

Computational Wind-Turbine Analysis with the ALE-VMS and ST-VMS Methods

Yuri Bazilevs, Kenji Takizawa, Tayfun E. Tezduyar, Ming-Chen Hsu, Nikolay Kostov and Spenser McIntyre

Abstract We provide an overview of the aerodynamic and FSI analysis of wind turbines the first three authors' teams carried out in recent years with the ALE-VMS and ST-VMS methods. The ALE-VMS method is the variational multiscale version of the Arbitrary Lagrangian–Eulerian (ALE) method. The VMS components are from the residual-based VMS (RBVMS) method. The ST-VMS method is the VMS version of the Deforming–Spatial-Domain/Stabilized Space–Time (DSD/SST) method. The techniques complementing these core methods include weak enforcement of the essential boundary conditions, NURBS-based isogeometric analysis, using NURBS basis functions in temporal representation of the rotor motion, mesh motion and also in remeshing, rotation representation with constant angular velocity, Kirchhoff–Love shell modeling of the rotor-blade structure, and full FSI coupling. The analysis cases include the aerodynamics of wind-turbine rotor and tower and the FSI that accounts for the deformation of the rotor blades. The specific wind turbines considered are NREL 5MW, NREL Phase VI and Micon 65/13M, all at full

Yuri Bazilevs
Structural Engineering, University of California, San Diego
9500 Gilman Drive, La Jolla, CA 92093, USA
e-mail: jbazilevs@ucsd.edu

Kenji Takizawa
Department of Modern Mechanical Engineering and Waseda Institute for Advanced Study
Waseda University, 1-6-1 Nishi-Waseda, Shinjuku-ku, Tokyo 169-8050, Japan

Tayfun E. Tezduyar
Mechanical Engineering, Rice University, 6100 Main Street, Houston, TX 77005, USA

Ming-Chen Hsu
Department of Mechanical Engineering, Iowa State University
2025 Black Engineering, Ames, IA 50011, USA

Nikolay Kostov and Spenser McIntyre
Mechanical Engineering, Rice University, 6100 Main Street, Houston, TX 77005, USA

scale, and our analysis for NREL Phase VI and Micon 65/13M includes comparison with the experimental data.

1 Introduction

Countries around the world are putting substantial effort into the development of wind energy technologies. The ambitious wind energy goals put pressure on the wind energy industry research and development to significantly enhance the current wind generation capabilities in a short period of time and decrease the associated costs. This calls for transformative concepts and designs (e.g., floating offshore wind turbines) that must be created and analyzed with high-precision methods and tools. These include complex-geometry, 3D, time-dependent, multi-physics predictive simulation methods and software that will play an increasingly important role as the demand for wind energy grows.

Currently most wind-turbine aerodynamics and aeroelasticity simulations are performed using low-fidelity methods, such as the Blade Element Momentum (BEM) theory for the rotor aerodynamics employed in conjunction with simplified structural models of the wind-turbine blades and tower (see, e.g., [1, 2]). These methods are very fast to implement and execute. However, the cases involving unsteady flow, turbulence, 3D details of the wind-turbine blade and tower geometry, and other similarly-important features, are beyond their range of applicability.

To obtain high-fidelity results for wind turbines, 3D modeling is essential. However, simulation of wind turbines at full scale engenders a number of challenges: the flow is fully turbulent, requiring highly accurate methods and increased grid resolution. The presence of fluid boundary layers, where turbulence is created, complicates the situation further. Wind-turbine blades are long and slender structures, with complex distribution of material properties, for which the numerical approach must have good approximation properties and avoid locking. Wind-turbine simulations involve moving and stationary components, and the fluid–structure coupling must be accurate, efficient and robust to preclude divergence of the computations. These explain the current, modest nature of the state-of-the-art in wind-turbine simulations.

Fluid–structure interaction (FSI) simulations at full scale are essential for accurate modeling of wind turbines. The motion and deformation of the wind-turbine blades depend on the wind speed and air flow, and the air flow patterns depend on the motion and deformation of the blades. In order to simulate the coupled problem, the equations governing the air flow and the blade motions and deformations need to be solved simultaneously, with proper kinematic and dynamic conditions coupling the two physical systems. Without that the modeling cannot be realistic: unsteady blade deformation affects aerodynamic efficiency and noise generation, and response to wind gusts. Flutter analysis of large blades operating in offshore environments is of great importance and cannot be accomplished without FSI.

In recent years, several attempts were made to address the above mentioned challenges and to raise the fidelity and predictability levels of wind-turbine simulations.

Standalone aerodynamics simulations of wind-turbine configurations in 3D were reported in [3–6], while standalone structural analyses of rotor blades of complex geometry and material composition, but under assumed wind-load conditions or wind-load conditions coming from separate aerodynamic computations were reported in [7–11]. In a recent work [12] it was shown that coupled FSI modeling and simulation of wind turbines is important for accurately predicting their mechanical behavior at full scale.

To address the above mentioned challenges one should employ a combination of numerical techniques, which are general, accurate, robust and efficient for the targeted class of problems. Such techniques are summarized in what follows, with some of them described in greater detail in the body of this review article.

Isogeometric Analysis (IGA), first introduced in [13] and further expanded on in [14–20], is adopted as the geometry modeling and simulation framework for wind turbines in some of the examples presented in this article. We use the IGA based on NURBS (non-uniform rational B-splines), which are more efficient than standard finite elements for representing complex, smooth geometries, such as wind-turbine blades. The IGA was successfully employed for computation of turbulent flows [21–26], nonlinear structures [10, 27–31], and FSI [32–35], and, in most cases, gave a clear advantage over standard low-order finite elements in terms of solution accuracy per-degree-of-freedom. This is in part attributable to the higher-order smoothness of the basis functions employed. Flows about rotating components are naturally handled in an isogeometric framework because all conic sections, and in particular, circular and cylindrical shapes, are represented exactly [36].

The blade structure is governed by the isogeometric rotation-free shell formulation with the aid of the bending-strip method [10]. The method is appropriate for thin-shell structures comprised of multiple C^1 - or higher-order continuous surface patches that are joined or merged with continuity no greater than C^0 . The Kirchhoff–Love shell theory that relies on higher-order continuity of the basis functions is employed in the patch interior as in [31]. Although NURBS-based IGA is employed in this work, other discretizations such as T-splines [19, 20] or subdivision surfaces [37–39], are perfectly suited for the proposed structural modeling method.

In addition, an isogeometric representation of the analysis-suitable geometry can be used in generating tetrahedral and hexahedral meshes for computations with the finite element method (FEM). In this article, we use tetrahedral meshes generated that way in wind-turbine computations with the ALE-VMS and ST-VMS methods. The ALE-VMS method [5, 34] is the variational multiscale (VMS) version of the Arbitrary Lagrangian–Eulerian (ALE) method [40]. The VMS components are from the residual-based VMS (RBVMS) method given in [21, 26, 41, 42]. The ST-VMS method [43, 44] is the VMS version of the Deforming-Spatial-Domain/Stabilized Space–Time (DSD/SST) method [45–49]. Earlier it was called “DSD/SST-VMST” (i.e. the version with the VMS turbulence model) in [43]. The original DSD/SST formulation was named “DSD/SST-SUPS” in [43] (i.e. the version with the SUPG/PSPG stabilization), which was also called “ST-SUPS” in [50].

The ALE-VMS method originated from the RBVMS formulation of incompressible turbulent flows proposed in [21] for stationary meshes, and may be thought of as

an extension of the RBVMS method to moving meshes. As such, it was presented for the first time in [34] in the context of FSI. Although ALE-VMS gave reasonably good results for several important turbulent flows, it was evident in [21, 24] that to obtain accurate results for wall-bounded turbulent flows the method required relatively fine resolution of the boundary layers. This fact makes ALE-VMS a somewhat costly technology for full-scale wall-bounded turbulent flows at high Reynolds numbers, which are characteristic of the present application. For this reason, weakly-enforced essential boundary condition formulation was introduced in [51], which significantly improved the performance of the ALE-VMS formulation in the presence of unresolved boundary layers [22, 23, 26]. The weak boundary condition formulation may be thought of as an extension of Nitsche’s method [52] to the Navier–Stokes equations of incompressible flows. Another interpretation of the weak boundary condition formulation is that it is a discontinuous Galerkin method (see, e.g., [53]), where the continuity of the basis functions is enforced everywhere in the domain interior, but not at the domain boundary.

The DSD/SST formulation was introduced in [45–47] as a general-purpose interface-tracking (moving-mesh) technique for flows with moving boundaries and interfaces, including FSI and flows with moving objects. Its stabilization components are the Streamline-Upwind/Petrov-Galerkin (SUPG) [54] and Pressure-Stabilizing/Petrov-Galerkin (PSPG) [45, 55] stabilizations. It also includes the “LSIC” (least-squares on incompressibility constraint) stabilization. Some of the earliest FSI computations with the DSD/SST formulation were reported in [56] for vortex-induced vibrations of a cylinder and in [57] for flow-induced vibrations of a flexible, cantilevered pipe (1D structure with 3D flow). The DSD/SST formulation has been used extensively in 3D computations of parachute FSI, starting with the 3D computations reported in [58] and evolving to computations with direct coupling [59]. New versions of the DSD/SST formulation introduced in [49] are the core technologies of the Stabilized ST FSI (SSTFSI) technique, which was also introduced in [49]. The ST-VMS method and SSTFSI technique, combined with a number of special techniques (see [60–63] and references therein) have been used in some of the most challenging parachute FSI computations (see [60, 64–66] and references therein), and also in a good number of patient-specific cardiovascular FSI and fluid mechanics computations (see [61–63, 67] and references therein). Computations with the SSTFSI technique also received a substantial attention in research related to iterative solution of large linear systems [68, 69].

In application of the DSD/SST formulation to flows with moving objects, the Shear–Slip Mesh Update Method (SSMUM) [70–72] has been very instrumental. The SSMUM was first introduced for computation of flow around two high-speed trains passing each other in a tunnel (see [70]). The challenge was to accurately and efficiently update the meshes used in computations based on the DSD/SST formulation and involving two objects in fast, linear relative motion. The idea behind the SSMUM was to restrict the mesh moving 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 [71] and flow around a helicopter [72].

