Received: 27 March 2023 | Revised: 27 July 2023 | Accepted: 24 November 2023

DOI: 10.1002/we.2886

RESEARCH ARTICLE

WILEY

ExaWind: Open-source CFD for hybrid-RANS/LES geometry-resolved wind turbine simulations in atmospheric flows

Ashesh Sharma[1] | Michael J. Brazell[1] | Ganesh Vijayakumar[1] | Shreyas Ananthan[2] | Lawrence Cheung[3] | Nathaniel deVelder[3] | Marc T. Henry de Frahan[1] | Neil Matula[3] | Paul Mullowney[4] | Jon Rood[1] | Philip Sakievich[3] | Ann Almgren[5] | Paul S. Crozier[3] | Michael Sprague[1]

[1]National Renewable Energy Laboratory, Golden, Colorado, USA

[2]Ascent OS, Concord, California, USA

[3]Sandia National Laboratories, Albuquerque, New Mexico, USA

[4]Advanced Micro Devices, Inc., Fort Collins, Colorado, USA

[5]Lawrence Berkeley National Laboratory, Berkeley, California, USA

Correspondence
Ashesh Sharma, National Renewable Energy Laboratory, Golden, CO, USA.
Email: ashesh.sharma@nrel.gov; ashesh.sharma@colorado.edu

Funding information
Exascale Computing Project, joint project of the U.S. Department of Energy, Office of Science and the National Nuclear Security Administration, Grant/Award Number: 17-SC-20-SC; U.S. Department of Energy, Office of Energy Efficiency and Renewable Energy, Wind Energy Technologies Office

Abstract

Predictive high-fidelity modeling of wind turbines with computational fluid dynamics, wherein turbine geometry is resolved in an atmospheric boundary layer, is important to understanding complex flow accounting for design strategies and operational phenomena such as blade erosion, pitch-control, stall/vortex-induced vibrations, and aftermarket add-ons. The biggest challenge with high-fidelity modeling is the realization of numerical algorithms that can capture the relevant physics in detail through effective use of high-performance computing. For modern supercomputers, that means relying on GPUs for acceleration. In this paper, we present ExaWind, a GPU-enabled open-source incompressible-flow hybrid-computational fluid dynamics framework, comprising the near-body unstructured grid solver Nalu-Wind, and the off-body block-structured-grid solver AMR-Wind, which are coupled using the Topology Independent Overset Grid Assembler. Turbine simulations employ either a pure Reynolds-averaged Navier–Stokes turbulence model or hybrid turbulence modeling wherein Reynolds-averaged Navier–Stokes is used for near-body flow and large eddy simulation is used for off-body flow. Being two-way coupled through overset grids, the two solvers enable simulation of flows across a huge range of length scales, for example, 10 orders of magnitude going from O(μm) boundary layers along the blades to O(10 km) across a wind farm. In this paper, we describe the numerical algorithms for geometry-resolved turbine simulations in atmospheric boundary layers using ExaWind. We present verification studies using canonical flow problems. Validation studies are presented using megawatt-scale turbines established in literature. Additionally presented are demonstration simulations of a small wind farm under atmospheric inflow with different stability states.

KEYWORDS
atmospheric boundary layer, CFD, geometry-resolved, overset grids, wind turbine

1 | INTRODUCTION

A clean energy future will require advances in wind energy technology for increased deployment. With continued cost reduction in wind energy, wind is anticipated to fuel up to 33% of the planet's future energy demands.[1] Realizing this goal will require continual technology innovation in system design, control, and integration with the electric grid. Recent review articles have outlined wind energy science challenges to facilitate widespread deployment of wind energy.[1,2] The complexity of wind as a resource and its energy capture cannot be overstated. For example, relevant scales in wind farm fluid dynamics include blade-boundary layers that are microns thick and atmosphere structures that are kilometers in length. Key to design innovation and optimization in wind energy is a suite of modeling tools, including predictive high-fidelity models, as well as low-fidelity models that can be quickly run within design optimization loops. The former, while computationally costly, can provide a virtual environment for evaluating potentially disruptive concepts such as vortex generators, de-icing technologies, and novel pitch-control algorithms. High-fidelity models can also serve as benchmarks for next-generation low-fidelity models in their quest to capture the relevant physics as the wind industry diversifies across complex terrains and moves offshore. Moreover, with the advent of machine learning, massive data sets generated by high-fidelity simulations present immense potential for "informing" the machine-learning-based next generation of engineering models.[3]

In the last few decades, state-of-the-art wind farm modeling techniques have largely focused on parameterization of wind turbines using actuator disks[4–6] and actuator lines[7–9] within a computational fluid dynamics (CFD) solver. In an actuator disk model, the turbine rotor is represented as a body force distributed over a disk, whereas in an actuator line model, the turbine blades are represented as body forces distributed along blade lines. Actuator parameterizations rely on the blade-element theory wherein forces are determined from two dimensional (2D) aerodynamic look-up tables. Although corrections exist to compensate for the various approximations in these models, the full three dimensional (3D) turbulent flow field presents many complications that airfoil look-up tables cannot handle satisfactorily, as recently demonstrated for inflow conditions with shear and yaw.[10] A recent study on direct comparisons between actuator line and geometry-resolved simulations for MW-scale turbines showed underprediction of thrust by up to 8% under complex inflow conditions, with the discrepancies attributed to wake flow, induction, the reliance of the actuator line on airfoil look-up data, and the force application into the computational domain.[11] The topic of 3D corrections to 2D airfoil data is a still a matter of considerable debate and no single model works consistently across different rotors.[12] Given the balance of accuracy and computational cost accorded by actuator methods, these techniques continue to be valid for many design load cases, and their place in the wind community is unquestionable. However, the next-generation of actuator methods is expected to be informed by higher fidelity geometry-resolved CFD simulations.[10,13]

Key to overcoming some of the aforementioned challenges and building the next-generation of engineering tools is developing modeling tools capable of capturing otherwise underresolved phenomena across a wide range of temporal and spatial scales such as blade-boundary layers, especially for nonstandard blade geometries and off-design blade conditions, for example, in the presence of blade erosion, icing, after market add-ons such as vortex generators; blade–wake and blade–atmosphere interactions in stall- and deep-stall-inducing flows such as but not limited to extreme-weather conditions; interactional aerodynamics between blades, nacelle, tower, and hub; and complex terrain impacts.[14] With the advances in, and increased accessibility to high-performance computing (HPC), including the establishment of exascale-class computing,[15] geometry-resolved high-fidelity modeling presents an attractive avenue to overcoming the aforementioned modeling barriers in wind farm physics.

Geometry-resolved CFD simulations, wherein the turbine geometry is represented by a body-fitted grid, are rare primarily due to the vast temporal and spatial length scales associated with the problem. Geometry-resolved computational domains can require grid resolutions of $O(10^{-6}$ m) near blade surfaces to model the boundary layers, and time-step sizes of $O(10^{-3}$ s) to resolve the turbulent scales of interest.[16] Discretizing the wind farm spatial and temporal scales ranging from the blade-boundary layer to the atmospheric boundary layer (ABL) could require a computational grid with $O(10^{10})$ grid points to be run for $O(10^{5})$ time steps. From an engineering perspective, the primary challenge is to strike a balance between manageable computational cost and the accuracy requirements. The primary approaches to achieving geometry-resolved wind farm simulations use either sliding grids[17–19] or overset grids[20–22] to resolve flow past the turbine geometry. During operation, turbine controllers can often cause the blade to rotate about its pitch axis or about the tower axis (yaw) resulting in rotor movement outside the turbine plane, rendering sliding-grid frameworks impractical. Overset grids, on the other hand, present immense flexibility to allow for relative motion between the overlapping grids while retaining control of local grid characteristics during domain traversal.[23–25] Consequently, CFD solvers employing overset grids represent the state of the art in geometry-resolved wind physics modeling, and it is also the strategy we adopt in this paper.

