Mathematical Models and Methods in Applied Sciences

ALE-VMS AND ST-VMS METHODS FOR COMPUTER MODELING OF WIND-TURBINE ROTOR AERODYNAMICS AND FLUID–STRUCTURE INTERACTION

YURI BAZILEVS

Department of Structural Engineering, University of California, San Diego,
9500 Gilman Drive, La Jolla, California 92037, USA.
yuri@ucsd.edu

MING-CHEN HSU

Department of Structural Engineering, University of California, San Diego,
9500 Gilman Drive, La Jolla, California 92037, USA

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 – MS 321
6100 Main Street, Houston, TX 77005, USA

Received (Day Month Year)
Revised (Day Month Year)
Communicated by (xxxxxxxxxx)

We provide an overview of the Arbitrary Lagrangian–Eulerian Variational Multiscale (ALE-VMS) and Space–Time Variational Multiscale (ST-VMS) methods we have developed for computer modeling of wind-turbine rotor aerodynamics and fluid–structure interaction (FSI). The related techniques described include weak enforcement of the essential boundary conditions, Kirchhoff–Love shell modeling of the rotor-blade structure, NURBS-based isogeometric analysis, and full FSI coupling. We present results from application of these methods to computer modeling of NREL 5MW and NREL Phase VI wind-turbine rotors at full scale, including comparison with experimental data.

Keywords: ALE-VMS method; ST-VMS method; DSD/SST formulation; DSD/SST-VMST formulation; weak enforcement of essential boundary conditions; finite elements; isogeometric analysis; NURBS; rotation-free shells; wind-turbine rotor; NREL Phase VI wind turbine; NREL 5MW offshore wind turbine; fluid–structure interaction; blade pre-bending.

AMS Subject Classification: 22E46, 53C35, 57S20

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 predictive simulation 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 [3–8], 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 [9–14]. In the recent work of [15] it was shown that coupled FSI modeling and simulation of wind turbines is important for

accurately predicting their mechanical behavior at full scale.

We feel that in order 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 and are described in greater detail in the body of this review paper.

Isogeometric Analysis (IGA), first introduced in [16] and further expanded on in [17–28], is adopted as the geometry modeling and simulation framework for wind turbines in some of the examples presented in this paper. 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 [29–34], nonlinear structures [13,35–39], and FSI [40–43], 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 [44].

The blade structure is governed by the isogeometric rotation-free shell formulation with the aid of the bending-strip method [13]. 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 [39]. Although NURBS-based IGA is employed in this work, other discretizations such as T-splines [23, 24] or subdivision surfaces [45–47], are perfectly suited for the proposed structural modeling method.

In addition, an isogeometric representation of the analysis-suitable geometry may be used to construct tetrahedral and hexahedral meshes for computations using the finite element method (FEM). In this paper, we use such tetrahedral meshes for wind-turbine rotor computation using the Arbitrary Lagrangian–Eulerian (ALE) [48] and Deforming-Spatial-Domain/Stabilized Space–Time (DSD/SST) [49–53] versions of the Residual-Based Variational Multiscale (RBVMS) method [29, 54–56]. We call the ALE version [6, 42] "ALE-VMS", and the DSD/SST version [57] "DSD/SST-VMST". We will use the acronym "ST-VMS" (meaning "Space–Time VMS") interchangeably with DSD/SST-VMST.

The ALE-VMS formulation originated from the RBVMS formulation of incompressible turbulent flows proposed in [29] 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 [42] in the context of FSI. Although ALE-VMS gave reasonably good results for several important turbulent flows, it was evident in [29, 32] 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 [58], which significantly improved the performance of the ALE-VMS formulation in the presence of unresolved boundary layers [30, 31, 34]. The

weak boundary condition formulation may be thought of as an extension of Nitsche's method [59] 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., [60]), 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 [49–51] as a general-purpose interface-tracking (moving-mesh) technique for flow computations involving moving boundaries and interfaces, including FSI and flows with moving mechanical components. It is based on the Streamline-Upwind/Petrov-Galerkin (SUPG) [61] and Pressure-Stabilizing/Petrov-Galerkin (PSPG) [49, 62] 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 [63] for vortex-induced vibrations of a cylinder and in [64] 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 [65–67] and evolving to computations with direct coupling [68, 69]. New versions of the DSD/SST formulation introduced in [53] are the core technologies of the Stabilized Space–Time FSI (SSTFSI) technique, which was also introduced in [53]. The SSTFSI technique, combined with a number of special techniques [53,70–83], have been used in some of the most challenging parachute FSI computations [72, 75, 78, 80, 82, 84, 85], and also in a good number of patient-specific arterial FSI computations [74, 76, 77, 79, 81, 83]. Computations with the SSTFSI technique also received a substantial attention in research related to iterative solution of large linear systems [86–90]. In application of the DSD/SST formulation to flows with moving mechanical components, the Shear–Slip Mesh Update Method (SSMUM) [91–93] 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 [91]). 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 [92] and flow around a helicopter with its rotor in motion [93].

The DSD/SST formulation was elevated in [57] to a new version with an advanced turbulence model. This most recent DSD/SST formulation with the turbulence model is a space–time version [57] of the RBVMS method [29, 54–56]. It was named in [57] "DSD/SST-VMST" (i.e. the version with the VMS turbulence model), which is also called "ST-VMS" in this paper. It was successfully tested on wind-turbine rotor aerodynamics

in [7,94,95]. The original DSD/SST version was named in [57] "DSD/SST-SUPS" (i.e. the version with the SUPG/PSPG stabilization).

In Section 2, we review the discrete formulations of aerodynamics equations employed in this work, namely, the ALE-VMS and ST-VMS formulations. We also give the formulation for weakly-enforced essential boundary conditions. In Section 3, we provide a description of the wind-turbine rotor geometry modeling and summarize the aerodynamics computations for the 5MW wind-turbine rotor defined in [2]. The results presented are from reference [6] and include simulations using the NURBS-based ALE-VMS and FEM-based ST-VMS formulations. In Section 4, we present the computational results from [96], which make use of the FEM-based ALE-VMS formulation with weakly-enforced boundary conditions. Here the aerodynamics simulations of the National Renewable Energy Lab (NREL) Phase VI two-bladed rotor (see [97]) are performed to validate our computations against an extensive set of experimental data available for this test case. In Section 5, we review the structural mechanics formulation, which is based on the Kirchhoff–Love thin-shell theory and the bending-strip method (see [13, 15, 39]). The FSI coupling is described in Section 6, where we also provide a discussion of the mesh motion procedures employed in the FSI simulations of the 5MW wind-turbine rotor reported in Section 7, which was reported earlier in [15]. In Section 8, we present a method and algorithm for pre-bending of wind-turbine blades, which was recently proposed in [14]. We draw conclusions in Section 9.

2. ALE-VMS and ST-VMS Formulations of the Navier–Stokes Equations of Incompressible Flows

2.1. *Continuous problem*

Let $\Omega_t \in \mathbb{R}^d$, $d = 2, 3$, be the spatial domain of the aerodynamics problem with boundary Γ_t at time $t \in (0, T)$. The subscript t indicates that the fluid mechanics spatial domain is time-dependent. The Navier–Stokes equations of incompressible flows [a] in the ALE frame may be written on Ω_t and $\forall t \in (0, T)$ as

$$\rho\left(\left.\frac{\partial \mathbf{u}}{\partial t}\right|_{\hat{x}} + (\mathbf{u} - \hat{\mathbf{u}}) \cdot \boldsymbol{\nabla}_x \mathbf{u} - \mathbf{f}\right) - \boldsymbol{\nabla}_x \cdot \boldsymbol{\sigma} = \mathbf{0},$$
$$\boldsymbol{\nabla}_x \cdot \mathbf{u} = 0, \tag{2.1}$$

where ρ, $\mathbf{u}$, and $\mathbf{f}$ are the density, velocity and the external force, respectively, and the stress tensor $\boldsymbol{\sigma}$ is defined as

$$\boldsymbol{\sigma}(\mathbf{u}, p) = -p\mathbf{I} + 2\mu\boldsymbol{\varepsilon}(\mathbf{u}). \tag{2.2}$$

Here p is the pressure, $\mathbf{I}$ is the identity tensor, μ is the dynamic viscosity, and $\boldsymbol{\varepsilon}(\mathbf{u})$ is the strain-rate tensor given by

$$\boldsymbol{\varepsilon}(\mathbf{u}) = \frac{1}{2}\left(\boldsymbol{\nabla}_x \mathbf{u} + \boldsymbol{\nabla}_x \mathbf{u}^T\right). \tag{2.3}$$

[a]Although aerodynamic phenomena are generally described using the Navier–Stokes equations of compressible flows, the incompressible-flow assumption is valid for the present application.

In Eq. (2.1), the notation $\left|_{\hat{x}}\right.$ implies that the time derivative is taken with respect to a fixed referential-domain spatial coordinates $\hat{\mathbf{x}}$, and $\hat{\mathbf{u}}$ is the velocity of the fluid domain Ω_t. The spatial gradients are taken with respect to the spatial coordinates $\mathbf{x}$ of the current configuration, and that is reflected in the subscript on the gradient operator.

2.2. *ALE-VMS formulation*

The ALE-VMS formulation of the continuum aerodynamics formulation is given as follows: find $\mathbf{u}^h \in \mathcal{S}_u^h$ and $p^h \in \mathcal{S}_p^h$, such that $\forall \mathbf{w}^h \in \mathcal{V}_u^h$ and $\forall q^h \in \mathcal{V}_p^h$:

$$\begin{aligned}
&\int_{\Omega_t} \mathbf{w}^h \cdot \rho \left(\left.\frac{\partial \mathbf{u}^h}{\partial t}\right|_{\hat{x}} + \left(\mathbf{u}^h - \hat{\mathbf{u}}^h\right) \cdot \boldsymbol{\nabla}_x \mathbf{u}^h - \mathbf{f}^h \right) \mathrm{d}\Omega + \int_{\Omega_t} \boldsymbol{\varepsilon}\left(\mathbf{w}^h\right) : \boldsymbol{\sigma}\left(\mathbf{u}^h, p^h\right) \mathrm{d}\Omega \\
&- \int_{(\Gamma_t)_{\mathrm{h}}} \mathbf{w}^h \cdot \mathbf{h}^h \, \mathrm{d}\Gamma + \int_{\Omega_t} q^h \boldsymbol{\nabla}_x \cdot \mathbf{u}^h \, \mathrm{d}\Omega \\
&+ \sum_{e=1}^{N_{\mathrm{el}}} \int_{\Omega_t^e} \tau_{\mathrm{SUPS}} \left(\left(\mathbf{u}^h - \hat{\mathbf{u}}^h\right) \cdot \boldsymbol{\nabla}_x \mathbf{w}^h + \frac{\boldsymbol{\nabla}_x q^h}{\rho} \right) \cdot \mathbf{r}_{\mathrm{M}}\left(\mathbf{u}^h, p^h\right) \mathrm{d}\Omega \\
&+ \sum_{e=1}^{N_{\mathrm{el}}} \int_{\Omega_t^e} \rho \nu_{\mathrm{LSIC}} \boldsymbol{\nabla}_x \cdot \mathbf{w}^h r_{\mathrm{C}}(\mathbf{u}^h, p^h) \, \mathrm{d}\Omega \\
&- \sum_{e=1}^{N_{\mathrm{el}}} \int_{\Omega_t^e} \tau_{\mathrm{SUPS}} \mathbf{w}^h \cdot \left(\mathbf{r}_{\mathrm{M}}\left(\mathbf{u}^h, p^h\right) \cdot \boldsymbol{\nabla}_x \mathbf{u}^h \right) \mathrm{d}\Omega \\
&- \sum_{e=1}^{N_{\mathrm{el}}} \int_{\Omega_t^e} \frac{\boldsymbol{\nabla}_x \mathbf{w}^h}{\rho} : \left(\tau_{\mathrm{SUPS}} \mathbf{r}_{\mathrm{M}}\left(\mathbf{u}^h, p^h\right)\right) \otimes \left(\tau_{\mathrm{SUPS}} \mathbf{r}_{\mathrm{M}}\left(\mathbf{u}^h, p^h\right)\right) \mathrm{d}\Omega = 0. \qquad (2.4)
\end{aligned}$$

Here Ω_t is divided into N_{el} individual spatial finite element subdomains denoted by Ω_t^e. The finite-dimensional trial function spaces $\mathcal{S}_u^h$ for the velocity and $\mathcal{S}_p^h$ for the pressure, as well as the corresponding test function spaces $\mathcal{V}_u^h$ and $\mathcal{V}_p^h$ are assumed to be of equal order. In Eq. (2.4), $\mathbf{h}$ is the natural boundary condition, $(\Gamma_t)_{\mathrm{h}}$ is the part of the boundary where we specify that natural boundary condition, $\hat{\mathbf{u}}^h$ is the mesh velocity, and $\mathbf{r}_{\mathrm{M}}$ and r_{C} are the residuals of the momentum and continuity (incompressibility constraint) equations, respectively, given by

$$\mathbf{r}_{\mathrm{M}}(\mathbf{u}^h, p^h) = \rho \left(\left.\frac{\partial \mathbf{u}^h}{\partial t}\right|_{\hat{x}} + \left(\mathbf{u}^h - \hat{\mathbf{u}}^h\right) \cdot \boldsymbol{\nabla}_x \mathbf{u}^h - \mathbf{f}^h \right) - \boldsymbol{\nabla}_x \cdot \boldsymbol{\sigma}\left(\mathbf{u}^h, p^h\right) \qquad (2.5)$$

and

$$r_{\mathrm{C}}(\mathbf{u}^h, p^h) = \boldsymbol{\nabla}_x \cdot \mathbf{u}^h. \qquad (2.6)$$

Also in Eq. (2.4), τ_{SUPS} and ν_{LSIC} are the stabilization parameters defined in [42] as

$$\tau_{\mathrm{SUPS}} = \left(\frac{4}{\Delta t^2} + \left(\mathbf{u}^h - \hat{\mathbf{u}}^h\right) \cdot \mathbf{G} \left(\mathbf{u}^h - \hat{\mathbf{u}}^h\right) + C_I \nu^2 \mathbf{G} : \mathbf{G} \right)^{-1/2} \qquad (2.7)$$

and

$$\nu_{\mathrm{LSIC}} = \left(\mathrm{tr}\mathbf{G} \; \tau_{\mathrm{SUPS}}\right)^{-1}, \qquad (2.8)$$

where

$$\mathrm{tr}\mathbf{G} = \sum_{i=1}^{d} G_{ii} \tag{2.9}$$

is the trace of the element metric tensor $\mathbf{G}$, Δt is the time-step size, and C_I is a positive constant, independent of the mesh size, derived from an appropriate element-wise inverse estimate (see, e.g., [98–100]).

Remark 1. The stabilization parameters τ_{SUPS} and ν_{LSIC} in the above equations originate from stabilized finite element methods for fluid dynamics (see, e.g., [52, 61, 101–105]). The notation "SUPS", introduced in [57], indicates that there is a single stabilization parameter for the SUPG and PSPG stabilizations, instead of two separate parameters. The notation "LSIC", introduced in [104], denotes the stabilization based on least-squares on the incompressibility constraint. The stabilization parameters were designed and studied extensively in the context of stabilized finite element formulations of linear model problems of direct relevance to fluid mechanics. These model problems include advection–diffusion and Stokes equations. The design of τ_{SUPS} and ν_{LSIC} is such that optimal convergence with respect to the mesh size and polynomial order of discretization is attained for these cases (see, e.g., [105] and references therein). Furthermore, enhanced stability for advection-dominated flows and the ability to conveniently employ the same basis functions for velocity and pressure variables for incompressible flow are some of the attractive outcomes of this method. More recently, the stabilization parameters were derived in the context of the variational multiscale methods [54,106] and were interpreted as the appropriate averages of the small-scale Green's function, a key mathematical object in the theory of VMS methods (see [56] for an elaboration).

Remark 2. The ALE-VMS formulation is a moving-mesh extension of the RBVMS turbulence modeling technique proposed for stationary meshes in [29]. It was also presented in [42] for moving meshes in the context of FSI. In [107] the authors derived a variant of the ALE-VMS method in the context of residual-free bubbles [108]. Recently, a space–time (DSD/SST) version of the RBVMS method was introduced in [57] for computations with moving meshes (see next subsection).

2.3. *ST-VMS formulation*

In the DSD/SST method (see, e.g., [49–53, 57]), the finite element formulation is written over a sequence of N space–time slabs Q_n, where Q_n is the slice of the space–time domain between the time levels t_n and t_{n+1} (see Figure 1). At each time step, the integrations are performed over Q_n. The space–time finite element interpolation functions are continuous within a space–time slab, but discontinuous from one space–time slab to another. The notation $(\cdot)_n^-$ and $(\cdot)_n^+$ will denote the function values at t_n as approached from below and above. Each Q_n is decomposed into elements Q_n^e, where $e = 1, 2, \ldots, (N_{\mathrm{el}})_n$. The subscript n used with N_{el} is for the general case where the number of space–time elements may change from one space–time slab to another. The essential and natural boundary conditions are enforced

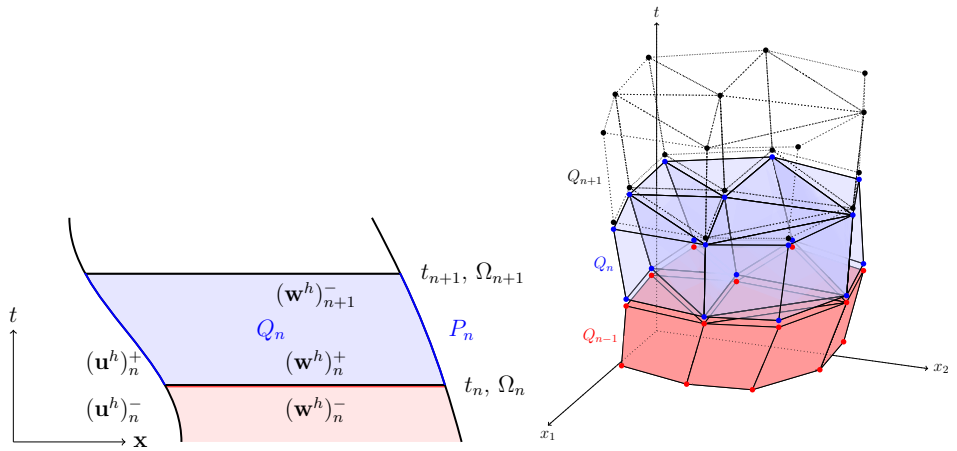


Fig. 1. Space–time slab in an abstract representation (left) and in a 2D context (right).

over $(P_n)_{\mathrm{g}}$ and $(P_n)_{\mathrm{h}}$, the complementary subsets of the lateral boundary of the space–time slab. The finite element trial function spaces $\mathcal{S}_u^h$ for velocity and $\mathcal{S}_p^h$ for pressure, and the test function spaces $\mathcal{V}_u^h$ and $\mathcal{V}_p^h = \mathcal{S}_p^h$ are defined by using, over Q_n, first-order polynomials in space and time. [b]

The ST (DSD/SST) version of the RBVMS method, first proposed in [57], is written as follows: given $(\mathbf{u}^h)_n^-$, find $\mathbf{u}^h \in \mathcal{S}_u^h$ and $p^h \in \mathcal{S}_p^h$, such that $\forall \mathbf{w}^h \in \mathcal{V}_u^h$ and $\forall q^h \in \mathcal{V}_p^h$:

$$
\begin{aligned}
&\int_{Q_n} \mathbf{w}^h \cdot \rho \left(\frac{\partial \mathbf{u}^h}{\partial t} + \boldsymbol{\nabla} \cdot (\mathbf{u}^h \otimes \mathbf{u}^h) - \mathbf{f}^h \right) \mathrm{d}Q + \int_{Q_n} \boldsymbol{\varepsilon}(\mathbf{w}^h) : \boldsymbol{\sigma}(\mathbf{u}^h, p^h) \mathrm{d}Q \\
&- \int_{(P_n)_{\mathrm{h}}} \mathbf{w}^h \cdot \mathbf{h}^h \mathrm{d}P + \int_{Q_n} q^h \boldsymbol{\nabla} \cdot \mathbf{u}^h \mathrm{d}Q + \int_{\Omega_n} (\mathbf{w}^h)_n^+ \cdot \rho \left((\mathbf{u}^h)_n^+ - (\mathbf{u}^h)_n^- \right) \mathrm{d}\Omega \\
&+ \sum_{e=1}^{(N_{\mathrm{el}})_n} \int_{Q_n^e} \frac{\tau_{\mathrm{SUPS}}}{\rho} \left[\rho \left(\frac{\partial \mathbf{w}^h}{\partial t} + \mathbf{u}^h \cdot \boldsymbol{\nabla} \mathbf{w}^h \right) + \boldsymbol{\nabla} q^h \right] \cdot \mathbf{r}_{\mathrm{M}} \mathrm{d}Q \\
&+ \sum_{e=1}^{(N_{\mathrm{el}})_n} \int_{Q_n^e} \rho \nu_{\mathrm{LSIC}} \boldsymbol{\nabla} \cdot \mathbf{w}^h r_{\mathrm{C}} \mathrm{d}Q \\
&+ \sum_{e=1}^{(N_{\mathrm{el}})_n} \int_{Q_n^e} \tau_{\mathrm{SUPS}} \mathbf{r}_{\mathrm{M}} \cdot \left(\boldsymbol{\nabla} \mathbf{w}^h \right) \mathbf{u}^h \mathrm{d}Q \\
&- \sum_{e=1}^{(N_{\mathrm{el}})_n} \int_{Q_n^e} \frac{\tau_{\mathrm{SUPS}}^2}{\rho} \mathbf{r}_{\mathrm{M}} \cdot \left(\boldsymbol{\nabla} \mathbf{w}^h \right) \mathbf{r}_{\mathrm{M}} \mathrm{d}Q = 0.
\end{aligned}
\tag{2.10}
$$

We call this "DSD/SST-VMST" (i.e. the version with the VMS turbulence model), and use "ST-VMS" (meaning "Space–Time VMS") interchangeably with DSD/SST-VMST.

