Mathematical Models and Methods in Applied Sciences
Vol. 29, No. 14 (2019) 2719–2753
© The Author(s)
DOI: 10.1142/S0218202519500581

World Scientific
www.worldscientific.com

A node-numbering-invariant directional length scale for simplex elements

Kenji Takizawa

Department of Modern Mechanical Engineering,
Waseda University, 3-4-1 Ookubo,
Shinjuku-ku, Tokyo 169-8555, Japan
Kenji.Takizawa@tafsm.org

Yuki Ueda

Department of Applied Mathematics,
The Hong Kong Polytechnic University,
Hung Hom, Kowloon, Hong Kong

Tayfun E. Tezduyar

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

Faculty of Science and Engineering, Waseda University,
3-4-1 Ookubo, Shinjuku-ku, Tokyo 169-8555, Japan
tezduyar@tafsm.org

Received 29 August 2019
Accepted 3 October 2019
Published 1 November 2019
Communicated by N. Bellomo

Variational multiscale methods, and their precursors, stabilized methods, have been very popular in flow computations in the past several decades. Stabilization parameters embedded in most of these methods play a significant role. The parameters almost always involve element length scales, most of the time in specific directions, such as the direction of the flow or solution gradient. We require the length scales, including the directional length scales, to have node-numbering invariance for all element types, including simplex elements. We propose a length scale expression meeting that requirement. We analytically evaluate the expression in the context of simplex elements and compared to one of the most widely used length scale expressions and show the levels of noninvariance avoided.

Keywords: Stabilization parameter; simplex element; directional element length; node-numbering; invariance.

AMS Subject Classification: 65M60, 65N30

1. Introduction

In this paper, we introduce a node-numbering-invariant directional length scale for simplex elements 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 been playing a crucial role in computational flow analysis. Two of the earliest and most widely used stabilized methods are the incompressible-flow Streamline-Upwind/Petrov-Galerkin (SUPG)[1,2] and compressible-flow SUPG[3–5] methods. The incompressible-flow Pressure-Stabilizing/Petrov-Galerkin (PSPG) method,[6,7] with its Stokes-flow version introduced in Ref. 8, also has a long history and is 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 they are residual-based, they accomplish stabilization 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 shock or some other discontinuity, stabilized methods are often supplemented with a DC term. Supplementing the SUPG with a DC term goes back to 1986.[13,14] The DC term played a key role in the evolution of the compressible-flow SUPG, which was originally introduced in 1982 in the context of conservation variables. That 1982 method is now called "$(\text{SUPG})_{82}$". At first, $(\text{SUPG})_{82}$ was not used with any DC (shock-capturing) term, and the test computations clearly showed the need for additional stabilization at the shocks. Later, $(\text{SUPG})_{82}$ was recast in entropy variables, but also supplemented with a DC term.[15] This of course resulted in better shock profiles. In an ASME paper in 1991,[16] $(\text{SUPG})_{82}$ was supplemented with a very similar DC term. It was shown in Refs. 16 and 17 that, with the added DC term, $(\text{SUPG})_{82}$ was very comparable in accuracy to $(\text{SUPG})_{82}$ recast in entropy variables. The stabilized and DC methods introduced in Ref. 14 for the advection–diffusion–reaction equation accounted for the interaction between the DC and SUPG terms. Taking that interaction into account precludes "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 space–time computational methods*

Stabilized methods have been an important component of the space–time (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 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. 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 Refs. 22 and 23), 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[25–31] is the VMS version of the ALE. It succeeded the ST-SUPS and ALE-SUPS[32] 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,[33–35] "sliding interfaces"[36,37] and backflow stabilization.[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,[32] wind-turbine aerodynamics and FSI,[39–49] more specifically, vertical-axis wind turbines,[50,51,48,49] floating wind turbines,[52] wind turbines in atmospheric boundary layers,[53,47–49] and fatigue damage in wind-turbine blades,[54] patient-specific cardiovascular fluid mechanics and FSI,[55,25,56–60] 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 Ref. 81, for divergence-conforming discretizations of incompressible flows in Ref. 82, and for compressible flows with emphasis on gas-turbine modeling in Ref. 83.

The ST-SUPS and ST-VMS have also been applied to many classes of FSI, MBI and fluid mechanics problems (see Ref. 84 for a comprehensive summary). The classes of problems include spacecraft parachute analysis for the landing-stage parachutes,[85,28,86–88] cover-separation parachutes[89] and the drogue parachutes,[90–92] wind-turbine aerodynamics for horizontal-axis wind-turbine rotors,[39,93,94,28] full horizontal-axis wind-turbines[95–97,45] and vertical-axis wind-turbines,[98,48,49] flapping-wing aerodynamics for an actual locust,[99,100,28,101] bioinspired MAVs[102,103,96,97] and wing-clapping,[104,105] blood flow analysis of cerebral aneurysms,[106,96] stent-blocked aneurysms,[107,108,106] aortas[109–113] and heart valves,[104,97,114–117,111,113] spacecraft aerodynamics,[89,118] thermo-fluid analysis of ground vehicles and their tires,[24,115] thermo-fluid analysis of disk brakes,[119] flow-driven string dynamics in turbomachinery,[120–122] flow analysis of turbocharger turbines,[123–126] flow around tires with road contact and deformation,[115,127–130] fluid films,[131,130] ram-air parachutes,[132] and compressible-flow spacecraft parachute aerodynamics.[133,134]

1.4. *ST Slip Interface method*

The ST Slip Interface (ST-SI) method was introduced in Ref. 98, 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.[36,37] 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 Ref. 98. 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 Ref. 119 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,[98,48,49] thermo-fluid analysis of disk brakes,[119] flow-driven string dynamics in turbomachinery,[120–122] flow analysis of turbocharger turbines,[123–126] flow around tires with road contact and deformation,[115,127–130] fluid films,[131,130] aerodynamic analysis of ram-air parachutes,[132] and flow analysis of heart valves.[116,117,111,113]

In the ST-SI version presented in Ref. 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.[132] The compressible-flow ST-SI methods were introduced in Ref. 133, 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 Ref. 133. These, together with the compressible-flow ST SUPG method,[135] 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.[133,134]

1.5. *ST Isogeometric Analysis*

The success of using Isogeometric Analysis (IGA) basis functions in space[136,55,25,36] motivated the integration of the ST methods with isogeometric discretization, which we broadly call "ST-IGA". The ST-IGA was introduced in Ref. 22. Computations with the ST-VMS and ST-IGA were first reported in Ref. 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 getting full benefit out of using higher-order basis functions in space. This was demonstrated with the stability and accuracy analysis given in Ref. 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 Refs. 22 and 23 and demonstrated in Refs. 99, 100 and 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 (STNMUM)[99,100,102]; the name was given in Ref. 95. 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"[22,99,23,28] 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.[137,24,115,119–122] 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,[99,100,28,101] bioinspired MAVs[102,103,96,97] and wing-clapping,[104,105] separation aerodynamics of spacecraft,[89] aerodynamics of horizontal-axis[95–97,45] and vertical-axis[98,48,49] wind-turbines, thermo-fluid analysis of ground vehicles and their tires,[24,115] thermo-fluid analysis of disk brakes,[119] flow-driven string dynamics in turbomachinery,[120–122] and flow analysis of turbocharger turbines.[123–126]

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,[123–126] flow-driven string dynamics in turbomachinery,[121,122] ram-air parachutes,[132] spacecraft parachutes,[134] aortas,[111–113] heart valves,[116,117,111,113] tires with road contact and deformation,[128–130] and fluid films.[131,130] Using IGA basis functions in space is now a key part of some of the newest arterial zero-stress-state estimation methods[138–141,113] and related shell analysis.[142]

1.6. *Stabilization parameters and element lengths*

In all the stabilized and VMS methods discussed in the earlier subsections, an embedded stabilization parameter, known as "τ", plays a significant role (see Ref. 28). 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 Refs. 1–5, followed by the the ones introduced in Refs. 13 and 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 Refs. 3–5 in conjunction with $(\text{SUPG})_{82}$ is now called "τ_{82}". The τ definition introduced in Ref. 14, which is for the advective limit and is now called "τ_{SUGN1}" (and the corresponding element length is now called "h_{UGN}"), automatically yields lower values for higher-order finite element basis functions (see Refs. 143 and 144). Later, other τ definitions that are applicable to higher-order elements were proposed in Ref. 145 in the context of advective–diffusive systems. The τ used in Ref. 16 with $(\text{SUPG})_{82}$ was a slightly modified version of τ_{82}. Subsequent minor modifications of τ_{82} took into account the interaction between the DC and $(\text{SUPG})_{82}$ terms in a fashion similar to how it was done in Ref. 14 for the advection–diffusion–reaction equation. Until 2004, all these slightly modified versions of τ_{82} were always used with the same DC parameter, which was introduced in the 1991 ASME paper[16] and is now called "δ_{91}". This DC parameter was derived from the one given in Ref. 15 for the entropy variables.

Calculating the τs based on the element-level matrices and vectors was introduced in Ref. 146 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[146,147] to address the difficulties reported at small time-step sizes. A second element length scale, in the solution-gradient direction and called "h_{RGN}", was introduced in 2001 Refs. 20 and 148. Recognizing this as a diffusion length scale, a new stabilization parameter for the diffusive limit, "τ_{SUGN3}", was introduced in Refs. 20 and 149, to be used together with τ_{SUGN1} and "τ_{SUGN2}", the parameters for the advective and transient limits. For the stabilized ST methods, "τ_{SUGN12}", representing both the advective and transient limits, was also introduced in Ref. 20.

New ways of calculating the τ and DC parameter to be used with $(\text{SUPG})_{82}$ were introduced in 2004 Refs. 149, 150 and 151. The new τs, now categorized under the label "τ_{04}", 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

DC Directional Dissipation[20,151,152] parameter was defined, and one that is now called "YZβ" DC parameter. The YZβ DC parameter is residual based, and it is simpler than δ_{91}. It has options for smoother or sharper computed shocks. A number of 2D and 3D test computations with YZβ DC were reported in Refs. 153–155. These computations showed that in addition to being simpler than δ_{91}, the YZβ DC parameter was superior in accuracy. The computations reported in Refs. 153–155 were based on the compressible-flow ST SUPG.

Some new options for the stabilization parameters used with the SUPS and VMS were proposed in Refs. 21, 24, 94, 95 and 99. These include a fourth τ component, "τ_{SUGN4}",[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 Refs. 72, 156–166 and 167.

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 Ref. 168 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 Ref. 168 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. *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 we introduce here meets that requirement. We analytically evaluate that element length expression in the context of simplex elements and compared to one of the most widely used element length expressions and show the levels of noninvariance avoided.

1.9. *Outline of the remaining sections*

In Sec. 2, we briefly describe the directional element length definition introduced in Ref. 168, which we will start with. In Sec. 3, we describe the parametric spaces we use in calculating the directional element length for simplex elements: the integration parametric space, which is not node-numbering-invariant, and the preferred parametric space, which is. In Sec. 4, we show the level of noninvariance we get with the integration parametric space, and we compare the element lengths based on the integration and preferred parametric spaces. We provide our concluding remarks in Sec. 5.

2. Overview of the Starting Element Length Definition

We briefly describe the element length definition introduced in Ref. 168, which we will start with. Given $\mathbf{x} \in \mathbb{R}^{n_{\mathrm{sd}}}$ and $\boldsymbol{\xi} \in \mathbb{R}^{n_{\mathrm{pd}}}$, where n_{sd} and n_{pd} are the spatial and parametric dimensions, We define the Jacobian tensor as

$$\mathbf{Q} = \frac{\partial \mathbf{x}}{\partial \boldsymbol{\xi}} \tag{2.1}$$

and the matrix of its components for given basis sets $\mathbf{e}_i$ $(i = 1, \ldots, n_{\mathrm{sd}})$ and $\mathbf{e}_\alpha$ $(\alpha = 1, \ldots, n_{\mathrm{pd}})$ as

$$\mathsf{Q} = [\mathsf{Q}_{i\alpha}] = \left[\frac{\partial x_i}{\partial \xi_\alpha}\right] \in \mathbb{R}^{n_{\mathrm{sd}} \times n_{\mathrm{pd}}}. \tag{2.2}$$

We introduce a transformation tensor $\mathbf{D}$, and based on that we define the following tensor:

$$\hat{\mathbf{Q}} = \mathbf{Q} \cdot \mathbf{D}^{-1}. \tag{2.3}$$

With that, the directional element-length is defined as

$$\begin{align} \hat{h}(\mathbf{r}) &= 2\|\mathbf{r} \cdot \hat{\mathbf{Q}}^{-T}\|^{-1} \tag{2.4} \\ &= 2\|\mathbf{r} \cdot (\mathbf{Q}^{-T} \cdot \mathbf{D}^{T})\|^{-1} \tag{2.5} \\ &= 2\|\mathbf{D} \cdot \mathbf{Q}^{-1} \cdot \mathbf{r}\|^{-1}, \tag{2.6} \end{align}$$

where $\mathbf{r}$ is the unit vector representing the direction, defined in the physical space.

Remark 1. The factor 2 comes from a typical parametric space, which is of square or cubic shape with length 2. Other shapes and different lengths can be accounted for with $\mathbf{D}$.

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. Some of the most widely used element length definitions do not have this invariance. The transformation tensor $\mathbf{D}$ was introduced in Ref. 168 to avoid such invariance. It was identified as a "scaling" tensor in Ref. 168, but considering its nondiagonal nature, we now identify it as a "transformation" tensor.

