ExaWind: A multifidelity modeling and simulation environment for wind energy

M A Sprague, S Ananthan, G Vijayakumar, and M Robinson
National Renewable Energy Laboratory, Golden, CO, USA
E-mail: `michael.a.sprague@nrel.gov`

Abstract. We introduce the open-source ExaWind modeling and simulation environment for wind energy. The primary physics codes of ExaWind are Nalu-Wind and OpenFAST. Nalu-Wind is a wind-focused computational fluid dynamics (CFD) code that is coupled to the whole-turbine simulation code OpenFAST. The ExaWind environment was created under U.S. Department of Energy funding to achieve the highest-fidelity simulations of wind turbines and wind farms to date, with the goal of enabling disruptive changes to turbine and plant design and operation. Innovation will be gleaned through better understanding of the complex flow dynamics in wind farms, including wake evolution and the impact of wakes on downstream turbines and turbulent flow from complex terrain. High-fidelity predictive simulations employ hybrid turbulence models, geometry/boundary-layer-resolving CFD meshes, atmospheric turbulence, nonlinear structural dynamics, and fluid-structure interaction. While there is an emphasis on very high-fidelity simulations (e.g., blade resolved with full fluid-structure coupling), the ExaWind environment supports lower-fidelity modeling capabilities including actuator-line and -disk methods. Important in the development of ExaWind codes is that the codes scale well on today's largest petascale supercomputers and on the next-generation platforms that will enable exascale computing.

1. Introduction

A key to achieving wide-scale deployment of wind energy is enabling a new understanding of, and ability to predict, the fundamental flow physics and coupled structural dynamics governing whole wind plant performance, including wake formation, complex-terrain impacts, and turbine-turbine interactions through wakes. Based on an improved understanding of the driving flow physics and interactions with turbine and plant structures, new technology innovations can be proposed to advance performance and resiliency. High-fidelity modeling (HFM), coupled with high-performance computing (HPC), offers a potential path to drive significant reductions in the cost of wind energy by providing researchers and engineers with a virtual environment for exploring technology innovations and new operational strategies with confidence.

In early 2015, the U.S. Department of Energy (DOE) Wind Energy Technologies Office sponsored a strategic-planning meeting [1] at which about 70 participants from industry, academia, and national laboratories were challenged to define the requirements for an *open-source* modeling and simulation environment for wind turbines and plants. Guiding principles for the planning meeting were that the environment be the foundation for state-of-the-art predictive, physics-based simulations of whole wind plants; leverage existing software/library assets where appropriate; be designed to accommodate future exascale systems; target simulations that aspire to "ground truth"; and be an open-source community model. From our perspective, a *predictive* simulation capability:

- Employs mathematical models that are derived from, and adhere to, first principles,
- provides solutions to those mathematical models,
- provides user control of numerical-approximation errors,
- provides assessment of uncertainties in results,
- enables study of the fundamental behavior of the system.

This is an aspirational list, as some physics will require modeling approximations of first principles in order to solve the equations with practical resource requirements. For example, while the Navier-Stokes equations provide a first-principles model of turbulent fluid flow, the need to resolve wind turbine and plant length scales combined with the cascade of turbulent energy down to the Kolmogorov microscales present a daunting range of scales. Practical simulation times require that the smallest scales be modeled (or filtered) through, e.g., Reynolds averaging or large-eddy simulation (LES). Regardless of the modeling approach, verification, validation, and uncertainty quantification of simulation results are necessary to bound applicability, accuracy, and confidence.

Based on the viable modeling pathways determined at the 2015 planning meeting [1], we chose the following for our highest-fidelity capability: an acoustically incompressible fluid dynamics model, two-way-coupled fluid-structure interaction (FSI), hybrid Reynolds-averaged-Navier-Stokes/LES (RANS/LES) turbulence modeling, turbine-geometry-resolved fluid meshes with mesh-motion capabilities (e.g., overset meshes), nonlinear structural dynamics models (e.g., large blade deflections), and one-way coupling to weather-scale forcing via, e.g., numerical weather prediction. This modeling pathway is realized through a suite of open-source codes and libraries. Our primary physics-based codes are Nalu-Wind and OpenFAST, which are for fluid dynamics and turbine dynamics, respectively, and which are based on a number of libraries described in Section 2. We refer to our software stack as ExaWind, which acknowledges the goal of enabling efficient simulation on next-generation computer architecture, including that of the first exascale systems [2]. While ExaWind is focused on enabling simulations of the highest fidelity, a range of fidelity options is available, depending upon the dominant physics of interest. Lower-fidelity, but computationally affordable models are key for high-throughput calculations required for uncertainty quantification and exploration of parameter spaces.

The choice for open-source software development and deployment is motivated by the desire for transparency and broad community engagement. It is the hope that the open-source approach will accelerate sharing and adoption of ideas across the wind energy community including research institutions, industry, and commercial software developers.

Key motivations of this paper are to introduce the ExaWind software stack to the wind energy community, and to document its key features and planned enhancements. The paper is organized as follows. Section 2 describes at a high level the models and codes currently in the ExaWind software stack. Section 3 describes the open-source ExaWind environment. Section 4 describes our preparation for next-generation computer architectures. Section 5 presents preliminary results, and Section 6 provides a summary and planned development.

2. Models, algorithms, and codes/libraries

The ExaWind software stack is a collection of integrated, physics-based solvers supported by a number of libraries for, e.g., solving linear systems. As noted, the primary physics-based codes are Nalu-Wind and OpenFAST. Key features and applicable references are described in this section.