The ST-VMS method was successfully tested on computation of wind-turbine rotor aerodynamics in [6, 73, 74]. Those computations did not include a wind-turbine tower, and therefore a mesh update method was not required. In [75], the ST-VMS method was applied to computation of wind-turbine rotor and tower aerodynamics. The presence of a tower requires a mesh update method that can handle the fast, rotational relative motion between the rotor and tower. The SSMUM would have been an option, but we decided to use a method that is more general. We use NURBS basis functions for the temporal representation of the rotor motion, mesh motion and also in remeshing. This is essentially the same computational technology used in the ST-VMS computations of flapping-wing aerodynamics reported in [76–79]. We named it “ST/NURBS Mesh Update Method (STNMUM)” in [75]. The motion of the rotor surface mesh created from the NURBS geometry is represented by quadratic temporal NURBS basis functions, with sufficient number of temporal patches for one rotation. This enables us to represent the circular paths associated with the rotor motion exactly and, with a “secondary mapping” [43, 44, 50, 76], specify a constant angular velocity corresponding to the invariant speeds along those paths. Given the motion of the surface mesh, we compute meshes that serve as temporal-control points. This is done by creating with an automatic mesh generator a new mesh at the central control point of the temporal patch, and computing the meshes at the other two control points by using the mesh moving technique [49, 80–83] developed earlier in conjunction with the DSD/SST method. The STNMUM allows us to do mesh computations with longer time in between, but get the mesh-related information for each ST slab, such as the coordinates and their time derivatives, from the temporal representation whenever we need. This approach where the mesh-related information is computed “directly” was called in [75] “Direct Temporal Representation (DTR).” In an alternative approach, we can obtain the mesh-related data after first computing the finite element meshes associated with each ST slab by interpolation from the temporal NURBS representation of the mesh. This approach was called “Interpolated-Mesh Temporal Representation (IMTR).” in [75]. For better mesh resolution, we use layers of thin elements near the blade surfaces. These layers of elements are created with a special mesh generation process and are not part of what we create with the automatic mesh generation process. They undergo rigid-body motion with the rotor. Despite the fast, rotational relative motion between the rotor and tower, the computations reported in [75] were carried out by using an automatic mesh generator only a total of 6 times during an entire computation.

We refer the interested reader to [50, 74] for the following methods that are not reviewed in this article: ALE-VMS and ST-VMS methods, formulation for weakly-enforced essential boundary conditions, structural mechanics formulation, which is based on the Kirchhoff–Love thin-shell theory and the bending-strip method

(see [10, 12, 31]), FSI coupling, mesh update, and a method for pre-bending of wind-turbine blades, which was recently proposed in [11].

In Section 2, we present the sliding-interface formulation from [36, 84, 85], which enables the simulation of rotor–tower interaction. The formulation was used in [85] for ALE-VMS aerodynamic simulations of the National Renewable Energy Lab (NREL) Phase VI wind turbine (see [86]) for comparison to the extensive set of experimental data available for this test case. We also present those simulations in Section 2. In Section 3, we describe, from [75], the ST-VMS computations of the wind-turbine rotor and tower aerodynamics. NURBS basis functions are used in temporal representation of the rotor and volume mesh motion and in remeshing. Simulations of the Micon 65/13M wind turbine with FSI, reported earlier in [87], are described in Section 4. We end with concluding remarks in Section 5.

2 Sliding-Interface Formulation and Rotor–Tower Interaction

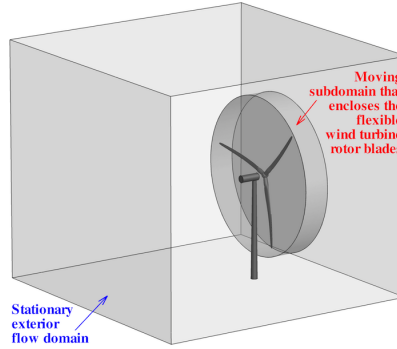


Fig. 1 Setup for the simulation of a full machine. The interior moving subdomain, which encloses the wind turbine rotor, and the exterior stationary subdomain, which houses the nacelle and tower.

2.1 Sliding-Interface Formulation

In order to simulate the full wind turbine configuration and investigate the rotor–tower interaction, we consider an approach that makes use of a moving subdomain, which encloses the entire wind turbine rotor, and a stationary subdomain that contains the rest of the wind turbine (see Figure 1). The two domains are in relative motion and share a sliding cylindrical interface. The meshes on each side of the interface are nonmatching because of the relative motion (see Figure 2). As a result, a numerical procedure is needed to impose the continuity of the kinematics and

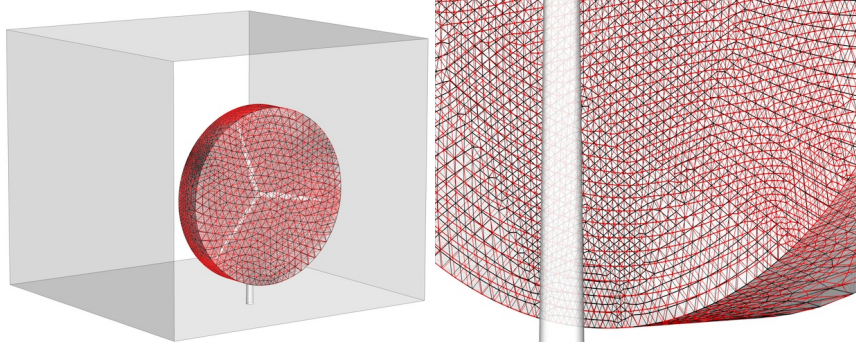


Fig. 2 Nonmatching meshes at the sliding interface between the stationary and moving subdomains. Left: Full domain. Right: Zoom on the sliding interface.

tractions at the stationary and rotating subdomain interface despite the fact that the interface discretizations are incompatible. Such a procedure was developed in [36] in the context of IGA for computing flows about rotating components. The advantage of IGA for rotating-component flows is that the cylindrical sliding interfaces are represented exactly and no geometry errors are incurred. In the case of standard FEM employed here, the geometric compatibility is only approximate. The sliding-interface coupling was successfully tested on the NREL Phase VI wind turbine in [85] and is presented in what follows.

Let the subscripts S and M denote the quantities pertaining to the fluid mechanics problem on the stationary and moving subdomains, respectively. The subdomain that encloses the rotor rotates with it, and the interior of the rotating subdomain is allowed to deflect to accommodate the motion of the blades. However, the motion of the outer boundary of the rotor subdomain is restricted to a rigid rotation to maintain geometric compatibility with the stationary subdomain. To enforce the compatibility of the flow kinematics and tractions at the sliding interface, we add the following terms to the ALE-VMS formulation, which is now assumed to hold in both the stationary and moving subdomains:

$$\begin{aligned}
 & - \sum_{b=1}^{n_{\text{eb}}} \int_{\Gamma_t^b \cap (\Gamma_t)_{\text{SI}}} (\mathbf{w}_S^h - \mathbf{w}_M^h) \cdot \frac{1}{2} (\boldsymbol{\sigma}_S \mathbf{n}_S - \boldsymbol{\sigma}_M \mathbf{n}_M) d\Gamma \\
 & - \sum_{b=1}^{n_{\text{eb}}} \int_{\Gamma_t^b \cap (\Gamma_t)_{\text{SI}}} \frac{1}{2} (\delta \boldsymbol{\sigma}_S \mathbf{n}_S - \delta \boldsymbol{\sigma}_M \mathbf{n}_M) \cdot (\mathbf{u}_S^h - \mathbf{u}_M^h) d\Gamma \\
 & - \sum_{b=1}^{n_{\text{eb}}} \int_{\Gamma_t^b \cap (\Gamma_t)_{\text{SI}}} \mathbf{w}_S^h \cdot \rho \{ (\mathbf{u}_S^h - \hat{\mathbf{u}}_S^h) \cdot \mathbf{n}_S \}_- (\mathbf{u}_S^h - \mathbf{u}_M^h) d\Gamma
 \end{aligned}$$

$$\begin{aligned}
& - \sum_{b=1}^{n_{\text{eb}}} \int_{\Gamma_t^b \cap (\Gamma_t)_{\text{SI}}} \mathbf{w}_M^h \cdot \rho \{ (\mathbf{u}_M^h - \hat{\mathbf{u}}_M^h) \cdot \mathbf{n}_M \}_- (\mathbf{u}_M^h - \mathbf{u}_S^h) d\Gamma \\
& + \sum_{b=1}^{n_{\text{eb}}} \int_{\Gamma_t^b \cap (\Gamma_t)_{\text{SI}}} \frac{C_I^B \mu}{h_n} (\mathbf{w}_S^h - \mathbf{w}_M^h) \cdot (\mathbf{u}_S^h - \mathbf{u}_M^h) d\Gamma = 0,
\end{aligned} \tag{1}$$

where $\delta\sigma$ is given by $\delta\sigma(\mathbf{w}, q) \mathbf{n} = 2\mu\boldsymbol{\varepsilon}(\mathbf{w})\mathbf{n} + q\mathbf{n}$. $(\Gamma_t)_{\text{SI}}$ is the sliding interface, and $\{\mathcal{A}\}_-$ denotes the negative part of \mathcal{A} , that is, $\{\mathcal{A}\}_- = \mathcal{A}$ if $\mathcal{A} < 0$ and $\{\mathcal{A}\}_- = 0$ if $\mathcal{A} \geq 0$. The sliding-interface formulation may be seen as a DG method, where the continuity of the basis function is enforced everywhere in the interior of the two subdomains, but not at the sliding interface between them. The structure of the terms on the sliding interface is similar to that of the weak enforcement of essential boundary conditions. The significance of each term is explained in detail in [36]. In the current application, $\hat{\mathbf{u}}_S^h = \mathbf{0}$, because the subdomain S is stationary. However, the formulation is able to handle situations where both subdomains are in motion.

Remark 1 *Nonmatching interface discretizations in the FSI and sliding-interface problems necessitate the use of interpolation or projection of kinematic and traction data between the nonmatching surface meshes (see, e.g., [43, 44, 88], where [44] is more comprehensive than [43]). A computational procedure, which can simultaneously handle the data transfer for IGA and FEM discretizations, was proposed in [88]. The procedure also includes a robust approach in identifying “closest points” for arbitrary shaped surfaces. While such interface projections are rather straightforward for weakly-coupled FSI algorithms, they require special techniques [44, 49, 60] for strongly-coupled, direct and quasi-direct methods [44, 49, 59, 60, 89], which become monolithic for matching discretizations.*

2.2 NREL Phase VI Wind Turbine

The computational results in this section make use of the ALE-VMS technique and are taken from [85]. The sliding-interface formulation is applied to the simulation of the full NREL Phase VI wind turbine configuration, including the rotor (blades and hub), nacelle and tower. The tower is composed of two cylinders with diameters of 0.6096 m and 0.4064 m that are connected with a short conical section. The tower height is 11.144 m above the wind tunnel floor. The detailed geometry of the tower and nacelle can be found in Hand *et al.* [86]. For this study, wind speeds of 7 and 10 m/s were selected from the experimental sequence S. The experimental sequence S setup consists the wind turbine rotor in the upwind configuration, 0° yaw angle, 0° cone angle, rotational speed of 72 rpm, and blade tip pitch angle of 3° . The air density and viscosity are 1.23 kg/m^3 and $1.78 \times 10^{-5} \text{ kg/(m}\cdot\text{s)}$, respectively.

Figure 3 shows the mesh resolution used in the computation. The mesh is highly refined near the rotor, nacelle and tower, as well as downstream of the wind turbine

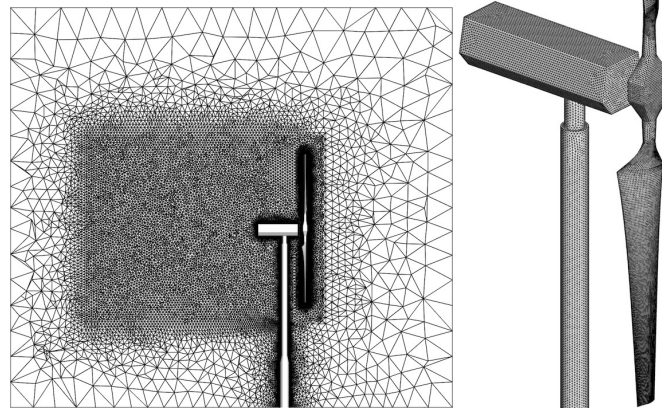


Fig. 3 Meshes used in the full-wind-turbine simulation. Left: 2D cut at $x = 0$ to show the flow domain mesh quality. Right: Rotor, nacelle, and tower surface mesh.

to better capture the wake turbulence. The mesh is comprised of 6,835,647 linear elements and 1,603,377 nodes. The size of the first boundary-layer element in the wall-normal direction is 0.002 m, and 15 layers of prismatic elements were generated with a growth ratio of 1.2. The time step size is set to 1.0×10^{-5} s.

