3D Simulation of Wind Turbine Rotors at Full Scale. Part I: Geometry Modeling and Aerodynamics

Y. Bazilevs[a,*], M.-C. Hsu[a], I. Akkerman[a], S. Wright[b], K. Takizawa[b], B. Henicke[b], T. Spielman[b], T.E. Tezduyar[b]

[a]*Department of Structural Engineering, University of California, San Diego, 9500 Gilman Drive, La Jolla, CA 92093, USA*
[b]*Mechanical Engineering, Rice University - MS 321, 6100 Main Street, Houston, TX 77005, USA*

Abstract

In this two-part article we present a collection of numerical methods combined into a single framework, which has the potential for a successful application to wind turbine rotor modeling and simulation. In Part 1 of this article we focus on: 1. The basics of geometry modeling and analysis-suitable geometry construction for wind turbine rotors; 2. The fluid mechanics formulation and its suitability and accuracy for rotating turbulent flows; 3. The coupling of air flow and a rotating rigid body. In Part 2 we focus on the structural discretization for wind turbine blades and the details of the fluid–structure interaction computational procedures. The methods developed are applied to the simulation of the NREL 5MW offshore baseline wind turbine rotor. The simulations are performed at realistic wind velocity and rotor speed conditions and at full spatial scale. Validation against published data is presented and possibilities of the newly developed computational framework are illustrated on several examples.

Keywords: wind turbine rotor, wind turbine blades, fluid–structure interaction, geometry modeling, isogeometric analysis, NURBS, rotating turbulent flow, aerodynamic torque

1. Introduction

Countries in Europe and Asia are putting substantial effort behind the development of wind energy technologies. Currently in the EU, 50 GW of electricity comes from the on-land and 1 GW from offshore wind turbines. The EU target is to raise the on-land production to 130 GW and offshore to 50 GW by 2020. The latter figure represents a fifty-fold increase, which will require a significant investment and engineering effort. The US Government recently established the objective that wind

*Corresponding author
Email address: yuri@ucsd.edu (Y. Bazilevs)

power should supply 25% of the US energy needs by 2025. Achieving this objective will require nearly a 1200% increase in wind power capacity (from 25 GW to 305 GW). We believe that in order to achieve this goal, the US must also invest significant resources in the development of offshore wind turbines. We also believe that leading-edge wind energy research and development, which includes advanced simulation, will be essential in meeting this goal.

The present cost for wind energy is nonetheless strongly dominated by rising operations and maintenance costs over the current 20-year design lifetime of the system [1]. Although the design lifetime is 20 years, a typical wind turbine fails 2.6 times during the first 10 years, usually within the gearbox, generator and rotor assembly, and the rotor assembly has been identified as the top opportunity for major advancement in design and performance improvements [2]. While wind turbine rotor failures occur due to a variety of reasons, fatigue failure of the wind turbine blades due to their everyday operation is recognized as one of the major causes. However, the industry is currently unable to predict these failure mechanisms, which leads to the unscheduled downtime, expensive maintenance and reduced capacity.

In offshore environments the winds are typically stronger and are more sustained than inland, providing a more reliable source of energy. This in part explains the attraction to offshore designs. However, offshore wind turbines are exposed to harsh environments and must be designed to reliably sustain increased wind loads. Increased wind speeds also imply that the blades of much larger diameter (120 m – 190 m) must be designed and built for better performance. These are significant engineering challenges that must be addressed through advanced research and development, which also involves large-scale advanced simulation.

The current practice in wind turbine simulation makes use of steady (time-independent), 2D lumped-parameter aerodynamic models for airfoil cross-sections that are coupled with 1D beam-type structures to evaluate wind turbine blade designs and, specifically, their aerodynamic performance. These models are simple to implement and fast to execute, which makes them attractive for industrial applications, especially if they are routinely used as part of the design cycle. However, due to the steady nature of the flow conditions and the lack of real 3D geometry and physics, these models are unable to adequately represent the system response to time-dependent phenomena, such as wind gusts, or phenomena attributable to complex blade geometry and material composition, such as flow separation and reattachment and detailed blade deformations and stress distributions. It is precisely these more extreme events that cause gearbox and blade failures and significantly reduce the life cycle of wind turbines, leading to premature maintenance and repair and, as a result, to increased cost of wind energy. A more fundamental problem with these simple models is their non-hierarchical nature: it is virtually not possible to enhance them with features necessary for predicting more extreme events or richer physics without going to a more advanced modeling framework all together.

In this work we propose to introduce a paradigm shift in wind turbine model-

ing and simulation by developing 3D, complex geometry, time-dependent, multi-physics modeling procedures for wind turbine fluid–structure interaction (FSI). In particular, we focus on predicting wind turbine blade–air flow interaction phenomena for real wind turbines, operating under real wind conditions, and at full design scale. The particular focus on turbine blades is motivated by the fact that improved blade efficiency directly translates to lower cost of wind energy conversion. However, full-scale 3D FSI simulations of wind turbines engender a set of challenges. The air flow Reynolds number is on the order of $10^7 - 10^8$, which is challenging in terms of both flow modeling and simulation. This Reynolds number necessitates the use of fine grids, good quality basis functions, and large-scale high-performance computing. Wind turbine blades are long and slender structures that are made of several structural components with complex distribution of material properties, requiring both advanced computational model generation and simulation methods. The numerical approach for structural mechanics must have good accuracy and avoid locking. The latter is typical of thin structures [3]. Wind turbine blades are manufactured using multi-layer composite materials that also require appropriate numerical treatment. The numerical simulations simultaneously involve moving and stationary components, which must be handled correctly. The fluid–structure coupling must be efficient and robust to preclude divergence of the computations. Some of these challenges will be addressed in this work.

Isogeometric Analysis, first introduced in [4] and further expanded on in [5–16], is adopted as the geometry modeling and simulation framework for wind turbines. It is particularly well suited for this application for the following reasons. We use NURBS (non-uniform rational B-splines), the most developed basis function technology for isogeometric analysis. NURBS are more efficient than standard finite elements for representing complex, smooth geometrical shapes, such as wind turbine blades. Because the geometry and solution fields are represented using the same functional description, the integration of geometry modeling with structural design and computational analysis is greatly simplified. Isogeometric analysis was successfully employed for computation of turbulent flows [17–22], nonlinear structures [23–28] and FSI [29–32]. In most cases, isogeometric analysis gave a clear advantage over standard low-order finite elements in terms of solution per-degree-of-freedom accuracy, which 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 [33]. In addition, an isogeometric representation of the analysis-suitable geometry may be used to construct tetrahedral and hexahedral meshes for computations using finite elements. In this paper, we use such tetrahedral meshes for wind turbine computation with the Deforming-Spatial-Domain/Stabilized Space–Time (DSD/SST) formulation [34–38].

The DSD/SST formulation was introduced in [34–36] as a general-purpose interface-tracking (moving-mesh) technique for flow computations involving moving boundaries and interfaces, including FSI and flows with moving mechani-

cal components. Some of earliest FSI computations with the DSD/SST formulation were reported in [39] for vortex-induced vibrations of a cylinder and in [40] 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 [41–43] and evolving to computations with direct coupling [44, 45]. New versions of the DSD/SST formulation introduced in [38] are the core technologies of the stabilized space–time FSI (SSTFSI) technique, which was also introduced in [38]. The SSTFSI technique, combined with a number of special techniques [38, 46–56], have been used in some of the most challenging parachute FSI computations [48, 51, 54, 56, 57], and also in a good number of patient-specific arterial FSI computations [50, 52, 53, 55]. In application of the DSD/SST formulation to flows with moving mechanical components, the Shear–Slip Mesh Update Method (SSMUM) [58–60] 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 [58]). 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 [59] and flow around a helicopter with its rotor in motion [60].

Part 1 of this paper, which focuses on geometry modeling and aerodynamics of wind turbines, is outlined as follows. In Section 2, we describe a template-based method for creating an analysis-suitable NURBS geometry for FSI simulation of wind turbine rotors. We apply the method developed to construct the NREL 5MW offshore baseline wind turbine rotor given in [61]. In Section 3, we develop the residual-based variational multiscale formulation of the Navier–Stokes equations of incompressible flow on a moving domain. We develop a generalization of the quasi-static fine scale assumption to the moving domain case, which leads to a semi-discrete formulation that globally conserves linear momentum. In Section 4, we perform a partial validation of the proposed computational method on a turbulent Taylor–Couette flow. In Section 5, we formulate a coupled problem, which involves incompressible flow with a rigidly rotating body. Rotationally-periodic boundary conditions, which allow us to take advantage of the problem symmetry for better computational efficiency, are also described. To assess the accuracy of the numerical method, the NREL 5MW offshore baseline wind turbine rotor is simulated using

prescribed inflow wind conditions and rotor speed. The NURBS-based simulation results compare favorably to the reference data from [61]. A coupled simulation for the case of high inflow wind speed and free rotation, which corresponds to the case of rotor over-spinning in the case of the control system failure, is also performed to illustrate the possibilities of the proposed computational procedures. In Section 6, we draw conclusions and outline future research directions.

2. Analysis-Suitable Geometry Construction for Wind Turbine Rotors

We propose a template-based wind turbine geometry modeling approach that makes use of volumetric NURBS. The method entails construction of one or more (small number of) template geometries of wind turbine designs. The templates include and closely approximate the geometry of the rotor blades and hub (nacelle), and the flow domain around. The template geometry is then deformed to the actual geometry of the wind turbine by appropriately minimizing the error between them. Once the model is generated, an analysis-ready geometry is produced with user control over mesh refinement and domain partitioning for efficient parallel processing. The advantage of this approach is that it can be specialized and optimized for a particular class of geometries. For example, a template-based geometry modeling approach was developed and successfully employed for NURBS modeling and FSI simulation of vascular blood flow with patient-specific data in [29].

2.1. Wind turbine rotor blade and hub geometry construction

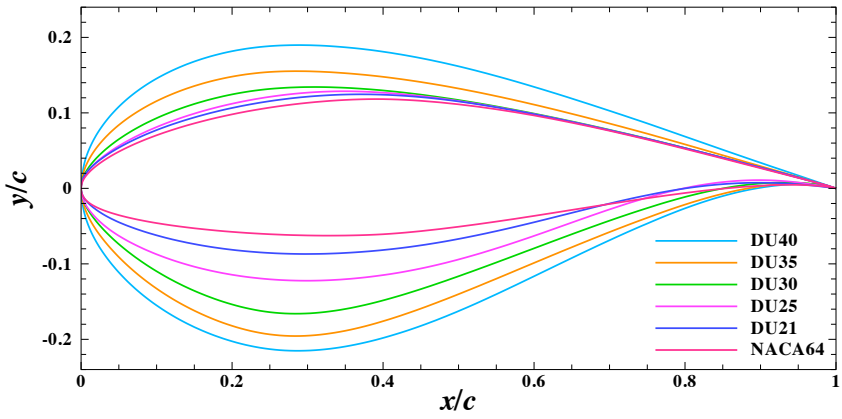


Figure 1: Airfoil cross-sections used in the design of the wind turbine rotor blades.

As a first step we construct a template for the structural model of the rotor. In this paper, 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

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 from [61].

axis direction. Other structural components, such as shear webs and spar caps [62] will be added to our template in the subsequent work.

The geometry of the rotor blade considered in this work is based on the NREL 5MW offshore baseline wind turbine described in [61]. 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 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 [61], but mentioned in [63].

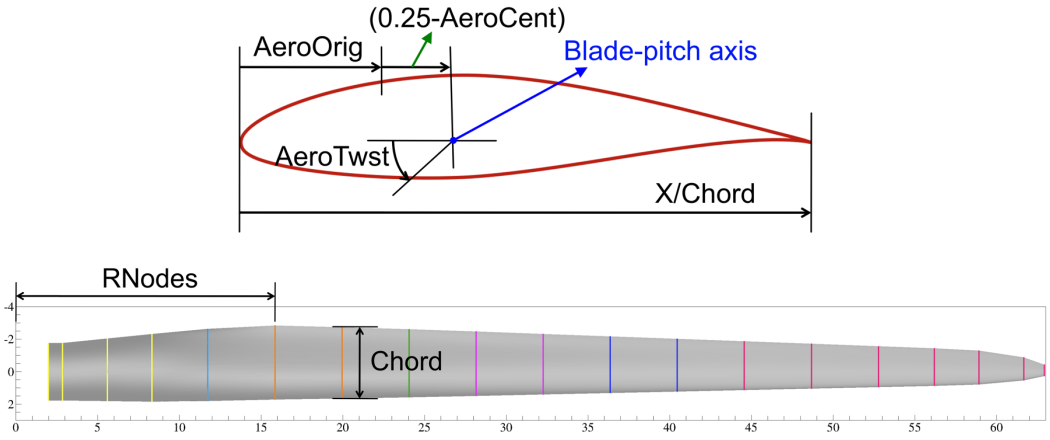


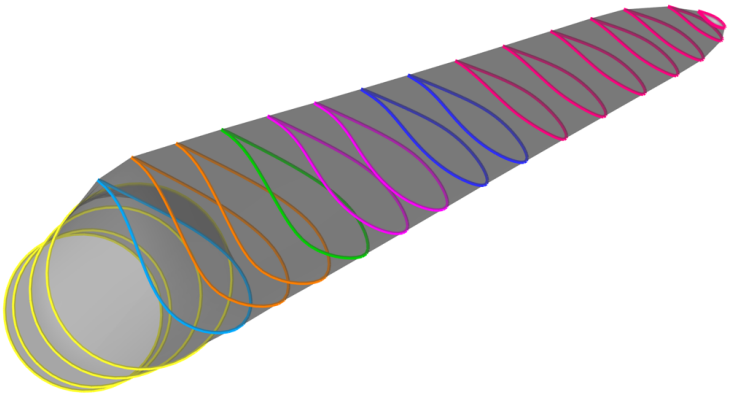
Figure 2: Illustration of quantities from Table 1.

Remark: There is some redundancy in the parameters given in Table 1. The variable "AeroCent" is used as an input to FAST [64], which is the aerodynamics modeling software that is typically used for wind turbine rotor computations. FAST is based on look-up tables and provides blade cross-section steady-state lift and drag forces given the airfoil type, relative wind speed, and angle of attack. The effects of the hub, trailing edge turbulence, and blade tip are modeled using empirical relationships. FAST 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 reference reports.