2.1. Nalu-Wind

Nalu-Wind is an open-source computational fluid dynamics (CFD) code written in C++, and it is a wind-specific version of the Nalu code [3], which is a large-eddy-simulation research code developed at Sandia National Laboratories. Nalu (and in turn, Nalu-Wind) is open source, it leverages well-supported open-source libraries (e.g., Trilinos [4]), it was demonstrated to scale well on large HPC systems [5], and it was developed with modern software engineering best practices including rigorous code verification.

Nalu-Wind employs an unstructured-grid, finite-volume method for spatial discretization and solves the acoustically incompressible Navier-Stokes equations for which mass continuity is maintained through approximate pressure projection. Two finite-volume formulations are provided: an edge-based method and a control-volume finite element method (CVFEM). Nalu-Wind contains the infrastructure for discretization of the underlying models, and heavily utilizes the Trilinos [4] Sierra Toolkit (STK) [6], providing an unstructured-mesh in-memory parallel-distributed database.

A key challenge to blade-resolved simulation of wind turbines is the need to handle meshes undergoing general large-scale motions. In addition to the rotor rotation, the nacelle/rotor yaws, blades undergo large deflections, and the whole nacelle-rotor system effectively moves because of tower bending. Floating offshore turbines have additional complexity caused by large platform motions. While early team efforts focused on a sliding-mesh approach [7], the overset-mesh method has become preferred, for which meshes around each turbine component (e.g., each blade, nacelle, and tower) can be created independently. In the ExaWind stack, mesh connectivity and constraints are created with the Topology Independent Overset Grid Assembler (TIOGA)[1]. Under this connectivity, turbine components can undergo large deformations and arbitrarily large rigid-body motions. With moving meshes comes the significant cost associated with mesh searches to build the connectivity between mesh points and the need to rebuild all of the matrices (associated with discretization of the governing equations) at *every* time step. Unlike static-mesh simulations, for which mesh- and matrix-creation costs can be amortized over the simulation, these every-time-step costs must be minimized for efficient simulations.

The governing equations for momentum (velocity), pressure, and scalar quantities (e.g., temperature) are discretized in time with a split-operator approach and with either a first- or second-order backwards-differentiation formula (BDF). An "outer-loop" (i.e., Picard) iteration surrounds a linearized momentum equation solve, an approximate pressure-projection equation (i.e., pressure-Poisson equation) solve to maintain continuity, and any relevant scalar-equation solves. Multiple outer-loop iterations are enabled to reduce the nonlinear residual at each time step.

To enable robust RANS, hybrid-RANS/LES, or detached-eddy simulations (DES), for which near body RANS-region meshes include elements with large aspect ratios (e.g., $O(10^5)$) and large local Courant-Friedrich-Lewy (CFL) numbers, CFL $\gg 1$, the time-stepping algorithm in Nalu-Wind was modified (from the base algorithm inherited from the Nalu code) based on the approach described in Sørensen [8]. The modified algorithm introduces two changes to the base algorithm: 1. The projection timescale is approximated as the inverse of the diagonal term of the momentum linear system, and 2. The system is under-relaxed to increase the diagonal dominance of the linear system at large time steps. Finally, the full pressure update is used for the velocity and mass-flux updates, but the pressure solution is under-relaxed at each outer-loop Picard iteration. Details can be found in the Nalu-Wind documentation[2].

In order to resolve a wide range of spatial length scales and cater to different applications, Nalu-Wind is equipped with different turbulence models. The codebase has capabilities to use RANS, DES, or LES models. Currently, Nalu-Wind supports the k-ω SST RANS model [9],

[1] `https://github.com/jsitaraman/tioga`
[2] `https://nalu-wind.readthedocs.io`

a model based on blending the k-ω and k-ϵ RANS models to leverage the advantages of the ω treatment near the wall and the ϵ treatment in the free stream. Nalu-Wind has the SST-DES model [10] to model separated flows. The key aspect of this model is to relax the RANS model and allow the CFD solver to partially resolve the turbulent flow away from the pure RANS region using LES if the grid resolution allows for it. For LES modeling of turbulent flows, the code is equipped with the standard Smagorinsky model, wall-adapting local eddy viscosity (WALE) model, and subgrid-scale kinetic-energy one-equation k^{sgs} model used for atmospheric-boundary-layer flows [11]. The code also includes the necessary wall functions that use the surface roughness height and surface heat flux to calculate the appropriate shear stress at the wall.

As is typical of CFD simulations, the vast majority of simulation time is spent solving the linear systems (at every time step) that are associated with the underlying spatial and temporal discretization. The Nalu-Wind CFD solver has been equipped to utilize the linear solvers and preconditioners in the Trilinos software stack and/or those in the *hypre*[3] solver library [12].

2.2. OpenFAST

OpenFAST is a whole-turbine-simulation code written in Fortran 2003 that grew out of FAST version 8 [13]. OpenFAST employs a modularization framework that facilitates the choice of different models for particular turbine components. OpenFAST contains a collection of physics modules necessary for modeling a turbine, including the turbine control system, a model for tower bending deformation, and a high-order nonlinear finite-element model called BeamDyn [14] for blade dynamics, which is based on geometrically exact beam theory. Also included are a number of reduced-order models for aerodynamics and offshore wind, including hydrodynamics and support structures. In regard to blade modeling, nonlinear beam models can capture the dynamics of modern wind turbine blades (see, e.g., [15]) for which the complex material layups and cross sections are modeled through two-dimensional sectional mass and stiffness matrices.

