Gas turbine computational flow and structure analysis with isogeometric discretization and a complex-geometry mesh generation method

Yuri Bazilevs · Kenji Takizawa · Michael C. H. Wu · Takashi Kuraishi · Reha Avsar · Zhaojing Xu · Tayfun E. Tezduyar

Received: 10 August 2020 / Accepted: 24 August 2020 © The Author(s) 2020

Abstract
A recently introduced NURBS mesh generation method for complex-geometry Isogeometric Analysis (IGA) is applied to building a high-quality mesh for a gas turbine. The compressible flow in the turbine is computed using the IGA and a stabilized method with improved discontinuity-capturing, weakly-enforced no-slip boundary-condition, and sliding-interface operators. The IGA results are compared with the results from the stabilized finite element simulation to reveal superior performance of the NURBS-based approach. Free-vibration analysis of the turbine rotor using the structural mechanics NURBS mesh is also carried out and shows that the NURBS mesh generation method can be used also in structural mechanics analysis. With the flow field from the NURBS-based turbine flow simulation, the Courant number is computed based on the NURBS mesh local length scale in the flow direction to show some of the other positive features of the mesh generation framework. The work presented further advances the IGA as a fully-integrated and robust design-to-analysis framework, and the IGA-based complex-geometry flow computation with moving boundaries and interfaces represents the first of its kind for compressible flows.

Keywords Gas turbine · Isogeometric discretization · Complex-geometry NURBS mesh generation · Compressible-flow SUPG method · ALE-SUPG method · Sliding interface · Direction-dependent local length scale

1 Introduction
The designs of commercial and military vehicles are continuously improved to deliver increased performance. The vehicles are often costly to build and maintain, and, as a result, optimal performance and operational reliability become the key aspects of the vehicle design in the commercial and military mobility applications. As the vehicle and engine designs get more sophisticated, the corresponding computational analysis methods must also mature to support advances in the mobility technology. Recent advances in geometry modeling, mesh generation, computational mechanics of fluids and structures, multiphysics modeling, and high-performance computing create a unique opportunity for the development of the next-generation predictive simulation methods and tools for general mobility applications involving complex geometry and fluid–structure interaction (FSI). These new-generation methods and tools, described in some detail and demonstrated on a single-stage gas turbine design in this article, enable high-fidelity computational analysis with greatly improved representation of the complex geometries and multiphysics phenomena.
1.1 Moving-mesh methods

Using the terminologies and categorizations used in [1–3], a method for flows with moving boundaries and interfaces (MBI), including FSI, can be an interface-tracking (moving-mesh) method or an interface-capturing (nonmoving-mesh) method, or possibly a combination of the two. In a moving-mesh method, as the interface moves and the fluid mechanics domain changes its shape, the mesh moves to adjust to the shape change and to follow (i.e. “track”) the interface. Moving the fluid mechanics mesh to follow a fluid–solid interface enables us to control the mesh resolution near the interface, have high-resolution representation of the boundary layers, and obtain accurate solutions in such critical flow regions. The arbitrary Lagrangian–Eulerian (ALE) method, with the ALE finite element method introduced in 1981 [4], is one of the earliest and most commonly used moving-mesh methods.

In incompressible-flow MBI computations the ALE is quite often used with residual-based stabilized methods, with the most notable ones being the streamline-upwind/Petrov–Galerkin (SUPG) [5], pressure-stabilizing/Petrov–Galerkin (PSPG)1 [6] and residual-based variational multiscale (RBVMS) [13–16] methods. The ALE method with the SUPG and PSPG stabilizations [17] is now called “ALE-SUPS,” and the ALE method with the VMS stabilization, “ALE-VMS.” The ALE-VMS has two more stabilization terms beyond those in the ALE-SUPS, and the additional terms give the method better turbulence modeling features. To increase their scope and accuracy, the ALE-VMS and RBVMS are often supplemented with special methods, such as those for weakly-enforced Dirichlet boundary conditions [18–20], sliding interfaces [21,22] (see Sect. 1.3), and backflow stabilization [23]. 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 [17], wind turbine aerodynamics and FSI [24–36], more specifically, vertical-axis wind turbines (VAWTs) [33,34,37,38], floating wind turbines [39], wind turbines in atmospheric boundary layers [32–34,40–42], and fatigue damage in wind turbine blades [43], patient-specific cardiovascular fluid mechanics and FSI [44–50], biomedical-device FSI [51–58], ship hydrodynamics with free-surface flow and fluid–object interaction [59,60], hydrodynamics and FSI of a hydraulic arresting gear [61,62], hydrodynamics of tidal-stream turbines with free-surface flow [63], passive-morphing FSI in turbomachinery [64], bioinspired FSI for marine propulsion [65,66], bridge aerodynamics and fluid–object interaction [67–69], and mixed ALE-VMS/immersogeometric computations [54–56,70,71] in the framework of the Fluid–Solid Interface-Tracking/Interface-Capturing Technique [72]. Recent advances in stabilized and multiscale methods may be found for stratified incompressible flows in [73] and for divergence-conforming discretizations of incompressible flows in [74].

The deforming-spatial-domain/stabilized space–time (DSD/SST) method [6,8,9] was introduced for computation of flows with MBI, including FSI, and functions as a moving-mesh method in such computations. Because the stabilization components of the original DSD/SST are the SUPG and PSPG, it is now called “ST-SUPS.” The ST-VMS is the VMS version of the DSD/SST. The VMS components of the ST-VMS are from the RBVMS. The ST-SUPS and ST-VMS, because of the higher-order accuracy of the ST framework (see [10,11]), are desirable also in computations without MBI. The ST-SUPS and ST-VMS have also been applied to many classes of FSI, MBI and fluid mechanics problems (see [75] for a comprehensive summary of the work prior to July 2018). The classes of problems include spacecraft parachute analysis for the landing-stage parachutes [2,76–79], cover-separation parachutes [80] and the drogue parachutes [81–83], wind turbine aerodynamics for horizontal-axis wind turbine (HAWT) rotors [2,3,24,84], full HAWTs [30,85–87] and VAWTs [33–36,88,89], flapping-wing aerodynamics for an actual locust [2,90–92], bioinspired MAVs [86,87,93,94] and wing-clapping [95,96], blood flow analysis of cerebral aneurysms [86,97], stent-blocked aneurysms [97–99], aortas [57,58,100–104], heart valves [57,58,87,95,104–109] and coronary arteries in motion [100], spacecraft aerodynamics [80,111], thermo-fluid analysis of ground vehicles and their tires [41,42,106,112], thermo-fluid analysis of disk brakes [113], flow-driven string dynamics in turbomachinery [35,36,114–116], flow analysis of turbocharger turbines [117–121], flow around tires with road contact and deformation [106,122–125], fluid films [125,126], ram-air parachutes [41,42,127], and compressible-flow spacecraft parachute aerodynamics [128,129].

1.2 Compressible-flow SUPG method

The compressible-flow SUPG method was first introduced, in the context of conservation variables, in a 1982 NASA technical report [130], with a concise version published as a 1983 AIAA paper [131] and more thorough version with additional examples as a 1984 journal paper [132]. That 1982 method is now called “(SUPG)82.” Several SUPG-like compressible-flow methods were developed after that. Taylor–Galerkin method [133], for example, is very similar, and under certain conditions is identical, to one of the versions of (SUPG)82. Another example of the subsequent

---

1 The name “PSPG”, coined in [6,7], refers to the specific pressure stabilization method introduced in [6,7] and the descendants of that method given in [2,8–11]. It does not refer to a general concept of pressure stabilization. An earlier version of the pressure stabilization, for Stokes flows, was introduced in [12].
SUPG-like compressible-flow methods in conservation variables is the streamline-diffusion method [134].

When the flow field is expected to have a discontinuity, SUPG and VMS methods are often supplemented with a discontinuity-capturing (DC) term, which is also called “shock-capturing term” in the context of compressible flows. Supplementing the SUPG with a DC term started as early as 1986 [135,136]. The DC term played a significant role in the evolution of the compressible-flow SUPG. At its inception, \((\text{SUPG})_{\text{DC}}\) was not used with a DC term, and the need for additional stabilization at the shocks was obvious from the test computations. Then \((\text{SUPG})_{\text{DC}}\) was recast in entropy variables and supplemented with a DC term [137]. This, as expected, resulted in better shock profiles. In a 1991 ASME paper [138], \((\text{SUPG})_{\text{DC}}\) was supplemented with a very similar DC term. All the test computations presented in [138,139], which were among the most widely used test computations at that time, showed that, with the DC term, \((\text{SUPG})_{\text{DC}}\) was very comparable in accuracy to \((\text{SUPG})_{\text{DC}}\) recast in entropy variables. The stabilized and DC methods introduced in [136] for the advection–diffusion–reaction equation included safeguards to prevent “compounding”, i.e., augmentation of the SUPG effect by the DC effect when the advection and discontinuity directions are not orthogonal.

The compressible-flow ST-SUPG method [140] is essentially the same as the compressible-flow DSD/SST method, but without necessarily implying a mesh motion. The compressible-flow DSD/SST is a straightforward mixture of the DSD/SST concept and the compressible-flow SUPG. The first 3D MBI computation with the compressible-flow ST-SUPG was reported in 1996 [141] for two high-speed trains passing each other in a tunnel.

In a recent article [142], the ALE-SUPG formulation of compressible flows using pressure-primitive variables with DC was developed and applied to the simulation of a single-stage gas turbine. Weakly-enforced boundary conditions and sliding-interface formulations for the compressible-flow equations were added in [142] to enable gas turbine simulations with boundary-layer flow and stator–rotor interaction. In [143], the formulation was applied to study off-design performance of the gas turbine stage, and in [144] it was used to simulate the full-rotorcraft aeroacoustics. Here we build on the developments in [142] and introduce improvements to the DC, weakly-enforced essential boundary condition, and sliding-interface operators to increase the accuracy and robustness of the ALE-based compressible flow formulation, especially when it is used in combination with the Isogeometric Analysis (IGA) [21,44,45,145,146].

1.3 ALE Sliding-Interface method

The ALE sliding-interface (ALE-SI) method [21,22] was introduced, in the context of incompressible-flow equations, to retain the desirable moving-mesh features of the ALE-VMS and ALE-SUPS in computations involving spinning solid surfaces, such as a turbine rotor. The mesh covering the spinning surface spins with it, retaining the high-resolution representation of the boundary layers, while the mesh on the other side of the SI remains unaffected. This is accomplished by adding to the ALE-VMS formulation interface terms that account for the compatibility conditions for the velocity and stress at the SI, accurately connecting the two sides of the solution. The ALE-SI has been applied to wind turbine aerodynamics and FSI [28–31] (more specifically, VAWTs [33,34,37,38], floating wind turbines [39], wind turbines in atmospheric boundary layers [32–34], and fatigue damage in wind turbine blades [43]), hydrodynamics and FSI of a hydraulic arresting gear [61,62], hydrodynamics of tidal-stream turbines with free-surface flow [63], bioinspired FSI for marine propulsion [65,66], and compressible flows with emphasis on gas-turbine analysis [142,143] and full-rotorcraft simulation [144].