Suppose, we have a preferred parametric space $\hat{\boldsymbol{\xi}}$ for calculating the element length. Then,

$$\hat{\mathbf{Q}} = \frac{\partial \mathbf{x}}{\partial \hat{\boldsymbol{\xi}}} = \frac{\partial \mathbf{x}}{\partial \boldsymbol{\xi}} \cdot \frac{\partial \boldsymbol{\xi}}{\partial \hat{\boldsymbol{\xi}}} \tag{2.7}$$

$$= \frac{\partial \mathbf{x}}{\partial \boldsymbol{\xi}} \cdot \mathbf{D}^{-1}. \tag{2.8}$$

That gives

$$\mathbf{D} \equiv \frac{\partial \hat{\boldsymbol{\xi}}}{\partial \boldsymbol{\xi}}, \tag{2.9}$$

and

$$\mathsf{D} = [\mathsf{D}_{\alpha\beta}] = \left[\frac{\partial \hat{\xi}_\alpha}{\partial \xi_\beta}\right] \in \mathbb{R}^{n_{\mathrm{pd}} \times n_{\mathrm{pd}}}. \tag{2.10}$$

We use a different basis set, $\hat{\mathbf{e}}_\alpha$, for the preferred parametric space.

3. Simplex Element Parametric Spaces

We consider n_{pd}-dimensional simplex elements ($n_{\mathrm{pd}} \geq 2$).

3.1. *Integration parametric space*

A typical integration parametric space is defined as

$$\left\{ \boldsymbol{\xi} \in \mathbb{R}^{n_{\mathrm{pd}}} \,|\, \xi_\alpha \geq 0 \text{ for } \alpha = 1, \ldots, n_{\mathrm{pd}} \text{ and } \sum_{\alpha=1}^{n_{\mathrm{pd}}} \xi_\alpha \leq 1 \right\}. \tag{3.1}$$

The βth component of the first n_{pd} vertex-position vectors is

$$\{\boldsymbol{\xi}_\alpha\}_\beta = \delta_{\alpha\beta}, \tag{3.2}$$

and the last-vertex position is

$$\boldsymbol{\xi}_{n_{\mathrm{pd}}+1} = \mathbf{0}. \tag{3.3}$$

Based on these, we define a set of linear shape functions:

$$N_\alpha(\boldsymbol{\xi}) = \boldsymbol{\xi} \cdot \mathbf{e}_\alpha, \tag{3.4}$$

for $\alpha = 1, \ldots n_{\mathrm{pd}}$, and

$$N_{n_{\mathrm{pd}}+1}(\boldsymbol{\xi}) = 1 - \sum_{\beta=1}^{n_{\mathrm{pd}}} \boldsymbol{\xi} \cdot \mathbf{e}_\beta. \tag{3.5}$$

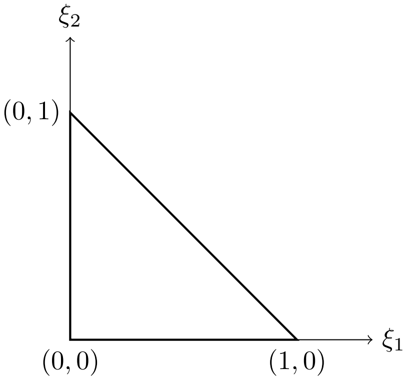

Fig. 1. Parametric space for a typical triangular element.

The derivatives are

$$\frac{\partial N_\alpha(\boldsymbol{\xi})}{\partial \xi_\beta} = \delta_{\alpha\beta}, \tag{3.6}$$

for $\beta = 1, \ldots, n_{\mathrm{pd}}$, and

$$\frac{\partial N_{n_{\mathrm{pd}}+1}(\boldsymbol{\xi})}{\partial \xi_\beta} = -\{\mathbf{1}\}_\beta, \tag{3.7}$$

where

$$\mathbf{1} = \begin{pmatrix} 1 \\ \vdots \\ 1 \end{pmatrix} \in \mathbb{R}^{n_{\mathrm{pd}}}. \tag{3.8}$$

Figure 1 shows a 2D example.

3.2. *Preferred parametric space*

The integration parametric space, as described in Sec. 3.1, is not ideal for calculating the stabilization parameters. For example, it does not have node-numbering invariance for the directional element length definition given by Eq. (2.6) without $\mathbf{D}$.

As the preferred parametric space, which we want to be free from the node-numbering order, we select a regular simplex. Here, the regular simplex has vertex distances of 2. The first n_{pd} vertex-position vectors are

$$\hat{\boldsymbol{\xi}}_\alpha = \sqrt{2}\hat{\mathbf{e}}_\alpha. \tag{3.9}$$

To make it a regular simplex, the last-vertex location will be determined following the steps below. We start with defining a centroid based on the first n_{pd} vertices:

$$\hat{\boldsymbol{\xi}}_{\mathrm{g}} = \frac{1}{n_{\mathrm{pd}}} \sum_{\beta=1}^{n_{\mathrm{pd}}} \hat{\boldsymbol{\xi}}_\beta, \tag{3.10}$$

and defining $\hat{\mathbf{1}} = \sum_{\beta=1}^{n_{\mathrm{pd}}} \hat{\mathbf{e}}_\beta$, we write

$$\hat{\boldsymbol{\xi}}_{\mathrm{g}} = \frac{\sqrt{2}}{n_{\mathrm{pd}}}\hat{\mathbf{1}}. \tag{3.11}$$

Because of the symmetry, the last vertex will be along that vector:

$$\hat{\boldsymbol{\xi}}_{n_{\mathrm{pd}}+1} = -d\hat{\boldsymbol{\xi}}_{\mathrm{g}}, \tag{3.12}$$

where d is a positive value. Next, we write the equations representing the distance constraints. For $\alpha = 1, \ldots, n_{\mathrm{pd}}$:

$$\left\|\hat{\boldsymbol{\xi}}_{n_{\mathrm{pd}}+1} - \hat{\boldsymbol{\xi}}_\alpha\right\| = 2. \tag{3.13}$$

From that, we obtain

$$4 = \left(\hat{\boldsymbol{\xi}}_{n_{\mathrm{pd}}+1} - \hat{\boldsymbol{\xi}}_\alpha\right) \cdot \left(\hat{\boldsymbol{\xi}}_{n_{\mathrm{pd}}+1} - \hat{\boldsymbol{\xi}}_\alpha\right) \tag{3.14}$$

$$= \left\|\hat{\boldsymbol{\xi}}_{n_{\mathrm{pd}}+1}\right\|^2 + 2 - 2\hat{\boldsymbol{\xi}}_\alpha \cdot \hat{\boldsymbol{\xi}}_{n_{\mathrm{pd}}+1}. \tag{3.15}$$

Substituting Eq. (3.12) into this, we obtain

$$2 = d^2\left\|\hat{\boldsymbol{\xi}}_{\mathrm{g}}\right\|^2 + 2d\hat{\boldsymbol{\xi}}_\alpha \cdot \hat{\boldsymbol{\xi}}_{\mathrm{g}}. \tag{3.16}$$

From Eq. (3.11),

$$\left\|\hat{\boldsymbol{\xi}}_{\mathrm{g}}\right\|^2 = \frac{2}{n_{\mathrm{pd}}}. \tag{3.17}$$

Substituting Eqs. (3.9), (3.11) and (3.17) into Eq. (3.16), we obtain

$$2 = \frac{2d^2}{n_{\mathrm{pd}}} + 2d\sqrt{2}\hat{\mathbf{e}}_\alpha \cdot \frac{\sqrt{2}}{n_{\mathrm{pd}}}\hat{\mathbf{1}} \tag{3.18}$$

$$= \frac{2d^2}{n_{\mathrm{pd}}} + \frac{4d}{n_{\mathrm{pd}}}, \tag{3.19}$$

which gives

$$d^2 + 2d - n_{\mathrm{pd}} = 0, \tag{3.20}$$

and the solution is

$$d = -1 \pm \sqrt{1 + n_{\mathrm{pd}}}. \tag{3.21}$$

Because $d > 0$, we select

$$d = \sqrt{1 + n_{\mathrm{pd}}} - 1. \tag{3.22}$$

With that, the last vertex-position vector is

$$\hat{\boldsymbol{\xi}}_{n_{\mathrm{pd}}+1} = \frac{\sqrt{2}}{n_{\mathrm{pd}}}\left(1 - \sqrt{1 + n_{\mathrm{pd}}}\right)\hat{\mathbf{1}}. \tag{3.23}$$

Figure 2 is the 2D case.

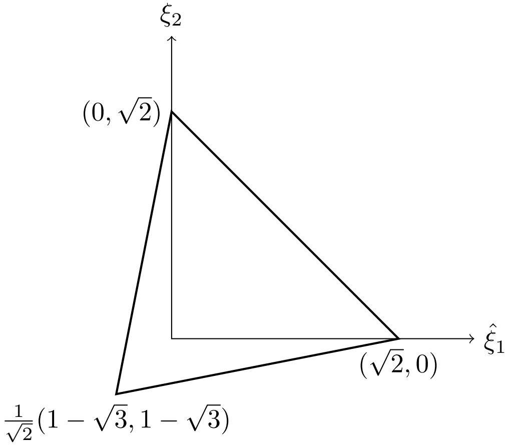

Fig. 2. A triangular element suitable for directional element length calculation.

From the vertex-position vectors, we obtain

$$\frac{\partial \hat{\xi}_\alpha}{\partial \xi_\beta} = \sum_{a=1}^{n_{\mathrm{pd}}+1} \frac{\partial N_a}{\partial \xi_\beta} \big\{\hat{\boldsymbol{\xi}}_a\big\}_\alpha \tag{3.24}$$

$$= \sum_{\gamma=1}^{n_{\mathrm{pd}}} \delta_{\gamma\beta} \big\{\hat{\boldsymbol{\xi}}_\gamma\big\}_\alpha + \frac{\partial N_{n_{\mathrm{pd}}+1}}{\partial \xi_\beta} \big\{\hat{\boldsymbol{\xi}}_{n_{\mathrm{pd}}+1}\big\}_\alpha \tag{3.25}$$

$$= \sqrt{2} \sum_{\gamma=1}^{n_{\mathrm{pd}}} \delta_{\gamma\beta}\delta_{\gamma\alpha} - \frac{\sqrt{2}}{n_{\mathrm{pd}}} \big(1 - \sqrt{1+n_{\mathrm{pd}}}\big) \{\hat{\mathbf{1}}\}_\beta \{\hat{\mathbf{1}}\}_\alpha \tag{3.26}$$

$$= \frac{\sqrt{2}}{n_{\mathrm{pd}}} \big(n_{\mathrm{pd}}\delta_{\alpha\beta} - \big(1 - \sqrt{1+n_{\mathrm{pd}}}\big) \{\hat{\mathbf{1}}\}_\alpha \{\hat{\mathbf{1}}\}_\beta\big). \tag{3.27}$$

While the tensor $\mathbf{D}$ depends on the orientation of the parametric space $\hat{\boldsymbol{\xi}}$, the following product does not:

$$\frac{\partial \hat{\xi}_\gamma}{\partial \xi_\alpha} \frac{\partial \hat{\xi}_\gamma}{\partial \xi_\beta} = 2 \left(\delta_{\gamma\alpha} - \frac{1 - \sqrt{1+n_{\mathrm{pd}}}}{n_{\mathrm{pd}}} \{\hat{\mathbf{1}}\}_\gamma \{\hat{\mathbf{1}}\}_\alpha\right)$$

$$\times \left(\delta_{\gamma\beta} - \frac{1 - \sqrt{1+n_{\mathrm{pd}}}}{n_{\mathrm{pd}}} \{\hat{\mathbf{1}}\}_\gamma \{\hat{\mathbf{1}}\}_\beta\right) \tag{3.28}$$

$$= 2 \left(\delta_{\gamma\alpha}\delta_{\gamma\beta} - \frac{1 - \sqrt{1+n_{\mathrm{pd}}}}{n_{\mathrm{pd}}} \left(\delta_{\gamma\alpha}\{\hat{\mathbf{1}}\}_\beta + \delta_{\gamma\beta}\{\hat{\mathbf{1}}\}_\alpha\right) \{\hat{\mathbf{1}}\}_\gamma \right.$$

$$\left. + n_{\mathrm{pd}} \left(\frac{1 - \sqrt{1+n_{\mathrm{pd}}}}{n_{\mathrm{pd}}}\right)^2 \{\hat{\mathbf{1}}\}_\alpha \{\hat{\mathbf{1}}\}_\beta \right) \tag{3.29}$$

$$= 2\left(\delta_{\alpha\beta} - 2\frac{1-\sqrt{1+n_{\rm pd}}}{n_{\rm pd}}\{\hat{\mathbf{1}}\}_\alpha\{\hat{\mathbf{1}}\}_\beta \right.$$

$$\left. + \frac{2+n_{\rm pd}-2\sqrt{1+n_{\rm pd}}}{n_{\rm pd}}\{\hat{\mathbf{1}}\}_\alpha\{\hat{\mathbf{1}}\}_\beta\right) \tag{3.30}$$

$$= 2\left(\delta_{\alpha\beta} + \{\hat{\mathbf{1}}\}_\alpha\{\hat{\mathbf{1}}\}_\beta\right). \tag{3.31}$$

From that,

$$\{\mathsf{D}^T\mathsf{D}\}_{\alpha\beta} = 2\big(\delta_{\alpha\beta} + \{\hat{\mathbf{1}}\}_\alpha\{\hat{\mathbf{1}}\}_\beta\big). \tag{3.32}$$