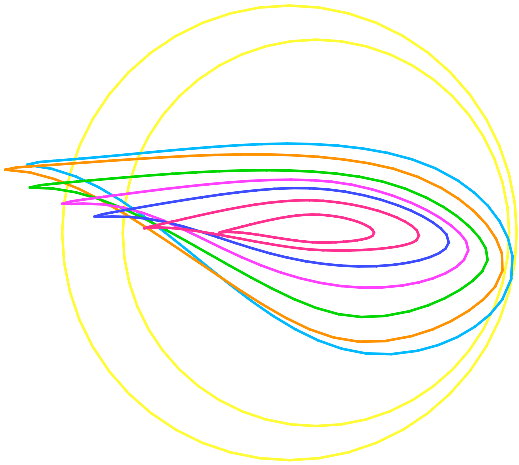
For each blade cross-section we use quadratic NURBS to represent the 2D airfoil shape. The weights of the NURBS functions are set to unity. The weights are adjusted near the root to represent the circular cross-sections of the blade exactly. The cross-sections are lofted in the blade axis direction, 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. The final shape of the blade along with the airfoil cross-sections is shown in Figure 3a. Figure 3b shows a top view of the blade in which the twisting of the cross-sections is evident.

2.2. Volumetric NURBS

Given the rotor blade surface description, the surrounding fluid domain volume is constructed next. The blade surface is split into four patches of similar size, which we call the blade surface patches. The splitting is done at the leading and trailing



(a)



(b)

Figure 3: (a) Airfloil cross-sections superposed on the wind turbine blade. (b) Top view of a subset of the airfoil cross-sections illustrating blade twisting.

edges, as well as half-way in between on both sides of the blade. The volumetric 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 fluid 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 (see Figure 4a). As a next step, we perform knot insertion and move the new control points such that their locations coincide with those of the blade surface patch. See Figure 4b and Figure 4c for knot insertion and control point re-positioning, respectively. This generates an a-priori conforming discretization between the volumetric 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 4d 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) and is shown in Figure 5a. The fluid mesh cross-section that also shows the details of mesh refinement in the boundary layer is shown in Figure 5b. The wind turbine rotor surface model and outer cylindrical domain of the three-blade configuration are shown in Figure 6. Mesh refinement was concentrated near the wind turbine blade surface. No special attention was paid to mesh design in the outer regions of the computational domain.

Remark: We note that the template-based nature of the proposed approach and a relatively small number of input parameters for model building allow one to alter the wind turbine rotor models with minimal effort.

3. Navier–Stokes Equations of Incompressible Flow and Residual-Based Turbulence Modeling for Moving Domains

3.1. The weak formulation

In what follows, $\Omega \subset \mathbb{R}^d$, $d = 2, 3$ denotes the spatial domain occupied by the fluid at the current time and $\Gamma = \partial\Omega$ is its boundary. Let $\mathcal{V}$ and $\mathcal{W}$ denote the infinite-dimensional trial solution and weighting function spaces, respectively, and let $(\cdot, \cdot)_\Omega$ denote the L^2-inner product over Ω. The weak form of the Navier–Stokes equations of incompressible flow over a moving domain is stated as follows: find the velocity-pressure pair $\{\boldsymbol{v}, p\} \in \mathcal{V}$ such that for all momentum and continuity weighting functions $\{\boldsymbol{w}, q\} \in \mathcal{W}$,

$$B(\{\boldsymbol{w}, q\}, \{\boldsymbol{v}, p\}) - (\boldsymbol{w}, \rho \boldsymbol{f})_\Omega = 0, \tag{1}$$

where

$$B(\{\boldsymbol{w}, q\}, \{\boldsymbol{v}, p\}) = \left(\boldsymbol{w}, \rho \frac{\partial \boldsymbol{v}}{\partial t}\right)_\Omega + (\boldsymbol{w}, \rho(\boldsymbol{v} - \hat{\boldsymbol{v}}) \cdot \nabla \boldsymbol{v})_\Omega + (q, \nabla \cdot \boldsymbol{v})_\Omega - (\nabla \cdot \boldsymbol{w}, p)_\Omega + (\nabla^s \boldsymbol{w}, 2\mu \nabla^s \boldsymbol{v})_\Omega \,. \tag{2}$$

In the above, $\hat{\boldsymbol{v}}$ is the fluid domain velocity, μ is the kinematic viscosity, ρ is the density, and

$$\nabla^s \boldsymbol{v} = \frac{1}{2}(\nabla \boldsymbol{v} + \nabla \boldsymbol{v}^T) \tag{3}$$

is the symmetric gradient of the particle velocity field. Equation (1) is a weak form corresponding to the Arbitrary Lagrangian–Eulerian (ALE) formulation of the Navier–Stokes equations and the time derivative in the first term on the right-hand-side of Eq. (2) is taken with respect to a fixed spatial coordinate in the reference configuration. In the absence of the fluid domain motion, the above formulation reverts to a standard incompressible flow formulation on a stationary domain.

Direct replacement of the infinite-dimensional spaces $\mathcal{V}$ and $\mathcal{W}$ by their finite dimensional counterparts leads to the Galerkin formulation, which is unstable for advection-dominated flow and for the choice of equal-order velocity–pressure discretization. In what follows, we make use of the variational multiscale method (see, e.g., [65, 66]) to generate a stable and accurate discrete formulation over a moving domain that is suitable for equal-order velocity–pressure discretization.

3.2. Residual-based variational multiscale (RBVMS) formulation of flow over a moving domain

Following the developments in [17], the trial solution and weighting function spaces are split into coarse and fine scales as

$$\{\boldsymbol{v}, p\} = \{\boldsymbol{v}^h, p^h\} + \{\boldsymbol{v}', p'\}, \tag{4}$$

$$\{\boldsymbol{w}, q\} = \{\boldsymbol{w}^h, q^h\} + \{\boldsymbol{w}', q'\}. \tag{5}$$

In the above, the superscript h denotes resolved coarse scales represented by the finite element or isogeometric discretization. The primed quantities correspond to the unresolved scales and will be modeled. The above decomposition of the weighting functions leads to two variational sub-problems:

$$B(\{\boldsymbol{w}^h, q^h\}, \{\boldsymbol{v}^h, p^h\} + \{\boldsymbol{v}', p'\}) - (\boldsymbol{w}^h, \rho\boldsymbol{f})_\Omega = 0, \quad \forall \{\boldsymbol{w}^h, q^h\} \in \mathcal{W}^h, \tag{6}$$

$$B(\{\boldsymbol{w}', q'\}, \{\boldsymbol{v}^h, p^h\} + \{\boldsymbol{v}', p'\}) - (\boldsymbol{w}', \rho\boldsymbol{f})_\Omega = 0, \quad \forall \{\boldsymbol{w}', q'\} \in \mathcal{W}', \tag{7}$$

where $\mathcal{W}^h$ is the finite-dimensional space of finite element or isogeometric functions and $\mathcal{W}'$ is an infinite dimensional space of the unresolved fine scales. We proceed as in [17] and model the fine-scale velocity and pressure as being proportional to the strong form of the Navier–Stokes partial differential equation residual

$$\boldsymbol{v}' = -\tau_M \boldsymbol{r}_M(\{\boldsymbol{v}^h, p^h\}), \tag{8}$$

$$p' = -\tau_C r_C(\boldsymbol{v}^h), \tag{9}$$

$$\boldsymbol{r}_M(\{\boldsymbol{v}, p\}) = \rho\frac{\partial \boldsymbol{v}}{\partial t} + \rho(\boldsymbol{v} - \hat{\boldsymbol{v}}) \cdot \nabla\boldsymbol{v} + \nabla p - \mu\Delta\boldsymbol{v} - \rho\boldsymbol{f}, \tag{10}$$

$$r_C(\boldsymbol{v}) = \nabla \cdot \boldsymbol{v}. \tag{11}$$

The parameters τ_M and τ_C in the above equations originate from stabilized finite element methods for fluid dynamics (see, e.g., [34, 37, 67–74]) and are referred to as stabilization parameters. Recently, they were interpreted as the appropriate

averages of the small-scale Green's function, which is a key mathematical object in the theory of variational multiscale methods (see [75] for an elaboration). We note that in this paper the stabilization parameters are defined in a form where they have the units of time divided by density, instead of the units of time, which is more typical.

To generate the numerical method, $\boldsymbol{v}'$ and p' in Eqs. (8) and (9) are inserted directly into the coarse-scale equations given by Eq. (6). To simplify the formulation, two assumptions are typically employed: 1) the fine scales are orthogonal to the coarse scales with respect to the inner-product generated by the viscous term; and 2) the fine scales are quasi-static. The latter condition is expressed as

$$\frac{\partial \boldsymbol{v}'}{\partial t} = 0. \tag{12}$$

In this work, we would like to illustrate that the quasi-static assumption on the fine scales takes on a different form in the case of a moving fluid domain. In particular, we postulate the following generalized quasi-static condition on the fine scales

$$\frac{\partial \hat{J}\boldsymbol{v}'}{\partial t} = 0, \tag{13}$$

where $\hat{J}$ is the determinant Jacobian of the mapping from the referential to the spatial domain in the ALE framework (see, e.g., [31]), and the time derivative is taken with respect to the fixed referential coordinate. Differentiating the product and rearranging terms, gives

$$\frac{\partial \boldsymbol{v}'}{\partial t} = -(\nabla \cdot \hat{\boldsymbol{v}})\boldsymbol{v}'. \tag{14}$$

Note that in the case of divergence-free fluid domain motions (e.g., stationary or rotating domains), the above definition reverts to the original quasi-static subgrid scale assumption.

Examining the influence of the fine scales in the time derivative and convective terms, we obtain

$$\left(\boldsymbol{w}^h, \rho\frac{\partial(\boldsymbol{v}^h + \boldsymbol{v}')}{\partial t}\right)_\Omega = \left(\boldsymbol{w}^h, \rho\frac{\partial \boldsymbol{v}^h}{\partial t}\right)_\Omega - \left(\boldsymbol{w}^h, \rho(\nabla \cdot \hat{\boldsymbol{v}})\boldsymbol{v}'\right)_\Omega, \tag{15}$$

and

$$\begin{aligned}\left(\boldsymbol{w}^h, \rho(\boldsymbol{v}^h + \boldsymbol{v}' - \hat{\boldsymbol{v}}) \cdot \nabla(\boldsymbol{v}^h + \boldsymbol{v}')\right)_\Omega &= \left(\boldsymbol{w}^h, \rho(\boldsymbol{v}^h + \boldsymbol{v}' - \hat{\boldsymbol{v}}) \cdot \nabla \boldsymbol{v}^h\right)_\Omega \\ &\quad - \left(\nabla \boldsymbol{w}^h, \rho\boldsymbol{v}' \otimes (\boldsymbol{v}^h + \boldsymbol{v}' - \hat{\boldsymbol{v}})\right)_\Omega \\ &\quad - \left(\boldsymbol{w}^h, \rho\nabla \cdot (\boldsymbol{v}^h + \boldsymbol{v}' - \hat{\boldsymbol{v}})\boldsymbol{v}'\right)_\Omega,\end{aligned} \tag{16}$$

respectively. Assuming incompressibility of the particle velocity field (i.e., $\nabla \cdot (\boldsymbol{v}^h + \boldsymbol{v}') = 0$) and rearranging the terms in Eq. (16), we obtain

$$\begin{aligned}\left(\boldsymbol{w}^h, \rho(\boldsymbol{v}^h + \boldsymbol{v}' - \hat{\boldsymbol{v}}) \cdot \nabla(\boldsymbol{v}^h + \boldsymbol{v}')\right)_\Omega &= \left(\boldsymbol{w}^h, \rho(\boldsymbol{v}^h - \hat{\boldsymbol{v}}) \cdot \nabla \boldsymbol{v}^h\right)_\Omega \\ &- \left(\nabla \boldsymbol{w}^h, \rho \boldsymbol{v}' \otimes (\boldsymbol{v}^h - \hat{\boldsymbol{v}})\right)_\Omega \\ &+ \left(\boldsymbol{w}^h, \rho \boldsymbol{v}' \cdot \nabla \boldsymbol{v}^h\right)_\Omega \\ &- \left(\nabla \boldsymbol{w}^h, \rho \boldsymbol{v}' \otimes \boldsymbol{v}'\right)_\Omega \\ &+ \left(\boldsymbol{w}^h, \rho(\nabla \cdot \hat{\boldsymbol{v}}) \boldsymbol{v}'\right)_\Omega . \end{aligned} \tag{17}$$

The above development leads to the following semi-discrete variational formulation: find $\{\boldsymbol{v}^h, p^h\} \in \mathcal{V}^h,\ \boldsymbol{v}^h = \boldsymbol{g}$ on Γ_g, such that $\forall \{\boldsymbol{w}^h, q^h\} \in \mathcal{V}^h,\ \boldsymbol{w}^h = \boldsymbol{0}$ on Γ_g,

$$B(\{\boldsymbol{w}^h, q^h\}, \{\boldsymbol{v}^h, p^h\}) + B_{vms}(\{\boldsymbol{w}^h, q^h\}, \{\boldsymbol{v}^h, p^h\}) - (\boldsymbol{w}^h, \rho \boldsymbol{f})_\Omega = 0, \tag{18}$$

where the modeled subgrid-scale terms in B_{vms} are

$$\begin{aligned} B_{vms}(\{\boldsymbol{w}, q\}, \{\boldsymbol{v}, p\}) &= (\nabla \boldsymbol{w}, \rho \tau_M \boldsymbol{r}_M(\{\boldsymbol{v}, p\}) \otimes (\boldsymbol{v} - \hat{\boldsymbol{v}}))_\Omega \\ &+ (\nabla \cdot \boldsymbol{w}, \tau_C \nabla \cdot \boldsymbol{v})_\Omega \\ &- (\boldsymbol{w}, \rho \tau_M \boldsymbol{r}_M(\{\boldsymbol{v}, p\}) \cdot \nabla \boldsymbol{v})_\Omega \\ &- (\nabla \boldsymbol{w}, \rho \tau_M \boldsymbol{r}_M(\{\boldsymbol{v}, p\}) \otimes \tau_M \boldsymbol{r}_M(\{\boldsymbol{v}, p\}))_\Omega \\ &+ (\nabla q, \tau_M \boldsymbol{r}_M(\{\boldsymbol{v}, p\}))_\Omega , \end{aligned} \tag{19}$$

and the integrals are taken element-wise.