The ST Slip Interface (ST-SI) method (the acronym “SI” will also refer to “slip interface”) was introduced in [88], also in the context of incompressible-flow equations. The interface terms added to the ST-VMS formulation are similar to those in the ALE-SI. An ST-SI version where the SI is between fluid and solid domains was also presented in [88]. The SI in that case is a “fluid–solid SI” rather than a standard “fluid–fluid SI” and enables weak enforcement of the Dirichlet boundary conditions for the fluid. The ST-SI introduced in [113] for the coupled incompressible-flow and thermal-transport equations retains the high-resolution representation of the thermo-fluid boundary layers near spinning solid surfaces. These ST-SI methods have been applied to aerodynamic analysis of VAWTs [33–36,88,89], thermo-fluid analysis of disk brakes [113], flow-driven string dynamics in turbomachinery [35,36,114–116], flow analysis of turbocharger turbines [117–121], flow around tires with road contact and deformation [106,122–125], fluid films [125,126], aerodynamic analysis of ram-air parachutes [41,42,127], and flow analysis of heart valves [57,58,102,107–109] and ventricle-valve-aorta sequences [104]. In the ST-SI version presented in [88] 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 aero-dynamic analysis of ram-air parachutes with fabric porosity [41,42,127]. The compressible-flow ST-SI methods were introduced in [128], 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 [128]. These, together with the compressible-flow
1.4 Moving-mesh Isogeometric Analysis

The IGA [21,44,145,146] was proposed in an effort to “bridge the gap” between computer-aided design (CAD) and the finite element method. As such, at its foundation, the IGA, unlike other computational methods, has a strong link to CAD, which is routinely used for the geometric design of many objects, including vehicles and their mechanical components. For the discretization of the solution fields, the IGA uses B-splines, non-uniform rational B-splines (NURBS), and related basis-function technology, which are essentially piece-wise polynomial or rational functions joined with higher-order continuity. For many problems arising in both fluid and solid/structural mechanics, the higher-order approximation accuracy and continuity of the splines result in high per-degree-of-freedom accuracy necessary to resolve the spatially multiscale phenomena present in many physical systems. This yields superior robustness, necessary for engineering applications. Computational flow analysis 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.

The success with IGA basis functions in space motivated the integration of the ST methods with isogeometric discretization, which is broadly called “ST-IGA.” The ST-IGA was introduced in [10]. Computations with the ST-VMS and ST-IGA were first reported in [10] in a 2D context, with IGA basis functions in space for flow past an airfoil, and in both space and time for the advection equation. Using higher-order basis functions in time enables deriving full benefit from using higher-order basis functions in space. This was demonstrated with the stability and accuracy analysis given in [10] 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 [10,11] and demonstrated in [90,91,93]. 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) [90,91,93], with the name coined in [85]. 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” [2,10,11,90] 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 [106,112–116,147]. 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 [2,90–92], bioinspired MAVs [86,93,94] and wing-clapping [95,96], separation aerodynamics of spacecraft [80], aerodynamics of horizontal-axis [30,85–87] and vertical-axis [33–36,88,89] wind turbines, thermo-fluid analysis of ground vehicles and their tires [41,106,112], thermo-fluid analysis of disk brakes [113], flow-driven string dynamics in turbomachinery [35,36,114–116], flow analysis of turbocharger turbines [117–121], and flow analysis of coronary arteries in motion [110].

The ST-IGA with IGA basis functions in space has been used in ST computational flow analysis of turbocharger turbines [117–121], flow-driven string dynamics in turbomachinery [35,36,115,116], ram-air parachutes [41,127], spacecraft parachutes [129], aortas [57,102,103], heart valves [57,102,107–109], ventricle-valve-aorta sequences [104], coronary arteries in motion [110], tires with road contact and deformation [123–125], fluid films [125,126], and VAWTs [35,36,89]. The image-based arterial geometries used in patient-specific arterial FSI computations do not come from the zero-stress state (ZSS) of the artery. A number of methods [2,148–158] have been proposed for estimating the ZSS required in the computations. Using IGA basis functions in space is now a key part of some of the newest ZSS estimation methods [57,156–159] and related shell analysis [160].

1.5 Complex-geometry NURBS mesh generation method

Mesh generation in the IGA, such as NURBS meshes generation, is not as established and straightforward as mesh generation in the standard discretization methods such as the finite differences and finite elements. To make the IGA-based flow and FSI computations even more powerful and more practical, the mesh generation has to be less arduous, more straightforward, more automated, and basically as established as it is now for the finite difference and finite element methods. A complex-geometry NURBS mesh generation method was introduced in [118] to make the ST-IGA use, and in a wider context the IGA use, more practical in computational flow analysis with complex geometries. The method is based on block-structured mesh generation with
existing techniques, projection of that mesh to a NURBS mesh made of patches that correspond to the blocks, and recovery of the original model surfaces. The recovery of the original surfaces is to the extent they are suitable for accurate and robust fluid mechanics computations. The method is expected to retain the refinement distribution and element quality of the block-structured mesh that we start with. Because there are ample good techniques and software for generating block-structured meshes, the method makes complex-geometry mesh generation relatively easy.

Mesh-generation tests and mesh-quality studies reported in [118,119] clearly demonstrated the good performance of the method. Practicality of the method in flow computations in real-world applications was also demonstrated in [118,119], with computational analysis of a turbocharger turbine with exhaust manifold, a disk-gap-band parachute, and a human aorta with branches. The method was used in more detailed computation of the aorta in [57,58,102,103] and turbocharger turbine in [120], with additional computational analysis in [121]. The method was shown to be very effective also in IGA-based computational flow analysis of other classes of problems, such as the pump-flow analysis part of the flow-driven string dynamics presented in [51,52,116], flow analysis of a tsunami-shelter VAWT [36,89], and ventricle-valve-aorta flow analysis [104].

1.6 ALE-SI-IGA

An SI provides mesh generation flexibility in a general context by accurately connecting the two sides of the solution computed over nonmatching meshes. This type of mesh generation flexibility is especially valuable in complex-geometry flow computations with isogeometric discretization, removing the matching requirement between the NURBS patches without loss of accuracy. This feature was used, in the context of the ST-IGA and under the name “ST-SI-IGA”, in flow analysis of heart valves [57,58,102,107–109], ventricle-valve-aorta flow analysis [104], turbocharger-turbine flow analysis [117–121], spacecraft parachute compressible-flow analysis [129], flow analysis around tires with road contact and deformation [124,125], and flow analysis of a tsunami-shelter VAWT [36,89]. It is also used, in the context of the ALE-IGA and under the name “ALE-SI-IGA”, in the gas turbine flow analysis presented here.

1.7 Stabilization and DC parameters and element lengths

In all the semi-discrete and ST stabilized and VMS methods discussed in the earlier subsections, an embedded stabilization parameter, almost universally known as “τ”, plays a significant role (see [2]). 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 ALE-SI and ST-SI also involve element length, in the direction normal to the interface. Various element lengths and τs were proposed, starting with those in [5,161] and [130–132], followed by the ones introduced in [135,136]. In many cases, the element length was seen as an advection length scale, in the flow-velocity direction. The set of τs introduced in [130–132] in conjunction with (SUPG)82 is now called “τs82”. The τ definition introduced in [136], which is for the advective limit and is now called “τSUGN1” (and the corresponding element length is now called “hSUGN”), automatically yields lower values for higher-order finite element basis functions (see [162,163]). The τ used in [138] with (SUPG)82 was a slightly modified version of τs82. Subsequent minor modifications of τs82 took into account the interaction between the DC and (SUPG)82 terms in a fashion similar to how it was done in [136] for the advection–diffusion–reaction equation. Until 2004, all these slightly modified versions of τs82 were always used with the same DC parameter, which was introduced in the 1991 ASME paper [138] and is now called “δ01”. This DC parameter was derived from the one given in [137] for the entropy variables.

Calculating the τs based on the element-level matrices and vectors was introduced in [164] 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 [164,165] to address the difficulties reported at small time-step sizes. A second element length scale, in the solution-gradient direction and called “hRGN,” was introduced in 2001 [8,166]. Recognizing this as a diffusion length scale, a new stabilization parameter for the diffusive limit, “τSUGN3,” was introduced in [8,167], to be used together with τSUGN1 and “τSUGN2,” the parameters for the advective and transient limits. For the stabilized ST methods, “τSUGN12,” representing both the advective and transient limits, was also introduced in [8].

New ways of calculating the τ and DC parameter to be used with (SUPG)82 were introduced in 2004 [167–169]. The new τs, now categorized under the label “τ04,” have a matrix structure for viscous flows and reduce to a scalar for inviscid flows. The new DC parameters were of two types: one defined in a style the Discontinuity-Capturing Directional Dissipation [8,169,170] parameter was defined, and one that is now called “YZβ” DC parameter. The YZβ DC parameter is residual-based, and it is simpler than δ01. It has options for smoother or sharper computed shocks. A number of 2D and 3D test computations with YZβ DC were reported in [171–173]. These computations showed that in addition to
being simpler than $\delta_1$, the YZ$\beta$ DC parameter was superior in accuracy. The computations reported in [171–173] were based on the compressible-flow ST SUPG.

Some new options for the stabilization parameters used with the SUPS and VMS were proposed in [9,84,85,90,112]. These include a fourth $\tau$ component, “$\tau_{\text{SUGN4}}$” [112], 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 [112] 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 [64,174–186].

### 1.8 Direction-dependent 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 [187] 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 computations presented in [187] 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. The stabilization parameters given in [124], which were mostly from [187], were the latest ones designed in conjunction with the ST-VMS.

The direction-dependent local-length-scale expressions introduced in [188] for B-spline meshes are outcome of a clear and convincing derivation and more suitable for element-level evaluation. The new expressions are based on a preferred parametric space, instead of the standard integration parametric space, and a transformation tensor that represents the relationship between the integration and preferred parametric spaces. We do not want the element splitting to influence the actual discretization, which is represented by the control or nodal points. Therefore, the local length scale should be invariant with respect to element splitting. That invariance is a crucial requirement in element definition, because unlike the element definition choices based on implementation convenience or computational efficiency, it influences the solution. It was proven in [189] that the local-length-scale expressions introduced in [188] meet that requirement.

The direction-dependent local-length-scale expressions introduced in [187,188] have been used in computational flow analysis of turbocharger turbines [120,121], compressible-flow spacecraft parachutes [129], tires with road contact and deformation [124,125], fluid films [125,126], heart valves [109], ventricle-valve-aorta sequences [104], and tsunami-shelter VAWTs [89].

### 1.9 Outline of the remaining sections

The NURBS mesh generation method is described in Sect. 2. The governing equations of compressible flows, including their quasi-linear and ALE forms, are summarized in Sect. 3. The compressible-flow computational method, including the SUPG, DC, weak-boundary-condition and SI operators, is presented in Sect. 4. The gas turbine geometry is described in Sect. 5. The meshes for the flow and structure analyses are presented in Sect. 6, together with the mesh quality evaluations. The results from the flow simulations are presented in Sect. 7. Structural mechanics simulation of the rotor free vibration is presented in Sect. 8. In Sect. 9, as an example of calculating the direction-dependent local length scales in isogeometric discretization, we calculate the Courant number in the turbine computation. The concluding remarks are given in Sect. 10.

### 2 NURBS mesh generation method

For completeness, we include, mostly from [118,119], a shortened description of the NURBS mesh generation method. First we provide a brief introduction to the basic concepts and terminologies of NURBS meshes. Figure 1 shows, as an example, a NURBS control mesh for a stator blade of a turbocharger turbine, depicting the NURBS patches and control points. Figure 2 shows the corresponding physical mesh, depicting the physical patches and elements. The relationship between the control and physical meshes is given by NURBS basis functions, defined on each patch by the knot vectors of the patch, with each space direction having its own knot vector.

#### 2.1 Basic method

We start with a block-structured mesh (grid). Such meshes are very common in finite difference and finite volume computations. The mesh quality can be measured in terms of the grid point distribution, grid line orthogonality, element aspect ratio, and the adjacent-element-length ratio. Generating high-
quality meshes requires some skills and experience, but there are ample good techniques and software for generating block-structured meshes. We assume that the block-structured mesh we start with is the outcome of such techniques or software and is of high quality. The mesh consists of trilinear elements. We see each block as a precursor to a NURBS patch. The time it would take to generate the block-structured mesh would be the time the user would spend on generating the finite difference, finite volume or finite element mesh. Beyond that, the NURBS mesh generation process is fully automated and takes essentially no time.

The second step involves a sequence of projections. For those projections, we define a common parametric space between each block and the corresponding patch. We choose the parametric space to be \( \zeta \in [0, 1]^{n_{pd}} \), where \( n_{pd} \) is the number of parametric dimensions. Then the knot vector of the patch for a given direction would be of the form

\[
\Xi = \{0, 0, \ldots, 0, 1, \ldots, 1\},
\]

where \( p \) is the polynomial order for the NURBS basis functions used in that direction. Two methods were proposed in [118,119] to determine the knot values. We refer the reader interested in knowing those methods to [118,119].

