# High-performance computing of wind turbine aerodynamics using isogeometric analysis

Ming-Chen Hsu[a,*], Ido Akkerman[a,b], Yuri Bazilevs[a]

[a]*Department of Structural Engineering, University of California, San Diego, 9500 Gilman Drive, Mail Code 0085, La Jolla, CA 92093, USA*
[b]*Coastal and Hydraulics Laboratory, US Army Engineer Research and Development Center, 3909 Halls Ferry Road, Vicksburg, MS 39180-6133, USA*

**Abstract**

In this article we present a high-performance computing framework for advanced flow simulation and its application to wind energy based on the residual-based variational multiscale (RBVMS) method and isogeometric analysis. The RBVMS formulation and its suitability and accuracy for turbulent flow in a moving domain are presented. Particular emphasis is placed on the parallel implementation of the methodology and its scalability. Two challenging flow cases were considered: the turbulent Taylor–Couette flow and the NREL 5MW offshore baseline wind turbine rotor at full scale. In both cases, flow quantities of interest from the simulation results compare favorably with the reference data and near-perfect linear parallel scaling is achieved.

*Keywords:* wind turbine, RBVMS, turbulence modeling, isogeometric analysis, NURBS, high-performance computing, parallel scalability

## 1. Introduction

The present costs for wind energy are dominated by the operations and maintenance of the wind turbine system. It is shown in [1] that a typical wind turbine has averaged of 2.6 component failures per year during the first 10 years of operation. However, the industry is currently unable to predict these failure mechanisms and the component failure leads to the unscheduled downtime and reduced capacity. At the same time, offshore wind turbines are receiving increased attention. Winds in the offshore environment are usually stronger and more sustained, providing a more reliable source of energy. However, offshore wind turbines are exposed to harsh environments and must be designed to withstand more severe loads than the inland wind turbines. Rotor blades of much larger diameter (> 120 m) are being designed and built for better performance. These are significant engineering challenges that must be addressed through advanced research and development, which also involves advanced and large-scale simulations.

*Corresponding author
*Email address:* m5hsu@ucsd.edu (Ming-Chen Hsu)

Due to the computational modeling challenges involved (and only recently developed interest in the application), state-of-the-art in wind turbine simulation is not as advanced as in other fields of engineering. In recent years, standalone fluid mechanics simulations of wind turbine configurations were reported in [2–5], while standalone structural analyses of rotor blades under assumed load conditions were reported in [6, 7]. Our recent work [8] has shown that coupled fluid–structure interaction (FSI) modeling of wind turbines is important in order to accurately predict their mechanical behavior. However, in order to perform fully-coupled FSI simulation of wind turbines at full spatial scale, advanced high-performance computing (HPC) resources, robust and accurate numerical methodology, and software with good parallel scalability are required. In this paper, we describe our computational procedures that enable efficient simulation of wind turbine rotors at full scale.

This paper is outlined as follows. In Section 2, we introduce the Arbitrary Lagrangian–Eulerian (ALE) form of the Navier–Stokes equations of incompressible flow suitable for moving domain problems. We also present the residual-based variational multiscale (RBVMS) formulation of the Navier–Stokes equations and turbulence modeling [9]. We review the basics and state-of-the-art of Isogeometric Analysis [10]. In Section 3, we describe our parallel implementation strategy in detail. In Section 4, we present our simulation and parallel scalability results for the turbulent Taylor–Couette flow and the NREL 5MW offshore baseline wind turbine rotor. In Section 5, we draw conclusions.

## 2. Numerical methods

### *2.1. Navier–Stokes equations of incompressible flow in a moving domain*

We begin by considering a weak formulation of the Navier–Stokes equations of incompressible flow in a moving domain. Let $\Omega \subset \mathbb{R}^3$ denote the fluid domain at the current time and $\Gamma = \partial\Omega$ is its boundary. Let $\mathcal{V}$ and $\mathcal{W}$ be the infinite-dimensional trial solution and weighting function spaces, respectively, and $(\cdot,\cdot)_\Omega$ denote the $L^2$-inner product over $\Omega$. The variational formulation corresponding to the arbitrary Lagrangian–Eulerian (ALE) form 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

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

In the above equations, $\rho$ is the density, $\hat{\boldsymbol{v}}$ is the fluid domain velocity, $\mu$ is the kinematic viscosity, $\nabla^s$ is the symmetric gradient operator, and the time derivative is taken with respect to a fixed spatial coordinate in the reference configuration. In the absence of the fluid domain motion, Eq. (2) reverts to a standard incompressible flow formulation in a stationary domain.

*2.2. Residual-based variational multiscale formulation in a moving domain*

Replacing the infinite-dimensional spaces $\mathcal{V}$ and $\mathcal{W}$ with their finite-dimensional counterparts $\mathcal{V}^h$ and $\mathcal{W}^h$ leads to the Galerkin formulation, which is unstable for advection-dominated flow and for the use of equal-order velocity–pressure discretization. In what follows, we present a stable and accurate discrete formulation based on the variational multiscale (VMS) framework (see, e.g. [11, 12]) that circumvents the aforementioned issues.

Let the superscript $h$ denote the resolved coarse scales represented by the finite element or isogeometric discretization. The trial solution and weighting function spaces are split into coarse and fine scales as

$$\{\boldsymbol{v}, p\} = \{\boldsymbol{v}^h, p^h\} + \{\boldsymbol{v}', p'\}, \tag{3}$$

$$\{\boldsymbol{w}, q\} = \{\boldsymbol{w}^h, q^h\} + \{\boldsymbol{w}', q'\}, \tag{4}$$

where the primed quantities correspond to the unresolved scales that will be modeled in what follows.

The above decomposition of the weighting functions leads to two variational subproblems:

$$B(\{\boldsymbol{w}^h, q^h\},\ \{\boldsymbol{v}^h, p^h\} + \{\boldsymbol{v}', p'\}) - (\boldsymbol{w}^h, \rho\boldsymbol{f})_\Omega = 0 \quad \forall \{\boldsymbol{w}^h, q^h\} \in \mathcal{W}^h, \tag{5}$$

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

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. The fine-scale velocity and pressure are modeled as being proportional to the strong form of the Navier–Stokes partial differential equation residuals [9]:

$$\boldsymbol{v}' = -\tau_{\mathrm{M}} \boldsymbol{r}_{\mathrm{M}}(\{\boldsymbol{v}^h, p^h\}), \tag{7}$$

$$p' = -\tau_{\mathrm{C}} r_{\mathrm{C}}(\boldsymbol{v}^h), \tag{8}$$

where

$$\boldsymbol{r}_{\mathrm{M}}(\{\boldsymbol{v}, p\}) = \rho \frac{\partial \boldsymbol{v}}{\partial t} + \rho(\boldsymbol{v} - \hat{\boldsymbol{v}}) \cdot \nabla \boldsymbol{v} + \nabla p - \mu \nabla^2 \boldsymbol{v} - \rho \boldsymbol{f}, \tag{9}$$

$$r_{\mathrm{C}}(\boldsymbol{v}) = \nabla \cdot \boldsymbol{v}. \tag{10}$$

In (7) and (8), $\tau_{\mathrm{M}}$ and $\tau_{\mathrm{C}}$ are the stabilization parameters originating from stabilized finite element methods for fluid dynamics (see, e.g. [13–17]). Recently, they were interpreted as appropriate approximations of the fine-scale Green's function, which is a key mathematical object in the VMS method (see [18] for an elaboration).