[b] Although the trial and test function spaces for the ALE and ST formulations are different, in order to avoid introducing unneeded extra notation, we use the same symbols to denote these objects in both cases.

The stabilization parameters τ_{SUPS} and ν_{LSIC} can be calculated using the definitions given by Eqs. (2.7)–(2.9). However, the definitions that are commonly employed in conjunction with the DSD/SST formulation are those given in [52]:

$$\tau_{\text{SUPS}} = \left(\frac{1}{\tau_{\text{SUGN12}}^2} + \frac{1}{\tau_{\text{SUGN3}}^2} \right)^{-\frac{1}{2}}, \tag{2.11}$$

$$\tau_{\text{SUGN12}} = \left(\sum_{a=1}^{n_{en}} \left| \frac{\partial N_a}{\partial t} + \mathbf{u}^h \cdot \boldsymbol{\nabla} N_a \right| \right)^{-1}, \tag{2.12}$$

$$\tau_{\text{SUGN3}} = \frac{h_{\text{RGN}}^2}{4\nu}, \tag{2.13}$$

$$h_{\text{RGN}} = 2 \left(\sum_{a=1}^{n_{en}} |\mathbf{r} \cdot \boldsymbol{\nabla} N_a| \right)^{-1}, \tag{2.14}$$

$$\mathbf{r} = \frac{\boldsymbol{\nabla} \|\mathbf{u}^h\|}{\| \boldsymbol{\nabla} \|\mathbf{u}^h\| \|}, \tag{2.15}$$

and in [53]:

$$\nu_{\text{LSIC}} = \tau_{\text{SUPS}} \, \|\mathbf{u}^h - \hat{\mathbf{u}}^h\|^2, \tag{2.16}$$

where n_{en} is the number of (space–time) element nodes, and N_a is the space–time shape function associated with the space–time node a. As an alternative to the construction of τ_{SUPS} as given by Eqs. (2.11)–(2.15), another option was introduced in [53]. In that option, τ_{SUPS} is constructed based on separate definitions for the advection-dominated and transient-dominated limits:

$$\tau_{\text{SUPS}} = \left(\frac{1}{\tau_{\text{SUGN1}}^2} + \frac{1}{\tau_{\text{SUGN2}}^2} + \frac{1}{\tau_{\text{SUGN3}}^2} \right)^{-\frac{1}{2}}, \tag{2.17}$$

$$\tau_{\text{SUGN1}} = \left(\sum_{a=1}^{n_{en}} \left| \left(\mathbf{u}^h - \hat{\mathbf{u}}^h \right) \cdot \boldsymbol{\nabla} N_a \right| \right)^{-1}, \tag{2.18}$$

$$\tau_{\text{SUGN2}} = \frac{\Delta t}{2}. \tag{2.19}$$

It was noted in [53] that separating τ_{SUGN12} into its advection- and transient-dominated components as given by Eqs. (2.18)–(2.19) is equivalent to excluding the $\frac{\partial N_a}{\partial t}\big|_{\boldsymbol{\xi}}$ part of $\frac{\partial N_a}{\partial t}$ in Eq. (2.12), making that the definition for τ_{SUGN1}, and accounting for $\frac{\partial N_a}{\partial t}\big|_{\boldsymbol{\xi}}$ in the definition for τ_{SUGN2} given by Eq. (2.19). Here $\boldsymbol{\xi}$ is the vector of element coordinates, and $\frac{\partial}{\partial t}\big|_{\boldsymbol{\xi}}$ is equivalent to $\frac{\partial}{\partial t}\big|_{\hat{x}}$. Both notations for the same partial derivative are kept in the interest of backward compatibility with prior articles. For more ways of calculating τ_{SUPS} and ν_{LSIC}, see [33, 52, 104, 109–121]. References [52, 117] also include the Discontinuity-Capturing Directional Dissipation (DCDD) stabilization, which was introduced as an alternative to the LSIC stabilization.

Remark 3. The 6th and 7th terms in Eq. (2.10) are the SUPG/PSPG and LSIC stabilization terms, respectively. If we exclude the last two terms in Eq. (2.10), the formulation reduces

to the original DSD/SST formulation (with the advection term retained in the conservation-law form) under the conditions that $\tau_{\text{PSPG}} = \tau_{\text{SUPG}}$. This was named in [57] "DSD/SST-SUPS" (i.e., the version with the SUPG/PSPG stabilization).

For completeness, we also provide here the DSD/SST-SUPS formulation (from [52]): given $(\mathbf{u}^h)_n^-$, find $\mathbf{u}^h \in \mathcal{S}_u^h$ and $p^h \in \mathcal{S}_p^h$, such that $\forall \mathbf{w}^h \in \mathcal{V}_u^h$ and $\forall q^h \in \mathcal{V}_p^h$:

$$\begin{aligned}
&\int_{Q_n} \mathbf{w}^h \cdot \rho \left(\frac{\partial \mathbf{u}^h}{\partial t} + \mathbf{u}^h \cdot \boldsymbol{\nabla} \mathbf{u}^h - \mathbf{f}^h \right) \mathrm{d}Q + \int_{Q_n} \boldsymbol{\varepsilon}(\mathbf{w}^h) : \boldsymbol{\sigma}(\mathbf{u}^h, p^h) \mathrm{d}Q \\
&- \int_{(P_n)_{\text{h}}} \mathbf{w}^h \cdot \mathbf{h}^h \mathrm{d}P + \int_{Q_n} q^h \boldsymbol{\nabla} \cdot \mathbf{u}^h \mathrm{d}Q + \int_{\Omega_n} (\mathbf{w}^h)_n^+ \cdot \rho \left((\mathbf{u}^h)_n^+ - (\mathbf{u}^h)_n^- \right) \mathrm{d}\Omega \\
&+ \sum_{e=1}^{(N_{\text{el}})_n} \int_{Q_n^e} \frac{1}{\rho} \left[\tau_{\text{SUPG}} \rho \left(\frac{\partial \mathbf{w}^h}{\partial t} + \mathbf{u}^h \cdot \boldsymbol{\nabla} \mathbf{w}^h \right) + \tau_{\text{PSPG}} \boldsymbol{\nabla} q^h \right] \cdot \mathbf{r}_{\text{M}} \mathrm{d}Q \\
&+ \sum_{e=1}^{(N_{\text{el}})_n} \int_{Q_n^e} \rho \nu_{\text{LSIC}} \boldsymbol{\nabla} \cdot \mathbf{w}^h r_{\text{C}} \mathrm{d}Q = 0. \qquad (2.20)
\end{aligned}$$

Remark 4. One of the main differences between the ALE and DSD/SST forms of the VMS method is that the DSD/SST formulation retains the fine-scale time derivative term $\frac{\partial \mathbf{u}'}{\partial t}\big|_{\boldsymbol{\xi}}$. Dropping this term is called the "quasi-static" assumption (see [6] for the terminology). This is the same as the WTSE option in the DSD/SST formulation (see Remark 2 of [53]). We believe that this makes a significant difference, especially when the polynomial orders in space or time are higher (see [57]).

Remark 5. With the function spaces defined in the paragraph preceding Eq. (2.10), for each space–time slab velocity and pressure assume double unknown values at each spatial node. One value corresponds to the lower end of the slab, and the other one the upper end. In [53], the option of using double unknown values at a spatial node is called "DV" for velocity and "DP" for pressure. In this case, as pointed out in [53], we use two integration points over the time interval of the space–time slab, and this time-itegration option is called "TIP2". This version of the DSD/SST formulation, with the options set DV, DP and TIP2, is called "DSD/SST-DP".

2.4. *Weakly-enforced essential boundary conditions*

In this section we state the formulation of the weakly-enforced essential boundary conditions. This was first proposed in [58] for the advection–diffusion equation and Navier–Stokes equations of incompressible flows in an effort to improve the accuracy of stabilized and multiscale formulations in the presence of unresolved boundary layers. In [30, 31, 34], the method for the weakly-enforced boundary condition was further refined and studied in a set of challenging wall-bounded turbulent flows. Here, we apply the method to the aerodynamics of wind turbines at full spatial scale.

To account for the weak enforcement of the essential boundary conditions, we remove them from the trial and test function sets $\mathcal{S}_u^h$ and $\mathcal{V}_u^h$, and add the following terms to the

left-hand-side of Eq. (2.4):

$$
\begin{aligned}
&- \sum_{b=1}^{N_{\text{eb}}} \int_{\Gamma_t^b \bigcap (\Gamma_t)_{\text{g}}} \mathbf{w}^h \cdot \boldsymbol{\sigma}\left(\mathbf{u}^h, p^h\right) \mathbf{n} \, \mathrm{d}\Gamma \\
&- \sum_{b=1}^{N_{\text{eb}}} \int_{\Gamma_t^b \bigcap (\Gamma_t)_{\text{g}}} \left(2\mu\boldsymbol{\varepsilon}\left(\mathbf{w}^h\right)\mathbf{n} + q^h\mathbf{n}\right) \cdot \left(\mathbf{u}^h - \mathbf{g}^h\right) \, \mathrm{d}\Gamma \\
&- \sum_{b=1}^{N_{\text{eb}}} \int_{\Gamma_t^b \bigcap (\Gamma_t)_{\text{g}}^-} \mathbf{w}^h \cdot \rho\left(\left(\mathbf{u}^h - \hat{\mathbf{u}}^h\right) \cdot \mathbf{n}\right)\left(\mathbf{u}^h - \mathbf{g}^h\right) \, \mathrm{d}\Gamma \\
&+ \sum_{b=1}^{N_{\text{eb}}} \int_{\Gamma_t^b \bigcap (\Gamma_t)_{\text{g}}} \tau_{\text{TAN}}^B \left(\mathbf{w}^h - \left(\mathbf{w}^h \cdot \mathbf{n}\right)\mathbf{n}\right) \cdot \left(\left(\mathbf{u}^h - \mathbf{g}^h\right) - \left(\left(\mathbf{u}^h - \mathbf{g}^h\right) \cdot \mathbf{n}\right)\mathbf{n}\right) \, \mathrm{d}\Gamma \\
&+ \sum_{b=1}^{N_{\text{eb}}} \int_{\Gamma_t^b \bigcap (\Gamma_t)_{\text{g}}} \tau_{\text{NOR}}^B \left(\mathbf{w}^h \cdot \mathbf{n}\right)\left(\left(\mathbf{u}^h - \mathbf{g}^h\right) \cdot \mathbf{n}\right) \, \mathrm{d}\Gamma. \qquad (2.21)
\end{aligned}
$$

Here $(\Gamma_t)_{\text{g}}$ is the part of the boundary where the velocity boundary condition $\mathbf{g}$ is set and $\mathbf{n}$ is the unit normal vector. The boundary $(\Gamma_t)_{\text{g}}$ is decomposed into N_{eb} surface elements denoted by Γ_t^b, and $(\Gamma_t)_{\text{g}}^-$ is defined as the "inflow" part of $(\Gamma_t)_{\text{g}}$:

$$
(\Gamma_t)_{\text{g}}^- = \left\{ \mathbf{x} \;\middle|\; \left(\mathbf{u}^h - \hat{\mathbf{u}}^h\right) \cdot \mathbf{n} < 0, \ \forall \mathbf{x} \subset (\Gamma_t)_{\text{g}} \right\}. \qquad (2.22)
$$

If $(\Gamma_t)_{\text{g}}$ coincides with the moving wall (rigid or flexible), then $\mathbf{g}$ is the prescribed wall velocity.

The term in the first line is the so-called consistency term. It is necessary to ensure that the discrete formulation is identically satisfied by the exact solution of the Navier–Stokes equations, which, in turn, has implications on the accuracy of the discrete formulation. Also note that this term cancels with the contributions coming from the integration-by-parts of the stress terms in Eq. (2.4), thus correctly removing traction boundary conditions from the no-slip boundary. The term in the second line is the so-called adjoint consistency term. Its role is less intuitive, as it ensures that the analytical solution of the adjoint equations, when introduced in place of the linear momentum and continuity equation test functions, also satisfies the discrete formulation. Adjoint consistency is linked to optimal convergence of the discrete solution in lower-order norms (see, e.g., [60]). The term in the third line leads to better satisfaction of the inflow boundary conditions. The last two terms are penalty-like, in that they penalize the deviation of the discrete solution from its prescribed value at the boundary. These terms are necessary to ensure the stability (or coercivity) of the discrete formulation, which may be lost due to the introduction of the consistency and adjoint consistency terms.

The weak boundary condition formulation is numerically stable if

$$
\tau_{\text{TAN}}^B = \tau_{\text{NOR}}^B = \frac{C_I^B \mu}{h_n}, \qquad (2.23)
$$

where h_n is the wall-normal element size, and C_I^B is a sufficiently large positive constant computed from an appropriate element-level inverse estimate (see, e.g., [98–100]). The

constant C_I^B depends on the space dimension d, the element type (tetrahedron, hexahedron, etc.), and the polynomial order of the finite element approximation. For a linear tetrahedron, it is sufficient to take $4.0 \leq C_I^B \leq 8.0$ to obtain a stable discrete solution. The wall-normal element size may be computed from the element metric tensor:

$$h_n = (\mathbf{n} \cdot \mathbf{G}\mathbf{n})^{1/2}. \tag{2.24}$$

Remark 6. Rather than setting the no-slip boundary conditions exactly, the weak boundary condition formulation gives the no-slip solution only in the limit as $h_n \to 0$. As a result, coarse discretizations do not need to struggle to resolve the boundary layers; the flow simply slips on the solid boundary. Because of this added flexibility, the weak boundary condition enforcement tends to produce more accurate results on meshes that are too coarse to capture the boundary layer solution. However, as the mesh is refined to capture the boundary layer, the weak and strong boundary condition formulations produce nearly identical results (see [30]).

Remark 7. Although the weak boundary condition formulation is also stable for very large values of C_I^B, we do not favor that. Large values of C_I^B place a heavy penalization on the no-slip condition, and the above mentioned flexibility of the method is lost together with the associated accuracy benefits. We favor using a C_I^B that is just large enough to guarantee the stability of the discrete formulation.

Remark 8. In reference [30], a connection was identified between the weakly-enforced boundary conditions and wall functions. The latter are commonly employed in conjunction with RANS formulations of turbulent flows (see, e.g., [122, 123]). In the case of wall function formulation, a no-slip boundary condition is replaced with a tangential traction boundary condition, where the traction direction is given by that of the local slip velocity, and the traction magnitude is computed by invoking the "law-of-the-wall". This is an empirical relationship between the flow speed and the normal distance to the wall, both appropriately normalized (see, e.g., [122]). The penalty parameter τ_{TAN}^B may be defined as

$$\tau_{\text{TAN}}^B = \frac{\rho u^{*2}}{\|\mathbf{u}_{\text{TAN}}^h\|}, \tag{2.25}$$

where $\mathbf{u}_{\text{TAN}}^h = \left(\left(\mathbf{u}^h - \mathbf{g}^h\right) - \left(\left(\mathbf{u}^h - \mathbf{g}^h\right) \cdot \mathbf{n}\right)\mathbf{n}\right)$ is the tangential slip velocity, and u^* is the so-called friction velocity, which, among other factors, depends on the magnitude of the slip velocity, and is computed from the law-of-the-wall formula by nonlinear iterations. It was shown in [30], however, that when the boundary layer mesh is fine enough, τ_{TAN}^B from Eq. (2.25) is independent of the local flow solution, and reverts to the definition given by Eq. (2.23). This fact is remarkable in that Eq. (2.23) is purely based on considerations of numerical stability, while Eq. (2.25) derives from the physics of wall-bounded turbulent flows. In our limited experience, both the "numerics-based" and "physics-based" definitions of the penalty parameter τ_{TAN}^B give very similar results.

3. Aerodynamic Simulations of a 5MW Wind-Turbine Rotor

In this section we begin with a careful definition of the 5MW wind-turbine rotor geometry. We then present the NURBS-based and FEM-based simulations of the wind-turbine rotor. In this section, we only present pure aerodynamic simulations. Structural and FSI modeling and simulations will be presented in the later sections.

3.1. *5MW wind-turbine rotor geometry definition*

RNodes (m)	AeroTwst (°)	Chord (m)	AeroCent (-)	AeroOrig (-)	Airfoil
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.

As a first step we construct a template for the structural model of the rotor. Here, the structural model is limited to a surface (shell) representation of the wind-turbine blade, the hub, and their attachment zone. The blade surface is assumed to be composed of a collection of airfoil shapes that are lofted in the blade axis direction.

The geometry of the rotor blade is based on the NREL 5MW offshore baseline wind-turbine described in [2]. The blade geometry data taken from the reference is summarized in Table 1. A 61 m blade is attached to a hub with radius of 2 m, which gives the total rotor radius of 63 m. The blade is composed of several airfoil types provided in the rightmost column of the table. The first portion of the blade is a perfect cylinder. Further away from the root the cylinder is smoothly blended into a series of DU (Delft University) airfoils. At the 44.55 m location away from the root the NACA64 profile is used to define the blade all the way to the tip (see Figure 2). The remaining parameters from Table 1 are defined in Figure 2: “RNodes” is the distance from the rotor center to the airfoil cross-section in the blade axis direction. “AeroTwst” is the twist angle for a given cross-section. The blades

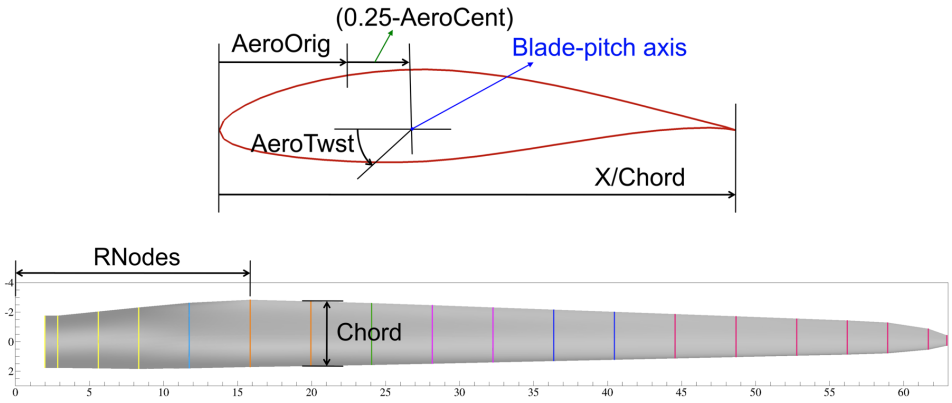


Fig. 2. Illustration of quantities from Table 1.

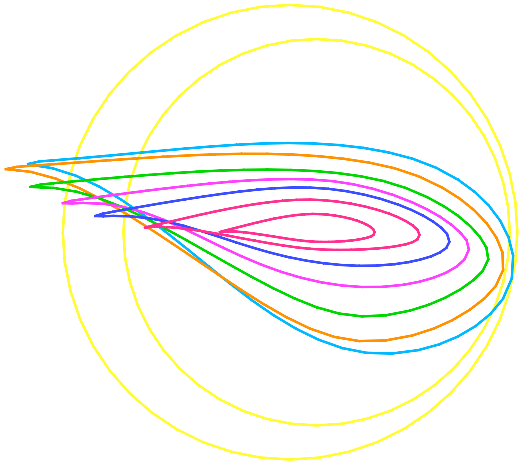


Fig. 3. Top view of a subset of the airfoil cross-sections illustrating blade twisting.

are twisted to enhance the aerodynamic performance. "Chord" is the chord length of the airfoil. "AeroOrig" is the location of the aerodynamic center. For most of the blade airfoil cross-sections, the aerodynamic center is taken at 25% of the chord length from the leading edge. To accommodate the cylindrical shape at the root, the aerodynamic center is gradually moved to 50% of the chord length. This is not reported in [2], but mentioned in [124].

Remark 9. There is some redundancy in the parameters given in Table 1. The variable "AeroCent" is used as an input to FAST [1], which is the aerodynamics modeling software that is typically used for wind-turbine rotor computations. FAST assumes that the blade-pitch axis passes through each airfoil section at 25% chord length, and defines AeroCent –

0.25 to be the fractional distance 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 of where the blade-pitch axis passes through each airfoil cross-section. Although for our purposes this added complexity is unnecessary, the same naming system is used for backward compatibility with the referenced reports.