2.1.1 Projections

For each patch, the projections are done hierarchically. First the control points at the corners of the patch are set to the same locations as the corner grid points of the block. Then the edges between the corners are projected:

\[
\int_0^1 w^h \cdot (x^h - \chi^h) \, d\zeta = 0.
\]

Here \( \chi^h \in \Omega \subset \mathbb{R}^{n_{sd}} \), where \( n_{sd} \) is the number of space dimensions, is the position representation in the block, \( x^h \in \Omega \) is the position representation in the patch, and \( w^h \) is the test function of the projection. The parametric coordinate \( \zeta \) used in the integration is a general representation for all edges. Next the surfaces between the edges are projected:

\[
\int_0^1 \int_0^1 w^h \cdot (x^h - \chi^h) \, d\zeta_1 d\zeta_2 = 0.
\]

Here, the combination of the parametric coordinates \( \zeta_1 \) and \( \zeta_2 \) is a general representation for all surfaces. The projection sequence is completed with the projection of the volumes between the surfaces:

\[
\int_0^1 \int_0^1 \int_0^1 w^h \cdot (x^h - \chi^h) \, d\zeta_1 d\zeta_2 d\zeta_3 = 0.
\]

2.1.2 Merging the patches

For two adjacent blocks, the projections described in Sect. 2.1.1 result in the same control point positions over the surface they share, provided that the two blocks have the same knot vectors over the shared surface. This might happen automatically or require some additional steps, depending on the method used in determining the knot values of the parametric space. The control point variables are declared to be the same between the adjacent patches over the shared surface, which results in \( C^0 \) continuity for the basis functions across the surface.

Remark 1 Alternatively, as proposed and used in [108,117], we can declare the control point variables over the shared surface to be separate, resulting in \( C^{-1} \) continuity, and connect them with an SI.

2.2 Element reduction

We know from our experience that, to generate good quality NURBS meshes with the method proposed, in many cases we may need more elements in the finite element mesh than the target number of NURBS elements. Therefore we apply a reduction factor \( r \) to the number of elements in the block, which we can possibly be different for each direction in
the block. However, if we use different factors in different directions, two adjacent blocks must have the same reduction factor in each of the two directions of the surface they share. To make the number of elements an integer, after applying the factor \( r \), we raise the value to the next integer. After the element reduction, we use the modified versions of the methods for determining the knot values of the parametric space, and we refer the reader interested in knowing how the modifications are done to [118,119].

2.3 Methods for recovering the exact surfaces

In the NURBS mesh generation method we have, we can recover the exact surfaces, instead of just relying on the surfaces represented by the block-structured mesh.

2.3.1 Special method for arc surfaces

Suppose we have a finite element mesh with a boundary that is an arc in the exact representation (see Fig. 3). We first represent the arc by three control points and the corresponding quadratic NURBS basis functions. The arc template is shown in Fig. 4, and the corresponding knot vector and weights are

\[
\Xi = \{0, 0, 0, 1, 1, 1\} \quad (5)
\]

and

\[
w^1 = 1, \quad w^2 = \cos q, \quad w^3 = 1. \quad (6)
\]

For details, see [2,11]. Figure 5 shows the NURBS representation of the arc, superimposed on the finite element mesh. Then we insert to the knot vector given by Eq. (5) the knots of the initial NURBS mesh. From that, we obtain the new control points and weights representing the boundary. The new mesh can be seen in Fig. 6.

2.3.2 General method for CAD surfaces

If the CAD surface basis function space is a subset of our NURBS mesh basis function space, we can recover the exact surface by knot insertion to the CAD space to obtain our NURBS space. If not, the best we can do is to project the CAD surface to our NURBS space.
\section*{3 Governing equations of compressible flows}

The Navier–Stokes equations of compressible flows are often expressed using the vector of conservation variables $\mathbf{U}$:

\[
\mathbf{U} = \begin{bmatrix} \rho \\ \rho u_1 \\ \rho u_2 \\ \rho u_3 \\ \rho e_{\text{tot}} \end{bmatrix},
\]

where $\rho$ is the density, $u_i$ is the $i$-th velocity component, $i = 1, \ldots, n_{\text{sd}}$, $e_{\text{tot}} = e + \|\mathbf{u}\|^2/2$, and $e$ is the internal-energy density.

We also introduce the vector of pressure-primitive variables $\mathbf{Y}$:

\[
\mathbf{Y} = \begin{bmatrix} p \\ u_1 \\ u_2 \\ u_3 \\ T \end{bmatrix},
\]

where $p$ and $T$ are the pressure and temperature. Pressure, density, and temperature are related through the equation of state. Here we use the ideal-gas equation of state:

\[
p = \rho RT,
\]

where $R$ is the ideal-gas constant. We assume a calorically perfect gas and define the internal-energy density as

\[
e = c_v T,
\]

where $c_v = R/(\gamma - 1)$ is the specific heat at constant volume, and $\gamma$ is the heat capacity ratio.

Throughout the article, we use $(\cdot)_t$ to denote a partial time derivative holding the spatial position $\mathbf{x}$ fixed, and we use $(\cdot)_i$ to denote the spatial gradient.

With the above definitions, the Navier–Stokes equations of compressible flows, which represent the conservation of mass, momentum, and energy, can be written in terms of $\mathbf{U}$ as

\[
\mathbf{Res} = \mathbf{U}_t + \mathbf{F}^{\text{adv}}_{i,i} - \mathbf{F}^{\text{diff}}_{i,i} - \mathbf{S} = \mathbf{0},
\]

where $\mathbf{F}^{\text{adv}}_{i,i}$ and $\mathbf{F}^{\text{diff}}_{i,i}$ are the vectors of advective and diffusive fluxes, $\mathbf{S}$ is the source term, and $\mathbf{Res}$ is the strong-form residual.

The fluxes are defined as

\[
\mathbf{F}^{\text{adv}}_{i} = \begin{bmatrix} \rho u_i \\ \rho u_i u_1 + p \delta_{1i} \\ \rho u_i u_2 + p \delta_{2i} \\ \rho u_i u_3 + p \delta_{3i} \\ \rho u_i (e + \|\mathbf{u}\|^2/2) + p u_i \end{bmatrix},
\]

\[
\mathbf{F}^{\text{diff}}_{i} = \begin{bmatrix} 0 \\ \tau_{1i} \\ \tau_{2i} \\ \tau_{3i} \end{bmatrix} - q_i,
\]

where $\delta_{ij}$ is the Kronecker delta. The viscous stress $\tau_{ij}$ and heat flux $q_i$ are given as

\[
\tau_{ij} = \lambda u_{k,k} \delta_{ij} + \mu (u_{i,j} + u_{j,i}),
\]

\[
q_i = -\kappa T_i,
\]

where $\mu$ is the dynamic viscosity, $\lambda = -2\mu/3$ based on Stokes' hypothesis, and $\kappa$ is the thermal conductivity.

Introducing the mass and momentum conservation into the energy equation of Eq. (11), we get

\[
\mathbf{Res} = \mathbf{U}_t + \mathbf{F}^{\text{adv}}_{i,i} + \mathbf{F}^{\text{sp}} - \mathbf{F}^{\text{diff}}_{i,i} - \mathbf{S} = \mathbf{0},
\]

where

\[
\mathbf{F}^{\text{sp}} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ p u_{i,i} - \tau_{ij} u_{j,i} \end{bmatrix}
\]

is the contribution of the stress power in the energy equation, and $\mathbf{Res}$ is the modified strong-form residual.
3.1 Quasi-linear and ALE forms of the governing equations

The quasi-linear form of Eq. (16) is

\[ U_{,t} + \hat{A}_i U_{,i} + \hat{A}^{sp}_{i} U_{,i} - \left( \hat{K}_{ij} U_{,j} \right)_{,i} - S = 0, \quad (21) \]

where \( \hat{A}_i = \frac{\partial \text{adv}}{\partial U_i} \), \( \hat{A}^{sp}_{i} \) is such that \( \hat{A}^{sp}_{i} U_{,i} = F^{sp} \), and \( \hat{K}_{ij} \) is such that \( \hat{K}_{ij} U_{,j} = F^{\text{diff}} \).

The quasi-linear form of Eq. (16) in terms of \( Y \) is

\[ A_0 Y_{,t} + A_i Y_{,i} + A^{sp}_{i} Y_{,i} - (K_{ij} Y_{,j})_{,i} - S = 0, \quad (22) \]

where \( A_0 = \frac{\partial U}{\partial Y}, A_i = \frac{\partial \text{adv}}{\partial Y_i} = \frac{\partial \text{adv}}{\partial U} \frac{\partial U}{\partial Y_i} = \hat{A}_j A_0 \). \( A^{sp}_{i} \) is such that \( A^{sp}_{i} Y_{,i} = F^{sp} \), and \( K_{ij} \) is such that \( K_{ij} Y_{,j} = F^{\text{diff}} \). Explicit expressions for the matrices appearing in the quasi-linear forms above can be found in [142].

With the space–time Piola transformation and following the steps in [45], the convective ALE form of Eq. (16) is written as

\[ U_{,t} |_{X} + F^{\text{adv}}_{,t} + F^{sp} - \hat{u}_i U_{,i} - F^{\text{diff}}_{,t} - S = 0, \quad (23) \]

where \( \hat{u}_i \) is the \( i \)th component of the domain velocity \( \hat{u} \) and \( (\cdot)_{,X} \) denotes a partial time derivative holding the referential position \( X \) fixed. The quasi-linear form of Eq. (23) in terms of \( U \) is

\[ U_{,t} |_{X} + \left( \hat{A}_i + \hat{A}^{sp}_i - \hat{u}_i I \right) U_{,i} - \left( \hat{K}_{ij} U_{,j} \right)_{,i} - S = 0, \quad (24) \]

and in terms of \( Y \),

\[ A_0 Y_{,t} |_{X} + \left( A_i + A^{sp}_i - \hat{u}_i A_0 \right) Y_{,i} - \left( K_{ij} Y_{,j} \right)_{,i} - S = 0. \quad (25) \]

The corresponding strong-form residual is

\[ \text{Res} = A_0 Y_{,t} |_{X} + \left( A_i + A^{sp}_i - \hat{u}_i A_0 \right) Y_{,i} - \left( K_{ij} Y_{,j} \right)_{,i} - S. \quad (26) \]

For notational convenience, we define the matrices \( \hat{A}^{\text{ALE}}_i = \hat{A}_i - \hat{u}_i I \) and \( A^{\text{ALE}}_i = A_i - \hat{u}_i A_0 \).

4 Compressible-flow computational method

Let \( \Omega \subset \mathbb{R}^{nd} \) denote the domain in the current configuration, and let \( \Gamma \) be its boundary. In addition, let \( \mathcal{S} \) and \( \mathcal{V} \) denote the vector-valued pressure-primitive trial- and test-function spaces, respectively. The weak form of Eq. (25) is stated as follows: find \( Y \in \mathcal{S} \), such that \( \forall W \in \mathcal{V} \),

\[ B (W, Y) - F (W) = 0, \quad (27) \]

where

\[ B (W, Y) = \int_{\Gamma} W \cdot \left( A_0 Y_{,t} |_{X} + \left( A^{\text{ALE}}_i + A^{sp}_i \right) Y_{,i} \right) \, d\Omega + \int_{\Gamma} W_{,i} \cdot K_{ij} Y_{,j} \, d\Omega, \quad (28) \]

\[ F (W) = \int_{\Omega} W \cdot S \, d\Omega + \int_{\Gamma_{h}} W \cdot H \, d\Gamma. \quad (29) \]

Here \( \Gamma_{h} \) is the subset of \( \Gamma \) where the viscous stress and heat flux boundary conditions \( H \) are enforced weakly.