Remark: Comparing Eqs. (15) and (17) we note that the last term on the right-hand-side of Eq. (17) cancels that of Eq. (15). This leads to the discrete formulation given by Eq. (18) that globally conserves linear momentum (see [31] for a derivation). This would not be the case if the quasi-static assumption given by Eq. (12) is employed.

Remark: The above discrete formulation may be augmented with weakly-enforced Dirichlet boundary conditions [76] to enhance solution accuracy in the presence of unresolved boundary layers.

Remark: A similar approach for deriving the multiscale formulation of the Navier–Stokes equations on a moving domain was presented in [77]. However, a quasi-static fine scale assumption given by Eq. (12) was used instead of Eq. (14).

4. Application to Turbulent Rotating Flow: Taylor–Couette Flow at $Re = 8000$

We simulate the turbulent Taylor–Couette flow, which is a flow between two concentric cylinders. We feel that this test case is relevant to wind turbines, because

the flow exhibits several complex features in common with the present application, such as rotation, curved walls, boundary layers, and highly complex time-dependent evolution of the velocity and pressure fields, all of which are challenging to compute. At the same time, much is known about this problem experimentally (see, e.g., [78]), and numerically (see, e.g., [22, 79–83]), which makes this test case a very good candidate for verification and validation of numerical formulations for turbulent fluid flow.

One of the bigger challenges involved in simulation of rotating turbulent flow is turbulence modeling. Eddy-viscosity models rely on the continuous energy cascade, i.e., the continuous energy transfer from low to high modes in the turbulent flow (see, e.g., [84]). However, as explained in [85], in the presence of high rotation rates the energy cascade is arrested. This renders most well-accepted eddy-viscosity models inconsistent in the rotation dominated regime. The advantage of the present turbulence modeling approach is that the unresolved scales are modeled using the variational multiscale methodology, and the explicit use of eddy viscosities is avoided all together. As a result, the present approach is not likely to suffer from the shortcomings of eddy-viscosity-based turbulence models for flows with high rotation rates.

The problem setup is as follows. The inner cylinder rotates at a constant angular velocity, while the outer cylinder is stationary (see Figure 7a). Periodic boundary conditions are imposed in the axial direction, and no-slip boundary conditions are applied at the cylinder surfaces. The no-slip conditions are imposed weakly using the methodology presented in [19]. The Reynolds number of the flow based on the speed of the inner cylinder and the gap between the cylinders is $Re = U_\theta(R_1 - R_0)\rho/\mu = 8000$. The fluid density is set to $\rho = 1$, the dynamic viscosity is set to $\mu = 1/8000$, the inner and outer radii are set to $R_0 = 1$ and $R_1 = 2$, respectively, and the inner cylinder wall speed is set to $U_\theta = 1$. Due to the rotational symmetry of the problem geometry, the simulations are performed over a stationary domain. The results of our simulation are compared with the DNS of [86], where this test case was computed using 256 Fourier modes in the axial direction and 400 9^{th}-order spectral finite elements in the remaining directions. This is the highest Reynolds number DNS available for this flow. For an overview of spectral finite elements applied to fluid dynamics, see, for example, [87].

We solve the problem on meshes consisting of linear finite elements and quadratic NURBS. The quadratic NURBS give exact representation of the geometry, while linear FEM only approximates it. However, the geometry approximation improves with refinement. We deliberately sacrifice the boundary layer resolution by using uniform meshes in all directions, including the wall-normal. Our goal is to see the performance of the proposed method in the case of unresolved boundary layers and rotational dominance, as well as to assess the accuracy benefits of the NURBS-based approach relative to standard FEM for this challenging situation. We note that two of the authors have successfully computed this test case using quadratic NURBS with boundary layer refinement in [22].

The simulation is started using a linear laminar profile of the azimuthal velocity. After transition to turbulence, the flow is advanced in time until the solution reaches a statistically-stationary state. Figure 7b shows the isosurfaces of a scalar quantity Q, which is defined as

$$Q = \mathbf{\Omega} : \mathbf{\Omega} - \nabla^s \boldsymbol{v}^h : \nabla^s \boldsymbol{v}^h, \tag{20}$$

where $\mathbf{\Omega} = \frac{1}{2}(\nabla \boldsymbol{v}^h - \nabla \boldsymbol{v}^{h^T})$ is the skew-symmetric part of the velocity gradient. Q is designed to be an objective quantity for identifying vortical features in the turbulent flow (see, e.g., [88]). The figure illustrates the complexity of the turbulent flow and the high demand on the numerical method to adequately represent it.

Flow statistics are given in the form of mean azimuthal velocity and angular momentum, both as functions of the radial coordinate. The velocity data is collected at mesh knots, rotated to the cylindrical coordinate system, and ensemble-averaged over the periodic directions and time.

Figure 8 shows mean azimuthal velocity results and compares linear FEM and quadratic NURBS. Linear elements give a reasonable approximation of the mean flow, especially considering the few degrees of freedom employed and no boundary layer resolution. The quadratic NURBS, which use nearly the same number of degrees of freedom as linear elements, produces results of remarkable accuracy. Even with the coarse mesh, for which the boundary layer lies completely within the first element in the wall-normal direction, the mean velocity curve nearly interpolates the DNS. The same conclusions hold for the mean angular momentum results shown in Figure 9.

This problem illustrates that the variational multiscale method in conjunction with NURBS-based isogeometric analysis can deliver very accurate results for rotating turbulent flow. In what follows, we apply the proposed method to wind turbine simulation. We are cognizant of the fact that the Reynolds number in the case of wind turbines is orders of magnitude higher than for the test case considered in this section. Hence, the results presented in this section constitute only partial validation of the proposed computational method.

5. Wind Turbine Simulations

5.1. Coupling of a rigid rotor with incompressible flow

In this section we present the formulation of the coupled problem that involves incompressible fluid and a rigid body that undergoes rotational motion. We specialize the formulation to the case of a wind turbine rotor whose axis of rotation coincides with the z-direction. We denote by θ the angle of rotation of the wind turbine rotor about the z-axis from the reference configuration. The rotor angular velocity and acceleration are found by differentiating θ with respect to time, and are denoted by $\dot{\theta}$ and $\ddot{\theta}$, respectively. The coupled problem amounts to finding the triple

$\{\boldsymbol{v}^h, p^h, \theta\}$ such that for all test functions $\{\boldsymbol{w}^h, q^h\}$

$$B(\{\boldsymbol{w}^h, q^h\}, \{\boldsymbol{v}^h, p^h\}) + B_{vms}(\{\boldsymbol{w}^h, q^h\}, \{\boldsymbol{v}^h, p^h\}) - (\boldsymbol{w}^h, \rho \boldsymbol{f})_\Omega = 0, \tag{21}$$

$$I\ddot{\theta} + f(\dot{\theta}, t) - T_f(\boldsymbol{v}^h, p^h) = 0, \tag{22}$$

where I is the rotor moment of inertia, $f(\dot{\theta}, t)$ may represent forcing due to friction in the system or interaction with the generator, and $T_f(\boldsymbol{v}^h, p^h)$ is the torque exerted on the rotor by the fluid, i.e. the aerodynamic torque. This is a two-way coupled problem: the fluid solution depends on the rotor position and speed, while the rotor equation is driven by the fluid torque.

The fluid domain rotates with the same angular speed as the rotor. This means the fluid domain current configuration is defined as

$$\Omega = \{\boldsymbol{x} \mid \boldsymbol{x} = \boldsymbol{R}(\theta)[\boldsymbol{X} - \boldsymbol{X}_0] + \boldsymbol{X}_0,\ \boldsymbol{X} \in \Omega_0\}, \tag{23}$$

where Ω_0 is the fluid domain in the reference configuration, $\boldsymbol{X}_0$ is the rotor center of mass, and $\boldsymbol{R}(\theta)$ is the rotation matrix given by

$$\boldsymbol{R}(\theta) = \begin{bmatrix} \cos\theta & -\sin\theta & 0 \\ \sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{bmatrix}. \tag{24}$$

With the above definition of the fluid domain motion, the fluid domain velocity becomes

$$\hat{\boldsymbol{v}} = \frac{d\boldsymbol{R}(\theta)}{d\theta}[\boldsymbol{X} - \boldsymbol{X}_0]\ \dot{\theta}. \tag{25}$$

The aerodynamic torque is given by

$$T_f(\boldsymbol{v}, p) = \int_{\Gamma_{rot}} \boldsymbol{r} \times (p\boldsymbol{n} - 2\mu\nabla^s \boldsymbol{v} \cdot \boldsymbol{n}) d\Gamma, \tag{26}$$

where Γ_{rot} is the rotor surface, $\boldsymbol{r} = \boldsymbol{x} - \boldsymbol{x}_0$ is the radius-vector from the rotor center of mass in the current configuration, and $\boldsymbol{n}$ is the unit normal to the rotor surface. In the computations, instead of using Eq. (26) directly, we opt for a conservative definition of the aerodynamic torque. For this, we first define the discrete momentum equation residual at every finite element node or isogeometric control point A and cartesian direction i as

$$R_{A,i} = B(\{N_A \boldsymbol{e}_i, 0\}, \{\boldsymbol{v}^h, p^h\}) + B_{vms}(\{N_A \boldsymbol{e}_i, 0\}, \{\boldsymbol{v}^h, p^h\}) - (N_A \boldsymbol{e}_i, \rho \boldsymbol{f})_\Omega, \tag{27}$$

where N_A's are the velocity basis functions and $\boldsymbol{e}_i$'s are the Cartesian basis vectors. The residual $R_{A,i}$ vanishes everywhere in the domain except at the rotor surface control points (or nodes). The conservative aerodynamic torque along the z-axis

may now be computed as

$$T_f(\boldsymbol{v}^h, p^h) = \sum_{A,j,k} \epsilon_{3jk} r_{A,j} R_{A,k}, \tag{28}$$

where ϵ_{ijk}'s are the cartesian components of the alternator tensor and $r_{A,j}$ are the nodal or control point coordinates.

The coupled system given by Eqs. (21)–(22) is advanced in time using the Generalized-alpha method (see [31, 89, 90]) for both the fluid and rotor equations. Within each time step, the coupled equations are solved using an inexact Newton approach. For every Newton iteration we obtain the fluid solution increment, update the fluid solution, compute the aerodynamic torque, obtain the rotor solution increment, and update the rotor solution and the fluid domain velocity and position. The iteration is repeated until convergence. Because the wind turbine rotor is a relatively heavy structure, the proposed approach, also referred to as "block-iterative" (see [44] for the terminology), is stable.

Remark: The case of prescribed angular velocity is also of interest and can be obtained by providing a time history of θ, which obviates Eq. (22). We make use of this in the sequel, where we perform a validation study of our wind turbine rotor configuration.

Remark: We note that the torque definition given by Eq. (28) makes use of the isoparametric property of the finite element and isogeometric methods. Besides the possible accuracy benefits associated with a conservative definition of torque, the formulation avoids performing boundary integration and makes use of the same "right-hand-side" routine as the fluid assembly itself, which simplifies implementation.

Remark: The conservative torque definition given by Eq. (28) is only applicable for strongly enforced velocity boundary conditions, which are employed in this work for wind turbine simulations. For the case of weak boundary conditions, a conservative definition of the torque given in [22] may be used instead.

5.2. Rotationally-periodic boundary conditions

To enhance the efficiency of the simulations, we take advantage of the problem rotational symmetry and only construct a 120° slice of the computational domain (see Section 2). However, to carry out the simulations, rotationally-periodic boundary conditions must be imposed. Denoting by $\boldsymbol{v}_l$ and $\boldsymbol{v}_r$ the fluid velocities at the left and right boundary, respectively (see Figure 10), and by p_l and p_r the corresponding pressures, we set

$$p_l = p_r, \tag{29}$$

$$\boldsymbol{v}_l = \boldsymbol{R}(2/3\pi)\, \boldsymbol{v}_r. \tag{30}$$

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 with the rotation matrix from Eq. (24) evaluated at $\theta = 2/3\pi$. Note that the transformation matrix is independent of the current domain position.

Rotationally periodic boundary conditions are implemented through standard master-slave relationships. During the discrete residual assembly (see Eq. (27)), the nodal or control-point momentum residual at the slave boundary is rotated using $\boldsymbol{R}^T(2/3\pi)$ and added to the master residual (assumed to reside at the right boundary). Once the fluid solution update is computed, the velocity solution at the master boundary is rotated using $\boldsymbol{R}(2/3\pi)$ prior to communicating it to the slave boundary. All of the numerical examples presented in this article use rotationally-periodic boundary conditions. We note that rotationally periodic boundary conditions were employed earlier in [54, 56, 91] for parachute simulations.

5.3. Simulations with prescribed rotor speed: A validation study

We compute the aerodynamics of the NREL 5MW offshore baseline wind turbine [61] with prescribed rotor shape and speed with a rotating mesh. The rotor radius, R, is 63 m. 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 [92] 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 of 4.67×10^{-4} s. Both NURBS and tetrahedral finite element simulations of this setup are performed. The time step size was originally set in computational units (non-dimensional) to 0.01 based on the Courant number considerations. Using a significantly larger time step size, such as 5 times larger, results in solutions which we do not find accurate.

The chosen wind velocity and rotor speed correspond to one of the cases given in [61], where the aerodynamics simulations were performed using FAST [64]. 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 [61] 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 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 [61]. Nevertheless, we feel that this value of the aerodynamic torque is close to what it is in reality, given the vast experience of NREL with wind turbine rotor simulations employing FAST.