For each blade 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 of the blade exactly. The cross-sections are lofted in the blade axis direction, also using quadratic NURBS and unity weights. This geometry modeling procedure produces a smooth rotor blade surface using a relatively small number of input parameters, which is an advantage of the isogeometric representation. Figure 3 shows a top view of the blade in which the twisting of the cross-sections is evident. Given the rotor blade surface description, the fluid-domain volume is constructed

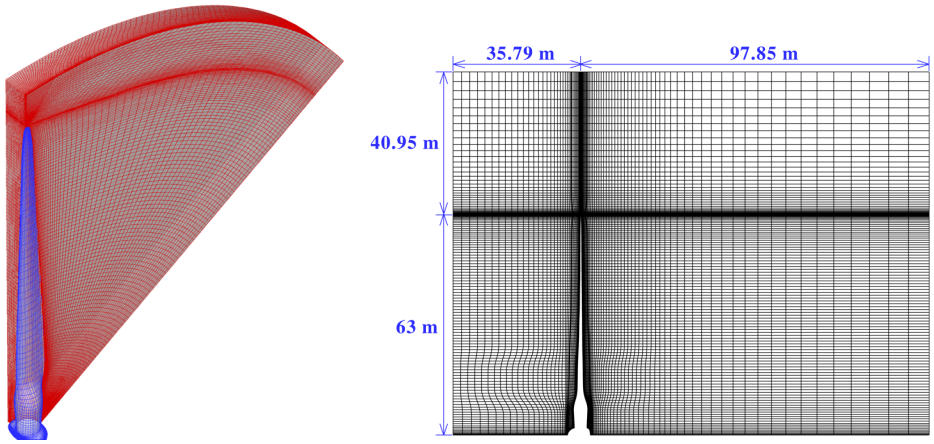


Fig. 4. (a) Volume NURBS mesh of the computational domain. (b) A planar cut to illustrate mesh grading toward the rotor blade.

next. The blade surface is split into four patches of similar sizes, which we call the blade surface patches. The splitting is done at the leading and trailing edges, as well as half-way in between on both sides of the blade. The fluid domain near the blade is generated for each one of the blade surface patches. As a final step, the fluid domain patches are merged such that the outer boundary of the domain is a perfect cylinder.

For each of the blade surface patches, we create a 60° pie-shaped domain using a minimum required number of control points. The control points at the bottom of the patch are moved to accommodate the shape of the rotor hub. As a next step, we perform knot in-

sertion and move the new control points such that their locations coincide with those of the blade surface patch. This generates an a-priori conforming discretization between the volume fluid domain and the surface of the structural model, suitable for FSI analysis. Finally, the fluid domain is refined in all parametric directions for analysis. Figure 4a shows the rotor surface mesh and one of the fluid-mesh subdomains adjacent to it. The remaining fluid subdomains are generated in the same manner.

The resultant fluid NURBS mesh may be embedded into a larger domain for the purposes of simulation. In this work we take this larger domain to also be a cylinder. For computational efficiency, only one-third of the domain is modeled. The fluid volume mesh, corresponding to one-third of the fluid domain, consists of 1,449,000 quadratic NURBS elements (and a similar number of control points). The fluid mesh cross-section that also shows the details of mesh refinement in the boundary layer is shown in Figure 4b. To carry

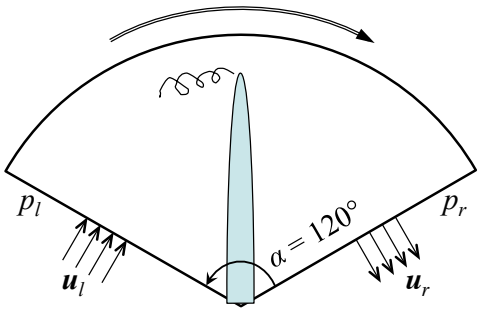


Fig. 5. Rotationally-periodic boundary conditions.

out the simulations, rotationally-periodic boundary conditions must be imposed. Denoting by $\mathbf{u}_l^h$ and $\mathbf{u}_r^h$ the discrete fluid velocities at the left and right boundary, respectively (see Figure 5), and by p_l^h and p_r^h the corresponding pressures, we set

$$p_l^h = p_r^h, \tag{3.1}$$

$$\mathbf{u}_l^h = \mathbf{R}(2/3\pi)\, \mathbf{u}_r^h, \tag{3.2}$$

where $\mathbf{R}(2/3\pi)$ is the rotation matrix evaluated at $\alpha = 2/3\pi$. That is, while the pressure degrees-of-freedom take on the same values, the fluid velocity degrees-of-freedom are related through a linear transformation corresponding to a rotation by $2/3\pi$ radians. Note that the transformation matrix is independent of the current domain position. Rotationally-periodic boundary conditions are implemented through standard master-slave relationships. We note that rotationally-periodic boundary conditions were employed earlier in [78, 80, 125] for parachute simulations.

We compute the aerodynamics of the wind-turbine rotor with prescribed speed using 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 [126] 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 radial boundary the radial component of the velocity is set to zero. We start from a flow field where the velocity is equal to the inflow velocity everywhere in the domain except on the rotor surface, where the velocity matches the rotor velocity.

We carry out the computations at a constant time-step size of 4.67×10^{-4} s. Both NURBS and tetrahedral FEM simulations of this setup are performed.

The chosen wind velocity and rotor speed correspond to one of the cases given in [2], where the aerodynamics simulations were performed using FAST [1]. We note that FAST is based on look-up tables for airfoil cross-sections, which give planar, steady-state lift and drag data for a given wind speed and angle of attack. The effects of trailing edge turbulence, hub, and tip are incorporated through empirical models. It was reported in [2] that at these wind conditions and rotor speed, no blade pitching takes place and the rotor develops a favorable aerodynamic torque (i.e., torque in the direction of the rotation) of 2500 kN m. Although this value is used for comparison with our simulations, the exact match is not expected, as our computational modeling is very different than the one in [2]. Nevertheless, we feel that this value of the aerodynamic torque is close to what is expected in reality, given the vast experience of NREL with wind-turbine rotor simulations employing FAST.

3.2. *ALE-VMS simulations using NURBS-based isogeometric analysis*

The computation is carried out on 240 cores on Ranger, a Sun Constellation Linux Cluster at the Texas Advanced Computing Center [127] with 62,976 processing cores. Near the blade surface, the size of the first element in the wall-normal direction is about 2 cm. The GMRES search technique [128] is used with a block-diagonal preconditioner. Each nodal block consists of a 3×3 and 1×1 matrices, corresponding to the discrete momentum and continuity equations, respectively. The number of nonlinear iterations per time step is 4 and the number of GMRES iterations is 200 for the first Newton iteration, 300 for the second, and 400 for the third and fourth. Figure 6 shows the air speed at $t = 0.8$ s at 1 m behind the rotor plane. Note the fine-grained turbulent features at the trailing edge of the blade, which require enhanced mesh resolution for accurate representation. The fluid traction vectors projected to the plane of rotation are shown in Figure 7. The traction vectors point in the direction of rotation and grow in magnitude toward the blade tip, creating favorable aerodynamic torque. However, at the blade tip the traction vectors rapidly decay to zero and even change sign, which introduces a small amount of inefficiency. The time history of the aerodynamic torque is shown in Figure 8, where the steady-state result from [2] is also shown for reference. The figure shows that in less than 0.8 s the torque settles at a statistically-stationary value of 2,670 kN m, which is within 6.4% of the reference. Given the significant differences in the computational modeling approaches, the two values are remarkably close. This result is encouraging in that 3D time-dependent simulation with a manageable number of degrees of freedom and without any empiricism is able to predict important quantities of interest for wind-turbine rotors simulated at full scale. This result also gives us confidence that our procedures are accurate and may be applied to simulations

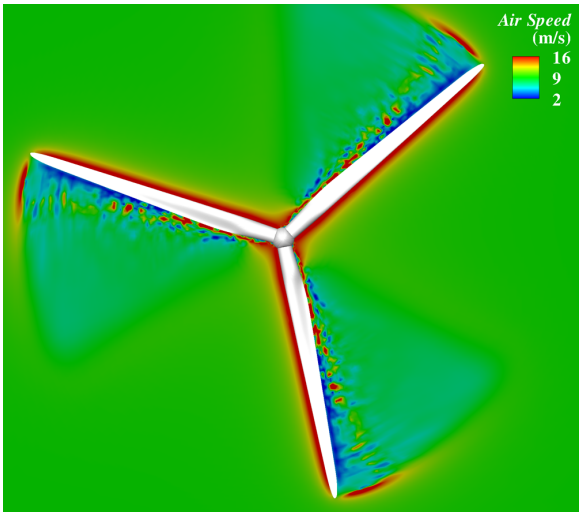


Fig. 6. Air speed at $t = 0.8$ s.

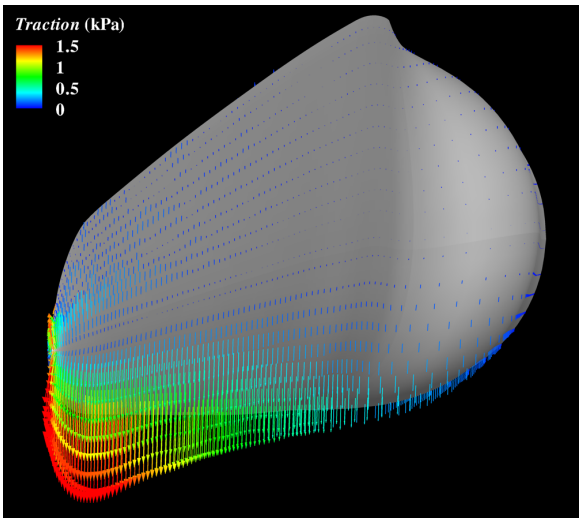


Fig. 7. Fluid traction vectors at $t = 0.8$ s viewed from the back of the blade. The traction vectors, colored by magnitude, are projected to the rotor plane and illustrate the mechanism by which the favorable aerodynamic torque is created.

cases where 3D, time-dependent modeling is indispensable (e.g., simulation of wind gusts or blade pitching).

Given the aerodynamic torque and the rotor speed, the power extracted from the wind

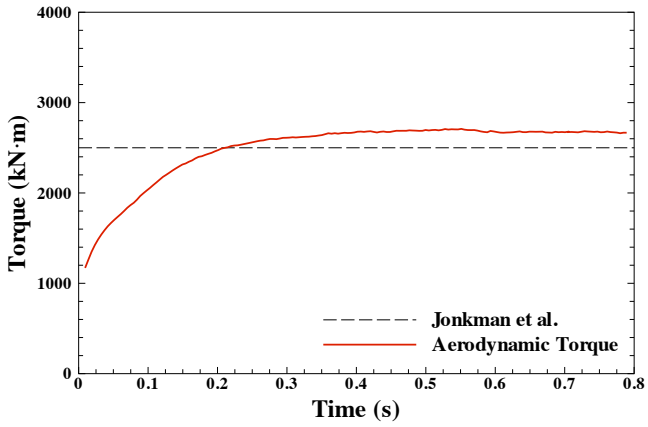


Fig. 8. Time history of the aerodynamic torque. Statistically-stationary torque is attained in less than 0.8 s. The reference steady-state result from NREL is also shown for comparison.

with these wind conditions (based on our simulations) is

$$P = T_f \dot{\theta} \approx 2.88\text{MW}. \tag{3.3}$$

According to the Betz law (see, e.g., [129]), the maximum power that a horizontal-axis wind turbine is able to extract from the wind is

$$P_{max} = \frac{16}{27} \frac{\rho A \|\mathbf{u}_{in}\|^3}{2} \approx 3.23\text{MW}, \tag{3.4}$$

where $A = \pi R^2$ is the cross-sectional area swept by the rotor, and $\|\mathbf{u}_{in}\|$ is the inflow speed. From this we conclude that the wind-turbine aerodynamic efficiency at the simulated wind conditions is

$$\frac{P}{P_{max}} \approx 89\%, \tag{3.5}$$

which is quite high even for modern wind-turbine designs. The blade is segmented into 18 spanwise "patches" to investigate how the aerodynamic torque distribution varies along the blade span. The patch-wise torque distribution is shown in Figure 9. The torque is nearly zero in the cylindrical section of the blade. A favorable aerodynamic torque is created on Patch 4 and its magnitude continues to increase until Patch 15. The torque magnitude decreases rapidly after Patch 15, however, the torque remains favorable all the way to the last patch.

The importance of 3D modeling and simulation is further illustrated in Figure 10, where the axial component of the flow velocity is displayed at a blade cross-section located at 56 m above the rotor center. The magnitude of the axial velocity component exceeds 15 m/s in the boundary layer, showing that 3D effects are important, especially in the regions of the blade with the largest contribution to the aerodynamic torque.

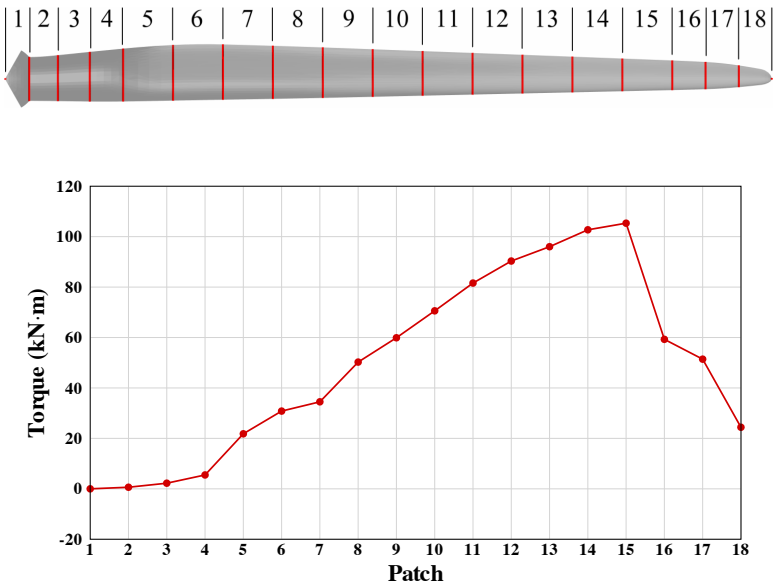


Fig. 9. Patches along the blade (top) and the aerodynamic torque contribution from each patch (bottom) at t = 0.8 s.

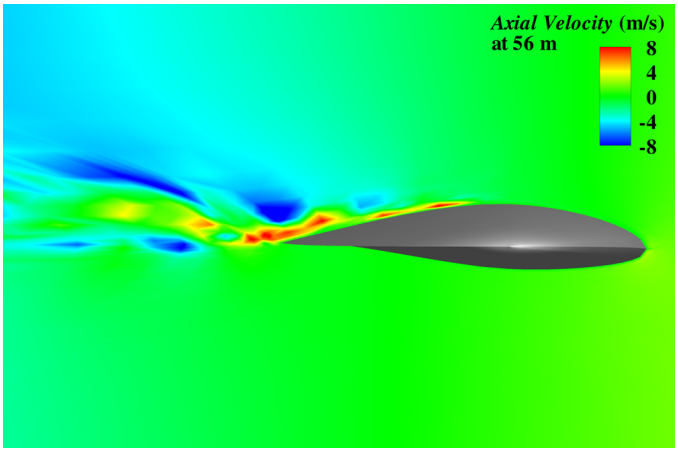


Fig. 10. Axial flow velocity over the blade cross-section at 56 m at t = 0.8 s. The level of axial flow in the boundary layer is significant, which illustrates the importance of 3D modeling.

3.3. *Computations with the DSD/SST formulation using finite elements*

We compute the same problem using the DSD/SST formulation with linear finite elements. To generate the triangular mesh on the rotor surface, we started with a quadrilateral surface

mesh generated by interpolating the NURBS geometry of the rotor at each knot intersection. We subdivided each quadrilateral element into triangles and then made minor modifications to improve the mesh quality near the hub. We use three different meshes: Mesh-2, Mesh-3 and Mesh-4, with the surface mesh refined along the blade 2, 3 and 4 times, respectively, compared to the finite element mesh used in [6]. The number of nodes and elements for each blade surface mesh is shown in Table 2, and Figure 11 shows the surface mesh for Mesh-4. For computational efficiency, rotational-periodicity [78, 80] is utilized so that

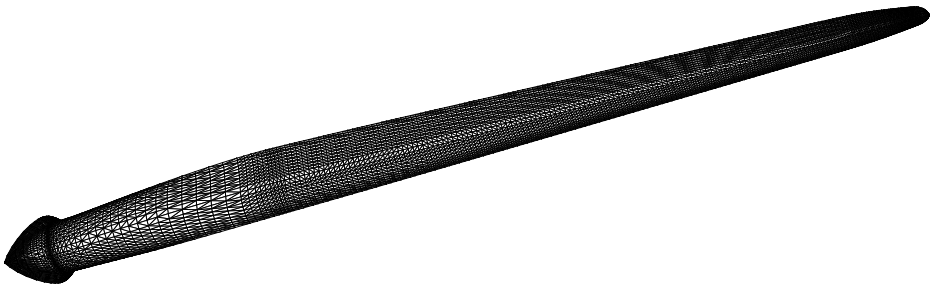


Fig. 11. Rotor surface mesh (Mesh-4).

the domain includes only one of three blades, as shown in Figure 12. The inflow, outflow

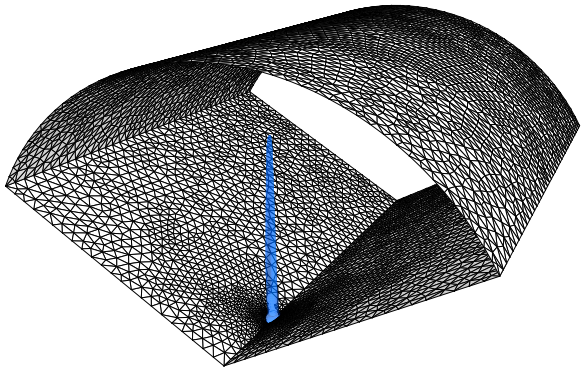


Fig. 12. Rotationally-periodic domain with wind-turbine blade shown in blue.

and radial boundaries lie $0.5R$, $2R$ and $1.43R$ from the hub center, respectively. This can be more easily seen in Figure 13, where the inflow, outflow, and radial boundaries are the left, right and top edges, respectively, of the cut plane along the rotation axis. 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 14. The number of nodes and elements

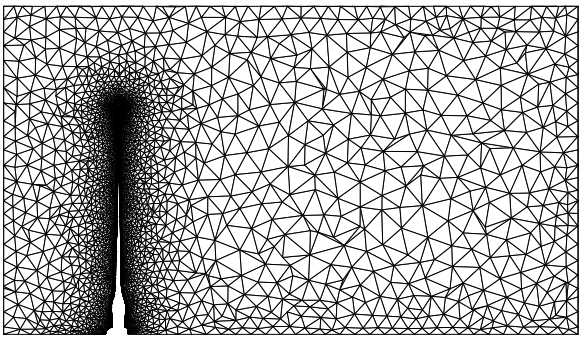


Fig. 13. Cut plane of the fluid volume mesh along rotor axis (Mesh-4).

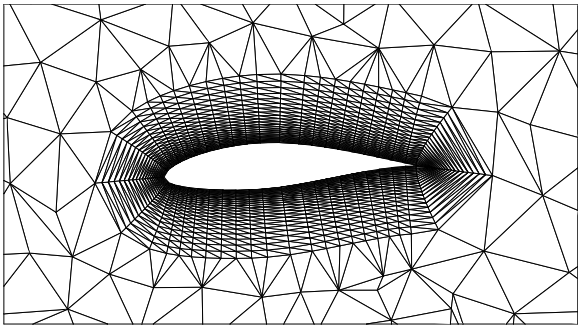


Fig. 14. Boundary layer mesh at $\frac{3}{4}R$.

	Surface		Volume	
	nn	*ne*	*nn*	*ne*
Mesh-2	5,748	11,452	155,494	898,640
Mesh-3	7,552	15,060	205,855	1,195,452
Mesh-4	9,268	18,492	253,340	1,475,175

Table 2. Summary of the meshes. Here *nn* and *ne* are the number of nodes and elements.

for each volume mesh is shown in Table 2.

We compute the problem with the DSD/SST-DP-SUPS and DSD/SST-DP-VMST techniques. The SUPS technique is used without the LSIC stabilization (i.e., $\nu_{\text{LSIC}} = 0$), while for the VMST technique ν_{LSIC} is defined according to Eq. (18) in Ref. [57], and is referred to as "TGI".

In solving the linear equation systems involved at every nonlinear iteration, the GMRES search technique [128] is used with a diagonal preconditioner. The computation is carried out in a parallel computing environment, using PC clusters. The mesh is partitioned

to enhance the parallel efficiency of the computations. Mesh partitioning is based on the METIS algorithm [130]. 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.

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.

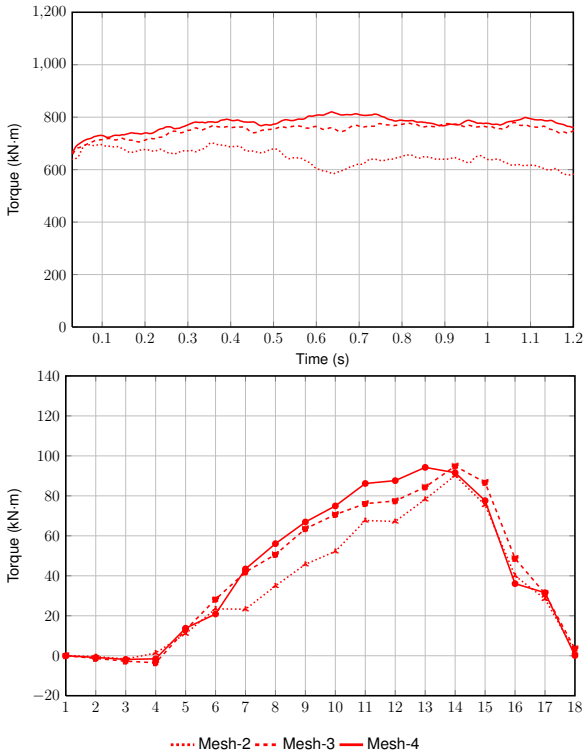


Fig. 15. The aerodynamic torque generated by a single blade. Comparison between different meshes with the DSD/SST-DP-SUPS technique. Time history (top). The torque contribution from each patch at $t = 1.0$ s (bottom).