For $n_{\rm pd} = 2$, we obtain

$$\mathsf{D} = \frac{\sqrt{2}}{2}\begin{bmatrix}\sqrt{3}+1 & \sqrt{3}-1\\ \sqrt{3}-1 & \sqrt{3}+1\end{bmatrix}, \tag{3.33}$$

$$\mathsf{D}^T\mathsf{D} = 2\begin{bmatrix}2 & 1\\ 1 & 2\end{bmatrix}, \tag{3.34}$$

for $n_{\rm pd} = 3$,

$$\mathsf{D} = \frac{\sqrt{2}}{3}\begin{bmatrix}4 & 1 & 1\\ 1 & 4 & 1\\ 1 & 1 & 4\end{bmatrix}, \tag{3.35}$$

$$\mathsf{D}^T\mathsf{D} = 2\begin{bmatrix}2 & 1 & 1\\ 1 & 2 & 1\\ 1 & 1 & 2\end{bmatrix}, \tag{3.36}$$

and for $n_{\rm pd} = 4$ (ST element),

$$\mathsf{D} = \frac{\sqrt{2}}{4}\begin{bmatrix}\sqrt{5}+3 & \sqrt{5}-1 & \sqrt{5}-1 & \sqrt{5}-1\\ \sqrt{5}-1 & \sqrt{5}+3 & \sqrt{5}-1 & \sqrt{5}-1\\ \sqrt{5}-1 & \sqrt{5}-1 & \sqrt{5}+3 & \sqrt{5}-1\\ \sqrt{5}-1 & \sqrt{5}-1 & \sqrt{5}-1 & \sqrt{5}+3\end{bmatrix}, \tag{3.37}$$

$$\mathsf{D}^T\mathsf{D} = 2\begin{bmatrix}2 & 1 & 1 & 1\\ 1 & 2 & 1 & 1\\ 1 & 1 & 2 & 1\\ 1 & 1 & 1 & 2\end{bmatrix}. \tag{3.38}$$

4. Analysis of the Element Lengths

4.1. *Ratio of element lengths with and without D*

From Eq. (2.6), the element length definition without **D**:

$$h(\mathbf{r}) = 2\|\mathbf{Q}^{-1}\cdot\mathbf{r}\|^{-1}. \tag{4.1}$$

We define two vectors:

$$\mathbf{z} = \mathbf{Q}^{-1} \cdot \mathbf{r}, \tag{4.2}$$

$$\begin{aligned} \hat{\mathbf{z}} &= \hat{\mathbf{Q}}^{-1} \cdot \mathbf{r} && (4.3) \\ &= \mathbf{D} \cdot \mathbf{Q}^{-1} \cdot \mathbf{r} && (4.4) \\ &= \mathbf{D} \cdot \mathbf{z}. && (4.5) \end{aligned}$$

They are the vectors we get when we bring the vector $\mathbf{r}$ from the physical space to the two parametric spaces. With that, we have

$$\begin{aligned} \hat{h}(\mathbf{r}) &= \frac{2}{\|\mathbf{D} \cdot \mathbf{z}\|} && (4.6) \\ &= \frac{2}{\|\hat{\mathbf{z}}\|}, && (4.7) \\ h(\mathbf{r}) &= \frac{2}{\|\mathbf{z}\|} && (4.8) \\ &= \frac{2}{\|\mathbf{D}^{-1} \cdot \hat{\mathbf{z}}\|}. && (4.9) \end{aligned}$$

From Eqs. (4.6) and (4.8), we obtain

$$\frac{h(\mathbf{r})}{\hat{h}(\mathbf{r})} = \frac{\|\mathbf{D} \cdot \mathbf{z}\|}{\|\mathbf{z}\|} \leq \|\mathbf{D}\|_2, \tag{4.10}$$

where $\| \bullet \|_2$ is the spectral norm. From Eqs. (4.7) and (4.9), we obtain

$$\frac{h(\mathbf{r})}{\hat{h}(\mathbf{r})} = \frac{\|\hat{\mathbf{z}}\|}{\|\mathbf{D}^{-1} \cdot \hat{\mathbf{z}}\|} \geq \|\mathbf{D}^{-1}\|_2^{-1}. \tag{4.11}$$

Combining Eqs. (4.10) and (4.11), we obtain

$$\|\mathbf{D}^{-1}\|_2^{-1} \leq \frac{h(\mathbf{r})}{\hat{h}(\mathbf{r})} \leq \|\mathbf{D}\|_2, \tag{4.12}$$

while noting that we will always have $\left\|\mathbf{D}^{-1}\right\|_2^{-1} \leq \|\mathbf{D}\|_2$. For the Cartesian basis sets, the matrix form of that is

$$\|\mathsf{D}^{-1}\|_2^{-1} \leq \frac{h(\mathbf{r})}{\hat{h}(\mathbf{r})} \leq \|\mathsf{D}\|_2. \tag{4.13}$$

4.2. *Dependence on the node-numbering order*

Without $\mathbf{D}$, the element length depends on the node-numbering order because the last vertex $n_{\mathrm{pd}} + 1$ is at an irregular location, and that is the only reason. Therefore, the element length depends only on which node corresponds to the last vertex in the parametric space. With the affine transformation

$$\tilde{\boldsymbol{\xi}} = \mathbf{R} \cdot \boldsymbol{\xi} + \mathbf{b}, \tag{4.14}$$

where

$$\mathbf{R} = \sum_{a=1}^{n_{\rm pd}} (\boldsymbol{\xi}_{a+1} - \boldsymbol{\xi}_1)\boldsymbol{\xi}_a, \tag{4.15}$$

$$\mathbf{b} = \boldsymbol{\xi}_1, \tag{4.16}$$

we can have all the configurations by applying this transformation 1 to $n_{\rm pd}$ times. Applying it $n_{\rm pd} + 1$ times takes us back to the configuration prior to any transformation. The element length with $\mathbf{R}$ is

$$\tilde{h}(\mathbf{r}) = 2\|\mathbf{R} \cdot \mathbf{Q}^{-1} \cdot \mathbf{r}\|^{-1}. \tag{4.17}$$

We compare that to the length without $\mathbf{R}$. From Eq. (4.12),

$$\|\mathbf{R}^{-1}\|_2^{-1} \le \frac{h(\mathbf{r})}{\tilde{h}(\mathbf{r})} \le \|\mathbf{R}\|_2. \tag{4.18}$$

First, we prove that $\mathbf{R}^k$ can be expressed as

$$\mathbf{R}^k = \sum_{a=1}^{n_{\rm pd}+1} (\boldsymbol{\xi}_{m(a+k)} - \boldsymbol{\xi}_{m(k)})\boldsymbol{\xi}_a, \tag{4.19}$$

where $m(a) = 1 + ((a-1) \bmod (n_{\rm pd} + 1))$. We note that, to deal with the algebra in the proof in a more systematic way, we have extended the summation range to $n_{\rm pd} + 1$, which has no actual effect because $\boldsymbol{\xi}_{n_{\rm pd}+1} = \mathbf{0}$.

Proof. Let us define

$$\mathbf{S}_k = \sum_{a=1}^{n_{\rm pd}+1} (\boldsymbol{\xi}_{m(a+k)} - \boldsymbol{\xi}_{m(k)})\boldsymbol{\xi}_a. \tag{4.20}$$

Obviously,

$$\mathbf{R} = \mathbf{S}_1. \tag{4.21}$$

We now show that

$$\mathbf{R} \cdot \mathbf{S}_k = \mathbf{S}_{k+1}. \tag{4.22}$$

From Eqs. (4.20) and (4.21),

$$\mathbf{R} \cdot \mathbf{S}_k = \sum_{a=1}^{n_{\rm pd}+1} (\boldsymbol{\xi}_{m(a+1)} - \boldsymbol{\xi}_1)\boldsymbol{\xi}_a \cdot \sum_{b=1}^{n_{\rm pd}+1} (\boldsymbol{\xi}_{m(b+k)} - \boldsymbol{\xi}_{m(k)})\boldsymbol{\xi}_b \tag{4.23}$$

$$= \sum_{a=1}^{n_{\rm pd}+1} \sum_{b=1}^{n_{\rm pd}+1} (\boldsymbol{\xi}_a \cdot (\boldsymbol{\xi}_{m(b+k)} - \boldsymbol{\xi}_{m(k)}))(\boldsymbol{\xi}_{m(a+1)} - \boldsymbol{\xi}_1)\boldsymbol{\xi}_b \tag{4.24}$$

$$= \sum_{a=1}^{n_{\mathrm{pd}}+1} \sum_{b=1}^{n_{\mathrm{pd}}+1} (\boldsymbol{\xi}_a \cdot \boldsymbol{\xi}_{m(b+k)})(\boldsymbol{\xi}_{m(a+1)} - \boldsymbol{\xi}_1)\boldsymbol{\xi}_b$$

$$- \sum_{a=1}^{n_{\mathrm{pd}}+1} \sum_{b=1}^{n_{\mathrm{pd}}+1} (\boldsymbol{\xi}_a \cdot \boldsymbol{\xi}_{m(k)})(\boldsymbol{\xi}_{m(a+1)} - \boldsymbol{\xi}_1)\boldsymbol{\xi}_b \tag{4.25}$$

$$= \sum_{b=1}^{n_{\mathrm{pd}}+1} \sum_{a=b+k}^{n_{\mathrm{pd}}+b+k} \underbrace{(\boldsymbol{\xi}_{m(a)} \cdot \boldsymbol{\xi}_{m(b+k)})(\boldsymbol{\xi}_{m(a+1)} - \boldsymbol{\xi}_1)}_{=\delta_{m(a)\,m(b+k)}(\boldsymbol{\xi}_{m(a+1)} - \boldsymbol{\xi}_1)}\boldsymbol{\xi}_b$$

$$- \sum_{b=1}^{n_{\mathrm{pd}}+1} \sum_{a=k}^{n_{\mathrm{pd}}+k} \underbrace{(\boldsymbol{\xi}_{m(a)} \cdot \boldsymbol{\xi}_{m(k)})(\boldsymbol{\xi}_{m(a+1)} - \boldsymbol{\xi}_1)}_{=\delta_{m(a)\,m(k)}(\boldsymbol{\xi}_{m(a+1)} - \boldsymbol{\xi}_1)}\boldsymbol{\xi}_b \tag{4.26}$$

$$= \sum_{b=1}^{n_{\mathrm{pd}}+1} (\boldsymbol{\xi}_{m(b+k+1)} - \boldsymbol{\xi}_1)\boldsymbol{\xi}_b - \sum_{b=1}^{n_{\mathrm{pd}}+1} (\boldsymbol{\xi}_{m(k+1)} - \boldsymbol{\xi}_1)\boldsymbol{\xi}_b \tag{4.27}$$

$$= \sum_{a=1}^{n_{\mathrm{pd}}+1} (\boldsymbol{\xi}_{m(a+k+1)} - \boldsymbol{\xi}_{m(k+1)})\boldsymbol{\xi}_a \tag{4.28}$$

$$= \mathbf{S}_{k+1}. \tag{4.29}$$

Actually, the expressions $\delta_{m(a)\,m(b+k)}$ and $\delta_{m(a)\,m(k)}$ cannot represent $\boldsymbol{\xi}_{n_{\mathrm{pd}}+1} \cdot \boldsymbol{\xi}_{n_{\mathrm{pd}}+1}$, however, for that a, $\boldsymbol{\xi}_{m(a+1)} - \boldsymbol{\xi}_1 = \mathbf{0}$. From Eqs. (4.21) and (4.22),

$$\mathbf{R}^k = \mathbf{S}_k. \tag{4.30}$$

□

We note that

$$\mathbf{R}^k = \mathbf{R}^{m(k)}. \tag{4.31}$$

To evaluate the spectral norm of $\mathbf{R}^k$, we first evaluate $(\mathbf{R}^k)^T \cdot (\mathbf{R}^k)$:

$$(\mathbf{R}^k)^T \cdot (\mathbf{R}^k) = \sum_{a=1}^{n_{\mathrm{pd}}+1} \boldsymbol{\xi}_a(\boldsymbol{\xi}_{m(a+k)} - \boldsymbol{\xi}_{m(k)}) \cdot \sum_{b=1}^{n_{\mathrm{pd}}+1} (\boldsymbol{\xi}_{m(b+k)} - \boldsymbol{\xi}_{m(k)})\boldsymbol{\xi}_b \tag{4.32}$$

$$= \sum_{a=1}^{n_{\mathrm{pd}}+1} \sum_{b=1}^{n_{\mathrm{pd}}+1} ((\boldsymbol{\xi}_{m(a+k)} - \boldsymbol{\xi}_{m(k)}) \cdot (\boldsymbol{\xi}_{m(b+k)} - \boldsymbol{\xi}_{m(k)}))\boldsymbol{\xi}_a\boldsymbol{\xi}_b \tag{4.33}$$

$$= \sum_{a=1}^{n_{\mathrm{pd}}+1} \sum_{b=1}^{n_{\mathrm{pd}}+1} (\boldsymbol{\xi}_{m(a+k)} \cdot \boldsymbol{\xi}_{m(b+k)} - \boldsymbol{\xi}_{m(k)} \cdot \boldsymbol{\xi}_{m(a+k)}$$

$$- \boldsymbol{\xi}_{m(b+k)} \cdot \boldsymbol{\xi}_{m(k)} + \boldsymbol{\xi}_{m(k)} \cdot \boldsymbol{\xi}_{m(k)})\boldsymbol{\xi}_a\boldsymbol{\xi}_b \tag{4.34}$$

$$= \sum_{a=1}^{n_{\mathrm{pd}}+1} \sum_{b=1}^{n_{\mathrm{pd}}+1} \delta_{m(a+k)\,m(b+k)}\boldsymbol{\xi}_a\boldsymbol{\xi}_b - \boldsymbol{\xi}_{m(-k)}\boldsymbol{\xi}_{m(-k)}$$

$$\underbrace{- \boldsymbol{\xi}_{m(k)} \cdot \sum_{a=1}^{n_{\mathrm{pd}}+1} \boldsymbol{\xi}_{m(a+k)}\boldsymbol{\xi}_a}_{=\mathbf{0}} \mathbf{1}$$

$$- \mathbf{1} \underbrace{\boldsymbol{\xi}_{m(k)} \cdot \sum_{b=1}^{n_{\mathrm{pd}}+1} \boldsymbol{\xi}_{m(b+k)}\boldsymbol{\xi}_b}_{=\mathbf{0}} + \boldsymbol{\xi}_{m(k)} \cdot \boldsymbol{\xi}_{m(k)} \mathbf{1}\mathbf{1} \tag{4.35}$$

$$= \mathbf{I} - \boldsymbol{\xi}_{m(-k)}\boldsymbol{\xi}_{m(-k)} + \boldsymbol{\xi}_{m(k)} \cdot \boldsymbol{\xi}_{m(k)}\mathbf{1}\mathbf{1}. \tag{4.36}$$

