Kenji Takizawa · Bradley Henicke · Tayfun E. Tezduyar · Ming-Chen Hsu · Yuri Bazilevs

Stabilized Space–Time Computation of Wind-Turbine Rotor Aerodynamics

Abstract We show how we use the Deforming-Spatial-Domain/Stabilized Space–Time (DSD/SST) formulation for accurate 3D computation of the aerodynamics of a wind-turbine rotor. As the test case, we use the NREL 5MW offshore baseline wind-turbine rotor. This class of computational problems are rather challenging, because they involve large Reynolds numbers and rotating turbulent flows, and computing the correct torque requires an accurate and meticulous numerical approach. We compute the problem with both the original version of the DSD/SST formulation and a recently-introduced version with an advanced turbulence model. The DSD/SST formulation with the advanced turbulence model is a space–time version of the residual-based variational multiscale method. We compare our results to those reported recently, which were obtained with the residual-based variational multiscale Arbitrary Lagrangian–Eulerian (ALE) method using NURBS for spatial discretization and which we take as the reference solution. While the original DSD/SST formulation yields torque values not far from the reference solution, the DSD/SST formulation with the variational multiscale turbulence model yields torque values very close to the reference solution.

Keywords DSD/SST formulation, Space–time variational multiscale method, Wind-turbine aerodynamics, Rotating turbulent flow, Torque values

1 Introduction

Recent worldwide re-emphasis on renewable energy and the expectation that advanced computing techniques and large-scale computing can do much in supporting the technology in this area motivated a determined focus on 3D wind-turbine simulation. For example, two recent journal articles [1; 2] document the first comprehensive effort to simulate wind-turbine rotors in 3D at full scale, including rotor-geometry definition, meshing, aerodynamic and structural modeling, and fully-coupled fluid–structure interaction (FSI) computation. Isogeometric analysis, proposed in [3] as an alternative to the standard finite element method, was employed for the bulk of the computations reported in [1; 2]. Wind-turbine simulation is a computationally very challenging class of problems, because they involve large Reynolds numbers and rotating turbulent flows, and computing the correct torque, which is of course one of the most important pieces of computer modeling, requires an accurate and meticulous numerical approach. Addressing the computational challenges involved in this class of problems has been a part of the computational mechanics research targeting flows with moving boundaries and interfaces (see, for example, [4; 5; 6; 7; 8; 9; 10; 11; 12; 13; 14; 15; 16; 17; 18; 19; 20; 21; 22; 23; 24; 25; 26; 27; 28; 29; 30; 31; 32; 33; 34; 35; 36; 37; 38; 39; 40; 41; 42; 43; 44; 45; 46; 47; 48; 49; 50; 51; 52; 53; 54; 55; 56; 57; 58; 1; 2; 59; 60; 61; 62; 63]), including FSI and flows with mechanical components in fast, linear or rotational relative motion [9; 12; 15; 18; 43].

Using the terminology and categorization used in [64], we can view a method for flow problems with moving boundaries and interfaces as an interface-tracking (moving-mesh) technique or an interface-capturing (nonmoving-mesh) technique, or a combination of the two. In interface-tracking methods, as the interface moves and the spatial domain occupied by the fluid changes its shape, the mesh moves to accommodate this shape change and to follow (i.e. "track") the interface. As mentioned in [36], moving the fluid mesh to track a fluid–solid interface enables us, at least for interfaces with reasonable geometric complexity, to control the mesh resolution near that interface and obtain accurate solutions in such critical flow regions. As also mentioned in [36], sometimes the geometric complexity of the interface may require a fluid mechanics mesh that is not affordable or not desirable or

Kenji Takizawa, Bradley Henicke and Tayfun E. Tezduyar
Mechanical Engineering, Rice University – MS 321
6100 Main Street, Houston, TX 77005, USA
E-mail: tezduyar@rice.edu

Ming-Chen Hsu and Yuri Bazilevs
Structural Engineering, University of California, San Diego
9500 Gilman Drive, La Jolla, CA 92093, USA
E-mail: yuri@ucsd.edu

just not manageable in mesh moving, and this is one of the most common reasons given for favoring a interface-capturing method. This approach can be seen as a special case of interface representation techniques where the interface geometry is somehow represented over a nonmoving fluid mechanics mesh, the main point being that the fluid mesh does not move to track the interfaces. However, as pointed out in [65], a consequence of the mesh not moving to track the interface is that for fluid-solid interfaces, independent of how accurately the interface geometry is represented, the resolution of the boundary layer will be limited by the resolution of the fluid mesh where the interface is. Therefore, for interfaces with reasonable geometric complexity, if a moving-mesh method can be used with a reasonable remeshing (see [33] for various remeshing options) cost, its fluid mechanics accuracy near the interface will be superior to that of an nonmoving-mesh method. Furthermore, as pointed out in [36], "while it is understandable that fixed-mesh methods become more favored when the interface geometric complexity appears to be too high for a moving-mesh method, we need to remember that there is a difference between making the problem computable and obtaining good fluid mechanics accuracy near the interface." Therefore, as also pointed out in [36], "it is not difficult to imagine that if we lower our expectations of good fluid mechanics accuracy near the interfaces with high geometric complexity, we can find a number of ways to make the problem computable also with moving-mesh methods, and can still expect to obtain good accuracy near the interfaces with reasonable geometric complexity." Examples of that were given in [36].

There has been significant effort in recent decades to improve the accuracy of nonmoving-mesh methods, including those based on the finite element method. For example, the Enhanced-Discretization Interface-Capturing Technique (EDICT) [66; 64] was introduced in 1997 to increase the accuracy in representing an interface. The EDICT is a multi-level finite element formulation where the trial and test functions come from two overlapping meshes — a base mesh and a higher-refinement second-level mesh built over the zones that require higher accuracy. Based on the EDICT concept, the EDICT-Clustered-Mesh-2 and EDICT-Layered-Mesh-2 were proposed [15] in 2001 as methods for increasing the mesh refinement near fluid–solid interfaces. It is clear that the EDICT-Layered-Mesh-2 is applicable whether Mesh-1 (first-level mesh) is moving to track the interface or not. It is important to realize while the EDICT-Layered-Mesh-2 is quite similar to the Chimera overset grid technique [67] in the intended functionality, the two techniques are quite different in the way the solutions over the two meshes are coupled. In the EDICT-Layered-Mesh-2, the basis-function set is a combination of the basis-function sets corresponding to Mesh-1 and Mesh-2. The Mixed Interface-Tracking/Interface-Capturing Technique (MITICT) [15] was introduced in 2001 for computation of flow problems that involve both interfaces that can be accurately tracked with a moving-mesh method and interfaces that are too complex or unsteady to be tracked and therefore require an interface-capturing technique. The MITICT was successfully tested in [68; 69]. We believe that it is only meaningful to propose additional nonmoving-mesh techniques (that are claimed to be better alternatives to moving-mesh techniques) if the categories of problems that cannot be solved with moving-mesh techniques that the new technique is targeting are clearly identified, with computed examples of complex, real-world problems in those categories. We rarely see that. We also believe that it will be a while before techniques without body-conforming meshes can meaningfully be applied to wind-turbine rotor aerodynamics.

Much of the work cited in the first paragraph is based on moving-mesh techniques, and the preferred method has mostly been the Arbitrary Lagrangian–Eulerian (ALE) formulation [4]. One of the earliest space–time formulations targeting flows with moving boundaries and interfaces is the Deforming-Spatial-Domain/Stabilized Space–Time (DSD/SST) formulation [70; 71; 72; 65]. The formulation was introduced in 1991 as a general-purpose interface-tracking technique for computation of flow problems with moving boundaries or interfaces. It is based on the Streamline-Upwind/Petrov-Galerkin (SUPG) [73] and Pressure-Stabilizing/Petrov-Galerkin (PSPG) [70; 74] methods. An earlier version of the pressure stabilization, for Stokes flows, was introduced in [75]. The stabilized space–time formulations were introduced and tested earlier by other researchers in the context of problems with fixed spatial domains (see [76]). The DSD/SST formulation has been applied to several different classes of problems involving moving boundaries and interfaces (see, for example, [5; 6; 7; 8; 9; 10; 11; 12; 13; 14; 15; 16; 18; 19; 20; 21; 22; 24; 25; 26; 27; 30; 31; 33]).

The DSD/SST formulation, like most stabilized formulations, involves stabilization parameters that play an important role in determining the accuracy of the formulation. There are various ways of defining the stabilization parameters (see, for example, [77; 70; 78; 65; 79; 80; 81; 82; 83; 84; 85; 86; 33; 87; 88; 89; 90]). The ones used with the DSD/SST formulation in recent years have mostly been those given in [65; 33].