OpenFAST is primarily focused on time-domain simulations. Through the modularization framework [13], physics modules can have independent time-update algorithms (either implicit or explicit), use different time-step sizes, and interact through nonmatching spatial meshes. Details regarding discrete-time and -space coupling can be found in [16–18].

2.3. Fluid-structure interaction

The ExaWind software stack provides the capability to simulate wind turbines under realistic inflow conditions with fluid-structure interaction by coupling the Nalu-Wind and OpenFAST codes as illustrated in Figure 1, which shows loose coupling and the data types transferred between models.

Nalu-Wind allows for multifidelity simulations with both actuator-line methods (like those described in [19; 20]) and turbine-resolved simulations. Actuator methods represent the effect of the wind turbine on the flow field using a series of body forces, whereas turbine-resolved simulations resolve the geometry of the blades, tower, and nacelle and exchange information with OpenFAST at the surface boundaries. In actuator methods, Nalu-Wind provides the fluid velocities at the actuator points and OpenFAST computes the response of the turbine as a whole to provide displacements and forces at the actuator nodes. In blade-resolved simulations, Nalu-Wind provides the loads on the blades, nacelle, and tower to OpenFAST, while OpenFAST provides the deformations and velocities to Nalu-Wind.

OpenFAST models the blades and the tower as slender beams along with point masses for the nacelle and hub. For blade-resolved FSI simulations, Nalu-Wind provides surface-line and line-surface mapping algorithms to transfer the loads and deflections between the line/point and

[3] https://github.com/hypre-space/hypre

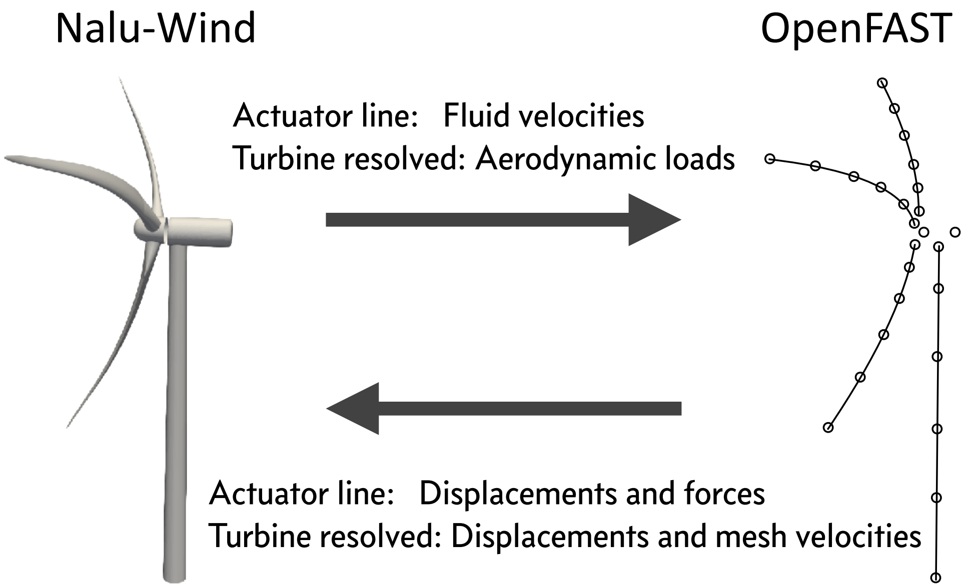


Figure 1: Overview of the fluid-structure-interaction framework for a Nalu-Wind fluid model coupled to an OpenFAST turbine model. The arrows describe data types transferred between models in a loosely coupled simulation for both turbine-resolved and actuator-line representations of the turbine in the fluid.

the surface representation of the turbine structure. The mapping algorithms work in parallel across several processors.

We use the conventional-serial-staggered algorithm for fluid-structure interaction [21], along with a specified number of "outer" or nonlinear iterations to couple Nalu-Wind and OpenFAST. There is also the option for time-step subcycling, for which the structural time step can be smaller than that of the fluid. Each fluid-structure-coupling outer iteration can encompass a number of Picard iterations of the fluid solver in order to reduce the nonlinear residual of the momentum equation. Guidance on the number of nonlinear iterations as well as a convergence criterion will be developed in the future for wind energy problems.

3. Community modeling and simulation environment

3.1. Software repositories, testing, and documentation

Nalu-Wind and OpenFAST are developed in the open domain, are under Git software version control and hosted on GitHub (`https://github.com/exawind/nalu-wind`, `https://github.com/openfast/openfast`). Contributions to the software can be made readily by external and internal collaborators through "pull requests." The GitHub issue tracker is employed to submit and respond to usage questions, bug reports, and feature requests. All codes undergo nightly automated regression and unit testing, for which reports are posted publicly (`https://my.cdash.org/index.php?project=Nalu-Wind`, `https://my.cdash.org/index.php?project=OpenFAST`). Finally, code documentation resides with the codes (on GitHub), and is meant to evolve with the code (`https://nalu-wind.readthedocs.io`, `https://openfast.readthedocs.io`).

3.2. Verification and validation

The ExaWind team strives to perform rigorous verification of its codes through comparison of solutions to analytical or manufactured solutions [22]. With verification solutions in hand, researchers can demonstrate for a set of problems that errors converge as expected, thereby giving developers confidence that numerical algorithms are implemented properly (and bug free).