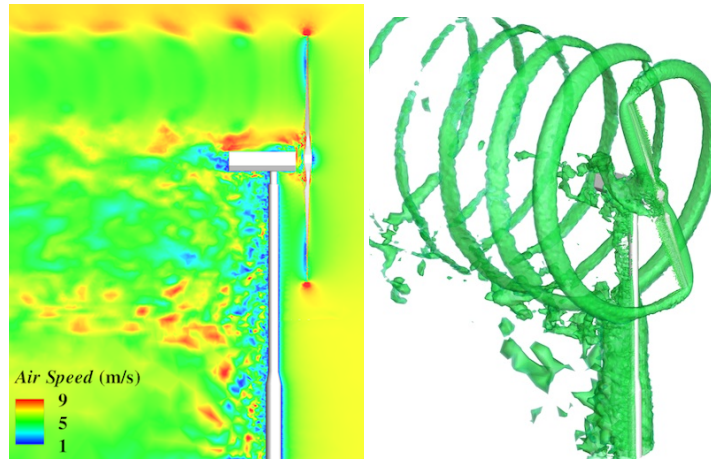


Fig. 4 Air speed planar distribution and isosurfaces at an instant for the 7 m/s case.

Figures 4 and 5 show the flow visualization of the full-wind-turbine simulations of the 7 and 10 m/s cases, respectively. The flow structures are different between the two cases. The tip vortex for the 7 m/s case decays very slowly as it is convected downstream, while the tip vortex breaks down quickly for the 10 m/s case. No visible

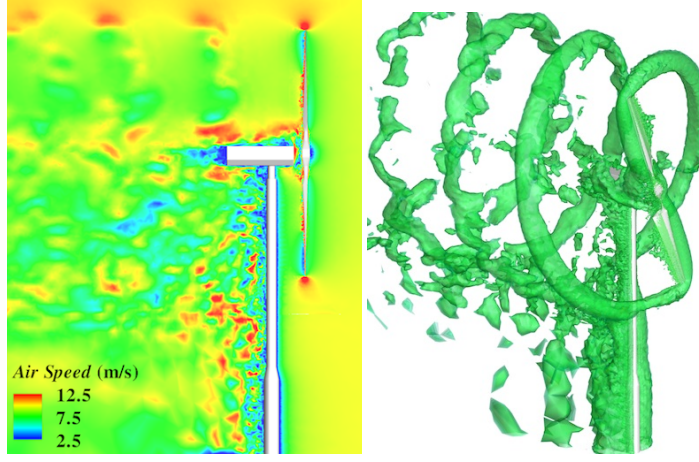


Fig. 5 Air speed planar distribution and isosurfaces at an instant for the 10 m/s case.

discontinuities are present in the flow field at the sliding interface, which indicates that the method correctly handles the kinematic compatibility conditions.

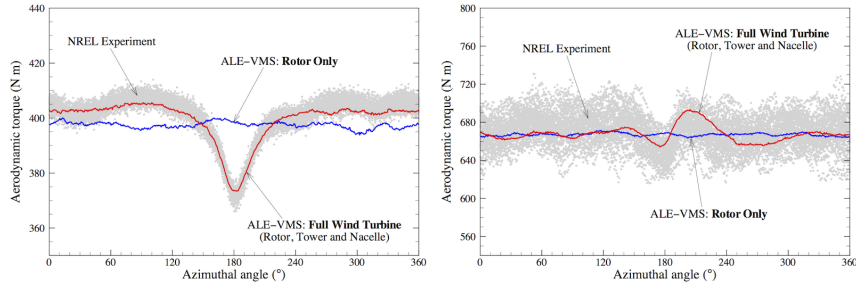


Fig. 6 Single-blade aerodynamic torque over a full revolution for 7 m/s (left) and 10 m/s (right) cases. The 180° azimuthal angle corresponds to the instant when the blade passes in front of the tower. The tower effect is clearly pronounced in the 7 m/s case. It is also present in the 10 m/s case, but is not as significant. The results in both cases are in very good agreement with the experimental data.

To see the influence of the tower, the single-blade aerodynamic torque over a full revolution is plotted in Figure 6 for both 7 and 10 m/s cases. The results of the full-wind-turbine computations are compared with the experimental data, as well as with the results of the rotor-only computations. For the full-wind-turbine simulation of the 7 m/s case, Figure 6(a) clearly shows the drop in the aerodynamic torque at an instant when the blade passes in front of the tower, which corresponds to the azimuthal angle of 180°. The drop in the torque is about 8% relative to its value when the blade is away from the tower. These results are in good agreement with the experimental data. The rotor-only computation, which is also shown in

the figure, is obviously unable to predict this feature, which may be important for the transient structural response of the blades. It should be noted, however, that the cycle-averaged aerodynamic torque is nearly identical for the full-wind-turbine and the rotor-only simulations. The picture is completely different for the 10 m/s case, where the influence of the tower, although clearly present, is a lot less pronounced.

3 ST-VMS Computation of the Wind-Turbine Rotor and Tower Aerodynamics

This section is from [75].

3.1 Rotation Representation with Constant Angular Velocity

We use quadratic NURBS functions, as described in [43, 44, 50, 76], to represent a circular arc. We discretize time and position as follows:

$$t = \sum_{\alpha=1}^{n_{\text{ent}}} T^{\alpha}(\Theta_t(\theta)) t^{\alpha}, \quad \mathbf{x} = \sum_{\alpha=1}^{n_{\text{ent}}} T^{\alpha}(\Theta_x(\theta)) \mathbf{x}^{\alpha}. \quad (2)$$

Here n_{ent} is the number of temporal element nodes, T^{α} is the basis function, $\Theta_t(\theta)$ and $\Theta_x(\theta)$ are the secondary mappings for time and position, and t^{α} and \mathbf{x}^{α} are the time and position values corresponding to the basis function T^{α} . The basis functions could be finite element or NURBS basis functions. For the circular arc, $n_{\text{ent}} = 3$ and they are quadratic NURBS. The secondary mapping concept above was introduced in [43], and the velocity can be expressed as follows:

$$\frac{d\mathbf{x}}{dt} = \left(\sum_{\alpha=1}^{n_{\text{ent}}} \frac{dT^{\alpha}}{d\Theta_x} \frac{d\Theta_x}{d\theta} \mathbf{x}^{\alpha} \right) \left(\sum_{\alpha=1}^{n_{\text{ent}}} \frac{dT^{\alpha}}{d\Theta_t} \frac{d\Theta_t}{d\theta} t^{\alpha} \right)^{-1}, \quad (3)$$

leading to

$$\frac{d\mathbf{x}}{dt} = \left(\sum_{\alpha=1}^{n_{\text{ent}}} \frac{dT^{\alpha}}{d\Theta_x} \mathbf{x}^{\alpha} \right) \left(\sum_{\alpha=1}^{n_{\text{ent}}} \frac{dT^{\alpha}}{d\Theta_t} t^{\alpha} \right)^{-1} \left(\frac{d\Theta_x}{d\theta} \frac{d\theta}{d\Theta_t} \right). \quad (4)$$

Thus, the speed along the path can be specified only by modifying the secondary mapping. For a circular arc, two methods were introduced in [44, 76]; one is modifying the secondary mapping for position and the other one is modifying both such that $\frac{dt}{d\theta}$ is constant. We note that, in theory, the secondary mapping selections do not make any difference as long as the relationship $\frac{d\Theta_x}{d\Theta_t}$ is the same. In our implementation, to keep the process general, we search for the parametric coordinate θ by using

an iterative solution method [44, 50, 76]. We use the latter set of the secondary mappings, having constant $\frac{dt}{d\theta}$. For the IMTR, we find the parametric coordinate corresponding to each time level and interpolate the position to obtain the corresponding mesh. For the DTR, we first calculate time corresponding to each integration point, including the time step size because of the jump term, and then calculate θ_x and θ_t to interpolate the position and velocity from Eqs. (2) and (4).

3.2 Geometry Construction

The geometry construction for the wind-turbine rotor blade and hub was described in [5, 6], and also partially in [73, 75]. For completeness we repeat some of that information here. The geometry of the rotor blade is based on the NREL 5MW off-shore baseline wind turbine reported in [90]. 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. 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 profile is used. 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. Images of the airfoil types used in the wind-turbine rotor blade and the final blade including the twisting cross-sections can be found in [5, 6, 73]. Starting from this rotor surface geometry, we generate a quadratic NURBS surface with G^2 and G^1 continuity between the patches around and along the blade, respectively. The tower geometry was created based on the tower design specified for the NREL 5MW wind turbine, which describes a circular tower with a height of 87.6 m, a base diameter of 6 m, and a top diameter of 3.87 m. This geometry was generated by lofting between NURBS curves for the top and base of the tower. The rotor axis is 90° to the tower, and there is no tilt or precone. The distance between the tower axis and the point where the three blade axes intersect is 5 m. For most of the blade, the clearance from the tower is in the range 2.3 m to 2.8 m.

3.3 Problem Setup

We compute the aerodynamics of the rotor with and without its tower for a given rotor shape and wind speed and a specified rotor speed. 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 [91] for wind-turbine terminology). We use air properties at standard sea-level conditions.

The Reynolds number (based on the cord length at $\frac{3}{4}R$ and the relative velocity there) is approximately 12 million. 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 top, side, and bottom boundaries slip conditions are imposed.

3.4 Rotor Motion

The circular turbine rotation is represented with temporal NURBS basis functions and secondary mapping, described in Section 3.1. Because the 3 blades of the turbine are 120° apart, rotational geometric periodicity is used such that a full 360° rotation is defined by 3 identical 120° segments. Each 120° segment is divided into 6 patches to keep the mesh distortion under control. Each patch has 3 temporal-control points. The 6 temporal patches and their control points are shown in Figure 7.

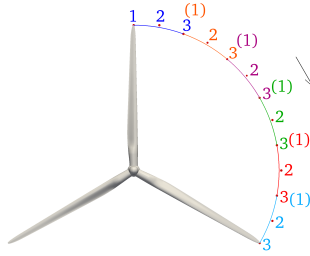


Fig. 7 Path of a blade tip with temporal patches and control point numbering local to each patch. A control point at the start of a patch and colocated with a control point at the end of the previous patch is in parentheses. Patch colors: 1: Blue, 2: Orange, 3: Purple, 4: Green, 5: Red, 6: Teal.