One of the first applications of overset grids to wind energy was following the NREL Phase VI rotor experiments,[26] when researchers recognized the need for higher fidelity wind turbine simulations to predict potentially separating flow around a rotating, twisted, and tapered blade under uniform inflow for comparison against wind tunnel experiments.[27–31] Most of these seminal simulations were performed over a decade ago, and to date, the researchers who performed the aforementioned work are also the only developers of a handful of overset-based simulation tools capable of geometry-resolved wind farm simulations, largely due to the complexity associated with building such a framework. Among commercial CFD software, Simcenter STAR-CCM+ is the only tool known to possess similar modeling capabilities.[32]

The first CFD software to pioneer use of overset grids for wind turbine simulations was EllipSys3D,[30] developed in cooperation between the Department of Mechanical Engineering at Technical University of Denmark and the Department of Wind Energy at the Risø National Laboratory,

Denmark. EllipSys3D uses the finite volume method with collocated-grid arrangement to solve the incompressible-flow Navier–Stokes equations in a curvilinear coordinate system. It is also equipped with turbulence models necessary to simulate geometry-resolved flow around turbines in ABLs. Both the near-body and off-body flows are restricted to structured grids modeled using the same CFD solver. The source code is MPI-enabled but lacks support for hybrid CPU/GPU architectures. Over the years, EllipSys3D has been used to perform many studies ranging from validating geometry-resolved CFD turbine simulations in uniform flow[20,33] to enabling fluid–structure interaction for blade-resolved CFD simulations.[34]

Another research group that actively innovates in overset technology for wind energy applications are the developers of W^2A^2KE3D at the University of Wyoming. This tool has been used to demonstrate flow over airfoils[35,36] as well as flow over turbines and wind farms in an ABL.[37] W^2A^2KE3D consists of the near-body flow solver NSU3D,[38] a well-known unsteady RANS solver for unstructured grids based on a node-centered finite volume method. The off-body solver is a discontinuous Galerkin method implemented using the dynamic adaptive grid refinement framework *p4est*.[39] Like EllipSys3D, W^2A^2KE3D too is designed to run on massive number of MPI ranks but lacks support for GPUs. Related work at the University of Wyoming prior to W^2A^2KE3D has used the HELIOS framework to study coupled mesoscale–microscale effects on wind farm aerodynamics.[40] HELIOS is primarily developed for prediction of vortex-dominated flows in rotary-wing systems and uses a multiple grid multiple solver paradigm coupled using overset. Funded by the U.S. Department of Defense, the details of the software are not published, and access is highly restricted.

The only other tool that has demonstrated overset-based geometry-resolved wind turbine simulations in the past is CFDShip-Iowa.[31] CFDShip-Iowa, although a general-purpose finite-difference-based CFD software, is primarily geared at research on ship hydrodynamics. Similar to EllipSys3D, CFDShip-Iowa solves the incompressible flow in the near-body and off-body domains using structured grids simulated using the same CFD solver. Turbulence models available include unsteady-RANS and detached eddy simulation. Previous work has also demonstrated fluid–structure interaction to analyze the NREL 5-MW turbine's aeroelastic[41] and aero-servo-elastic[42] response under atmospheric turbulence modeled using the Mann model. CFDShip-Iowa is closed-source and funded by the Office of Naval Research with use restricted to projects funded by U.S. government agencies.

Motivated by creating a framework capable of predicting the highest fidelity of wind farm physics possible, designed to accommodate future exascale systems and in the spirit of creating a community-driven platform, the US Department of Energy has invested in the development of open-source high-fidelity modeling tools for wind farm physics for the better part of the last decade. This framework, referred to as ExaWind[*], is an open-source hybrid-solver platform designed to perform large-scale wind farm simulations in atmospheric flows. ExaWind was developed with the physics goals of solving acoustically incompressible fluid dynamics involving geometry-resolved two-way coupled fluid–structure interaction using nonlinear structural dynamics models, hybrid RANS/LES turbulence models, and one-way coupling to weather-scale forcing such as numerical weather prediction. ExaWind utilizes a multisolver strategy of Nalu-Wind,[†] a fully implicit node-centered finite volume unstructured-grid near-body solver, and AMR-Wind,[‡] a semi-explicit semistaggered finite volume/difference block-structured grid off-body solver built using the AMReX framework.[43] Both CFD solvers model acoustically incompressible flow. Nalu-Wind is used to resolve the flow around blades, including the thin boundary layers. The background flow is simulated with AMR-Wind, and the two CFD solvers are coupled through overset grids processed by the Topology Independent Overset Grid Assembler (TIOGA)[§] library which has been modified to enable handling the mixed arrangement of flow variables presented by the two CFD solvers. This multisolver strategy allows each CFD solver to use tailored linear-system solvers which operate in parallel to solve the equations on the near- and off-body grids. Both Nalu-Wind and AMR-Wind have been designed to be GPU-enabled on modern computing architectures. We have achieved this by constructing the entire framework using programming models that support performance portability, that is, the same source code runs on varying GPU and CPU architectures. In addition to the CFD solvers, the ExaWind framework also consists of the widely used whole-turbine-simulation tool, OpenFAST,[44] which includes a turbine control system, a linear model for tower bending deformation, and a high-order nonlinear finite element blade dynamic model, BeamDyn,[45] designed to capture the dynamics of modern wind turbine blades. Under development is an interface between OpenFAST and Nalu-Wind to enable fluid–structure interaction coupling between the geometry-resolved turbine and the atmospheric flow.

In a previously published study about ExaWind,[46] the authors presented an overview of some preliminary results where flow past a turbine under uniform inflow was demonstrated using Nalu-Wind-only simulations coupled with OpenFAST to achieve fluid–structure interaction. Single turbine simulations may achieve petascale computational levels, but simulating problems for whole wind farms with multiple turbines can be categorized as exascale-class,[47] requiring algorithms capable of harnessing the potential of exascale hardware. Consequently, in this paper, we present for the first time our hybrid-solver strategy described above. We present the discretization details surrounding the two CFD solvers alongside the appropriate turbulence models implemented in the solvers. We further present our overset strategy for interfacing the two CFD solvers and accommodating the different spatial and temporal discretization schemes therein. The linear solver strategy tailored to the near- and off-body grids is discussed for effective convergence of the highly nonlinear flow field for large-scale problems. For verifying and validating the ExaWind

[*] https://github.com/ExaWind.
[†] https://github.com/ExaWind/nalu-wind.
[‡] https://github.com/ExaWind/amr-wind.
[§] https://github.com/ExaWind/tioga.

solver, we present a formal order-of-accuracy study, a first-ever overset-based flow past a sphere analysis for a varying range of Reynolds number (*Re*) for examining the accuracy of the ExaWind solver for flow past bluff bodies, demonstration of the ExaWind solver for turbulent flow past different types of megawatt-scale turbines and subsequent validation against published CFD results and experimental data, and demonstration of the ExaWind solver for flow past multiple turbines in atmospheric inflow for different stability states. Note, as mentioned above, that the fluid–structure interaction capabilities between OpenFAST and Nalu-Wind are underway, rendering fluid–structure interaction outside the scope of this paper. Consequently, for the purposes of the current study all near-body flow assume rigid body behavior.

The remainder of the manuscript is organized as follows: Section 2 presents the governing equations of interest for both the near- and off-body CFD solvers including the relevant turbulence modeling strategy for geometry-resolved simulation under atmospheric inflow, Section 3 presents the overset approach for coupling the near-body and off-body solvers alongside an overview of the solution-exchange algorithm employed in this work, Section 4 presents an overview of the linear solvers used in ExaWind, Section 5 presents the numerical examples to analyze and validate the ExaWind solver, and Section 6 presents the summary and concluding remarks.

Note that most of the code documentation is provided in the online GitHub repositories, and the present manuscript aims to present only the primary algorithms of the ExaWind software stack, and their validation. Throughout the manuscript, where deemed appropriate, footnotes have been added to guide the reader to the more detailed online documentation.

2 | CFD SOLVERS

This section details the discretization adopted by the CFD solvers that make up the ExaWind hybrid-solver framework: Nalu-Wind and AMR-Wind. We also describe the turbulence modeling approach adopted for a hybrid RANS/LES scheme. The ExaWind solvers solve the acoustically incompressible Navier–Stokes equations, which can be summarized as

$$\text{Momentum}: \rho\frac{\partial \boldsymbol{u}}{\partial t}+\rho(\boldsymbol{u}\cdot\nabla\boldsymbol{u})+\nabla p-\nabla\cdot\boldsymbol{\sigma}(\boldsymbol{u})=\boldsymbol{F}, \tag{1}$$

$$\text{Continuity}: \nabla\cdot\boldsymbol{u}=0, \tag{2}$$

$$\text{Potential temperature}: \rho\frac{\partial \Theta}{\partial t}+\rho(\boldsymbol{u}\cdot\nabla\Theta)-\nabla\cdot\boldsymbol{\sigma}(\Theta)=0, \tag{3}$$

where the velocity vector, pressure, and potential temperature solution degree of freedom (DOF) are denoted by $\boldsymbol{u}, p$, and Θ, respectively. The variables $\boldsymbol{\sigma}$ and $\boldsymbol{F}$ denote the viscous and Reynolds/subfilter-scale (SFS) stress tensor, and the forcing (Coriolis, buoyancy, body force, etc.) vector, respectively. For brevity, only the discretization of the velocity and pressure equations is described. Other scalar equations (potential temperature, turbulent kinetic energy, specific dissipation rate, etc.) are discretized in a similar manner.