5.3.1. NURBS-based simulation

The problem setup and the details of the computational domain are shown in Figures 5 and 6. The mesh of the 120° slice of the domain is comprised of 1,449,000 quadratic NURBS elements, which yields about the same number of mesh control points (analogues of nodes in finite elements). The computation is carried out on 240 processors on Ranger, a Sun Constellation Linux Cluster at the Texas Advanced Computing Center [93] 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 [94] 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 11 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 resultion for accurate representation. Figure 12 shows the air pressure contours at the suction side (i.e., the back side) of the wind turbine blade. It is precisely the large negative pressure that creates the desired lift. The fluid traction vector projected to the plane of rotation is shown in Figure 13. The traction vector points in the direction of rotation and grows in magnitude toward the blade tip, creating favorable aerodynamic torque. However, at the blade tip the traction vector rapidly decays to zero and even changes sign, which introduces a small amount of inefficiency. The time history of the aerodynamic torque is shown in Figure 14, where the steady-state result from [61] 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 simulate 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 by the rotor for these wind conditions (based on our simulations) is

$$P = T_f \dot{\theta} \approx 2.88\text{MW}. \tag{31}$$

According to Betz law (see, e.g., [62]), the maximum power that a horizontal axis wind turbine is able to extract from the wind is

$$P_{max} = \frac{16}{27} \frac{\rho A \|\boldsymbol{v}_{in}\|^3}{2} \approx 3.23\text{MW}, \tag{32}$$

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

$$\frac{P}{P_{max}} \approx 89\%, \tag{33}$$

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. To better resolve the torque gradient at the blade root and tip, Patch 1 has 0.366 the span length of the middle patches, and Patches 2–4 and 16–18 have $\frac{2}{3}$. The patch-wise torque distribution is shown in Figure 15. 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 16, where the axial component of the flow velocity is plotted 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.

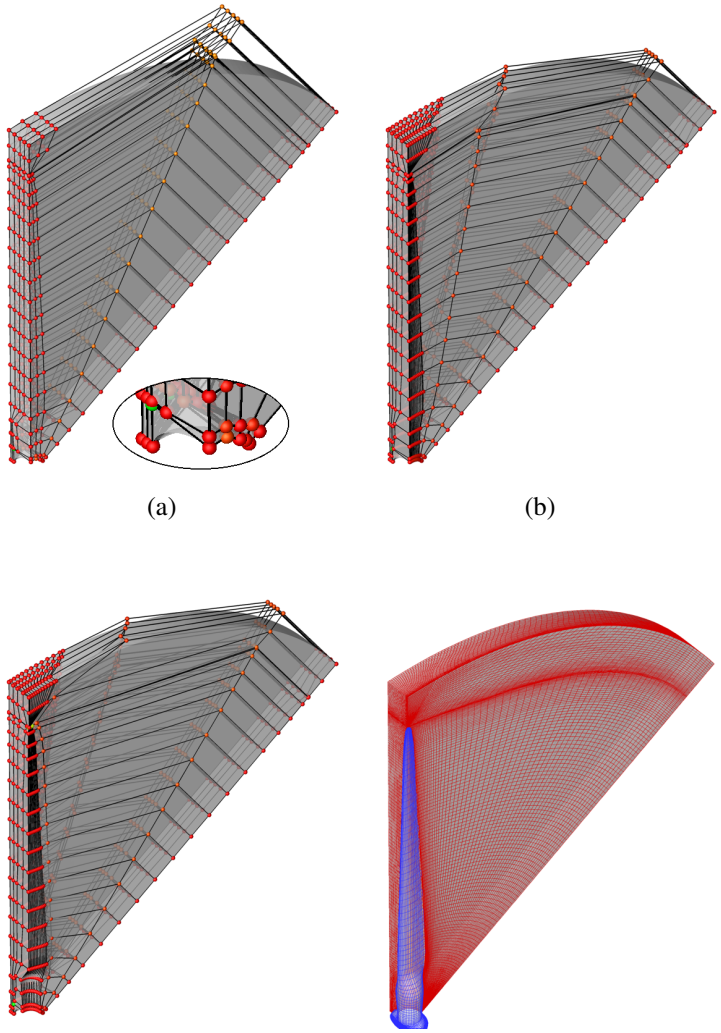
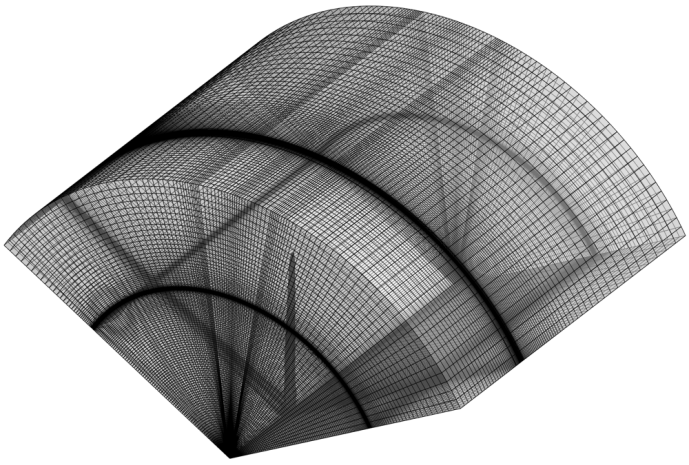
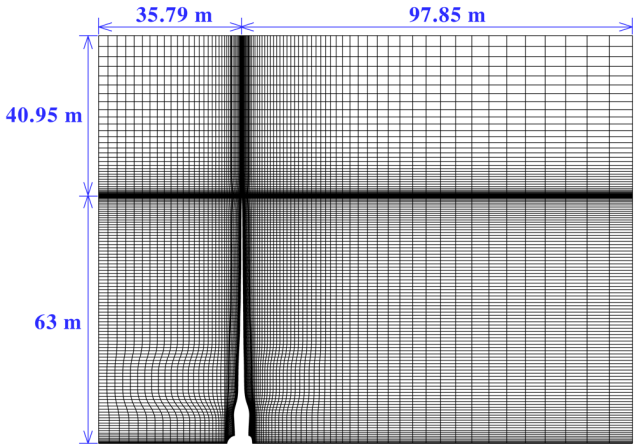


Figure 4: Stages of analysis-suitable geometry construction for wind turbine rotor simulations.

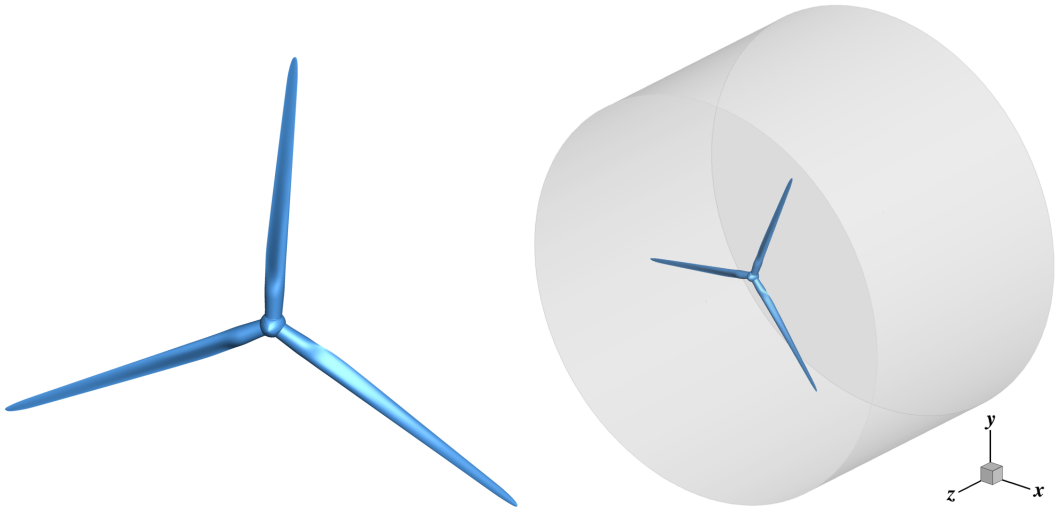


(a)



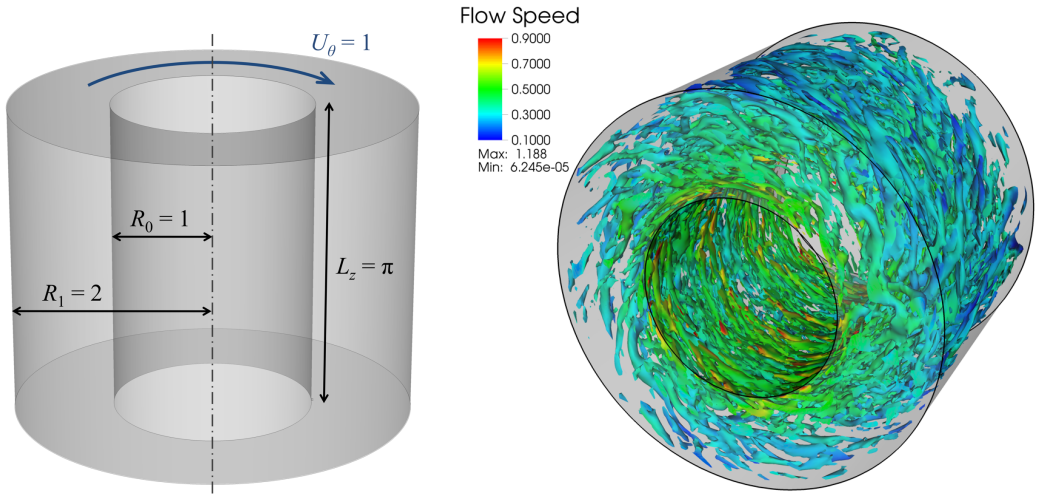
(b)

Figure 5: (a) Volumetric NURBS mesh of the computational domain. (b) A planar cut to illustrate mesh grading toward the rotor blade.



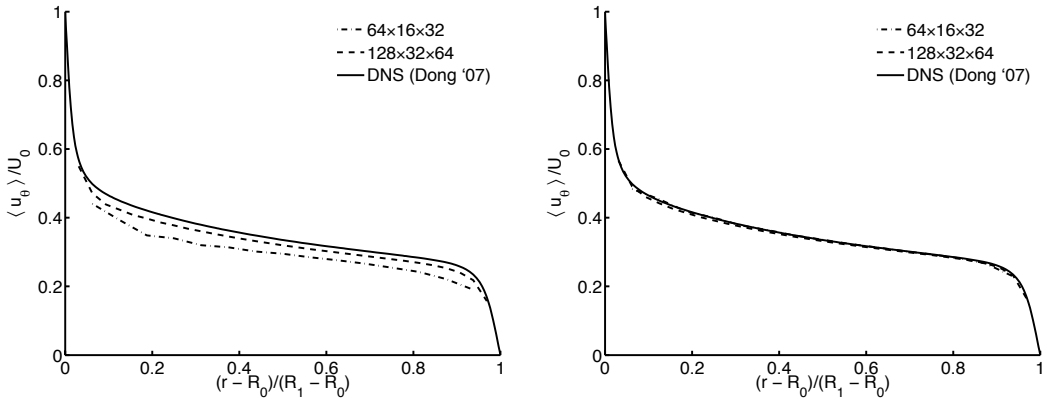
(a) (b)

Figure 6: (a) Wind turbine rotor surface. (b) Full problem domain. Both are obtained by merging the 120° rotationally-periodic domains.



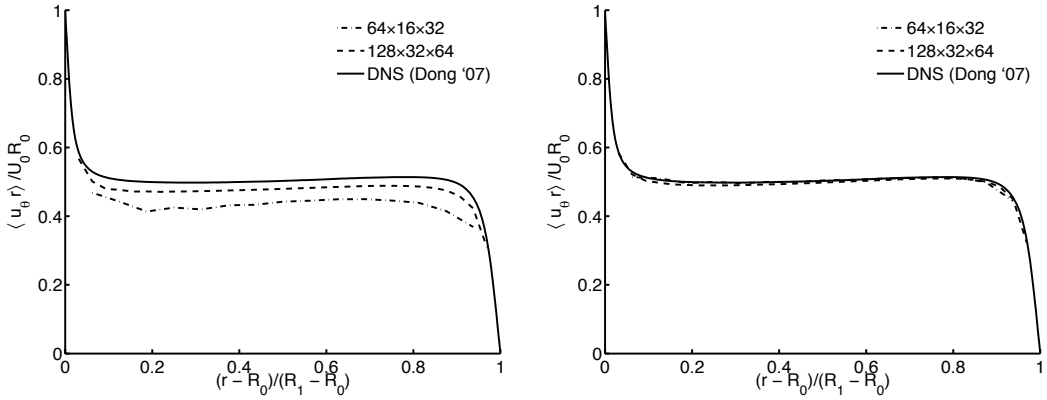
(a) (b)

Figure 7: Turbulent Taylor–Couette flow. (a) Problem setup. (b) Instantaneous flow speed. Isosurfaces of Q from Eq. (20) colored by flow speed.



(a) Linear finite elements (b) Quadratic NURBS

Figure 8: Turbulent Taylor–Couette flow. Mean azimuthal velocity.



(a) Linear finite elements (b) Quadratic NURBS

Figure 9: Turbulent Taylor–Couette flow. Mean angular momentum.

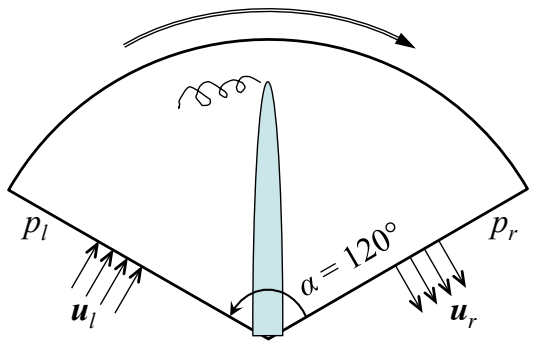


Figure 10: Rotationally-periodic boundary conditions.

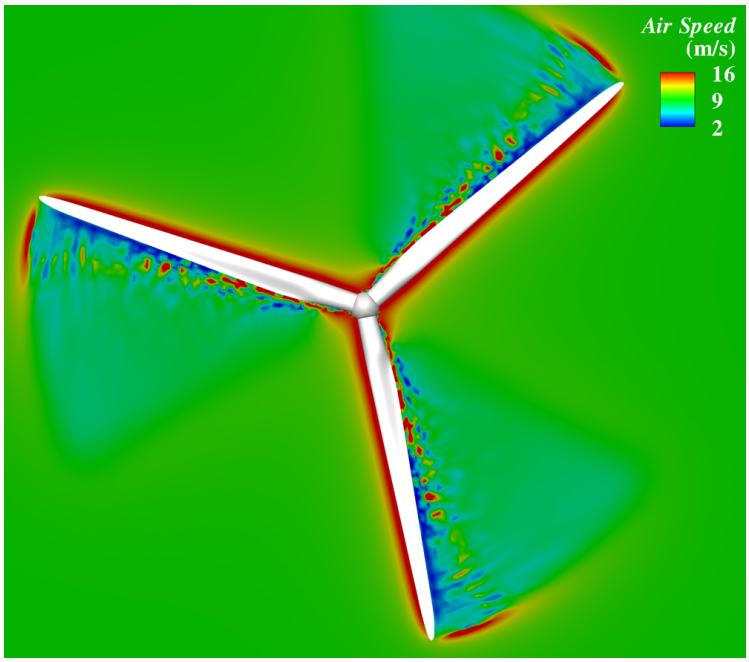


Figure 11: Air speed at $t = 0.8$ s

.