For model validation, a series of benchmark problems is being defined to assess the progression of predictive ExaWind capabilities under DOE funding. Code validation will start with well-described, well-understood aerodynamic fundamentals (e.g., fixed airfoils and wings) historically used in aerospace validation, increasing in complexity with fully resolved single-turbine simulations, and concluding with capstone multiturbine wind farm simulations. The ability to validate model performance in the area of wake dynamics and vortical flow behavior (e.g., formation, evolution, merging, dissipation), and the effect of wake dynamics on turbine-centric quantities of interest (e.g., power, loads) are the key science and engineering modeling challenges of interest. Benchmark problems will be fully defined with sufficient specificity to duplicate the simulations for validation by external code developers and placed in public domain for easy access and download. Results from the ExaWind codes will be posted along with performance analysis metrics describing computational efficiency and accuracy obtained from each benchmark simulation. Other institutions, domestic and international, will be invited to post their results in the open forum as well. Our intent is to provide a suite of clearly defined computational challenges for the wind community that will both facilitate code development and provide an easily accessible resource to enable code-to-code comparisons and independent experimental data validation for the wind community.

4. Next-generation high-performance computing

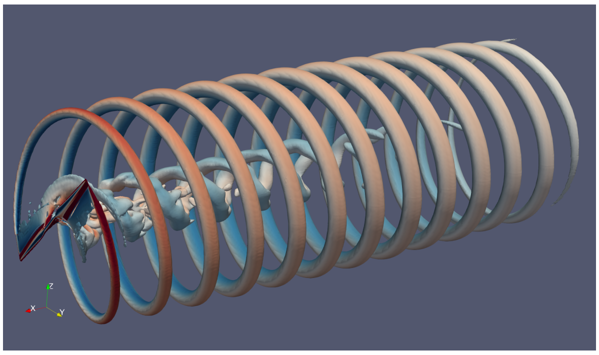
Developers of codes like Nalu-Wind, which require massively parallel supercomputers for their target applications, must be informed by the transition to next-generation, power-efficient computer architectures that will enable exascale class computing [2; 23]. For example, the latest DOE supercomputer, Summit, employs graphical processing units (GPUs) in addition to traditional CPUs. In order to get competitive allocations on such systems, proposals require demonstration of effective use of GPUs. However, enabling CFD codes to run effectively on GPUs is no small task. Nalu-Wind and Trilinos developer teams are actively preparing for next-generation architectures, like GPUs, with Kokkos[4], a parallel-performance abstraction layer. The Nalu-Wind team is also working closely with the *hypre* team in preparing it for effective use of GPUs.

5. Example results

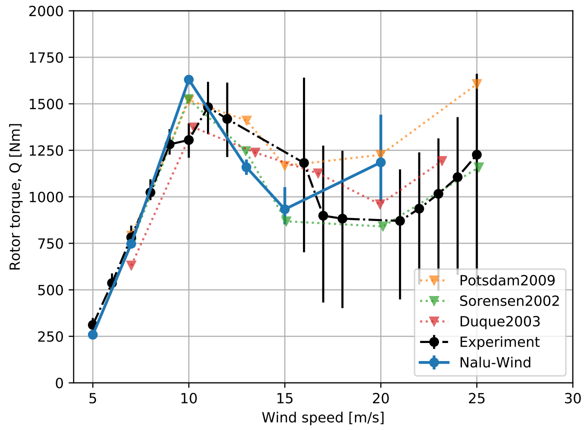
5.1. NREL UAE Phase VI rotor

In this section we describe preliminary validation of the blade-resolved, overset-mesh simulation capability in Nalu-Wind by comparing simulation results against those from the NREL Unsteady Aerodynamics Experiment (UAE) Phase VI [24]. For this study, simulations were performed to match the run conditions of the Test Sequence H, the upwind baseline configuration, for which the blades were rigid (i.e., no teeter of the two-bladed configuration) with zero blade coning. The blade pitch was set at 3° and the rotor was run at a fixed speed of 72 rpm for all wind speeds. Simulations were performed for the zero-yaw condition at six different wind speeds: 5, 7, 10, 13, 15, and 20 m/s, respectively. The turbine geometry was simplified for the simulation by neglecting the tower, nacelle, and the aerodynamic interference effects arising from the instrumentation near the rotor hub. Furthermore, the hub section of the two-bladed rotor

[4] https://github.com/kokkos/kokkos



(a) Flowfield



(b) Torque

Figure 2: (a) Flowfield (isosurfaces of Q-criterion colored by vorticity magnitude) for the NREL Phase VI rotor operating in uniform inflow with velocity of 7 m/s and (b) rotor torque a function of inflow velocity as predicted by Nalu-Wind, measurements [24], and other codes [25–27].

was idealized as a cylindrical connecting rod that joined the two blades in the computational model.

The boundary-layer resolving, near-body mesh was embedded in a structured, hexahedral-element-only, cylindrical wake-capturing mesh with an O-H (or "butterfly") topology. The cylindrical mesh extended half a rotor diameter upstream and 5 diameters downstream. The section of the cylindrical mesh around the near-body mesh had constant spacing in the flow direction up to half a diameter upstream and downstream, and mesh stretching was introduced in the flow direction further downstream. The wake-capturing mesh was embedded inside a fully unstructured mesh that covered the rest of the domain that had extended 5 diameters upstream and 10 diameters downstream and in lateral directions.

Simulations were performed in a fixed reference frame and rotor rotation was simulated by rotating the near-body mesh at each time step. This required re-computation of the overset domain connectivity and the reinitialization of the linear systems and preconditioners at each time step. Calculations used the $k-\omega$ SST RANS turbulence model. A fixed time-step size was chosen such that the rotor blade would rotate 0.25° per time step (10^{-4} s), and one rotor revolution would require 1440 time steps to complete. At least 12 rotor revolutions were simulated at each wind speed to achieve statistical convergence of the integrated thrust and power for the turbine before comparison with experiments.