3.5 Surface Mesh

The rotor surface mesh is generated by discretizing the NURBS surface geometry at each knot intersection, subdividing the knot spans into quadrilateral finite elements in a structured way, and subdividing the quadrilateral elements into two triangles. Small adjustments are made to improve the mesh near the hub. The surface mesh position is calculated at each temporal-control point shown in Figure 7. Figure 8 shows the rotor surface at the three temporal-control points of the first patch. We note that control points 1 and 3 lie on the path traveled by the points on the blades and a portion of the hub at the start and end of the 20° rotation, but control point 2 lies outside the circular arc. This means that the temporal-control mesh 2 is deformed compared to the temporal-control meshes 1 and 3. A temporal-control mesh 2 has to be generated for the part of the surface between the hub cross-sections rotat-

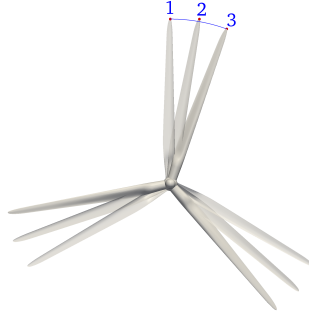


Fig. 8 Rotor surface at the three temporal-control points of the first patch.

ing with the blades and fixed to the tower. The tower surface mesh is generated from the NURBS representation of the surface by using an unstructured triangular mesh generator and matched with the previously generated hub mesh at the intersection. The rotor surface mesh has 34,087 nodes and 68,112 triangles. The tower surface mesh has 6,952 nodes and 13,806 triangles.

3.6 Volume Mesh

3.6.1 Boundary-Layer Mesh

The layers of thin elements near the blades are generated by extruding the NURBS surface geometry into NURBS volume representation, subdividing the knot spans into hexahedral finite elements in a structured way, and subdividing the hexahedral elements into six tetrahedral elements. The resulting boundary-layer mesh for each blade consists of 4 layers with a first-layer thickness of about 2.85×10^{-2} m and a total thickness of about 2.85×10^{-1} m, 52 nodes in the circumferential direction around the blade, and approximately 145 nodes in the longitudinal direction. The tower boundary-layer mesh is generated by extruding the tower surface mesh to layers of prismatic elements, which are then subdivided into 3 tetrahedral elements each. It consists of 4 layers, with a first-layer thickness of 2.85×10^{-2} m and a total thickness of 3.0×10^{-1} m. The blade and tower boundary-layer meshes do not undergo any mesh deformation. This maintains the mesh quality in the boundary-layer regions. Figure 9 shows the tower and blade boundary-layer meshes.

3.6.2 Overall Mesh

Three different meshes are used in the computations: Mesh 1, Mesh 2, and Mesh 3. Mesh 2 has both the rotor and the tower, with boundary-layer mesh only for the blades. Mesh 1 has only the rotor, and is identical to Mesh 2 except the tower is filled

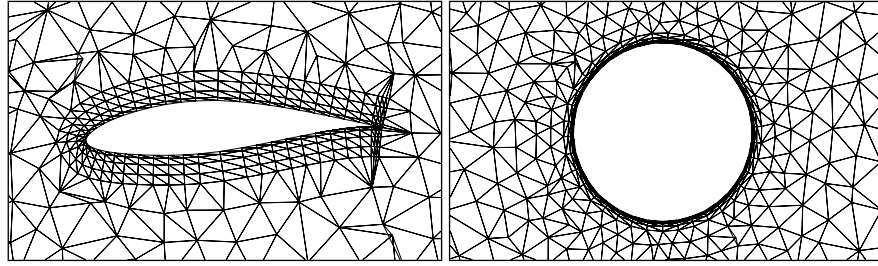


Fig. 9 Left: Boundary-layer mesh at $\frac{3}{4}R$. Right: Tower boundary-layer mesh.

with volume elements. Mesh 3 has both the rotor and the tower, with boundary-layer mesh for both the blades and the tower, and a mesh refinement region downstream of the tower. All three meshes have an outer, coarser region, with an inner cylindrical refinement region surrounding the rotor. This inner refinement region includes most of the tower for Mesh 2 and Mesh 3, and the mesh refinement region downstream of the tower for Mesh 3. Figure 10 illustrates, as an example, a cut plane of Mesh 3. The inflow and outflow boundaries are at $3.79R$ and $10.35R$ from the hub center,

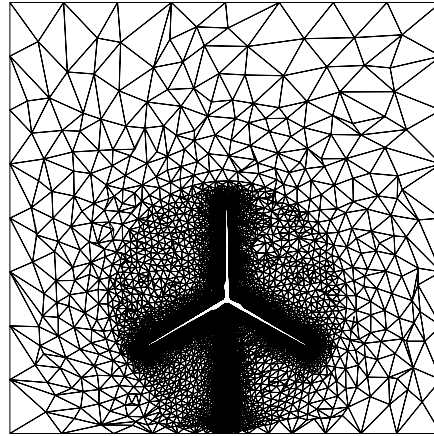


Fig. 10 A cut plane of temporal-control mesh 1 of patch 1 for Mesh 3.

respectively. The side, top, and bottom boundaries are at $2.29R$, $3.17R$, and $1.43R$, respectively (see Figure 10). The volume mesh is generated once per patch using an automatic mesh generator (a total of 6 times). The mesh is generated at control point 2 of each patch to minimize mesh distortion between control points. We note that only the mesh in the inner cylindrical refinement region surrounding the rotor is generated for each patch. The outer, coarser mesh is generated only once, and is kept the same when the inner meshes are generated for each patch. The mesh moving technique [49, 80–83] developed earlier in conjunction with the DSD/SST

method is used for computing the mesh position for control points 1 and 3. The outer surfaces of the boundary-layer meshes serve as the boundaries where we specify the inner boundary conditions for the mesh motion. The external boundaries of the computational domain serve as the boundaries where we specify the outer boundary conditions, with zero displacement. In the elasticity equations of the mesh moving technique, a Young's modulus of 1.0, a Poisson's ratio of -0.20, and a stiffening exponent of 1.5 are used. We use 1,500 GMRES [92] iterations for each step of the mesh motion, with diagonal preconditioner. Each 10° range of motion is computed over 40 steps. The approximate number of nodes for Mesh 1, Mesh 2 and Mesh 3 are 465,000, 440,000 and 595,000.

3.7 Computational Conditions

In the ST-VMS computations, the stabilization parameters are given by Eq. (7) in [49] for $\tau_M (= \tau_{SUPS}) = \tau_{SUPG}$ and Eq. (19) in [75] for $\nu_C (= \nu_{LSIC}) = \nu_{LSIC-HRGN}$. They are both used with $h_{RGN} = h_{RGNT}$, given by Eq. (15) in [75], which was originally introduced in [76]. The DTR and IMTR approaches are used on all three meshes. Least-squares projection is used to interpolate the velocity and pressure between temporal patches. Because the boundary-layer meshes and the tower and rotor surface meshes remain identical between temporal patches, the velocity values are transferred exactly for those nodes. The time-step size is 2.23×10^{-3} s (145 time steps per patch), with 4 nonlinear iterations per time-step. First we develop the flow field for 500 time steps while the rotor is static, ramping up the inflow velocity during the first 300 steps from zero to the wind speed using a cosine ramp. During this flow-development stage, we use 150, 150, 200, and 400 GMRES iterations for the 4 nonlinear iterations. In computations with the rotor in motion, we use 150, 150, 200, and 400 GMRES iterations for Mesh 1, and 150, 250, 350, and 500 GMRES iterations for Mesh 2 and Mesh 3. With the GMRES iterations in flow computations, we use nodal-block-diagonal preconditioner. The mesh is partitioned based on the METIS algorithm [93] to improve parallel efficiency in the computations.

3.8 Results

Figure 11 shows the torque for Mesh 1 with the DTR approach, for the last 360° rotation of a blade, with the rotation amount measured from the orientation seen in Figure 7. For reference purposes, Figure 11 includes the NREL data. The torque is within 8% of the NREL data. Figure 12 shows the torque for the last 80° rotation of a single blade of Mesh 1 with the DTR approach, compared with the torque from an earlier, single-blade computation [73] using the TGI option of $\nu_C (= \nu_{LSIC})$. The single-blade computation has the same blade geometry, wind speed, and rotor speed, but has a single-blade mesh in a rotationally-periodic domain. It has a more refined

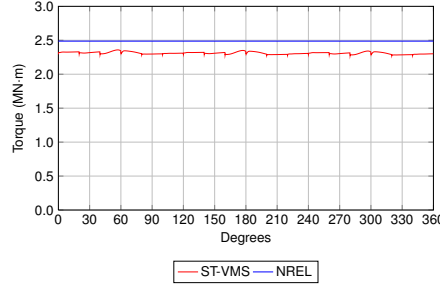


Fig. 11 Torque for Mesh 1 with the DTR approach, compared with the NREL data.

boundary-layer mesh and a time-step size that is approximately 5 times smaller. The higher torque seen for the single-blade computation may be due to the fact that the computation was carried out for a much shorter duration, only 80° of rotation versus $1,080^\circ$ for the Mesh 1 computation. Therefore the current computation likely represents a more settled torque value. The higher torque for the single-blade computation may also be due the fact that the computation was carried out using a computational domain with significantly nearer lateral boundaries. Figures 13 and 14 show the

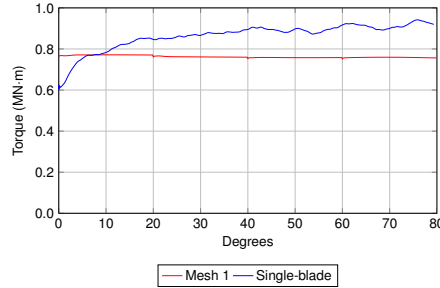


Fig. 12 Torque for a single blade of Mesh 1 with the DTR approach, compared with the torque from an earlier single-blade computation [73] using the TGI option of $\nu_C (= \nu_{LSIC})$.

torque for all three meshes with the DTR and IMTR approaches. As can be seen from these figures, Mesh 1 (no tower) has a very stable torque, while Mesh 2 and Mesh 3 (with tower) exhibit a significant but expected drop in torque each time a blade passes the tower. Figure 15 shows, for each of the three meshes, the torque obtained with the DTR and IMTR approaches. The figure illustrates that the DTR and IMTR approaches result in a nearly identical torque magnitude for all 3 meshes. Figure 16 shows the torque for Mesh 1 with the DTR approach, using two different time-step sizes: 2.23×10^{-3} s (145 time steps per patch) and 4.49×10^{-3} s (72 time steps per patch). Doubling the time-step size still yields a comparable torque value, within 10% of the value for the smaller time-step size. We also carried out a computation with the convective form of the ST-VMS formulation (see Eq. (8.17) in [44]),

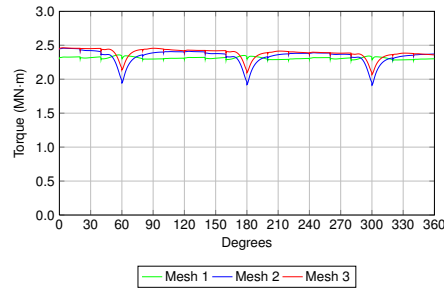


Fig. 13 Torque for Mesh 1, Mesh 2 and Mesh 3 with the DTR approach.