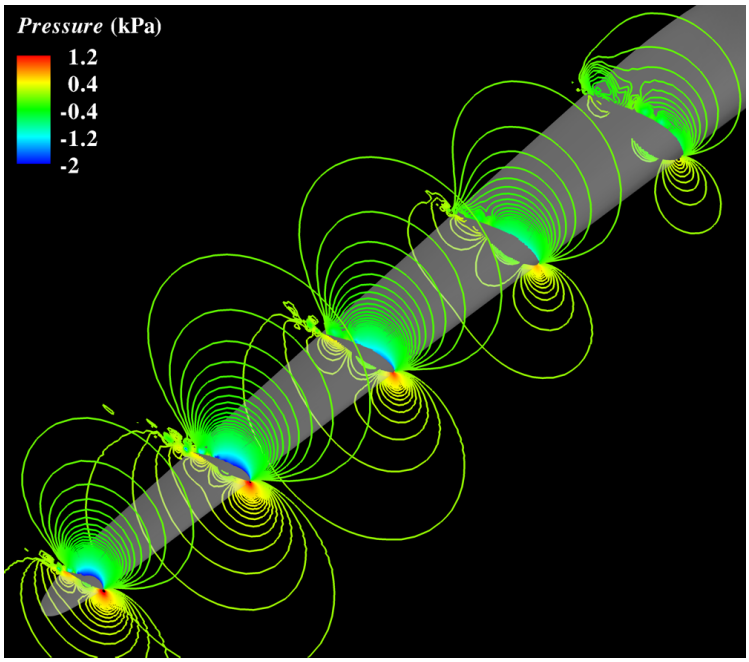


Figure 12: Pressure contours at several blade cross-sections at $t = 0.8$ s viewed from the back of the blade. The large negative pressure at the suction side of the airfoil creates the desired lift.

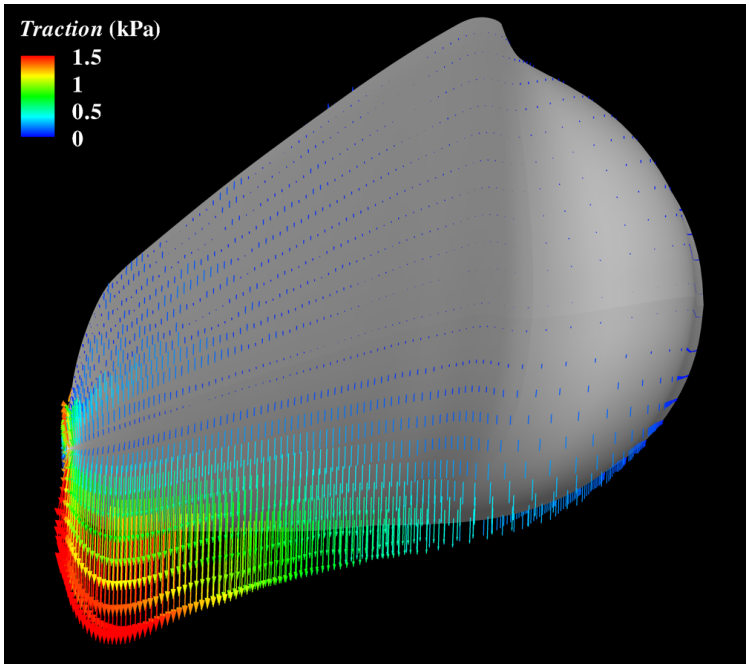


Figure 13: Fluid traction vector colored by magnitude at $t = 0.8$ s viewed from the back of the blade. The traction vector is projected to the rotor plane and illustrates the mechanism by which the favorable aerodynamic torque is created.

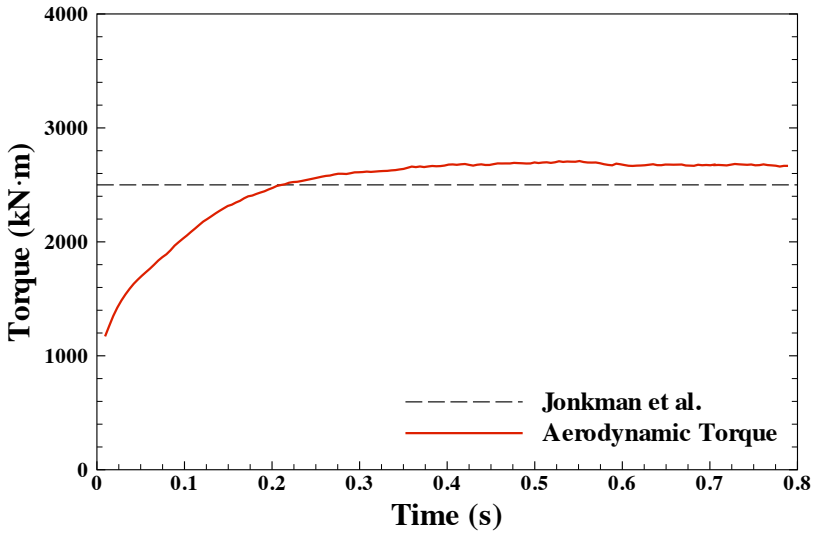


Figure 14: Time history of the aerodynamic torque. Statistically-stationary torque is attained in less than 0.8 s. The reference steady-state result from [61] is also shown for comparison.

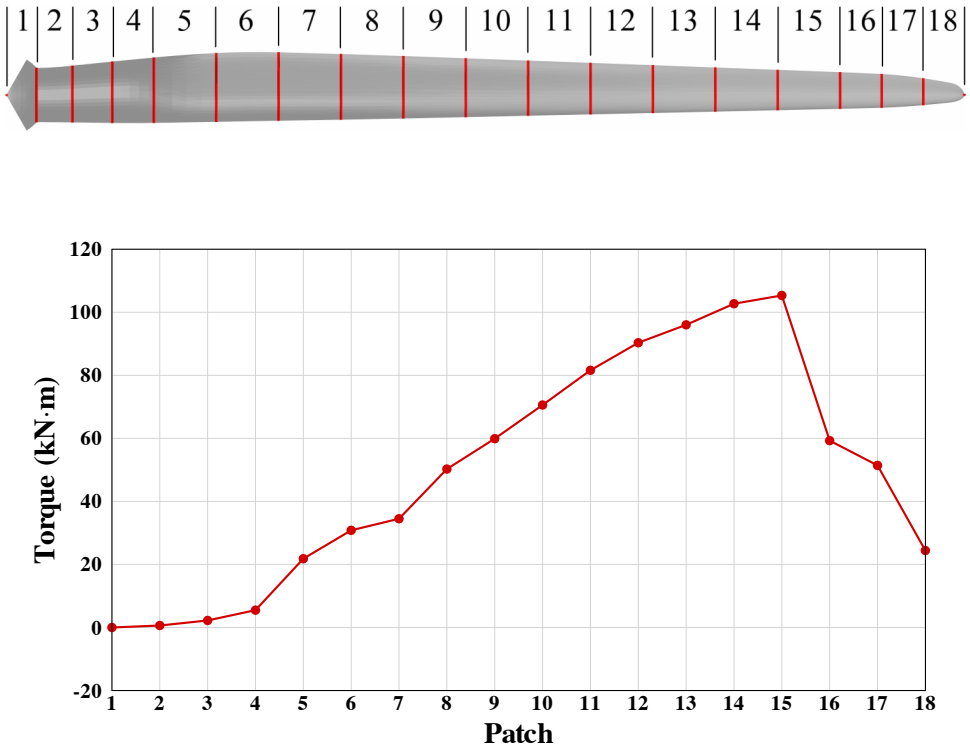


Figure 15: Patches along the blade (top) and the aerodynamic torque contribution from each patch (bottom) at t = 0.8 s.

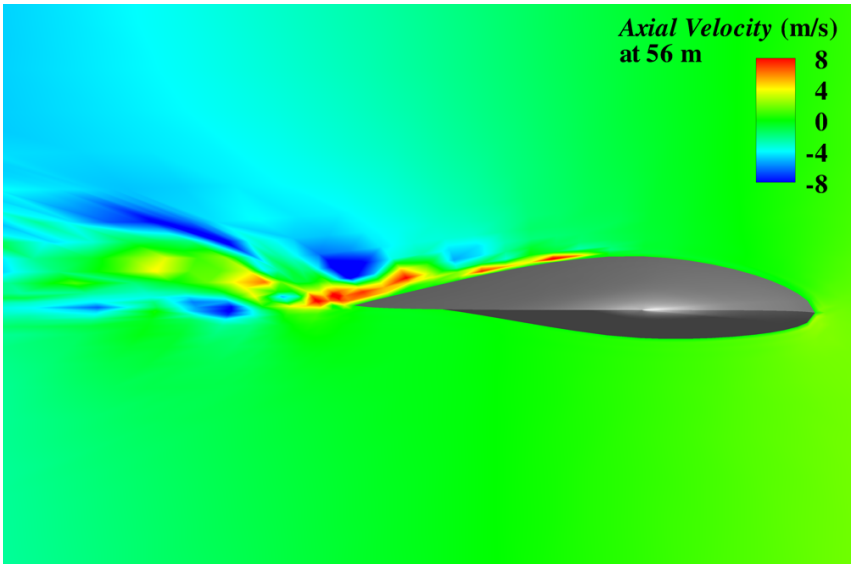


Figure 16: Axial flow velocity over the blade cross-section at 56 m at t = 0.8 s. The level of axial flow in the boundary layer is significant, which illustrates the importance of 3D computational modeling.

5.3.2. FEM-based simulation

As an added benefit, the isogeometric representation of the analysis geometry may be used to create a mesh for standard finite element analysis. To generate the triangular mesh on the rotor surface, we started with a quadrilateral surface mesh generated by interpolating the NURBS mesh at knots. We then subdivided each quadrilateral element into two triangles and then made some minor modifications to improve the mesh quality near the hub. The fluid domain tetrahedral mesh was then generated from the rotor surface triangulation. The computational domain, which

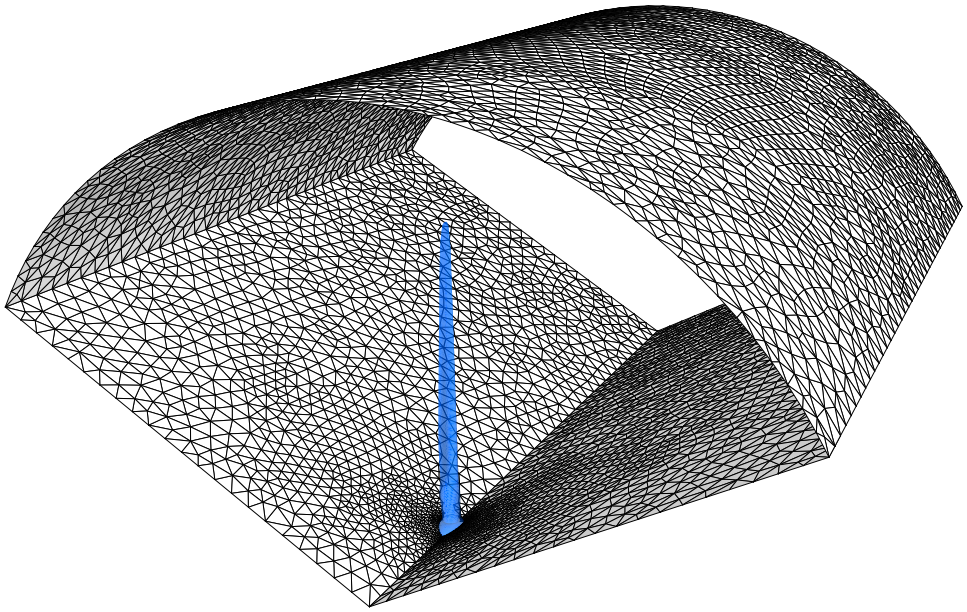


Figure 17: Rotationally-periodic domain with wind turbine blade shown in blue.

includes only one of the three blades, is shown in Figure 17. The inflow, outflow and radial boundaries lie $0.5R$, $2R$ and $1.43R$ from the hub center, respectively. This can be more easily seen in Figure 18, where the inflow, outflow, and radial boundaries are the left, right and top edges, respectively, of the cut plane along the rotation axis. We use the DSD/SST-TIP1 technique [38], with the SUPG test function option WTSA (see Remark 2 in [38]), with the "LSIC" stabilization dropped. The stabilization parameters used are those given in [38] by Eqs. (7)–(12). In solving the linear equation systems involved at every nonlinear iteration, the GMRES search technique 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 [95]. The fluid volume mesh consists of 241,609 nodes and 1,416,782 four-node tetrahedral elements, with 3,862 nodes and 7,680 triangles on the rotor surface. Each periodic boundary contains 1,430 nodes and 2,697 triangles. Near the rotor surface, we have 15 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

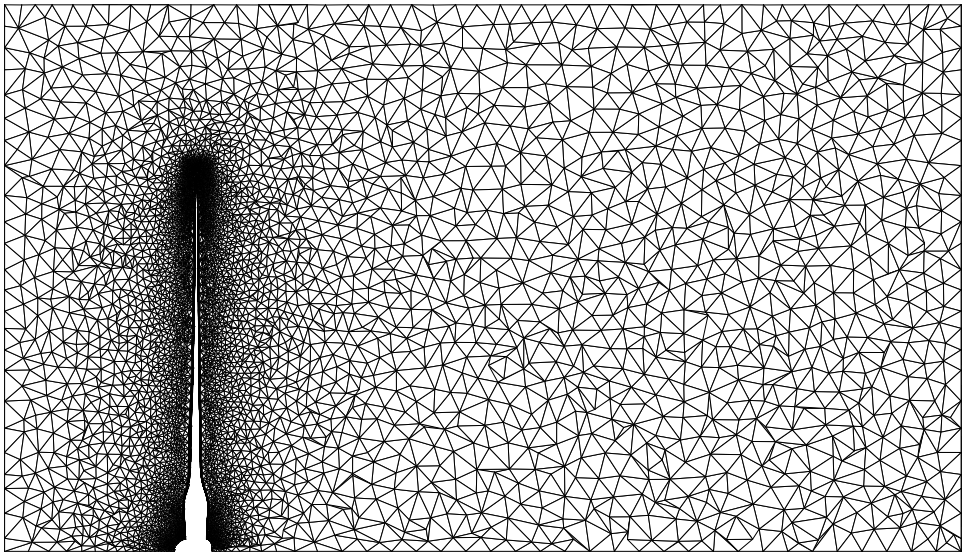


Figure 18: Cut plane of fluid volume mesh along rotor axis.