Figures 15–17 show the time history of the aerodynamic torque and the torque contribution from each patch for a single blade at $t = 1.0$ s. The patches are defined as shown in Figure 9. Figure 18 shows the pressure coefficient at $t = 1.0$ s for Patch 16 (at $0.90R$), which is a representative section of the blade. 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

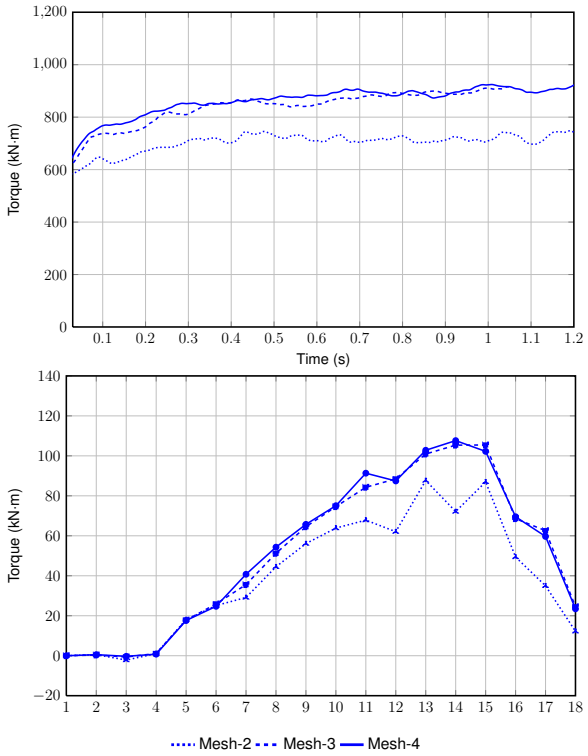


Fig. 16. The aerodynamic torque generated by a single blade. Comparison between different meshes with the DSD/SST-DP-VMST (TGI) technique. Time history (top). The torque contribution from each patch at $t = 1.0$ s (bottom).

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 $0.90R$).

Mesh refinement studies for both the SUPS and VMST techniques indicate good convergence in the quantities of interest such as the aerodynamic torque and pressure coefficient. The VMST results on the finest mesh give more or less the same value of the aerodynamic torque as the ALE-VMS simulation using NURBS, which is taken as a reference solution for this study. The results of the SUPS technique are also very good, however the torque is slightly under-predicted with respect to the VMST and NURBS-based ALE-VMS simulation. Figure 18 indicates that smoother (i.e., more stable) pressure solution is obtained with the VMST technique. 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 the figure. The lower pressure indicates that the flow is attached; thus, the VMST technique, for the level of mesh refinement presented here, is able to better represent the turbulent boundary layer solution than the SUPS technique.

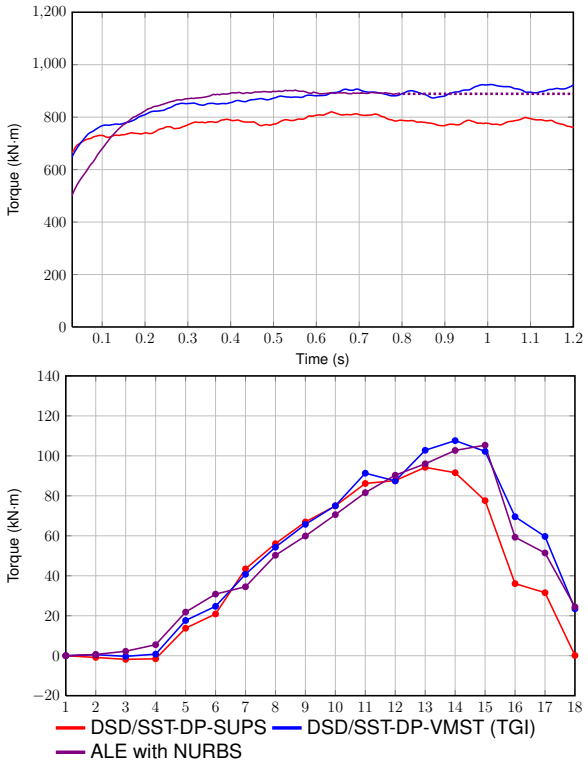


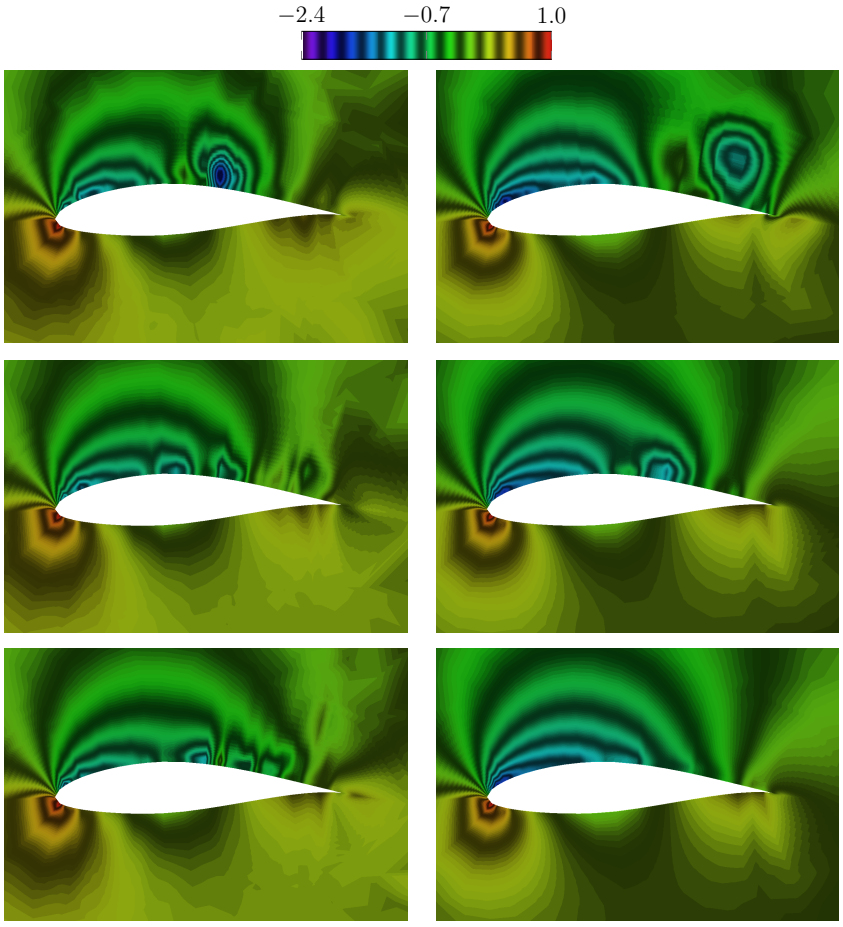
Fig. 17. The aerodynamic torque generated by a single blade. Computed with different techniques using Mesh-4. Time history (top). The torque contribution from each patch at $t = 1.0$ s (bottom). We note that the curve labeled "ALE with NURBS" represents the aerodynamic torque data from the NURBS computation presented in the previous section.

Radial Distance r (m)	Span Station (r/5.029 m)	Chord Length (m)	Twist (degrees)	Twist Axis (% chord)	Airfoil (-)
0.508	0.100	0.218	0.0	50	Cylinder
1.510	0.300	0.711	14.292	30	NREL S809
2.343	0.466	0.627	4.715	30	NREL S809
3.185	0.633	0.542	1.115	30	NREL S809
4.023	0.800	0.457	-0.381	30	NREL S809
4.780	0.950	0.381	-1.469	30	NREL S809
5.029	1.000	0.355	-1.815	30	NREL S809

Table 3. Selected cross-sectional data for NREL UAE Phase VI blade.

4. NREL Phase VI Wind-Turbine Rotor: Validation and the Role of Weakly-Enforced Essential Boundary Conditions

The proposed method is applied to predict the aerodynamics of the Unsteady Aerody-

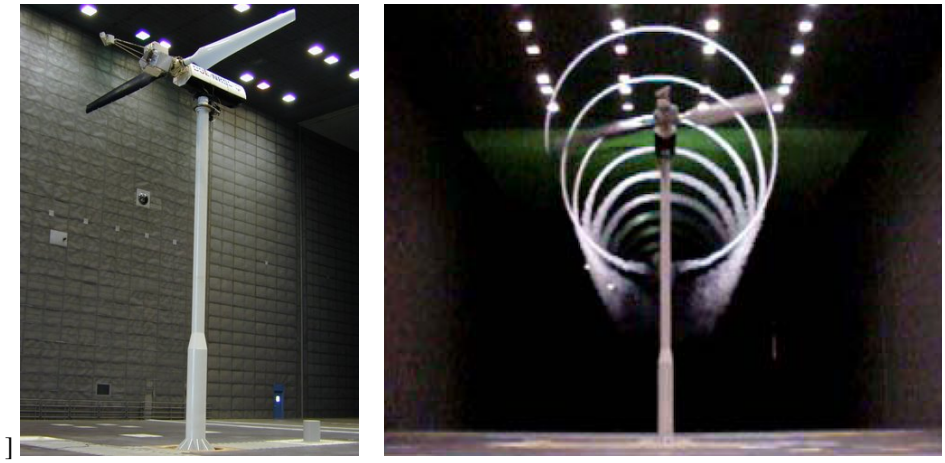


(a) DSD/SST-DP-SUPS (b) DSD/SST-DP-VMST (TGI)

Fig. 18. Pressure coefficient at t = 1.0 s for Patch 16 (at 0.90R). Top: Mesh-2. Middle: Mesh-3. Bottom: Mesh-4.

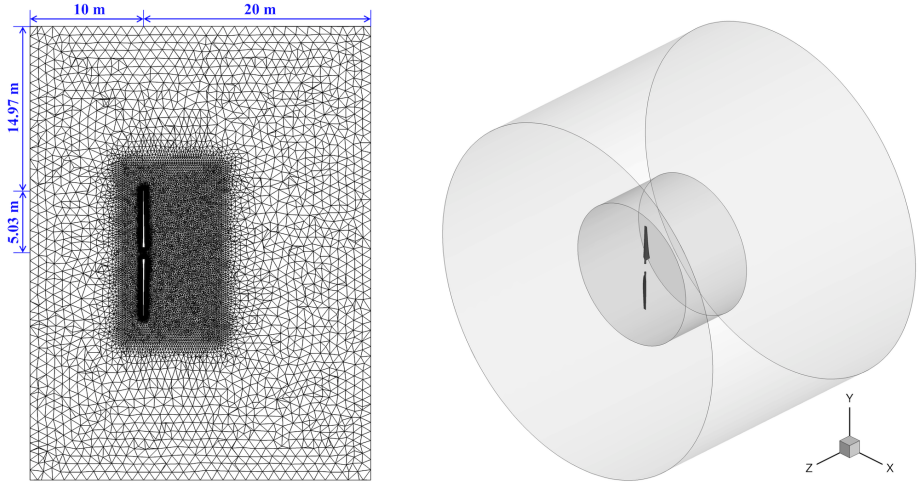
namics Experiment (UAE) Phase VI two-bladed wind-turbine rotor [97] from NREL. In this experiment, a two-bladed twisted and tapered 10.058 m diameter wind turbine, which has a rated power of 19.8 kW, was tested in the NASA Ames 80 ft × 120 ft wind tunnel in 2000 (see Figure 19). This is one of the most comprehensive, accurate, and reliable experiments carried out on a full-scale wind turbine. This test case was also studied by many computational researchers [3–5, 131–133] for the purposes of validating their simulations and improving their ability to predict wind-turbine aerodynamic performance.

The Phase VI rotor geometry makes use of a single NREL S809 airfoil [97]. Selected blade cross-section geometry data are summarized in Table 3. The detailed documentation of the rotor configuration and its technical specifications are available in [97]. Two cases from the experiment were selected for the validation study. The first case has a wind speed



(a) (b)

Fig. 19. (a) UAE mounted in 80 ft × 120 ft test section. (b) UAE wake flow visualization in the field at the National Wind Technology Center and in the wind tunnel at NASA Ames Research Center.



(a) (b)

Fig. 20. A 2D cut of the mesh (left) and the computational domain (right).

of 5 m/s and the second case 25 m/s. For both cases, we have upwind configuration, 0° yaw angle, 0° cone angle, blade tip pitch angle of 3°, and rotational speed of 72 rpm. The influence of the hub and tower on the rotor aerodynamics was neglected, which is a fair approximation for an upwind turbine (see, e.g., [3]). The two cases we consider here present very different flow conditions. For the 5 m/s case the flow is fully attached for the

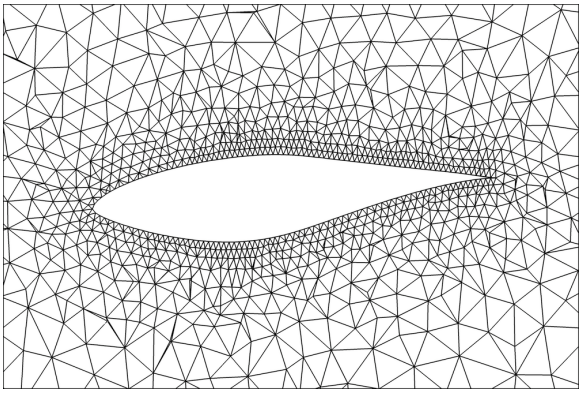


Fig. 21. A 2D blade cross-section at 0.8R illustrating the boundary layer mesh used in the computations, which is relatively coarse. The size of the first element in the wall-normal direction is about 0.008 m.

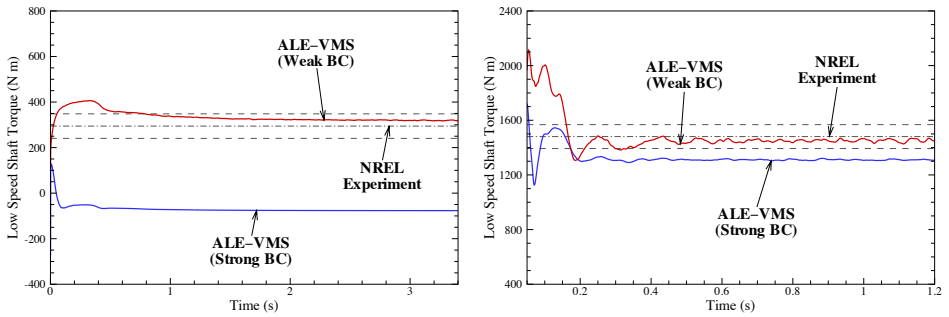
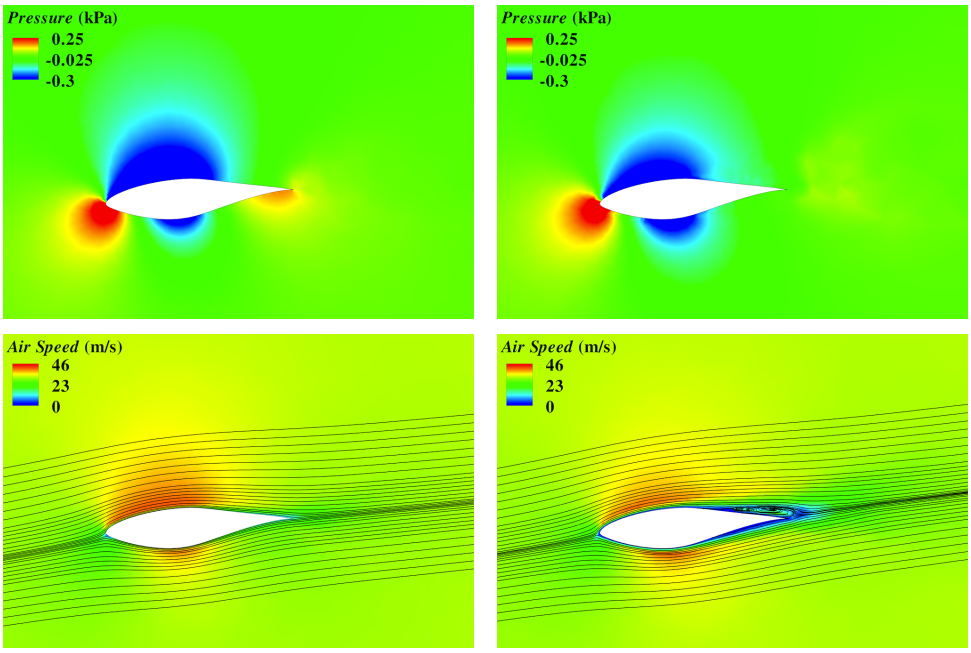


Fig. 22. The time history of the aerodynamic (low-speed shaft) torque for both weak and strong boundary condition simulations at (a) 5 m/s and (b) 25 m/s. The results are compared to the NREL experimental data. Dashed lines represent the standard deviation in the experimental data.

entire blade. On the contrary, the stall occurs in most part of the blade for the 25 m/s case and the simulation is considered to be more challenging [8].

The mesh resolution and computational domain are shown in Figure 20. The rotor radius R is 5.029 m and the blade is assumed to be rigid. At the inflow boundary the wind speed is set to either 5 m/s or 25 m/s. At the outflow boundary the traction vector is set to zero. At the radial boundary the radial component of the velocity is set to zero. The air density and viscosity are 1.23 kg/m^3 and 1.78×10^{-5} kg/(m·s), respectively. The mesh is comprised of 1,508,983 nodes and 8,494,182 linear tetrahedral elements. Figure 21 shows a 2D blade cross-section at 0.8R to illustrate the type of mesh near the boundary. Near the blade surface at 0.8R, the size of the first element in the wall-normal direction is about 0.008 m. No special boundary layer meshing was used in this study, in part to test the ability of the ALE-VMS method to deal with coarse boundary layer meshes.



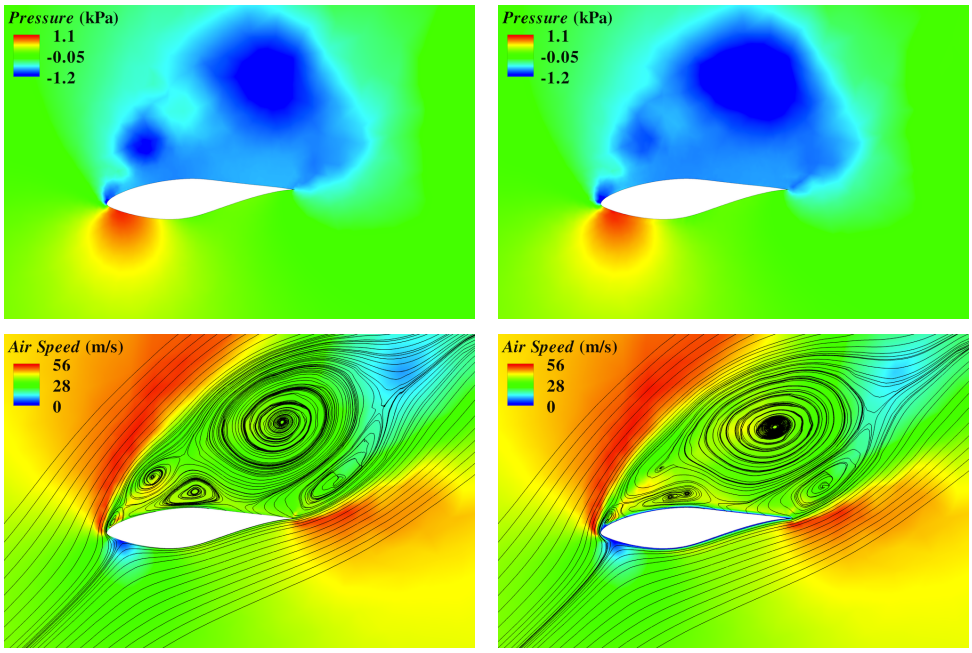
(a) (b)

Fig. 23. Pressure, air speed contours, and streamlines at 0.8R for the 5 m/s case. (a) Weakly-enforced boundary condition. (b) Strongly-enforced boundary condition.

The computations were carried out in a parallel computing environment on a Dell Cluster at the Texas Advanced Computing Center (TACC) [127].The linear system is solved using a block-diagonal preconditioned GMRES method [128, 134]. The time-step size is 0.0001 s. The number of nonlinear iterations per time step is 3, with 50 GMRES iterations for the first and second nonlinear iterations, and 50 to 80 GMRES iterations for the third nonlinear iteration, depending on the nonlinear convergence.

The time history of aerodynamic (low-speed shaft) torque is shown in Figure 22. Good agreement in the aerodynamic torque is found between the weak-boundary-condition simulations and experimental data for both flow conditions. However, the results for the strongly-enforced boundary condition are not at all accurate.