**Remark 2** In compressible-flow computations, at the outflow boundary one often sets the pressure, viscous stress, and heat flux. One way to accomplish this is to set the nodal or control-point pressure to the desired values, and enforce the remaining conditions weakly by defining \( H \) as

\[ H = \begin{bmatrix} 0 & (\tau_{1i} n_1)_{\text{out}} & (\tau_{2i} n_1)_{\text{out}} & (\tau_{3i} n_1)_{\text{out}} \\ -(p - p_{\text{out}}) n_1 + (\tau_{1i} n_1)_{\text{out}} \\ -(p - p_{\text{out}}) n_2 + (\tau_{2i} n_1)_{\text{out}} \\ -(p - p_{\text{out}}) n_3 + (\tau_{3i} n_1)_{\text{out}} \\ -(q_i n_1)_{\text{out}} \end{bmatrix}. \quad (30) \]

An alternative approach, which we use here, is to relax the strong imposition of the outlet pressure and define \( H \) as

\[ H = \begin{bmatrix} 0 \\ -(p - p_{\text{out}}) n_1 + (\tau_{1i} n_1)_{\text{out}} \\ -(p - p_{\text{out}}) n_2 + (\tau_{2i} n_1)_{\text{out}} \\ -(p - p_{\text{out}}) n_3 + (\tau_{3i} n_1)_{\text{out}} \\ -(q_i n_1)_{\text{out}} \end{bmatrix}. \quad (31) \]

Here \( n_i \) is the \( i \)th component of outward unit surface normal vector \( n \), and the subscript “out” denotes the prescribed outflow quantities.

Equations (27)–(29) are used as a point of departure to develop a formulation suitable for the discretization using the IGA or finite elements. In what follows, we present the additional method components, which are the SUPG stabilization, DC, weak enforcement of the essential boundary conditions, and the SI method.

4.1 SUPG operator

We assume the time-dependent domain \( \Omega \) is divided into \( N_{el} \) spatial finite elements each denoted by \( \Omega^e \), and define the SUPG operator as
The stabilization parameter is defined as
\[
\tau_{\text{SUPG}}(W, Y) = \frac{N_{el}}{2} \int_{\Omega^e} \left( \left( A_{i}^{\text{ALE}} \right)^T \hat{W}_i \right) \cdot \tau_{\text{SUPG}} \text{Res} \, d\Omega. 
\]  
(32)

This SUPG operator is an extension of \((\text{SUPG})_{82}\) (see Sect. 1.2) from its original version with the scalar stabilization parameter \(\tau_{82}\) (see Sect. 1.7) to a version with a matrix stabilization parameter \(\tau_{\text{SUPG}}\). When working with \(Y\), the stabilization parameter is
\[
\tau_{\text{SUPG}} = A_0^{-1} \hat{r}_{\text{SUPG}}, 
\]  
(33)

where \(\hat{r}_{\text{SUPG}}\) is the stabilization parameter when working with \(U\), which was given in [190] as
\[
\hat{r}_{\text{SUPG}} = \left( \frac{4}{\Delta t^2} I + G_{ij} \hat{A}_{i}^{\text{ALE}} \hat{A}_{j}^{\text{ALE}} + C_{i} G_{ij} \hat{K}_{ij} \right)^{-\frac{1}{2}}. 
\]  
(34)

Here \(\Delta t\) is the time-step size, \(C_{i}\) is a positive constant derived from an appropriate element-wise inverse estimate [191], and \(G_{ij}\) are the components of the element metric tensor. The expression in Eq. (34) requires calculation of the square-root-inverse of a \(5 \times 5\) matrix in 3D, which is done iteratively using a modified version of the Denman–Beavers algorithm [192,193]. More details can be found in [142].

### 4.2 DC operator

Following the approach for the SUPG operator, we first define the DC operator for working with \(U\):
\[
\sum_{e=1}^{N_{el}} \int_{\Omega^e} W_{i,e} \cdot \hat{K}_{DC} U_{i,e} \, d\Omega, 
\]  
(35)

where \(\hat{K}_{DC}\) is the matrix DC parameter. Changing the variables we are working with from \(U\) to \(Y\) gives
\[
\sum_{e=1}^{N_{el}} \int_{\Omega^e} W_{i,e} \cdot \hat{K}_{DC} A_0 Y_{i,e} \, d\Omega, 
\]  
(36)

which, in turn, defines the matrix DC parameter for working with \(Y\):
\[
K_{DC} = \hat{K}_{DC} A_0. 
\]  
(37)

We assume a diagonal form for \(\hat{K}_{DC}\):
\[
\hat{K}_{DC} = \text{diag} (\hat{k}_C, \hat{k}_M, \hat{k}_M, \hat{k}_E), 
\]  
(38)

where
\[
\hat{k}_C = \min \left( C_C, \frac{h \| \text{Res} \|_w}{\| \nabla U \|_w}, \hat{k}_{\text{cap}} \right), 
\]  
(39)

\[
\hat{k}_M = \min \left( C_M, \frac{h \| \text{Res} \|_w}{\| \nabla U \|_w}, \hat{k}_{\text{cap}} \right), 
\]  
(40)

\[
\hat{k}_E = \min \left( C_E, \frac{h \| \text{Res} \|_w}{\| \nabla U \|_w}, \hat{k}_{\text{cap}} \right). 
\]  
(41)

Here \(h\) is the element length, \(C_C, C_M,\) and \(C_E\) are \(O(1)\) positive constants, \(\| \text{Res} \|_w\) and \(\| \nabla U \|_w\) are the weighted norms
\[
\| \text{Res} \|_w = c^2 \| \text{Res}_1 \| + \| u \| \| \text{Res}_{2,\text{sd}} \| + \| \text{Res}_{\text{sd}} \|, 
\]  
(42)

\[
\| \nabla U \|_w = c^2 \| \nabla U_1 \| + \| u \| \| \nabla U_{2,\text{sd}} \| + \| U_{\text{sd}} \|, 
\]  
(43)

\(\hat{k}_{\text{cap}}\) is the maximum value of the DC parameter:
\[
\hat{k}_{\text{cap}} = \left( (u_i - \hat{u}_i) \hat{G}_{ij} (u_j - \hat{u}_j) + c^2 \hat{G}_{ii} \right)^{\frac{1}{2}}, 
\]  
(44)

where \(\hat{G}_{ij} = [G_{ij}]^{-1}\), and \(c\) is the acoustic speed.

#### Remark 3
Note that the weighted norm appropriately scales the components that have different dimensions.

#### Remark 4
The DC parameter upper bound or “cap”, \(\hat{k}_{\text{cap}}\), is a multi-dimensional generalization of the upwinding viscosity \(h(u + c)/2\). While it is expected that, on average, the residual-based definition of the DC parameter will stay well below the upwinding limit, the division by the gradient norm can lead to local spikes, which are mitigated by the cap. The cap idea was introduced and successfully employed in [194], a recent reference focusing on developing residual-based shock-capturing methods for solids. We also note that the introduction of the cap reduces the degree of nonlinearity associated with the DC terms and, as a result, improves convergence of the Newton–Raphson iterations.

#### Remark 5
The DC parameter originates from the “CAU DC” [195], which, in turn, is just an extension of \(\delta_{01}\) (see Sect. 1.7), originally designed in the context of computing steady-state solutions, from its form based on the steady-state residual to its form based on the time-dependent residual. The DC parameter may also be viewed as \(YZ\beta\) DC with \(\beta = 1\).

### 4.3 Weak-boundary-condition operator

Weakly-enforced essential boundary conditions act as near-wall models for under-resolved boundary-layer flows while converging to their strongly-enforced counterparts at optimal
rate with mesh refinement. Here we develop an improved, relative to [142], weak-boundary-condition formulation for compressible flows. The essential boundary conditions for where, defining $u_n = (u - \bar{u}) \cdot n$, the diagonal matrix of the eigenvalues of $\hat{A}_n$, is the matrix constructed using the corresponding eigenvectors as columns,

\[
\Gamma_n = \begin{bmatrix}
0 & -n_3 & n_2 & \rho \\
-n_3 & 0 & -n_1 & \rho \\
n_2 & -n_1 & 0 & \rho \\
-n_1 & n_2 & -n_3 & \rho \\
0 & 0 & 0 & \rho \sqrt{2} \\
0 & 0 & 0 & -\rho \sqrt{2}
\end{bmatrix}
\tag{48}
\]

\[
B_{WBC}(W, Y) = - \int_{\Gamma_0} W \cdot (\hat{A}_n)_- (U(Y) - G) \, d\Gamma
- \int_{\Gamma_0} W \cdot (\mu (\nabla u + \nabla^T u) + \lambda (\nabla \cdot u) I) n \, d\Gamma
- \int_{\Gamma_0} (\mu (\nabla w + \nabla^T w) + \lambda (\nabla \cdot w) I) n \cdot (u - g) \, d\Gamma
+ \int_{\Gamma_0} w \cdot \tau_{\mu} (u - g) \, d\Gamma
- \int_{\Gamma_0} w_{\theta k} \kappa T \cdot n \, d\Gamma
- \int_{\Gamma_0} \kappa \nabla w_{\theta} \cdot n (T - T_b) \, d\Gamma
+ \int_{\Gamma_0} w_{\theta k} \kappa (T - T_b) \, d\Gamma.
\tag{45}
\]

The first line on the right-hand side of Eq. (45) represents a convective contribution to the weak-boundary-condition operator. We note that the weak-boundary-condition operator is defined in terms of $U$ and $G = [\rho \; \rho g^T \; \rho c_s T_b]^T$, where $g$ and $T_b$ are the prescribed velocity and temperature. Furthermore, $[\hat{A}_n]_-$ is the “negative” part of $\hat{A}_n = \hat{A}_n n_1$. It $\Gamma_n$ is computed, based on the eigenvalue decomposition $\Gamma_n T^{-1}$ associated with hyperbolic equation systems (see, e.g., [196]), using the expression

\[
\hat{A}_n = MT[\Gamma_n]_- T^{-1} M^{-1},
\tag{46}
\]

\[
\Gamma_n = \begin{bmatrix}
u_n & 0 & 0 & 0 & 0 \\
0 & \nu_n & 0 & 0 & 0 \\
0 & 0 & \nu_n & 0 & 0 \\
0 & 0 & 0 & \nu_n + c & 0 \\
0 & 0 & 0 & 0 & \nu_n - c
\end{bmatrix}
\tag{47}
\]

\[
B_{SI}(W, Y) = - \int_{\Gamma_1} W_1 \cdot (\hat{A}_n)_- (U_1(Y_1) - U_2(Y_2)) \, d\Gamma
- \int_{\Gamma_1} W_2 \cdot (\hat{A}_n)_- (U_2(Y_2) - U_1(Y_1)) \, d\Gamma
- \int_{\Gamma_1} (w_1 - w_2) \cdot \frac{1}{2} (t(u_1, n_1) - t(u_2, n_2)) \, d\Gamma
- \int_{\Gamma_1} \frac{1}{2} (t(w_1, n_1) - t(w_2, n_2)) \cdot (u_1 - u_2) \, d\Gamma
+ \int_{\Gamma_1} (w_1 - w_2) \cdot \tau_{\mu} (u_1 - u_2) \, d\Gamma
- \int_{\Gamma_1} \frac{\kappa}{2} (\nabla w_{\theta_1} n_1 - \nabla T_1) \cdot n_2 \, d\Gamma
- \int_{\Gamma_1} \frac{\kappa}{2} (\nabla w_{\theta_2} n_2 - \nabla T_2) \cdot n_1 \, d\Gamma
+ \int_{\Gamma_1} (w_{\theta_1} - w_{\theta_2}) \, \tau_{\kappa} (T_1 - T_2) \, d\Gamma.
\tag{50}
\]
where \( t(u, n) = (\mu (\nabla u + (\nabla u)^T) + \lambda (\nabla \cdot u) I) n \) is the viscous stress operator. The convective contributions to the SI operator, given by the first two terms on the right-hand side of Eq. (50), make use of \( \{\hat{A}_n\} \) evaluated on each side of the SI. The remainder of the terms account for the viscous stress and heat flux contributions.