Figure 19. The number of nonlinear iterations per time step is 5, and the number of

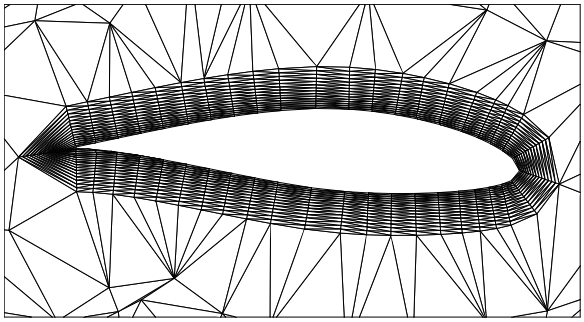


Figure 19: Boundary layer mesh at $\frac{3}{4}R$.

GMRES iterations per nonlinear iteration is 500.

Figure 20 shows, at 0.8 s, velocity magnitudes at the rotor plane of a whole domain formed by merging three periodic-domain solutions. The total aerodynamic torque is 1,701 kN m at 0.2 s and 821 kN m at 0.8 s, increases back to a level of 922 kN m at 0.887 s, and exhibits fluctuations in time. While in the earlier stages of the computation the torque level is more favorable, as the computation proceeds, it becomes less favorable because of flow separation, which should not occur for this wind turbine at the specified flow regime, becomes more widespread. We believe possible sources of flow separation are lack of sufficient mesh resolution near the blade surface, sudden increase in element size outside the boundary layer mesh, and lack of turbulence modeling. The patches and the torque contribution from each patch for a single blade are shown in Figure 21 at 0.2 s and 0.8 s. The torques for Patches 1–3 are approximately zero since these correspond to the hub

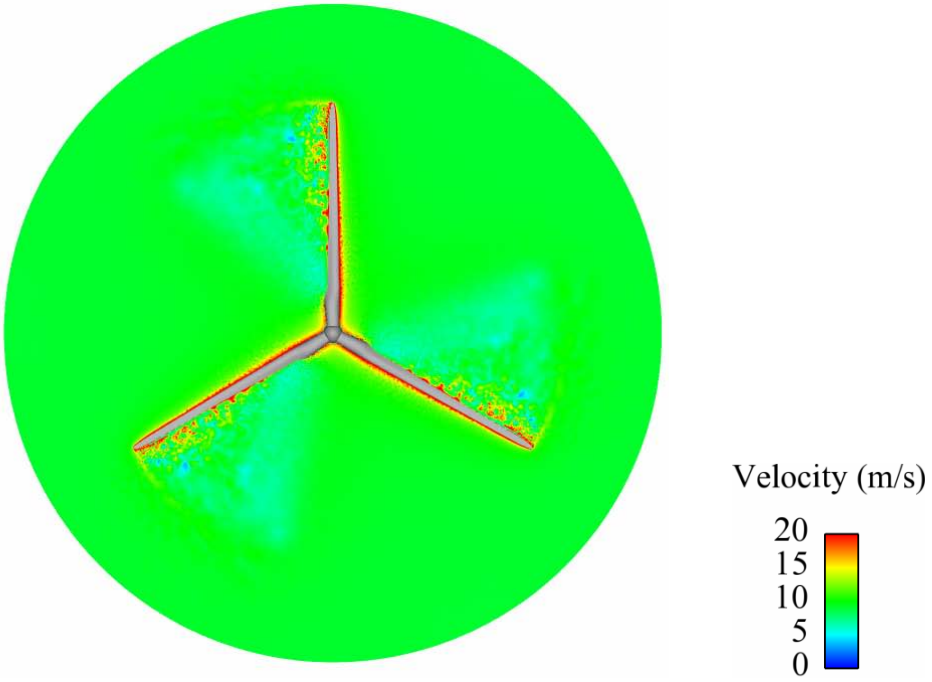


Figure 20: Velocity magnitude (ground reference frame) at 0.8 s at the rotor plane of a merged whole domain. High-speed flow trailing from the blade tips is readily observed. We note that the rotor plane velocities range up to 70 m/s but the coloring range is only up to 20 m/s to provide more contrast.

and cylindrical blade sections. The velocity and pressure fields at $\frac{3}{4}R$ ($\frac{1}{4}$ patch length down-span from Patch 13/14 border) are shown in Figures 22 and 23 at 0.2 s and 0.8 s. The velocity and pressure fields at the center of Patch 17 (approximately $\frac{15}{16}R$) are shown in Figures 24 and 25 at 0.2 s and 0.8 s. Possible solutions we have in mind to control the flow separation include more mesh refinement in regions just outside the boundary layer mesh and using, as a quick, simple attempt, the Discontinuity-Capturing Directional Dissipation (DCDD) [37]. The DCDD was shown in [96] to function like the Smagorinsky turbulence model [97] and was used in [98] for computation of air circulation problems with thermal coupling.

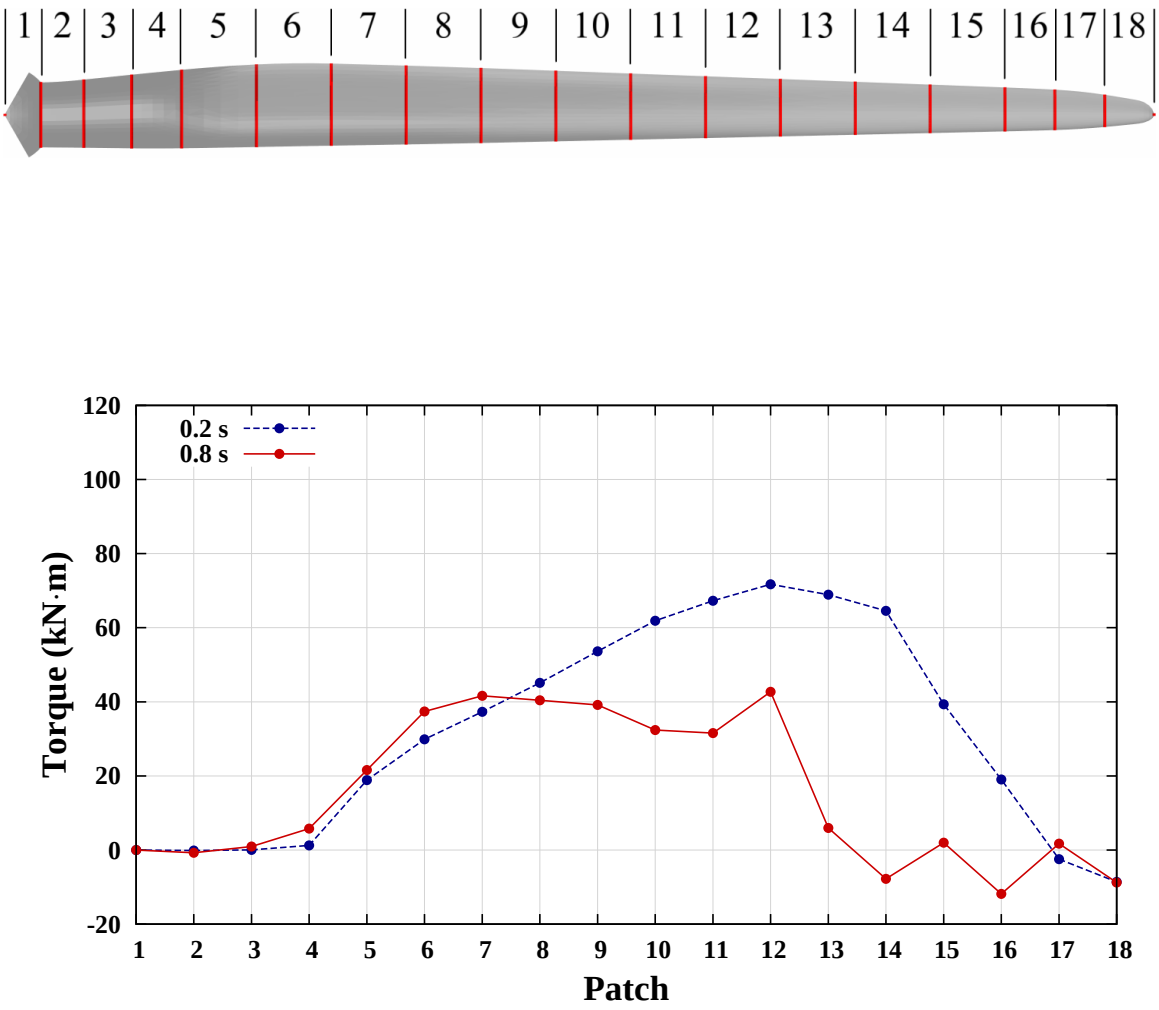


Figure 21: Patches along the blade (top) and the aerodynamic torque contribution from each patch at 0.2 s and 0.8 s (bottom).

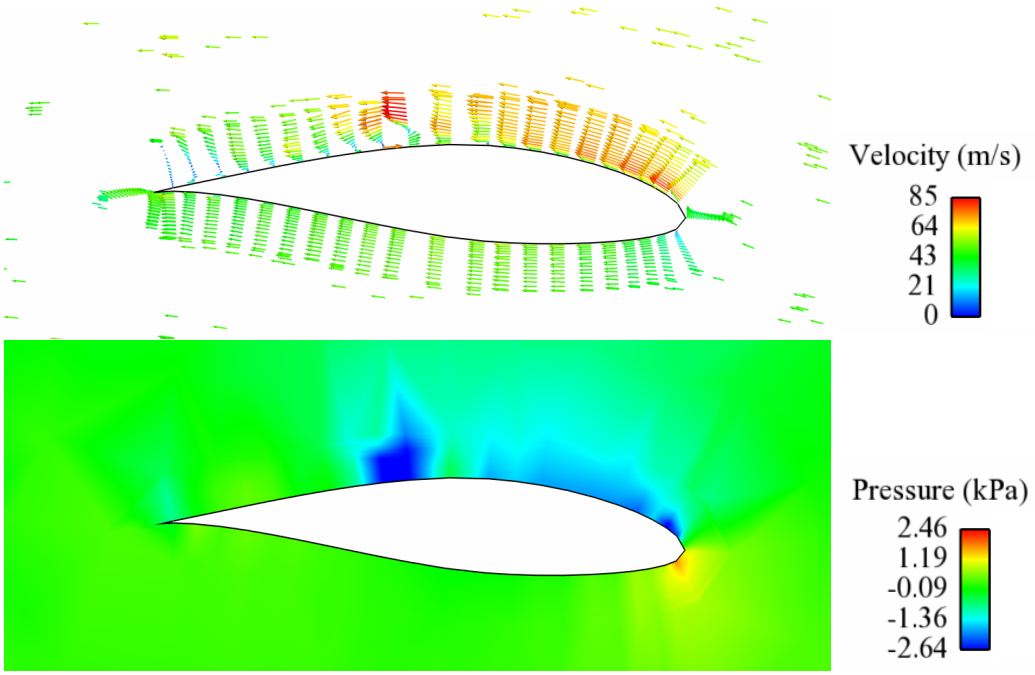


Figure 22: Velocity vectors in blade reference frame (top) and pressure field (bottom) for airfoil section at $\frac{3}{4}R$ at 0.2 s.

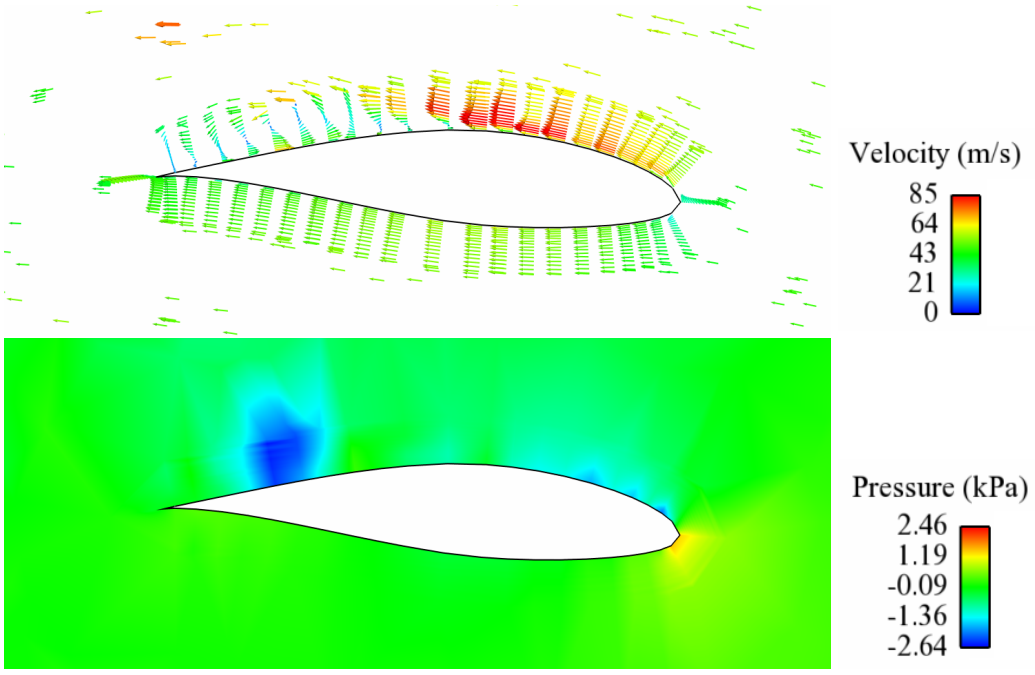


Figure 23: Velocity vectors in blade reference frame (top) and pressure field (bottom) for airfoil section at $\frac{3}{4}R$ at 0.8 s.