Pressure, air speed contours, and streamlines at 0.8R for the 5 m/s and 25 m/s cases are shown in Figures 23 and 24, respectively. Figure 23a shows the weak-boundary-condition prediction of the air flow for the 5 m/s case. The flow is fully attached, and the torque is correctly predicted. However, the strong-boundary-condition simulation predicts flow separation at the trailing edge (see Figure 23b). The blade stalls and, as a result, the torque is under-predicted by 126% (see Figure 22a). For the 25 m/s case, small differences are found in the pressure contours and air flow patterns between the weak- and strong-boundary-condition computations. This is due to the fact that the flow is already separated at the edges, the entire airfoil is stalled, and the boundary layer resolution is not so important



(a) (b)

Fig. 24. Pressure, air speed contours, and streamlines at 0.8R for the 25 m/s case. (a) Weakly-enforced boundary condition. (b) Strongly-enforced boundary condition.

for these type of flow conditions. In this case, the weak boundary condition again correctly predicts the torque, while the strong boundary condition under-predicts the torque, but only by 11% (see Figure 22b).

These results are not surprising. For strongly-enforced boundary conditions, the coarse boundary layer discretization gives rise to artificially "thick" boundary layers, which retard the flow and lead to non-physical aerodynamics, such as premature flow separation. In the case of weakly-enforced boundary conditions, the flow is allowed to slip on the solid surface without forming these undesired thick boundary layers. Of course, with sufficient boundary layer mesh refinement, both approaches will capture the boundary layer, and the strongly-enforced boundary condition will also produce the correct result (see [30]).

Figure 25 shows the pressure coefficient at 0.466R, 0.633R, and 0.8R for the 5 m/s and 25 m/s cases. The predicted values (using the weak boundary condition) are plotted against the experimental data. Very good agreement is likewise achieved for both attached and separated flow conditions at different radial locations.

Figure 26 shows the flow visualization (isosurfaces of air speed) for the 5 m/s case. The tip vortex generated by the blade is carried downstream of the rotor with little decay. The figure also shows the pressure contours on the rotor surface.

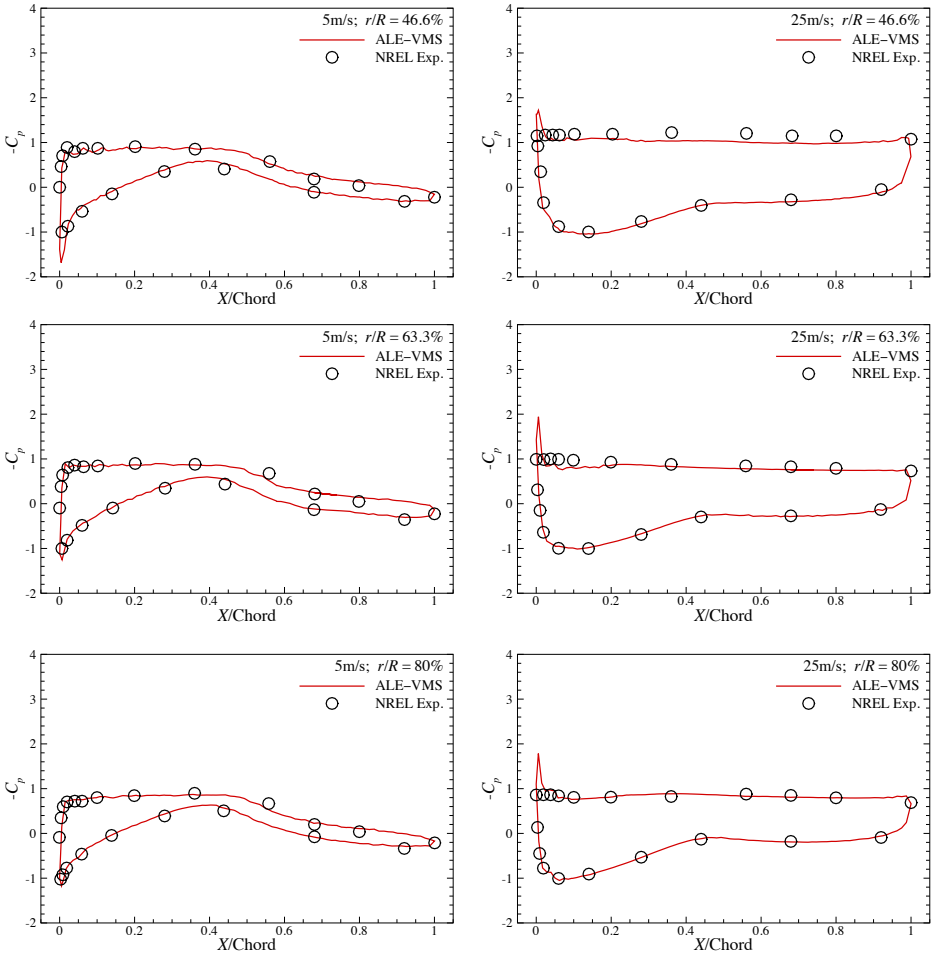


Fig. 25. Pressure coefficients at $0.466R$, $0.633R$, and $0.8R$ for (a) 5 m/s and (b) 25 m/s. The predicted values (using the weak boundary condition) are plotted against the NREL experimental data.

5. Governing Equations of Structural Mechanics: Isogeometric Kirchhoff–Love Composite Shell and the Bending-Strip Method

5.1. *Kirchhoff–Love shell*

In this section we follow the developments of [13, 15, 39] and present the governing equations of the Kirchhoff–Love shell theory. The theory is appropriate for thin-shell structures and requires no rotational degrees of freedom. The variational formulation of a Kirchhoff–Love shell is based on the principle of virtual work expressed as

$$\delta W = \delta W_{int} + \delta W_{ext} = 0\,, \tag{5.1}$$

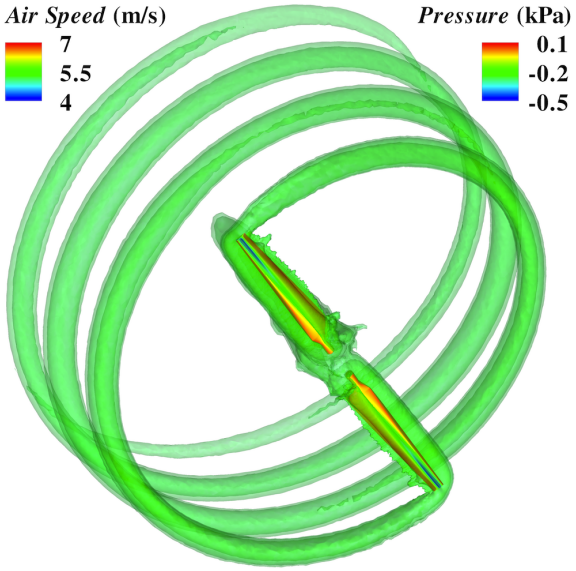


Fig. 26. Isosurfaces of air speed at an instant for the 5 m/s case. The tip vortex generated by the blade is carried downstream of the rotor with little decay. The figure also shows the pressure contours on the rotor surface.

where W, W_{int}, and W_{ext} denote the total, internal, and external work, respectively, and δ denotes a variation with respect to the virtual displacement variables $\delta\mathbf{d}$, that is

$$\delta W = \frac{\partial W}{\partial \mathbf{d}} \delta \mathbf{d} \,. \tag{5.2}$$

The internal virtual work is defined by (see, e.g., [135])

$$\delta W_{int} = - \int_{\Omega_0^s} (\delta \mathbf{E} \,:\, \mathbf{S}) \, \mathrm{d}\Omega \,, \tag{5.3}$$

where Ω_0^s is the shell volume in the reference configuration (the total Lagrangian approach is adopted in this work), $\mathbf{E}$ is the Green–Lagrange strain tensor, $\delta\mathbf{E}$ is its variation with respect to virtual displacements $\delta\mathbf{d}$, and $\mathbf{S}$ is the energetically conjugate second Piola–Kirchhoff stress tensor.

In the case of shells, the 3D continuum description is reduced to that of the shell midsurface, and the transverse normal stress is neglected. Furthermore, the Kirchhoff–Love theory assumes that the shell director remains normal to its middle surface during the deformation, which implies that the transverse shear strains are zero. As a result, only in-plane stress and strain tensors are considered, and the indices $\alpha = 1, 2$ and $\beta = 1, 2$ are employed to denote their components. We denote by Γ_0^s the shell midsurface in the undeformed reference configuration, h_{th} is the (variable) shell thickness, and $\xi_3 \in [-0.5h_{th}, 0.5h_{th}]$ is the through-thickness coordinate.

We introduce the following standard shell kinematic quantities and relationships

(see [39, 136] for more details):

$$E_{\alpha\beta} = \varepsilon_{\alpha\beta} + \xi_3 \kappa_{\alpha\beta}, \tag{5.4}$$

$$\varepsilon_{\alpha\beta} = \frac{1}{2}\left(\mathbf{g}_\alpha \cdot \mathbf{g}_\beta - \mathbf{G}_\alpha \cdot \mathbf{G}_\beta\right), \tag{5.5}$$

$$\kappa_{\alpha\beta} = -\frac{\partial \mathbf{g}_\alpha}{\partial \xi_\beta} \cdot \mathbf{g}_3 - (-\frac{\partial \mathbf{G}_\alpha}{\partial \xi_\beta} \cdot \mathbf{G}_3), \tag{5.6}$$

$$\mathbf{g}_\alpha = \frac{\partial \mathbf{x}}{\partial \xi_\alpha}, \tag{5.7}$$

$$\mathbf{G}_\alpha = \frac{\partial \mathbf{X}}{\partial \xi_\alpha}, \tag{5.8}$$

$$\mathbf{g}_3 = \frac{\mathbf{g}_1 \times \mathbf{g}_2}{\|\mathbf{g}_1 \times \mathbf{g}_2\|}, \tag{5.9}$$

$$\mathbf{G}_3 = \frac{\mathbf{G}_1 \times \mathbf{G}_2}{\|\mathbf{G}_1 \times \mathbf{G}_2\|}, \tag{5.10}$$

$$\mathbf{G}^\alpha = (\mathbf{G}_\alpha \cdot \mathbf{G}_\beta)^{-1} \mathbf{G}_\beta. \tag{5.11}$$

Here, $E_{\alpha\beta}$, $\varepsilon_{\alpha\beta}$, and $\kappa_{\alpha\beta}$ are the contravariant components of the in-plane Green–Lagrange strain, membrane strain, and curvature tensors, respectively. The spatial coordinates of the *shell midsurface* in the current and reference configurations are $\mathbf{x} = \mathbf{x}(\xi_1, \xi_2)$ and $\mathbf{X} = \mathbf{X}(\xi_1, \xi_2)$, parameterized by ξ_1 and ξ_2. The covariant surface basis vectors in the current and reference configurations are $\mathbf{g}_\alpha$ and $\mathbf{G}_\alpha$. The unit outward normal vectors to the shell midsurface in the current and reference configurations are $\mathbf{g}_3$ and $\mathbf{G}_3$. The contravariant surface basis vectors in the reference configuration are denoted by $\mathbf{G}^\alpha$.

We select the *local Cartesian basis* vectors as follows:

$$\bar{\mathbf{e}}_1 = \frac{\mathbf{G}_1}{\|\mathbf{G}_1\|}, \tag{5.12}$$

$$\bar{\mathbf{e}}_2 = \frac{\mathbf{G}_2 - (\mathbf{G}_2 \cdot \mathbf{e}_1)\mathbf{e}_1}{\|\mathbf{G}_2 - (\mathbf{G}_2 \cdot \mathbf{e}_1)\mathbf{e}_1\|}, \tag{5.13}$$

that is, the first local basis vector is the normalized first covariant basis vector in the reference configuration. The local Cartesian basis vectors $\bar{\mathbf{e}}_\alpha$ are used in expressing a constitutive relationship for the shell. Because the local basis is orthonormal, we make no distinction between covariant and contravariant quantities, which are expressed with respect to it.

With the above definitions, we calculate the components of the Green–Lagrange strain tensor and its variation in the local coordinate system as

$$\overline{E}_{\alpha\beta} = \overline{\varepsilon}_{\alpha\beta} + \xi_3 \overline{\kappa}_{\alpha\beta}, \tag{5.14}$$

$$\delta\overline{E}_{\alpha\beta} = \delta\overline{\varepsilon}_{\alpha\beta} + \xi_3 \delta\overline{\kappa}_{\alpha\beta}, \tag{5.15}$$

$$\overline{\varepsilon}_{\alpha\beta} = \varepsilon_{\gamma\delta}(\mathbf{G}^\gamma \cdot \bar{\mathbf{e}}_\alpha)(\mathbf{G}^\delta \cdot \bar{\mathbf{e}}_\beta), \tag{5.16}$$

$$\overline{\kappa}_{\alpha\beta} = \kappa_{\gamma\delta}(\mathbf{G}^\gamma \cdot \bar{\mathbf{e}}_\alpha)(\mathbf{G}^\delta \cdot \bar{\mathbf{e}}_\beta), \tag{5.17}$$

$$\delta\overline{\varepsilon}_{\alpha\beta} = \delta\varepsilon_{\gamma\delta}(\mathbf{G}^\gamma \cdot \bar{\mathbf{e}}_\alpha)(\mathbf{G}^\delta \cdot \bar{\mathbf{e}}_\beta), \tag{5.18}$$

$$\delta\overline{\kappa}_{\alpha\beta} = \delta\kappa_{\gamma\delta}(\mathbf{G}^\gamma \cdot \bar{\mathbf{e}}_\alpha)(\mathbf{G}^\delta \cdot \bar{\mathbf{e}}_\beta). \tag{5.19}$$

The variations $\delta\varepsilon_{\gamma\delta}$ and $\delta\kappa_{\gamma\delta}$ may be computed directly by taking the variational derivatives of the expressions given by Eqs. (5.5) and (5.6) with respect to the displacement vector.

We define the vectors of membrane strain and curvature components in the local coordinate system as

$$\overline{\boldsymbol{\varepsilon}} = \begin{bmatrix} \overline{\varepsilon}_{11} \\ \overline{\varepsilon}_{22} \\ \overline{\varepsilon}_{12} \end{bmatrix} \tag{5.20}$$

and

$$\overline{\boldsymbol{\kappa}} = \begin{bmatrix} \overline{\kappa}_{11} \\ \overline{\kappa}_{22} \\ \overline{\kappa}_{12} \end{bmatrix}, \tag{5.21}$$

together with a Green–Lagrange strain vector

$$\overline{\mathbf{E}} = \overline{\boldsymbol{\varepsilon}} + \xi_3 \overline{\boldsymbol{\kappa}}. \tag{5.22}$$

We assume St. Venant–Kirchhoff material law and write the following stress–strain relationship in the local coordinate system:

$$\overline{\mathbf{S}} = \overline{\mathbf{C}}\,\overline{\mathbf{E}}, \tag{5.23}$$

where $\overline{\mathbf{S}}$ is a vector of components of the second Piola–Kirchhoff stress tensor in the local coordinate system, and $\overline{\mathbf{C}}$ is a constitutive material matrix, which is symmetric. Introducing Eqs. (5.22) and (5.23) into the expression for the internal virtual work given by Eq. (5.3), we obtain

$$\delta W_{int} = -\int_{\Omega_0^s} \delta\overline{\mathbf{E}} \cdot \overline{\mathbf{S}}\, \mathrm{d}\Omega \tag{5.24}$$

$$= -\int_{\Gamma_0^s} \left(\int_{h_{th}} \delta\overline{\mathbf{E}} \cdot \overline{\mathbf{C}}\,\overline{\mathbf{E}}\, d\xi_3 \right) \mathrm{d}\Gamma \tag{5.25}$$

$$= -\int_{\Gamma_0^s} \delta\overline{\boldsymbol{\varepsilon}} \cdot \left(\left(\int_{h_{th}} \overline{\mathbf{C}}\, d\xi_3 \right) \overline{\boldsymbol{\varepsilon}} + \left(\int_{h_{th}} \xi_3 \overline{\mathbf{C}}\, d\xi_3 \right) \overline{\boldsymbol{\kappa}} \right) \mathrm{d}\Gamma$$

$$- \int_{\Gamma_0^s} \delta\overline{\boldsymbol{\kappa}} \cdot \left(\left(\int_{h_{th}} \xi_3 \overline{\mathbf{C}}\, d\xi_3 \right) \overline{\boldsymbol{\varepsilon}} + \left(\int_{h_{th}} {\xi_3}^2 \overline{\mathbf{C}}\, d\xi_3 \right) \overline{\boldsymbol{\kappa}} \right) \mathrm{d}\Gamma. \tag{5.26}$$

For a general orthotropic material,

$$\overline{\mathbf{C}}_{ort} = \begin{bmatrix} \dfrac{E_1}{(1-\nu_{12}\nu_{21})} & \dfrac{\nu_{21}E_1}{(1-\nu_{12}\nu_{21})} & 0 \\ \dfrac{\nu_{12}E_2}{(1-\nu_{12}\nu_{21})} & \dfrac{E_2}{(1-\nu_{12}\nu_{21})} & 0 \\ 0 & 0 & G_{12} \end{bmatrix}. \tag{5.27}$$

In Eq. (5.27), E_1 and E_2 are the Young's moduli in the directions defined by the local basis vectors, ν_{12} and ν_{21} are the Poisson ratios, G_{12} is the shear modulus, and $\nu_{21}E_1 = \nu_{12}E_2$

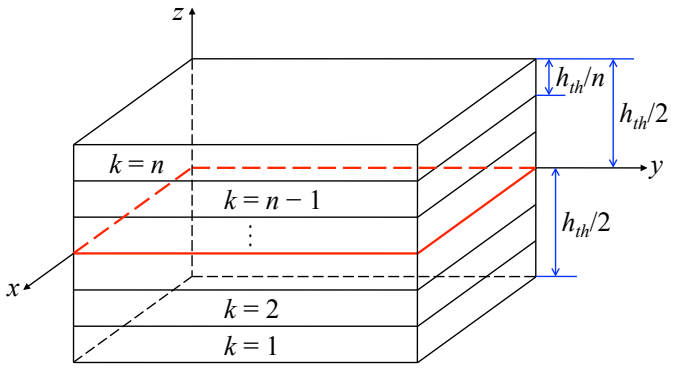


Fig. 27. Schematic of a composite laminate.

to ensure the symmetry of the constitutive material matrix $\overline{\mathbf{C}}_{ort}$. In the case of an isotropic material, $E_1 = E_2 = E$, $\nu_{21} = \nu_{12} = \nu$, and $G_{12} = E/(2(1+\nu))$.

In the case of composite materials, which are used in the manufacturing of modern wind-turbine blades, we assume that the structure is composed of a set of plies, each modeled as an orthotropic material. We use the classical laminated-plate theory [137], and homogenize the material through-thickness constitutive behavior for a given composite ply layout. Let k denotes the k^{th} ply (or lamina) and let n be the total number of plies (see Figure 27). We assume each ply has the same thickness h_{th}/n. Pre-integrating through the shell thickness in Eq. (5.26), the extensional stiffness **A**, coupling stiffness **B**, and bending stiffness **D** are given by

$$\mathbf{A} = \int_{h_{th}} \overline{\mathbf{C}}\, d\xi_3 = \frac{h_{th}}{n} \sum_{k=1}^{n} \overline{\mathbf{C}}_k, \tag{5.28}$$

$$\mathbf{B} = \int_{h_{th}} \xi_3 \overline{\mathbf{C}}\, d\xi_3 = \frac{h_{th}^2}{n^2} \sum_{k=1}^{n} \overline{\mathbf{C}}_k \left(k - \frac{n}{2} - \frac{1}{2}\right), \tag{5.29}$$

$$\mathbf{D} = \int_{h_{th}} {\xi_3}^2 \overline{\mathbf{C}}\, d\xi_3 = \frac{h_{th}^3}{n^3} \sum_{k=1}^{n} \overline{\mathbf{C}}_k \left(\left(k - \frac{n}{2} - \frac{1}{2}\right)^2 + \frac{1}{12}\right), \tag{5.30}$$

where

$$\overline{\mathbf{C}}_k = \mathbf{T}^T(\phi_k)\, \overline{\mathbf{C}}_{ort}\, \mathbf{T}(\phi_k), \tag{5.31}$$

$$\mathbf{T}(\phi) = \begin{bmatrix} \cos^2\phi & \sin^2\phi & \sin\phi\cos\phi \\ \sin^2\phi & \cos^2\phi & -\sin\phi\cos\phi \\ -2\sin\phi\cos\phi & 2\sin\phi\cos\phi & \cos^2\phi - \sin^2\phi \end{bmatrix}. \tag{5.32}$$

In the above equations, ϕ is the fiber orientation angle in each ply, Eq. (5.31) transforms $\overline{\mathbf{C}}_{ort}$ from the principal material coordinates to the laminate coordinates (defined by the local Cartesian basis) for each ply, and $\overline{\mathbf{C}}_k$ is constant within each ply.

With the above definitions, the expression for the internal virtual work for a composite shell may now be compactly written as

$$\delta W_{int} = -\int_{\Gamma_0^s} \delta\overline{\boldsymbol{\varepsilon}} \cdot \left(\mathbf{A}\overline{\boldsymbol{\varepsilon}} + \mathbf{B}\overline{\boldsymbol{\kappa}}\right) \mathrm{d}\Gamma - \int_{\Gamma_0^s} \delta\overline{\boldsymbol{\kappa}} \cdot \left(\mathbf{B}\overline{\boldsymbol{\varepsilon}} + \mathbf{D}\overline{\boldsymbol{\kappa}}\right) \mathrm{d}\Gamma. \tag{5.33}$$

Remark 10. Setting $n = 1$ and $\overline{\mathbf{C}}_k = \overline{\mathbf{C}}_{ort}$ in Eqs. (5.28)–(5.30), we get $\mathbf{B} = \mathbf{0}$ and

$$\mathbf{A} = h_{th}\overline{\mathbf{C}}_{ort}, \tag{5.34}$$

$$\mathbf{D} = \frac{h_{th}^3}{12}\overline{\mathbf{C}}_{ort}, \tag{5.35}$$

which are the classical membrane and bending stiffnesses for an orthotropic shell.