2.1 | Nalu-Wind

Nalu-Wind* is an open-source CFD solver written in C++. It is a fork of Nalu[48] which is a variable density turbulent flow research tool developed at Sandia National Laboratories that supports a wide range of energy applications such as combustion and participating media radiation. Nalu-Wind has removed physics capabilities not applicable to wind energy and tailored the algorithms for wind energy applications.

To be able to resolve complex geometries encountered in wind farms, Nalu-Wind uses unstructured-grid CFD to solve the acoustically incompressible Navier–Stokes equations, maintaining mass continuity through approximate pressure projection. Governing equations in Nalu-Wind are discretized in space using a second-order node-centered finite volume scheme (see, e.g., Diskin et al[49] for discretization details). An implicit second-order backward difference formula is used for time integration. The time-stepping algorithm in Nalu-Wind is detailed below.

An incremental approximate pressure-projection algorithm[50,51] calculates momentum and pressure solutions at each time step. We solve a set number of Picard iterations, K, with the solution initialized to $\boldsymbol{u}_0^{n+1}=\boldsymbol{u}_K^n$ and $p_0^{n+1}=p_K^n$ at each time step. The final equations solving for $\Delta\hat{\boldsymbol{u}}_{k+1}=\hat{\boldsymbol{u}}_{k+1}-\boldsymbol{u}_k$ and $\Delta p_{k+1}=p_{k+1}-p_k$ at each Picard iteration, k, are presented below without the time-step superscripts.

$$\text{Momentum}: A_d\Delta\hat{\boldsymbol{u}}_{d,k+1}+A_{nb}\Delta\hat{\boldsymbol{u}}_{nb,k+1}=\boldsymbol{F}-A\boldsymbol{u}_k-\nabla p_k, \tag{4}$$

$$\text{Pressure Poisson}: -\tau\nabla^2(\Delta p_{k+1})=-\nabla\cdot(\rho\hat{\boldsymbol{u}}_{k+1}+\tau\nabla p_k)+\tau\nabla^2 p_k, \tag{5}$$

*https://nalu-wind.readthedocs.io/en/latest/source/theory/index.html.

$$\text{Projection}: \boldsymbol{u}_{k+1} = \hat{\boldsymbol{u}}_{k+1} - \left(\frac{\tau}{\rho}\right)\nabla(\Delta p_{k+1}), \tag{6}$$

where

$$A_d = -\sum A_{nb} + \frac{\gamma}{\Delta t}\rho V_d;\ \gamma = \frac{(1+2\frac{\Delta t^n}{\Delta t^{n-1}})}{1+\frac{\Delta t^n}{\Delta t^{n-1}}}. \tag{7}$$

Here, A_d and A_{nb} represent the diagonal and the off-diagonal contributions, respectively, to the momentum equation from the time derivative, diffusion, and linearized advection terms,[52] and V_d denotes the control volume of the cell. Note for an approximate pressure-projection scheme, $\nabla \cdot \tau\nabla \neq \tau\nabla^2$.[50] For further details, the reader is referred to the derivations in literature[50,53]. In summary, one Picard iteration (4)-(6) predicts a nondivergence-free velocity field and then projects it onto a divergence-free subspace. The above formulation is well suited for unsteady LES flows and has several advantages. For stationary-grid problems, the left-hand side of (5) is purely geometric, allowing the preconditioner to be computed only once for the entire simulation. However, one needs to be careful with their choice of the stabilization factor τ which if chosen to be too high can result in error due to stabilization dominating. Consequently, τ is chosen to be

$$\tau = A_d^{-1}. \tag{8}$$

In RANS simulations at large Courant numbers, some underrelaxation might be necessary, as the viscous terms dominate near the blade-boundary layer, to restore the diagonal dominance of the transport equations using a relaxation factor η such that

$$\phi^{k+1} = \eta\phi^{k+1} + (1-\eta)\phi^k;\ A_d = -\frac{\sum A_{nb}}{\eta} + \frac{\gamma}{\Delta t}\rho V_d, \tag{9}$$

where ϕ is any solution field. When underrelaxation is applied, only the advection and diffusion contributions to the diagonal term are modified by dividing these terms by the underrelaxation factor. Also, while the pressure solution may be underrelaxed at each Picard iteration, the full pressure update from (5) is used for the velocity and mass-flux updates. Unless mentioned otherwise, for all turbulent simulations in this study, the underrelaxation parameter η is set to 0.7 for momentum and scalar transport equations and 0.3 for the pressure update. For RANS simulations, upwinding of the advection terms is necessary. Hence, we use linear upwinding, the details of which are available in the Nalu-Wind user manual.*

It must be emphasized that the robustness of the above time-stepping algorithm at large Courant numbers comes at a price. The introduction of underrelaxation in the linear system means that the temporal order of accuracy is only satisfied upon the convergence of nonlinear residuals within each time step to machine precision. However, convergence of the nonlinear residuals to machine precision using Picard iterations can be quite slow and would require several iterations. The common practice is to perform a fixed number of Picard iterations and move to the next time step. In general, the authors recommend using enough Picard iterations to drop the nonlinear residual by about 2 to 3 orders of magnitude within a time step. In the authors' experience, for turbine simulations, four Picard iterations are sufficient to drop the nonlinear residual as recommended. For more benign laminar flows, fewer number of Picard iterations might be sufficient. Consequently, unless specified otherwise, numerical simulations in the current work use four Picard iterations for robust convergence of the linear systems. Accelerating the nonlinear convergence within each time step and exploring techniques to eliminate the need for underrelaxation of the linear systems in Nalu-Wind are ongoing topics of research.

Nalu-Wind has the infrastructure for discretizing underlying models, relying heavily on the Trilinos[54] Sierra Toolkit (STK) which offers an in-memory parallel-distributed database for unstructured grids. Nalu-Wind employs modern software engineering practices and utilizes reliable, well-supported open-source libraries (*hypre*,[55] Trilinos, etc.) that scale well on large HPC systems. In addition, Nalu-Wind is designed using the Kokkos[56] programming model for a GPU-enabled performance-portable source code capable of running on modern HPC platforms.[57]

2.2 | AMR-Wind

AMR-Wind† is an open-source block-structured adaptive grid incompressible-flow solver specialized for efficiency and scalability. The solver uses the AMReX library,[43] which offers grid data structures, grid adaptivity, linear solvers, and corresponding performance-portable parallel algorithms

*https://nalu-wind.readthedocs.io/en/latest/source/theory/advectionStabilization.html.
†https://exawind.github.io/amr-wind/theory/theory.html.

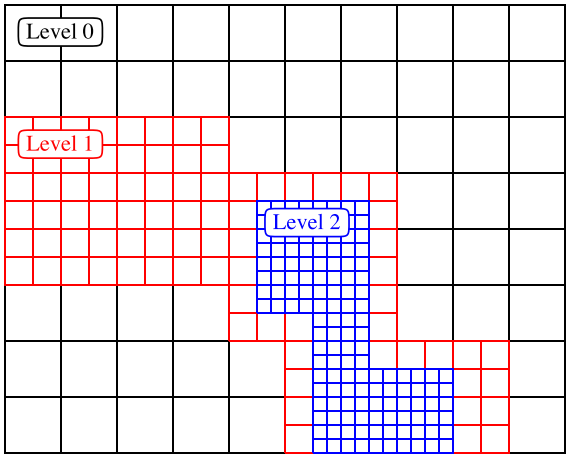

FIGURE 1 Example AMR-Wind grid with two levels of patch-based refinement.

for various GPU architectures. AMR-Wind is designed to perform LES of ABL flows which are coupled to Nalu-Wind using an overset methodology (Section 3) for geometry-resolved simulations of multiple wind turbines in a wind farm. Figure 1 depicts an example of an AMR-Wind grid as created using AMReX.