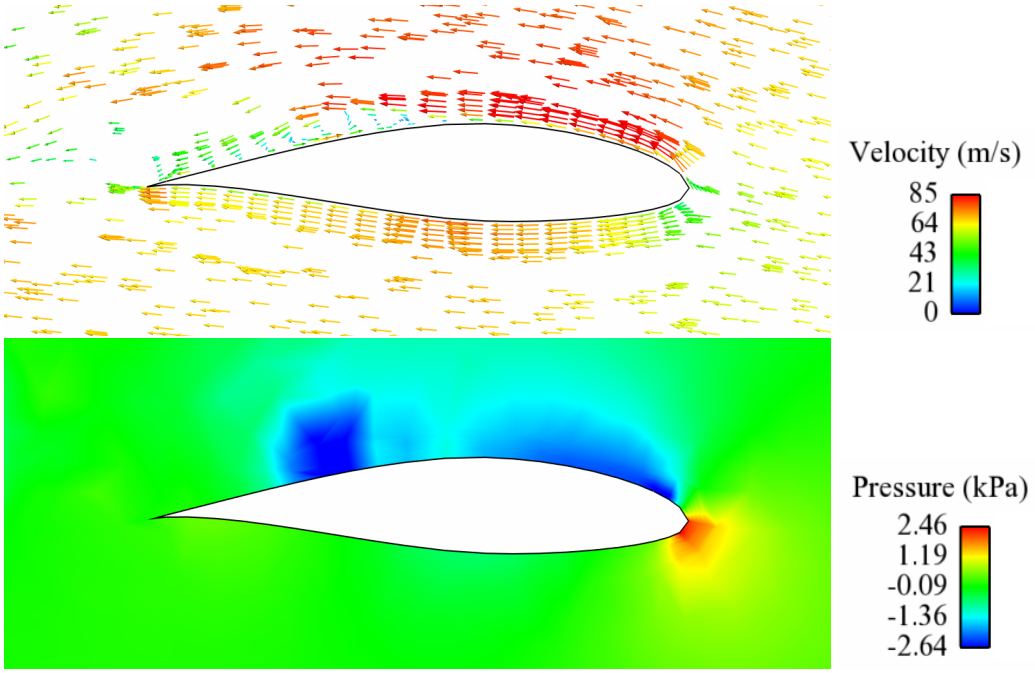


Figure 24: Velocity vectors in blade reference frame (top) and pressure field (bottom) for airfoil section at center of Patch 17 at 0.2 s.

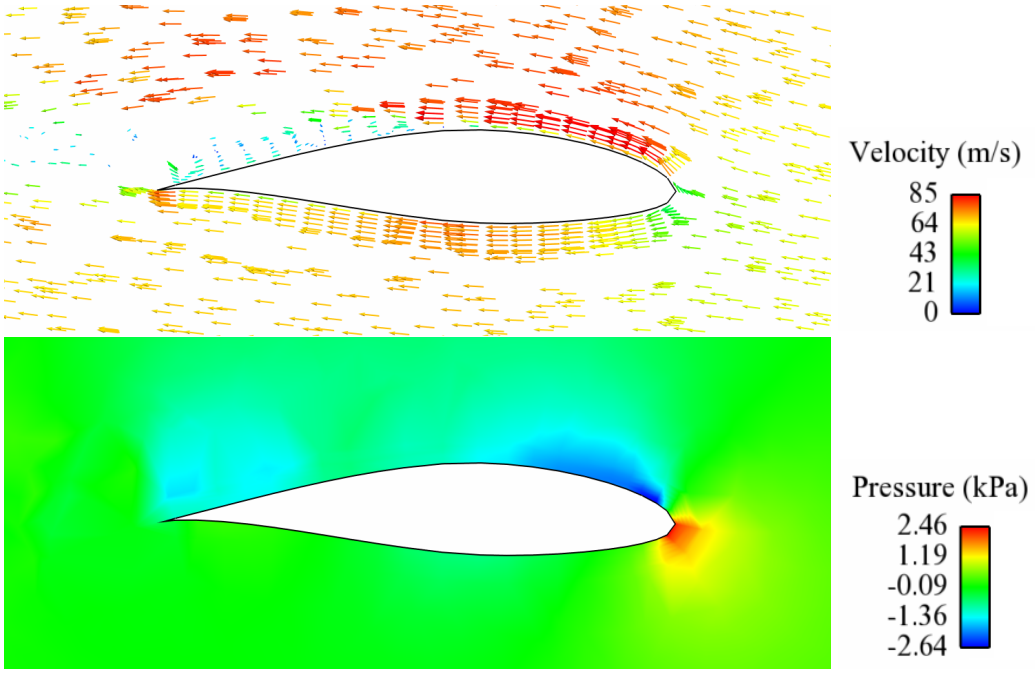


Figure 25: Velocity vectors in blade reference frame (top) and pressure field (bottom) for airfoil section at center of Patch 17 at 0.8 s.

5.4. Coupled air flow and rigid rotor simulation

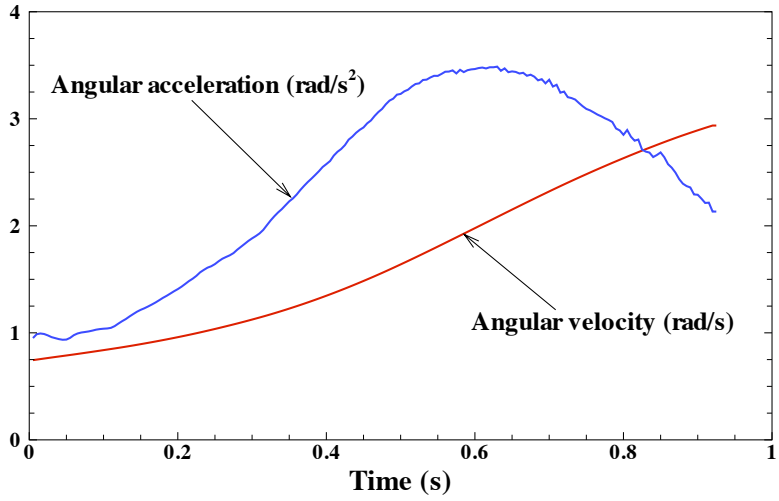


Figure 26: Time history of the rotor angular velocity and acceleration.

In this section, we perform a coupled aerodynamic analysis of the same NREL 5MW offshore baseline wind turbine using a NURBS discretization. The wind speed at the inflow is now set to 12 m/s. Reference [61] reports that at this wind speed the rotor blades are pitched to prevent over-rotation. Our interest was to study a scenario in which no blade pitching controls are applied and the rotor is free to spin under the influence of incoming wind. This situation may occur due to the failure of the rotor control system, in which case, under the prescribed wind condition, the rotor will spin at speeds much faster than the limits set by the design strength of the rotor structure.

The simulation is started with the rotor spinning at a prescribed angular velocity of 0.8 rad/s. Then the rotor is released to spin freely under the action of wind forces. We neglect the frictional losses and interaction with the generator, that is, we set $f(\dot{\theta}, t) = 0$ in Eq. (22). Figure 26 shows the time history of the rotor angular speed and acceleration. Within 1 s the rotor attains an angular velocity of 3 rad/s. The computation was stopped at this point, however, the rotor continues to accelerate. Due to the size and the dimension of the blade, such rotor speeds lead to unsustainable structural loads and, ultimately, wind turbine failure.

Figure 27 shows a snapshot of the rotor spinning at nearly 3 rad/s and the isosurfaces of the air speed. The trailing edge of the blade and especially the tip create significant amount of turbulence. Snapshots of air pressure on the front and back of the blade are shown in Figure 28. Note that, compared to the previous case of 9 m/s wind and 1.08 rad/s rotation, the pressure magnitude is significantly higher. Furthermore, the pressure field at the trailing edge and the blade tip exhibits fluctuating

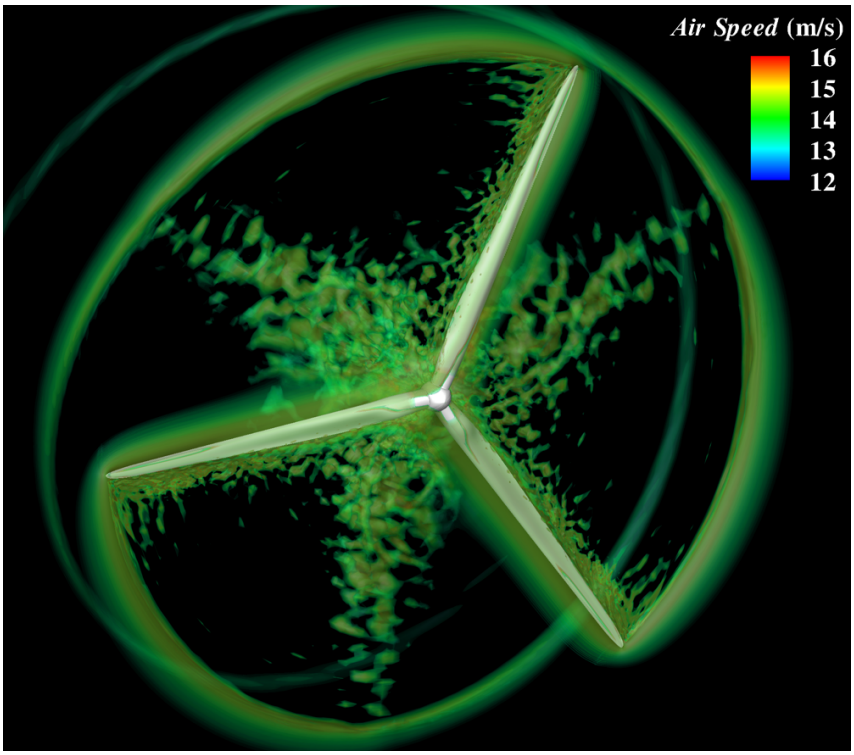
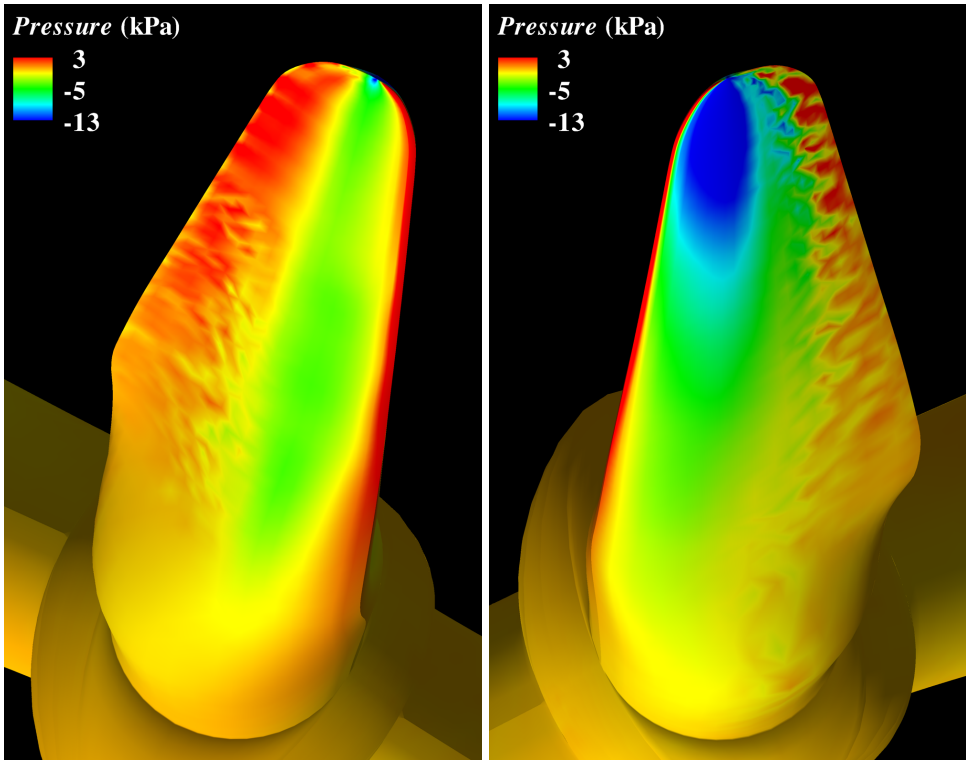


Figure 27: Isosurfaces of air speed at $t = 0.8$ s.

behavior, which leads to high-frequency loading and aerodynamic noise generation in these areas.



(a) Front (b) Back

Figure 28: Air pressure distribution on the blade surface.

6. Conclusions

This two-part paper presents a comprehensive computational framework for advanced simulation of wind turbines at full scale and its application to the NREL 5MW offshore baseline wind turbine. Part 1 of the paper focused on the wind turbine geometry modeling, mesh generation, and numerical simulation in which the rotor is assumed to be rigid. Isogeometric analysis was adopted as the primary geometry modeling and simulation framework. A template-based approach was developed in which the analysis-suitable geometry for both fluid and structural wind turbine rotor domains is constructed. The fluid and structural meshes are compatible at the interface and may be employed for the coupled FSI analysis, which is presented in Part 2 of this paper. The residual-based variational multiscale formulation was employed for the fluid simulation and turbulence modeling. The quasi-static assumption on the fine scales was generalized to moving domain problems in the ALE framework leading to a formulation that globally conserves linear momentum.

Partial validation of the turbulence modeling technique was performed on a turbulent Taylor–Couette flow, which has features in common with wind turbine flows, such as rotation, curved solid walls, and boundary layers. Comparison of linear finite elements and quadratic NURBS revealed the superior per-degree-of-freedom accuracy of the NURBS-based discretization. Accurate mean flow results are obtained even for very coarse meshes with no boundary layer resolution. Weakly-imposed boundary conditions were employed in this simulation and were partly responsible for the good quality of the computational results.

Further validation of the proposed method was performed by computing the NREL 5MW offshore baseline wind turbine rotor at full scale. Both NURBS and standard finite elements were employed for this simulation. The NURBS-based representation of the wind turbine geometry was used to create a tetrahedral mesh for the FEM simulation, which is an added benefit of the isogeometric framework. The results for aerodynamic torque, a key quantity in evaluating the wind turbine performance, were in close agreement with those reported in [61] for the NURBS simulation. This suggests that 3D, complex-geometry, time-dependent computational modeling of wind turbine rotors, which is fully predictive and does not rely on empiricism, is capable of accurately approximating the aerodynamic quantities of interest while keeping the number of degrees of freedom at a manageable level. The tetrahdral discretization, which used a significantly smaller number of degrees of freedom, produced a somewhat lower value of the torque than anticipated due to the premature flow separation that starts near the blade tip and spreads. The results of the tetrahedral simulation are expected to improve with further mesh refinement.