A special mesh update method, which was later named the Shear–Slip Mesh Update Method (SSMUM), was introduced in [9] for DSD/SST computation of flows with mechanical components in fast, linear or rotational relative motion. The application in [9] was 3D computation of the aerodynamics of two high-speed trains in a tunnel, passing each other in fast, linear relative motion. The idea behind the SSMUM was to restrict the mesh deformation and remeshing to a thin layer of elements between the objects in relative motion. The mesh update at each time step can be accomplished by a "shear" deformation of the elements in this layer, followed by a "slip" in node connectivities. The slip in the node connectivities, to an extent, un-does the deformation of the elements and results in elements with better shapes than those that were shear-deformed. Because the remeshing consists of simply re-defining the node connectivities, both the projection errors and the mesh generation cost are minimized. A few years after the high-speed train computations, the SSMUM was implemented for objects in fast, rotational relative motion and applied to computation of flow past a rotating propeller [12] and flow around a helicopter with its rotor in motion [18]. Enhancements of the SSMUM that give the method more flexibility and wider scope were proposed in [15]. A newer method for objects in fast, rotational relative motion can be found in [43].

New-generation DSD/SST formulations were introduced in [33]. The new versions were named “DSD/SST-SP”, “DSD/SST-TIP1” and “DSD/SST-SV” to differentiate them from the original version introduced in [70; 71; 72], which was named “DSD/SST-DP” in [33]. The new formulations have been the core technologies of the stabilized space–time FSI (SSTFSI) technique, which was also introduced in [33]. The SSTFSI technique, supplemented with special FSI techniques targeting specific classes of problems, has been successfully applied to complex, real-world problems, such as computer modeling of the Orion Spacecraft parachutes (see [36; 91; 57; 59; 60]) and patient-specific modeling of cerebral aneurysms (see [30; 37; 44; 49; 51; 52; 92]).

The variational multiscale (VMS) method was first proposed in [93] and its application to fluid mechanics and turbulence was given in [94; 95]. Mathematical analysis of the method was performed in [96]. The residual-based instantiation of the VMS method for incompressible turbulent flow was first given in [97] and further studied in [98] with emphasis on rotating turbulent flows. A space–time version of the residual-based VMS method was introduced in [63] as a new DSD/SST formulation that has an advanced turbulence model. This was implemented specifically for DSD/SST-DP, and the new DSD/SST version was named in [63] “DSD/SST-DP-VMST” (implying the version with the variational multiscale turbulence model). To differentiate it from this new version, the original DSD/SST-DP version was named in [63] “DSD/SST-DP-SUPS” (implying the version with the SUPG/PSPG stabilization). In this paper we apply the DSD/SST-DP-VMST formulation to compute the aerodynamics of a wind-turbine rotor, specifically the NREL 5MW offshore baseline wind-turbine rotor, with the geometry coming from [1]. We compute this problem also with the DSD/SST-DP-SUPS version and compare the solutions obtained with both versions to the solution [1] obtained with the residual-based variational multiscale ALE method using NURBS. We note that the ALE solution, which we take as the reference solution, was computed with strong Dirichlet boundary conditions on the rotor, and that is the case in our computations.

The geometry construction for the wind-turbine rotor blade and hub is described in Section 2. The problem setup, mesh generation, computations, and discussion of the results are presented in Section 3. The concluding remarks are given in Section 4.

2 Geometry construction for the wind-turbine rotor blade and hub

The geometry construction for the wind-turbine rotor blade and hub we are using in the computations was described in [1]. We repeat that description here. The geometry of the rotor blade is based on the NREL 5MW offshore baseline wind turbine reported in [99]. The blade-geometry data given in [99] is summarized in Table 1. A 61 m blade is attached to a hub with radius of 2 m, making the total rotor radius, R, 63 m. The blade is composed of several airfoil types (see Figure 1), which are listed in the rightmost column of the table. The first portion of the blade is a perfect cylinder. Farther away from the root the cylinder is smoothly blended into a series of DU (Delft University) airfoils. Starting at 44.55 m from the root and all the way to the tip, the NACA64 is profile used. The remaining parameters used in Table 1 are defined in Figure 2. In Table 1, “RNodes” is the distance from the rotor center to the airfoil cross-section along the blade axis, “AeroTwist” is the twist angle of the cross-section (blades are twisted to enhance the aerodynamic performance), “Chord” is the chord length of the air-

RNodes	AT	Chord	AC	AO	Type
(m)	(°)	(m)			
2.0000	0.000	3.542	0.2500	0.50	Cylinder
2.8667	0.000	3.542	0.2500	0.50	Cylinder
5.6000	0.000	3.854	0.2218	0.44	Cylinder
8.3333	0.000	4.167	0.1883	0.38	Cylinder
11.7500	13.308	4.557	0.1465	0.30	DU40
15.8500	11.480	4.652	0.1250	0.25	DU35
19.9500	10.162	4.458	0.1250	0.25	DU35
24.0500	9.011	4.249	0.1250	0.25	DU30
28.1500	7.795	4.007	0.1250	0.25	DU25
32.2500	6.544	3.748	0.1250	0.25	DU25
36.3500	5.361	3.502	0.1250	0.25	DU21
40.4500	4.188	3.256	0.1250	0.25	DU21
44.5500	3.125	3.010	0.1250	0.25	NACA64
48.6500	2.310	2.764	0.1250	0.25	NACA64
52.7500	1.526	2.518	0.1250	0.25	NACA64
56.1667	0.863	2.313	0.1250	0.25	NACA64
58.9000	0.370	2.086	0.1250	0.25	NACA64
61.6333	0.106	1.419	0.1250	0.25	NACA64
62.9000	0.000	0.700	0.1250	0.25	NACA64

Table 1 Wind-turbine rotor geometry definition from [99]. AT: AeroTwist, AC: AeroCent, AO: AeroOrig.

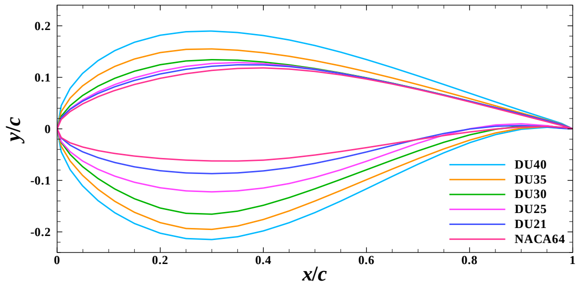

Fig. 1 Airfoil types used in the design of the wind-turbine rotor blade.

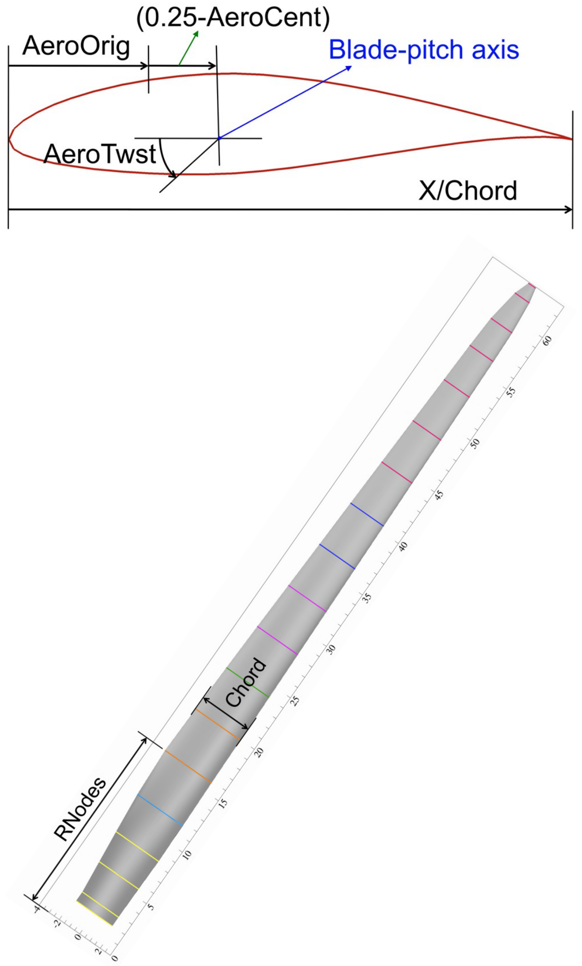

Fig. 2 Definition of the parameters used in Table 1.

foil, and "AeroOrig" is the location of the aerodynamic center. For most of the airfoil cross-sections, the aerodynamic center is positioned at 25% of the chord length from the leading edge. To transition to the cylindrical shape at the root, the aerodynamic center is gradually moved to 50% of the chord length. This is not reported in [99] but mentioned in [100].

For each cross-section, we use quadratic NURBS to represent the 2D airfoil shape. The weights of the NURBS functions are set to unity. The weights are adjusted near the root to represent the circular cross-sections exactly. The cross-sections are lofted along the blade axis direction, also using quadratic NURBS and unit weights. This geometry-construction process yields a smooth blade surface with a relatively small number of input parameters, which is an advantage of the isogeometric representation. The final blade shape is shown in Figure 3, together with the airfoil shapes. Figure 3 also shows the airfoils seen with a viewing direction parallel to the blade axis, and that illustrates the twisting of the cross-sections.