Here, $\mathbf{I}$ is the unit tensor, and $\mathbf{1} = \sum_{\alpha=1}^{n_{\mathrm{pd}}} \mathbf{e}_\alpha$. We preclude $m(k) = n_{\mathrm{pd}} + 1$ because that would take us back to the starting parametric space. With that,

$$(\mathbf{R}^k)^T \cdot (\mathbf{R}^k) = \mathbf{I} - \boldsymbol{\xi}_{m(-k)}\boldsymbol{\xi}_{m(-k)} + \mathbf{1}\mathbf{1}. \tag{4.37}$$

In the matrix version of that, all entries are 1, excluding the diagonal entries, where the value is 2, but not excluding the $m(-k)$th column. For example, for $k = n_{\mathrm{pd}}$, we get

$$(\mathsf{R}^{n_{\mathrm{pd}}})^T(\mathsf{R}^{n_{\mathrm{pd}}}) = \begin{pmatrix} 1 & 1 & \cdots & 1 & 1 \\ 1 & 2 & & 1 & 1 \\ \vdots & & \ddots & & \vdots \\ 1 & 1 & & 2 & 1 \\ 1 & 1 & \cdots & 1 & 2 \end{pmatrix} \in \mathbb{R}^{n_{\mathrm{pd}} \times n_{\mathrm{pd}}}. \tag{4.38}$$

From this example, we can see that the eigenvalues of $(\mathsf{R}^k)^T\mathsf{R}^k$ are independent of the k value, which implies $\|\mathsf{R}\|_2 = \|\mathsf{R}^2\|_2 = \cdots = \|\mathsf{R}^{n_{\mathrm{pd}}}\|_2$. Therefore it will be sufficient to inspect the eigenvalues of $(\mathsf{R}^{n_{\mathrm{pd}}})^T\mathsf{R}^{n_{\mathrm{pd}}}$. From Eq. (4.31), $\mathsf{R}^{n_{\mathrm{pd}}} = \mathsf{R}^{-1}$. With that, for notation convenience, we use the characteristic polynomial from $\det(\lambda\mathsf{I} - \mathsf{R}^{-T}\mathsf{R}^{-1})$, where I is the unit matrix. In deriving that polynomial, we use the special method given in Appendix A, specifically we use c_n (Eq. (A.22)) with $p = \lambda - 2$, $q = -1$ and $r = \lambda - 1$, obtaining

$$\det(\lambda\mathsf{I} - \mathsf{R}^{-T}\mathsf{R}^{-1}) = (\lambda - 1)^{(n_{\mathrm{pd}}-2)}(\lambda(\lambda - 1) - (n_{\mathrm{pd}}\lambda - 1)) \tag{4.39}$$

$$= (\lambda - 1)^{(n_{\mathrm{pd}}-2)}(\lambda^2 - (n_{\mathrm{pd}} + 1)\lambda + 1). \tag{4.40}$$

From that, the eigenvalues are

$$\lambda = \frac{n_{\mathrm{pd}} + 1 \pm \sqrt{(n_{\mathrm{pd}} + 1)^2 - 4}}{2}, \tag{4.41}$$

and for $n_{\mathrm{pd}} \geq 3$, $\lambda = 1$ is also an eigenvalue with multiplicity of $n_{\mathrm{pd}} - 2$. The maximum eigenvalue is

$$\lambda_{\max} = \frac{n_{\mathrm{pd}} + 1 + \sqrt{(n_{\mathrm{pd}} + 1)^2 - 4}}{2} \tag{4.42}$$

$$= \frac{n_{\mathrm{pd}} + 1 + \sqrt{(n_{\mathrm{pd}} - 1)(n_{\mathrm{pd}} + 3)}}{2}. \tag{4.43}$$

This means that

$$\|\mathbf{R}^k\|_2 = \sqrt{\frac{n_{\mathrm{pd}} + 1 + \sqrt{(n_{\mathrm{pd}} - 1)(n_{\mathrm{pd}} + 3)}}{2}} \tag{4.44}$$

$$= \frac{\sqrt{n_{\mathrm{pd}} - 1} + \sqrt{n_{\mathrm{pd}} + 3}}{2}. \tag{4.45}$$

The eigenvector corresponding to $\lambda_{\max}$, $\mathbf{v} = \mathsf{v}_\alpha \mathbf{e}_\alpha$, will be calculated using the matrix

$$\lambda_{\max}\mathsf{I} - \mathsf{R}^T\mathsf{R} = \begin{pmatrix} \lambda_{\max} - 2 & -1 & \cdots & -1 & -1 \\ -1 & \lambda_{\max} - 2 & & -1 & -1 \\ \vdots & & \ddots & & \vdots \\ -1 & -1 & & \lambda_{\max} - 2 & -1 \\ -1 & -1 & \cdots & -1 & \lambda_{\max} - 1 \end{pmatrix} \in \mathbb{R}^{n_{\mathrm{pd}} \times n_{\mathrm{pd}}}, \tag{4.46}$$

and it can be verified that

$$\mathsf{v}_\alpha = \begin{cases} n_{\mathrm{pd}} - 1 & (\text{for } \alpha = m(-k)) \\ \lambda_{\max} - 1 & (\text{otherwise}) \end{cases} \tag{4.47}$$

$$= \begin{cases} n_{\mathrm{pd}} - 1 & (\text{for } \alpha = m(-k)), \\ \dfrac{n_{\mathrm{pd}} - 1 + \sqrt{(n_{\mathrm{pd}} - 1)(n_{\mathrm{pd}} + 3)}}{2} & (\text{otherwise}). \end{cases} \tag{4.48}$$

From Eq. (4.2),

$$\mathbf{r} = \frac{\mathbf{Q} \cdot \mathbf{v}}{\|\mathbf{Q} \cdot \mathbf{v}\|}, \tag{4.49}$$

and in that direction,

$$\frac{h}{\tilde{h}} = \frac{\sqrt{n_{\mathrm{pd}} - 1} + \sqrt{n_{\mathrm{pd}} + 3}}{2}. \tag{4.50}$$

See Fig. 3 for the case with $n_{\mathrm{pd}} = 2$.

From Eq. (4.45),

$$\|\mathbf{R}^{-1}\|_2^{-1} = \frac{2}{\sqrt{n_{\mathrm{pd}} - 1} + \sqrt{n_{\mathrm{pd}} + 3}} \tag{4.51}$$

$$= \frac{\sqrt{n_{\mathrm{pd}} + 3} - \sqrt{n_{\mathrm{pd}} - 1}}{2}, \tag{4.52}$$

and from Eq. (4.18),

$$\frac{\sqrt{n_{\mathrm{pd}} + 3} - \sqrt{n_{\mathrm{pd}} - 1}}{2} \leq \frac{h(\mathbf{r})}{\tilde{h}(\mathbf{r})} \leq \frac{\sqrt{n_{\mathrm{pd}} - 1} + \sqrt{n_{\mathrm{pd}} + 3}}{2}. \tag{4.53}$$

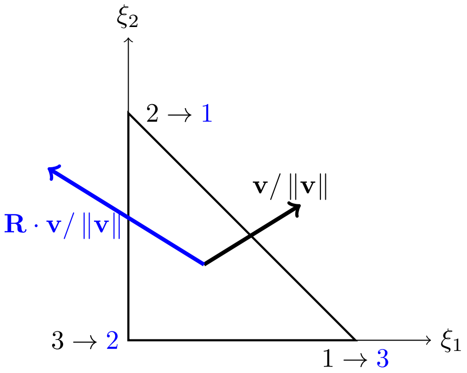
ξ_2
$2 \to 1$
$\mathbf{v}/\|\mathbf{v}\|$
$\mathbf{R}\cdot\mathbf{v}/\|\mathbf{v}\|$
$3 \to 2$
$1 \to 3$
ξ_1

Fig. 3. (Color online) An example of the different parametric spaces (black and blue). The directions of maximum $h/\tilde{h}$ for the black and blue parametric spaces, expressed as $\mathbf{v}/\|\mathbf{v}\|$ and $\mathbf{R}\cdot\mathbf{v}/\|\mathbf{v}\|$.

For $n_{\mathrm{pd}} = 2$,

$$\frac{\sqrt{5}-1}{2} \leq \frac{h(\mathbf{r})}{\tilde{h}(\mathbf{r})} \leq \frac{1+\sqrt{5}}{2} \approx 1.6, \tag{4.54}$$

for $n_{\mathrm{pd}} = 3$,

$$\frac{\sqrt{6}-\sqrt{2}}{2} \leq \frac{h(\mathbf{r})}{\tilde{h}(\mathbf{r})} \leq \frac{\sqrt{2}+\sqrt{6}}{2} \approx 1.9, \tag{4.55}$$

and for $n_{\mathrm{pd}} = 4$,

$$\frac{\sqrt{7}-\sqrt{3}}{2} \leq \frac{h(\mathbf{r})}{\tilde{h}(\mathbf{r})} \leq \frac{\sqrt{3}+\sqrt{7}}{2} \approx 2.2. \tag{4.56}$$

The range expands at both ends with increasing parametric-space dimension.

4.3. *Comparison of the element lengths based on the integration and preferred parametric spaces*

Here, we compare h and $\hat{h}$. From Eq. (3.32),

$$\mathsf{D}^T\mathsf{D} = 2\begin{pmatrix} 2 & 1 & \dots & 1 & 1 \\ 1 & 2 & & 1 & 1 \\ \vdots & & \ddots & & \vdots \\ 1 & 1 & & 2 & 1 \\ 1 & 1 & \dots & 1 & 2 \end{pmatrix} \in \mathbb{R}^{n_{\mathrm{pd}}\times n_{\mathrm{pd}}}. \tag{4.57}$$

In deriving the characteristic polynomial from $\det(\lambda\mathsf{I} - \mathsf{D}^T\mathsf{D})$, we use the special method given in Appendix A, specifically we use a_n (Eq. (A.17)) with $p = \lambda - 4$ and $q = -2$, obtaining

$$\det(\lambda\mathsf{I} - \mathsf{D}^T\mathsf{D}) = (\lambda-2)^{n_{\mathrm{pd}}-1}(\lambda - 2(1+n_{\mathrm{pd}})). \tag{4.58}$$

The maximum eigenvalue is $\lambda_{\max} = 2(1 + n_{\text{pd}})$, and

$$\|\mathbf{D}\|_2 = \sqrt{2(1 + n_{\text{pd}})}. \tag{4.59}$$

The eigenvector corresponding to $\lambda_{\max}$, $\mathbf{v} = \mathsf{v}_\alpha \mathbf{e}_\alpha$, will be calculated using the matrix

$$\lambda_{\max}\mathsf{I} - \mathsf{D}^T\mathsf{D} = \begin{pmatrix} 2n_{\text{pd}} - 2 & -2 & \dots & -2 & -2 \\ -2 & 2n_{\text{pd}} - 2 & & -2 & -2 \\ \vdots & & \ddots & & \vdots \\ -2 & -2 & & 2n_{\text{pd}} - 2 & -2 \\ -2 & -2 & \dots & -2 & 2n_{\text{pd}} - 2 \end{pmatrix} \in \mathbb{R}^{n_{\text{pd}} \times n_{\text{pd}}}, \tag{4.60}$$

and it can be verified that

$$\mathsf{v} = \mathbf{1}. \tag{4.61}$$

From Eq. (4.2),

$$\mathbf{r} = \frac{\mathbf{Q} \cdot \mathbf{1}}{\|\mathbf{Q} \cdot \mathbf{1}\|}, \tag{4.62}$$

and in that direction,

$$\frac{h}{\hat{h}} = \sqrt{2(1 + n_{\text{pd}})}. \tag{4.63}$$

Because D is symmetric, $\mathsf{D}^{-T}\mathsf{D}^{-1} = (\mathsf{D}^T\mathsf{D})^{-1}$. The eigenvalues of $(\mathsf{D}^T\mathsf{D})^{-1}$ are the inverses of the eigenvalues of $\mathsf{D}^T\mathsf{D}$, and the corresponding eigenvectors are the same. Therefore, the spectral norm of D^{-1} is the inverse of the square root of the minimum eigenvalue of $\mathsf{D}^T\mathsf{D}$, $\lambda_{\min} = 2$, with multiplicity $n_{\text{pd}} - 1$, and the span of the eigenvectors associated with that is $\mathbb{R}^{n_{\text{pd}}-1}$, which is orthogonal to $\mathbf{1}$. Consequently, $\|\mathsf{D}^{-1}\|_2^{-1} = \sqrt{2}$, and from Eq. (4.3),

$$\mathbf{r} = \frac{\hat{\mathbf{Q}} \cdot \left(\mathbf{I} - \frac{1}{n_{\text{pd}}}\mathbf{1}\mathbf{1}\right) \cdot \mathbf{y}}{\left\|\hat{\mathbf{Q}} \cdot \left(\mathbf{I} - \frac{1}{n_{\text{pd}}}\mathbf{1}\mathbf{1}\right) \cdot \mathbf{y}\right\|}, \tag{4.64}$$

$\forall\, \mathbf{y}$ that gives $\mathbf{r} \neq \mathbf{0}$, and in that direction,

$$\frac{h}{\hat{h}} = \sqrt{2}. \tag{4.65}$$

In summary,

$$\sqrt{2} \leq \frac{h(\mathbf{r})}{\hat{h}(\mathbf{r})} \leq \sqrt{2(1 + n_{\text{pd}})}. \tag{4.66}$$

For $n_{\text{pd}} = 2$,

$$\sqrt{2} \leq \frac{h(\mathbf{r})}{\hat{h}(\mathbf{r})} \leq \sqrt{6} \approx 2.4, \tag{4.67}$$

for $n_{\mathrm{pd}} = 3$,

$$\sqrt{2} \leq \frac{h(\mathbf{r})}{\hat{h}(\mathbf{r})} \leq 2\sqrt{2} \approx 2.8, \tag{4.68}$$

and for $n_{\mathrm{pd}} = 4$,

$$\sqrt{2} \leq \frac{h(\mathbf{r})}{\hat{h}(\mathbf{r})} \leq \sqrt{10} \approx 3.2. \tag{4.69}$$

The range expands at the upper end with increasing parametric-space dimension.