**4.5 Semi-discrete formulation and time integration**

The final semi-discrete form of the Navier–Stokes equations of compressible flows may be stated as: find \( Y^h \in S^h \), such that \( \forall W^h \in V^h \),

\[
B(W^h, Y^h) - F(W^h) + B_{SUPG}(W^h, Y^h) + B_{DC}(W^h, Y^h) + B_{WBC}(W^h, Y^h) + B_{SI}(W^h, Y^h) = 0, \tag{51}
\]

where \( S^h \) and \( V^h \) are the discrete counterparts of \( S \) and \( V \). The generalized-\( \alpha \) method \([197,198]\) is used for time integration. At each time step, the nonlinear equation system is solved with the Newton–Raphson method. At each Newton–Raphson iteration the resulting linear equation system is solved iteratively using the GMRES search technique \([199]\) with nodal-block-diagonal preconditioning \([200]\).

**5 Turbine geometry**

The turbine geometry comes from the interactive geometry modeling platform described in \([201]\) and was used in earlier finite element simulations reported in \([142]\). The geometry and dimensions are shown in Fig. 7. The axial length is 211.36 mm, the casing radius is 95.524 mm, and the shaft inner and outer radii are 38.862 and 77.724 mm. The stator has 24 blades, and the rotor 34. This is a smaller gas turbine design, similar to that used as part of a turboshift for Black Hawk and Apache helicopters. We note that the gap between the stator and rotor blades is larger than that in the design used earlier in \([142]\).

**6 Turbine mesh generation**

**6.1 Fluid mechanics mesh**

We divide the domain into four parts and generate a block-structured mesh for each part. Figure 8 shows the four parts. We have five SIs in the mesh. One of them is with an actual slip, and the other four are just for mesh generation purposes, giving us the flexibility to have different mesh resolutions on the two sides of the SI. Figure 9 shows the five SIs. Due to the rotational periodicity, the meshes around the stator and rotor blades are created by generating only one stator blade mesh and one rotor blade mesh and repeating those in a rotationally periodic fashion. In the cylindrical parts, the NURBS weights are selected to represent the exact geometry. Figure 10 shows the control mesh. The number of control points is 716,706.
Fig. 10  Fluid mechanics mesh. Control mesh

Fig. 11  Fluid mechanics mesh. The mesh around the stator blades. The checkerboard coloring is for differentiating between the NURBS elements. (Color figure online)

Fig. 12  Fluid mechanics mesh. The mesh around the rotor blades. The checkerboard coloring is for differentiating between the NURBS elements. (Color figure online)

Fig. 13  Rotor structural mechanics mesh. The three parts of the domain. The two shaft parts (yellow and cyan) and the part with the blades (blue) and rotational periodicity. (Color figure online)

Fig. 14  Rotor structural mechanics mesh. Control mesh

mesh and repeating that in a rotationally periodic fashion. Figure 14 shows the control mesh. The number of control points is 246,738, and the number of elements is 126,922. Figure 15 shows the blade mesh.

6.3 Mesh quality evaluation

We evaluate the mesh quality based on the mesh line orthogonality and the aspect ratio. The mesh lines are defined along the parametric coordinates \( \xi^\alpha \) in each element:

\[
\mathbf{g}_\alpha = \frac{\partial \mathbf{x}}{\partial \xi^\alpha}.
\]

(52)

where \( \alpha = 1, \ldots, n_{pd} \). For a given pair of two directions \( \alpha \) and \( \beta \), we define the angle \( \phi_{\alpha\beta} \) between the two mesh lines as

\[
\cos \left( \frac{\phi_{\alpha\beta}}{2} + \frac{\pi}{2} \right) = \frac{\mathbf{g}_\alpha \cdot \mathbf{g}_\beta}{\| \mathbf{g}_\alpha \| \| \mathbf{g}_\beta \|}.
\]

(53)

Since the choice of the parametric directions is arbitrary, in the mesh quality visualization, we report the maximum value of the mesh line angle. We also report the \( L_p \) norm, which
is defined in a standard fashion:

\[
L_p(\phi) = \left( \frac{\int_{\Omega} \frac{2}{n_{pd}(n_{pd}-1)} \sum_{\beta=1}^{n_{pd}-1} \sum_{\alpha=\beta+1}^{n_{pd}} |\phi_{\alpha\beta}|^p \, d\Omega}{\int_{\Omega} d\Omega} \right)^{\frac{1}{p}}.
\]

(54)

To define the aspect ratio, we first define the length in each parametric direction:

\[
h_\alpha = \|g_\alpha\|.
\]

(55)

With that, for a given pair of two directions \(\alpha\) and \(\beta\), the aspect ratio \(r_{\alpha\beta}\) is defined as

\[
r_{\alpha\beta} = \max \left( \frac{h_\alpha}{h_\beta}, \frac{h_\beta}{h_\alpha} \right) = \exp \left( \ln h_\alpha - \ln h_\beta \right).
\]

(56) (57)

In the mesh quality visualization, we again report the maximum value. The \(L_p\) norm we report in this case is defined in a logarithmic fashion:

\[
\ln L_p(r) = \left( \frac{\int_{\Omega} \frac{2}{n_{pd}(n_{pd}-1)} \sum_{\beta=1}^{n_{pd}-1} \sum_{\alpha=\beta+1}^{n_{pd}} |\ln h_\alpha - \ln h_\beta|^p \, d\Omega}{\int_{\Omega} d\Omega} \right)^{\frac{1}{p}}.
\]

(58)

**Remark 6** We note that here we use the “transformation tensor” definition \(D = I\) in calculating the local length scale (see [121] for the terminology and symbol definitions). There is no technical obstacle to using other definitions, such as “RQD-MAX” in [121].

Figures 16–19 show the mesh line angle and aspect ratio in the parts of the mesh around the stator and rotor blades. Tables 1 and 2 show the \(L_2\) and max norms of the mesh line angle and aspect ratio.

### 6.3.1 Rotor structural mechanics mesh

Figures 20 and 21 show the mesh line angle and aspect ratio in the part of the mesh with the blades. Tables 3 and 4 show the \(L_2\) and max norms of the mesh line angle and aspect ratio.
Fig. 18 Fluid mechanics mesh. Aspect ratio in the part of the mesh around the stator blades

Fig. 19 Fluid mechanics mesh. Aspect ratio in the part of the mesh around the rotor blades

Table 1 Fluid mechanics mesh

| Part                  | $L_2$ norm | Max norm |
|-----------------------|------------|----------|
| Cylindrical (yellow)  | 0.0°       | 0.0°     |
| Cylindrical (cyan)    | 0.0°       | 0.0°     |
| Stator blades (red)   | 15.2°      | 65.5°    |
| Rotor blades (blue)   | 15.9°      | 67.9°    |

$L_2$ and max norms of the mesh line angle. Colors indicates the parts in Fig. 8

Table 2 Fluid mechanics mesh

| Part                  | $L_2$ norm | Max norm |
|-----------------------|------------|----------|
| Cylindrical (yellow)  | 3.1        | 8.2      |
| Cylindrical (cyan)    | 3.1        | 6.9      |
| Stator blades (red)   | 2.9        | 28.3     |
| Rotor blades (blue)   | 3.1        | 13.3     |

$L_2$ and max norms of the aspect ratio. Colors indicates the parts in Fig. 8

Fig. 20 Rotor structural mechanics mesh. Mesh line angle in the part with the blades

Fig. 21 Rotor structural mechanics mesh. Aspect ratio in the part with the blades
Table 3 Rotor structural mechanics mesh

| Part       | $L_2$ norm | Max norm |
|------------|------------|----------|
| Shaft (yellow) | 0.005°     | 0.023°   |
| Shaft (cyan)  | 0.005°     | 0.035°   |
| Blades (blue) | 21.4°      | 80.7°    |

$L_2$ and max norms of the mesh line angle. Colors indicates the parts in Fig. 13

Table 4 Rotor structural mechanics mesh

| Part       | $L_2$ norm | Max norm |
|------------|------------|----------|
| Shaft (yellow) | 5.7        | 29.6     |
| Shaft (cyan)  | 8.4        | 63.7     |
| Blades (blue) | 4.0        | 62.4     |

$L_2$ and max norms of the aspect ratio. Colors indicates the parts in Fig. 13

Table 5 Mesh data for the quadratic NURBS and linear tetrahedral meshes

|               | NURBS       | Tetrahedral |
|---------------|-------------|-------------|
| Number of elements | 393,274    | 109,50,909 |
| Number of points/nodes | 716,706    | 2037,853   |
| Blade element height (mm) | 0.3        | 0.6        |

7 Flow simulation

7.1 Problem setup

The problem setup is given in Fig. 7. A uniform axial velocity of 82.3 m/s, a pressure of 2,012,790 Pa and a temperature of 1669.78 K are prescribed (strongly) at the inlet boundary. At the outlet, the pressure is set weakly to 1,212,790 Pa, and the viscous stress and heat flux to zero. No-slip conditions are enforced weakly on the stator and rotor blades. In addition, temperatures of $T = 1423.15$ K and $T = 1673.15$ K are set strongly on the stator and rotor blades. On the shaft and casing surfaces, zero normal heat flux and slip conditions are applied. The dynamic viscosity is $5.551 \times 10^{-5}$ Pa·s. We note that the flow is allowed to slip freely on the shaft and casing surfaces, different from the earlier simulations, where no-slip conditions were used.

We conduct two simulations, one using the quadratic NURBS mesh described in Sect. 6, and the other using a linear tetrahedral mesh similar to the one used in [142]. The meshes near the stator and rotor blades are shown in Fig. 22, and the mesh data is given in Table 5. The time-step size is $3 \times 10^{-7}$ s in both simulations.

7.2 Results

Figures 23 and 24 show the global distribution of the flow speed, pressure, temperature, and Mach number for the computations with the NURBS and tetrahedral meshes. The results are in good agreement intrinsically, as well as with earlier simulations of this case. We note that the solution appears continuous across all five SIs of the NURBS mesh. Figure 25 shows the temperature distribution in the stator–rotor region. The quadratic NURBS mesh with a carefully constructed boundary layer mesh is able to much better represent the thin thermal boundary layer near the stator blade than the tetrahedral mesh. The more diffuse thermal boundary layer obtained with the tetrahedral mesh leads to much less flow acceleration behind the stator blade and slower flow in the rotor channels. This can be seen in Fig. 26, which shows, for both simulations, the relative flow speed and streamlines in the stator and rotor channels. The slower flow in the rotor channels in the tetrahedral-mesh simulation translates to lower suction pressure and underestimation of the shaft torque relative to the NURBS-mesh simulation. The shaft power obtained is 354.0 kW with the NURBS mesh and 242.3 kW with the tetrahedral mesh.

Gas turbine performance can also be assessed from the adiabatic efficiency of the turbine stage $\eta_{ad}$, calculated using the expression (see [202])

$$\eta_{ad} = \frac{1 - \frac{T_{RBO}^0}{T_{SBI}^0}}{1 - \left(\frac{P_{RBO}^0}{P_{SBI}^0}\right)^{-\frac{\gamma - 1}{\gamma}}}. \quad (59)$$

Here, $T_{RBO}^0$ and $P_{RBO}^0$ are the total temperature and pressure at the rotor blade outlet, and $T_{SBI}^0$ and $P_{SBI}^0$ at the stator blade inlet. The total temperature and pressure are calculated as

$$T_0 = T \left(1 + \frac{\gamma - 1}{2} M^2\right) \quad (60)$$

and

$$P_0 = P \left(1 + \frac{\gamma - 1}{2} M^2\right)^{-\gamma / (\gamma - 1)} \quad (61),$$

where $M$ is the local Mach number. The computed values of $\eta_{ad}$ are 98.6 and 98.9% in the simulations with the NURBS and tetrahedral meshes. This is higher than the adiabatic efficiency of 85% reported for a similar case in [143], however, as we pointed out earlier, the simulation here has slip boundary conditions on the shaft and casing and a larger gap between...
Computational Mechanics

Fig. 22  Mesh near the stator and rotor blades, showing the boundary-layer resolution. The red line is the SI. (Color figure online)

Fig. 23  Flow quantities computed with the NURBS mesh

the stator and rotor. As also evident from Fig. 26, in both simulations the flow stays attached to the blade surfaces, showing almost no recirculation and resulting in higher adiabatic efficiency values.