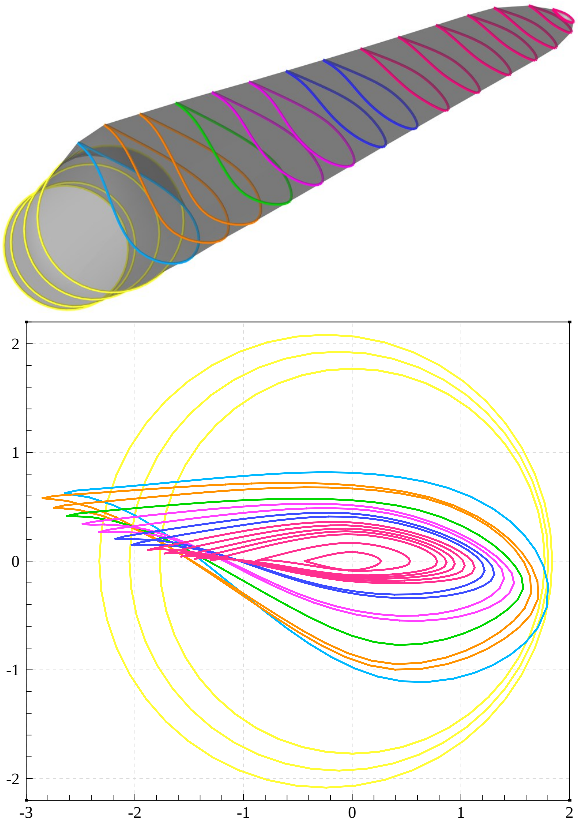

Fig. 3 Top: Airfloils superposed on the blade. Bottom: Airfoils seen with a viewing direction parallel to the blade axis, illustrating the twisting of the cross-sections. Axes units are meters.

Remark 1 *The parameter "AeroCent" is used as an input to FAST [101], which is the aerodynamics modeling software that is typically used for wind-turbine rotor computations. FAST is based on look-up tables and provides steady-state lift and drag for a blade cross-section with given airfoil type, relative wind speed, and angle of attack. The effects of the hub, trailing edge turbulence, and blade tip are modeled using empirical relationships. FAST defines* $(AeroCent - 0.25)$ *to be the fractional dis-*

tance to the aerodynamic center from the blade-pitch axis along the chordline, positive toward the trailing edge. Therefore, $AeroOrig + (0.25 - AeroCent)$ gives the location where the blade-pitch axis passes through each airfoil cross-section.

3 Computation with the DSD/SST formulation

3.1 Problem setup and mesh generation

We compute the aerodynamics of the rotor, shown in Figure 4, with a prescribed shape and speed with a rotating mesh. The wind speed is uniform at 9 m/s and the rotor speed is 1.08 rad/s, giving a tip speed ratio of 7.55 (see [102] for wind-turbine terminology). We use air properties at standard sea-level conditions. The Reynolds number (based on the chord length at $\frac{3}{4}R$ and the relative velocity there) is approximately 12 million. For computational efficiency, rotational-periodicity [59; 103; 60] is utilized so that the domain includes only one of three blades, as shown in Figure 5. The inflow, outflow and radial boundaries lie $0.5R$, $2R$ and $1.43R$ from the hub center, respectively. This can be more easily seen in Figure 6, where the inflow, outflow, and radial boundaries are the left, right and top edges, respectively, of the cut plane along the rotation axis. At the inflow boundary the velocity is set to the wind velocity, at the outflow boundary the stress vector is set to zero, and at the radial boundary the radial and tangential components of the velocity are set to zero.

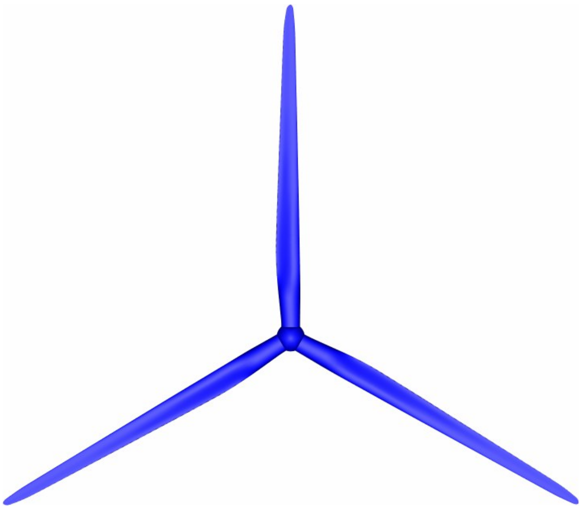

Fig. 4 Wind-turbine rotor.

The fluid volume mesh consists of 253,340 nodes and 1,475,175 four-node tetrahedral elements, with 9,268 nodes and 18,492 triangles on the rotor surface as shown in Figures 5 and 6. Along the rotor axis, we have 22 nodes upstream and 43 nodes downstream. To generate the triangular mesh on the rotor surface, we started with a quadrilateral surface mesh generated by interpolating the NURBS geometry at each knot intersection. We subdivided each quadrilateral element into triangles and then made minor modifications to improve the mesh quality near the hub. Figure 7 shows the surface mesh. Each periodic boundary contains 1,430 nodes and 2,697 triangles. Near the rotor surface, we have 22 layers of refined mesh with first-layer thickness of 1 cm and a progression factor of 1.1. The boundary layer mesh at $\frac{3}{4}R$ is shown in Figure 8.

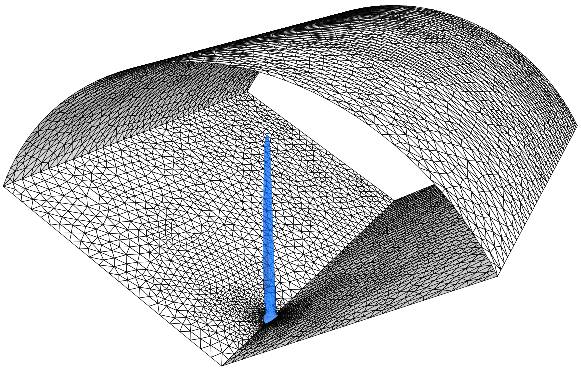

Fig. 5 Rotationally-periodic domain with wind-turbine blade shown in blue.

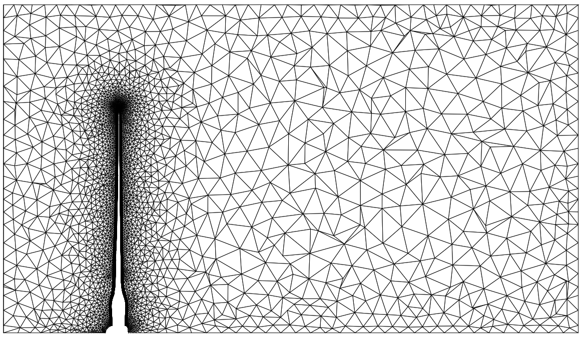

Fig. 6 Cut plane of the fluid volume mesh along rotor axis.

3.2 Computation

We compute the problem with the DSD/SST-DP-SUPS and DSD/SST-DP-VMST [63] techniques. For the VMST technique, we test both definitions of "ν_C" given in [63]. We will call the one given by Eq. (17) in [63] "TC2", and the one given by Eq. (18), "TGI". With the SUPS technique, we do not use the "LSIC" stabilization. In solving the linear equation systems involved at every nonlinear iteration, the GMRES search technique [104] is used with a diagonal preconditioner. The computation is carried out in a parallel computing environment, using PC clusters, specifically, IBM Power 7, with one or two nodes, 32 cores each. The mesh is partitioned to enhance the par-

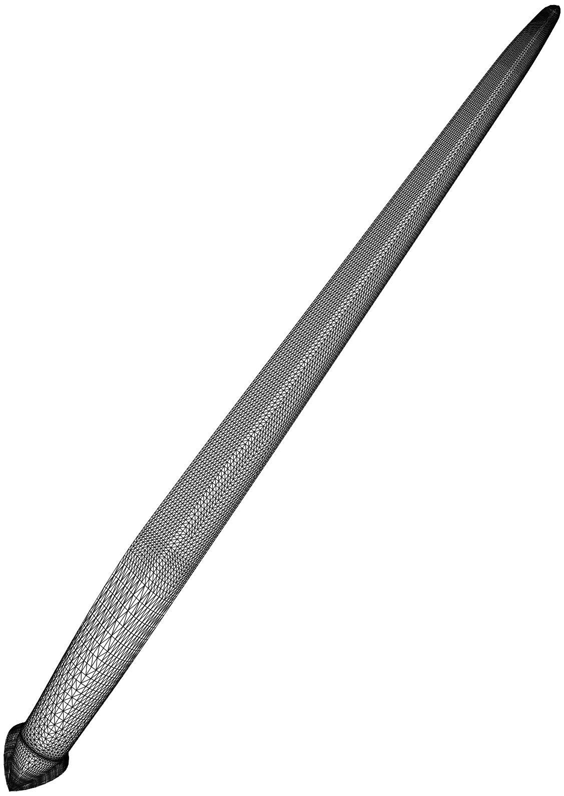

Fig. 7 Rotor surface mesh.