Figure 2a shows the flow field after 15 revolutions for a uniform inflow of 7 m/s. Figure 2b shows the rotor-torque predictions as a function of wind speed in comparison with measurements (shown in black) and other simulations from the literature [25–27]. The black vertical bars about the measurements indicate the variation in the measurements over the duration that data were collected. The torque predictions show good agreement with measurements and other computational results for the low wind speeds. In this regime, the flow is mostly attached across the entire blade span, and the experimental measurements show very little deviation from the mean values. At higher wind speeds (> 10 m/s), the measurements show significant variation; this is a result of flow separation and stall in the inboard sections of the blade. In this regime, there is greater mismatch between Nalu-Wind-computed torque values and the experimental data, as well as other computed results.

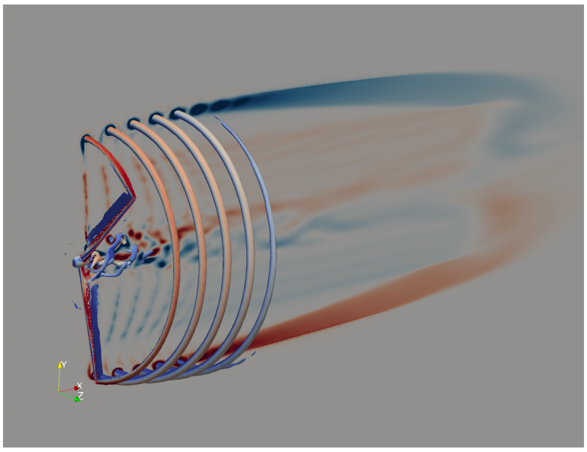


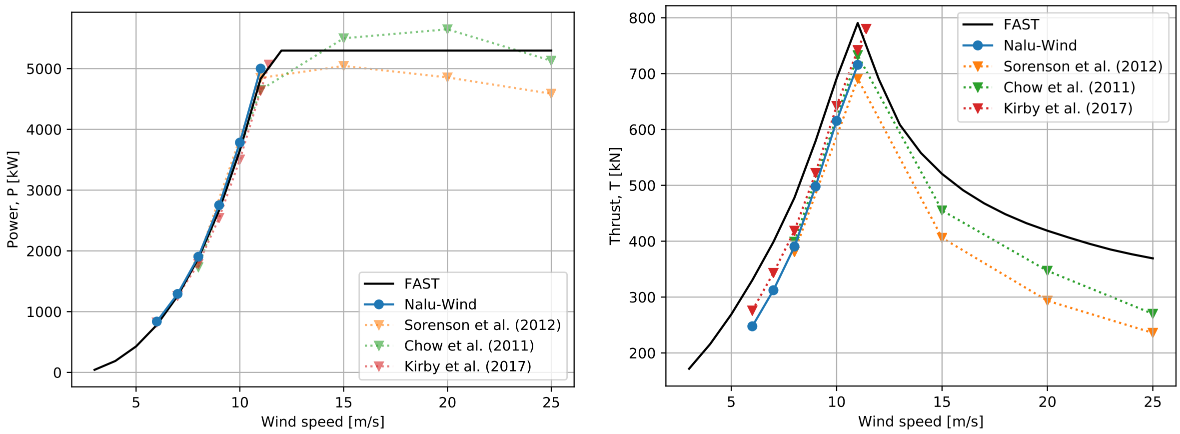
Figure 3: Flowfield (isosurfaces of Q-criterion colored by vorticity magnitude and a plane with vorticity-magnitude isocontours) for the NREL 5-MW rotor with rigid blades operating in uniform inflow of 8 m/s.

5.2. NREL 5-MW turbine with rigid blades

The NREL 5-MW turbine [28] is a 126 m diameter reference turbine, designed for use in research of offshore wind. It is a notional turbine that is widely used in the wind research community and thus provides a good baseline for studying code capabilities and performing code-to-code comparisons with other simulations published in the literature. For the purposes of this study, the turbine geometry was simplified in that only the three blades and the hub were modeled.

Meshing best-practices from the Phase VI turbine study were used to generate the blade surface mesh. In order to transition smoothly to the hub structure, the structured mesh on the blade surface was constructed outboard of the 20% span. The sections inboard used unstructured mesh to transition smoothly to the hub mesh. Like the Phase VI simulations, the near-body mesh was embedded in a wake-capturing mesh that extended half a rotor diameter upstream and about 5 rotor diameters downstream. The wake-capturing mesh was enclosed within a fully unstructured mesh that formed the outer domain. The overall computational domain extended 5 rotor diameters upstream, 10 diameters downstream, and 10 diameters in the lateral directions. The mesh contained a total of 38 million elements (23 million nodes), and the near-body mesh contained 7 million elements for all three blades.

Simulations were performed with a fixed time-step size such that the rotor rotated 0.25° at each time step (for the particular constant rpm at each wind speed). Computations were performed on the NREL Eagle HPC system with 1080 Message-Passing Interface (MPI) ranks (30 compute nodes). Figure 3 shows the flowfield (isocontours of Q-criterion and vorticity contours) for the NREL 5-MW rotor operating at uniform inflow of 8 m/s. The qualitative flow structures are similar to those observed in the Phase VI results with the tip vortex dissipating quickly with coarsening of the mesh in the wake region. Figures 4a and 4b show the NREL 5-MW turbine power curve and thrust, respectively, for Nalu-Wind predictions compared to other simulation results [29–31] alongside results from FAST blade element momentum (BEM) theory simulations [28]. Nalu-Wind simulations were only performed for wind speeds below rated wind speed (11.4 m/s) because there was no controller active, and pitch control is necessary for relevant simulations above rated wind speed. Results show good agreement with the other CFD simulations published in literature and provide confidence in the capability of Nalu-Wind simulations to predict the performance of megawatt-scale rotors operating in uniform inflow.