To generate the numerical method, $\boldsymbol{v}'$ and $p'$ from Eqs. (7) and (8) are inserted directly into the coarse-scale equation (5). The formulation is typically simplified based on the assumptions that the fine scales are orthogonal to the coarse scales with respect to the inner-product generated by the viscous term, and the fine scales are quasi-static [9]. These lead to the following semi-discrete variational formulation: find $\{\boldsymbol{v}^h, p^h\} \in \mathcal{V}^h$, $\boldsymbol{v}^h = \boldsymbol{g}$ on $\Gamma_g$, such that $\forall \{\boldsymbol{w}^h, q^h\} \in \mathcal{V}^h$, $\boldsymbol{w}^h = \boldsymbol{0}$ on $\Gamma_g$,

$$B(\{\boldsymbol{w}^h, q^h\}, \{\boldsymbol{v}^h, p^h\}) + B_{\mathrm{VMS}}(\{\boldsymbol{w}^h, q^h\}, \{\boldsymbol{v}^h, p^h\}) - (\boldsymbol{w}^h, \rho\boldsymbol{f})_\Omega = 0, \tag{11}$$

where the modeled subgrid-scale terms in $B_{\mathrm{VMS}}$ are

$$\begin{aligned} B_{\mathrm{VMS}}(\{\boldsymbol{w}, q\}, \{\boldsymbol{v}, p\}) = & (\nabla \boldsymbol{w}, \rho \tau_{\mathrm{M}} \boldsymbol{r}_{\mathrm{M}}(\{\boldsymbol{v}, p\}) \otimes (\boldsymbol{v} - \hat{\boldsymbol{v}}))_{\Omega} \\ & + (\nabla q, \tau_{\mathrm{M}} \boldsymbol{r}_{\mathrm{M}}(\{\boldsymbol{v}, p\}))_{\Omega} \\ & + (\nabla \cdot \boldsymbol{w}, \tau_{\mathrm{C}} r_{\mathrm{C}}(\boldsymbol{v}))_{\Omega} \\ & - (\boldsymbol{w}, \rho \tau_{\mathrm{M}} \boldsymbol{r}_{\mathrm{M}}(\{\boldsymbol{v}, p\}) \cdot \nabla \boldsymbol{v})_{\Omega} \\ & - (\nabla \boldsymbol{w}, \rho \tau_{\mathrm{M}} \boldsymbol{r}_{\mathrm{M}}(\{\boldsymbol{v}, p\}) \otimes \tau_{\mathrm{M}} \boldsymbol{r}_{\mathrm{M}}(\{\boldsymbol{v}, p\}))_{\Omega}, \end{aligned} \tag{12}$$

and the integrals are taken element-wise. We note that the quasi-static assumption on the fine scales was generalized to moving domain problems in [19] such that the semi-discrete formulation (11) globally conserves linear momentum.

The discrete formulation (11) derives from a VMS paradigm, which is viewed as a framework for generating models of turbulence (see, e.g. [9, 20–25]). Equation (11) is known as the residual-based variational multiscale (RBVMS) turbulence model [9]. Note that, in contrast to classical turbulence modeling approaches, no *ad hoc* eddy viscosity terms are present in the RBVMS formulation, which is one of its distinguishing features. Because the explicit use of eddy viscosities is avoided, the present turbulence modeling approach is not likely to suffer from the shortcomings of eddy-viscosity-based turbulence models for flows dominated by rotation in the presence of curved boundaries (see [26, 27]). This fact also makes the current methodology attractive for simulating wind turbine aerodynamics.

### *2.3. Isogeometric analysis*

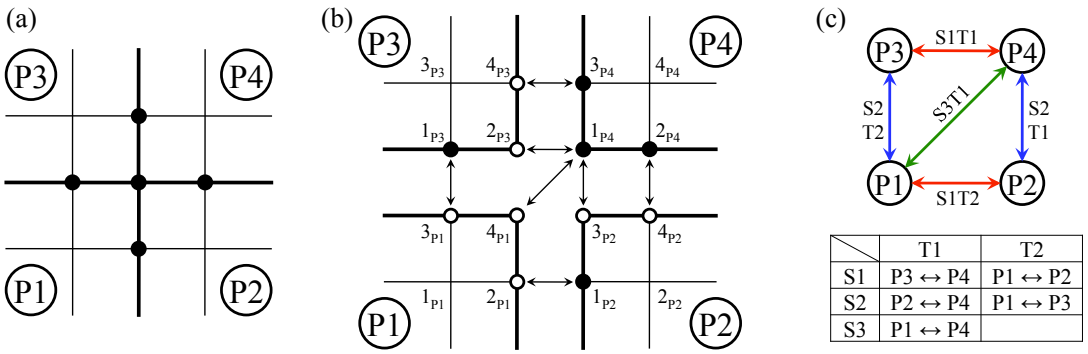
To discretize Eq. (11) in space we use isogeometric analysis, a new computational framework that was first proposed in [10] and further detailed in [28]. Isogeometric analysis is based on the technologies originating from computer graphics and computer-aided design (CAD). The basic idea of the isogeometric concept is to use the same basis for design and computational analysis. It can be thought of as a generalization of the finite element method and has similar features such as variational framework, compactly supported basis functions and geometric flexibility. However, isogeometric analysis has several advantages and offers new possibilities that do not exist in standard finite elements. It has precise and efficient modeling of complex geometry and smooth basis functions with degree of continuity beyond $C^0$. Smooth basis functions that are $C^1$-continuous or higher can be directly employed to discretize higher-order differential operators (see, e.g. [29–32]).