8 Free-vibration analysis

We carry out a free-vibration analysis of the gas turbine rotor with the structural mechanics NURBS mesh described in Sect. 6. We use the material properties of Nimonic C-263,
Fig. 24 Flow quantities computed with the tetrahedral mesh

(a) Flow speed
(b) Pressure
(c) Temperature
(d) Mach number

Fig. 25 Temperature distribution in the stator–rotor region

(a) NURBS
(b) Tetrahedral

with Young’s modulus 142 GPa, density 8360 kg/m³ and Poisson’s ratio 0.3. Figure 27 shows the displacement magnitude corresponding to one of the lower and one of the higher natural frequencies. The lower frequency corresponds to a global deformation mode, while the higher frequency has deformation localized to the rotor blades.
Fig. 26  Average relative flow speed and streamlines in the stator (left) and rotor (right) channels.

(a) NURBS mesh  (b) NURBS mesh

(c) Tetrahedral mesh  (d) Tetrahedral mesh

Fig. 27  Structural displacement magnitude corresponding to two of the rotor natural frequencies. Color range from blue to red indicates the normalized-displacement range from low to high.

(a) 3558 Hz  (b) 10.271 Hz
9 An example of local-length-scale calculation in isogeometric discretization: Courant number

In the context of the fluid mechanics mesh and computation presented in the earlier sections, we give an example of calculating the direction-dependent local length scales in isogeometric discretization. We calculate the Courant number based on the local length scale in the direction of the flow velocity computed. The local length scale is based on the RQD-MAX version of the method in [121]. The time-step size, from Sect. 7, is $3 \times 10^{-7}$ s. Figures 28 and 29 show the Courant number.

10 Concluding remarks

We have made further advances in establishing the IGA as a fully-integrated and robust design-to-analysis framework. We have integrated a recently introduced NURBS mesh generation method for complex-geometry IGA and a state-of-the-art moving-mesh compressible-flow computation method. With that, we have successfully carried out IGA simulation of a gas turbine stage similar to the one used as part of a turboshift for Black Hawk and Apache helicopters. In the course of this development, we have further improved the DC, weak-boundary-condition, and SI methods for compressible flows. A comparison between the IGA and finite element simulations of the gas turbine stage shows that the NURBS mesh generation method, which is as easy to use as a block-structured mesh generation method in finite volume or finite element analysis, enables higher-order and higher-continuity representation and superior accuracy. The NURBS discretization is able to represent better the thin thermal boundary layers and higher flow velocity on the suction side of the stator blade, leading to higher flow rate in the rotor channels and resulting in higher shaft power compared to the tetrahedral finite element discretization. A free-vibration analysis of the gas turbine rotor carried out using the structural mechanics NURBS crate with the mesh generation method shows that the method can be used very effectively also in structural mechanics analysis, producing informative global and local deformation patterns. The Courant number computation based on the NURBS mesh local length scale in the flow direction demonstrates some of the other positive features of the mesh generation framework.

Acknowledgements This work was supported in part by ARO STTR Contract W911NF-20-P-0011. It was also supported in part by (second author) Grant-in-Aid for Challenging Exploratory Research 16K13779 from Japan Society for the Promotion of Science; Grant-in-Aid for Scientific Research (S) 26220002 from the Ministry of Education, Culture, Sports, Science and Technology of Japan (MEXT); Council for Science, Technology and Innovation (CSTI), Cross-Ministerial Strategic Innovation Promotion Program (SIP), “Innovative Combustion Technology” (Funding agency: JST); Rice–Waseda research agreement; (third author) Postdoctoral fellowship at Brown University; (fourth, fifth and seventh authors) ARO Grant W911NF-17-1-0046; and (seventh author) Top Global University Project of Waseda University.

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

1. Tezduyar T, Aliabadi S, Behr M (1998) Enhanced-discretization interface-capturing technique (EDICT) for computation of unsteady flows with interfaces. Comput Methods Appl Mech Eng 155:235–248. https://doi.org/10.1016/S0045-7825(97)00194-1

2. Bazilevs Y, Takizawa K, Tezduyar TE (2013) Computational fluid–structure interaction: methods and applications. Wiley. ISBN 978-0470978771

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

4. Hughes TJR, Liu WK, Zimmermann TK (1981) Lagrangian–Eulerian finite element formulation for incompressible viscous flows. Comput Methods Appl Mech Eng 29:329–349

5. Brooks AN, Hughes TJR (1982) Streamline upwind/Petrov–Galerkin formulations for convection dominated flows with particular emphasis on the incompressible Navier–Stokes equations. Comput Methods Appl Mech Eng 32:199–259

6. Tezduyar TE (1992) Stabilized finite element formulations for incompressible flow computations. Adv Appl Mech 28:1–44. https://doi.org/10.1016/S0065-2156(08)70153-4

7. Tezduyar TE, Mittal S, Ray SE, Shih R (1992) Incompressible flow computations with stabilized bilinear and linear equal-order-interpolation velocity-pressure elements. Comput Methods Appl Mech Eng 95:221–242. https://doi.org/10.1016/0045-7825(92)90141-6

8. Tezduyar TE (2003) Computation of moving boundaries and interfaces and stabilization parameters. Int J Numer Methods Fluids 43:555–575. https://doi.org/10.1002/fld.505

9. Tezduyar TE, Sathe S (2007) Modeling of fluid–structure interactions with the space–time finite elements: solution techniques. Int J Numer Methods Fluids 54:855–900. https://doi.org/10.1002/fld.1430

10. Takizawa K, Tezduyar TE (2011) Multiscale space–time fluid–structure interaction techniques. Comput Mech 48:247–267. https://doi.org/10.1007/s00446-011-0571-z

11. Takizawa K, Tezduyar TE (2012) Space–time fluid–structure interaction methods. Math Models Methods Appl Sci 22(suppl2):1230001. https://doi.org/10.1142/S0218202512300013

12. Hughes TJR, Franca LP, Balestra M (1986) A new finite element formulation for computational fluid dynamics: V. circumventing the Babuška–Brezzi condition: a stable Petrov–Galerkin formulation of the Stokes problem accommodating equal-order interpolations. Comput Methods Appl Mech Eng 59:85–99

13. Hughes TJR (1995) Multiscale phenomena: Green’s functions, the Dirichlet-to-Neumann formulation, subgrid scale models, bubbles, and the origins of stabilized methods. Comput Methods Appl Mech Eng 127:387–401

14. Hughes TJR, Oberai AA, Mazzei L (2001) Large eddy simulation of turbulent channel flows by the variational multiscale method. Phys Fluids 13:1784–1799

15. Bazilevs Y, Calo VM, Cottrell JA, Hughes TJR, Reali A, Scovazzi G (2007) Variational multiscale residual-based turbulence modeling for large eddy simulation of incompressible flows. Comput Methods Appl Mech Eng 197:173–201

16. Bazilevs Y, Akkerman I (2010) Large eddy simulation of turbulent Taylor–Couette flow using isogeometric analysis and the residual-based variational multiscale method. J Comput Phys 229:3402–3414

17. Kalro V, Tezduyar TE (2000) A parallel 3D computational method for fluid–structure interactions in parachute systems. Comput Methods Appl Mech Eng 190:321–332. https://doi.org/10.1016/S0045-7825(00)00204-8

18. Bazilevs Y, Hughes TJR (2007) Weak imposition of Dirichlet boundary conditions in fluid mechanics. Comput Fluids 36:12–26

19. Bazilevs Y, Michler C, Calo VM, Hughes TJR (2010) Isogeometric variational multiscale modeling of wall-bounded turbulent flows with weakly enforced boundary conditions on unstretched meshes. Comput Methods Appl Mech Eng 199:780–790

20. Hsu M-C, Akkerman I, Bazilevs Y (2012) Wind turbine aerodynamics using ALE-VMS: validation and role of weakly enforced boundary conditions. Comput Mech 50:499–511

21. Bazilevs Y, Hughes TJR (2008) NURBS-based isogeometric analysis for the computation of flows about rotating components. Comput Mech 43:143–150

22. Hsu M-C, Bazilevs Y (2012) Fluid–structure interaction modeling of wind turbines: simulating the full machine. Comput Mech 50:821–833

23. Moghadam ME, Bazilevs Y, Hsia T-Y, Vignon-Clementel IE, Marsden AL, Modeling of Congenital Hearts Alliance (MOCHA) (2011) A comparison of outlet boundary treatments for prevention of backflow divergence with relevance to blood flow simulations. Comput Mech 48:277–291. https://doi.org/10.1007/s00466-011-0599-0

24. Bazilevs Y, Hsu M-C, Akkerman I, Wright S, Takizawa K, Henicke B, Spilman T, Tezduyar TE (2011) 3D simulation of wind turbine rotors at full scale. Part I: geometry modeling and aerodynamics. Int J Numer Methods Fluids 65:207–235. https://doi.org/10.1002/fld.2400

25. Bazilevs Y, Hsu M-C, Kiendl J, Wüchner R, Bletzinger K-U (2011) 3D simulation of wind turbine rotors at full scale. Part II: fluid-structure interaction modeling with composite blades. Int J Numer Methods Fluids 65:236–253

26. Hsu M-C, Akkerman I, Bazilevs Y (2011) High-performance computing of wind turbine aerodynamics using isogeometric analysis. Comput Fluids 49:93–100

27. Bazilevs Y, Hsu M-C, Scott MA (2012) Isogeometric fluid-structure interaction analysis with emphasis on non-matching discretizations, and with application to wind turbines. Comput Methods Appl Mech Eng 249–252:28–41

28. Hsu M-C, Akkerman I, Bazilevs Y (2014) Finite element simulation of wind turbine aerodynamics: validation study using NREL Phase VI experiment. Wind Energy 17:461–481

29. Korobenko A, Hsu M-C, Akkerman I, Tippmann J, Bazilevs Y (2013) Structural mechanics modeling and FSI simulation of wind turbines. Math Models Methods Appl Sci 23:249–272

30. Bazilevs Y, Takizawa K, Tezduyar TE, Hsu M-C, Kostov N, McIntyre S (2014) Aerodynamic and FSI analysis of wind turbines with the ALE-VMS and ST-VMS methods. Arch Comput Methods Eng 21:359–398. https://doi.org/10.1007/s11831-014-9119-7

31. Bazilevs Y, Korobenko A, Deng X, Yan J (2015) Novel structural modeling and mesh moving techniques for advanced FSI simulation of wind turbines. Int J Numer Methods Eng 102:766–783. https://doi.org/10.1002/nme.4738

32. Korobenko A, Yan J, Gohari SMI, Sarkar S, Bazilevs Y (2017) FSI simulation of two back-to-back wind turbines in atmospheric boundary layer flow. Comput Fluids 158:167–175. https://doi.org/10.1016/j.compfluid.2017.05.010