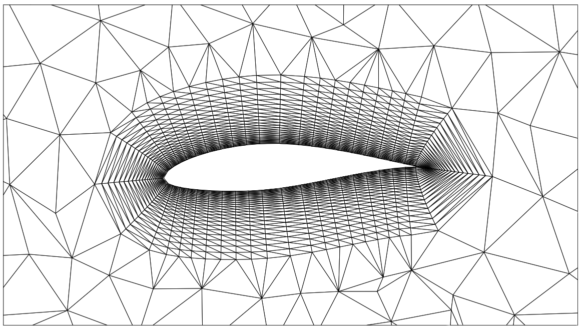

Fig. 8 Boundary layer mesh at $\frac{3}{4}R$.

allel efficiency of the computations. Mesh partitioning is based on the METIS algorithm [105]. The time-step size is 4.67×10^{-4} s. The number of nonlinear iterations per time step is 3 with 30, 60 and 500 GMRES iterations for the first, second and third nonlinear iterations, respectively. The nonlinear-iteration residuals we monitor show that we are using sufficient number of nonlinear iterations per time step. The number of GMRES iterations we use at the third nonlinear iteration is the same as the number of GMRES iterations used at every nonlinear iteration of the DSD/SST computations reported in [1]. However, we believe that the computations we report here can also be carried out with fewer number of GMRES iterations, perhaps as low as 150 even at the third nonlinear iteration.

Prior to the computations reported here, we performed a series of brief computations with the DSD/SST-DP-SUPS technique, starting from a lower Reynolds number and gradually reaching the actual Reynolds number. This solution is used as the initial condition also for the computations with the DSD/SST-DP-VMST technique. The purpose is to generate a divergence-free and reasonable flow field at this Reynolds number. We note that it was especially difficult with the VMST option to start from non-physical conditions, such as setting all nodes except those on the blade to the inflow velocity.

Figure 9 shows the magnitude of the vorticity at t = 1.0 s, computed with the DSD/SST-DP-VMST (TGI) technique. Figure 10 shows the time history of the aerodynamic torque computed with the DSD/SST-DP-SUPS and DSD/SST-DP-VMST (TGI) techniques. The blade is segmented into 18 spanwise "patches" to investigate how the aerodynamic torque distribution varies along the blade span. To better resolve the torque gradient at the blade root and tip, Patch 1 has 0.366 the span length of the middle patches, and Patches 2–4 and 16–18 have $\frac{2}{3}$. The airfoil types for the patches are shown in Table 2. The patches and for the DSD/SST-DP-SUPS and DSD/SST-DP-VMST (TGI) techniques, the torque contribution from each patch for a single blade at $t = 1.0$ s are shown in Figure 11.

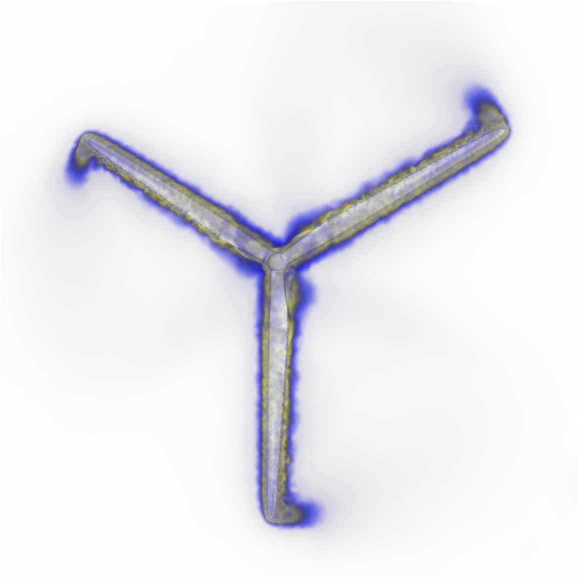

Fig. 9 Vorticity magnitude at $t = 1.0$ s, computed with the DSD/SST-DP-VMST (TGI) technique.

In Figure 12, we compare the time histories of the aerodynamic torque computed with the DSD/SST-DP-VMST (TGI) and DSD/SST-DP-VMST (TC2) techniques. In Figure 13, we compare the DSD/SST-DP-VMST (TGI) and DSD/SST-DP-VMST (TC2) techniques in terms of

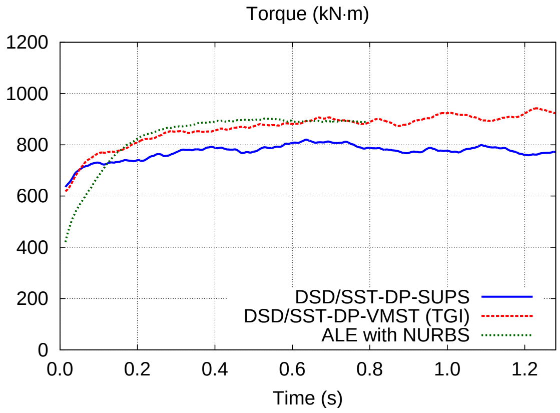

Fig. 10 Time history of the aerodynamic torque generated by a single blade. Computed with the DSD/SST-DP-SUPS and DSD/SST-DP-VMST (TGI) techniques. The torque curve labeled "ALE with NURBS" is from [1].

Patch	Type
1	Cylinder
2	Cylinder
3	Cylinder
4	Cylinder–DU40
5	Cylinder–DU40, DU40–DU35
6	DU35–DU30
7	DU35–DU30
8	DU35–DU30, DU30–DU25
9	DU30–DU25
10	DU25–DU21
11	DU21
12	DU21–NACA64
13	DU21–NACA64
14	NACA64
15	NACA64
16	NACA64
17	NACA64
18	NACA64

Table 2 Airfoil types for the patches. Some patches contain zones of transition between different airfoil types.

the torque contribution from each patch for a single blade at t = 1.0 s.

Figures 14 and 15 show the pressure coefficients at t = 1.0 s for Patch 12 (at 0.65R) and Patch 16 (at 0.90R), respectively. For most of the patches, the angle of attack and Reynolds number do not vary much from one patch to another. For example, the angle of attack and Reynolds number are 7.4° and 9.9×10^6 at 0.65R for Patch 12 (at 0.65R) and 7.6° and 9.6×10^6 for Patch 16 (at 0.90R).

3.3 Discussion

We take the results obtained with the variational multiscale method using NURBS [1] as the reference solution.

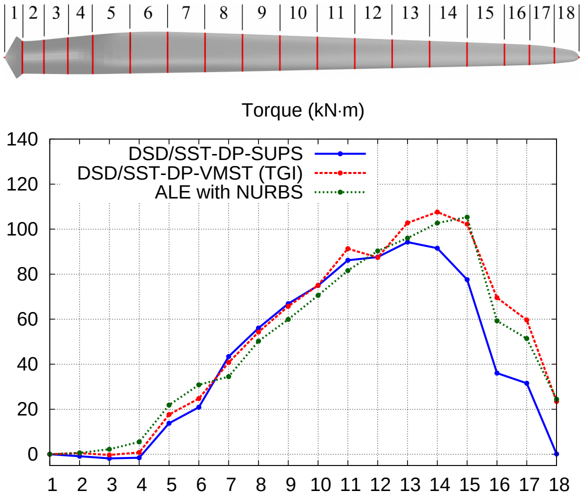

Fig. 11 Top: Patches along the blade. Bottom: The aerodynamic torque contribution from each patch at t = 1.0 s. Computed with the DSD/SST-DP-SUPS and DSD/SST-DP-VMST (TGI) techniques. We note that the curve labeled "ALE with NURBS" is from [1] and corresponds to t = 0.8 s.

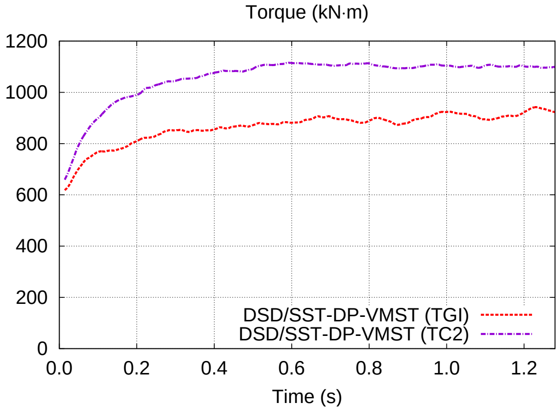

Fig. 12 Time history of the aerodynamic torque generated by a single blade. Comparison between the DSD/SST-DP-VMST (TGI) and DSD/SST-DP-VMST (TC2) techniques.