Non-uniform rational B-splines (NURBS) [33] were the first and currently the most developed basis function technology in isogeometric analysis. Mathematical theory of NURBS-based isogeometric analysis was originally developed in [34] and further refinements and insights into approximation properties of NURBS were studied in [35]. NURBS-based isogeometric analysis was applied with good success to the study of fluids [36–38], structures [30, 31, 39–41] and fluid–structure interaction [42–45]. The RBVMS formulation (11), in conjunction with NURBS-based isogeometric analysis, was employed in the simulation of several important turbulent flows in [27, 46–48]. In most cases, isogeometric analysis gave an 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.

Recent developments in isogeometric analysis include efficient quadrature rules [49], isogeometric model quality assessment and improvement [50–52], and T-Splines [53, 54]. It is also shown in [55] that the existing finite element code can be converted to an isogeometric analysis code without too many modifications by using the Bézier extraction operator. We note that because conic sections and cylindrical shapes can be represented exactly using quadratic or higher-order NURBS, they present an added benefit for computation of flows involving rotating components [37], such as wind turbine rotors.

## 3. Parallel Implementation




| | T1 | T2 |
|---|---|---|
| S1 | P3 ↔ P4 | P1 ↔ P2 |
| S2 | P2 ↔ P4 | P1 ↔ P3 |
| S3 | P1 ↔ P4 | |

Figure 1: An example of the domain decomposition and parallel communications. (a) The computational domain is partitioned into four subdomains and each subdomain is individually assigned to a processor (P). Black dots are the control points shared by the subdomains. (b) The shared control points are designated to be "masters" (black dots) or "slaves" (white dots) in each subdomain. The arrows indicate the correspondence between the master and slave control points. (c) Communication stages (S) and tasks (T) for this example.

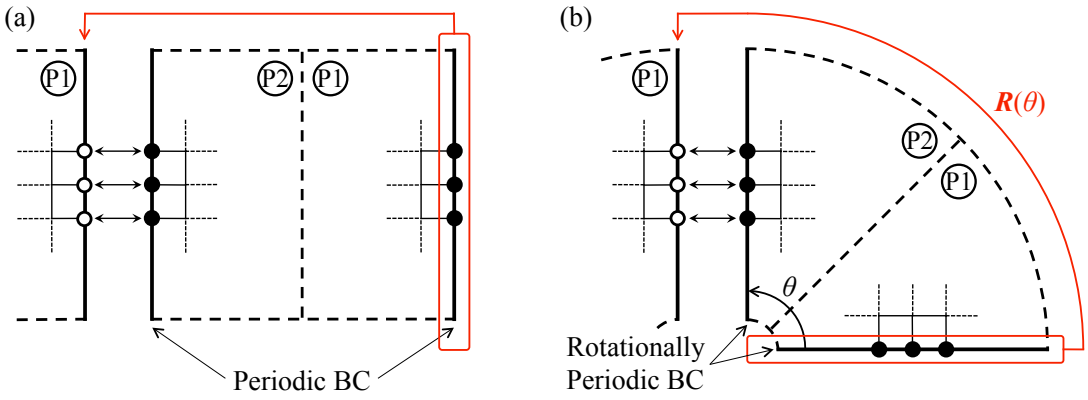



Figure 2: Illustration of parallelization procedures for (a) periodic boundaries, including (b) boundaries with rotational periodicity. Here $\boldsymbol{R}(\theta)$ is the rotation matrix for the rotationally periodic boundary conditions.

Turbulent flows, especially in the regime of large eddy simulation (LES), require substantial grid resolution for accuracy. Parallel computing is thus essential to efficiently compute

turbulence. Although several references present applications of isogeometric analysis to turbulent flow, no discussion is given with regard to parallel implementation employed and parallel performance achieved by NURBS discretizations. For this reason, here, for the first time, we describe our parallel implementation of isogeometric analysis that allows us to achieve strong scalability on massively parallel computers.

There are mainly two stages that are involved in the computational work for standard finite elements or isogeometric analysis. The first stage consists of numerical integration over the elements and formation of the equation system (i.e. assembly of the left-hand-side tangent matrix and right-hand-side residual vector). The second stage consists of solving the linear system of equations using an iterative method. In this work we employ the Generalized Minimum Residual (GMRES) method [56].

Our parallel implementation makes use of the Message Passing Interface (MPI) libraries and is adapted from [57]. We use the non-overlapping element-based partitioning of the computational domain. The domain is partitioned into subdomains with similar work load as a preprocessing step, and each subdomain is individually assigned to a processor (or core). As a result, each element is uniquely assigned to a single partition and the assembly stage is highly scalable, both in terms of speed and memory.

However, element-based partitioning leads to shared control points[1] at the inter-processor boundaries. Figure 1a shows a computational domain that is partitioned into four subdomains. Black dots are the control points shared by the subdomains, while the rest of the control points reside solely in their subdomains and are not shared. Typically, a good balance of elements with sufficient work load in each partition and a minimum amount of communication between the partitions result in a reasonable control-point balance as well. This also helps maintain good scalability in solving the resultant linear system of equations using an iterative solver.

Each subdomain will generally have shared control points requiring inter-processor communication, as shown in Figure 1b. The shared control points are the points that reside on different partitions that originally referred to the same global control point number in the non-partitioned domain. In a set of control points that referred to the same global control point number, only one is designated to be a "master" and the rest are "slaves". Figure 1b shows an example where the black dots are the master control points and the white dots are the slave control points. The arrows indicate the correspondence between the master and slave control points.

Every partition contains information about its portion of the work and its interaction with neighboring partitions. The interaction between neighboring partitions is defined based on shared control points. The design is such that only the master control point will be in charge of data accumulation and update. As a result, the communications will only exist between master and slave control points. No communications are needed between slave control points.

The process of communicating the information to and from the master control points is decomposed into so-called communication stages. Each stage involves several communication tasks, which are one-to-one communications between two unique processors. During a communication task the data for the shared control points on a given processor is packaged into a

[1]Control points in isogeometric analysis are analogs of nodes in standard finite elements. In isogeometric analysis the degrees of freedom reside on control points and are referred to as control variables [10].

single array (using MPI derived data types) and sent (or received) as a single message. Figure 1c illustrates the communication stages and tasks for the example shown. Note that no communication will take place between processors 2 and 3, because the share control point is designated to be a slave on these processors. The master–slave relationships and the communication stages and tasks are defined in the pre-processing step.

The master–slave framework is also well suited for handling periodic boundaries (see Figure 2a). For rotationally periodic boundary conditions, in addition to defining the necessary inter-processor communication structures, further operations are needed and are discussed in what follows.