33. Korobenko A, Bazilevs Y, Takizawa K, Tezduyar TE (2018) Recent advances in ALE-VMS and ST-VMS computational aerodynamic and FSI analysis of wind turbines. In: Tezduyar TE (ed) Frontiers in computational fluid–structure interaction and flow simulation: research from lead investigators under Forty—2018, Modeling and simulation in science, engineering and technology. Springer, pp 253–336. https://doi.org/10.1007/978-3-319-96469-0_7, ISBN 978-3-319-96468-3
34. Korobenko A, Bazilevs Y, Takizawa K, Tezduyar TE (2019) Computer modeling of wind turbines: I. ALE-VMS and ST-VMS aerodynamic and FSI analysis. Arch Comput Methods Eng 26:1059–1099. https://doi.org/10.1007/s11831-018-9292-1
35. Bazilevs Y, Takizawa K, Tezduyar TE, Hsu M-C, Otozuro Y, Mohchizuki H, Wu MCH (2020) Wind turbine and turbomachinery computational analysis with the ALE and space–time variational multiscale methods and isogeometric discretization. J Adv Eng Comput 4:1–32. https://doi.org/10.25073/jaec.2020.41.278
36. Bazilevs Y, Takizawa K, Tezduyar TE, Hsu M-C, Otozuro Y, Mohchizuki H, Wu MCH (2020) ALE and space–time variational multiscale isogeometric analysis of wind turbines and turbomachinery. In: Graa A, Sameh A (eds) Parallel algorithms in computational science and engineering, Modeling and simulation in science, engineering and technology. Springer, pp 195–233. https://doi.org/10.1007/978-3-030-43736-7_7. ISBN 978-3-030-43735-0
37. Korobenko A, Hsu M-C, Akkerman I, Bazilevs Y (2013) Aerodynamic simulation of vertical-axis wind turbines. J Appl Mech 81:021011. https://doi.org/10.1115/1.4024415
38. Bazilevs Y, Korobenko A, Deng X, Yan J, Kinzel M, Dabiri JO (2014) FSI modeling of vertical-axis wind turbines. J Appl Mech 81:081006. https://doi.org/10.1115/1.4027466
39. Yan J, Korobenko A, Deng X, Bazilevs Y (2016) Computational free-surface fluid–structure interaction with application to floating offshore wind turbines. Comput Fluids 141:155–174. https://doi.org/10.1016/j.compfluid.2016.03.008
40. Bazilevs Y, Korobenko A, Yan J, Pal A, Gohari SMI, Sarkar S (2015) ALE-VMS formulation for stratified turbulent incompressible flows with applications. Math Models Methods Appl Sci 25:2349–2375. https://doi.org/10.1142/S0218202515400114
41. Takizawa K, Bazilevs Y, Tezduyar TE, Korobenko A (2020) Variational multiscale flow analysis in aerospace, energy and transportation technologies. J Adv Eng Comput 4:83–117. https://doi.org/10.25073/jaec.2020.42.279
42. Takizawa K, Bazilevs Y, Tezduyar TE, Korobenko A (2020) Variational multiscale flow analysis in aerospace, energy and transportation technologies. In: Graa A, Sameh A (eds) Parallel algorithms in computational science and engineering, Modeling and simulation in science, engineering and technology. Springer, pp 235–280. https://doi.org/10.1007/978-3-030-43736-7_8. ISBN 978-3-030-43735-0
43. Bazilevs Y, Korobenko A, Deng X, Yan J (2016) FSI modeling for fatigue-damage prediction in full-scale wind turbine blades. J Appl Mech 83:061010
44. Bazilevs Y, Calo VM, Zhang Y, Hughes TJR (2006) Isogeometric fluid–structure interaction analysis with applications to arterial blood flow. Comput Mech 38:310–322
45. Bazilevs Y, Calo VM, Hughes TJR, Zhang Y (2008) Isogeometric fluid–structure interaction: theory, algorithms, and computations. Comput Mech 43:3–37
46. Bazilevs Y, Gheean JR, Hughes TJR, Moser RD, Zhang Y (2009) Patient-specific isogeometric fluid–structure interaction analysis of thoracic aortic blood flow due to implantation of the Jarvik 2000 left ventricular assist device. Comput Methods Appl Mech Eng 198:3534–3550
47. Bazilevs Y, Hsu M-C, Benson D, Sankaran S, Marsden A (2009) Computational fluid–structure interaction: methods and application to a total cavopulmonary connection. Comput Mech 45:77–89
48. Bazilevs Y, Hsu M-C, Zhang Y, Wang W, Liang X, Kvensdal T, Brekker R, Isaksen J (2010) A fully-coupled fluid–structure interaction simulation of cerebral aneurysms. Comput Mech 46:3–16
49. Bazilevs Y, Hsu M-C, Zhang Y, Wang W, Kvensdal T, Hentschel S, Isaksen J (2010) Computational fluid–structure interaction: methods and application to cerebral aneurysms. Biomech Model Mechanobiol 9:481–498
50. Hsu M-C, Bazilevs Y (2011) Blood vessel tissue prestress modeling for vascular fluid–structure interaction simulations. Finite Elem Anal Des 47:593–599
51. Long CC, Marsden AL, Bazilevs Y (2013) Fluid–structure interaction simulation of pulsatile ventricular assist devices. Comput Mech 52:971–981. https://doi.org/10.1007/s00466-013-0858-3
52. Long CC, Esmaiily-Moghadam M, Marsden AL, Bazilevs Y (2014) Computation of residence time in the simulation of pulsatile ventricular assist devices. Comput Mech 54:911–919. https://doi.org/10.1007/s00466-013-0931-y
53. Long CC, Marsden AL, Bazilevs Y (2014) Shape optimization of pulsatile ventricular assist devices using FSI to minimize thrombotic risk. Comput Mecl 54:921–932. https://doi.org/10.1007/s00466-013-0967-z
54. Hsu M-C, Kamensky D, Bazilevs Y, Sacks MS, Hughes TJR (2014) Fluid–structure interaction analysis of bioprosthetic heart valves: significance of arterial wall deformation. Comput Mech 54:1055–1071. https://doi.org/10.1007/s00466-014-1059-4
55. Hsu M-C, Kamensky D, Xu F, Kiendl J, Wang C, Wu MCH, Minneroff J, Reali A, Bazilevs Y, Sacks MS (2015) Dynamic and fluid-structure interaction simulations of bioprosthetic heart valves using parametric design with T-splines and Fung-type material models. Comput Mech 55:1211–1225. https://doi.org/10.1007/s00466-015-1166-x
56. Kamensky D, Hsu M-C, Schillinger D, Evans JA, Aggarwal A, Bazilevs Y, Sacks MS, Hughes TJR (2015) An isogeometric variational framework for fluid-structure interaction: application to bioprosthetic heart valves. Comput Methods Appl Mech Eng 284:1005–1053
57. Takizawa K, Bazilevs Y, Tezduyar TE, Hsu M-C (2019) Computational cardiovascular flow analysis with the variational multiscale methods. J Adv Eng Comput 3:366–405. https://doi.org/10.25073/jaec.201932.245
58. Hughes TJR, Takizawa K, Bazilevs Y, Tezduyar TE, Hsu M-C (2020) Computational cardiovascular analysis with the variational multiscale methods and isogeometric discretization. In: Graa A, Sameh A (eds) Parallel algorithms in computational science and engineering, Modeling and simulation in science, engineering and technology. Springer, pp 151–193. https://doi.org/10.1007/978-3-030-43736-7_6. ISBN 978-3-030-43735-0
59. Akkerman I, Bazilevs Y, Benson DJ, Furthing MW, Kees CE (2012) Free-surface flow and fluid-object interaction modeling with emphasis on ship hydrodynamics. J Appl Mech 79:010905
60. Akkerman I, Duanaway J, Kvanjil J, Spinks J, Bazilevs Y (2012) Toward free-surface modeling of planning vessels: simulation of the Fridsma hull using ALE-VMS. Comput Mech 50:719–727
61. Wang C, Wu MCH, Xu F, Hsu M-C, Bazilevs Y (2017) Modeling of a hydraulic arresting gear using fluid–structure interaction and isogeometric analysis. Comput Fluids 142:3–14. https://doi.org/10.1016/j.compfluid.2015.12.004
62. Wu MCH, Kamensky D, Wang C, Herrema AJ, Xu F, Pigazzini MS, Verma A, Marsden AL, Bazilevs Y, Hsu M-C (2017) Optimizing fluid–structure interaction systems with isogeometric analysis and surrogate modeling: application to a hydraulic arresting gear. Comput Methods Appl Mech Eng 316:668–693
63. Yan J, Deng X, Korobenko A, Bazilevs Y (2017) Free-surface flow modeling and simulation of horizontal-axis tidal-stream turbines. Comput Fluids 158:157–166. https://doi.org/10.1016/j.compfluid.2016.06.016
64. Castorini A, Corsini A, Rispoli F, Takizawa K, Tezduyar TE (2019) A stabilized ALE method for computational fluid-structure interaction analysis of passive morphing in turbomachinery. Math Models Methods Appl Sci 29:967–994. https://doi.org/10.1142/S0218202519410057
65. Augier B, Yan J, Korobenko A, Czarnowski J, Ketterman G, Bazilevs Y (2015) Experimental and numerical FSI study of com-
98. Takizawa K, Schjodt K, Puntel A, Kostov N, Tezduyar TE (2012) Patient-specific computer modeling of blood flow in cerebral arteries with aneurysm and stent. Comput Mech 50:675–686. https://doi.org/10.1007/s00466-012-0760-4
99. Takizawa K, Schjodt K, Puntel A, Kostov N, Tezduyar TE (2013) Patient-specific computational analysis of the influence of a stent on the unsteady flow in cerebral aneurysms. Comput Mech 51:1061–1073. https://doi.org/10.1007/s00466-012-0790-y
100. Suito H, Takizawa K, Huynh VQH, Sze D, Ueda T (2014) FSI analysis of the blood flow and geometrical characteristics in the thoracic aorta. Comput Mech 54:1035–1045. https://doi.org/10.1007/s00466-014-1017-1
101. Suito H, Takizawa K, Huynh VQH, Sze D, Ueda T, Tezduyar TE (2016) A geometrical-characteristics study in patient-specific FSI analysis of blood flow in the thoracic aorta. In: Bazilevs Y, Takizawa K (eds) Advances in computational fluid–structure interaction and flow simulation: new methods and challenging computations, Modeling and simulation in science, engineering and technology. Springer, pp 379–386. https://doi.org/10.1007/978-3-319-40827-9_29. ISBN 978-3-319-40825-5
102. Takizawa K, Tezduyar TE, Uchikawa H, Terahara T, Sasaki T, Shiozaki K, Yoshida A, Komiya K, Inoue G (2018) Aorta flow analysis and heart valve flow and structure analysis. In: Tezduyar TE (ed) Frontiers in computational fluid–structure interaction and flow simulation: research from lead investigators under Forty—2018, Modeling and simulation in science, engineering and technology. Springer, pp 29–89. https://doi.org/10.1007/978-3-319-96469-0_2. ISBN 978-3-319-96468-3
103. Takizawa K, Tezduyar TE, Uchikawa H, Terahara T, Sasaki T, Yoshida A (2019) Mesh refinement influence and cardiac-cycle flow periodicity in aorta flow analysis with isogeometric discretization. Comput Fluids 179:790–798. https://doi.org/10.1016/j.compfluid.2018.05.025
104. Terahara T, Takizawa K, Tezduyar TE, Tsushima A, Shiozaki K (2020) Ventricle-valve-aorta flow analysis with the space–time isogeometric discretization and topology change. Comput Mech 65:1343–1363. https://doi.org/10.1007/s00466-020-01822-4
105. Takizawa K, Tezduyar TE, Buscher A, Asada S (2014) Space–time fluid mechanics computation of heart valve models. Comput Mech 54:973–986. https://doi.org/10.1007/s00466-014-1046-9
106. Takizawa K, Tezduyar TE (2016) New directions in space–time computational methods. In: Bazilevs Y, Takizawa K (eds) Advances in computational fluid–structure interaction and flow simulation: new methods and challenging computations, Modeling and simulation in science, engineering and technology. Springer, pp 159–178. https://doi.org/10.1007/978-3-319-40827-9_13. ISBN 978-3-319-40825-5
107. Takizawa K, Tezduyar TE, Terahara T, Sasaki T (2018) Heart valve flow computation with the space–time slip interface topology change (ST-SI-TC) method and isogeometric analysis (IGA). In: Wriggers P, Lenarz T (eds) Biomedical technology: modeling, experiments and simulation. Lecture notes in applied and computational mechanics. Springer, pp 77–99. https://doi.org/10.1007/978-3-319-59548-1_6. ISBN 978-3-319-59547-4
108. Takizawa K, Tezduyar TE, Terahara T, Sasaki T (2017) Heart valve flow computation with the integrated space–time VMS, slip interface, topology change and isogeometric discretization methods. Comput Fluids 158:176–188. https://doi.org/10.1016/j.compfluid.2016.11.012
109. Terahara T, Takizawa K, Tezduyar TE, Bazilevs Y, Hsu M-C (2020) Heart valve isogeometric sequentially-coupled FSI analysis with the space–time topology change method. Comput Mech 65:1167–1187. https://doi.org/10.1007/s00466-019-01813-0
110. Yu Y, Zhang YJ, Takizawa K, Tezduyar TE, Sasaki T (2020) Anatomically realistic lumen motion representation in patient-specific space–time isogeometric flow analysis of coronary arteries with time-dependent medical-image data. Comput Mech 65:395–404. https://doi.org/10.1007/s00446-019-01774-4
111. Takizawa K, Montes D, McIntyre S, Tezduyar TE (2013) Space–time VMS methods for modeling of incompressible flows at high Reynolds numbers. Math Models Methods Appl Sci 23:223–248. https://doi.org/10.1142/s218220513400022
112. Takizawa K, Tezduyar TE, Kuraishi T (2015) Multiscale ST methods for thermo-fluid analysis of a ground vehicle and its tires. Math Models Methods Appl Sci 25:2227–2255. https://doi.org/10.1142/S2182205115400072
113. Takizawa K, Tezduyar TE, Kuraishi T, Tabata S, Takagi H (2016) Computational thermo-fluid analysis of a disk brake. Comput Mech 57:965–977. https://doi.org/10.1007/s00446-016-1272-4
114. Takizawa K, Tezduyar TE, Hattori H (2017) Computational analysis of flow-driven string dynamics in turbomachinery. Comput Fluids 142:109–117. https://doi.org/10.1016/j.compfluid.2016.02.019
115. Komiya K, Kanai T, Otoguro Y, Kaneko M, Hirota K, Zhang Y, Takizawa K, Tezduyar TE, Nomih M, Tsuneda T, Kawai M, Isono M (2019) Computational analysis of flow-driven string dynamics in a pump and residence time calculation. IOP Conf Ser Earth Environ Sci 240:062014. https://doi.org/10.1088/1755-1315/240/6/062014
116. Kanai T, Takizawa K, Tezduyar TE, Komiya K, Kaneko M, Hirota K, Nomih M, Tsuneda T, Kawai M, Isono M (2019) Methods for computation of flow-driven string dynamics in a pump and residence time. Math Models Methods Appl Sci 29:839–870. https://doi.org/10.1142/s201820251941001x
117. Takizawa K, Tezduyar TE, Otoguro Y, Terahara T, Kuriashi T, Hattori H (2017) Turbocharger flow computations with the space–time isogeometric analysis (ST-IGA). Comput Fluids 142:15–20. https://doi.org/10.1016/j.compfluid.2016.02.021
118. Otoguro Y, Takizawa K, Tezduyar TE (2017) Space–time VMS computational flow analysis with isogeometric discretization and a general-purpose NURBS mesh generation method. Comput Fluids 158:189–200. https://doi.org/10.1016/j.compfluid.2017.04.017
119. Otoguro Y, Takizawa K, Tezduyar TE (2018) A general-purpose NURBS mesh generation method for complex geometries. In: Tezduyar TE (ed) Frontiers in computational fluid–structure interaction and flow simulation: research from lead investigators under Forty—2018, Modeling and simulation in science, engineering and technology. Springer, pp 390–434. https://doi.org/10.1007/978-3-319-96469-0_10. ISBN 978-3-319-96468-3
120. Otoguro Y, Takizawa K, Tezduyar TE, Nagaoka K, Mei S (2019) Turbocharger turbine and exhaust manifold flow computation with the space–time variational multiscale method and isogeometric analysis. Comput Fluids 179:764–776. https://doi.org/10.1016/j.compfluid.2018.05.019
121. Otoguro Y, Takizawa K, Tezduyar TE, Nagaoka K, Avsar R, Zhang Y (2019) Space–time VMS flow analysis of a turbocharger turbine with isogeometric discretization: computations with time-dependent and steady-inflow representations of the intake/exhaust cycle. Comput Mech 64:1403–1419. https://doi.org/10.1007/s00446-019-01722-2
122. Takizawa K, Tezduyar TE, Asada S, Kuraishi T (2016) Space–time method for flow computations with slip interfaces and topology changes (ST-SI-TC). Comput Fluids 142:124–134. https://doi.org/10.1016/j.compfluid.2016.05.006
123. Kuriashi T, Takizawa K, Tezduyar TE (2018) Space–time computational analysis of tire aerodynamics with actual geometry, road contact and tire deformation. In: Tezduyar TE (ed) Frontiers in computational fluid–structure interaction and flow sim-
156. Takizawa K, Tezduyar TE, Sasaki T (2018) Estimation of element-based zero-stress state in arterial FSI computations with isogeometric wall discretization. In: Wrzggs P, Lenarz T (eds) Biomedical technology: modeling, experiments and simulation. Lecture notes in applied and computational mechanics. Springer, pp 101–122. https://doi.org/10.1007/978-3-319-59548-1_7. ISBN 978-3-319-59547-4