The grid is a collection of patches where each patch is defined by nine integers and six reals. The integers define the lower and upper bounds of the patch relative to the global structured grid in addition to the number of cells in each direction. The six reals represent the location of the patch and the spacing in each Cartesian direction. Such a compact grid data structure has several advantages for the overset algorithm as discussed in Section 3. Typically, a patch size of 16 to 32 cells per dimension on a CPU and 64 to 128 cells per dimension on a GPU are chosen to maximize efficiency of computing resources. Details on AMReX are presented in the paper by Weiqun et al,[58] as well as in the source code's online documentation.[‡]

The spatial discretization combines the finite volume method and the finite-difference method. Important aspects of the discretization are discussed below, but for more detail, the reader is referred to the seminal work by Almgren et al.[59,60] Velocity, scalars, and pressure gradients are located at cell centers, whereas pressure is located at nodes. Using partial staggering and an approximate projection method,[60] AMR-Wind constructs linear systems that are well-researched, have manageable stencils, and can be effectively solved using standard techniques like multilevel multigrid (MLMG). These discretization choices provide a good blend of efficiency and accuracy. In addition to the spatial staggering, there is also staggering in time similar to a Crank–Nicholson scheme such that

$$\frac{\boldsymbol{u}^* - \boldsymbol{u}^n}{\Delta t} + \left(\boldsymbol{u}^{MAC} \cdot \nabla \boldsymbol{u}\right)^{n+1/2} = \frac{1}{\rho^{n+1/2}} \left(-\nabla p^{n-1/2} + \frac{1}{2}\left(\nabla \cdot \boldsymbol{\sigma}^n + \nabla \cdot \boldsymbol{\sigma}^{n+1}\right)\right) + \boldsymbol{F}^{n+1/2}, \tag{10}$$

where density at $n+1/2$ is obtained using a simple average:

$$\rho^{n+1/2} = \frac{1}{2}\left(\rho^n + \rho^{n+1}\right). \tag{11}$$

The advection term is constructed by extrapolating in time using a Godunov method.[60] To do this, the velocity is first extrapolated in space to the faces and in time to $n+1/2$ in a predictor step. This predicted face velocity is not guaranteed to be divergence-free, so a Marker-And-Cell (MAC) projection is used to correct it, which takes the form

$$P^{MAC}\left(\boldsymbol{u}^f\right) = \boldsymbol{u}^f - \frac{\Delta t}{2\rho^n} \nabla \psi, \tag{12}$$

where $\boldsymbol{u}^f$ represents the face velocity and ψ is a Lagrange multiplier located on the cells. Taking the divergence and setting it equal to zero forms a Poisson equation:

[‡]https://amrex-codes.github.io/amrex/docs_html.

$$\nabla \cdot \left(\frac{\Delta t}{2\rho^n} \nabla \psi \right) = \nabla \cdot \boldsymbol{u}^f. \tag{13}$$

The Poisson equation is discretized using a cell-centered seven-point stencil in 3D, which is efficiently solved using MLMG as a linear solver.[43] There are several available options for spatially extrapolating the velocities from the cell centers to the faces. These include Piecewise Linear Method (PLM),[61] Piecewise Parabolic Method (PPM),[62] Weighted Essentially Nonoscillatory (WENO)-Z,[63] and WENO-JS.[64] PPM is the default advection scheme used in AMR-Wind, although in the authors' experience, for ABL and hybrid-CFD turbine simulations, the WENO-Z scheme has demonstrated a good balance between low numerical dissipation and high numerical stability, and that method was used for those cases. PLM, PPM, and WENO variants offer different numerical dissipation properties, and the interaction of the numerical dissipation with the LES models is currently under investigation.

With the advection term computed, the intermediate velocity $\boldsymbol{u}^*$ in (10) is advanced by solving the Helmholtz problem discretized using a cell-centered finite-difference method forming a seven-point stencil in 3D.[60] Note, for the Helmholtz solve in (10), the already computed advection term is treated as a constant. The intermediate velocity vector $\boldsymbol{u}^*$ is not guaranteed to be divergence-free. Consequently, an approximate projection method is used to solve for the velocity at $n+1$ such that $\boldsymbol{u}^{n+1} = P(\boldsymbol{u}^*)$ where the nodal projection P is defined to be:

$$P(\boldsymbol{u}^*) = \boldsymbol{u}^* + \frac{\Delta t}{\rho^{n+1/2}} \left(\nabla p^{n-1/2} - \nabla \chi \right). \tag{14}$$

This approximate projection is different from the original work[60] and more similar to the projection introduced later by Almgren et al.[65] Taking the divergence of (14) and setting it equal to zero yields the following:

$$\nabla \cdot \left(\frac{\Delta t}{\rho^{n+1/2}} \nabla \chi \right) = \nabla \cdot \left(\boldsymbol{u}^* + \frac{\Delta t}{\rho^{n+1/2}} \nabla p^{n-1/2} \right), \tag{15}$$

where χ is a Lagrange multiplier related to the pressure field and solved on the nodes. To solve this nodal projection, a variational form is used giving a 27-point stencil in 3D.[60] Once the solution χ is obtained, the velocity is updated using (14) and the pressure and its gradient are updated such that

$$\nabla \chi \rightarrow \nabla p^{n+1/2}; \; \chi \rightarrow p^{n+1/2}. \tag{16}$$

The gradient $\nabla \chi$ is approximated by taking finite-differences of χ along edges and averaging each edge to the cell center. For each coordinate direction, there are four edges that contribute to the value at the cell center.

The time stepping scheme in (10)–(16) can be summarized as a predictor–corrector approach wherein a lagged pressure in (10) is used to predict the velocity u^* which is not guaranteed to be divergence free, followed by a projection to enforce the divergence-free constraint on the velocity field and update the pressure (and pressure gradient). The alternative to this predictor–corrector approach would be an implicit method in which we solve for the velocity and pressure in the same step. We have chosen the predictor–corrector approach for a variety of reasons explained in previous papers.[59,60] Additionally, the pressure gradient is defined at a "half-time" level; this follows from the fact that in the Navier–Stokes equations (1), the pressure gradient should live at the same time as the time derivative of velocity rather than velocity itself.

Note that given the staggered arrangement of pressure and velocity in time, there is a mismatch across the two CFD solvers in regard to solution update. While a Nalu-Wind time step advances both pressure and velocity from $n \rightarrow n+1$, an AMR-Wind time step advances the velocity from $n \rightarrow n+1$ and the pressure from $n-1/2 \rightarrow n+1/2$. Considering that pressure in incompressible flow exists solely to satisfy a divergence-free velocity, this inconsistency is not considered problematic for the physics of interest to ExaWind.

2.3 | Turbulence modeling

Turbulence models in ExaWind's CFD solvers include a combination of RANS, hybrid RANS/LES, and LES models employed depending on the problems of interest. Geometry-resolved simulations of turbines in Nalu-Wind are typically handled using RANS, while the background flow in AMR-Wind is handled using RANS or LES models for uniform and atmospheric inflow problems, respectively. All turbulence models use the Boussinesq approximation to model the deviatoric part of the stress tensor $\boldsymbol{\sigma}$, that is,

$$\boldsymbol{\sigma} - \boldsymbol{I}\frac{tr(\boldsymbol{\sigma})}{3} = 2\mu_t \boldsymbol{S}, \tag{17}$$

where μ_t is the computed eddy viscosity and $\boldsymbol{S}$ is the strain-rate tensor. The average normal stress $tr(\boldsymbol{\sigma})/3$ is added to the pressure and modeled as a modified pressure as is typical in such simulation tools. Depending on the choice of the turbulence model employed, the stress tensor $\boldsymbol{\sigma}$ is either the SFS stress in LES or Reynolds stress in RANS simulations. For both LES and RANS, we choose models that are based on prognostic equations for the SFS/total turbulent kinetic energy so that the two models can be blended together.

The one-equation SFS model for the LES of ABL from Moeng[66] solves for the SFS kinetic energy k^{SFS} as follows:

$$\begin{aligned}
\rho\frac{\partial k^{SFS}}{\partial t} + \rho\boldsymbol{u}\cdot\nabla k^{SFS} &= -\left(\boldsymbol{\sigma}^{SFS}:\boldsymbol{S}\right) + \nabla\cdot 2\mu_t\nabla k^{SFS} + \frac{g}{\Theta_0}\sigma_\Theta - \rho C_\epsilon\frac{{k^{SFS}}^{3/2}}{l},\\
\sigma_\Theta &= -\frac{\mu_t}{Pr_t}\frac{\partial\Theta}{\partial n},\\
\mu_t &= \rho C_k l\sqrt{k^{SFS}},\\
l &= \begin{cases} \Delta x, & \text{Negative stratification } \frac{g}{\Theta_0}\frac{\partial\Theta}{\partial n} \le 10^{-10}\\ \min\left(\Delta x, 0.76\sqrt{k^{SFS}}\left(\frac{g}{\Theta_0}\frac{\partial\Theta}{\partial n}\right)^{-1/2}\right) & \text{Positive stratification } \frac{g}{\Theta_0}\frac{\partial\Theta}{\partial n} > 10^{-10}, \end{cases}\\
Pr_t &= 1/\left(1+\frac{2l}{\Delta x}\right).
\end{aligned} \tag{18}$$

where Θ_0 is a a constant reference potential temperature, Pr_t is the turbulent Prandtl number, n denotes the direction along which stratification occurs, and Δx is the characteristic length of the cell. The state of stratification required to determine the length scale l is determined by the local gradient of the filtered potential temperature. The default choices for the constants are $C_\epsilon = 0.93$ and $C_k = 0.1$.