5. Concluding Remarks

Stabilized and VMS methods are widely used in flow computations. Stabilization parameters embedded in most of these methods almost always involve element lengths, most of the time in specific directions, such as the direction of the flow or solution gradient. We require the element lengths, including the directional element lengths, to have node-numbering invariance for all element types, including simplex elements. We have introduced a directional element length expression meeting that requirement. We accomplished that by using in the element length calculations for simplex elements a preferred parametric space instead of the standard integration parametric space. In the context of simplex elements, we analytically evaluated the element length expressions based on the two parametric spaces. We showed 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. We also showed 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.

Appendix A. Special Matrices and Their Determinants

A.1. $\mathbf{A}_n$ *and* $\mathbf{B}_n$

The first two special matrices are

$$\mathbf{A}_n = \begin{pmatrix} p & q & \cdots & q & q \\ q & p & & & q \\ \vdots & & \ddots & & \vdots \\ q & & & p & q \\ q & q & \cdots & q & p \end{pmatrix} \in \mathbb{R}^{n\times n}, \quad \mathbf{B}_n = \begin{pmatrix} q & q & \cdots & q & q \\ q & p & & & q \\ \vdots & & \ddots & & \vdots \\ q & & & p & q \\ q & q & \cdots & q & p \end{pmatrix} \in \mathbb{R}^{n\times n}. \tag{A.1}$$

We define $a_n = \det \mathsf{A}_n$ and $b_n = \det \mathsf{B}_n$. To obtain direct expressions for these, we look for recurrence relationships:

$$a_{n+1} = \det \begin{pmatrix} p & q \cdots q \\ 0 & \\ & \mathsf{A}_n \\ \vdots & \\ 0 & \end{pmatrix} + \underbrace{\det \begin{pmatrix} 0 & q \cdots q \\ q & \\ 0 & \\ \vdots & \mathsf{A}_n \\ 0 & \end{pmatrix} + \cdots + \det \begin{pmatrix} 0 & q \cdots q \\ & \\ \vdots & \mathsf{A}_n \\ 0 & \\ q & \end{pmatrix}}_{n} \tag{A.2}$$

$$= pa_n - \underbrace{\det \begin{pmatrix} q & p & q & \cdots & & q \\ 0 & q & & \cdots & & q \\ & & p & & & \\ \vdots & \vdots & & \ddots & & \vdots \\ & & & & & q \\ 0 & q & & \cdots & q & p \end{pmatrix} - \cdots - \det \begin{pmatrix} q & q & & \cdots & q & p \\ 0 & p & q & \cdots & & q \\ & q & & & & \\ \vdots & \vdots & & \ddots & & \vdots \\ & & & & p & q \\ 0 & q & & \cdots & q & q \end{pmatrix}}_{n} \tag{A.3}$$

$$= pa_n - nqb_n. \tag{A.4}$$

Note that we used the following relationship:

$$b_n = \det \begin{pmatrix} q & q & \cdots & q & q \\ q & p & & & q \\ \vdots & & \ddots & & \vdots \\ q & & & p & q \\ q & q & \cdots & q & p \end{pmatrix}$$

$$= \det \begin{pmatrix} p & q & \cdots & q & q \\ q & q & & & q \\ \vdots & & \ddots & & \vdots \\ q & & & p & q \\ q & q & \cdots & q & p \end{pmatrix} = \cdots = \det \begin{pmatrix} p & q & \cdots & q & q \\ q & p & & & q \\ \vdots & & \ddots & & \vdots \\ q & & & p & q \\ q & q & \cdots & q & q \end{pmatrix}. \tag{A.5}$$

Similarly, we obtain

$$b_{n+1} = qa_n - nqb_n. \tag{A.6}$$

Thus,

$$\begin{pmatrix} a_{n+1} \\ b_{n+1} \end{pmatrix} = \begin{pmatrix} p & -nq \\ q & -nq \end{pmatrix} \begin{pmatrix} a_n \\ b_n \end{pmatrix}, \tag{A.7}$$

with $a_1 = p$ and $b_1 = q$. We eliminate a_n and obtain

$$qa_{n+1} = pb_{n+1} + nq(p-q)b_n. \tag{A.8}$$

From Eq. (A.6), we get

$$b_{n+2} = qa_{n+1} - (n+1)qb_{n+1}, \tag{A.9}$$

substituting Eq. (A.8) into that, we obtain

$$b_{n+2} = (p-(n+1)q)b_{n+1} + nq(p-q)b_n, \tag{A.10}$$

and rearranging the terms further,

$$\begin{aligned} b_{n+2} - (p-q)b_{n+1} &= -nq(b_{n+1} - (p-q)b_n) && \text{(A.11)}\\ &= (-1)^n q^n \underbrace{(b_2 - (p-q)b_1)}_{qp-q^2-(p-q)q=0} && \text{(A.12)}\\ &= 0. && \text{(A.13)} \end{aligned}$$

From that we have

$$b_{n+2} = (p-q)^n b_2. \tag{A.14}$$

Noting from Eq. (A.6) that $b_2 = (p-q)q$, we obtain

$$b_n = (p-q)^{n-1}q. \tag{A.15}$$

We note that this expression represents b_n even for $p - q = 0$ by letting $(p-q)^0 = 1$.

We substitute this into Eq. (A.6) and obtain

$$(p-q)^n q = qa_n - nq(p-q)^{n-1}q. \tag{A.16}$$

Dividing the equation by q under the condition $q \neq 0$, we obtain

$$a_n = (p-q)^{n-1}(p+(n-1)q). \tag{A.17}$$

We note that even when $q = 0$, the above equation represents the obvious solution $a_n = p^n$, and it generally holds, again by letting $(p-q)^0 = 1$.

A.2. C_n

The determinant calculation for the third special matrix will rely on the determinant of A_n and B_n, and therefore it is more convenient to provide its definition for $n+1$:

$$\mathsf{C}_{n+1} = \begin{pmatrix} r & q & \cdots & q & q \\ q & p & & & q \\ \vdots & & \ddots & & \vdots \\ q & & & p & q \\ q & q & \cdots & q & p \end{pmatrix} \in \mathbb{R}^{(n+1)\times(n+1)}. \tag{A.18}$$

We define $c_{n+1} = \det \mathbf{C}_{n+1}$. Similar to the way of obtaining the expression for a_{n+1} (see Eqs. (A.2)–(A.4)), we obtain

$$\begin{aligned} c_{n+1} &= ra_n - qnb_n && \text{(A.19)} \\ &= r(p-q)^{n-1}(p+(n-1)q) - qn(p-q)^{n-1}q && \text{(A.20)} \\ &= (p-q)^{n-1}(r(p+(n-1)q) - q^2 n). && \text{(A.21)} \end{aligned}$$

We again note that the expression represents c_{n+1} even for $p - q = 0$ by letting $(p-q)^0 = 1$. Changing the index $n+1$ to n and limiting n to $n \geq 2$, we obtain

$$c_n = (p-q)^{n-2}((r-q)(p-q) + q(n(r-q) + (p-r))). \qquad \text{(A.22)}$$

Acknowledgments

This work was supported in part by and Rice–Waseda research agreement. It was also supported (third author) in part by ARO Grant W911NF-17-1-0046 and Top Global University Project of Waseda University.

References

1. T. J. R. Hughes and A. N. Brooks, A multi-dimensional upwind scheme with no crosswind diffusion, in *Finite Element Methods for Convection Dominated Flows*, ed. T. J. R. Hughes, AMD-Vol. 34 (ASME, 1979), pp. 19–35.
2. A. N. Brooks and T. J. R. Hughes, Streamline upwind/Petrov-Galerkin formulations for convection dominated flows with particular emphasis on the incompressible Navier-Stokes equations, *Comput. Methods Appl. Mech. Eng.* **32** (1982) 199–259.
3. T. E. Tezduyar and T. J. R. Hughes, 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 (1982), http://www.researchgate.net/publication/24313718/.
4. T. E. Tezduyar and T. J. R. Hughes, Finite element formulations for convection dominated flows with particular emphasis on the compressible Euler equations, in *Proc. AIAA 21st Aerospace Sciences Meeting*, Reno, Nevada (1983), AIAA Paper 83-0125, doi:10.2514/6.1983-125.
5. T. J. R. Hughes and T. E. Tezduyar, Finite element methods for first-order hyperbolic systems with particular emphasis on the compressible Euler equations, *Comput. Methods Appl. Mech. Eng.* **45** (1984) 217–284, doi:10.1016/0045-7825(84)90157-9.
6. T. E. Tezduyar, Stabilized finite element formulations for incompressible flow computations, *Adv. Appl. Mech.* **28** (1992) 1–44, doi:10.1016/S0065-2156(08)70153-4.
7. T. E. Tezduyar, S. Mittal, S. E. Ray and R. Shih, Incompressible flow computations with stabilized bilinear and linear equal-order-interpolation velocity-pressure elements, *Comput. Methods Appl. Mech. Eng.* **95** (1992) 221–242, doi:10.1016/0045-7825(92)90141-6.
8. T. J. R. Hughes, L. P. Franca and M. Balestra, 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** (1986) 85–99.
9. T. J. R. Hughes, Multiscale phenomena: Green's functions, the Dirichlet-to-Neumann formulation, subgrid scale models, bubbles, and the origins of stabilized methods, *Comput. Methods Appl. Mech. Engrg.* **127** (1995) 387–401.

10. T. J. R. Hughes, A. A. Oberai and L. Mazzei, Large eddy simulation of turbulent channel flows by the variational multiscale method, *Phys. Fluids* **13** (2001) 1784–1799.
11. Y. Bazilevs, V. M. Calo, J. A. Cottrell, T. J. R. Hughes, A. Reali and G. Scovazzi, Variational multiscale residual-based turbulence modeling for large eddy simulation of incompressible flows, *Comput. Methods Appl. Mech. Engrg.* **197** (2007) 173–201.
12. Y. Bazilevs and I. Akkerman, Large eddy simulation of turbulent Taylor–Couette flow using isogeometric analysis and the residual–based variational multiscale method, *J. Comput. Phys.* **229** (2010) 3402–3414.
13. T. J. R. Hughes, M. Mallet and A. Mizukami, A new finite element formulation for computational fluid dynamics: II. Beyond SUPG, *Comput. Methods Appl. Mech. Eng.* **54** (1986) 341–355.
14. T. E. Tezduyar and Y. J. Park, Discontinuity capturing finite element formulations for nonlinear convection-diffusion-reaction equations, *Comput. Methods Appl. Mech. Engrg.* **59** (1986) 307–325, doi:10.1016/0045-7825(86)90003-4.
15. T. J. R. Hughes, L. P. Franca and M. Mallet, 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. Engrg.* **63** (1987) 97–112.
16. G. J. Le Beau and T. E. Tezduyar, Finite element computation of compressible flows with the SUPG formulation, in *Advances in Finite Element Analysis in Fluid Dynamics*, FED-Vol. 123 (ASME, 1991), pp. 21–27.
17. G. J. Le Beau, S. E. Ray, S. K. Aliabadi and T. E. Tezduyar, SUPG finite element computation of compressible flows with the entropy and conservation variables formulations, *Comput. Methods Appl. Mech. Engrg.* **104** (1993) 397–422, doi:10.1016/0045-7825(93)90033-T.
18. T. E. Tezduyar, M. Behr and J. Liou, 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. Engrg.* **94** (1992) 339–351, doi:10.1016/0045-7825(92)90059-S.
19. T. E. Tezduyar, M. Behr, S. Mittal and J. Liou, 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. Engrg.* **94** (1992) 353–371, doi:10.1016/0045-7825(92)90060-W.
20. T. E. Tezduyar, Computation of moving boundaries and interfaces and stabilization parameters, *Internat. J. Numer. Methods Fluids* **43** (2003) 555–575, doi:10.1002/fld.505.
21. T. E. Tezduyar and S. Sathe, Modeling of fluid–structure interactions with the space–time finite elements: Solution techniques, *Internat. J. Numer. Methods Fluids* **54** (2007) 855–900, doi:10.1002/fld.1430.
22. K. Takizawa and T. E. Tezduyar, Multiscale space–time fluid–structure interaction techniques, *Comput. Mech.* **48** (2011) 247–267, doi:10.1007/s00466-011-0571-z.
23. K. Takizawa and T. E. Tezduyar, Space–time fluid–structure interaction methods, *Math. Models Methods Appl. Sci.* **22** (2012) 1230001, doi:10.1142/S0218202512300013.
24. K. Takizawa, T. E. Tezduyar and T. Kuraishi, Multiscale ST methods for thermo-fluid analysis of a ground vehicle and its tires, *Math. Models Methods Appl. Sci.* **25** (2015) 2227–2255, doi:10.1142/S0218202515400072.
25. Y. Bazilevs, V. M. Calo, T. J. R. Hughes and Y. Zhang, Isogeometric fluid–structure interaction: Theory, algorithms, and computations, *Comput. Mech.* **43** (2008) 3–37.