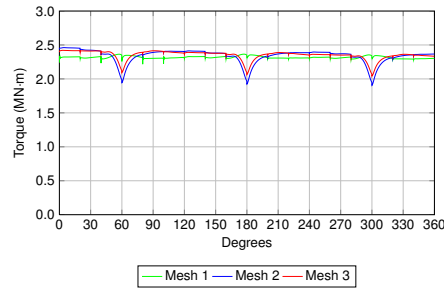


Fig. 14 Torque for Mesh 1, Mesh 2 and Mesh 3 with the IMTR approach.

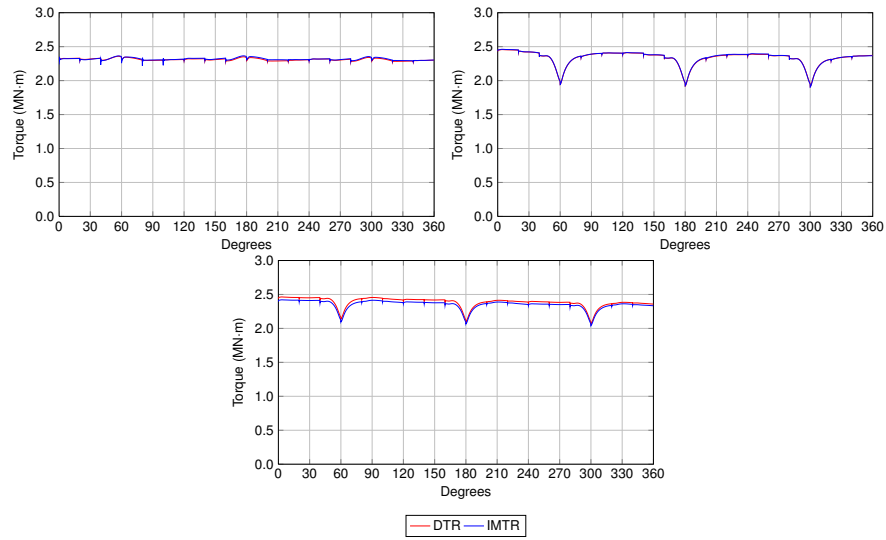


Fig. 15 Torque with the DTR and IMTR approaches for Mesh 1, Mesh 2, and Mesh 3.

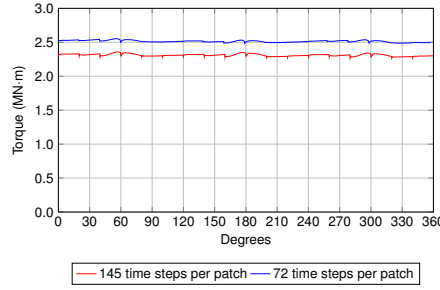


Fig. 16 Torque for Mesh 1 with the DTR approach, using two different time-step sizes: 2.23×10^{-3} s (145 time steps per patch) and 4.49×10^{-3} s (72 time steps per patch).

but with a smaller time-step size: 4.46×10^{-4} s (725 time steps per patch). Figure 17 shows the torque for Mesh 2 with the DTR approach and the conservative and convective forms of the ST-VMS formulation. The conservative-form computation is with the standard time-step size: 2.23×10^{-3} s (145 time steps per patch). Figure 18

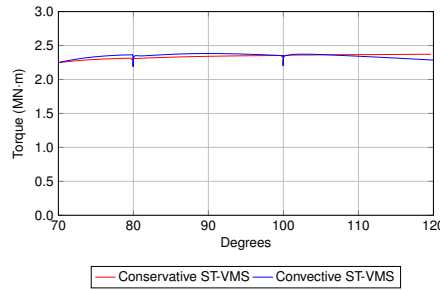


Fig. 17 Torque for Mesh 2 with the DTR approach and the conservative and convective forms of the ST-VMS formulation. The time-step sizes: 4.46×10^{-4} s (725 time steps per patch) for the convective form and 2.23×10^{-3} s (145 time steps per patch) for the conservative form. The torques are from the same period in a rotation cycle, but the conservative-form torque is from the last 360° of the computation, and the convective-form torque is from a recently-started, ongoing computation.

shows the torque for the individual blades of Mesh 2 with the DTR approach. The figure clearly shows the expected torque drop for each blade as it passes the tower, while the other two blades maintain relatively constant torque. Figure 19 shows the torque for 10 equal-length spanwise sections of a blade of Mesh 2 with the DTR approach. Greatest amount of torque is generated in sections 6–9 of the blade, while section 10 at the tip and the other lower sections generate less torque. Figure 20 shows a volume rendering of the vorticity for Mesh 2 with the DTR approach. The flow patterns vary considerably along each blade length, illustrating the necessity to carry out the computations in 3D. Figure 21 shows the pressure coefficient at $0.90R$ for the last 0° orientation of a blade of Mesh 2, with the DTR and IMTR approaches, with the last 0° orientation being common between the two computations. There is

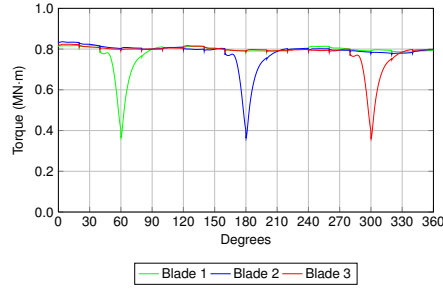


Fig. 18 Torque for the individual blades of Mesh 2 with the DTR approach.

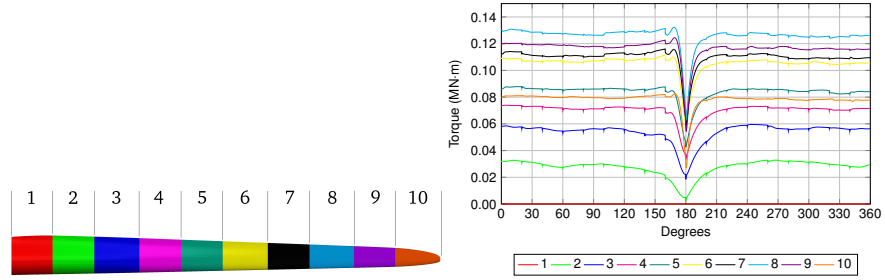


Fig. 19 Torque for 10 equal-length spanwise sections of a blade of Mesh 2 with the DTR approach.

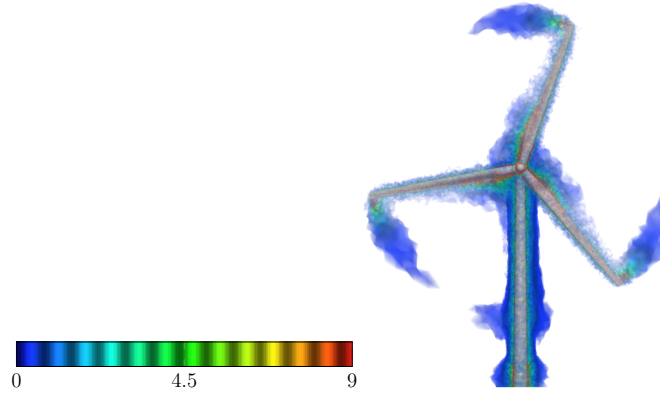


Fig. 20 Volume rendering of the vorticity (in s^{-1}) from the last 360° of the computation for Mesh 2 with the DTR approach.

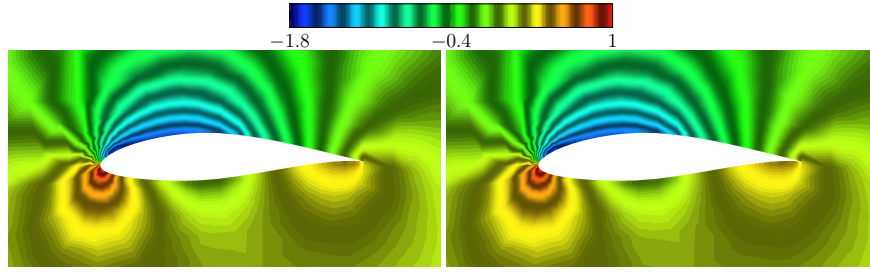


Fig. 21 Pressure coefficient at $0.90R$ for the last 0° orientation of a blade of Mesh 2, with the DTR (left) and IMTR (right) approaches.

very little difference in the pressure coefficient around the blades between the DTR and IMTR approaches. Figure 22 shows the pressure coefficient at $0.90R$ for the last 180° orientation of a blade of Mesh 1, Mesh 2 and Mesh 3, with the DTR approach, with the last 180° orientation being common between Mesh 2 and Mesh 3 computations. Averaged torque (in MN·m) for the last 360° rotation for Mesh 1, 2

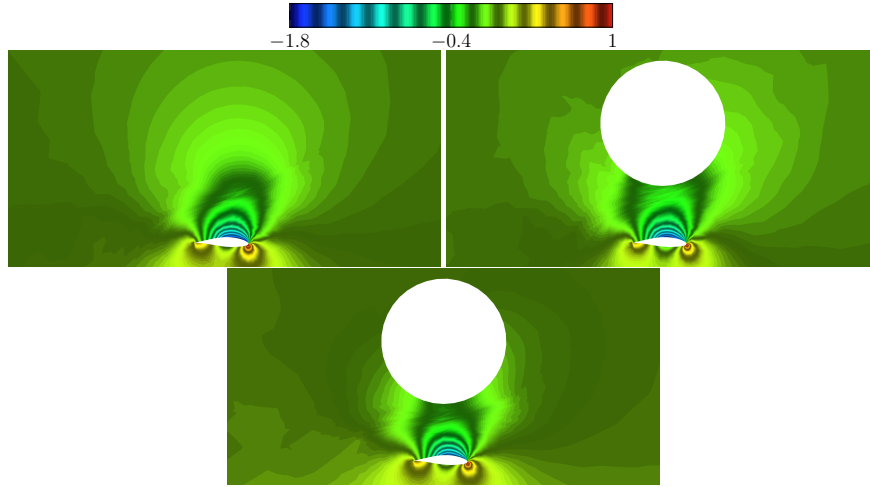


Fig. 22 Pressure coefficient at $0.90R$ for the last 180° orientation of a blade of Mesh 1, Mesh 2, and Mesh 3, with the DTR approach.

and 3 are 2.31, 2.34 and 2.39 for the DTR approach, and 2.32, 2.34 and 2.35 for the IMTR approach. The values show that the difference in torque between the DTR and IMTR approaches, and between Mesh 2 and Mesh 3, is rather small. The difference in torque between Mesh 1 and Mesh 2 and 3 illustrates effect of the tower.

4 Micon 65/13M Wind Turbine with a CX-100 Blade

This section is adapted from [87]. We simulate the Micon 65/13M wind turbine at field test conditions [94]. Micon 65/13M is a three-blade, horizontal-axis, fixed-pitch, upwind turbine with the total rotor diameter of 19.3 m and rated power of 100 kW. The hub is located at the height of 23 m. The wind turbine stands on a tubular steel tower, with a base diameter of 1.9 m. The drive train generator operates at 1200 rpm, while the rotor spins at a nominal speed of 55 rpm. The Micon 65/13M wind turbine was used for the Long-Term Inflow and Structural Testing (LIST) program [95] initiated by Sandia National Laboratories in 2001 to explore the use of carbon fiber in wind turbine blades. Three experimental blade prototypes, GX-100, CX-100 and TX-100, were developed specifically for this project. We use the CX-100 conventional carbon-spar blade design [94, 96]. The NREL S821, S819 and S820 airfoils are used to define the blade geometry. The details of the blade geometry definition are provided in Table 1.