Assuming z to be the wall normal direction, the SFS stresses at the terrain boundary, following Moeng,[66] are written as

$$\sigma^{SFS}_{xz_{wall}} = \overline{\sigma^{SFS}_{xz_{wall}}}\frac{||u||_1\overline{u_{x1}} + \overline{||u||_1}(u_{x1} - \overline{u_{x1}})}{\overline{||u||_1}\sqrt{\overline{u_{x1}}^2 + \overline{u_{y1}}^2}}, \tag{19}$$

$$\sigma^{SFS}_{yz_{wall}} = \overline{\sigma^{SFS}_{yz_{wall}}}\frac{||u||_1\overline{u_{y1}} + \overline{||u||_1}(u_{y1} - \overline{u_{y1}})}{\overline{||u||_1}\sqrt{\overline{u_{x1}}^2 + \overline{u_{y1}}^2}}, \tag{20}$$

$$\sigma^{SFS}_{\Theta z_{wall}} = \overline{\sigma^{SFS}_{\Theta z_{wall}}}\frac{||u||_1(\overline{\Theta_1} - \Theta_0) + \overline{||u||_1}(\Theta_1 - \overline{\Theta_1})}{\overline{||u||_1}(\overline{\Theta_1} - \Theta_0)}, \tag{21}$$

where subscript 1 denotes the value at the first cell in the wall normal direction and $\overline{\bullet}$ represents the horizontal plane average. The Monin–Obukhov similarity law is used to compute mean SFS stresses.[67] Note that the formulation for the potential temperature flux is singular when $\overline{\Theta_1} = \Theta_\circ$. Therefore, the model for the SFS potential temperature flux is reformulated using the Monin–Obukhov similarity law as

$$\sigma^{SFS}_{\Theta z_{wall}} = \left(\frac{u_*\kappa}{\Theta_h}\right)\frac{||u||_1(\overline{\Theta_1} - \Theta_0) + \overline{||u||_1}(\Theta_1 - \overline{\Theta_1})}{\overline{||u||_1}}, \tag{22}$$

where u_* is the friction velocity, κ is the von Kármán constant set to 0.41 by default, and Θ_h is the nondimensional potential temperature gradient from the Monin–Obukhov similarity law.[67] Note that the use of horizontally averaged velocity for the calculation of the Monin–Obukhov law is valid only for horizontally homogeneous flows such as an ABL simulation in the absence of turbines. For nonhorizontally homogeneous conditions, such as ABL simulations in the presence of turbines, the Monin–Obukhov law is applied to the locally sampled velocity at each grid point without averaging.

In the near-body solver, Nalu-Wind, we use the k-ω SST RANS model.[68] The model blends $k-\omega$ and $k-\epsilon$ RANS models, leveraging the advantages of the ω treatment near the wall and the ϵ treatment in the free stream. It solves two prognostic equations, one for the total turbulent kinetic energy k and one for the specific dissipation rate ω such that

$$\rho\frac{\partial k}{\partial t} + \rho\boldsymbol{u}\cdot\nabla k = \nabla\cdot\left((\mu + \varsigma_k\mu_t)\nabla k\right) + \min(2\mu_t(\boldsymbol{S}:\boldsymbol{S}), 10\rho\beta^* k\omega) - \rho\beta^* k\omega, \tag{23}$$

$$\begin{aligned}\rho\frac{\partial\omega}{\partial t}+\rho\boldsymbol{u}\cdot\nabla\omega &= \nabla\cdot((\mu+\varsigma_\omega\mu_t)\nabla\omega)\\ &\quad+\frac{\alpha}{\mu_t}\min(2\mu_t(\boldsymbol{S}:\boldsymbol{S}),10\rho\beta^*k\omega)-\rho\beta\omega^2+2\rho\varsigma_\omega\frac{1-F_1}{\omega}\nabla k\cdot\nabla\omega,\\ \mu_t &= \frac{\rho a_1 k}{\max(a_1\omega,F_2\sqrt{2(\boldsymbol{S}:\boldsymbol{S})})}\end{aligned}\tag{24}$$

where μ is the dynamic fluid viscosity; F_1, F_2 are blending functions; and $\varsigma_k, \varsigma_\omega, \alpha, a_1, \beta^*, \beta$ are constants[68]. Implementation of RANS models in AMR-Wind is an evolving topic. For the purposes of the current study, the k-ω SST model in (23) and (24) was implemented in AMR-Wind without limiting the production term for the ω equation.

To simulate the operation of turbines in the ABL, AMR-Wind only solves the one-equation model (18), while Nalu-wind solves (23) and (24) in the k-ω SST model. Since AMR-Wind does not use a prognostic equation for ω, it supplies the boundary condition for ω at the overset interface as $\omega = \sqrt{k}/(C_k\Delta x)$. Studies using validation data from field measurements are underway to incorporate a hybrid RANS/LES model in Nalu-Wind[69] and AMR-Wind, based on the improved delayed detached eddy simulation (IDDES) model by Gritskevich et al[68] wherein RANS near the turbine is blended with the LES in the ABL.

3 | OVERSET-GRID COUPLING

An overset-grid methodology is used to connect the near-body solver, Nalu-Wind, with the off-body solver, AMR-Wind. Figure 2 presents an example of an overset grid used for flow past a turbine simulation. The near-body grid is unstructured, whereas the background grid is