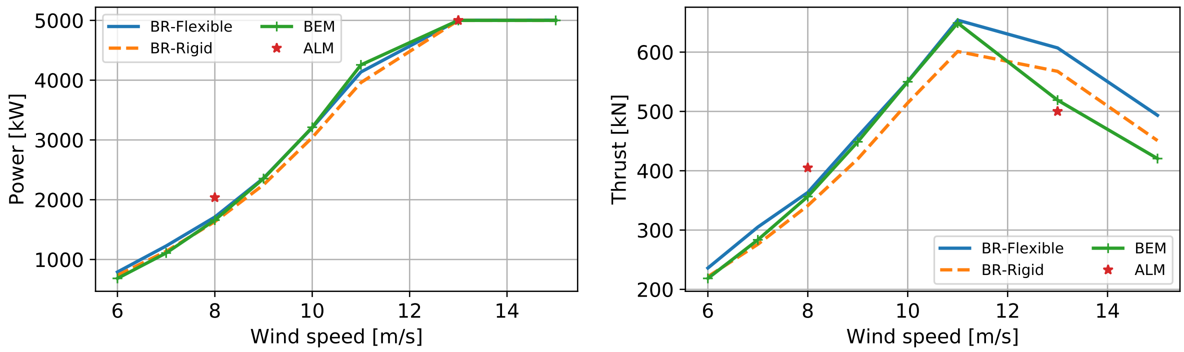


(a) Power (b) Thrust

Figure 4: Predictions of (a) rotor power and (b) thrust from blade-resolved, overset-mesh simulations of the NREL 5-MW rotor with rigid blades compared to FAST BEM simulations and other blade-resolved simulations from the literature [29–31].

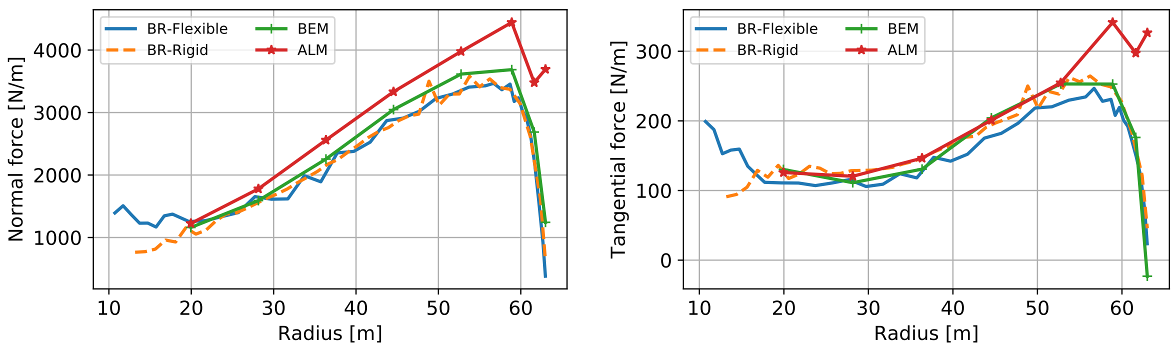
5.3. NREL 5-MW turbine with flexible blades and FSI

We describe here a demonstration of the fluid-structure-interaction capabilities in the ExaWind framework through simulation of the NREL 5-MW turbine [28] in uniform inflow conditions. We compare the effects of fluid-structure interaction with simulations of different-fidelity models: BEM, actuator-line-method (ALM), and blade-resolved simulations. The OpenFAST model of the NREL 5-MW turbine uses BeamDyn to represent the blades with geometrically exact beam theory. The blade-resolved simulations use the $k-\omega$ SST RANS turbulence model along with other best practices learned from Section 5.1. The mesh used in these simulations is similar to that described in Section 5.2, with the exception of the rotor hub, which was removed to simplify the inclusion of independent and arbitrary pitch motion of each blade. We use two fluid-structure-coupling Picard iterations per time step, each having two fluid Picard iterations. The CFD code uses a fixed time step corresponding to a rotation 0.25° along with 4 sub-time-steps for OpenFAST. Simulations are run for 10 revolutions and the results shown in Figures 5-7 are averaged over the last revolution. Figure 5 shows a comparison of predicted generator power and thrust for the NREL 5-MW rotor using a blade-resolved model (with and without FSI effects), an actuator-line model, and a BEM model; the latter two included FSI effects. The power and thrust predicted by blade-resolved and actuator-line simulations compare well with the predictions from BEM before rated speed, but differ significantly above rated speed. The blade-resolved simulations with no structural deflection show a lower thrust compared to the simulations with deflections enabled as observed in the literature and shown in Figure 4b. Figures 6-7 show the normal and tangential force-per-unit span along the blade in Region II (8 m/s) and Region III (13 m/s). The oscillations in the line loads from blade-resolved simulations are due to the load mapping from a coarse CFD mesh to a line mesh with higher resolution along the span. However, the surface-to-line mapping process conserves the total force and moment across each blade and the entire rotor. We do not expect any differences due to the spanwise load oscillations on the blade response because of the stiff nature of the NREL 5-MW blades. The detailed effects of the spanwise oscillations in the mapped load on the full fluid-structure interaction response is being investigated further. In Region II, blade-resolved, ALM, and BEM forces are nominally equivalent, with largest difference in tip and root regions. As expected, the actuator-line simulations predict a significantly larger load near the tip region compared to all



(a) Generator power (b) Rotor thrust

Figure 5: Comparison of predicted (a) generator power and (b) rotor thrust for the NREL 5-MW rotor using a blade-resolved (BR) model with (Flexible) and without (Rigid) FSI, an ALM model with FSI, and a BEM model with FSI.