RNodes	Chord	AeroTwst	Airfoil
0.200	0.356	29.6	Cylinder
0.600	0.338	24.8	Cylinder
1.000	0.569	20.8	Cylinder
1.400	0.860	17.5	NREL S821
1.800	1.033	14.7	NREL S821
2.200	0.969	12.4	NREL S821
3.200	0.833	8.3	NREL S821
4.200	0.705	5.8	NREL S819
5.200	0.582	4.0	NREL S819
6.200	0.463	2.7	NREL S819
7.200	0.346	1.4	NREL S819
8.200	0.232	0.4	NREL S819
9.000	0.120	0.0	NREL S820

Table 1 CX-100 blade with “RNodes” (m), “Chord” (m), “AeroTwst” ($^{\circ}$), and “Airfoil” type.

4.1 Eigenfrequency Analysis of the CX-100 Blade

The blade structure is comprised of five primary sections: leading edge, trailing edge, root, spar cap, and shear web. The sections are shown in Figure 23. Each section is further subdivided into zones, each consisting of a multilayer composite layup. There is a total of 32 zones with constant total thickness and unique laminate stacking. The effective material properties for each of the zones are computed using the procedures described in [50, 74]. All 32 zones are identified on the blade surface

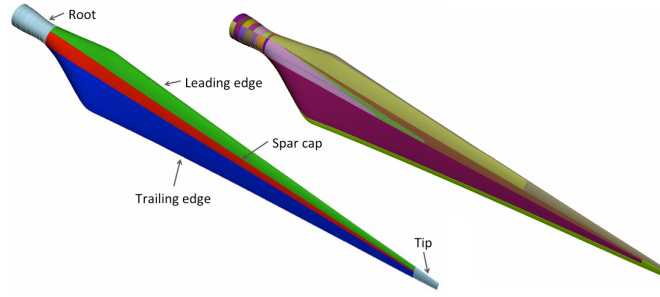


Fig. 23 Left: Five primary sections of the CX-100 blade; Right: 32 distinct material zones of the CX-100 blade.

and are shown in Figure 23. For more details of the material composition of the CX-100 blade see [87]. We perform eigenfrequency calculations of the CX-100

	Control points	Elements
Mesh 1	3,469	1,846
Mesh 2	7,411	4,647
Mesh 3	25,896	18,611

Table 2 NURBS blade meshes used in the eigenfrequency analysis.

	Mode 1	Mode 2	Mode 3
Mesh 1	8.28	15.92	19.26
Mesh 2	8.22	15.61	18.21
Mesh 3	8.22	15.6	18.01
Experiment	7.6–8.2	15.7–18.1	20.2–21.3

Table 3 Comparison of experimentally measured and computed natural frequencies (in Hz) for the free case. Mode 1 is the first flapwise mode, Mode 2 is the first edgewise mode, and Mode 3 is the second flapwise mode.

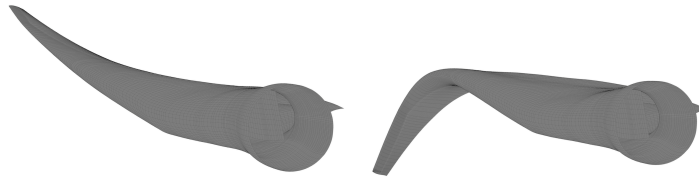


Fig. 24 First (left) and second (right) flapwise bending mode for the clamped case.

	Mode 1	Mode 2	Mode 3
Mesh 1	4.33	11.82	19.69
Mesh 2	4.29	11.61	19.08
Mesh 3	4.27	11.54	18.98
Experiment	4.35	11.51	20.54

Table 4 Comparison of experimentally measured and computed natural frequencies (in Hz) for the clamped case. Modes 1-3 are the first three flapwise bending modes.

blade using three quadratic NURBS meshes. The coarsest mesh has 1,846 elements, while the finest mesh has 18,611. The mesh statistics are summarized in Table 2. The eigenfrequency results are compared with the experimental data from [97, 98]. We compute the case with free boundary conditions and the case when the blade is clamped at the root. In both cases, the computed natural frequencies are in good agreement with the experimental data (see Tables 3 and 4). The medium mesh shows a good balance between the computational cost and accuracy. For this reason, this mesh is chosen for the FSI computations here. The mode shapes computed using the medium mesh for the clamped case are shown in Figure 24.

4.2 Aerodynamics and FSI Computations

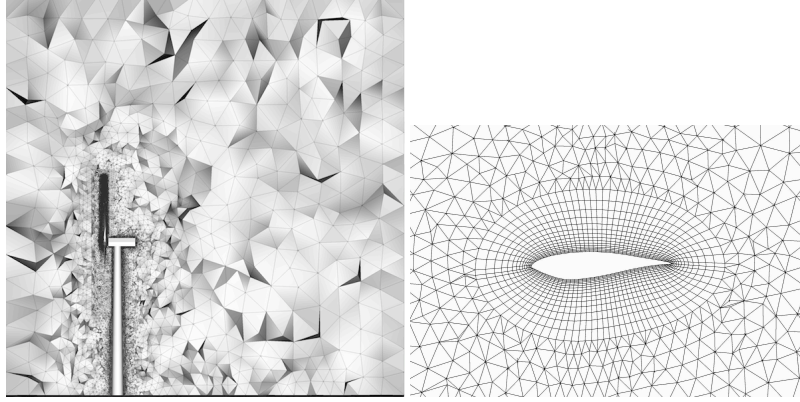


Fig. 25 Left: Computational domain and mesh with the refined inner region for better flow resolution near the rotor; Right: 2D blade cross-section at $r/R = 70\%$ and the boundary-layer mesh.

In this section, we present aerodynamic and FSI simulations. For both cases, a constant inflow wind speed of 10.5 m/s and fixed rotor speed of 55 rpm are prescribed. These correspond to the operating conditions reported for the field tests in [94]. The air density and viscosity are 1.23 kg/m^3 and $1.78 \times 10^{-5} \text{ kg/(m}\cdot\text{s)}$, re-

spectively. Zero traction boundary conditions are prescribed at the outflow and no-penetration boundary conditions are prescribed at the top, bottom, and side surfaces of the outer (stationary) computational domain. No-slip boundary conditions are prescribed at the rotor, nacelle, and tower, and are imposed weakly. Figure 25 shows the computational domain and mesh used in this study. The mesh consists of 5,134,916 linear elements, which are triangular prisms in the rotor boundary layers and tetrahedra everywhere else in the domain. The mesh is refined in the rotor and tower regions for better flow resolution near the wind turbine. The size of the first element in the wall-normal direction is 0.002 m, and 15 layers of prismatic elements were generated with a growth ratio of 1.2. Figure 25 shows a 2D blade cross-section at 70% spanwise station to illustrate the boundary-layer mesh used in the computations. The time-step size is set to 3.0×10^{-5} s. In Figure 26 the time history of

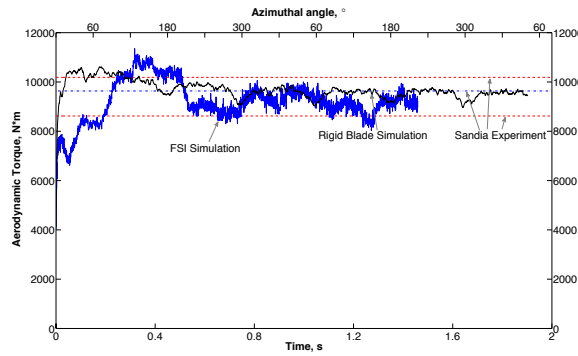


Fig. 26 Aerodynamic torque for the FSI and rigid-blade simulations. The experimental range for the torque and its average are provided for comparison and are plotted using dashed lines.

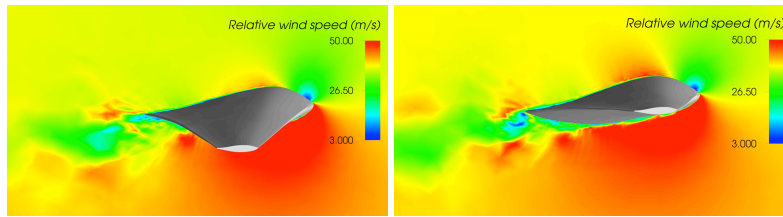


Fig. 27 Relative wind speed at the 70% spanwise station for the FSI simulation at $t = 0.86$ s (left) and $t = 1.06$ s (right). The blade deflection is clearly visible.

the aerodynamic torque is plotted. As can be seen from the plot, using FSI, we capture the high frequency oscillations caused by the bending and torsional motions of the blades. In the case of the rigid blade the only high-frequency oscillations in the torque curve are due to the trailing-edge turbulence. For the rigid blade case the effect of the tower on the aerodynamic torque is more pronounced, while in the case

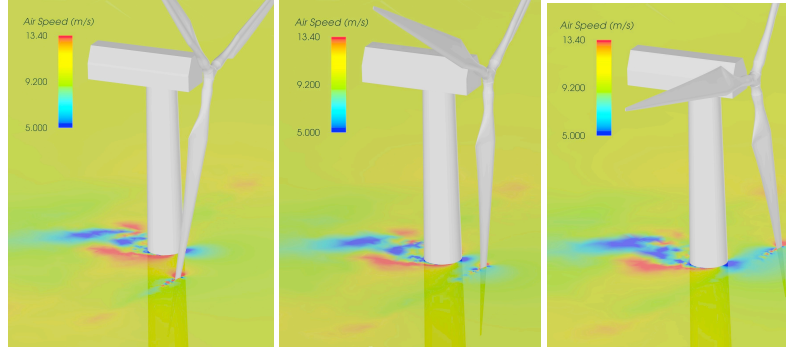


Fig. 28 Wind speed contours at 80% spanwise station as the blade passes the tower.

of FSI it is not as visible due to the relatively high torque oscillations. The “dips” in the aerodynamic torque can be seen at 60° , 180° , and 300° azimuthal angle, which is precisely when one of the three blades is passing the tower. The computed values of the aerodynamic torque are plotted together with field test results from [94]. The upper and lower dashed lines indicate the aerodynamic torque bounds, while the middle dashed line gives its average value. Both the aerodynamic and FSI results compare very well with the field test data. Figure 27 shows the relative wind speed at the 70% spanwise station rotated to the reference configuration to illustrate the blade deflection and complexity of boundary-layer turbulent flow. Figure 28 shows the flow field as the blade passes the tower.

5 Concluding Remarks

We provided an overview of the aerodynamic and FSI analysis of wind turbines carried out in recent years with the ALE-VMS and ST-VMS methods. The techniques complementing these core methods include weak enforcement of the essential boundary conditions, NURBS-based isogeometric analysis, using NURBS basis functions in temporal representation of the rotor motion, mesh motion and also in remeshing, rotation representation with constant angular velocity, Kirchhoff–Love shell modeling of the rotor-blade structure, and full FSI coupling. Some of these techniques were included in our overview. The wind-turbine analysis cases presented include the aerodynamics of wind-turbine rotor and tower and the FSI that accounts for the deformation of the rotor blades. The specific wind turbines considered were NREL 5MW, NREL Phase VI and Micon 65/13M, all at full scale. In the case of NREL Phase VI and Micon 65/13M we also presented a successful comparison with the experimental data. Overall, this article demonstrates that the ALE-VMS and ST-VMS methods, together with some new supporting techniques, have brought the aerodynamic and FSI analysis of wind turbines to a new level, where such analyses can contribute more to simulation-based design and testing.