As can be seen in Figures 10 and 11, while the torque values obtained with the DSD/SST-DP-SUPS technique are not far from the reference solution, the DSD/SST-DP-VMST (TGI) technique yields torque values very close to the reference solution. For Patches 14 to 18, there are clear differences between the SUPS and VMST (TGI) solutions. Since the angle of attack and Reynolds number do not vary much from one patch to another, the differences between the SUPS and VMST solutions must come from the airfoil geometries. In other words, the SUPS and VMST torque values are different for the patches with the NACA geometry, but comparable for the patches with DU geometries. We note that the main reason behind the higher VMST torque is the wider low-pressure region on the upper surface of the NACA64 geometry, as can be seen in Figure 15. The lower pressure indicates

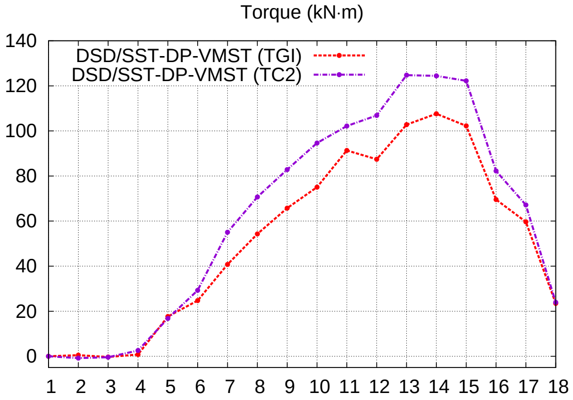

Fig. 13 Aerodynamic torque contribution from each patch at $t = 1.0$ s. Comparison between the DSD/SST-DP-VMST (TGI) and DSD/SST-DP-VMST (TC2) techniques.

that the flow is attached; thus, the DSD/SST-DP-VMST technique can represent a turbulent boundary layer well.

Remark 2 *As can be seen in Figures 12 and 13, the choice of ν_{C} is making a significant difference in the torque values computed with the DSD/SST-DP-VMST technique. This calls for further investigation.*

Remark 3 *We believe that the torque level reached with the TC2 definition of ν_{C} may still not be unreasonable, because we are computing with a computational domain that extends only 1.43R in the radial direction. This also calls for further investigation.*

When it comes to accurate representation of the eddies in the turbulent boundary layer, the mesh refinement level we use here, shown in Figures 7 and 8, is not sufficient for a large-eddy simulation (LES) type computation. Although the DSD/SST-DP-VMST technique does not have any additional turbulence modeling equations, it is still working well on a mesh that would normally be suitable for a Reynolds-averaged Navier–Stokes (RANS) type computation. This justifies our expectation that the DSD/SST-DP-VMST technique can also be used with meshes that would normally be suitable for a detached-eddy simulation (DES) type computation [106]. The same observation was made in [107] for the residual-based variational multiscale method using NURBS [1].

4 Concluding remarks

We applied the DSD/SST formulation to computation of the aerodynamics of a wind-turbine rotor, specifically the NREL 5MW offshore baseline wind-turbine rotor. We computed the problem with both the DSD/SST-DP-SUPS formulation, which is the original DSD/SST version with the SUPG and PSPG stabilizations, and the DSD/SST-DP-VMST formulation, which is a recently-introduced version that was derived as a space–time version of the residual-based variational multiscale method. We compared the solutions obtained with both versions to the solution obtained with the residual-based variational multiscale ALE method using NURBS, which we took as the reference solution. We based our comparisons on the torque values obtained because computing the correct torque, which is obviously crucial in computer modeling of wind-turbine rotor aerodynamics, requires an accurate and meticulous numerical approach. For the VMST technique, we tested both the “TGI” and “TC2” definitions of “ν_{C}”. While the DSD/SST-DP-SUPS formulation yielded torque values not far from the reference solution, the DSD/SST-DP-VMST (TGI) formulation yielded torque values very close to the reference so-

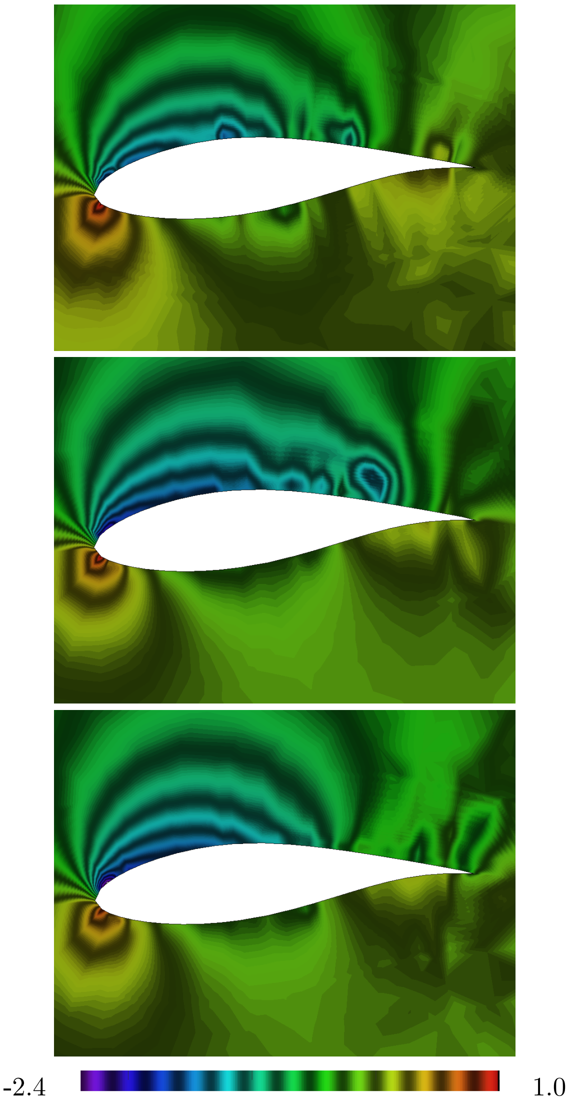

Fig. 14 Pressure coefficient at $t = 1.0$ s for Patch 12 (at $0.65R$). Top: DSD/SST-DP-SUPS. Middle: DSD/SST-DP-VMST (TGI). Bottom: DSD/SST-DP-VMST (TC2).

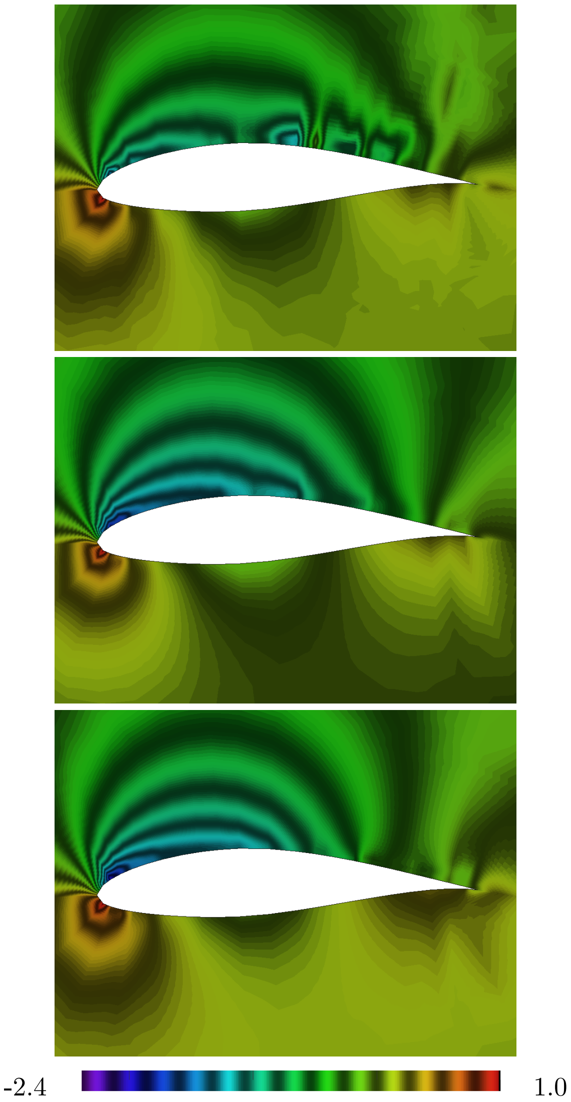

Fig. 15 Pressure coefficient at $t = 1.0$ s for Patch 16 (at $0.90R$). Top: DSD/SST-DP-SUPS. Middle: DSD/SST-DP-VMST (TGI). Bottom: DSD/SST-DP-VMST (TC2).

lution. We also observed that the choice of ν_C is making a significant difference in the torque values computed with the DSD/SST-DP-VMST technique, which we believe calls for further investigation. Overall, we demonstrated that we now have a reliable computational tool for modeling wind-turbine rotor aerodynamics, which enables us study rotor designs using 3D unsteady wind conditions (e.g., wind gusts and inflow turbulence).

Acknowledgment
This work was supported in part by the Rice Computational Research Cluster funded by NSF Grant CNS-0821727. Method analysis and evaluation components of this work were supported also in part by ARO Grant W911NF-09-1-0346. We thank Samuel E. Wright III whose previous work on this project contributed to mesh generation and data reduction. M.-C. Hsu and Y. Bazilevs were partially supported by the Los Alamos–UC San Diego Educational Collaboration Fellowship. Y. Bazilevs would also like to acknowledge the support of the Hellman Fellowship.