5.2. *The bending-strip method and a complete variational statement of the structural mechanics problem*

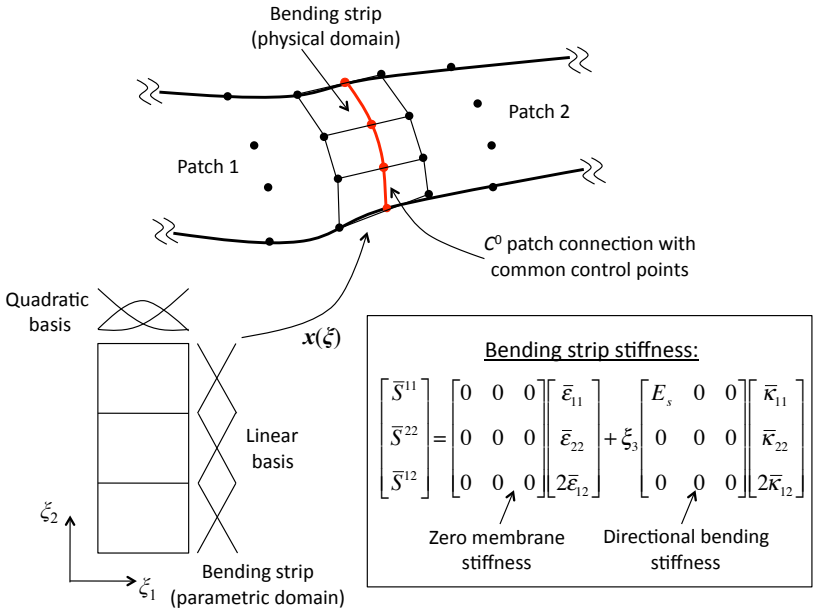


Fig. 28. Schematic of the bending-strip method.

The expression for of the internal virtual work given by Eq. (5.33) is only meaningful when the shell midsurface is described using a smooth geometrical mapping. In the case when the regularity of the mapping reduces to the C^0 level, the terms involving the curvature tensors, which rely on the second derivatives of the geometrical mapping, lead to

non-integrable singularities, and the formulation may not be used as is. However, for complex structures, the geometry definition often requires that the continuity of the geometrical mapping is reduced to the C^0 level (think of a trailing edge of an airfoil or an I-beam, the latter being a non-manifold surface).

In [13], a method was prosed to handle complex multi-patch shell structures in the context of the rotation-free Kirchhoff–Love theory, called the "bending-strip method" [c]. The main idea behind the method, illustrated in Figure 28, consists of the following. It is assumed that the shell structure is comprised of smooth subdomains, such as NURBS patches, that are joined with C^0-continuity. In addition, thin strips of fictitious material, also modeled as surface NURBS patches, are placed at patch intersections. The triples of control points at the patch interface, consisting of a shared control point and one on each side, are extracted and used as a control mesh for the bending strips. The parametric domain of each bending strip consists of one quadratic element in the direction transverse to the interface and, for simplicity and computational efficiency, of as many linear elements as necessary to accommodate all the control points along the length of the strip. The material is assumed to have zero mass, zero membrane stiffness, and *non-zero bending stiffness only in the direction transverse to the interface*. The transverse direction may be obtained using the local basis construction given by Eqs. (5.12) and (5.13), however, other options may be explored.

Let Γ_0^s and Γ_t^s denote the structure midsurface in the reference and deformed configurations, respectively, and let Γ_0^b denote the bending-strip domain, where Γ_0^b is a union of the bending-strip patch subdomains. Let $\mathcal{S}_d^h$ and $\mathcal{V}_d^h$ denote the discrete trial and test function spaces for the structural problem. We seek the displacement of the shell midsurface $\mathbf{d}^h \in \mathcal{S}_d^h$, such that $\forall\, \delta\mathbf{d}^h \in \mathcal{V}_d^h$:

$$\begin{aligned}
&\int_{\Gamma_t^s} \delta\mathbf{d}^h \cdot \rho h_{th} \left(\left.\frac{\partial^2 \mathbf{d}^h}{\partial t^2}\right|_{\mathbf{X}} - \mathbf{f}^h \right) \mathrm{d}\Gamma \\
&+ \int_{\Gamma_0^s} \delta\bar{\boldsymbol{\varepsilon}}^h \cdot \left(\mathbf{A}\bar{\boldsymbol{\varepsilon}}^h + \mathbf{B}\bar{\boldsymbol{\kappa}}^h \right) \mathrm{d}\Gamma \\
&+ \int_{\Gamma_0^s} \delta\bar{\boldsymbol{\kappa}}^h \cdot \left(\mathbf{B}\bar{\boldsymbol{\varepsilon}}^h + \mathbf{D}\bar{\boldsymbol{\kappa}}^h \right) \mathrm{d}\Gamma \\
&+ \int_{\Gamma_0^b} \delta\bar{\boldsymbol{\kappa}}^h \cdot \mathbf{D}^b \bar{\boldsymbol{\kappa}}^h \, \mathrm{d}\Gamma - \int_{(\Gamma_t^s)_h} \delta\mathbf{d}^h \cdot \mathbf{h}^h \, \mathrm{d}\Gamma = 0.
\end{aligned} \tag{5.36}$$

In the above, the superscript h denotes all the discrete quantities, ρ is the structural mass density in the deformed configuration, $\mathbf{f}$ is the body force (e.g. gravity), and $\mathbf{h}$ is the prescribed surface traction on $(\Gamma_t^s)_h$. The terms on the first line of Eq. (5.36) represent the inertial and body forces. The terms on the second and third lines were derived in the previous section. The term $\int_{\Gamma_0^b} \delta\bar{\boldsymbol{\kappa}}^h \cdot \mathbf{D}^b \bar{\boldsymbol{\kappa}}^h \, \mathrm{d}\Gamma$ is penalty-like, and represents the contribution of

[c]The method in its current form was developed and implemented at the University of California, San Diego, when J. Kiendl, at the time a PhD student in the group of K.-U. Bletzinger at the Technical University of Munich, was visiting the research group of Y. Bazilevs. The method has similarities with the concept of "continuity patches", introduced by K.-U. Bletzinger and collaborators in [138].

the bending strips to the structural formulation given by Eq. (5.36). Here $\mathbf{D}^b$ is the bending stiffness of the strips:

$$\mathbf{D}^b = \frac{h_{th}^3}{12}\mathbf{C}^b, \tag{5.37}$$

where

$$\mathbf{C}^b = \begin{bmatrix} E_s & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}, \tag{5.38}$$

and E_s is the scalar bending-strip stiffness, typically chosen as a multiple of the local Young's modulus of the shell. This design of the material constitutive matrix ensures that the bending strips add no extra stiffness to the structure. They only penalize the change in the angle during the deformation between the triples of control points at the patch interface. The stiffness E_s must be high enough so that the change in angle is within an acceptable tolerance. However, if E_s is chosen too high, the global stiffness matrix becomes badly conditioned, which may lead to divergence in the computations.

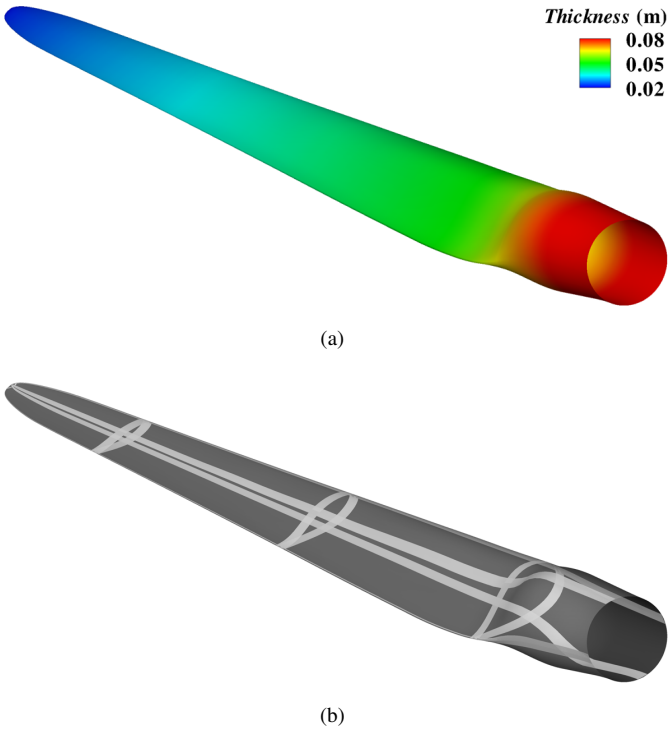


Fig. 29. NREL 5MW offshore baseline wind-turbine rotor blade model. (a) Blade thickness. (b) Bending strips.

The following illustrates the use of the bending-strip method applied to the construction of the structural model of the wind-turbine blade. A symmetric fiberglass/epoxy composite with $[\pm 45/0/90_2/0_3]_s$ lay-up, which enhances flap-wise and edge-wise stiffness, is considered for the rotor blade material. The 0° fiber points in the direction of a tangent vector to the airfoil cross-section curve. The orthotropic elastic moduli for each ply are given in Table 4. For simplicity, the entire blade is assumed to have the same lay-up. The resulting **A**, **B**, and **D** matrices from Eqs. (5.28)–(5.30) are

$$\mathbf{A} = h_{th} \begin{bmatrix} 26.315 & 4.221 & 0 \\ 4.221 & 18.581 & 0 \\ 0 & 0 & 5.571 \end{bmatrix} \times 10^9 \,(\mathrm{N/m})\,, \tag{5.39}$$

$$\mathbf{B} = \mathbf{0}, \tag{5.40}$$

$$\mathbf{D} = h_{th}^3 \begin{bmatrix} 1.727 & 0.545 & 0.053 \\ 0.545 & 1.627 & 0.053 \\ 0.053 & 0.053 & 0.658 \end{bmatrix} \times 10^9 \,(\mathrm{N{\cdot}m})\,. \tag{5.41}$$

The total laminate thickness distribution is shown in Figure 29a. The blade shell model together with the bending strips covering the regions of C^0-continuity is shown in Figure 29b. The FSI simulations with this blade model will be presented in the sequel. Further

E_1 (GPa)	E_2 (GPa)	G_{12} (GPa)	ν_{12}	ρ (g/cm^3)
39	8.6	3.8	0.28	2.1

Table 4. Material properties of a unidirectional E-glass/epoxy composite.

examples of calculations using the bending-strip method, which include verification and validation, and the possibility of using the approach for simple coupling of solids and shells, may be found in [13].

Remark 11. Because of the structure of the bending-strip term in Eq. (5.36), the method may be interpreted as a physically-motivated penalty formulation.

Remark 12. In IGA, the possibility to employ smooth surface descriptions directly in analysis has led to the development of new shell element formulations. Besides the references cited in this section, the reader is referred to [38, 45–47, 139] for relevant work on shells. We would also like to note that references [45–47] predate the development of IGA.

5.3. *Time integration of the structural mechanics equations*

In the case of wind-turbine rotors, the structural motions are dominated by the rotation of the blades around the hub axis. In [15], the authors proposed to take advantage of this fact and modify a class of standard time integration techniques to exactly account for the rotational part of the structural motion.

For this, as a first step, it is useful to decompose the structural displacement $\mathbf{d}$ into its rotation and deflection components as

$$\mathbf{d} = \mathbf{d}_\theta + \mathbf{d}_d. \tag{5.42}$$

The rotational component of the displacement may be computed as

$$\mathbf{d}_\theta = (\mathbf{R}(\theta) - \mathbf{I})\,(\mathbf{X} - \mathbf{X}_0)\,, \tag{5.43}$$

where $\mathbf{X}$ are the coordinates of the structure reference configuration, $\mathbf{X}_0$ is a fixed point, θ is the time-dependent angle of rotation, $\mathbf{R}(\theta)$ is the rotation matrix, and $\mathbf{I}$ is the identity matrix. We specialize to the case of rotation about the z-axis, which gives

$$\mathbf{R}(\theta) = \begin{bmatrix} \cos\theta & -\sin\theta & 0 \\ \sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{bmatrix}. \tag{5.44}$$

The total structural velocity and acceleration may be computed as

$$\frac{\partial \mathbf{d}}{\partial t} = \dot{\mathbf{d}} = \dot{\mathbf{d}}_\theta + \dot{\mathbf{d}}_d = \dot{\mathbf{R}}(\theta)(\mathbf{X} - \mathbf{X}_0) + \dot{\mathbf{d}}_d, \tag{5.45}$$

$$\frac{\partial^2 \mathbf{d}}{\partial t^2} = \ddot{\mathbf{d}} = \ddot{\mathbf{d}}_\theta + \ddot{\mathbf{d}}_d = \ddot{\mathbf{R}}(\theta)(\mathbf{X} - \mathbf{X}_0) + \ddot{\mathbf{d}}_d, \tag{5.46}$$

where

$$\dot{\mathbf{R}}(\theta) = \begin{bmatrix} -\sin\theta & -\cos\theta & 0 \\ \cos\theta & -\sin\theta & 0 \\ 0 & 0 & 0 \end{bmatrix} \dot{\theta}, \tag{5.47}$$

$$\ddot{\mathbf{R}}(\theta) = \begin{bmatrix} -\cos\theta & \sin\theta & 0 \\ -\sin\theta & -\cos\theta & 0 \\ 0 & 0 & 0 \end{bmatrix} \dot{\theta}^2 + \begin{bmatrix} -\sin\theta & -\cos\theta & 0 \\ \cos\theta & -\sin\theta & 0 \\ 0 & 0 & 0 \end{bmatrix} \ddot{\theta}. \tag{5.48}$$

We repeat this decomposition at the discrete level, where we operate directly on the nodal or control-point displacement degrees of freedom. For this, we let $\mathbf{U}$, $\dot{\mathbf{U}}$, and $\ddot{\mathbf{U}}$ be the vectors of nodal or control point displacements, velocities, and accelerations, respectively. We set

$$\mathbf{U} = \mathbf{U}_\theta + \mathbf{U}_d, \tag{5.49}$$

$$\dot{\mathbf{U}} = \dot{\mathbf{U}}_\theta + \dot{\mathbf{U}}_d, \tag{5.50}$$

$$\ddot{\mathbf{U}} = \ddot{\mathbf{U}}_\theta + \ddot{\mathbf{U}}_d, \tag{5.51}$$

where $\mathbf{U}_\theta$, $\dot{\mathbf{U}}_\theta$, and $\ddot{\mathbf{U}}_\theta$ are given by

$$\mathbf{U}_\theta = (\mathbf{R}(\theta) - \mathbf{I})\,(\mathbf{X} - \mathbf{X}_0), \tag{5.52}$$

$$\dot{\mathbf{U}}_\theta = \dot{\mathbf{R}}(\theta)(\mathbf{X} - \mathbf{X}_0), \tag{5.53}$$

$$\ddot{\mathbf{U}}_\theta = \ddot{\mathbf{R}}(\theta)(\mathbf{X} - \mathbf{X}_0). \tag{5.54}$$

The above Eqs. (5.52)–(5.54) present an exact relationship between the nodal or control point displacements, velocities, and accelerations corresponding to the rotational motion.

To relate the deflection degrees of freedom between time levels t_n and t_{n+1}, we make use of the standard Newmark formulas (see e.g., [140]):

$$\dot{\mathbf{U}}_d^{n+1} = \dot{\mathbf{U}}_d^n + \Delta t\left((1-\gamma)\ddot{\mathbf{U}}_d^n + \gamma\ddot{\mathbf{U}}_d^{n+1}\right), \tag{5.55}$$

$$\mathbf{U}_d^{n+1} = \mathbf{U}_d^n + \Delta t\dot{\mathbf{U}}_d^n + \frac{\Delta t^2}{2}\left((1-2\beta)\ddot{\mathbf{U}}_d^n + 2\beta\ddot{\mathbf{U}}_d^{n+1}\right), \tag{5.56}$$

where γ and β are the time integration parameters chosen to maintain second-order accuracy and unconditional stability of the method.

Combining exact rotations given by Eqs. (5.52)–(5.54) and time-discrete deflections given by Eqs. (5.55)–(5.56), we obtain the following modified Newmark formulas for the total discrete solution:

$$\begin{aligned}\dot{\mathbf{U}}^{n+1} = {} & \left\{\dot{\mathbf{R}}^{n+1} - \left[\dot{\mathbf{R}}^n + \Delta t\left((1-\gamma)\ddot{\mathbf{R}}^n + \gamma\ddot{\mathbf{R}}^{n+1}\right)\right]\right\}(\mathbf{X}-\mathbf{X}_0) \\ & + \dot{\mathbf{U}}^n + \Delta t\left((1-\gamma)\ddot{\mathbf{U}}^n + \gamma\ddot{\mathbf{U}}^{n+1}\right),\end{aligned} \tag{5.57}$$

$$\begin{aligned}\mathbf{U}^{n+1} = {} & \left\{\mathbf{R}^{n+1} - \left[\mathbf{R}^n + \Delta t\dot{\mathbf{R}}^n + \frac{\Delta t^2}{2}\left((1-2\beta)\ddot{\mathbf{R}}^n + 2\beta\ddot{\mathbf{R}}^{n+1}\right)\right]\right\}(\mathbf{X}-\mathbf{X}_0) \\ & + \mathbf{U}^n + \Delta t\dot{\mathbf{U}}^n + \frac{\Delta t^2}{2}\left((1-2\beta)\ddot{\mathbf{U}}^n + 2\beta\ddot{\mathbf{U}}^{n+1}\right).\end{aligned} \tag{5.58}$$

We employ Eqs. (5.57)–(5.58), in conjunction with the generalized-alpha method of [141], for the time discretization of the structure.

Remark 13. In the case of no rotation, for which $\mathbf{R}$ is an identity, Eqs. (5.57)–(5.58) reduce to the standard Newmark formulas. In the case of no deflection, pure rotation is likewise recovered.

6. FSI Coupling and Aerodynamics Mesh Update

In this section we briefly summarize our FSI coupling procedures for wind-turbine simulations. The fluid and structural equations are integrated in time using the generalized-alpha method. In the case of the structure, the modified Newmark formulas given by Eqs. (5.57)–(5.58) are employed to enhance the accuracy of the time integration procedures in the presence of large rotation. Within each time step, the coupled equations are solved using an inexact Newton approach. For every Newton iteration the following steps are performed. 1. We obtain the fluid solution increment holding the structure and mesh fixed. 2. We update the fluid solution, compute the aerodynamic force on the structure and compute the structural solution increment. The aerodynamic force at control points or nodes is computed using the conservative definition (see, e.g., [142, 143] for the importance of using the conservative definitions of fluxes near essential boundaries and in coupled problems). 3. We update the structural solution and use elastic mesh motion to update the fluid domain velocity and position. We note that only the deflection part of the mesh motion is computed using linear elastostatics, while the rotation part is computed exactly. This three-step iteration is repeated until convergence to an appropriately coupled discrete solution is achieved.

The proposed approach, also referred to as "block-iterative coupling" (see [53, 68, 69] for the terminology), is stable because the wind-turbine blades are relatively heavy structures.

Remark 14. For this coupling strategy the fluid and structural meshes may or may not be conforming. In the case of conforming meshes, the conservative nodal or control-point traction vector from the fluid side is applied directly at the nodes or control points of the structure, while the structural nodal or control-point kinematic data is applied directly to the nodes or control points of the fluid. When the fluid and structural meshes are non-conforming, additional projection of the traction and kinematic data is necessary before they are transferred to the neighboring subdomain. In this paper we only present conforming mesh simulations.

We conclude this section with the discussion of a special technique we devised in [15] to update the kinematics (position and velocity) of the fluid mesh. Typically, one employes the equations of linear elastostatics subject to dynamic boundary conditions coming from the structural displacement to update the position and velocity of the fluid mesh (see, for example, the elasticity-based mesh-moving method introduced in [144, 145]). In the case of wind turbines, which are dominated by rotation, this may not be a preferred procedure due to the fact that the linear elastostatics operator does not vanish on large rotational motions. This, in turn, may lead to the loss of the fluid mesh quality if one plans to simulate the FSI problem for many revolutions of the wind-turbine rotor. As a result, for the present application, we modify our fluid mesh motion strategy as follows. We take advantage of the fact that the structural displacement vector is already decomposed into the rotation and deflection parts. As a result, as the increment of the structural displacement is computed, we extract the deflection part, apply the elasticity-based mesh-moving method [144, 145] to computation of just the deflection part of the mesh displacement, rotate the (deformed) mesh from the previous time level to the current time, and add the mesh deflection increment to obtain its current position. For a precise mathematical formulation of this procedure see [15].

Remark 15. For a variety of other mesh update strategies the reader is referred to [94, 146, 147].

7. FSI Simulations of a 5MW Wind-Turbine Rotor

The wind-turbine rotor is simulated at prescribed steady inlet wind velocity of 11.4 m/s and rotor angular velocity of 12.1 rpm. This setup corresponds to one of the cases reported in [2]. The dimensions of the problem domain and the NURBS mesh employed are the same as in Section 3. The time-step size is 0.0003 s. The structural model of the wind-turbine blade is described in Section 5. A rigid rotor under the same wind and rotor speed conditions is simulated for comparison.