Acknowledgements

We wish to thank the Texas Advanced Computing Center (TACC) and the San Diego Supercomputing Center (SDSC) for providing HPC resources that have contributed to the research results reported in this article. The first author acknowledges the support of the NSF CAREER Award, the NSF Award CBET-1306869, and the Air Force Office of Scientific Research Award FA9550-12-1-0005. The ST-VMS part of the work was supported by ARO grants W911NF-09-1-0346 and W911NF-12-1-0162 (third author) and Rice–Waseda Research Agreement (second author).

References

- [1] J.M. Jonkman and M.L. Buhl, “FAST user’s guide”, Technical Report NREL/EL-500-38230, National Renewable Energy Laboratory, Golden, CO, 2005.
- [2] 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, Golden, CO, 2009.
- [3] N.N. Sørensen, J.A. Michelsen, and S. Schreck, “Navier–Stokes predictions of the NREL Phase VI rotor in the NASA Ames 80 ft \times 120 ft wind tunnel”, *Wind Energy*, **5** (2002) 151–169.
- [4] A.L. Pape and J. Lecanu, “3D Navier–Stokes computations of a stall-regulated wind turbine”, *Wind Energy*, **7** (2004) 309–324.
- [5] 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, doi: [10.1002/flid.2400](https://doi.org/10.1002/flid.2400).
- [6] K. Takizawa, B. Henicke, T.E. Tezduyar, M.-C. Hsu, and Y. Bazilevs, “Stabilized space–time computation of wind-turbine rotor aerodynamics”, *Computational Mechanics*, **48** (2011) 333–344, doi: [10.1007/s00466-011-0589-2](https://doi.org/10.1007/s00466-011-0589-2).
- [7] C. Kong, J. Bang, and Y. Sugiyama, “Structural investigation of composite wind turbine blade considering various load cases and fatigue life”, *Energy*, **30** (2005) 2101–2114.
- [8] M.O.L. Hansen, J.N. Sørensen, S. Voutsinas, N. Sørensen, and H.A. Madsen, “State of the art in wind turbine aerodynamics and aeroelasticity”, *Progress in Aerospace Sciences*, **42** (2006) 285–330.
- [9] F.M. Jensen, B.G. Falzon, J. Ankersen, and H. Stang, “Structural testing and numerical simulation of a 34 m composite wind turbine blade”, *Composite Structures*, **76** (2006) 52–61.
- [10] J. Kiendl, Y. Bazilevs, M.-C. Hsu, R. Wüchner, and K.-U. Bletzinger, “The bending strip method for isogeometric analysis of Kirchhoff–Love shell struc-

- tures comprised of multiple patches”, *Computer Methods in Applied Mechanics and Engineering*, **199** (2010) 2403–2416.
- [11] Y. Bazilevs, M.-C. Hsu, J. Kiendl, and D.J. Benson, “A computational procedure for pre-bending of wind turbine blades”, *International Journal for Numerical Methods in Engineering*, **89** (2012) 323–336.
 - [12] 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.
 - [13] 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.
 - [14] J.A. Cottrell, A. Reali, Y. Bazilevs, and T.J.R. Hughes, “Isogeometric analysis of structural vibrations”, *Computer Methods in Applied Mechanics and Engineering*, **195** (2006) 5257–5297.
 - [15] Y. Bazilevs, L.B. da Veiga, J.A. Cottrell, T.J.R. Hughes, and G. Sangalli, “Isogeometric analysis: Approximation, stability and error estimates for h -refined meshes”, *Mathematical Models and Methods in Applied Sciences*, **16** (2006) 1031–1090.
 - [16] J.A. Cottrell, T.J.R. Hughes, and A. Reali, “Studies of refinement and continuity in isogeometric structural analysis”, *Computer Methods in Applied Mechanics and Engineering*, **196** (2007) 4160–4183.
 - [17] J.A. Cottrell, T.J.R. Hughes, and Y. Bazilevs, *Isogeometric Analysis: Toward Integration of CAD and FEA*. Wiley, Chichester, 2009.
 - [18] J.A. Evans, Y. Bazilevs, I. Babuška, and T.J.R. Hughes, “ n -Widths, sup-infs, and optimality ratios for the k -version of the isogeometric finite element method”, *Computer Methods in Applied Mechanics and Engineering*, **198** (2009) 1726–1741.
 - [19] M.R. Dörfel, B. Jüttler, and B. Simeon, “Adaptive isogeometric analysis by local h -refinement with T-splines”, *Computer Methods in Applied Mechanics and Engineering*, **199** (2010) 264–275.
 - [20] Y. Bazilevs, V.M. Calo, J.A. Cottrell, J.A. Evans, T.J.R. Hughes, S. Lipton, M.A. Scott, and T.W. Sederberg, “Isogeometric analysis using T-splines”, *Computer Methods in Applied Mechanics and Engineering*, **199** (2010) 229–263.
 - [21] 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.
 - [22] Y. Bazilevs, C. Michler, V.M. Calo, and T.J.R. Hughes, “Weak Dirichlet boundary conditions for wall-bounded turbulent flows”, *Computer Methods in Applied Mechanics and Engineering*, **196** (2007) 4853–4862.
 - [23] Y. Bazilevs, C. Michler, V.M. Calo, and T.J.R. Hughes, “Isogeometric variational multiscale modeling of wall-bounded turbulent flows with weakly en-

- forced boundary conditions on unstretched meshes”, *Computer Methods in Applied Mechanics and Engineering*, **199** (2010) 780–790.
- [24] I. Akkerman, Y. Bazilevs, V.M. Calo, T.J.R. Hughes, and S. Hulshoff, “The role of continuity in residual-based variational multiscale modeling of turbulence”, *Computational Mechanics*, **41** (2008) 371–378.
 - [25] 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, doi: [10.1016/j.cma.2009.06.019](https://doi.org/10.1016/j.cma.2009.06.019).
 - [26] 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.
 - [27] T. Elguedj, Y. Bazilevs, V.M. Calo, and T.J.R. Hughes, “B-bar and F-bar projection methods for nearly incompressible linear and nonlinear elasticity and plasticity using higher-order nurbs elements”, *Computer Methods in Applied Mechanics and Engineering*, **197** (2008) 2732–2762.
 - [28] S. Lipton, J.A. Evans, Y. Bazilevs, T. Elguedj, and T.J.R. Hughes, “Robustness of isogeometric structural discretizations under severe mesh distortion”, *Computer Methods in Applied Mechanics and Engineering*, **199** (2010) 357–373.
 - [29] D.J. Benson, Y. Bazilevs, E. De Luycker, M.-C. Hsu, M. Scott, T.J.R. Hughes, and T. Belytschko, “A generalized finite element formulation for arbitrary basis functions: from isogeometric analysis to XFEM”, *International Journal for Numerical Methods in Engineering*, **83** (2010) 765–785.
 - [30] D.J. Benson, Y. Bazilevs, M.-C. Hsu, and T.J.R. Hughes, “Isogeometric shell analysis: The Reissner–Mindlin shell”, *Computer Methods in Applied Mechanics and Engineering*, **199** (2010) 276–289.
 - [31] J. Kiendl, K.-U. Bletzinger, J. Linhard, and R. Wüchner, “Isogeometric shell analysis with Kirchhoff–Love elements”, *Computer Methods in Applied Mechanics and Engineering*, **198** (2009) 3902–3914.
 - [32] Y. Zhang, Y. Bazilevs, S. Goswami, C. Bajaj, and T.J.R. Hughes, “Patient-specific vascular nurbs modeling for isogeometric analysis of blood flow”, *Computer Methods in Applied Mechanics and Engineering*, **196** (2007) 2943–2959.
 - [33] 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.
 - [34] 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.
 - [35] 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.

- [36] 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.
- [37] F. Cirak, M. Ortiz, and P. Schröder, “Subdivision surfaces: a new paradigm for thin shell analysis”, *International Journal for Numerical Methods in Engineering*, **47** (2000) 2039–2072.
- [38] F. Cirak and M. Ortiz, “Fully C^1 -conforming subdivision elements for finite deformation thin shell analysis”, *International Journal for Numerical Methods in Engineering*, **51** (2001) 813–833.
- [39] F. Cirak, M.J. Scott, E.K. Antonsson, M. Ortiz, and P. Schröder, “Integrated modeling, finite-element analysis, and engineering design for thin-shell structures using subdivision”, *Computer-Aided Design*, **34** (2002) 137–148.
- [40] 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.
- [41] 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.
- [42] 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.
- [43] K. Takizawa and T.E. Tezduyar, “Multiscale space–time fluid–structure interaction techniques”, *Computational Mechanics*, **48** (2011) 247–267, doi: [10.1007/s00466-011-0571-z](https://doi.org/10.1007/s00466-011-0571-z).
- [44] K. Takizawa and T.E. Tezduyar, “Space–time fluid–structure interaction methods”, *Mathematical Models and Methods in Applied Sciences*, **22** (2012) 1230001, doi: [10.1142/S0218202512300013](https://doi.org/10.1142/S0218202512300013).
- [45] T.E. Tezduyar, “Stabilized finite element formulations for incompressible flow computations”, *Advances in Applied Mechanics*, **28** (1992) 1–44, doi: [10.1016/S0065-2156\(08\)70153-4](https://doi.org/10.1016/S0065-2156(08)70153-4).
- [46] 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, doi: [10.1016/0045-7825\(92\)90059-S](https://doi.org/10.1016/0045-7825(92)90059-S).
- [47] 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, doi: [10.1016/0045-7825\(92\)90060-W](https://doi.org/10.1016/0045-7825(92)90060-W).
- [48] T.E. Tezduyar, “Computation of moving boundaries and interfaces and stabilization parameters”, *International Journal for Numerical Methods in Fluids*, **43** (2003) 555–575, doi: [10.1002/flid.505](https://doi.org/10.1002/flid.505).