References

1. Y. Bazilevs, M.-C. Hsu, I. Akkerman, S. Wright, K. Takizawa, B. Henicke, T. Spielman, and T.E. Tezduyar, “3D simulation of wind turbine rotors at full scale. Part I: Geometry modeling and aerodynamics”, *International Journal for Numerical Methods in Fluids*, **65** (2011) 207–235.
2. Y. Bazilevs, M.-C. Hsu, J. Kiendl, R. Wüchner, and K.-U. Bletzinger, “3D simulation of wind turbine rotors at full scale. Part II: Fluid–structure interaction modeling with composite blades”, *International Journal for Numerical Methods in Fluids*, **65** (2011) 236–253.
3. T.J.R. Hughes, J.A. Cottrell, and Y. Bazilevs, “Isogeometric analysis: CAD, finite elements, NURBS, exact geometry, and mesh refinement”, *Computer Methods in Applied Mechanics and Engineering*, **194** (2005) 4135–4195.
4. T.J.R. Hughes, W.K. Liu, and T.K. Zimmermann, “Lagrangian–Eulerian finite element formulation for incompressible viscous flows”, *Computer Methods in Applied Mechanics and Engineering*, **29** (1981) 329–349.
5. T. Tezduyar, S. Aliabadi, M. Behr, A. Johnson, and S. Mittal, “Parallel finite-element computation of 3D flows”, *Computer*, **26** (1993) 27–36.
6. T.E. Tezduyar, S.K. Aliabadi, M. Behr, and S. Mittal, “Massively parallel finite element simulation of compressible and incompressible flows”, *Computer Methods in Applied Mechanics and Engineering*, **119** (1994) 157–177.
7. S. Mittal and T.E. Tezduyar, “Massively parallel finite element computation of incompressible flows involving fluid-body interactions”, *Computer Methods in Applied Mechanics and Engineering*, **112** (1994) 253–282.
8. S. Mittal and T.E. Tezduyar, “Parallel finite element simulation of 3D incompressible flows – Fluid-structure interactions”, *International Journal for Numerical Methods in Fluids*, **21** (1995) 933–953.
9. T. Tezduyar, S. Aliabadi, M. Behr, A. Johnson, V. Kalro, and M. Litke, “Flow simulation and high performance computing”, *Computational Mechanics*, **18** (1996) 397–412.
10. A.A. Johnson and T.E. Tezduyar, “Parallel computation of incompressible flows with complex geometries”, *International Journal for Numerical Methods in Fluids*, **24** (1997) 1321–1340.
11. A.A. Johnson and T.E. Tezduyar, “Advanced mesh generation and update methods for 3D flow simulations”, *Computational Mechanics*, **23** (1999) 130–143.
12. M. Behr and T. Tezduyar, “The Shear-Slip Mesh Update Method”, *Computer Methods in Applied Mechanics and Engineering*, **174** (1999) 261–274.
13. V. Kalro and T.E. Tezduyar, “A parallel 3D computational method for fluid–structure interactions in parachute systems”, *Computer Methods in Applied Mechanics and Engineering*, **190** (2000) 321–332.
14. K. Stein, R. Benney, V. Kalro, T.E. Tezduyar, J. Leonard, and M. Accorsi, “Parachute fluid–structure interactions: 3-D Computation”, *Computer Methods in Applied Mechanics and Engineering*, **190** (2000) 373–386.

15. T.E. Tezduyar, "Finite element methods for flow problems with moving boundaries and interfaces", *Archives of Computational Methods in Engineering*, **8** (2001) 83–130.
16. T. Tezduyar and Y. Osawa, "Fluid–structure interactions of a parachute crossing the far wake of an aircraft", *Computer Methods in Applied Mechanics and Engineering*, **191** (2001) 717–726.
17. R. Ohayon, "Reduced symmetric models for modal analysis of internal structural-acoustic and hydroelastic-sloshing systems", *Computer Methods in Applied Mechanics and Engineering*, **190** (2001) 3009–3019.
18. M. Behr and T. Tezduyar, "Shear-slip mesh update in 3D computation of complex flow problems with rotating mechanical components", *Computer Methods in Applied Mechanics and Engineering*, **190** (2001) 3189–3200.
19. K. Stein, T. Tezduyar, and R. Benney, "Mesh moving techniques for fluid–structure interactions with large displacements", *Journal of Applied Mechanics*, **70** (2003) 58–63.
20. K. Stein, T.E. Tezduyar, and R. Benney, "Automatic mesh update with the solid-extension mesh moving technique", *Computer Methods in Applied Mechanics and Engineering*, **193** (2004) 2019–2032.
21. R. Torii, M. Oshima, T. Kobayashi, K. Takagi, and T.E. Tezduyar, "Influence of wall elasticity on image-based blood flow simulation", *Japan Society of Mechanical Engineers Journal Series A*, **70** (2004) 1224–1231, in Japanese.
22. T.E. Tezduyar, S. Sathe, R. Keedy, and K. Stein, "Space–time techniques for finite element computation of flows with moving boundaries and interfaces", in S. Gallegos, I. Herrera, S. Botello, F. Zarate, and G. Ayala, editors, *Proceedings of the III International Congress on Numerical Methods in Engineering and Applied Science*, CD-ROM, Monterrey, Mexico, 2004.
23. E.H. van Brummelen and R. de Borst, "On the nonnormality of subiteration for a fluid-structure interaction problem", *SIAM Journal on Scientific Computing*, **27** (2005) 599–621.
24. T.E. Tezduyar, S. Sathe, R. Keedy, and K. Stein, "Space–time finite element techniques for computation of fluid–structure interactions", *Computer Methods in Applied Mechanics and Engineering*, **195** (2006) 2002–2027.
25. T.E. Tezduyar, S. Sathe, and K. Stein, "Solution techniques for the fully-discretized equations in computation of fluid–structure interactions with the space–time formulations", *Computer Methods in Applied Mechanics and Engineering*, **195** (2006) 5743–5753.
26. R. Torii, M. Oshima, T. Kobayashi, K. Takagi, and T.E. Tezduyar, "Computer modeling of cardiovascular fluid–structure interactions with the Deforming-Spatial-Domain/Stabilized Space–Time formulation", *Computer Methods in Applied Mechanics and Engineering*, **195** (2006) 1885–1895.
27. R. Torii, M. Oshima, T. Kobayashi, K. Takagi, and T.E. Tezduyar, "Fluid–structure interaction modeling of aneurysmal conditions with high and normal blood pressures", *Computational Mechanics*, **38** (2006) 482–490.
28. Y. Bazilevs, V.M. Calo, Y. Zhang, and T.J.R. Hughes, "Isogeometric fluid–structure interaction analysis with applications to arterial blood flow", *Computational Mechanics*, **38** (2006) 310–322.
29. R.A. Khurram and A. Masud, "A multiscale/stabilized formulation of the incompressible Navier–Stokes equations for moving boundary flows and fluid–structure interaction", *Computational Mechanics*, **38** (2006) 403–416.
30. T.E. Tezduyar, S. Sathe, T. Cragin, B. Nanna, B.S. Conklin, J. Pausewang, and M. Schwaab, "Modeling of fluid–structure interactions with the space–time finite elements: Arterial fluid mechanics", *International Journal for Numerical Methods in Fluids*, **54** (2007) 901–922.
31. R. Torii, M. Oshima, T. Kobayashi, K. Takagi, and T.E. Tezduyar, "Influence of wall elasticity in patient-specific hemodynamic simulations", *Computers & Fluids*, **36** (2007) 160–168.
32. T. Sawada and T. Hisada, "Fluid–structure interaction analysis of the two dimensional flag-in-wind problem by an interface tracking ALE finite element method", *Computers & Fluids*, **36** (2007) 136–146.
33. T.E. Tezduyar and S. Sathe, "Modeling of fluid–structure interactions with the space–time finite elements: Solution techniques", *International Journal for Numerical Methods in Fluids*, **54** (2007) 855–900.
34. R. Torii, M. Oshima, T. Kobayashi, K. Takagi, and T.E. Tezduyar, "Numerical investigation of the effect of hypertensive blood pressure on cerebral aneurysm — Dependence of the effect on the aneurysm shape", *International Journal for Numerical Methods in Fluids*, **54** (2007) 995–1009.
35. M. Manguoglu, A.H. Sameh, T.E. Tezduyar, and S. Sathe, "A nested iterative scheme for computation of incompressible flows in long domains", *Computational Mechanics*, **43** (2008) 73–80.
36. T.E. Tezduyar, S. Sathe, J. Pausewang, M. Schwaab, J. Christopher, and J. Crabtree, "Interface projection techniques for fluid–structure interaction modeling with moving-mesh methods", *Computational Mechanics*, **43** (2008) 39–49.
37. T.E. Tezduyar, S. Sathe, M. Schwaab, and B.S. Conklin, "Arterial fluid mechanics modeling with the stabilized space–time fluid–structure interaction technique", *International Journal for Numerical Methods in Fluids*, **57** (2008) 601–629.
38. R. Torii, M. Oshima, T. Kobayashi, K. Takagi, and T.E. Tezduyar, "Fluid–structure interaction modeling of a patient-specific cerebral aneurysm: Influence of structural modeling", *Computational Mechanics*, **43** (2008) 151–159.
39. Y. Bazilevs, V.M. Calo, T.J.R. Hughes, and Y. Zhang, "Isogeometric fluid–structure interaction: theory, algorithms, and computations", *Computational Mechanics*, **43** (2008) 3–37.
40. J.G. Isaksen, Y. Bazilevs, T. Kvamsdal, Y. Zhang, J.H. Kaspersen, K. Waterloo, B. Romner, and T. Ingebrigtsen, "Determination of wall tension in cerebral artery aneurysms by numerical simulation", *Stroke*, **39** (2008) 3172–3178.
41. U. Kuttler and W.A. Wall, "Fixed-point fluid–structure interaction solvers with dynamic relaxation", *Computational Mechanics*, **43** (2008) 61–72.
42. W.G. Dettmer and D. Peric, "On the coupling between fluid flow and mesh motion in the modelling of fluid–structure interaction", *Computational Mechanics*, **43** (2008) 81–90.
43. Y. Bazilevs and T.J.R. Hughes, "NURBS-based isogeometric analysis for the computation of flows about rotating components", *Computational Mechanics*, **43** (2008) 143–150.
44. T.E. Tezduyar, M. Schwaab, and S. Sathe, "Sequentially-Coupled Arterial Fluid–Structure Interaction (SCAFSI) technique", *Computer Methods in Applied Mechanics and Engineering*, **198** (2009) 3524–3533.
45. R. Torii, M. Oshima, T. Kobayashi, K. Takagi, and T.E. Tezduyar, "Fluid–structure interaction modeling