There are two major types of communications. The first type is where the data is accumulated at the master control points from their slave control points to obtain a "complete value". A control point is said to have a complete value if it has the same value in the partitioned case as in the unpartitioned case [58]. For example, after numerical integration on each local partition, non-shared control points will have complete values in the right-hand-side vector. However, values for shared control points are incomplete on each subdomain because their contributions are distributed among the partitions that shares the same control point. Communications between processors are needed in this situation. Slave control points will send and add their incomplete values to the corresponding master control points. After finishing all the communication stages, the master control points have complete values, while the slave control points are assigned zero values for the right-hand-side vector.

The second type of the communication is where the complete values are copied from the masters to update their slave control points. For example, after solving the linear system, the master control points have complete updated value of the solutions. Communications will then take place to send back these solutions from the master control points to their corresponding slave control points. As a result, the slave control points will also have the complete value of the solutions.

For rotationally periodic boundaries the complete value of the right-hand-side vector is obtained by first rotating the linear momentum residual at the slave control points to the correct coordinate system, and then sending and adding the contributions to the master control points. Likewise, when the value of the velocity vector is computed on the master control points, it is first rotated back and then sent to the slave control points. In order to rotate the right-hand-side and solution vectors, we construct an appropriate rotation matrix at the rotationally periodic control points. Figure 2b illustrates this case. Note that no rotation is necessary to communicate the scalar pressure variable or the continuity equation residual at the rotationally periodic boundaries.

To solve the linear system using GMRES, only the product of the left-hand-side matrix with a right-hand-side vector is needed. To efficiently compute this product in parallel, a vector, which is assembled at the processor level and communicated globally, is multiplied against a local-to-the-processor assembled matrix in parallel. To assess convergence and decide whether to continue with the iteration process or update the solution, a new communication is performed.

Note that for a fixed-size problem partitioned into large number of subdomains, the number of elements per partition becomes relatively small. This may result in significant imbalance of control points since the control point balance is not explicitly requested in the procedure.

Furthermore, the percentage of shared control points increases and communication stages and tasks may also increase. All these may eventually becomes detrimental to scaling.

## 4. Computational results

### *4.1. Turbulent Taylor–Couette flow at Re = 8000*

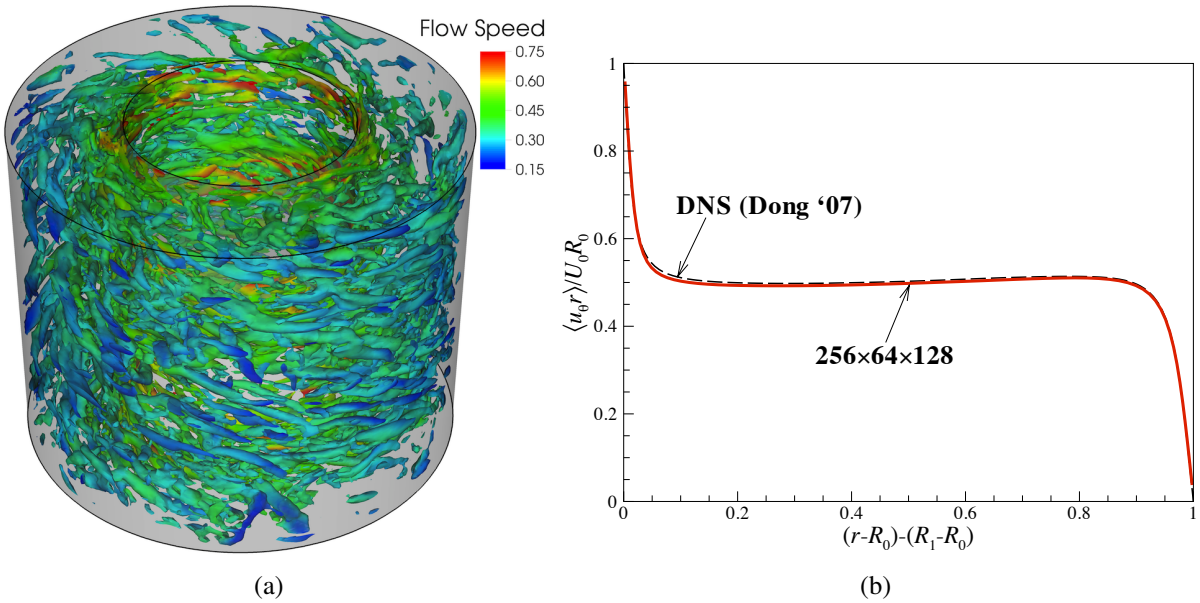



Figure 3: Turbulent Taylor–Couette flow: (a) Isosurfaces of $Q$ colored by flow speed. (b) Mean angular momentum compared with the DNS result of Dong [59].

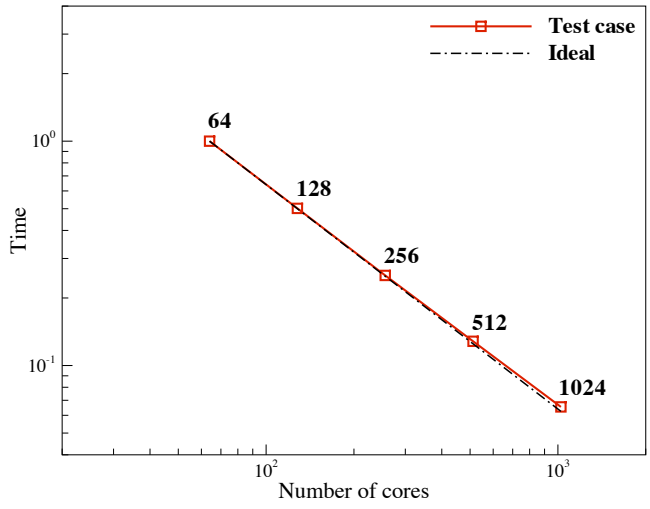



Figure 4: Scalability study for turbulent Taylor–Couette flow simulation. The computation time is normalized by the result of 64-processor case.

A turbulent Taylor–Couette flow at $Re = 8000$ is simulated on a mesh consisting of $256 \times 64 \times 128$ quadratic NURBS elements in the azimuthal, radial and axial directions, respectively. The problem setup is same as the one reported in [27]. A uniform mesh is used in the azimuthal and axial directions. In the radial direction, a hyperbolic tangent mesh stretching is used to better capture the boundary-layer turbulence. No-slip boundary conditions are imposed weakly using the methodology presented in [36, 47].