Rotationally-periodic boundary conditions for the fluid are imposed in order to reduce the computational cost (see Figure 5). However, because the rotor blades are subject to gravity forces, a fully rotationally-periodic structural solution is not expected in this case. Nevertheless, we feel that the use of rotationally-periodic boundary conditions for the fluid

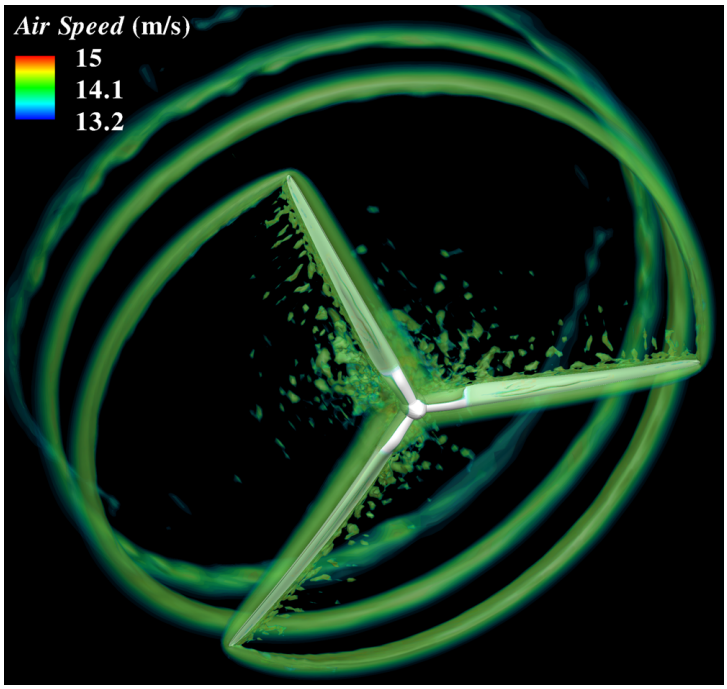


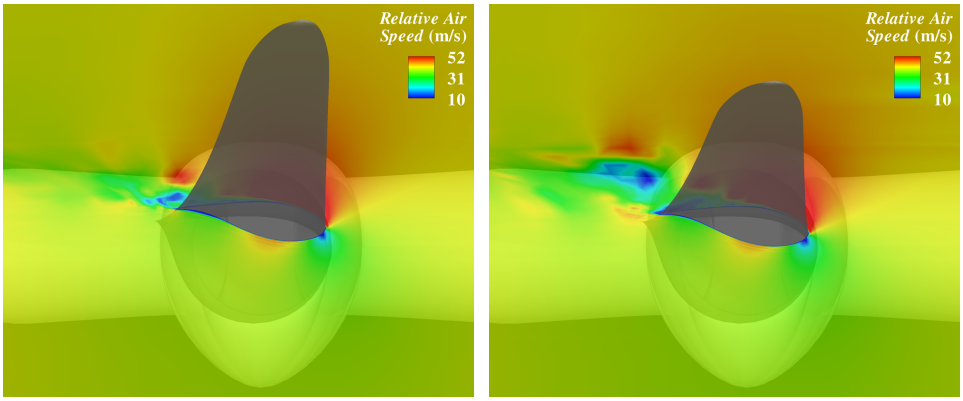
Fig. 30. Isosurfaces of air speed at an instant during the simulation, showing that the flow exhibits complex behavior. The tip vortex is advected downstream of the rotor with very little decay.

domain is justified due to the fact that the fluid periodic boundaries are located sufficiently far away from the structure and are not expected to affect the structural response.

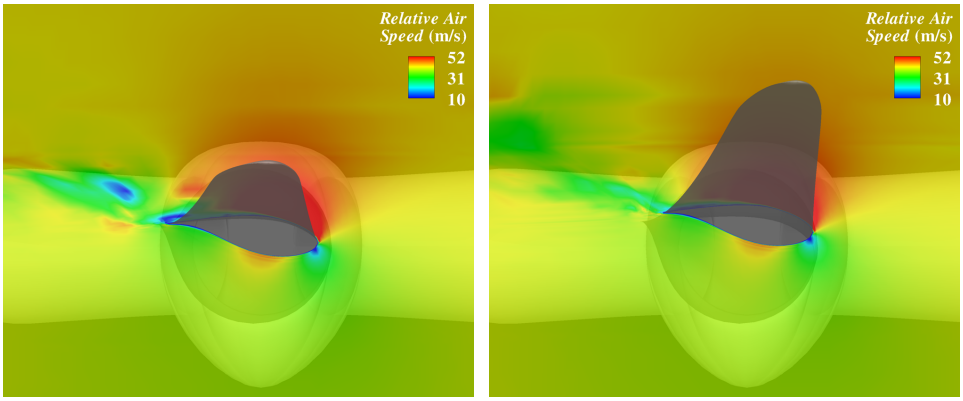
Isosurfaces of air speed at an instant during the simulation are shown in Figure 30. Note that, for visualization purposes the rotationally-periodic 120° domain was merged into a full 360° domain. Fine-grained turbulent structures are generated at the trailing edge of the blade along its entire length. The vortex forming at the tip of the blades is advected downstream of the rotor with little decay. There is also high-intensity turbulence in the blade aerodynamic zone, which is a segment of the blade where the cylindrical root rapidly transitions to a thin airfoil shape. This suggests that the blade trailing edge in this location is subjected to high-frequency loads that are fatiguing the blade.

Figure 31 shows the isocontours of relative wind speed at a 30 m radial cut at different instants during the simulation. For every snapshot the blade is rotated to the reference configuration to better illustrate the deflection part of the motion. The blade deflection is quite significant. On the pressure side, the air flow boundary layer is attached to the blade for the entire cord length. On the suction side, the flow detaches near the trailing edge and transitions to turbulence.

The aerodynamic torque (for a single blade) is plotted in Figure 32 for both rigid- and flexible-blade simulations. Both cases compare favorably to the data reported in [2]



(a) $t = 0.7$ s (b) $t = 1.2$ s



(c) $t = 2.0$ s (d) $t = 5.0$ s

Fig. 31. Isocontours of relative wind speed at a 30 m radial cut at different instants superposed on the moving blade. The air flow is fully attached on the pressure side of the blade and separates on the suction side. The flow separation point varies as the blade moves under the action of the wind, inertial, and gravitational forces.

for this setup obtained using FAST [1]. Note that the aerodynamic torque for the flexible blade exhibits low-magnitude, high-frequency oscillations, while the rigid-blade torque is smooth. To better understand this behavior, we examine the twisting motion of the wind-turbine blade about its axis. Time histories of the twist angle at four different cross-sections are shown in Figure 33. The twist angle increases with distance from the root and reaches almost 2° near the tip in the early stages of the simulation. However, starting at $t = 1.2$ s, when the blade tip reaches its lowest vertical position, the magnitude of the twist angle is reduced significantly. The reversal of the gravity vector with respect to the lift direction clearly affects the edge-wise bending and twisting behavior of the blade. The blade twist angle undergoes high frequency oscillations, which are in part driven by the trailing-edge vortex shedding and turbulence. Local oscillations in the twist angle lead to the temporal

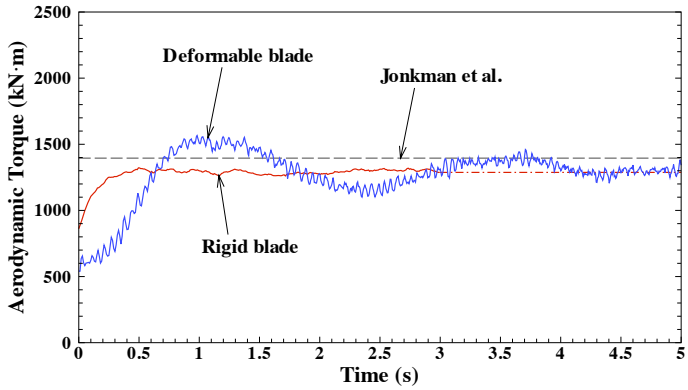


Fig. 32. Time history of the aerodynamic torque. Both rigid- and flexible-rotor results are plotted. The reference steady-state result from NREL is also shown for comparison.

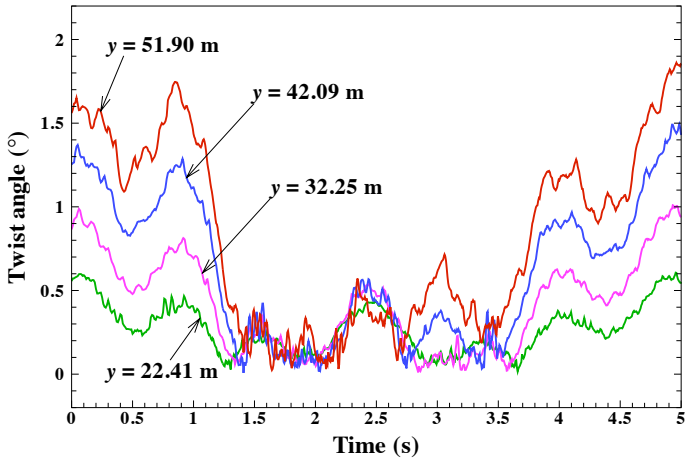


Fig. 33. Time histories of the twist angle at four cross-sections along the blade axis.

fluctuations in the aerodynamic torque.

8. Pre-Bending of the Wind-Turbine Blades

The rotor blades of a wind turbine need to be designed such that they do not strike the tower as the rotor turns in strong winds. This may be accomplished with blade pre-bending. In this case, the blades are manufactured to flex toward the wind when the rotor is mounted on the tower. Once the blades are exposed to the wind, and the rotor starts turning, the blades are straightened to achieve their design shape. This situation is graphically illustrated in

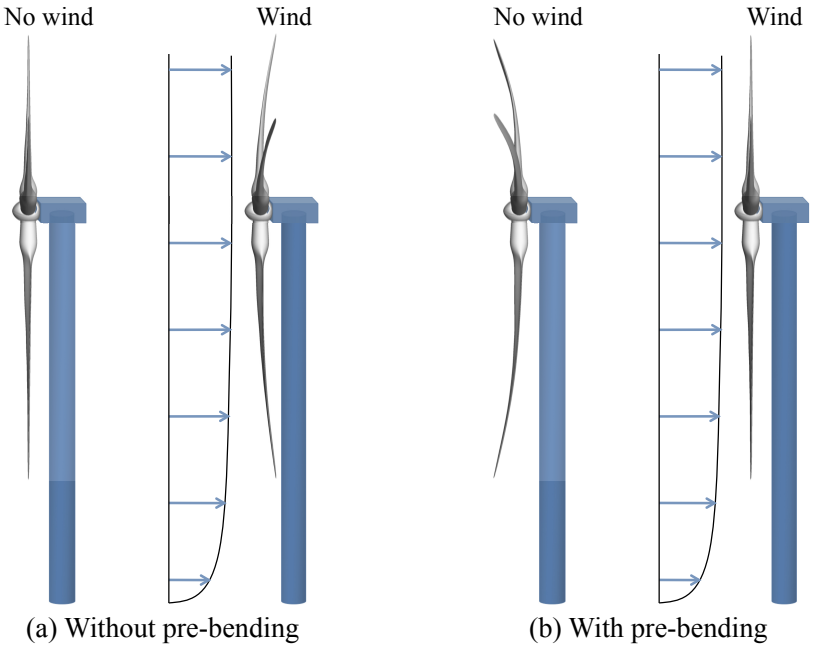


Fig. 34. Using pre-bent blades to ensure tower clearance and rotor operation in its design configuration.

Figure 34. Besides tower clearance, pre-bending of the blades engenders additional benefits. For example, the blades need not be quite as rigid because the amount of allowable deflection is greater. This makes it possible to use less material overall, and fewer processed materials, resulting in lighter and more economical blades. Pre-bending of the blades also results in a more compact nacelle design. During operation, the pre-bent blades straighten to their designed configuration, which is typically optimized for best possible aerodynamic performance.

Given the above advantages, it is important that one is able to determine the correct pre-bent shape given the blade structural and aerodynamic design, and the wind-turbine operating conditions (i.e., wind and rotor speeds). In [14], we proposed a method that makes use of standalone computational fluid and structural mechanics procedures to obtain a pre-bent shape of the wind-turbine blades. The main idea consists of performing an aerodynamics simulation of a rigidly-rotating rotor to obtain the aerodynamic load acting on the blade. Given the aerodynamic and inertial loads acting in the design configuration, a stress-free pre-bent blade configuration is found using a simple iterative procedure that requires a sequence of structural mechanics simulations. Note that in the proposed approach the aerodynamic and structural computations are decoupled, which avoids the challenges involved in solving the coupled FSI problem. In this section, we summarize the method and show the supporting computations. We follow the developments of [14].

8.1. *Problem statement and the pre-bending algorithm*

We begin with the statement of virtual work for the structure from Eq. (5.36), where only the stress terms are left on the left-hand-side: find the displacement of the shell midsurface $\mathbf{d}^h \in \mathcal{S}_d^h$, such that for $\forall\ \delta\mathbf{d}^h \in \mathcal{V}_d^h$:

$$\begin{aligned}
&\int_{\Gamma_0^s} \delta\bar{\boldsymbol{\varepsilon}}^h \cdot \left(\mathbf{A}\bar{\boldsymbol{\varepsilon}}^h + \mathbf{B}\bar{\boldsymbol{\kappa}}^h\right) \mathrm{d}\Gamma \\
&+ \int_{\Gamma_0^s} \delta\bar{\boldsymbol{\kappa}}^h \cdot \left(\mathbf{B}\bar{\boldsymbol{\varepsilon}}^h + \mathbf{D}\bar{\boldsymbol{\kappa}}^h\right) \mathrm{d}\Gamma \\
&+ \int_{\Gamma_0^b} \delta\bar{\boldsymbol{\kappa}}^h \cdot \mathbf{D}^b\bar{\boldsymbol{\kappa}}^h \,\mathrm{d}\Gamma = -\int_{\Gamma_t^s} \delta\mathbf{d}^h \cdot \rho h_{th} \left(\left.\frac{\partial^2 \mathbf{d}^h}{\partial t^2}\right|_{\mathbf{X}} - \mathbf{f}^h \right) \mathrm{d}\Gamma + \int_{(\Gamma_t^s)_h} \delta\mathbf{d}^h \cdot \mathbf{h}^h \,\mathrm{d}\Gamma. \qquad (8.1)
\end{aligned}$$

Although the virtual work equations hold true, the problem setup is unusual in that the stress-free reference configuration Γ_0^s is unknown and the final configuration Γ_t^s is given. The formulation given by Eq. (8.1) is a form of the inverse deformation problem, whose general formulation and treatment was proposed in [148], and further analyzed and studied computationally in [149]. In these references, the focus was placed on developing the right kinematic and stress measures for the inverse deformation problem. Here we develop a simple algorithm for the solution of the inverse deformation equations with application to wind-turbine blades.

We assume that the rotor spins around its axis at a constant angular speed and that the inflow wind conditions do not change. With this setup, the blade is subjected to a constant-in-time centripetal force density (per unit volume) given by

$$\rho \left.\frac{\partial^2 \mathbf{d}^h}{\partial t^2}\right|_{\mathbf{X}} = \rho\boldsymbol{\omega} \times (\boldsymbol{\omega} \times (\mathbf{x} - \mathbf{x}_0)), \qquad (8.2)$$

where the coordinate system of the current configuration is assumed to rotate with the blade, $\boldsymbol{\omega}$ is the vector of angular velocities, and $\mathbf{x}_0$ is a fixed point. The centripetal force density per unit volume may be directly computed as

$$\rho\boldsymbol{\omega} \times \boldsymbol{\omega} \times (\mathbf{x} - \mathbf{x}_0) = \begin{bmatrix} -\rho x\dot{\theta}^2 \\ -\rho y\dot{\theta}^2 \\ 0 \end{bmatrix}, \qquad (8.3)$$

where we assume that the coordinate system of the blade is chosen such that the y-axis is aligned with the blade axis, and the blade rotates around the z-axis with a constant angular speed $\dot{\theta}$.

The time-averaged aerodynamics traction vector $\mathbf{h}^h$ in Eq. (8.1) may be obtained from a separate aerodynamics computation of a rigidly-spinning rotor using the methods described in the earlier sections. (see [77] for the computation of time-averaged traction vector for moving boundary simulations.)

In [14], we proposed the following two-stage iterative approach to solve the variational equations (8.1) for the shell midsurface displacement, which, in turn, gives the stress-free reference configuration.

Initialization: We initialize the unknown reference configuration to coincide with the current configuration, that is,

$$\Gamma_0^s = \Gamma_t^s, \tag{8.4}$$

which implies

$$\mathbf{d}^h = \mathbf{0}. \tag{8.5}$$

Step 1: Given the reference configuration Γ_0^s, we solve the standard nonlinear structural problem: find the structural displacement $\mathbf{d}^h \in \mathcal{S}_d^h$ relative to Γ_0^s, such that $\forall\, \delta\mathbf{d}^h \in \mathcal{V}_d^h$:

$$\begin{aligned}
&\int_{\Gamma_0^s} \delta\overline{\boldsymbol{\varepsilon}}^h \cdot \left(\mathbf{A}\overline{\boldsymbol{\varepsilon}}^h + \mathbf{B}\overline{\boldsymbol{\kappa}}^h\right) \mathrm{d}\Gamma \\
&+ \int_{\Gamma_0^s} \delta\overline{\boldsymbol{\kappa}}^h \cdot \left(\mathbf{B}\overline{\boldsymbol{\varepsilon}}^h + \mathbf{D}\overline{\boldsymbol{\kappa}}^h\right) \mathrm{d}\Gamma \\
&+ \int_{\Gamma_0^b} \delta\overline{\boldsymbol{\kappa}}^h \cdot \mathbf{D}^b\overline{\boldsymbol{\kappa}}^h \,\mathrm{d}\Gamma = -\int_{\Gamma_t^s} \delta\mathbf{d}^h \cdot \left(\rho h_{th}\boldsymbol{\omega} \times (\boldsymbol{\omega} \times (\mathbf{x} - \mathbf{x}_0))\right) \,\mathrm{d}\Gamma + \int_{(\Gamma_t^s)_h} \delta\mathbf{d}^h \cdot \mathbf{h}^h \,\mathrm{d}\Gamma,
\end{aligned} \tag{8.6}$$

Standard Newton–Raphson iteration is employed in this work to compute the solution of the nonlinear structural problem given by Eq. (8.6).

Step 2: Given the displacement solution $\mathbf{d}^h$ from Step 1, we update the reference configuration as

$$\Gamma_0^s = \{\mathbf{X} \mid \mathbf{X} = \mathbf{x} - \mathbf{d}^h,\ \forall \mathbf{x} \in \Gamma_t^s\}, \tag{8.7}$$

and return to Step 1 using $\mathbf{d}^h$ as the initial data.

Steps 1–2 are repeated until convergence, that is, until $\mathbf{d}^h$ satisfies Eq. (8.6).

The above algorithm is based on the idea of computing negative increments of the displacement, or increments of the displacement away from the current configuration, until the reference configuration is found. The mathematical justification for this approach may be found in the appendix of [14]. In what follows, we will illustrate the good performance of the proposed algorithm on a full-scale wind-turbine blade subject to realistic wind and inertial loads.

8.2. *Pre-bending results for the NREL 5MW wind-turbine blade*

The same blade design and wind conditions as in the previous section are taken for the pre-bending computations presented here. Figure 35 shows the tip displacement convergence of the iterative pre-bending algorithm. After a few (5–6) iterations of the two-step pre-bending algorithm the tip exhibits no further visible displacements, and the computation is stopped after a total of 15 iterations. Figure 36 shows the initial and the final stress-free blade shapes. As expected, the blade bends into the wind. The tip deflection is predicted to be 5.61 m.

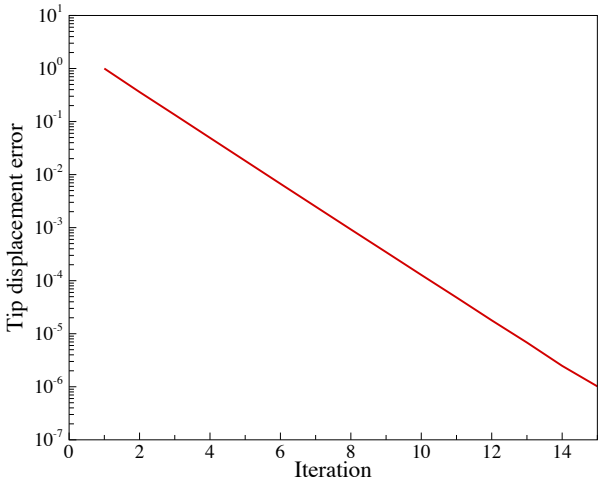


Fig. 35. Blade tip displacement convergence as a function of the iteration number.