of blood flow and cerebral aneurysm: Significance of artery and aneurysm shapes", *Computer Methods in Applied Mechanics and Engineering*, **198** (2009) 3613–3621.

46. M. Manguoglu, A.H. Sameh, F. Saied, T.E. Tezduyar, and S. Sathe, "Preconditioning techniques for nonsymmetric linear systems in the computation of incompressible flows", *Journal of Applied Mechanics*, **76** (2009) 021204.
47. Y. Bazilevs, J.R. Gohean, T.J.R. Hughes, R.D. Moser, and Y. Zhang, "Patient-specific isogeometric fluid–structure interaction analysis of thoracic aortic blood flow due to implantation of the Jarvik 2000 left ventricular assist device", *Computer Methods in Applied Mechanics and Engineering*, **198** (2009) 3534–3550.
48. Y. Bazilevs, M.-C. Hsu, D. Benson, S. Sankaran, and A. Marsden, "Computational fluid–structure interaction: Methods and application to a total cavopulmonary connection", *Computational Mechanics*, **45** (2009) 77–89.
49. K. Takizawa, J. Christopher, T.E. Tezduyar, and S. Sathe, "Space–time finite element computation of arterial fluid–structure interactions with patient-specific data", *International Journal for Numerical Methods in Biomedical Engineering*, **26** (2010) 101–116.
50. T.E. Tezduyar, K. Takizawa, and J. Christopher, "Multiscale Sequentially-Coupled Arterial Fluid–Structure Interaction (SCAFSI) technique", in S. Hartmann, A. Meister, M. Schaefer, and S. Turek, editors, *International Workshop on Fluid–Structure Interaction — Theory, Numerics and Applications*, 231–252, Kassel University Press, 2009.
51. T.E. Tezduyar, K. Takizawa, C. Moorman, S. Wright, and J. Christopher, "Multiscale sequentially-coupled arterial FSI technique", *Computational Mechanics*, **46** (2010) 17–29.
52. K. Takizawa, C. Moorman, S. Wright, J. Christopher, and T.E. Tezduyar, "Wall shear stress calculations in space–time finite element computation of arterial fluid–structure interactions", *Computational Mechanics*, **46** (2010) 31–41.
53. R. Torii, M. Oshima, T. Kobayashi, K. Takagi, and T.E. Tezduyar, "Influence of wall thickness on fluid–structure interaction computations of cerebral aneurysms", *International Journal for Numerical Methods in Biomedical Engineering*, **26** (2010) 336–347.
54. M. Manguoglu, K. Takizawa, A.H. Sameh, and T.E. Tezduyar, "Solution of linear systems in arterial fluid mechanics computations with boundary layer mesh refinement", *Computational Mechanics*, **46** (2010) 83–89.
55. R. Torii, M. Oshima, T. Kobayashi, K. Takagi, and T.E. Tezduyar, "Role of 0D peripheral vasculature model in fluid–structure interaction modeling of aneurysms", *Computational Mechanics*, **46** (2010) 43–52.
56. Y. Bazilevs, M.-C. Hsu, Y. Zhang, W. Wang, X. Liang, T. Kvamsdal, R. Brekken, and J. Isaksen, "A fully-coupled fluid–structure interaction simulation of cerebral aneurysms", *Computational Mechanics*, **46** (2010) 3–16.
57. T.E. Tezduyar, K. Takizawa, C. Moorman, S. Wright, and J. Christopher, "Space–time finite element computation of complex fluid–structure interactions", *International Journal for Numerical Methods in Fluids*, **64** (2010) 1201–1218.
58. Y. Bazilevs, M.-C. Hsu, Y. Zhang, W. Wang, T. Kvamsdal, S. Hentschel, and J. Isaksen, "Computational fluid–structure interaction: Methods and application to cerebral aneurysms", *Biomechanics and Modeling in Mechanobiology*, **9** (2010) 481–498.
59. K. Takizawa, C. Moorman, S. Wright, T. Spielman, and T.E. Tezduyar, "Fluid–structure interaction modeling and performance analysis of the Orion spacecraft parachutes", *International Journal for Numerical Methods in Fluids*, **65** (2011) 271–285.
60. K. Takizawa, S. Wright, C. Moorman, and T.E. Tezduyar, "Fluid–structure interaction modeling of parachute clusters", *International Journal for Numerical Methods in Fluids*, **65** (2011) 286–307.
61. M. Manguoglu, K. Takizawa, A.H. Sameh, and T.E. Tezduyar, "Nested and parallel sparse algorithms for arterial fluid mechanics computations with boundary layer mesh refinement", *International Journal for Numerical Methods in Fluids*, **65** (2011) 135–149.
62. T.E. Tezduyar, K. Takizawa, T. Brummer, and P.R. Chen, "Space–time fluid–structure interaction modeling of patient-specific cerebral aneurysms", *International Journal for Numerical Methods in Biomedical Engineering*, **27** (2011) 1665–1710.
63. K. Takizawa and T.E. Tezduyar, "Multiscale space–time fluid–structure interaction techniques", *Computational Mechanics*, **48** (2011) 247–267.
64. T. Tezduyar, S. Aliabadi, and M. Behr, "Enhanced-Discretization Interface-Capturing Technique (EDICT) for computation of unsteady flows with interfaces", *Computer Methods in Applied Mechanics and Engineering*, **155** (1998) 235–248.
65. T.E. Tezduyar, "Computation of moving boundaries and interfaces and stabilization parameters", *International Journal for Numerical Methods in Fluids*, **43** (2003) 555–575.
66. T. Tezduyar, S. Aliabadi, and M. Behr, "Enhanced-Discretization Interface-Capturing Technique", in Y. Matsumoto and A. Prosperetti, editors, *ISAC '97 High Performance Computing on Multiphase Flows*, 1–6, Japan Society of Mechanical Engineers, 1997.
67. J.A. Benek, P.G. Buning, and J.L. Steger, "A 3-D Chimera grid embedding technique", Paper 85-1523, AIAA, 1985.
68. J.E. Akin, T.E. Tezduyar, and M. Ungor, "Computation of flow problems with the mixed interface-tracking/interface-capturing technique (MITICT)", *Computers & Fluids*, **36** (2007) 2–11.
69. M.A. Cruchaga, D.J. Celentano, and T.E. Tezduyar, "A numerical model based on the Mixed Interface-Tracking/Interface-Capturing Technique (MITICT) for flows with fluid–solid and fluid–fluid interfaces", *International Journal for Numerical Methods in Fluids*, **54** (2007) 1021–1030.
70. T.E. Tezduyar, "Stabilized finite element formulations for incompressible flow computations", *Advances in Applied Mechanics*, **28** (1992) 1–44.
71. T.E. Tezduyar, M. Behr, and J. Liou, "A new strategy for finite element computations involving moving boundaries and interfaces – the deforming-spatial-domain/space–time procedure: I. The concept and the preliminary numerical tests", *Computer Methods in Applied Mechanics and Engineering*, **94** (1992) 339–351.
72. T.E. Tezduyar, M. Behr, S. Mittal, and J. Liou, "A new strategy for finite element computations involving moving boundaries and interfaces – the deforming-spatial-domain/space–time procedure: II. Computation of free-surface flows, two-liquid flows, and flows with drifting cylinders", *Computer Methods in Applied Mechanics and Engineering*, **94** (1992) 353–371.
73. A.N. Brooks and T.J.R. Hughes, "Streamline upwind/Petrov-Galerkin formulations for convection dominated flows with particular emphasis on the incompressible Navier-Stokes equations", *Com-*

puter Methods in Applied Mechanics and Engineering, **32** (1982) 199–259.