Figure 3a shows the isosurfaces of a scalar quantity $Q$, which is designed to be an objective quantity for identifying vortical features in the turbulent flow (see, e.g. [60]). The figure illustrates the complexity of the turbulent flow and the high demand on the numerical method to adequately represent it. Figure 3b shows the flow statistics of mean angular momentum, where the curve lies on top of the DNS result reported by Dong [59]. This is not surprising because very good accuracy for the same test case was achieved in our computations in [19, 27], but on meshes that are significantly coarser than the one used here. Such a fine NURBS mesh was chosen purely in the interest of performing a meaningful parallel scalability study of our isogeometric turbulence code.

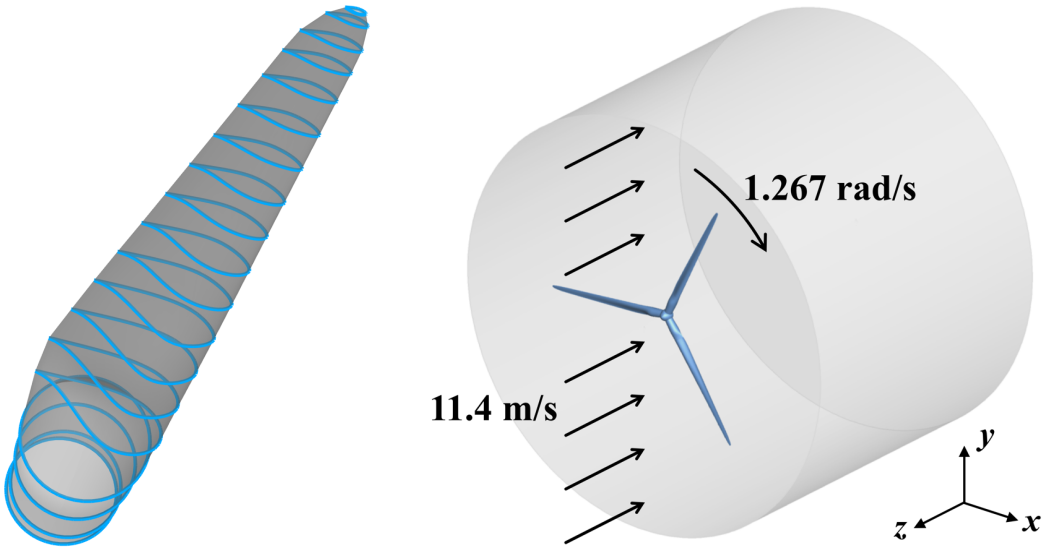
The parallel scalability tests were carried out on Ranger, a Sun Constellation Linux Cluster at the Texas Advanced Computing Center (TACC) [61]. The Ranger system is comprised of 3936 16-way SMP compute nodes providing 15744 AMD Opteron$^{TM}$ processors for a total of 62976 compute cores, 123 TB of total memory and 1.7 PB of raw global disk space. It has a theoretical peak performance of 579 TFLOPS. All Ranger nodes are interconnected using InfiniBand technology in a full-CLOS topology providing a 1GB/sec point-to-point bandwidth [62].

The $256 \times 64 \times 128$ quadratic NURBS mesh, which consists of nearly 2.1 million quadratic elements and about the same number of control points (analogues of nodes in finite elements), is decomposed into 64, 128, 256, 512 and 1024 subdomains and each subdomain is assigned to a processor. In the axial and radial directions the mesh is decomposed into 8 and 4 partitions, respectively, for all cases. In the azimuthal direction the mesh is decomposed into 2, 4, 8, 16 or 32 partitions. $C^1$-continuity at the inter-processor boundaries is maintained by imposing appropriate linear constraints on the velocity and pressure control variables. The scalability test results are shown in Figure 4. The code exhibits perfect linear scaling for the cases of 64, 128 and 256 processors. Near-perfect strong scaling is achieved for the 512- and 1024-processor cases.

### *4.2. 5MW offshore wind turbine simulation*

We use a template-based NURBS geometry modeling approach to construct the computational domain of the NREL 5MW offshore baseline wind turbine rotor [8, 19, 63]. The rotor blade surface is composed of a collection of airfoil shapes stacked along the blade axis. The blade and the airfoil cross-sections that define it is shown in Figure 5a. We compute the aerodynamics of the wind turbine rotor using the ALE approach on a rotating mesh.

The problem setup is shown in Figures 5b. The rotor radius is 63 m and the prescribed rotor speed is 1.267 rad/s. The blade is assumed to be rigid. At the inflow boundary the wind speed is set to 11.4 m/s, 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. The air density and viscosity are 1.2 kg/m$^3$




(a) (b)

Figure 5: Wind turbine simulation: (a) Constructed blade and airfoil cross-sections that define it. (b) Problem setup with prescribed inflow wind velocity and rotor speed.

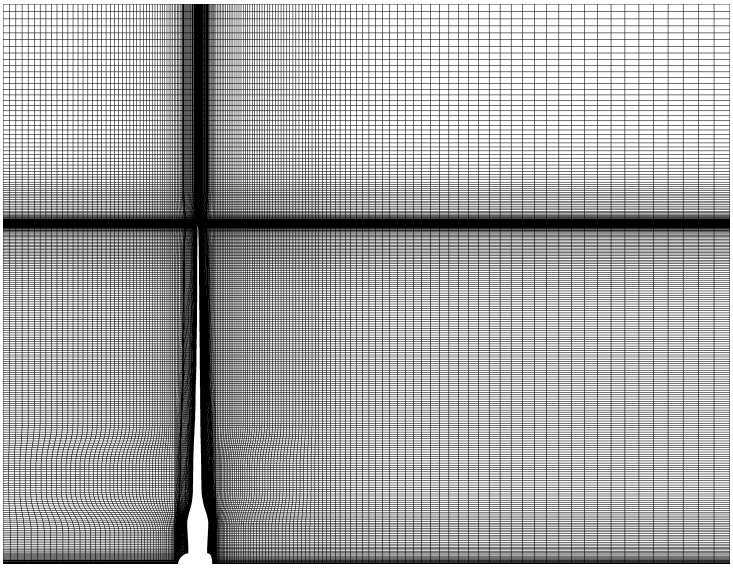
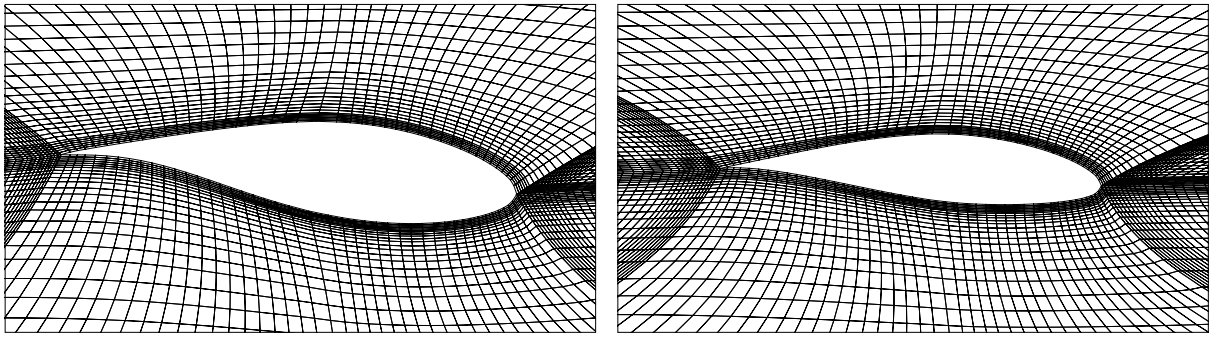