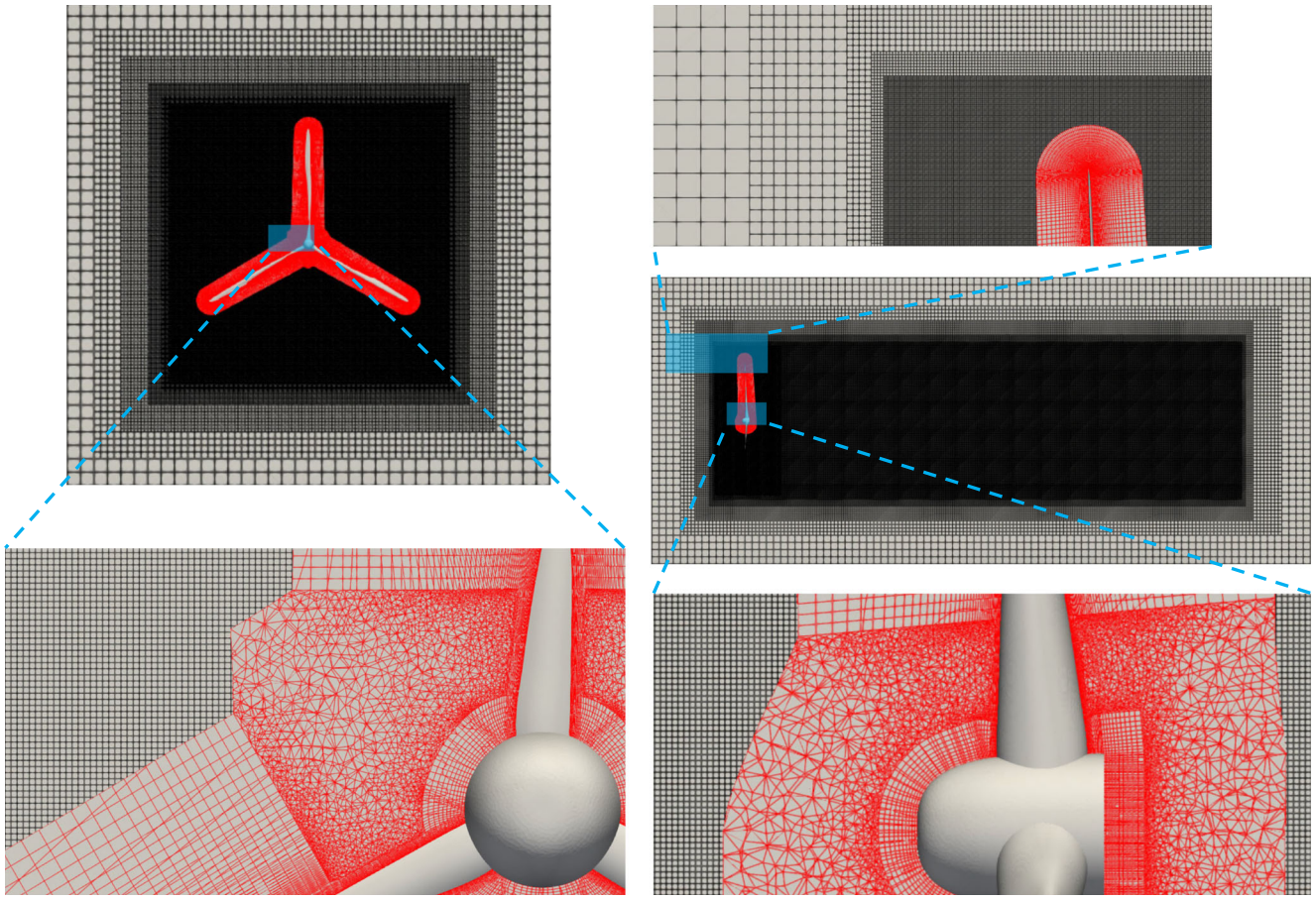

FIGURE 2 Nalu-Wind (red) and AMR-Wind (black) grids, coupled via overset, for the NREL 5-MW rotor. Unstructured grids resolve complex geometric features and boundary layers. Block-structured Cartesian grids are used to capture far-field flow features including wakes and the atmospheric boundary layer (ABL)

block-structured and consists of multiple levels of refinements. Such a setup is representative of overlapping grids utilized by the ExaWind solver. The overlapping grids in ExaWind are connected using TIOGA which is an open-source automated overset-grid-assembly library[35,70].

3.1 | Hole cutting

TIOGA is an implicit hole-cutting tool that automates the overset assembly process; that is, it classifies each node or cell (depending on the DOF location) in every grid as one of the following to determine the regions of information exchange:

- *Field*: Nodes/cells where the governing equations are solved on the grid.
- *Fringe*: Nodes/cells where information is exchanged between overlapping grids. The points used to interpolate solution from the *donor* grid to a recipient grid are referred to as the *donor* points. In the presence of multiple overlapping near-body grids, TIOGA determines the *donor* points to be those belonging to cells with the finest resolution (smallest volume) among all overlapping near-body grid cells. This is because flow field features can be captured more precisely by a finer resolution grid, as compared with grids with coarser resolution. Additional details surrounding *donor* classification can be found in the papers describing the hole-cut algorithms employed in TIOGA.[23,70]
- *Hole*: Nodes/cells where solution does not exist.

Figure 3 visually highlights the outcome of the overset assembly process as it pertains to the overlapping grids depicted in Figure 2.

The red regions depict the *field* points, and the blue regions depict the *fringe* points in both the near-body and off-body grids. The white regions depict the *hole* points in the off-body grid corresponding to the wall in the near-body grid.

The overset assembly process is split into determining the hole cut for near-body-near-body overlaps and near-body-off-body overlaps. For near-body-near-body overlaps, the hole-cutting process is described in detail in the seminal work introducing TIOGA.[70] For the near-body-off-body overlaps, modifications were made to TIOGA to develop hole-cutting similar to that implemented in the overset connectivity tool PUNDIT[23] while also accommodating staggered arrangement of flow field variables as defined on node and cell patches of the block-structured grid in AMR-Wind. Following the hole cut, solution is exchanged/interpolated using trilinear interpolation functions from red regions in a grid to blue regions in the corresponding overlapping grid. For effective interpolation across the overlapping grids, it is advisable that the cell resolutions along the near-body grid boundary be similar to the resolutions of the overlapping cells in the off-body grid.

The computational costs for hole-cutting largely depend on the size of the near-body grid. This is because the hole-cutting search algorithms[23,70] use the connectivity data of cell neighbors to traverse the unstructured mesh which can be split across a large number of MPI ranks as determined by the size of the near-body grid. Using a hybrid-solver framework is beneficial in this case because it enables us to use a structured off-body grid, rather than an unstructured one. This, in turn, helps reduce the computational cost of search algorithms. As discussed in Section 2.2, AMReX uses a patch-based arrangement of data, a huge benefit of which is that information can be stored in a compact form and duplicated on each MPI rank. Easy duplication of patch-based grid information on every MPI rank enables fast local search for operations including intersection checks and point containment enabled by trivial conversion of floating point numbers to indices in the patch-based grid system. For

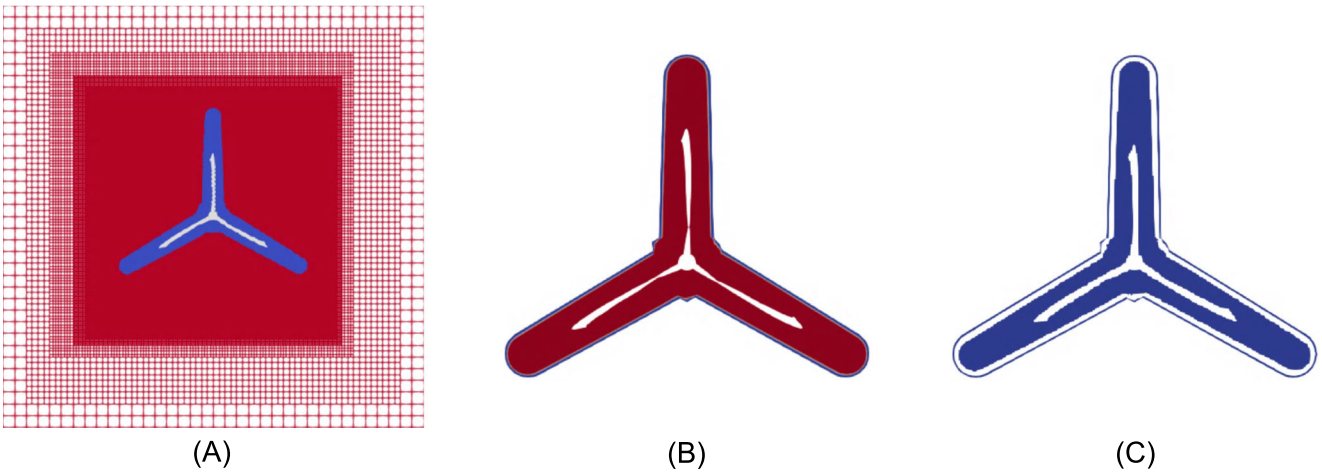

FIGURE 3 Hole cut associated with the NREL 5-MW overset grid in Figure 2 using Topology Independent Overset Grid Assembler (TIOGA): (A) Classification of cells in the off-body block-structured grid. (B) Classification of nodes in the near-body unstructured grid. (C) *Fringe* regions of both off-body and near-body grids. Red regions denote *field*, blue regions denote *fringe*, and white regions denote *hole*

simulations executed on GPUs, larger patch sizes are used to maximize efficiency of computing resources resulting in fewer number of overall patches enabling further speedup of the global search.

3.2 | Coupling algorithm

The coupling algorithm between Nalu-Wind and AMR-Wind follows an additive Schwarz-like approach which involves solving in parallel the system of discretized partial differential equations at the field points for both CFD solvers, followed by update of solution at the fringe points. There exists a lag in propagation of information between overlapping meshes due to the process of computing solution at field points followed by evaluating the solution at fringe points.[71] Compared with a single-linear-system solve[72–74], such a decoupled-linear-system solve provides several benefits to solving massively large system of equations,[71] including the use of grid-tailored linear solvers and grid-tailored algebraic multigrid (AMG) preconditioners which are essential for achieving rapid convergence of the pressure Poisson problem.[75] In addition, compared with a multiplicative/alternating Schwarz-like approach[76] which requires updating each solver with the latest solution of the other solver, an additive Schwarz-like approach allows for performing linear solves across the CFD solvers simultaneously, adding greatly to the parallelism of the entire system.

Algorithm 1 describes the AMR-Wind/Nalu-Wind coupling strategy. The linear solve at field points in steps 15 and 17 for every overlapping grid is computed with the solution at the fringe points constrained for the linear solve as discussed further in Section 4. The fringe points for every overlapping grid are then updated with the solution from the overlapping grid.