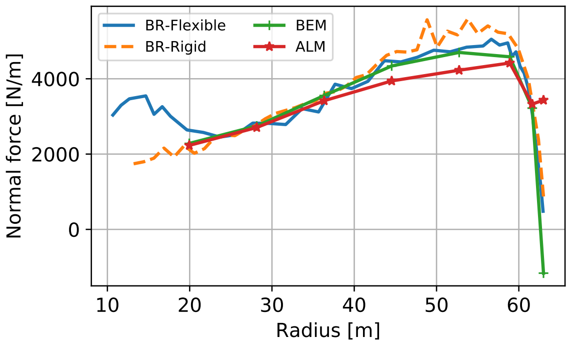
(a) Normal force (b) Tangential force

Figure 6: Comparison of predicted (a) normal and (b) tangential force-per-unit span for the NREL 5-MW rotor in uniform inflow of 8 m/s using a blade-resolved (BR) model with (Flexible) and without (Rigid) FSI, an ALM model with FSI, and a BEM model with FSI.

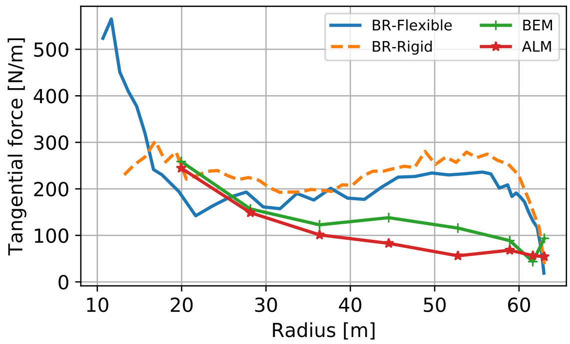
other simulations because of the isotropic spreading of body forces in the CFD simulation and lack of an explicit tip-loss model. In Region III, spanwise loading for blade-resolved simulations is significantly different compared to the BEM and ALM results.

6. Concluding remarks and next steps

We described in this paper the ExaWind software stack for wind turbine and wind plant simulations. Multiple levels of fidelity are provided, including turbine-resolved hybrid-RANS/LES capabilities with fluid-structure interaction and full turbine mobility. In regard to planned work, the ExaWind team is actively trying to minimize time to solution (i.e., time per time step) through optimizing time-step algorithms, optimizing linear-system solvers and preconditioners, and enabling the use of GPUs. In order to further reduce time to solution, there is a new effort examining the addition of a Cartesian structured-grid off-body background solver with adaptive-mesh-refinement (AMR) capabilities that will interface with Nalu-Wind as the near-body solver. Another significant new effort will be to implement high-fidelity hydrodynamics in Nalu-Wind for floating-offshore-turbine simulations. Verification and validation of ExaWind capabilities are ongoing and results will be published in the open domain.



(a) Normal force



(b) Tangential force

Figure 7: Comparison of predicted (a) normal and (b) tangential force-per-unit span for the NREL 5-MW rotor in uniform inflow of 13 m/s using a blade-resolved (BR) model with (Flexible) and without (Rigid) FSI, an ALM model with FSI, and a BEM model with FSI.

Acknowledgments
The authors would like to acknowledge members of the ExaWind and HFM teams and their contributions to the ExaWind software stack, and Stefan Domino, the developer of the Nalu code on which Nalu-Wind is based. This work was authored by the National Renewable Energy Laboratory, operated by Alliance for Sustainable Energy, LLC, for the U.S. Department of Energy (DOE) under Contract No. DE-AC36-08GO28308. Funding for Nalu-Wind/OpenFAST development is provided by the U.S. Department of Energy Office of Energy Efficiency and Renewable Energy Wind Energy Technologies Office and the Exascale Computing Project (17-SC-20-SC), a collaborative effort of two DOE organizations (Office of Science and the National Nuclear Security Administration). The research was performed using computational resources sponsored by the DOE Office of Energy Efficiency and Renewable Energy and located at the National Renewable Energy Laboratory. The views expressed in the article do not necessarily represent the views of the DOE or the U.S. Government. The U.S. Government retains and the publisher, by accepting the article for publication, acknowledges that the U.S. Government retains a nonexclusive, paid-up, irrevocable, worldwide license to publish or reproduce the published form of this work, or allow others to do so, for U.S. Government purposes.