74. T.E. Tezduyar, S. Mittal, S.E. Ray, and R. Shih, "Incompressible flow computations with stabilized bilinear and linear equal-order-interpolation velocity-pressure elements", *Computer Methods in Applied Mechanics and Engineering*, **95** (1992) 221–242.
75. T.J.R. Hughes, L.P. Franca, and M. Balestra, "A new finite element formulation for computational fluid dynamics: V. Circumventing the Babuška–Brezzi condition: A stable Petrov–Galerkin formulation of the Stokes problem accommodating equal-order interpolations", *Computer Methods in Applied Mechanics and Engineering*, **59** (1986) 85–99.
76. T.J.R. Hughes and G.M. Hulbert, "Space–time finite element methods for elastodynamics: formulations and error estimates", *Computer Methods in Applied Mechanics and Engineering*, **66** (1988) 339–363.
77. T.E. Tezduyar and Y.J. Park, "Discontinuity capturing finite element formulations for nonlinear convection-diffusion-reaction equations", *Computer Methods in Applied Mechanics and Engineering*, **59** (1986) 307–325.
78. T.E. Tezduyar and Y. Osawa, "Finite element stabilization parameters computed from element matrices and vectors", *Computer Methods in Applied Mechanics and Engineering*, **190** (2000) 411–430.
79. J.E. Akin, T. Tezduyar, M. Ungor, and S. Mittal, "Stabilization parameters and Smagorinsky turbulence model", *Journal of Applied Mechanics*, **70** (2003) 2–9.
80. T.E. Tezduyar, "Finite element methods for fluid dynamics with moving boundaries and interfaces", in E. Stein, R.D. Borst, and T.J.R. Hughes, editors, *Encyclopedia of Computational Mechanics*, Volume 3: Fluids, Chapter 17, John Wiley & Sons, 2004.
81. J.E. Akin and T.E. Tezduyar, "Calculation of the advective limit of the SUPG stabilization parameter for linear and higher-order elements", *Computer Methods in Applied Mechanics and Engineering*, **193** (2004) 1909–1922.
82. L. Catabriga, A.L.G.A. Coutinho, and T.E. Tezduyar, "Compressible flow SUPG parameters computed from element matrices", *Communications in Numerical Methods in Engineering*, **21** (2005) 465–476.
83. A. Corsini, F. Rispoli, A. Santoriello, and T.E. Tezduyar, "Improved discontinuity-capturing finite element techniques for reaction effects in turbulence computation", *Computational Mechanics*, **38** (2006) 356–364.
84. L. Catabriga, A.L.G.A. Coutinho, and T.E. Tezduyar, "Compressible flow SUPG parameters computed from degree-of-freedom submatrices", *Computational Mechanics*, **38** (2006) 334–343.
85. T.E. Tezduyar, "Finite elements in fluids: Stabilized formulations and moving boundaries and interfaces", *Computers & Fluids*, **36** (2007) 191–206.
86. F. Rispoli, A. Corsini, and T.E. Tezduyar, "Finite element computation of turbulent flows with the discontinuity-capturing directional dissipation (DCDD)", *Computers & Fluids*, **36** (2007) 121–126.
87. L. Catabriga, D.A.F. de Souza, A.L.G.A. Coutinho, and T.E. Tezduyar, "Three-dimensional edge-based SUPG computation of inviscid compressible flows with YZβ shock-capturing", *Journal of Applied Mechanics*, **76** (2009) 021208.
88. A. Corsini, C. Iossa, F. Rispoli, and T.E. Tezduyar, "A DRD finite element formulation for computing turbulent reacting flows in gas turbine combustors", *Computational Mechanics*, **46** (2010) 159–167.
89. M.-C. Hsu, Y. Bazilevs, V.M. Calo, T.E. Tezduyar, and T.J.R. Hughes, "Improving stability of stabilized and multiscale formulations in flow simulations at small time steps", *Computer Methods in Applied Mechanics and Engineering*, **199** (2010) 828–840.
90. A. Corsini, F. Rispoli, and T.E. Tezduyar, "Stabilized finite element computation of NOx emission in aero-engine combustors", *International Journal for Numerical Methods in Fluids*, **65** (2011) 254–270.
91. T.E. Tezduyar, S. Sathe, M. Schwaab, J. Pausewang, J. Christopher, and J. Crabtree, "Fluid–structure interaction modeling of ringsail parachutes", *Computational Mechanics*, **43** (2008) 133–142.
92. K. Takizawa, C. Moorman, S. Wright, J. Purdue, T. McPhail, P.R. Chen, J. Warren, and T.E. Tezduyar, "Patient-specific arterial fluid–structure interaction modeling of cerebral aneurysms", *International Journal for Numerical Methods in Fluids*, **65** (2011) 308–323.
93. T.J.R. Hughes, "Multiscale phenomena: Green's functions, the Dirichlet-to-Neumann formulation, subgrid scale models, bubbles, and the origins of stabilized methods", *Computer Methods in Applied Mechanics and Engineering*, **127** (1995) 387–401.
94. T.J.R. Hughes, L. Mazzei, and K.E. Jansen, "Large-eddy simulation and the variational multiscale method", *Computing and Visualization in Science*, **3** (2000) 47–59.
95. T.J.R. Hughes, A.A. Oberai, and L. Mazzei, "Large eddy simulation of turbulent channel flows by the variational multiscale method", *Physics of Fluids*, **13** (2001) 1784–1799.
96. T.J.R. Hughes and G. Sangalli, "Variational multiscale analysis: the fine-scale Green's function, projection, optimization, localization, and stabilized methods", *SIAM Journal of Numerical Analysis*, **45** (2007) 539–557.
97. Y. Bazilevs, V.M. Calo, J.A. Cottrell, T.J.R. Hughes, A. Reali, and G. Scovazzi, "Variational multiscale residual-based turbulence modeling for large eddy simulation of incompressible flows", *Computer Methods in Applied Mechanics and Engineering*, **197** (2007) 173–201.
98. Y. Bazilevs and I. Akkerman, "Large eddy simulation of turbulent Taylor–Couette flow using isogeometric analysis and the residual–based variational multiscale method", *Journal of Computational Physics*, **229** (2010) 3402–3414.
99. J. Jonkman, S. Butterfield, W. Musial, and G. Scott, "Definition of a 5-MW reference wind turbine for offshore system development", Technical Report NREL/TP-500-38060, National Renewable Energy Laboratory, 2009.
100. H.J.T. Kooijman, C. Lindenburg, D. Winkelaar, and E.L. van der Hooft, "DOWEC 6 MW pre-design: Aero-elastic modelling of the DOWEC 6 MW pre-design in PHATAS", Technical Report DOWEC-F1W2-HJK-01-046/9, 2003.
101. J.M. Jonkman and M.L. Buhl Jr., "FAST user's guide", Technical Report NREL/EL-500-38230, National Renewable Energy Laboratory, Golden, CO, 2005.
102. D.A. Spera, "Introduction to modern wind turbines", in D.A. Spera, editor, *Wind Turbine Technology: Fundamental Concepts of Wind Turbine Engineering*, 47–72, ASME Press, 1994.
103. K. Takizawa, C. Moorman, S. Wright, and T.E. Tezduyar, "Computer modeling and analysis of the Orion spacecraft parachutes", in H.-J. Bungartz, M. Mehl, and M. Schafer, editors, *Fluid–Structure Interaction II – Modelling, Simulation, Optimization*, volume 73 of *Lecture Notes in Computational Science and Engineering*, Chapter 3, 53–81, Springer, 2010.
104. Y. Saad and M. Schultz, "GMRES: A generalized minimal residual algorithm for solving nonsymmetric linear systems", *SIAM Journal of Scientific and Statistical*

Computing, **7** (1986) 856–869.

105. G. Karypis and V. Kumar, “A fast and high quality multilevel scheme for partitioning irregular graphs”, *SIAM Journal of Scientific Computing*, **20** (1998) 359–392.
106. P.R. Spalart, “Strategies for turbulence modelling and simulations”, *International Journal of Heat and Fluid Flow*, **21** (2000) 252–263.
107. Y. Bazilevs, C. Michler, V.M. Calo, and T.J.R. Hughes, “Isogeometric variational multiscale modeling of wall-bounded turbulent flows with weakly enforced boundary conditions on unstretched meshes”, *Computer Methods in Applied Mechanics and Engineering*, **199** (2010) 780–790.