We examine the stress distribution in the composite plies of the blade skin. For each ply we compute the Cauchy stress tensor components with respect to the local Cartesian basis that is aligned with the material axes. The first basis vector points in the direction of the fiber and the second in the direction of the matrix, which is orthogonal to the fiber direction (see Eqs. (5.12) and (5.13)). The maximum values of the tensile (σ^t), compressive (σ^c), and in-plane shear stresses are computed for each ply. The highest ratio of the predicted Cauchy stress and the composite strength, i.e, $\sigma_2^t/\sigma_2^{t,u}$, occurs for the tensile stress in the direction of the matrix material. Although the ratio does not exceed 0.6, which means the predicted stress is below the composite failure strength, we feel this value is somewhat high. In the rest of the stress components the ratios are significantly lower. Figure 37 shows the distribution of σ_2 in the 0° ply number 14, which has the highest ratio of $\sigma_2^t/\sigma_2^{t,u}$. The pressure side of the blade is in tension, while the suction side of the blade is in compression as expected. However, the level of the tensile stress is not very far from the tensile failure strength, which suggests that stronger matrix material may be desirable for this blade design.

9. Conclusions

In this review paper we presented our computational procedures, which include both aerodynamics and FSI techniques, for the simulation of wind-turbine rotors at full scale. We applied these computational procedures to the simulation of the NREL Phase VI two-bladed rotor and the NREL 5MW rotor. We validated our aerodynamics procedures using the experimental results for the Phase IV rotor. The use of weakly-enforced essential boundary conditions was critical in obtaining accurate results when using a coarse boundary layer mesh. Furthermore, no "tuning" of the turbulence model was necessary as the same mul-

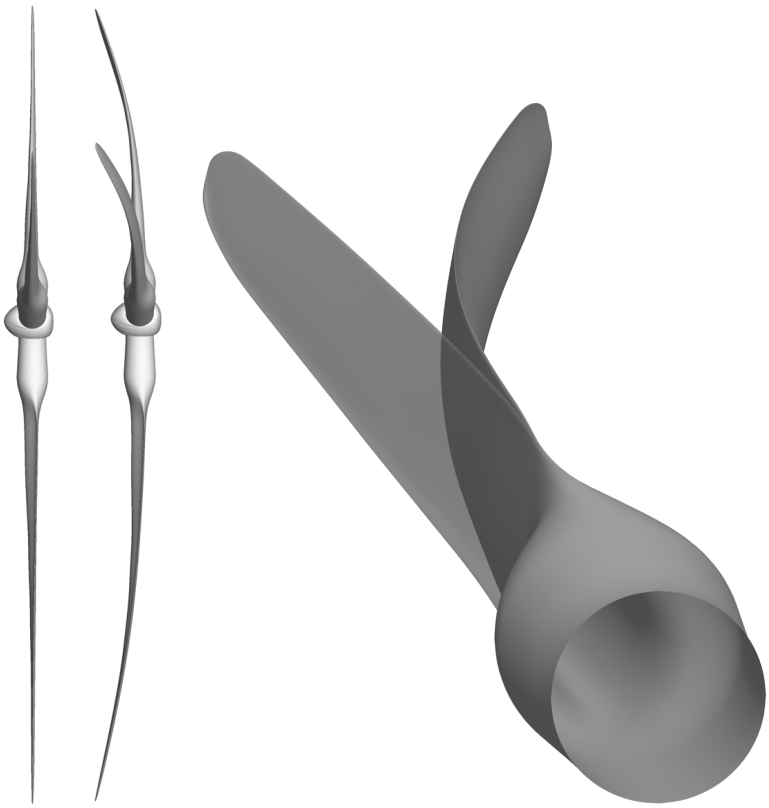


Fig. 36. Left: rotor design configuration. Middle: rotor pre-bent configuration. Right: rotor blade design and pre-bent configurations superposed.

tiscale formulation was able to correctly represent both the fully-attached and detached turbulent boundary layers.

The DSD/SST formulation and its VMS version were compared in the aerodynamics computations of the 5MW rotor. It was found that the VMS version produces more accurate results for the aerodynamic torque than the original DSD/SST formulation, suggesting that the VMS version is capable of better representing the turbulent boundary layer.

In the case of FSI simulations, the structure is governed by the rotation-free Kirchhoff–Love shell theory, complemented with the bending-strip method. The fluid and structure are strongly coupled at the interface. The strong coupling was in part facilitated by the fact that the structure has only displacement degrees of freedom. The coupled system was solved in a block-iterative fashion, which is robust enough for the present application due to the relatively high structural mass of the wind-turbine blades.

For wind-turbine rotors the structural motion is dominated by rotation about the horizontal axis. For this we found it advantageous for overall accuracy of the computations to separate the structural displacement into rotation and deflection parts. With this decompo-

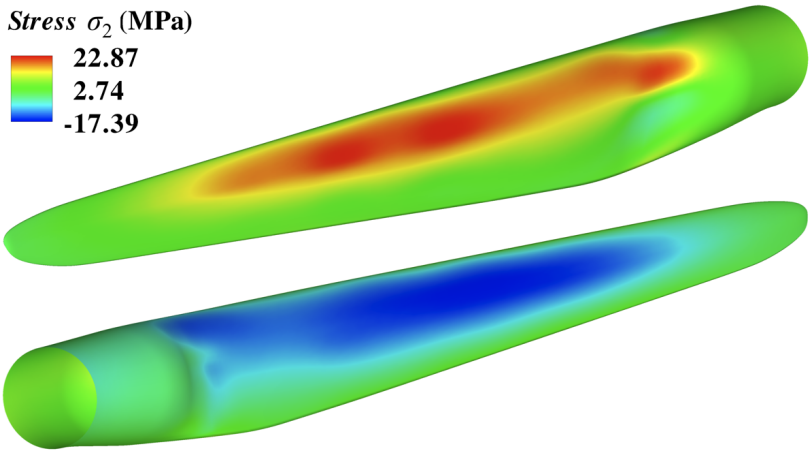


Fig. 37. Normal stress distribution in the direction of the matrix material for the 0° ply number 14. Top: pressure side. Bottom: suction side.

sition, we modified the time integration formulas to treat the rotation part of the structural motion exactly. The decomposition of the structural displacement also yields a more effective mesh-moving computation, where only the deflection part of the fluid-mesh motion is computed with the elasticity-based mesh-moving method. The rotational part of the mesh motion is calculated exactly.

We performed the FSI simulation of the NREL 5MW offshore baseline wind-turbine rotor. The rotor blades were modeled as symmetric composite laminates homogenized in the through-thickness direction. The computation yielded good prediction of the aerodynamic torque and blade-tip deflection.

We proposed a pre-bending algorithm, which is based on separate aerodynamics and structural mechanics computations and performs well for full-scale wind-turbine blades.

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 paper. M.-C. Hsu was supported by the Los Alamos – UC San Diego Educational Collaboration Fellowship. Y. Bazilevs would like to acknowledge the support of the NSF CAREER Award and the Air Force Office of Scientific Research Award FA9550-12-1-0005. This research was supported in part by the Rice Computational Research Cluster funded by NSF under Grant CNS-0821727. The ST-VMS part of the work was supported by the ARO Grant W911NF-09-1-0346 (fourth author) and Rice–Waseda research agreement (third author).

References

1. J.M. Jonkman and M.L.B. Jr., "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 × 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. F. Zahle, N.N. Sørensen, and J. Johansen, "Wind turbine rotor-tower interaction using an incompressible overset grid method", *Wind Energy*, **12** (2009) 594–619.
6. 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/fld.2400.
7. 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.
8. Y. Li and P.M.C. K. J. Paik, T. Xing, "Dynamic overset CFD simulations of wind turbine aerodynamics", *Renewable Energy*, **37** (2012) 285–298.
9. E. Guttierez, S. Primi, F. Taucer, P. Caperan, D. Tirelli, J. Mieres, I. Calvo, J. Rodriguez, F. Vallano, G. Galiotis, and D. Mouzakis, "A wind turbine tower design based on fibre-reinforced composites", Technical report, Joint Research Centre - Ispra, European Laboratory for Structural Assessment (ELSA), Institute For Protection and Security of the Citizen (IPSC), European Commission, 2003.
10. 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.
11. M.O.L. Hansen, J.N. Sørensen, S.V. andN. Sørensen, and H.A. Madsen, "State of the art in wind turbine aerodynamics and aeroelasticity", *Progress in Aerospace Sciences*, **42** (2006) 285–330.
12. 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.
13. J. Kiendl, Y. Bazilevs, M.-C. Hsu, R. Wüchner, and K.-U. Bletzinger, "The bending strip method for isogeometric analysis of Kirchhoff–Love shell structures comprised of multiple patches", *Computer Methods in Applied Mechanics and Engineering*, **199** (2010) 2403–2416.
14. 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*, (2011), doi:10.1002/nme.3244.
15. 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.
16. 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.
17. 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.
18. 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.
19. 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.
20. W.A. Wall, M.A. Frenzel, and C. Cyron, "Isogeometric structural shape optimization", *Computer Methods in Applied Mechanics and Engineering*, **197** (2008) 2976–2988.
21. J.A. Cottrell, T.J.R. Hughes, and Y. Bazilevs, *Isogeometric Analysis: Toward Integration of CAD and FEA*. Wiley, Chichester, 2009.
22. J.A. Evans, Y. Bazilevs, I.B. ska, 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.
23. 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.
24. 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.
25. F. Auricchio, L. Beirão da Veiga, C. Lovadina, and A. Reali, "The importance of the exact satisfaction of the incompressibility constraint in nonlinear elasticity: Mixed FEMs versus NURBS-based approximations", *Computer Methods in Applied Mechanics and Engineering*, **199** (2010) 314–323.
26. W. Wang and Y. Zhang, "Wavelets-based NURBS simplification and fairing", *Computer Methods in Applied Mechanics and Engineering*, **199** (2010) 290–300.
27. E. Cohen, T. Martin, R.M. Kirby, T. Lyche, and R.F. Riesenfeld, "Analysis-aware modeling: Understanding quality considerations in modeling for isogeometric analysis", *Computer Methods in Applied Mechanics and Engineering*, **199** (2010) 334–356.
28. V. Srinivasan, S. Radhakrishnan, and G. Subbarayan, "Coordinated synthesis of hierarchical engineering systems", *Computer Methods in Applied Mechanics and Engineering*, **199** (2010) 392–404.
29. Y. Bazilevs, V.M. Calo, J.A. Cottrel, 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.
30. 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.
31. 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.
32. 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.
33. 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.
34. 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.
35. 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.

36. 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.
37. 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.
38. 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.
39. 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.
40. 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.
41. 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.
42. 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.
43. 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.
44. 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.
45. 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.
46. 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.
47. 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.
48. 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.
49. 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.
50. 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.
51. 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.

52. 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/fld.505.
53. 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/fld.1430.
54. 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.
55. 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.
56. 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.
57. 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.
58. Y. Bazilevs and T.J.R. Hughes, "Weak imposition of Dirichlet boundary conditions in fluid mechanics", *Computers and Fluids*, **36** (2007) 12–26.
59. J. Nitsche, "Uber ein variationsprinzip zur losung von Dirichlet-problemen bei verwendung von teilraumen, die keinen randbedingungen unterworfen sind", *Abh. Math. Univ. Hamburg*, **36** (1971) 9–15.
60. 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.
61. 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.
62. 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.
63. 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/fld.1650150911.
64. 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/fld.1650211011.
65. 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.
66. 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, doi: 10.1016/S0045-7825(00)00208-5.
67. 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, doi: 10.1016/S0045-7825(01)00311-5.
68. 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.

69. 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.
70. 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, doi: 10.1002/fld.1443.
71. 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, doi: 10.1002/fld.1633.
72. 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, doi: 10.1007/s00466-008-0261-7.
73. 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, doi: 10.1016/j.cma.2008.05.024.
74. T.E. Tezduyar, K. Takizawa, C. Moorman, S. Wright, and J. Christopher, "Multiscale sequentially-coupled arterial FSI technique", *Computational Mechanics*, **46** (2010) 17–29, doi: 10.1007/s00466-009-0423-2.
75. 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, doi: 10.1002/fld.2221.
76. 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, doi: 10.1002/cnm.1241.
77. 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, doi: 10.1007/s00466-009-0425-0.
78. 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, doi: 10.1002/fld.2348.
79. 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, doi: 10.1002/fld.2360.
80. 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, doi: 10.1002/fld.2359.
81. 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.
82. K. Takizawa, T. Spielman, and T.E. Tezduyar, "Space–time FSI modeling and dynamical analysis of spacecraft parachutes and parachute clusters", *Computational Mechanics*, **48** (2011) 345–364, doi: 10.1007/s00466-011-0590-9.
83. K. Takizawa, T. Brummer, T.E. Tezduyar, and P.R. Chen, "A comparative study based on patient-specific fluid–structure interaction modeling of cerebral aneurysms", *Journal of Applied Mechanics*, to appear, 2011.
84. T.E. Tezduyar, S. Sathe, J. Pausewang, M. Schwaab, J. Christopher, and J. Crabtree, "Fluid–structure interaction modeling of ringsail parachutes", *Computational Mechanics*,

43 (2008) 133–142, doi: 10.1007/s00466-008-0260-8.

85. K. Takizawa, T. Spielman, C. Moorman, and T.E. Tezduyar, "Fluid–structure interaction modeling of spacecraft parachutes for simulation-based design", *Journal of Applied Mechanics*, to appear, 2011.
86. 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, doi: 10.1007/s00466-008-0276-0.
87. M. Manguoglu, A.H. Sameh, F. Saied, T.E. Tezduyar, and S. Sathe, "Preconditioning techniques for nonsymmetric linear systems in computation of incompressible flows", *Journal of Applied Mechanics*, **76** (2009) 021204, doi: 10.1115/1.3059576.
88. 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, doi: 10.1007/s00466-009-0426-z.
89. 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/fld.2415.
90. 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.
91. 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.
92. 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.
93. 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.
94. 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*, to appear, 2011.
95. 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.
96. M.C. Hsu, I. Akkerman, and Y. Bazilevs, "Wind turbine aerodynamics using ALE–VMS: Validation and the role of weakly enforced boundary conditions", *Computational Mechanics*, (2011), In review.
97. 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.
98. C. Johnson, *Numerical solution of partial differential equations by the finite element method*. Cambridge University Press, Sweden, 1987.
99. S.C. Brenner and L.R. Scott, *The Mathematical Theory of Finite Element Methods, 2nd ed.* Springer, 2002.
100. A. Ern and J.L. Guermond, *Theory and Practice of Finite Elements*. Springer, 2004.
101. T.J.R. Hughes and T.E. Tezduyar, "Finite element methods for first-order hyperbolic systems with particular emphasis on the compressible Euler equations", *Computer Methods in Applied*

Mechanics and Engineering, **45** (1984) 217–284, doi: 10.1016/0045-7825(84)90157-9.

102. 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, doi: 10.1016/0045-7825(86)90003-4.
103. 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.
104. 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, doi: 10.1016/S0045-7825(00)00211-5.
105. T.J.R. Hughes, G. Scovazzi, and L.P. Franca, "Multiscale and stabilized methods", in E. Stein, R. de Borst, and T.J.R. Hughes, editors, *Encyclopedia of Computational Mechanics, Vol. 3, Fluids*, Chapter 2, Wiley, 2004.
106. T.J.R. Hughes, G.R. Feijóo, L. Mazzei, and J.B. Quincy, "The variational multiscale method–A paradigm for computational mechanics", *Computer Methods in Applied Mechanics and Engineering*, **166** (1998) 3–24.
107. R. Calderer and A. Masud, "A multiscale stabilized ALE formulation for incompressible flows with moving boundaries", *Computational Mechanics*, **46** (2010) 185–197.
108. F. Brezzi, L.P. Franca, T.J.R. Hughes, and A. Russo, "$b = \int g$", *Computer Methods in Applied Mechanics and Engineering*, **145** (1997) 329–339.
109. J.E. Akin, T. Tezduyar, M. Ungor, and S. Mittal, "Stabilization parameters and Smagorinsky turbulence model", *Journal of Applied Mechanics*, **70** (2003) 2–9, doi: 10.1115/1.1526569.
110. 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, doi: 10.1016/j.cma.2003.12.050.
111. 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, doi: 10.1002/cnm.759.
112. T.E. Tezduyar, M. Senga, and D. Vicker, "Computation of inviscid supersonic flows around cylinders and spheres with the SUPG formulation and YZβ shock-capturing", *Computational Mechanics*, **38** (2006) 469–481, doi: 10.1007/s00466-005-0025-6.
113. T.E. Tezduyar and S. Sathe, "Enhanced-discretization selective stabilization procedure (EDSSP)", *Computational Mechanics*, **38** (2006) 456–468, doi: 10.1007/s00466-006-0056-7.
114. E. Onate, A. Valls, , and J. Garcia, "FIC/FEM formulation with matrix stabilizing terms for incompressible flows at low and high Reynolds numbers", *Computational Mechanics*, **38** (2006) 440–455.
115. 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, doi: 10.1007/s00466-006-0045-x.
116. 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, doi: 10.1007/s00466-006-0033-1.
117. T.E. Tezduyar, "Finite elements in fluids: Stabilized formulations and moving boundaries and interfaces", *Computers & Fluids*, **36** (2007) 191–206, doi: 10.1016/j.compfluid.2005.02.011.
118. 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, doi: 10.1016/j.compfluid.2005.07.004.

119. 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, doi: 10.1007/s00466-009-0441-0.
120. 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, doi: 10.1002/fld.2451.
121. A. Corsini, F. Rispoli, and T.E. Tezduyar, "Computer modeling of wave-energy air turbines with the SUPG/PSPG formulation and discontinuity-capturing technique", *Journal of Applied Mechanics*, to appear, 2011.
122. B.E. Launder and D.B. Spalding, "The numerical computation of turbulent flows", *Computer Methods in Applied Mechanics and Engineering*, **3** (1974) 269–289.
123. D.C. Wilcox, *Turbulence Modeling for CFD*. DCW Industries, La Canada, CA, 1998.
124. H.J.T. Kooijman, C. Lindenburg, and D.W.E.L. van der Hooft, "DOWEC 6 MW pre-design: Aero-elastic modelling ofthe DOWEC 6 MW pre-design in PHATAS", Technical Report DOWEC-F1W2-HJK-01-046/9, 2003.
125. 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*, 53–81, Springer, 2010, ISBN 3642142052.
126. 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.
127. T.A.C.C. (TACC), "Available at: http://www.tacc.utexas.edu", Accessed October 6, 2011.
128. 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.
129. E. Hau, *Wind Turbines: Fundamentals, Technologies, Application, Economics. 2nd Edition*. Springer, Berlin, 2006.
130. 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.
131. D.J. Laino, A.C. Hansen, and J.E. Minnema, "Validation of the AeroDyn subroutines using NREL Unsteady Aerodynamics Experiment data", *Wind Energy*, **5** (2002) 227–244.
132. C. Tongchitpakdee, S. Benjanirat, and L.N. Sankar, "Numerical simulation of the aerodynamics of horizontal axis wind turbines under yawed flow conditions", *Journal of Solar Energy Engineering*, **127** (2005) 464–474.
133. S. Schmitz and J.J. Chattot, "Characterization of three-dimensional effects for the rotating and parked NREL Phase VI wind turbine", *Journal of Solar Energy Engineering*, **128** (2006) 445–454.
134. F. Shakib, T.J.R. Hughes, and Z. Johan, "A multi-element group preconditionined GMRES algorithm for nonsymmetric systems arising in finite element analysis", *Computer Methods in Applied Mechanics and Engineering*, **75** (1989) 415–456.
135. T. Belytschko, W.K. Liu, and B. Moran, *Nonlinear Finite Elements for Continua and Structures*. Wiley, 2000.
136. M. Bischoff, W.A. Wall, K.U. Bletzinger, and E. Ramm, "Models and finite elements for thin-walled structures", in E. Stein, R. de Borst, and T.J.R. Hughes, editors, *Encyclopedia of Computational Mechanics, Vol. 2, Solids, Structures and Coupled Problems*, Chapter 3, Wiley, 2004.
137. J.N. Reddy, *Mechanics of Laminated Composite Plates and Shells: Theory and Analysis, 2nd ed.* CRC Press, Boca Raton, FL, 2004.
138. K.U. Bletzinger, S. Kimmich, and E. Ramm, "Efficient modeling in shape optimal design", *Computing Systems in Engineering*, **2** (1991) 483–495.
139. D.J. Benson, Y. Bazilevs, M.C. Hsu, and T.J.R. Hughes, "A large deformation, rotation-

free, isogeometric shell", *Computer Methods in Applied Mechanics and Engineering*, **200** (2011) 1367–1378.

140. T.J.R. Hughes, *The finite element method: Linear static and dynamic finite element analysis*. Prentice Hall, Englewood Cliffs, NJ, 1987.
141. J. Chung and G.M. Hulbert, "A time integration algorithm for structural dynamics withimproved numerical dissipation: The generalized-α method", *Journal of Applied Mechanics*, **60** (1993) 371–75.
142. H. Melbø and T. Kvamsdal, "Goal oriented error estimators for Stokes equations based on variationally consistent postprocessing", *Computer Methods in Applied Mechanics and Engineering*, **192** (2003) 613–633.
143. E.H. van Brummelen, K.G. van der Zee, V.V. Garg, and S. Prudhomme, "Flux evaluation in primal and dual boundary-coupled problems", *Journal of Applied Mechanics*, (2012), Published online. DOI: 10.1115/1.4005187.
144. 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.
145. 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.
146. 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.
147. 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.
148. R.T. Shield, "Inverse deformation results in finite elasticity", *ZAMP*, **18** (1967) 381–389.
149. S. Govindjee and P.A. Mihalic, "Computational methods for inverse finite elastostatics", *Computer Methods in Applied Mechanics and Engineering*, **136** (1996) 47–57.