26. K. Takizawa, Y. Bazilevs and T. E. Tezduyar, Space–time and ALE-VMS techniques for patient-specific cardiovascular fluid–structure interaction modeling, *Arch. Comput. Methods Engrg.* **19** (2012) 171–225, doi:10.1007/s11831-012-9071-3.
27. Y. Bazilevs, M.-C. Hsu, K. Takizawa and T. E. Tezduyar, ALE-VMS and ST-VMS methods for computer modeling of wind-turbine rotor aerodynamics and fluid–structure interaction, *Math. Models Methods Appl. Sci.* **22** (2012) 1230002, doi:10.1142/S0218202512300025.
28. Y. Bazilevs, K. Takizawa and T. E. Tezduyar, *Computational Fluid–Structure Interaction: Methods and Applications* (Wiley, 2013).
29. Y. Bazilevs, K. Takizawa and T. E. Tezduyar, Challenges and directions in computational fluid–structure interaction, *Math. Models Methods Appl. Sci.* **23** (2013) 215–221, doi:10.1142/S0218202513400010.
30. Y. Bazilevs, K. Takizawa and T. E. Tezduyar, New directions and challenging computations in fluid dynamics modeling with stabilized and multiscale methods, *Math. Models Methods Appl. Sci.* **25** (2015) 2217–2226, doi:10.1142/S0218202515020029.
31. Y. Bazilevs, K. Takizawa and T. E. Tezduyar, Computational analysis methods for complex unsteady flow problems, *Math. Models Methods Appl. Sci.* **29** (2019) 825–838, doi:10.1142/S0218202519020020.
32. V. Kalro and T. E. Tezduyar, A parallel 3D computational method for fluid–structure interactions in parachute systems, *Comput. Methods Appl. Mech. Engrg.* **190** (2000) 321–332, doi:10.1016/S0045-7825(00)00204-8.
33. Y. Bazilevs and T. J. R. Hughes, Weak imposition of Dirichlet boundary conditions in fluid mechanics, *Comput. Fluids* **36** (2007) 12–26.
34. Y. Bazilevs, C. Michler, V. M. Calo and T. J. R. Hughes, Isogeometric variational multiscale modeling of wall-bounded turbulent flows with weakly enforced boundary conditions on unstretched meshes, *Comput. Methods Appl. Mech. Engrg.* **199** (2010) 780–790.
35. M.-C. Hsu, I. Akkerman and Y. Bazilevs, Wind turbine aerodynamics using ALE-VMS: Validation and role of weakly enforced boundary conditions, *Comput. Mech.* **50** (2012) 499–511.
36. Y. Bazilevs and T. J. R. Hughes, NURBS-based isogeometric analysis for the computation of flows about rotating components, *Comput. Mech.* **43** (2008) 143–150.
37. M.-C. Hsu and Y. Bazilevs, Fluid–structure interaction modeling of wind turbines: Simulating the full machine, *Comput. Mech.* **50** (2012) 821–833.
38. M. E. Moghadam, Y. Bazilevs, T.-Y. Hsia, I. E. Vignon-Clementel and A. L. Marsden, Modeling of Congenital Hearts Alliance (MOCHA), A comparison of outlet boundary treatments for prevention of backflow divergence with relevance to blood flow simulations, *Comput. Mech.* **48** (2011) 277–291, doi:10.1007/s00466-011-0599-0.
39. Y. Bazilevs, M.-C. Hsu, I. Akkerman, S. Wright, K. Takizawa, B. Henicke, T. Spielman and T. E. Tezduyar, 3D simulation of wind turbine rotors at full scale. Part I: Geometry modeling and aerodynamics, *Internat. J. Numer. Methods Fluids* **65** (2011) 207–235, doi:10.1002/fld.2400.
40. Y. Bazilevs, M.-C. Hsu, J. Kiendl, R. Wüchner and K.-U. Bletzinger, 3D simulation of wind turbine rotors at full scale. Part II: Fluid–structure interaction modeling with composite blades, *Internat. J. Numer. Methods Fluids* **65** (2011) 236–253.
41. M.-C. Hsu, I. Akkerman and Y. Bazilevs, High-performance computing of wind turbine aerodynamics using isogeometric analysis, *Comput. Fluids* **49** (2011) 93–100.
42. Y. Bazilevs, M.-C. Hsu and M. A. Scott, Isogeometric fluid–structure interaction analysis with emphasis on non-matching discretizations, and with application to wind turbines, *Comput. Methods Appl. Mech. Engrg.* **249–252** (2012) 28–41.

43. M.-C. Hsu, I. Akkerman and Y. Bazilevs, Finite element simulation of wind turbine aerodynamics: Validation study using NREL Phase VI experiment, *Wind Energy* **17** (2014) 461–481.
44. A. Korobenko, M.-C. Hsu, I. Akkerman, J. Tippmann and Y. Bazilevs, Structural mechanics modeling and FSI simulation of wind turbines, *Math. Models Methods Appl. Sci.* **23** (2013) 249–272.
45. Y. Bazilevs, K. Takizawa, T. E. Tezduyar, M.-C. Hsu, N. Kostov and S. McIntyre, Aerodynamic and FSI analysis of wind turbines with the ALE-VMS and ST-VMS methods, *Arch. Comput. Methods Engrg.* **21** (2014) 359–398, doi:10.1007/s11831-014-9119-7.
46. Y. Bazilevs, A. Korobenko, X. Deng and J. Yan, Novel structural modeling and mesh moving techniques for advanced FSI simulation of wind turbines, *Int. J. Numer. Methods Engrg.* **102** (2015) 766–783, doi:10.1002/nme.4738.
47. A. Korobenko, J. Yan, S. M. I. Gohari, S. Sarkar and Y. Bazilevs, FSI simulation of two back-to-back wind turbines in atmospheric boundary layer flow, *Comput. Fluids* **158** (2017) 167–175, doi:10.1016/j.compfluid.2017.05.010.
48. A. Korobenko, Y. Bazilevs, K. Takizawa and T. E. Tezduyar, Recent advances in ALE-VMS and ST-VMS computational aerodynamic and FSI analysis of wind turbines, in *Frontiers in Computational Fluid–Structure Interaction and Flow Simulation: Research from Lead Investigators Under Forty — 2018*, ed. T. E. Tezduyar, Modeling and Simulation in Science, Engineering and Technology (Springer, 2018), pp. 253–336, doi:10.1007/978-3-319-96469-0_7.
49. A. Korobenko, Y. Bazilevs, K. Takizawa and T. E. Tezduyar, Computer modeling of wind turbines: 1. ALE-VMS and ST-VMS aerodynamic and FSI analysis, *Arch. Comput. Methods Engrg.* **26** (2019) 1059–1099.
50. A. Korobenko, M.-C. Hsu, I. Akkerman and Y. Bazilevs, Aerodynamic simulation of vertical-axis wind turbines, *J. Appl. Mech.* **81** (2013) 021011, doi:10.1115/1.4024415.
51. Y. Bazilevs, A. Korobenko, X. Deng, J. Yan, M. Kinzel and J. O. Dabiri, FSI modeling of vertical-axis wind turbines, *J. Appl. Mech.* **81** (2014) 081006, doi:10.1115/1.4027466.
52. J. Yan, A. Korobenko, X. Deng and Y. Bazilevs, Computational free-surface fluid–structure interaction with application to floating offshore wind turbines, *Comput. Fluids* **141** (2016) 155–174, doi:10.1016/j.compfluid.2016.03.008.
53. Y. Bazilevs, A. Korobenko, J. Yan, A. Pal, S. M. I. Gohari and S. Sarkar, ALE–VMS formulation for stratified turbulent incompressible flows with applications, *Math. Models Methods Appl. Sci.* **25** (2015) 2349–2375, doi:10.1142/S0218202515400114.
54. Y. Bazilevs, A. Korobenko, X. Deng and J. Yan, FSI modeling for fatigue-damage prediction in full-scale wind-turbine blades, *J. Appl. Mech.* **83** (2016) 061010.
55. Y. Bazilevs, V. M. Calo, Y. Zhang and T. J. R. Hughes, Isogeometric fluid–structure interaction analysis with applications to arterial blood flow, *Comput. Mech.* **38** (2006) 310–322.
56. Y. Bazilevs, J. R. Gohean, T. J. R. Hughes, R. D. Moser and Y. Zhang, 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. Engrg.* **198** (2009) 3534–3550.
57. Y. Bazilevs, M.-C. Hsu, D. Benson, S. Sankaran and A. Marsden, Computational fluid–structure interaction: Methods and application to a total cavopulmonary connection, *Comput. Mech.* **45** (2009) 77–89.
58. Y. Bazilevs, M.-C. Hsu, Y. Zhang, W. Wang, X. Liang, T. Kvamsdal, R. Brekken and J. Isaksen, A fully-coupled fluid–structure interaction simulation of cerebral aneurysms, *Comput. Mech.* **46** (2010) 3–16.

59. Y. Bazilevs, M.-C. Hsu, Y. Zhang, W. Wang, T. Kvamsdal, S. Hentschel and J. Isaksen, Computational fluid–structure interaction: Methods and application to cerebral aneurysms, *Biomech. Model. Mechanobiol.* **9** (2010) 481–498.
60. M.-C. Hsu and Y. Bazilevs, Blood vessel tissue prestress modeling for vascular fluid–structure interaction simulations, *Finite Elements Anal. Des.* **47** (2011) 593–599.
61. C. C. Long, A. L. Marsden and Y. Bazilevs, Fluid–structure interaction simulation of pulsatile ventricular assist devices, *Comput. Mech.* **52** (2013) 971–981, doi:10.1007/s00466-013-0858-3.
62. C. C. Long, M. Esmaily-Moghadam, A. L. Marsden and Y. Bazilevs, Computation of residence time in the simulation of pulsatile ventricular assist devices, *Comput. Mech.* **54** (2014) 911–919, doi:10.1007/s00466-013-0931-y.
63. C. C. Long, A. L. Marsden and Y. Bazilevs, Shape optimization of pulsatile ventricular assist devices using FSI to minimize thrombotic risk, *Comput. Mech.* **54** (2014) 921–932, doi:10.1007/s00466-013-0967-z.
64. M.-C. Hsu, D. Kamensky, Y. Bazilevs, M. S. Sacks and T. J. R. Hughes, Fluid–structure interaction analysis of bioprosthetic heart valves: Significance of arterial wall deformation, *Comput. Mech.* **54** (2014) 1055–1071, doi:10.1007/s00466-014-1059-4.
65. M.-C. Hsu, D. Kamensky, F. Xu, J. Kiendl, C. Wang, M. C. H. Wu, J. Mineroff, A. Reali, Y. Bazilevs and M. S. Sacks, Dynamic and fluid–structure interaction simulations of bioprosthetic heart valves using parametric design with T-splines and Fung-type material models, *Comput. Mech.* **55** (2015) 1211–1225, doi:10.1007/s00466-015-1166-x.
66. D. Kamensky, M.-C. Hsu, D. Schillinger, J. A. Evans, A. Aggarwal, Y. Bazilevs, M. S. Sacks and T. J. R. Hughes, An immersogeometric variational framework for fluid-structure interaction: Application to bioprosthetic heart valves, *Comput. Methods Appl. Mech. Engrg.* **284** (2015) 1005–1053.
67. I. Akkerman, Y. Bazilevs, D. J. Benson, M. W. Farthing and C. E. Kees, Free-surface flow and fluid–object interaction modeling with emphasis on ship hydrodynamics, *J. Appl. Mech.* **79** (2012) 010905.
68. I. Akkerman, J. Dunaway, J. Kvandal, J. Spinks and Y. Bazilevs, Toward free-surface modeling of planing vessels: Simulation of the Fridsma hull using ALE-VMS, *Comput. Mech.* **50** (2012) 719–727.
69. C. Wang, M. C. H. Wu, F. Xu, M.-C. Hsu and Y. Bazilevs, Modeling of a hydraulic arresting gear using fluid–structure interaction and isogeometric analysis, *Comput. Fluids* **142** (2017) 3–14, doi:10.1016/j.compfluid.2015.12.004.
70. M. C. H. Wu, D. Kamensky, C. Wang, A. J. Herrema, F. Xu, M. S. Pigazzini, A. Verma, A. L. Marsden, Y. Bazilevs and M.-C. Hsu, Optimizing fluid–structure interaction systems with immersogeometric analysis and surrogate modeling: Application to a hydraulic arresting gear, *Comput. Methods Appl. Mech. Engrg.* **316** (2017) 668–693.
71. J. Yan, X. Deng, A. Korobenko and Y. Bazilevs, Free-surface flow modeling and simulation of horizontal-axis tidal-stream turbines, *Comput. Fluids* **158** (2017) 157–166, doi:10.1016/j.compfluid.2016.06.016.
72. A. Castorrini, A. Corsini, F. Rispoli, K. Takizawa and T. E. Tezduyar, A stabilized ALE method for computational fluid–structure interaction analysis of passive morphing in turbomachinery, *Math. Models Methods Appl. Sci.* **29** (2019) 967–994, doi:10.1142/S0218202519410057.
73. B. Augier, J. Yan, A. Korobenko, J. Czarnowski, G. Ketterman and Y. Bazilevs, Experimental and numerical FSI study of compliant hydrofoils, *Comput. Mech.* **55** (2015) 1079–1090, doi:10.1007/s00466-014-1090-5.