Algorithm 1 ExaWind driver

```
1:  Create MPI sub-communicators near-body-comm and off-body-comm
2:  if rank ⊂ near-body-comm then
3:      Initialize near-body solver                    ▷ Multiple instances of near-body solver allowed
4:  end if
5:  if rank ⊂ off-body-comm then
6:      Initialize off-body solver                     ▷ Only single instance allowed
7:  end if
8:  Initialize TIOGA                                   ▷ For all ranks
9:  for t = 0; t < T; t → t + 1 do                     ▷ Where T is the total number of time steps
10:     if near-body grid movement || off-body grid adaption || t == 0 then
11:         Perform overset connectivity
12:     end if
13:     Exchange overset solution
14:     for k = 0; k < K; k → k + 1 do                 ▷ Where K is the total number of Picard iterations
15:         Perform near-body time step
16:     end for
17:     Perform off-body time step                     ▷ Executed in parallel to the near-body time step
18:     if additional Picard iterations then
19:         Exchange overset solution
20:         for k = 0; k < K; k → k + 1 do             ▷ Where K is the total number of Picard iterations
21:             Perform near-body time step
22:         end for
23:     end if
24: end for
```

Currently, the ExaWind solver also allows an optional *additional Picard iterations* of Nalu-Wind (steps 18–23) in the same time step preceded by a second solution exchange. Additional solves of AMR-Wind within a time step are not possible given the explicit advection strategy employed by AMR-Wind. However, *additional Picard iterations* of Nalu-Wind allows for a near-body solve using overset boundary conditions as determined by the off-body solution at the same time step and enables stronger coupling between the overlapping grids. Multiple solution exchanges in general, within a time step, between the overlapping grids have been shown to achieve better order of accuracy as well as a smoother flow field across the overset interface for certain problems solved using decoupled linear systems such as those solved using ExaWind.[71] As discussed in the authors' previous work,[71] the need for stronger coupling increases as the Reynolds number decreases. In accordance, the need for *additional Picard iterations* in ExaWind has been observed only when solving laminar flow problems. The authors recommend using enough *additional Picard*

iterations to drop the nonlinear residual in Nalu-Wind by an order of magnitude, for which two *additional Picard iterations* have been sufficient in the authors' experience. *Additional Picard iterations* are set to zero by default and, unless specified, were not used for the numerical examples studied in this paper.

Note that the lack of an outer Picard-like loop in AMR-Wind means that the solution at fringe points in AMR-Wind is always lagging behind the solution at field points. While this does not greatly affect the quantities of interest as discussed in Section 5, the order-of-accuracy can be affected depending on the mesh resolution.[71] Furthermore, as discussed toward the end of Section 2.2, there is a mismatch in the solution update across the two CFD solvers. While this inconsistency is not shown to have any impacts on ExaWind simulations of wind turbines and wind farms, the influence on the formal order-of-accuracy of the overset framework is unclear. Work is underway toward both, introducing an outer coupling loop enabling multiple AMR-Wind iterations at any particular time step, as well as ensuring consistent temporal exchange of pressure across the two CFD solvers.

4 | LINEAR SOLVERS

Wind farm simulations require solving large sparse systems, making iterative solvers preferable to direct solvers. The linear systems in Nalu-Wind are thus solved using a Krylov subspace method, for example, conjugate gradient (CG), generalized minimal residual (GMRES), biconjugate gradient (BiCG), and so on. Nalu-Wind interfaces with Trilinos[54*] through its Tpetra package, using a GMRES solver with preconditioners such as symmetric Gauss-Seidel, Jacobi, and Incomplete LU factorization preconditioner. Nalu-Wind also has an option to interface with *hypre*[55†] to leverage (AMG) algorithms which are imperative for rapid convergence of the pressure Poisson problem (5).[75] In this paper, linear solvers from both Trilinos and *hypre* libraries were used to solve the system of equations for the numerical examples in Section 5. As long as the linear system residuals reduce by the same amount, there is not a stark difference between using the two libraries, and interfacing with them is largely a matter of personal preference. In the case of AMR-Wind, the linear system is constructed and the solvers are called through the AMReX interface.[77] AMR-Wind has the option to solve the linear system using the AMReX-native geometric MLMG solvers or utilize the GMRES solver in *hypre* for solving the more challenging linear systems associated with the MAC projection (12) or the nodal projection (14).

A detailed description surrounding use of *hypre* in Nalu-Wind and related GPU optimizations was published recently[57]. This includes specialized algorithms for linear system assembly, AMG setup, and AMG/Krylov solve. When utilized in Nalu-Wind, for all equations, we use a GMRES solver alongside a Boomer AMG preconditioner. For AMR-Wind, the same *hypre* algorithms are accessible through AMReX's hypre interface.[*] For the MAC and nodal projection solves, depending on the grid anisotropy associated with the overset problem as discussed below, GMRES solve with Boomer AMG preconditioner may be leveraged. The parameters of the various preconditioners in Nalu-Wind and AMR-Wind are constantly evolving for performance optimality, though the authors acknowledge that there may be a more optimal set of parameters available. Further details surrounding linear solver parameters and considerations made therein are omitted given the scope of the current work.

The linear solvers in both Nalu-Wind and AMR-Wind have been modified to include overset-based masking resulting in Dirichlet boundary conditions as determined by the fringe points. While the linear solvers in Nalu-Wind treat the masking as any other Dirichlet boundary condition, the geometric MLMG linear solver within AMReX has coarsening limitations in the presence of such a masking. As the masking is coarsened alongside the rest of the linear system, there is not a community-wide established approach on how to handle the restriction of these values and boundary conditions. This is especially problematic given that the masking pattern has an unstructured appearance to it. Currently, this issue is alleviated by the use of the *hypre* interface in AMReX to solve the coarse grid problem, that is, MLMG operates on the fine grid levels, and then at coarser levels, the *hypre* solver is activated; *hypre* is well-suited for this problem since it is based on an algebraic multigrid and can handle unstructured data. Currently, a geometric MLMG algorithm is under development with focus on coarsening strategies that account for the unstructured overset-based masking.

The time-to-solution as determined by time per time step can vary greatly depending on the linear solver settings and computational resources available, the CPU (ARM, Intel, etc.) and GPU (NVIDIA, AMD, etc.) architectures, and the corresponding compilers (GNU, Intel, Clang, etc.) used to build the ExaWind software stack. Improving the time-to-solution is a constant focus of the ExaWind team. The strong scaling limit as of the current study for Nalu-Wind is ≈ 15 thousand cells (per CPU) on CPUs and ≈ 50 thousand cells (per GPU) on GPUs. The strong scaling limit as of the current study for AMR-Wind is ≈ 15 thousand cells (per CPU) on CPUs and ≈ 1 million cells (per GPU) on GPUs. Scaling studies are consistently conducted on a variety of platforms including Intel CPUs, NVIDIA GPUs, and AMD GPUs.[78]

*https://trilinos.github.io/linear_solver.html.
†https://hypre.readthedocs.io/en/latest/ch-solvers.html.
*https://amrex-codes.github.io/amrex/docs_html/LinearSolvers_Chapter.html.

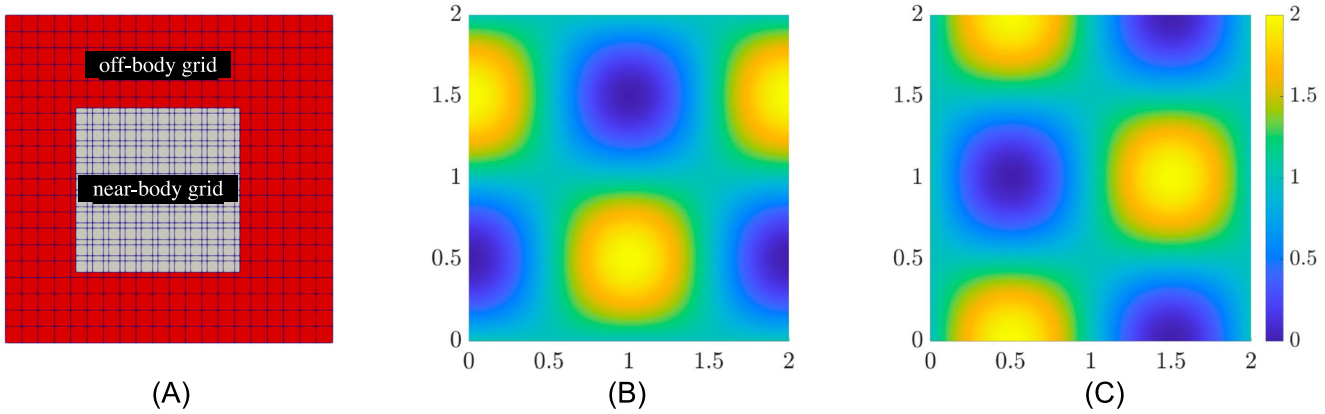

FIGURE 4 (A) Overset grid, initial solution for (B) u_x, and (C) u_y

5 | NUMERICAL EXAMPLES

In this section, we establish the formal order of accuracy for the ExaWind solver. We then establish the validity of the ExaWind solver for flow past bluff bodies by means of laminar flow past a sphere for a wide range of Reynolds numbers. Subsequently, we demonstrate the applicability of the ExaWind hybrid-solver framework to megawatt-scale turbine simulations. Validation studies against other CFD solvers and experimental results are presented for the NREL 5-MW[79] and the 2-MW NM80 rotor[80] under uniform inflow. Lastly, we demonstrate the applicability of the ExaWind solver to a small wind farm comprising NREL 5-MW turbines under different atmospheric stability states. It is important to note that, as mentioned in Section 1, for all studies presented in this manuscript, the near-body flow assumes rigid body behavior.