References

[1] Hammond S W, Sprague M A, Womble D and Barone M 2015 A2e High Fidelity Modeling: Strategic Planning Meetings Tech. Rep. NREL/TP-2C00-64697 National Renewable Energy Laboratory Golden, CO URL https://www.nrel.gov/docs/fy16osti/64697.pdf

[2] Robinson M C and Sprague M A 2019 *Wind Energy Modeling and Simulation Volume 1: Atmosphere and plant* ed Veers P (Institution of Engineering and Technology)

[3] Domino S 2015 Sierra low Mach module: Nalu theory manual 1.0 Tech. Rep. SAND2015-3107W Sandia National Laboratories

[4] Heroux M A, Bartlett R A, Howle V E, Hoekstra R J, Hu J J, Kolda T G, Lehoucq R B, Long K R, Pawlowski R P, Phipps E T, Salinger A G, Thornquist H K, Tuminaro R S, Willenbring J M, Williams A and Stanley K S 2005 *ACM Transactions on Mathematical Software (TOMS) - Special issue on the Advanced CompuTational Software (ACTS) Collection* **31** 397–423

[5] Lin P, Domino S, Fisher T, Hu J, Phipps E, Prokopenko A, Rajamanickam S, Siefert C and Kennon S 2014 *Parallel Processing Letters* **24** 1442005

[6] Edwards H C, Williams A B, Sjaardema G D, Baur D G and Cochran W K 2010 Sierra toolkit computational mesh conceptual model Tech. Rep. SAND2010-1192 Sandia National Laboratories URL https://trilinos.org/oldsite/packages/stk/STK_Mesh_Main.pdf

[7] Domino S P 2018 *Journal of Computational Physics* **359** 331–351

[8] Sørensen N N 1995 *General purpose flow solver applied to flow over hills* Ph.D. thesis URL https://orbit.dtu.dk/files/12280331/Ris_R_827.pdf

[9] Menter F R, Kuntz M and Langtry R 2003 *Turbulence, Heat and Mass Transfer 4* ed Hanjalic K, Nagano Y and Tummers M (Begell House, Inc.) pp 625–632

[10] Travin A, Shur M, Strelets M and Spalart P R 2002 *Advances in LES of Complex Flows* ed Friedrich R and Rodi W (Dordrecht: Springer Netherlands) pp 239–254 ISBN 978-0-306-48383-7

[11] Moeng C H 1984 *Journal of the Atmospheric Sciences* **41** 2052–2062

[12] Thomas S J, Ananthan S, Yellapantula S, Hu J J, Lawson M and Sprague M A 2019 *SIAM Journal of Scientific Computing* In press

[13] Jonkman J M 2013 *Proceedings of the 51st AIAA Aerospace Sciences Meeting including the New Horizons Forum and Aerospace Exposition* (Grapevine, Texas)

[14] Wang Q, Sprague M A, Jonkman J, Johnson N and Jonkman B 2017 *Wind Energy* **20** 1439–1462

[15] Guntur S, Jonkman J, Sievers R, Sprague M A, Schreck S and Wang Q 2017 *Wind Energy Science* **2** 443–468

[16] Gasmi A, Sprague M A, Jonkman J M and Jones W B 2013 *Proceedings of 32nd ASME Wind Energy Symposium, 51st AIAA Aerospace Sciences Meeting including the New Horizons Forum and Aerospace Exposition* (Grapevine, TX)

[17] Sprague M A, Jonkman J M and Jonkman B J 2014 *Proceedings of AIAA SciTech 2014* (National Harbor, MD) URL https://www.nrel.gov/docs/fy14osti/60742.pdf

[18] Sprague M A, Jonkman J M and Jonkman B J 2015 *Proceedings of AIAA SciTech 2015* (Kissimmee, FL) URL https://www.nrel.gov/docs/fy16osti/63203.pdf

[19] Troldborg N, Sørensen J N and Mikkelsen R 2007 *Journal of Physics: Conference Series* **75** 012063

[20] Churchfield M J, Lundquist J K, Quon E, Lee S and Clifton A 2014 Large-eddy simulations of wind turbine wakes subject to different atmospheric stabilities poster presented at the American Geophysical Union Fall Meeting

[21] Lesoinne M and Farhat C 1998 *AIAA Journal* **36** 1754–1757 URL http://dx.doi.org/10.2514/2.7555

[22] Roache P J 2002 *Journal of Fluids Engineering* **124** 4–10

[23] Sprague M, Boldyrev S, Fischer P, Grout R, Jr W G and Moser R 2017 Turbulent flow simulation at the exascale: Opportunities and challenges workshop Tech. rep. U.S. Department of Energy, Office of Science, Advanced Scientific Computing Research Golden, CO published as tech. report NREL/TP-2C00-67648 by the National Renewable Energy Laboratory URL https://www.nrel.gov/docs/fy17osti/67648.pdf

[24] Hand M M, Simms D A, Fingerish L J, Jager D W, Cotrell J R, Schreck S and Larwood S M 2001 Unsteady Aerodynamics Experiment Phase VI: Wind tunnel test configurations and available data campaigns Tech. Rep. NREL/TP-500-29955 National Renewable Energy Laboratory Golden, CO URL https://www.nrel.gov/docs/fy02osti/29955.pdf

[25] Sørensen N N, Michelsen J A and Schreck S 2002 *Wind Energy* **5** 151–169

[26] Duque E P N, Burklund M D and Johnson W 2003 *Journal of Solar Energy Engineering* **125** 457–467

[27] Potsdam M and Mavriplis D 2009 *Aerospace Sciences Meetings*

[28] Jonkman J, Butterfield S, Musial W and Scott G 2009 Definition of a 5-MW reference wind turbine for offshore system development Tech. Rep. NREL/TP-500-38060 National Renewable Energy Laboratory Golden, CO URL https://www.nrel.gov/docs/fy09osti/38060.pdf

[29] Sørensen N N and Johansen J 2008 *2007 European Wind Energy Conference and Exhibition* (Milan, Italy)

[30] Chow R and van Dam C 2012 *Wind Energy* **15** 967–981

[31] Kirby A C, Brazell M, Yang Z, Roy R, Reza Ahrabi B, Mavriplis D, Sitaraman J and Stoellinger M K 2017 *23rd AIAA Computational Fluid Dynamics Conference*