74. J. Yan, B. Augier, A. Korobenko, J. Czarnowski, G. Ketterman and Y. Bazilevs, FSI modeling of a propulsion system based on compliant hydrofoils in a tandem configuration, *Comput. Fluids* **141** (2016) 201–211, doi:10.1016/j.compfluid.2015.07.013.
75. T. A. Helgedagsrud, Y. Bazilevs, K. M. Mathisen and O. A. Oiseth, Computational and experimental investigation of free vibration and flutter of bridge decks, *Comput. Mech.* **63** (2019) 121–136, doi:10.1007/s00466-018-1587-4.
76. T. A. Helgedagsrud, Y. Bazilevs, A. Korobenko, K. M. Mathisen and O. A. Oiseth, Using ALE-VMS to compute aerodynamic derivatives of bridge sections, *Comput. Fluids* **179** (2019) 820–832, doi:10.1016/j.compfluid.2018.04.037.
77. T. A. Helgedagsrud, I. Akkerman, Y. Bazilevs, K. M. Mathisen and O. A. Oiseth, Isogeometric modeling and experimental investigation of moving-domain bridge aerodynamics, *ASCE J. Engrg. Mech.* **145** (2019) 04019026.
78. D. Kamensky, J. A. Evans, M.-C. Hsu and Y. Bazilevs, Projection-based stabilization of interface Lagrange multipliers in immersogeometric fluid–thin structure interaction analysis, with application to heart valve modeling, *Comput. Math. Appl.* **74** (2017) 2068–2088, doi:10.1016/j.camwa.2017.07.006.
79. Y. Yu, D. Kamensky, M.-C. Hsu, X. Y. Lu, Y. Bazilevs and T. J. R. Hughes, Error estimates for projection-based dynamic augmented Lagrangian boundary condition enforcement, with application to fluid–structure interaction, *Math. Models Meth. Appl. Sci.* **28** (2018) 2457–2509, doi:10.1142/S0218202518500537.
80. T. E. Tezduyar, K. Takizawa, C. Moorman, S. Wright and J. Christopher, Space–time finite element computation of complex fluid–structure interactions, *Int. J. Numer. Methods Fluids* **64** (2010) 1201–1218, doi:10.1002/fld.2221.
81. J. Yan, A. Korobenko, A. E. Tejada-Martinez, R. Golshan and Y. Bazilevs, A new variational multiscale formulation for stratified incompressible turbulent flows, *Comput. Fluids* **158** (2017) 150–156, doi:10.1016/j.compfluid.2016.12.004.
82. T. M. van Opstal, J. Yan, C. Coley, J. A. Evans, T. Kvamsdal and Y. Bazilevs, Isogeometric divergence-conforming variational multiscale formulation of incompressible turbulent flows, *Comput. Methods Appl. Mech. Engrg.* **316** (2017) 859–879, doi:10.1016/j.cma.2016.10.015.
83. F. Xu, G. Moutsanidis, D. Kamensky, M.-C. Hsu, M. Murugan, A. Ghoshal and Y. Bazilevs, Compressible flows on moving domains: Stabilized methods, weakly enforced essential boundary conditions, sliding interfaces and application to gas-turbine modeling, *Comput. Fluids* **158** (2017) 201–220, doi:10.1016/j.compfluid.2017.02.006.
84. T. E. Tezduyar and K. Takizawa, Space–time computations in practical engineering applications: A summary of the 25-year history, *Comput. Mech.* **63** (2019) 747–753, doi:10.1007/s00466-018-1620-7.
85. K. Takizawa and T. E. Tezduyar, Computational methods for parachute fluid–structure interactions, *Arch. Comput. Methods Engrg.* **19** (2012) 125–169, doi:10.1007/s11831-012-9070-4.
86. K. Takizawa, M. Fritze, D. Montes, T. Spielman and T. E. Tezduyar, Fluid–structure interaction modeling of ringsail parachutes with disreefing and modified geometric porosity, *Comput. Mech.* **50** (2012) 835–854, doi:10.1007/s00466-012-0761-3.
87. K. Takizawa, T. E. Tezduyar, J. Boben, N. Kostov, C. Boswell and A. Buscher, Fluid–structure interaction modeling of clusters of spacecraft parachutes with modified geometric porosity, *Comput. Mech.* **52** (2013) 1351–1364, doi:10.1007/s00466-013-0880-5.
88. K. Takizawa, T. E. Tezduyar, C. Boswell, Y. Tsutsui and K. Montel, Special methods for aerodynamic-moment calculations from parachute FSI modeling, *Comput. Mech.* **55** (2015) 1059–1069, doi:10.1007/s00466-014-1074-5.

89. K. Takizawa, D. Montes, M. Fritze, S. McIntyre, J. Boben and T. E. Tezduyar, Methods for FSI modeling of spacecraft parachute dynamics and cover separation, *Math. Models Methods Appl. Sci.* **23** (2013) 307–338, doi:10.1142/S0218202513400058.

90. K. Takizawa, T. E. Tezduyar, C. Boswell, R. Kolesar and K. Montel, FSI modeling of the reefed stages and disreefing of the Orion spacecraft parachutes, *Comput. Mech.* **54** (2014) 1203–1220, doi:10.1007/s00466-014-1052-y.

91. K. Takizawa, T. E. Tezduyar, R. Kolesar, C. Boswell, T. Kanai and K. Montel, Multiscale methods for gore curvature calculations from FSI modeling of spacecraft parachutes, *Comput. Mech.* **54** (2014) 1461–1476, doi:10.1007/s00466-014-1069-2.

92. K. Takizawa, T. E. Tezduyar and R. Kolesar, FSI modeling of the Orion spacecraft drogue parachutes, *Comput. Mech.* **55** (2015) 1167–1179, doi:10.1007/s00466-014-1108-z.

93. K. Takizawa, B. Henicke, T. E. Tezduyar, M.-C. Hsu and Y. Bazilevs, Stabilized space–time computation of wind-turbine rotor aerodynamics, *Comput. Mech.* **48** (2011) 333–344, doi:10.1007/s00466-011-0589-2.

94. K. Takizawa, B. Henicke, D. Montes, T. E. Tezduyar, M.-C. Hsu and Y. Bazilevs, Numerical-performance studies for the stabilized space–time computation of wind-turbine rotor aerodynamics, *Comput. Mech.* **48** (2011) 647–657, doi:10.1007/s00466-011-0614-5.

95. K. Takizawa, T. E. Tezduyar, S. McIntyre, N. Kostov, R. Kolesar and C. Habluetzel, Space–time VMS computation of wind-turbine rotor and tower aerodynamics, *Comput. Mech.* **53** (2014) 1–15, doi:10.1007/s00466-013-0888-x.

96. K. Takizawa, Y. Bazilevs, T. E. Tezduyar, M.-C. Hsu, O. Øiseth, K. M. Mathisen, N. Kostov and S. McIntyre, Engineering analysis and design with ALE-VMS and space–time methods, *Arch. Comput. Methods Engrg.* **21** (2014) 481–508, doi:10.1007/s11831-014-9113-0.

97. K. Takizawa, Computational engineering analysis with the new-generation space–time methods, *Comput. Mech.* **54** (2014) 193–211, doi:10.1007/s00466-014-0999-z.

98. K. Takizawa, T. E. Tezduyar, H. Mochizuki, H. Hattori, S. Mei, L. Pan and K. Montel, Space–time VMS method for flow computations with slip interfaces (ST-SI), *Math. Models Methods Appl. Sci.* **25** (2015) 2377–2406, doi:10.1142/S02182025 15400126.

99. K. Takizawa, B. Henicke, A. Puntel, T. Spielman and T. E. Tezduyar, Space–time computational techniques for the aerodynamics of flapping wings, *J. Appl. Mech.* **79** (2012) 010903, doi:10.1115/1.4005073.

100. K. Takizawa, B. Henicke, A. Puntel, N. Kostov and T. E. Tezduyar, Space–time techniques for computational aerodynamics modeling of flapping wings of an actual locust, *Comput. Mech.* **50** (2012) 743–760, doi:10.1007/s00466-012-0759-x.

101. K. Takizawa, B. Henicke, A. Puntel, N. Kostov and T. E. Tezduyar, Computer modeling techniques for flapping-wing aerodynamics of a locust, *Comput. Fluids* **85** (2013) 125–134, doi:10.1016/j.compfluid.2012.11.008.

102. K. Takizawa, N. Kostov, A. Puntel, B. Henicke and T. E. Tezduyar, Space–time computational analysis of bio-inspired flapping-wing aerodynamics of a micro aerial vehicle, *Comput. Mech.* **50** (2012) 761–778, doi:10.1007/s00466-012-0758-y.

103. K. Takizawa, T. E. Tezduyar and N. Kostov, Sequentially-coupled space–time FSI analysis of bio-inspired flapping-wing aerodynamics of an MAV, *Comput. Mech.* **54** (2014) 213–233, doi:10.1007/s00466-014-0980-x.

104. K. Takizawa, T. E. Tezduyar, A. Buscher and S. Asada, Space–time interface-tracking with topology change (ST-TC), *Comput. Mech.* **54** (2014) 955–971, doi:10.1007/s00466-013-0935-7.

105. K. Takizawa, T. E. Tezduyar and A. Buscher, Space–time computational analysis of MAV flapping-wing aerodynamics with wing clapping, *Comput. Mech.* **55** (2015) 1131–1141, doi:10.1007/s00466-014-1095-0.

106. K. Takizawa, Y. Bazilevs, T. E. Tezduyar, C. C. Long, A. L. Marsden and K. Schjodt, ST and ALE-VMS methods for patient-specific cardiovascular fluid mechanics modeling, *Math. Models Methods Appl. Sci.* **24** (2014) 2437–2486, doi:10.1142/S0218202514500250.

107. K. Takizawa, K. Schjodt, A. Puntel, N. Kostov and T. E. Tezduyar, Patient-specific computer modeling of blood flow in cerebral arteries with aneurysm and stent, *Comput. Mech.* **50** (2012) 675–686, doi:10.1007/s00466-012-0760-4.

108. K. Takizawa, K. Schjodt, A. Puntel, N. Kostov and T. E. Tezduyar, Patient-specific computational analysis of the influence of a stent on the unsteady flow in cerebral aneurysms, *Comput. Mech.* **51** (2013) 1061–1073, doi:10.1007/s00466-012-0790-y.

109. H. Suito, K. Takizawa, V. Q. H. Huynh, D. Sze and T. Ueda, FSI analysis of the blood flow and geometrical characteristics in the thoracic aorta, *Comput. Mech.* **54** (2014) 1035–1045, doi:10.1007/s00466-014-1017-1.

110. H. Suito, K. Takizawa, V. Q. H. Huynh, D. Sze, T. Ueda and T. E. Tezduyar, A geometrical-characteristics study in patient-specific FSI analysis of blood flow in the thoracic aorta, in *Advances in Computational Fluid–Structure Interaction and Flow Simulation*: *New Methods and Challenging Computations*, eds. Y. Bazilevs and K. Takizawa, Modeling and Simulation in Science, Engineering and Technology (Springer, 2016), pp. 379–386, doi:10.1007/978-3-319-40827-9_29.

111. K. Takizawa, T. E. Tezduyar, H. Uchikawa, T. Terahara, T. Sasaki, K. Shiozaki, A. Yoshida, K. Komiya and G. Inoue, Aorta flow analysis and heart valve flow and structure analysis, in *Frontiers in Computational Fluid–Structure Interaction and Flow Simulation*: *Research from Lead Investigators Under Forty — 2018*, ed. T. E. Tezduyar, Modeling and Simulation in Science, Engineering and Technology (Springer, 2018), pp. 29–89, doi:10.1007/978-3-319-96469-0_2.

112. K. Takizawa, T. E. Tezduyar, H. Uchikawa, T. Terahara, T. Sasaki and A. Yoshida, Mesh refinement influence and cardiac-cycle flow periodicity in aorta flow analysis with isogeometric discretization, *Comput. Fluids* **179** (2019) 790–798, doi:10.1016/j.compfluid.2018.05.025.

113. K. Takizawa, Y. Bazilevs, T. E. Tezduyar and M.-C. Hsu, Computational cardiovascular flow analysis with the variational multiscale methods, *J. Adv. Eng. Comput.* **3** (2019) 366–405, doi:10.25073/jaec.201932.245.

114. K. Takizawa, T. E. Tezduyar, A. Buscher and S. Asada, Space–time fluid mechanics computation of heart valve models, *Comput. Mech.* **54** (2014) 973–986, doi:10.1007/s00466-014-1046-9.

115. K. Takizawa and T. E. Tezduyar, New directions in space–time computational methods, in *Advances in Computational Fluid–Structure Interaction and Flow Simulation*: *New Methods and Challenging Computations*, eds. Y. Bazilevs and K. Takizawa, Modeling and Simulation in Science, Engineering and Technology (Springer, 2016), pp. 159–178, doi:10.1007/978-3-319-40827-9_13.

116. K. Takizawa, T. E. Tezduyar, T. Terahara and T. Sasaki, Heart valve flow computation with the Space–Time Slip Interface Topology Change (ST-SI-TC) method and Isogeometric Analysis (IGA), in *Biomedical Technology*: *Modeling*, *Experiments and Simulation*, eds. P. Wriggers and T. Lenarz, Lecture Notes in Applied and Computational Mechanics (Springer, 2018), pp. 77–99, doi:10.1007/978-3-319-59548-1_6.

117. K. Takizawa, T. E. Tezduyar, T. Terahara and T. Sasaki, Heart valve flow computation with the integrated Space–Time VMS, Slip Interface, Topology Change

and Isogeometric Discretization methods, *Comput. Fluids* **158** (2017) 176–188, doi:10.1016/j.compfluid.2016.11.012.