- [49] 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, doi: [10.1002/flid.1430](https://doi.org/10.1002/flid.1430).
- [50] Y. Bazilevs, K. Takizawa, and T.E. Tezduyar, *Computational Fluid–Structure Interaction: Methods and Applications*. Wiley, February 2013, ISBN 978-0470978771.
- [51] Y. Bazilevs and T.J.R. Hughes, “Weak imposition of Dirichlet boundary conditions in fluid mechanics”, *Computers and Fluids*, **36** (2007) 12–26.
- [52] J. Nitsche, “Über ein variationsprinzip zur losung von Dirichlet-problemen bei verwendung von teilraumen, die keinen randbedingungen unterworfen sind”, *Abh. Math. Univ. Hamburg*, **36** (1971) 9–15.
- [53] D.N. Arnold, F. Brezzi, B. Cockburn, and L.D. Marini, “Unified analysis of Discontinuous Galerkin methods for elliptic problems”, *SIAM Journal of Numerical Analysis*, **39** (2002) 1749–1779.
- [54] 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”, *Computer Methods in Applied Mechanics and Engineering*, **32** (1982) 199–259.
- [55] 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, doi: [10.1016/0045-7825\(92\)90141-6](https://doi.org/10.1016/0045-7825(92)90141-6).
- [56] S. Mittal and T.E. Tezduyar, “A finite element study of incompressible flows past oscillating cylinders and aerofoils”, *International Journal for Numerical Methods in Fluids*, **15** (1992) 1073–1118, doi: [10.1002/flid.1650150911](https://doi.org/10.1002/flid.1650150911).
- [57] 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, doi: [10.1002/flid.1650211011](https://doi.org/10.1002/flid.1650211011).
- [58] 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, doi: [10.1016/S0045-7825\(00\)00204-8](https://doi.org/10.1016/S0045-7825(00)00204-8).
- [59] 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, doi: [10.1016/j.cma.2004.09.014](https://doi.org/10.1016/j.cma.2004.09.014).
- [60] K. Takizawa and T.E. Tezduyar, “Computational methods for parachute fluid–structure interactions”, *Archives of Computational Methods in Engineering*, **19** (2012) 125–169, doi: [10.1007/s11831-012-9070-4](https://doi.org/10.1007/s11831-012-9070-4).
- [61] 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, doi: [10.1002/cnm.1433](https://doi.org/10.1002/cnm.1433).

- [62] K. Takizawa, Y. Bazilevs, and T.E. Tezduyar, “Space–time and ALE–VMS techniques for patient-specific cardiovascular fluid–structure interaction modeling”, *Archives of Computational Methods in Engineering*, **19** (2012) 171–225, doi: [10.1007/s11831-012-9071-3](https://doi.org/10.1007/s11831-012-9071-3).
- [63] K. Takizawa, K. Schjodt, A. Puntel, N. Kostov, and T.E. Tezduyar, “Patient-specific computer modeling of blood flow in cerebral arteries with aneurysm and stent”, *Computational Mechanics*, **50** (2012) 675–686, doi: [10.1007/s00466-012-0760-4](https://doi.org/10.1007/s00466-012-0760-4).
- [64] K. Takizawa, M. Fritze, D. Montes, T. Spielman, and T.E. Tezduyar, “Fluid–structure interaction modeling of ringsail parachutes with disreefing and modified geometric porosity”, *Computational Mechanics*, **50** (2012) 835–854, doi: [10.1007/s00466-012-0761-3](https://doi.org/10.1007/s00466-012-0761-3).
- [65] K. Takizawa, D. Montes, M. Fritze, S. McIntyre, J. Boben, and T.E. Tezduyar, “Methods for FSI modeling of spacecraft parachute dynamics and cover separation”, *Mathematical Models and Methods in Applied Sciences*, **23** (2013) 307–338, doi: [10.1142/S0218202513400058](https://doi.org/10.1142/S0218202513400058).
- [66] K. Takizawa, T.E. Tezduyar, J. Boben, N. Kostov, C. Boswell, and A. Buscher, “Fluid–structure interaction modeling of clusters of spacecraft parachutes with modified geometric porosity”, *Computational Mechanics*, **52** (2013) 1351–1364, doi: [10.1007/s00466-013-0880-5](https://doi.org/10.1007/s00466-013-0880-5).
- [67] K. Takizawa, K. Schjodt, A. Puntel, N. Kostov, and T.E. Tezduyar, “Patient-specific computational analysis of the influence of a stent on the unsteady flow in cerebral aneurysms”, *Computational Mechanics*, **51** (2013) 1061–1073, doi: [10.1007/s00466-012-0790-y](https://doi.org/10.1007/s00466-012-0790-y).
- [68] 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, doi: [10.1002/flid.2415](https://doi.org/10.1002/flid.2415).
- [69] M. Manguoglu, K. Takizawa, A.H. Sameh, and T.E. Tezduyar, “A parallel sparse algorithm targeting arterial fluid mechanics computations”, *Computational Mechanics*, **48** (2011) 377–384, doi: [10.1007/s00466-011-0619-0](https://doi.org/10.1007/s00466-011-0619-0).
- [70] 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, doi: [10.1007/BF00350249](https://doi.org/10.1007/BF00350249).
- [71] M. Behr and T. Tezduyar, “The Shear-Slip Mesh Update Method”, *Computer Methods in Applied Mechanics and Engineering*, **174** (1999) 261–274, doi: [10.1016/S0045-7825\(98\)00299-0](https://doi.org/10.1016/S0045-7825(98)00299-0).
- [72] 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, doi: [10.1016/S0045-7825\(00\)00388-1](https://doi.org/10.1016/S0045-7825(00)00388-1).
- [73] K. Takizawa, B. Henicke, D. Montes, T.E. Tezduyar, M.-C. Hsu, and Y. Bazilevs, “Numerical-performance studies for the stabilized space–time computation of wind-turbine rotor aerodynamics”, *Computational Mechanics*, **48** (2011) 647–657, doi: [10.1007/s00466-011-0614-5](https://doi.org/10.1007/s00466-011-0614-5).

- [74] Y. Bazilevs, M.-C. Hsu, K. Takizawa, and T.E. Tezduyar, “ALE-VMS and ST-VMS methods for computer modeling of wind-turbine rotor aerodynamics and fluid–structure interaction”, *Mathematical Models and Methods in Applied Sciences*, **22** (2012) 1230002, doi: [10.1142/S0218202512300025](https://doi.org/10.1142/S0218202512300025).
- [75] K. Takizawa, T.E. Tezduyar, S. McIntyre, N. Kostov, R. Kolesar, and C. Habluetzel, “Space–time VMS computation of wind-turbine rotor and tower aerodynamics”, *Computational Mechanics*, **53** (2014) 1–15, doi: [10.1007/s00466-013-0888-x](https://doi.org/10.1007/s00466-013-0888-x).
- [76] K. Takizawa, B. Henicke, A. Puntel, T. Spielman, and T.E. Tezduyar, “Space–time computational techniques for the aerodynamics of flapping wings”, *Journal of Applied Mechanics*, **79** (2012) 010903, doi: [10.1115/1.4005073](https://doi.org/10.1115/1.4005073).
- [77] K. Takizawa, B. Henicke, A. Puntel, N. Kostov, and T.E. Tezduyar, “Space–time techniques for computational aerodynamics modeling of flapping wings of an actual locust”, *Computational Mechanics*, **50** (2012) 743–760, doi: [10.1007/s00466-012-0759-x](https://doi.org/10.1007/s00466-012-0759-x).
- [78] K. Takizawa, N. Kostov, A. Puntel, B. Henicke, and T.E. Tezduyar, “Space–time computational analysis of bio-inspired flapping-wing aerodynamics of a micro aerial vehicle”, *Computational Mechanics*, **50** (2012) 761–778, doi: [10.1007/s00466-012-0758-y](https://doi.org/10.1007/s00466-012-0758-y).
- [79] K. Takizawa, B. Henicke, A. Puntel, N. Kostov, and T.E. Tezduyar, “Computer modeling techniques for flapping-wing aerodynamics of a locust”, *Computers & Fluids*, **85** (2013) 125–134, doi: [10.1016/j.compfluid.2012.11.008](https://doi.org/10.1016/j.compfluid.2012.11.008).
- [80] T.E. Tezduyar, M. Behr, S. Mittal, and A.A. Johnson, “Computation of unsteady incompressible flows with the finite element methods – space–time formulations, iterative strategies and massively parallel implementations”, in *New Methods in Transient Analysis*, PVP-Vol.246/AMD-Vol.143, ASME, New York, (1992) 7–24.
- [81] T. Tezduyar, S. Aliabadi, M. Behr, A. Johnson, and S. Mittal, “Parallel finite-element computation of 3D flows”, *Computer*, **26** (1993) 27–36, doi: [10.1109/2.237441](https://doi.org/10.1109/2.237441).
- [82] A.A. Johnson and T.E. Tezduyar, “Mesh update strategies in parallel finite element computations of flow problems with moving boundaries and interfaces”, *Computer Methods in Applied Mechanics and Engineering*, **119** (1994) 73–94, doi: [10.1016/0045-7825\(94\)00077-8](https://doi.org/10.1016/0045-7825(94)00077-8).
- [83] T.E. Tezduyar, “Finite element methods for flow problems with moving boundaries and interfaces”, *Archives of Computational Methods in Engineering*, **8** (2001) 83–130, doi: [10.1007/BF02897870](https://doi.org/10.1007/BF02897870).
- [84] M.-C. Hsu and Y. Bazilevs, “Fluid–structure interaction modeling of wind turbines: simulating the full machine”, *Computational Mechanics*, **50** (2012) 821–833.
- [85] M.-C. Hsu, I. Akkerman, and Y. Bazilevs, “Finite element simulation of wind turbine aerodynamics: Validation study using NREL Phase VI experiment”, *Wind Energy*, **17** (2014) 461–481.
- [86] M.M. Hand, D.A. Simms, L.J. Fingersh, D.W. Jager, J.R. Cotrell, S. Schreck, and S.M. Larwood, “Unsteady aerodynamics experiment phase VI: Wind

- tunnel test configurations and available data campaigns”, Technical Report NREL/TP-500-29955, National Renewable Energy Laboratory, Golden, CO, 2001.
- [87] A. Korobenko, M.-C. Hsu, I. Akkerman, J. Tippmann, and Y. Bazilevs, “Structural mechanics modeling and FSI simulation of wind turbines”, *Mathematical Models and Methods in Applied Sciences*, **23** (2013) 249–272.
 - [88] Y. Bazilevs, M.-C. Hsu, and M.A. Scott, “Isogeometric fluid–structure interaction analysis with emphasis on non-matching discretizations, and with application to wind turbines”, *Computer Methods in Applied Mechanics and Engineering*, **249-252** (2012) 28–41.
 - [89] 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, doi: [10.1016/j.cma.2005.08.023](https://doi.org/10.1016/j.cma.2005.08.023).
 - [90] 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.
 - [91] 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.
 - [92] 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.
 - [93] 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.
 - [94] J.R. Zayas and W.D. Johnson, “3X-100 blade field test”, Report of the Sandia National Laboratories, Wind Energy Technology Department, 2008.
 - [95] J.H. Sutherland, P.L. Jones, and B.A. Neal, “The long-term inflow and structural test program”, Proceedings of the 2001 ASME Wind Energy Symposium, p.162, 2001.
 - [96] D. Berry and T. Ashwill, “Design of 9-meter carbon-fiberglass prototype blades: CX-100 and TX-100”, Report of the Sandia National Laboratories, 2007.
 - [97] J.R. White, D.E. Adams, and M.A. Rumsey, “Modal analysis of CX-100 rotor blade and Micon 65/13 wind turbine”, Structural Dynamics and Renewable Energy, Volume 1, Conference Proceedings of the Society for Experimental Mechanics Series 10, 2011.
 - [98] T. Marinone, B. LeBlanc, J. Harvie, C. Niezrecki, and P. Avitabile, “Modal testing of a 9 m CX-100 turbine blade”, Topics in Experimental Dynamics Substructuring and Wind Turbine Dynamics, Volume 2, Conference Proceedings of the Society for Experimental Mechanics Series 27, 2012.