Figure 6: A planar cut of the volumetric NURBS mesh of the computational domain to illustrate the mesh resolution. The mesh is clustered toward the rotor blade. For the purpose of visualization, each quadratic NURBS element is interpolated with $2 \times 2 \times 2$ bilinear elements.

and $2.0 \times 10^{-5}$ kg/(m·s), respectively. To enhance the efficiency of the simulations, we take advantage of the problem symmetry. We construct a 120° slice of the computational domain and impose rotationally periodic boundary conditions [19, 64, 65]. The mesh of the 120° slice of the domain is comprised of 1449000 quadratic NURBS elements, which yields about the same number of mesh control points. Figure 6 shows a planar cut of the volumetric NURBS mesh of



(a) (b)

Figure 7: 2D blade cross-sections at (a) $0.6R$ and (b) $0.75R$ to illustrate the boundary layer mesh used in our computation. $R$ = 63 m is the radius of the rotor.

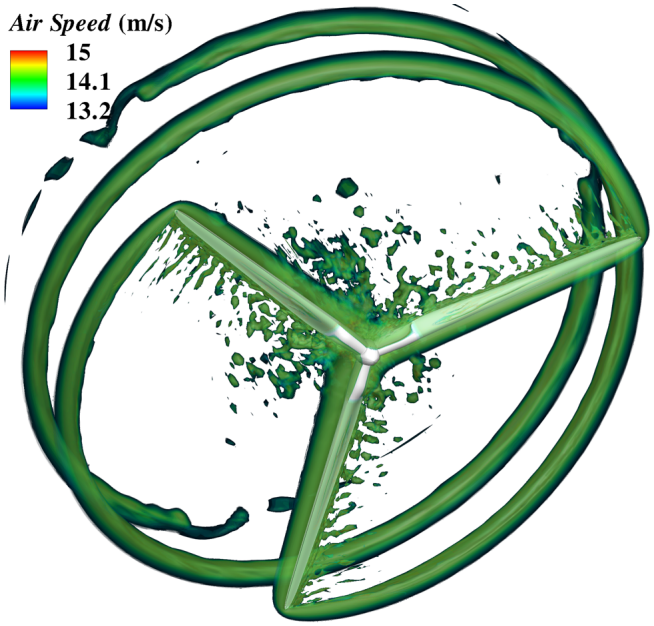
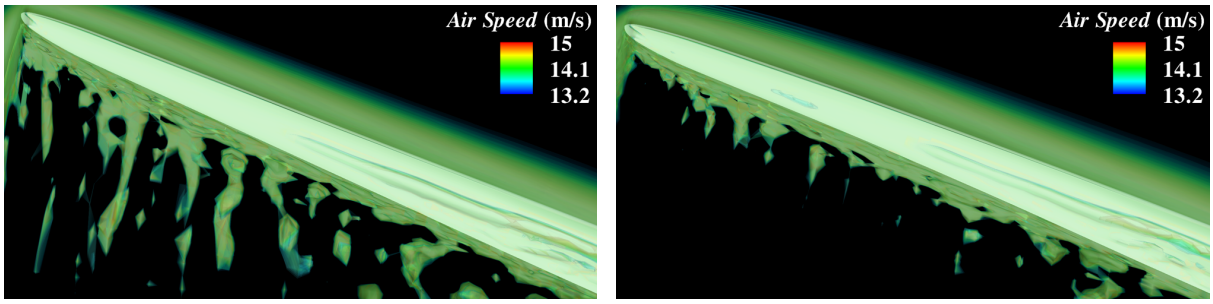



Figure 8: Isosurfaces of air speed at an instant in the wind turbine simulation. The flow exhibits complex behavior. The vortical feature generated at the blade tip is convected downstream of the rotor with very little decay.

the computational domain to illustrate the mesh resolution. The mesh is clustered toward the rotor blade. Figure 7 shows 2D blade cross-sections at $0.6R$ and $0.75R$, where $R$ is the rotor radius, to illustrate the boundary layer mesh used in our computation. Near the blade surface, the size of the first element in the wall-normal direction is about 2 cm.

Isosurfaces of the air speed at a time instant is shown in Figure 8. The vortex forming at the tip of the blades is convected downstream of the rotor with little decay, which is attributable to the use of NURBS functions. We note that the problem setup and mesh of this test case are identical to those reported in [8], where a fully-coupled FSI simulation was performed. Figure 9 presents a comparison of the trailing-edge turbulence for rigid and flexible blade cases. Note the




(a) (b)

Figure 9: Isosurfaces of air speed zoomed in on the wind turbine rotor blade tip. Comparison of the trailing-edge turbulence for (a) rigid and (b) flexible blade cases.

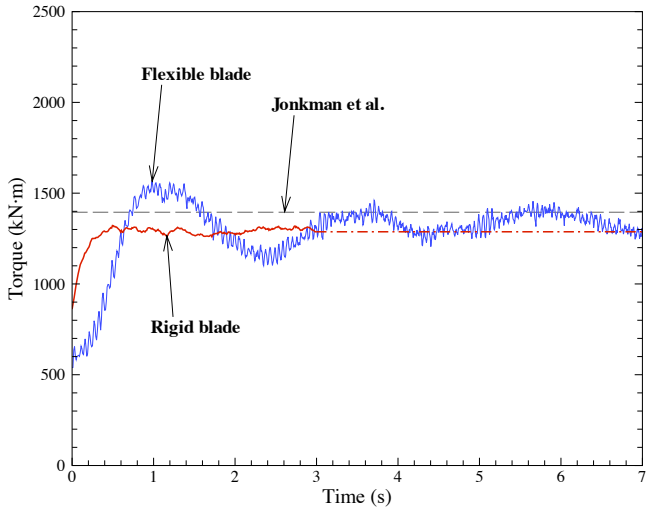



Figure 10: Time history of the aerodynamic torque. Both rigid and flexible rotor results are plotted. The reference result reported by Jonkman *et al.* [63] is also shown for comparison.

significant differences in the turbulence structures, which seem to be more fine-grained in the case of the rigid blade. Also note the presence of elongated streak-like flow structures coming off the trailing edge of the rigid blade, which are not present in the case of the flexible blade. It appears that the rotor structural deformation is influencing the generation of turbulence, and the effect is more pronounced near the blade tip where the relative air speed is higher than near the hub. The observed differences in the flow structures between the rigid and flexible blade cases suggest that FSI modeling may be important for accurate prediction of aerodynamically-generated noise.