The computational framework was extended to coupling of the flow with a rotating rigid body, and successfully applied to the simulation of the wind turbine rotor over-spinning under high wind speed inlet conditions. Such computation, performed using the NURBS-based discretization, would not be possible with the currently existing methods and tools for wind wind turbine simulation.

In the future work we plan to employ weak imposition of Dirichlet boundary conditions for wind turbine simulations, thus reducing the computational cost associated with boundary layer refinement and increasing accuracy. We also plan to incorporate the effect of the wind turbine tower in the simulations, which may have considerable influence on the results. It should be noted that the presence of the tower will preclude the use of rotationally-periodic boundary conditions, which will increase computational cost. Given the advanced nature of the simulations, the proposed framework may be developed further and used for rotor control and blade geometry optimization. Mesh refinement to increase the accuracy of the tetrahedral simulations is also planned.

Acknowledgement

We wish to thank the Texas Advanced Computing Center (TACC) at the University of Texas at Austin for providing HPC resources that have contributed to the research results reported within this paper. M.-C. Hsu was partially supported by the Los Alamos - UC San Diego Educational Collaboration Fellowship. This support is gratefully acknowledged. This research was supported in part by the Rice Computational Research Cluster funded by NSF under Grant CNS-0821727. We thank Creighton Moorman for his help in the very early stages of the FEM simulations.

References

[1] C.A. Walford. Wind turbine reliability: Understanding and minimizing wind turbine O & M costs. Technical Report SAND2006-1100, Sandia National Laboratories, 2006.

[2] E. Echavarria, B. Hahn, and G.J.W. van Bussel. Reliability of wind turbine technology through time. *Journal of Solar Energy Engineering*, 130, 2008.

[3] 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.

[4] 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*, 194:4135–4195, 2005.

[5] J.A. Cottrell, A. Reali, Y. Bazilevs, and T.J.R. Hughes. Isogeometric analysis of structural vibrations. *Computer Methods in Applied Mechanics and Engineering*, 195:5257–5297, 2006.

[6] Y. Bazilevs, L. Beirao 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*, 16:1031–1090, 2006.

[7] 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*, 196:4160–4183, 2007.

[8] W. A. Wall, M. A. Frenzel, and C. Cyron. Isogeometric structural shape optimization. *Computer Methods in Applied Mechanics and Engineering*, 197:2976–2988, 2008.

[9] J.A. Cottrell, T.J.R. Hughes, and Y. Bazilevs. *Isogeometric Analysis: Toward Integration of CAD and FEA*. Wiley, Chichester, 2009.

[10] J.A. Evans, Y. Bazilevs, I. Babuška, and T.J.R. Hughes. N-widths, sup-infs, and optimality ratios for the k-version of the isogeometric finite element method. *Computer Methods in Applied Mechanics and Engineering*, 198:1726–1741, 2009.

[11] 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*, 199:264–275, 2010.

[12] 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*, 199:264–275, 2010.

[13] 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*, 199:314–323, 2010.

[14] W. Wang and Y. Zhang. Wavelets-based NURBS simplification and fairing. *Computer Methods in Applied Mechanics and Engineering*, 199:290–300, 2010.

[15] 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*, 199:334–356, 2010.

[16] V. Srinivasan, S. Radhakrishnan, and G. Subbarayan. Coordinated synthesis of hierarchical engineering systems. *Computer Methods in Applied Mechanics and Engineering*, 199:392–404, 2010.

[17] 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*, 197:173–201, 2007.

[18] 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*, 196:4853–4862, 2007.

[19] 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*, 2010. doi:10.1016/j.cma.2008.11.020.

[20] 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*, 41:371–378, 2008.

[21] 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*, 199:828–840, 2010.

[22] 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*, 229:3402–3414, 2010.

[23] T. Elguedj, Y. Bazilevs, V.M. Calo, and T.J.R. Hughes. Bbar and Fbar projection methods for nearly incompressible linear and non-linear elasticity and plasticity using higher-order NURBS elements. *Computer Methods in Applied Mechanics and Engineering*, 197:2732–2762, 2008.

[24] 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*, 199:357–373, 2010.

[25] 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*, 2010. DOI: 10.1002/nme.2864.

[26] 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*, 199:276–289, 2010.

[27] J. Kiendl, K.-U. Bletzinger, J. Linhard, and R. Wüchner. Isogeometric shell analysis with Kirchhoff–Love elements. *Computer Methods in Applied Mechanics and Engineering*, 198:3902–3914, 2009.

[28] 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*, 2010. doi:10.1016/j.cma.2010.03.029.

[29] 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*, 196:2943–2959, 2007.

[30] 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*, 38:310–322, 2006.

[31] Y. Bazilevs, V.M. Calo, T.J.R. Hughes, and Y. Zhang. Isogeometric fluid-structure interaction: theory, algorithms, and computations. *Computatonal Mechanics*, 43:3–37, 2008.

[32] 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*, 2008. DOI: 10.1161/STROKEAHA.107.503698.

[33] Y. Bazilevs and T.J.R. Hughes. NURBS-based isogeometric analysis for the computation of flows about rotating components. *Computatonal Mechanics*, 43:143–150, 2008.

[34] T. E. Tezduyar. Stabilized finite element formulations for incompressible flow computations. *Advances in Applied Mechanics*, 28:1–44, 1992.

[35] 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(3):339–351, 1992.

[36] 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(3):353–371, 1992.

[37] T.E. Tezduyar. Computation of moving boundaries and interfaces and stabilization parameters. *International Journal for Numerical Methods in Fluids*, 43:555–575, 2003.

[38] 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*, 54:855–900, 2007.

[39] 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*, 15:1073–1118, 1992.

[40] S. Mittal and T. E. Tezduyar. Parallel finite element simulation of 3D incompressible flows – Fluid-structure interactions. *International Journal for Numerical Methods in Fluids*, 21:933–953, 1995.

[41] 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*, 190:321–332, 2000.

[42] 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*, 190:373–386, 2000.

[43] 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*, 191:717–726, 2001.

[44] 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*, 195:2002–2027, 2006.

[45] 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*, 195:5743–5753, 2006.

[46] 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*, 54:901–922, 2007.

[47] 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*, 57:601–629, 2008.

[48] 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*, 43:39–49, 2008.

[49] T. E. Tezduyar, M. Schwaab, and S. Sathe. Sequentially-Coupled Arterial Fluid–Structure Interaction (SCAFSI) technique. *Computer Methods in Applied Mechanics and Engineering*, 198:3524–3533, 2009.

[50] T. E. Tezduyar, K. Takizawa, C. Moorman, S. Wright, and J. Christopher. Multiscale sequentially-coupled arterial FSI technique. *Computational Mechanics*, 46:17–29, 2010.

[51] 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*, published online, DOI: 10.1002/d.2221, 2009.

[52] 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*, 26:101–116, 2010.

[53] 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*, 46:31–41, 2010.

[54] 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*, published online, DOI: 10.1002/d.2348, May 2010.

[55] 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*, published online, DOI: 10.1002/d.2360, May 2010.

[56] K. Takizawa, S. Wright, C. Moorman, and T. E. Tezduyar. Fluid–structure interaction modeling of parachute clusters. *International Journal for Numerical Methods in Fluids*, published online, DOI: 10.1002/d.2359, May 2010.

[57] T. E. Tezduyar, S. Sathe, J. Pausewang, M. Schwaab, J. Christopher, and J. Crabtree. Fluid–structure interaction modeling of ringsail parachutes. *Computational Mechanics*, 43:133–142, 2008.

[58] T. Tezduyar, S. Aliabadi, M. Behr, A. Johnson, V. Kalro, and M. Litke. Flow simulation and high performance computing. *Computational Mechanics*, 18:397–412, 1996.

[59] M. Behr and T. Tezduyar. The Shear-Slip Mesh Update Method. *Computer Methods in Applied Mechanics and Engineering*, 174:261–274, 1999.

[60] 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*, 190:3189–3200, 2001.

[61] 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.

[62] E. Hau. *Wind Turbines: Fundamentals, Technologies, Application, Economics. 2nd Edition*. Springer, Berlin, 2006.

[63] H.J.T. Kooijman, C. Lindenburg, D. Winkelaar, and E.L. van der Hooft. DOWEC 6 MW pre-design: Aero-elastic modelling of the DOWEC 6 MW pre-design in PHATAS. Technical Report DOWEC-F1W2-HJK-01-046/9, 2003.

[64] J.M. Jonkman and M.L. Buhl Jr. FAST user's guide. Technical Report NREL/EL-500-38230, National Renewable Energy Laboratory, Golden, CO, 2005.

[65] 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*, 127:387–401, 1995.

[66] T. J. R. Hughes, G. Feijóo., L. Mazzei, and J. B. Quincy. The variational multiscale method – A paradigm for computational mechanics. *Computer Methods in Applied Mechanics and Engineering*, 166:3–24, 1998.

[67] 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*, 32:199–259, 1982.

[68] 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*, 45:217–284, 1984.

[69] 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*, 59:307–325, 1986.

[70] 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*, 95:221–242, 1992.

[71] L. P. Franca and S. Frey. Stabilized finite element methods: II. The incompressible Navier-Stokes equations. *Computer Methods in Applied Mechanics and Engineering*, 99:209–233, 1992.

[72] T. E. Tezduyar and Y. Osawa. Finite element stabilization parameters computed from element matrices and vectors. *Computer Methods in Applied Mechanics and Engineering*, 190:411–430, 2000.

[73] 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, Computational Fluid Dynamics*, chapter 2. Wiley, 2004.

[74] L. Catabriga, A. L. G. A. Coutinho, and T. E. Tezduyar. Compressible flow SUPG parameters computed from degree-of-freedom submatrices. *Computational Mechanics*, 38:334–343, 2006.

[75] 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*, 45:539–557, 2007.

[76] Y. Bazilevs and T.J.R. Hughes. Weak imposition of Dirichlet boundary conditions in fluid mechanics. *Computers and Fluids*, 36:12–26, 2007.

[77] R. Calderer and A. Masud. A multiscale stabilized ALE formulation for incompressible flows with moving boundaries. *Computational Mechanics*, 46:185–197, 2010.

[78] G. Lewis and H. Swinney. Velocity structure functions, scaling and transitions in high-Reynolds-number Couette–Taylor flow. *Phys. Rev. E*, 59:5457–5467, 1999.

[79] T. Tezduyar, S. Aliabadi, M. Behr, A. Johnson, and S. Mittal. Parallel finite element computation of 3D flows. *Computer*, 26:27–36, 1993.

[80] T.E. Tezduyar, S.K. Aliabadi, M. Behr, and S. Mittal. Massively parallel finite element simulation of compressible and incompressible flows. *Computer Methods in Applied Mechanics and Engineering*, 119:157–177, 1994.

[81] M. Bilson and K. Bremhorst. Direct numerical simulation of turbulent Taylor-Couette flow. *Journal of Fluid Mechanics*, 579:227–270, 2007.

[82] D. Pirro and M. Quadrio. Direct numerical simulation of turbulent Taylor-Couette flow. *European Journal of Mechanics - B/Fluids*, 27:552–566, 2007.

[83] S. Dong. Turbulent flow between counter-rotating concentric cylinders: A DNS study. *Journal of Fluid Mechanics*, 615:371–399, 2008.

[84] S. B. Pope. Large-eddy simulation using projection onto local basis functions. In J. L. Lumley, editor, *Fluid Mechanics and the Environment: Dynamical Approaches*. Springer, 2000. To appear.

[85] C.G. Speziale, B.A. Younis, R. Rubinstein, and Y. Zhou. On consistency conditions for rotating turbulent flows. *Physics of Fluids*, 10:2108–2110, 1998.

[86] S. Dong. DNS of turbulent Taylor-Couette flow. *Journal of Fluid Mechanics*, 587:373–393, 2007.

[87] R.M. Kirby and G.E. Karniadakis. Spectral element and hp methods. In E. Stein, R. de Borst, and T. J. R. Hughes, editors, *Encyclopedia of Computational Mechanics, Vol. 3, Computational Fluid Dynamics*. Wiley, 2004.

[88] J. Jeong and F. Hussain. On the identification of a vortex. *Journal of Fluid Mechanics*, 285:69–94, 1995.

[89] K. E. Jansen, C. H. Whiting, and G. M. Hulbert. A generalized-α method for integrating the filtered Navier-Stokes equations with a stabilized finite element method. *Computer Methods in Applied Mechanics and Engineering*, 190:305–319, 1999.

[90] J. Chung and G. M. Hulbert. A time integration algorithm for structural dynamics with improved numerical dissipation: The generalized-α method. *Journal of Applied Mechanics*, 60:371–75, 1993.

[91] K. Takizawa, C. Moorman, S. Wright, and T. E. Tezduyar. Computer modeling and analysis of the Orion spacecraft parachutes. In Hans-Joachim Bungartz and M. Schafer, editors, *Fluid–Structure Interaction – Modelling, Simulation, Optimization, Part II*, Lecture Notes in Computational Science and Engineering. Springer, 2010.

[92] D. A. Spera. Introduction to modern wind turbines. In D. A. Spera, editor, *Wind Turbine Technology: Fundamental Concepts of Wind Turbine Engineering*, pages 47–72. ASME Press, 1994.

[93] Texas Advanced Computing Center (TACC). http://www.tacc.utexas.edu.

[94] Y. Saad and M. Schultz. GMRES: A generalized minimal residual algorithm for solving nonsymmetric linear systems. *SIAM Journal of Scientific and Statistical Computing*, 7:856–869, 1986.

[95] G. Karypis and V. Kumar. A fast and high quality multilevel scheme for partitioning irregular graphs. *SIAM Journal of Scientific Computing*, 20:359–392, 1998.

[96] F. Rispoli, A. Corsini, and T. E. Tezduyar. Finite element computation of turbulent flows with the discontinuity-capturing directional dissipation (DCDD). *Computers & Fluids*, 36:121–126, 2007.

[97] J. Smagorinsky. General circulation experiments with the primitive equations. *Monthly Weather Review*, 91(3):99–165, 1963.

[98] T. E. Tezduyar, S. Ramakrishnan, and S. Sathe. Stabilized formulations for incompressible flows with thermal coupling. *International Journal for Numerical Methods in Fluids*, 57:1189–1209, 2008.