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

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

159. Sasaki T, Takizawa K, Tezduyar TE (2019) Medical-image-based aorta modeling with zero-stress-state estimation. Comput Mech 64:249–271. https://doi.org/10.1007/s00466-019-1669-4

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

161. Hughes TJR, Brooks AN (1979) A multi-dimensional upwind scheme with no crosswind diffusion. In: Hughes TJR (ed) Finite element methods for convection dominated flows, AMD, vol 34. ASME, New York, pp 19–35

162. Akin JE, Tezduyar T, Ungor M, Mittal S (2003) Stabilization parameters and Smagorinsky turbulence model. J Appl Mech 60:2–9. https://doi.org/10.1115/1.1526569

163. Hsu M-C, Bazilevs Y, Calo VM, Tezduyar TE, Hughes TJR (2010) A multiscale finite element formulation with discontinuity capturing for turbulence models with dominant reactionlike terms. J Appl Mech 76:021211. https://doi.org/10.1115/1.3062967

164. Rispoli F, Saavedra R, Menichini F, Tezduyar TE (2009) Computation of inviscid supersonic flows around cylinders and spheres with the SUPG formulation and YZβ shock-capturing. Comput Mech 38:469–481. https://doi.org/10.1007/s00466-005-0025-6

165. Corsini A, Menichini C, Rispoli F, Santoriello A, Tezduyar TE (2009) A multiscale finite element formulation with discontinuity capturing for turbulence models with dominant reactionlike terms. J Appl Mech 76:021210. https://doi.org/10.1115/1.3057496

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

167. Corsini A, Rispoli F, Tezduyar TE (2011) Stabilized finite element computation of NOx emission in engine combustors. Int J Numer Methods Fluids 65:254–270. https://doi.org/10.1002/fld.2451

168. Corsini A, Rispoli F, Tezduyar TE (2012) Computer modeling of wave-energy air turbines with the SUPG/PSPG formulation and discontinuity-capturing technique. J Appl Mech 79:010910. https://doi.org/10.1115/1.4005060

169. Corsini A, Rispoli F, Sheard AG, Tezduyar TE (2012) Computational analysis of noise reduction devices in axial fans with stabilized finite element formulations. Comput Mech 50:695–705. https://doi.org/10.1007/s00466-012-0789-4

170. Kler PA, Dalcin LD, Paz RR, Tezduyar TE (2013) SUPG and discontinuity-capturing methods for coupled fluid mechanics and electrochemical transport problems. Comput Mech 51:171–185. https://doi.org/10.1007/s00466-012-0712-z

171. Corsini A, Rispoli F, Sheard AG, Takizawa K, Tezduyar TE, Venturini P (2014) A variational multiscale method for particle-cloud tracking in turbomachinery flows. Comput Mech 54:1191–1202. https://doi.org/10.1007/s00466-014-1050-0

172. Rispoli F, Delibra G, Venturini P, Corsini A, Saavedra R, Tezduyar TE (2015) Particle tracking and particle-shock interaction in compressible-flow computations with the V-SGS stabilization and YZβ shock-capturing. Comput Mech 55:1201–1209. https://doi.org/10.1007/s00466-015-1160-3

173. Cardillo L, Corsini A, Delibra G, Rispoli F, Tezduyar TE (2016) Flow analysis of a wave-energy air turbine with the SUPG/PSPG stabilization and discontinuity-capturing directional dissipation. Comput Fluids 141:184–190. https://doi.org/10.1016/j.compfluid.2016.07.011

174. Castorrini A, Corsini A, Rispoli F, Venturini P, Takizawa K, Tezduyar TE (2016) Computational analysis of wind-turbine blade rain erosion. Comput Fluids 141:175–183. https://doi.org/10.1016/j.compfluid.2016.08.013

175. Castorrini A, Corsini A, Rispoli F, Venturini P, Takizawa K, Tezduyar TE (2019) Computational analysis of performance deterioration of a wind turbine blade strip subjected to environmental erosion. Comput Mech 64:1133–1153. https://doi.org/10.1007/s00466-019-0197-0

176. Castorrini A, Venturini P, Corsini A, Rispoli F, Takizawa K, Tezduyar TE (2020) Computational analysis of particle-laden airflow erosion and experimental verification. Comput Mech 65:1549–1565. https://doi.org/10.1007/s00446-020-01834-0

177. Takizawa K, Tezduyar TE, Otoguro Y (2018) Stabilization and discontinuity-capturing parameters for space-time flow computations with finite element and isogeometric discretizations. Comput
188. Otoguro Y, Takizawa K, Tezduyar TE (2020) Element length calculation in B-spline meshes for complex geometries. Comput Mech 65:1085–1103. https://doi.org/10.1007/s00466-019-01809-w
189. Ueda Y, Otoguro Y, Takizawa K, Tezduyar TE (2020) Element-splitting-invariant local-length-scale calculation in B-spline meshes for complex geometries. Math Models Methods Appl Sci (published online). https://doi.org/10.1142/S0218202520500402
190. Shakib F, Hughes TJR, Johan Z (1991) A new finite element formulation for computational fluid dynamics: X. The compressible Euler and Navier–Stokes equations. Comput Methods Appl Mech Eng 89:141–219
191. Johnson C (1987) Numerical solution of partial differential equations by the finite element method. Cambridge University Press, Sweden
192. Denman ED, Beavers AN (1976) The matrix sign function and computations in systems. Appl Math Comput 2(1):63–94
193. Tejada-Martínez AE, Akkerman I, Bazilevs Y (2012) Large-eddy simulation of shallow water Langmuir turbulence using isogeometric analysis and the residual-based variational multiscale method. J Appl Mech 79(1):010909
194. Bazilevs Y, Kamensky D, Moutsanidis G, Shende S (2020) Residual-based shock capturing in solids. Comput Methods Appl Mech Eng 358:112638
195. Almeida RC, Galeão AC (1996) An adaptive Petrov–Galerkin formulation for the compressible Euler and Navier–Stokes equations. Comput Methods Appl Mech Eng 129(1):157–176
196. Warming RF, Beam RM, Hyett BJ (1975) Diagonalization and simultaneous symmetrization of the gas-dynamic matrices. Math Comput 29:1037–1045
197. Chung J, Hulbert GM (1993) A time integration algorithm for structural dynamics with improved numerical dissipation: the generalized-α method. J Appl Mech 60:371–75
198. Jansen KE, Whiting CH, Hulbert GM (2000) A generalized-α method for integrating the filtered Navier–Stokes equations with a stabilized finite element method. Comput Methods Appl Mech Eng 190:305–319
199. Saad Y, Schultz M (1986) GMRES: a generalized minimal residual algorithm for solving nonsymmetric linear systems. SIAM J Sci Stat Comput 7:856–869
200. Shakib F, Hughes TJR, Johan Z (1989) A multi-element preconditioning GMRES algorithm for nonsymmetric systems arising in finite element analysis. Comput Methods Appl Mech Eng 75:415–456
201. Hsu M-C, Wang C, Herrema AJ, Schillinger D, Ghoshal A, Bazilevs Y (2015) An interactive geometry modeling and parametric design platform for isogeometric analysis. Comput Math Appl 70:1481–1500
202. Boyce MP (2011) Gas turbine engineering handbook. Elsevier

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