The time history of the aerodynamic torque (for a single blade) is plotted in Figure 10 for both rigid and flexible blade simulations. Both cases compare favorably to the data reported in [63] obtained using FAST [66], which is the gold standard in the wind turbine aerodynamics simulation. Computational modeling in FAST makes use of look-up tables to obtain steady-state lift and drag data for airfoil cross-sections and incorporates empirical modeling to account

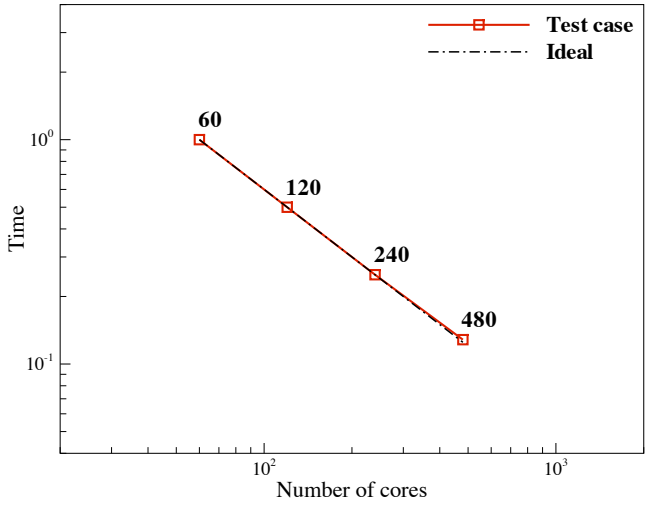



Figure 11: Scalability study for wind turbine rotor simulation. The computation time is normalized by the result of 60-processor case.

for complex effects such as trailing-edge turbulence. Nevertheless, we feel that the value of aerodynamic torque reported in [63] is close to what it is in reality, given the vast experience of NREL with wind turbine rotor simulations employing FAST. In our simulations we are able to capture this important quantity of interest using the proposed procedures, which do not rely on empiricism and are 100% predictive.

Note that the aerodynamic torque for the flexible blade exhibits low-magnitude high-frequency oscillations, while the rigid blade torque is smooth (see Figure 10). This is due to the twisting motion of the wind turbine blade about its axis as examined in [8]. The blade twist angle undergoes high frequency oscillations, which are driven by the trailing-edge vortex shedding and turbulence. Local oscillations of the twist angle lead to the temporal fluctuations in the aerodynamic torque.

As in the case of the Taylor–Couette flow, we assess the scalability of our aerodynamics code (without FSI) on the wind turbine simulation. The wind turbine mesh is decomposed into 60, 120, 240 and 480 subdomains. Five and six partitions were created in the downstream and radial directions, respectively, for all cases. In the azimuthal direction the mesh is decomposed into 2, 4, 8 or 16 partitions. The scalability test results are shown in Figure 11. Near-perfect linear scaling is likewise achieved.

## 5. Conclusions

This paper presents a computational framework for advanced flow simulation that is based on the RBVMS turbulence modeling and isogeometric analysis. Particular emphasis is placed on the parallel implementation of the methodology and scalability results. Near-perfect linear parallel scaling is shown on two challenging flow cases: the turbulent Taylor–Couette flow and the NREL 5MW offshore baseline wind turbine rotor at full scale. For wind turbine simulation the results of aerodynamic torque, a key quantity in evaluating the wind turbine performance,

were in close agreement with those reported in [63]. 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.

Recent advances in HPC and hardware design enable turbulence simulations with much finer grid resolution. Nowadays, in order to have confidence in the computational results for turbulent flows, several levels of mesh refinement are employed to achieve grid independence in the predicted quantities of interest (e.g. lift, drag, torque, turbulent kinetic energy, etc.). In the RBVMS method, the same discrete formulation is employed at all levels of mesh resolution. This is in contrast to standard turbulence modeling approaches, which use different turbulence models (of RANS, LES or DNS variety) depending on mesh resolution. As a result, in the RBVMS framework, one is not faced with the difficulties of selecting an appropriate turbulence model for a given level of mesh resolution, or trying to couple different turbulence models in different parts of the computational domain. This makes our methodology advantageous for large-scale simulations of turbulent flow that rely on mesh refinement to produce converged flow statistics and other quantities of interest.

## Acknowledgement

We 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. I. Akkerman is supported in part by an appointment to the Postgraduate Research Participation Program at the U.S. Army Engineering Research and Development Center, Coastal and Hydraulics Laboratory (ERDC-CHL) administered by the Oak Ridge Institute for Science and Education through an interagency agreement between the U.S. Department of Energy and ERDC. Y. Bazilevs would like to acknowledge the support of the Hellman Fellowship.

## References

[1] E. Echavarria, B. Hahn, G.J.W. van Bussel, and T. Tomiyama. Reliability of wind turbine technology through time. *Journal of Solar Energy Engineering*, 130:031005–1–8, 2008.

[2] C. Tongchitpakdee, S. Benjanirat, and L.N. Sankar. Numerical simulation of the aerodynamics of horizontal axis wind turbines under yawed flow conditions. *Journal of Solar Energy Engineering*, 127:464–474, 2005.

[3] N. Sezer-Uzol and L.N. Long. 3-D time-accurate CFD simulations of wind turbine rotor flow fields. *AIAA Paper 2006-0394*, 2006.

[4] F. Zahle and N.N. Sørensen. Overset grid flow simulation on a modern wind turbine. *AIAA Paper 2008-6727*, 2008.

[5] M.A. Potsdam and D.J. Mavriplis. Unstructured mesh CFD aerodynamic analysis of the NREL Phase VI rotor. *AIAA Paper 2009-1221*, 2009.

[6] C. Kong, J. Bang, and Y. Sugiyama. Structural investigation of composite wind turbine blade considering various load cases and fatigue life. *Energy*, 30:2101–2114, 2005.