118. K. Takizawa, D. Montes, S. McIntyre and T. E. Tezduyar, Space–time VMS methods for modeling of incompressible flows at high Reynolds numbers, *Math. Models Methods Appl. Sci.* **23** (2013) 223–248, doi:10.1142/s0218202513400022.
119. K. Takizawa, T. E. Tezduyar, T. Kuraishi, S. Tabata and H. Takagi, Computational thermo-fluid analysis of a disk brake, *Comput. Mech.* **57** (2016) 965–977, doi:10.1007/s00466-016-1272-4.
120. K. Takizawa, T. E. Tezduyar and H. Hattori, Computational analysis of flow-driven string dynamics in turbomachinery, *Comput. Fluids* **142** (2017) 109–117, doi:10.1016/j.compfluid.2016.02.019.
121. K. Komiya, T. Kanai, Y. Otoguro, M. Kaneko, K. Hirota, Y. Zhang, K. Takizawa, T. E. Tezduyar, M. Nohmi, T. Tsuneda, M. Kawai and M. Isono, Computational analysis of flow-driven string dynamics in a pump and residence time calculation, *IOP Conference Series Earth and Environmental Science*, Vol. 240 (2019), pp. 062014, doi:10.1088/1755-1315/240/6/062014.
122. T. Kanai, K. Takizawa, T. E. Tezduyar, K. Komiya, M. Kaneko, K. Hirota, M. Nohmi, T. Tsuneda, M. Kawai and M. Isono, Methods for computation of flow-driven string dynamics in a pump and residence time, *Math. Models Methods Appl. Sci.* **29** (2019) 839–870, doi:10.1142/S021820251941001X.
123. K. Takizawa, T. E. Tezduyar, Y. Otoguro, T. Terahara, T. Kuraishi and H. Hattori, Turbocharger flow computations with the Space–Time Isogeometric Analysis (ST-IGA), *Comput. Fluids* **142** (2017) 15–20, doi:10.1016/j.compfluid.2016.02.021.
124. Y. Otoguro, K. Takizawa and T. E. Tezduyar, Space–time VMS computational flow analysis with isogeometric discretization and a general-purpose NURBS mesh generation method, *Comput. Fluids* **158** (2017) 189–200, doi:10.1016/j.compfluid.2017.04.017.
125. Y. Otoguro, K. Takizawa and T. E. Tezduyar, A general-purpose NURBS mesh generation method for complex geometries, in *Frontiers in Computational Fluid–Structure Interaction and Flow Simulation*: *Research from Lead Investigators Under Forty — 2018*, ed. T. E. Tezduyar, Modeling and Simulation in Science, Engineering and Technology (Springer, 2018), pp. 399–434, doi:10.1007/978-3-319-96469-0_10.
126. Y. Otoguro, K. Takizawa, T. E. Tezduyar, K. Nagaoka and S. Mei, Turbocharger turbine and exhaust manifold flow computation with the space–time variational multiscale method and isogeometric analysis, *Comput. Fluids* **179** (2019) 764–776, doi:10.1016/j.compfluid.2018.05.019.
127. K. Takizawa, T. E. Tezduyar, S. Asada and T. Kuraishi, Space–time method for flow computations with slip interfaces and topology changes (ST-SI-TC), *Comput. Fluids* **141** (2016) 124–134, doi:10.1016/j.compfluid.2016.05.006.
128. T. Kuraishi, K. Takizawa and T. E. Tezduyar, Space–time computational analysis of tire aerodynamics with actual geometry, road contact and tire deformation, in *Frontiers in Computational Fluid–Structure Interaction and Flow Simulation*: *Research from Lead Investigators Under Forty — 2018*, ed. T. E. Tezduyar, Modeling and Simulation in Science, Engineering and Technology (Springer, 2018), pp. 337–376, doi:10.1007/978-3-319-96469-0_8.
129. T. Kuraishi, K. Takizawa and T. E. Tezduyar, Tire aerodynamics with actual tire geometry, road contact and tire deformation, *Comput. Mech.* **63** (2019) 1165–1185, doi:10.1007/s00466-018-1642-1.

130. T. Kuraishi, K. Takizawa and T. E. Tezduyar, Space–time computational analysis of tire aerodynamics with actual geometry, road contact, tire deformation, road roughness and fluid film, *Comput. Mech.* (2019), doi:10.1007/s00466-019-01746-8.

131. T. Kuraishi, K. Takizawa and T. E. Tezduyar, Space–time Isogeometric flow analysis with built-in Reynolds-equation limit, *Math. Models Methods Appl. Sci.* **29** (2019) 871–904, doi:10.1142/S0218202519410021.

132. K. Takizawa, T. E. Tezduyar and T. Terahara, Ram-air parachute structural and fluid mechanics computations with the space–time isogeometric analysis (ST-IGA), *Comput. Fluids* **141** (2016) 191–200, doi:10.1016/j.compfluid.2016.05.027.

133. K. Takizawa, T. E. Tezduyar and T. Kanai, Porosity models and computational methods for compressible-flow aerodynamics of parachutes with geometric porosity, *Math. Models Methods Appl. Sci.* **27** (2017) 771–806, doi:10.1142/S0218202517500166.

134. T. Kanai, K. Takizawa, T. E. Tezduyar, T. Tanaka and A. Hartmann, Compressible-flow geometric-porosity modeling and spacecraft parachute computation with isogeometric discretization, *Comput. Mech.* **63** (2019) 301–321, doi:10.1007/s00466-018-1595-4.

135. T. E. Tezduyar, S. K. Aliabadi, M. Behr and S. Mittal, Massively parallel finite element simulation of compressible and incompressible flows, *Comput. Methods Appl. Mech. Engrg.* **119** (1994) 157–177, doi:10.1016/0045-7825(94)00082-4.

136. T. J. R. Hughes, J. A. Cottrell and Y. Bazilevs, Isogeometric analysis: CAD, finite elements, NURBS, exact geometry and mesh refinement, *Comput. Methods Appl. Mech. Engrg.* **194** (2005) 4135–4195.

137. K. Takizawa and T. E. Tezduyar, Space–time computation techniques with continuous representation in time (ST-C), *Comput. Mech.* **53** (2014) 91–99, doi:10.1007/s00466-013-0895-y.

138. K. Takizawa, T. E. Tezduyar and T. Sasaki, Estimation of element-based zero-stress state in arterial FSI computations with isogeometric wall discretization, in *Biomedical Technology*: *Modeling*, *Experiments and Simulation*, eds. P. Wriggers and T. Lenarz, Lecture Notes in Applied and Computational Mechanics (Springer, 2018), pp. 101–122, doi:10.1007/978-3-319-59548-1_7.

139. K. Takizawa, T. E. Tezduyar and T. Sasaki, Aorta modeling with the element-based zero-stress state and isogeometric discretization, *Comput. Mech.* **59** (2017) 265–280, doi:10.1007/s00466-016-1344-5.

140. T. Sasaki, K. Takizawa and T. E. Tezduyar, Aorta zero-stress state modeling with T-spline discretization, *Comput. Mech.* **63** (2019) 1315–1331, doi:10.1007/s00466-018-1651-0.

141. T. Sasaki, K. Takizawa and T. E. Tezduyar, Medical-image-based aorta modeling with zero-stress-state estimation, *Comput. Mech.* **64** (2019) 249–271, doi:10.1007/s00466-019-01669-4.

142. K. Takizawa, T. E. Tezduyar and T. Sasaki, Isogeometric hyperelastic shell analysis with out-of-plane deformation mapping, *Comput. Mech.* **63** (2019) 681–700, doi:10.1007/s00466-018-1616-3.

143. J. E. Akin, T. Tezduyar, M. Ungor and S. Mittal, Stabilization parameters and Smagorinsky turbulence model, *J. Appl. Mech.* **70** (2003) 2–9, doi:10.1115/1.1526569.

144. J. E. Akin and T. E. Tezduyar, Calculation of the advective limit of the SUPG stabilization parameter for linear and higher-order elements, *Comput. Methods Appl. Mech. Engrg.* **193** (2004) 1909–1922, doi:10.1016/j.cma.2003.12.050.

145. L. P. Franca, S. L. Frey and T. J. R. Hughes, Stabilized finite element methods: I. Application to the advective-diffusive model, *Comput. Methods Appl. Mech. Engrg.* **95** (1992) 253–276.

146. T. E. Tezduyar and Y. Osawa, Finite element stabilization parameters computed from element matrices and vectors, *Comput. Methods Appl. Mech. Engrg.* **190** (2000) 411–430, doi:10.1016/S0045-7825(00)00211-5.

147. M.-C. Hsu, Y. Bazilevs, V. M. Calo, T. E. Tezduyar and T. J. R. Hughes, Improving stability of stabilized and multiscale formulations in flow simulations at small time steps, *Comput. Methods Appl. Mech. Engrg.* **199** (2010) 828–840, doi:10.1016/j.cma.2009.06.019.

148. T. E. Tezduyar, Adaptive determination of the finite element stabilization parameters, in *Proc. the ECCOMAS Computational Fluid Dynamics Conf. 2001* (*CD-ROM*), Swansea, Wales, UK (2001).

149. T. E. Tezduyar, Finite element methods for fluid dynamics with moving boundaries and interfaces, in *Encyclopedia of Computational Mechanics*, eds. E. Stein, R. D. Borst and T. J. R. Hughes, Vol. 3: Fluids, Chap. 17 (Wiley, 2004), doi:10.1002/0470091355.ecm069.

150. T. E. Tezduyar, Determination of the stabilization and shock-capturing parameters in SUPG formulation of compressible flows, in *Proc. the European Congress on Computational Methods in Applied Sciences and Engineering ECCOMAS 2004* (*CD-ROM*) (Jyvaskyla, 2004).

151. T. E. Tezduyar, Finite elements in fluids: Stabilized formulations and moving boundaries and interfaces, *Comput. Fluids* **36** (2007) 191–206, doi:10.1016/j.compfluid.2005.02.011.

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

153. T. E. Tezduyar and M. Senga, Stabilization and shock-capturing parameters in SUPG formulation of compressible flows, *Comput. Methods Appl. Mech. Engrg.* **195** (2006) 1621–1632, doi:10.1016/j.cma.2005.05.032.

154. T. E. Tezduyar and M. Senga, SUPG finite element computation of inviscid supersonic flows with YZβ shock-capturing, *Comput. Fluids* **36** (2007) 147–159, doi:10.1016/j.compfluid.2005.07.009.

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

156. A. Corsini, C. Menichini, F. Rispoli, A. Santoriello and T. E. Tezduyar, A multiscale finite element formulation with discontinuity capturing for turbulence models with dominant reactionlike terms, *J. Appl. Mech.* **76** (2009) 021211, doi:10.1115/1.3062967.

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

158. A. Corsini, C. Iossa, F. Rispoli and T. E. Tezduyar, A DRD finite element formulation for computing turbulent reacting flows in gas turbine combustors, *Comput. Mech.* **46** (2010) 159–167, doi:10.1007/s00466-009-0441-0.

159. A. Corsini, F. Rispoli and T. E. Tezduyar, Stabilized finite element computation of NOx emission in aero-engine combustors, *Int. J. Numer. Methods Fluids* **65** (2011) 254–270, doi:10.1002/fld.2451.

160. A. Corsini, F. Rispoli and T. E. Tezduyar, Computer modeling of wave-energy air turbines with the SUPG/PSPG formulation and discontinuity-capturing technique, *J. Appl. Mech.* **79** (2012) 010910, doi:10.1115/1.4005060.
161. A. Corsini, F. Rispoli, A. G. Sheard and T. E. Tezduyar, Computational analysis of noise reduction devices in axial fans with stabilized finite element formulations, *Comput. Mech.* **50** (2012) 695–705, doi:10.1007/s00466-012-0789-4.
162. P. A. Kler, L. D. Dalcin, R. R. Paz and T. E. Tezduyar, SUPG and discontinuity-capturing methods for coupled fluid mechanics and electrochemical transport problems, *Comput. Mech.* **51** (2013) 171–185, doi:10.1007/s00466-012-0712-z.
163. A. Corsini, F. Rispoli, A. G. Sheard, K. Takizawa, T. E. Tezduyar and P. Venturini, A variational multiscale method for particle-cloud tracking in turbomachinery flows, *Comput. Mech.* **54** (2014) 1191–1202, doi:10.1007/s00466-014-1050-0.
164. F. Rispoli, G. Delibra, P. Venturini, A. Corsini, R. Saavedra and T. E. Tezduyar, Particle tracking and particle–shock interaction in compressible-flow computations with the V-SGS stabilization and YZβ shock-capturing, *Comput. Mech.* **55** (2015) 1201–1209, doi:10.1007/s00466-015-1160-3.
165. L. Cardillo, A. Corsini, G. Delibra, F. Rispoli and T. E. Tezduyar, Flow analysis of a wave-energy air turbine with the SUPG/PSPG stabilization and discontinuity-capturing directional dissipation, *Comput. Fluids* **141** (2016) 184–190, doi:10.1016/j.compfluid.2016.07.011.
166. A. Castorrini, A. Corsini, F. Rispoli, P. Venturini, K. Takizawa and T. E. Tezduyar, Computational analysis of wind-turbine blade rain erosion, *Comput. Fluids* **141** (2016) 175–183, doi:10.1016/j.compfluid.2016.08.013.
167. A. Castorrini, A. Corsini, F. Rispoli, P. Venturini, K. Takizawa and T. E. Tezduyar, Computational analysis of performance deterioration of a wind turbine blade strip subjected to environmental erosion, *Comput. Mech.* **64** (2019) 1133–1153, doi:10.1007/s00466-019-01697-0.
168. K. Takizawa, T. E. Tezduyar and Y. Otoguro, Stabilization and discontinuity-capturing parameters for space–time flow computations with finite element and isogeometric discretizations, *Comput. Mech.* **62** (2018) 1169–1186, doi:10.1007/s00466-018-1557-x.