[7] M.O.L. Hansen, J.N. Sørensen, S. Voutsinas, N. Sørensen, and H.Aa. Madsen. State of the art in wind turbine aerodynamics and aeroelasticity. *Progress in Aerospace Sciences*, 42:285–330, 2006.

[8] Y. Bazilevs, M.-C. Hsu, J. Kiendl, R. Wüchner, and K.-U. Bletzinger. 3D simulation of wind turbine rotors at full scale. Part II: Fluid–structure interaction modeling with composite blades. *International Journal for Numerical Methods in Fluids*, 65:236–253, 2011.

[9] 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.

[10] 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.

[11] 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.

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

[13] 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.

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

[15] 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.

[16] 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.

[17] T.J.R. Hughes, G. Scovazzi, and L.P. Franca. Multiscale and stabilized methods. In E. Stein, R. de Borst, and T.J.R. Hughes, editors, *Encyclopedia of Computational Mechanics, Vol. 3, Fluids*, chapter 2. Wiley, 2004.

[18] 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.

[19] Y. Bazilevs, M.-C. Hsu, I. Akkerman, S. Wright, K. Takizawa, B. Henicke, T. Spielman, and T.E. Tezduyar. 3D simulation of wind turbine rotors at full scale. Part I: Geometry modeling and aerodynamics. *International Journal for Numerical Methods in Fluids*, 65:207–235, 2011.

[20] T.J.R. Hughes, L. Mazzei, and K.E. Jansen. Large-eddy simulation and the variational multiscale method. *Computing and Visualization in Science*, 3:47–59, 2000.

[21] B. Koobus and C. Farhat. A variational multiscale method for the large eddy simulation of compressible turbulent flows on unstructured meshes – application to vortex shedding. *Computer Methods in Applied Mechanics and Engineering*, 193:1367–1383, 2004.

[22] R. Codina, J. Principe, O. Guasch, and S. Badia. Time dependent subscales in the stabilized finite element approximation of incompressible flow problems. *Computer Methods in Applied Mechanics and Engineering*, 196:2413–2430, 2007.

[23] G. Houzeaux and J. Principe. A variational subgrid scale model for transient incompressible flows. *International Journal of Computational Fluid Dynamics*, 22:135–152, 2008.

[24] P. Gamnitzer, V. Gravemeier, and W.A. Wall. Time-dependent subgrid scales in residual-based large eddy simulation of turbulent channel flow. *Computer Methods in Applied Mechanics and Engineering*, 199:819–827, 2010.

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

[26] 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.

[27] 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.

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

[29] H. Gomez, V.M. Calo, Y. Bazilevs, and T.J.R. Hughes. Isogeometric analysis of the Cahn–Hilliard phase-field model. *Computer Methods in Applied Mechanics and Engineering*, 197:4333–4352, 2008.

[30] 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.

[31] J. Kiendl, Y. Bazilevs, M.-C. Hsu, R. Wüchner, and K.-U. Bletzinger. The bending strip method for isogeometric analysis of Kirchhoff–Love shell structures comprised of multiple patches. *Computer Methods in Applied Mechanics and Engineering*, 199:2403–2416, 2010.

[32] H. Gomez, T.J.R. Hughes, X. Nogueira, and V.M. Calo. Isogeometric analysis of the isothermal Navier–Stokes–Korteweg equations. *Computer Methods in Applied Mechanics and Engineering*, 199:1828–1840, 2010.

[33] L. Piegl and W. Tiller. *The NURBS Book (Monographs in Visual Communication), 2nd ed.* Springer-Verlag, New York, 1997.

[34] 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.

[35] 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.

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

[37] 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.

[38] 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.

[39] 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.

[40] 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.

[41] D.J. Benson, Y. Bazilevs, E. De Luycker, M.-C. Hsu, M. Scott, T.J.R. Hughes, and T. Belytschko. A generalized finite element formulation for arbitrary basis functions: from isogeometric analysis to XFEM. *International Journal for Numerical Methods in Engineering*, 83:765–785, 2010.

[42] 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.

[43] 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.

[44] 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.

[45] Y. Bazilevs, J.R. Gohean, T.J.R. Hughes, R.D. Moser, and Y. Zhang. Patient-specific isogeometric fluid-structure interaction analysis of thoracic aortic blood flow due to implantation of the Jarvik 2000 left ventricular assist device. *Computer Methods in Applied Mechanics and Engineering*, 198:3534–3550, 2009.

[46] 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.

[47] Y. Bazilevs, C. Michler, V.M. Calo, and T.J.R. Hughes. Isogeometric variational multiscale modeling of wall-bounded turbulent flows with weakly-enforced boundary conditions on unstretched meshes. *Computer Methods in Applied Mechanics and Engineering*, 199:780–790, 2010.

[48] 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.

[49] T.J.R. Hughes, A. Reali, and G. Sangalli. Efficient quadrature for NURBS-based isogeometric analysis. *Computer Methods in Applied Mechanics and Engineering*, 199:301–313, 2010.

[50] 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.

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

[52] 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.

[53] 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*, 199:229–263, 2010.

[54] 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.

[55] M.J. Borden, M.A. Scott, J.A. Evans, and T.J.R. Hughes. Isogeometric finite element data structures based on Bézier extraction of NURBS. *International Journal for Numerical Methods in Engineering*, 2010. doi:10.1002/nme.2968.

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

[57] A.K. Karanam, K.E. Jansen, and C.H. Whiting. Geometry based pre-processor for parallel fluid dynamic simulations using a hierarchical basis. *Engineering with Computers*, 24:17–26, 2008.

[58] O. Sahni, M. Zhou, M.S. Shephard, and K.E. Jansen. Scalable implicit finite element solver for massively parallel processing with demonstration to 160K cores. In *Proceedings of the ACM/IEEE Conference on High Performance Computing*, Portland, OR, USA, 2009.

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

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

[61] Texas Advanced Computing Center (TACC). http://www.tacc.utexas.edu. 2010.

[62] Ranger User Guide. http://services.tacc.utexas.edu/index.php/ranger-user-guide. 2010.

[63] 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.

[64] K. Takizawa, C. Moorman, S. Wright, T. Spielman, and T.E. Tezduyar. Fluid–structure interaction modeling and performance analysis of the Orion spacecraft parachutes. *International Journal for Numerical Methods in Fluids*, 65:271–285, 2010.

[65] K. Takizawa, S. Wright, C. Moorman, and T.E. Tezduyar. Fluid–structure interaction modeling of parachute clusters. *International Journal for Numerical Methods in Fluids*, 65:286–307, 2010.

[66] 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.