5.1 | Convecting Taylor vortex

Through the convecting Taylor vortex problem, we verify the order of accuracy of the ExaWind hybrid-solver framework. We consider a non-dimensionalized setup of the decaying convecting Taylor vortex problem[71] composed of two overlapping grids, an AMR-Wind off-body domain of $4 \times 4 \times 2$, and a Nalu-Wind near-body domain of $2 \times 2 \times 1$ as shown in Figure 4A. Similar grid size, Δx, is used in both grids with the near-body grid offset so as to not align with the off-body grid. The analytical solution, shown in Figure 4B,C at $t = 0$, is given by

$$\begin{aligned} u_x &= 1 - \cos(\pi(x-t))\sin(\pi(y-t))e^{-2\pi^2\mu t}, \\ u_y &= 1 + \sin(\pi(x-t))\cos(\pi(y-t))e^{-2\pi^2\mu t}, \end{aligned} \tag{25}$$

where $\mu = 0.001$ is the dynamic viscosity of the fluid. We simulate the problem till $t = 0.2$. Periodic boundary conditions are used in the off-body grid along all domain boundaries in the plane of flow, while symmetry boundary conditions are used for boundaries normal to the plane of flow.

Given the 2D nature of the flow, symmetry boundary conditions are also enforced on the near-body grid for boundaries normal to the plane of flow. Choices of time step sizes are dependent on the grid size as discussed below. At each time step, two *additional Picard iterations* are solved to achieve the tightest overset coupling possible as outlined in Section 3.2.

Given the separate discretization schemes, we compute separate errors for the two CFD solvers for the problem defined by (25). The L_2 error norm is defined as follows:

$$L_{2,\mathrm{NW}} = \sqrt{\frac{\sum_i^N \left(\phi_i - \phi_i^{exact}\right)^2}{N}}; \; L_{2,\mathrm{AW}} = \sqrt{\frac{\sum_i^N \left(\phi_i - \phi_i^{exact}\right)^2 V_i}{\sum V_i}}; \tag{26}$$

where ϕ_i^{exact} is the analytical solution at grid node i and cell i for Nalu-Wind and AMR-Wind, respectively. In the case of Nalu-Wind, the error is computed over the total number of nodes N including fringe and field points. In the case of AMR-Wind too, the error is computed over all cells with V_i denoting the volume of cell i.

While AMR-Wind* and Nalu-Wind[71] are independently second-order accurate in space and time, the overset-coupled simulations are limited to first-order accuracy in time. This is demonstrated by Figure 5 which presents the L_2 error norm comparison for two convergence studies.

*https://ExaWind.github.io/amr-wind/developer/verification.html.

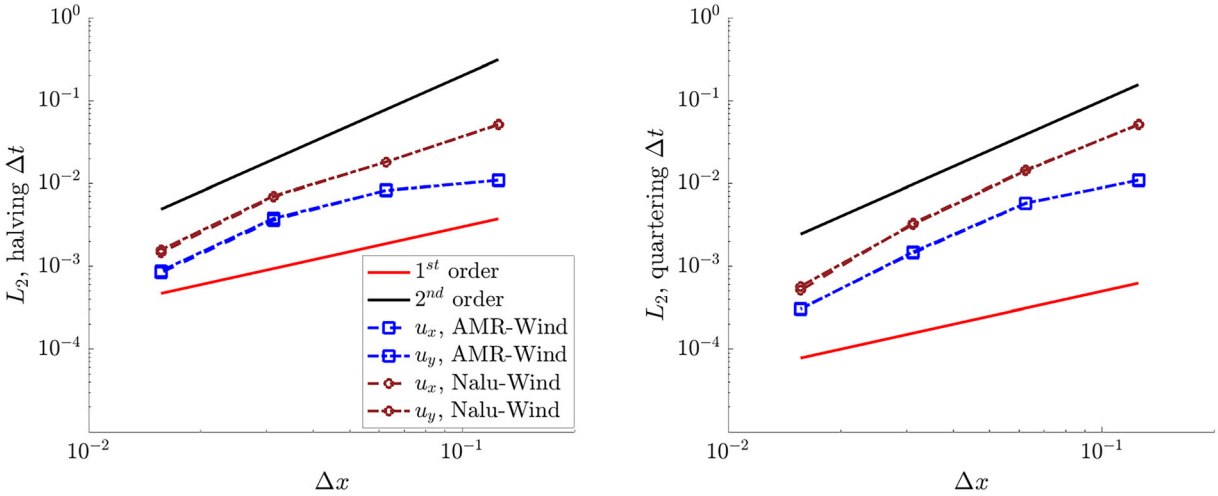

FIGURE 5 L_2 error for the convecting Taylor vortex as a function of grid size Δx. Order-of-accuracy studies are presented corresponding to when the time-step size is halved (left) and quartered (right) for each halving of the spatial grid size.

In the first comparison, the time step size is halved for every data point at which the grid size is halved. For the second convergence study, we quarter the time-step size for every data point at which the grid size is halved. While the former presents us with first-order accuracy in the L_2 error norm, the latter presents us with second-order accuracy provided Δx is sufficiently small. This is because when halving the time steps, the temporal error dominates signifying first-order accuracy in time which is hypothesized to be a combination of lack of an outer coupling loop which would enable multiple solution exchanges[71] between Nalu-Wind and AMR-Wind within a time step and staggered-in-time flow variables across the two CFD solvers as discussed in Section 2. Confirming these hypotheses would require development of new algorithms and is deemed outside the scope of the present study. Quartering the time step allows the spatial error to dominate which is expected to be second order for advection-dominated flows.[71]

While the current numerical example studies the order-of-accuracy for laminar flows and associated algorithms in ExaWind, the order-of-accuracy of hybrid RANS/LES simulations has not been addressed explicitly. The presented convecting Taylor vortex flow has a maximum cell Reynolds number of ≈ 50 and 400 for the smallest and largest grid sizes, respectively, and denotes advection-dominated flows. Considering that the order of accuracy for overset algorithms has been shown to have a heavy dependence on the Reynolds number of the flows,[71] the authors hypothesize that for heavily advection-dominated hybrid RANS/LES flows, the order of accuracy would follow that of other advection-dominated problems such as the convecting Taylor vortex presented here, provided sufficient number of solution exchanges are performed within a time step to arrive at a solution consistent with a fully coupled (single linear system) overset solve.[71] A more thorough study to establish the formal order of accuracy for the ExaWind solver is a topic of future research following the implementation of an outer coupling loop which would enable multiple solution exchanges between Nalu-Wind and AMR-Wind within a time step. Additionally, as discussed by Sharma et al[71] and as is demonstrated throughout rest of the paper, lack of stronger coupling in the absence of an outer coupling loop does not affect the aerodynamic quantities of interest or the smoothness of the flow fields visualized across the overset interfaces.

5.2 | Direct numerical simulation (DNS) of flow past a sphere

We use the ExaWind solver to study laminar flow past a sphere. This is a fundamental study of canonical flow past a bluff body, for which there is experimental and numerical reference data.[81–85] The physical phenomena present herein such as complex asymmetric flow, shedding, and interacting vortical wake structures, are commonly found in wind farm simulations. Of particular interest is the transition of the flow and wake from one solver to the next considering that the errors in the transition can rapidly degrade the solution in the entire domain. We study DNS of flow past a sphere across a wide range of Reynolds numbers: $Re = 100$, 300, 500, 750, and 1000. This range of conditions was chosen based on the availability of reference experimental and numerical results.[81–85] To the best of the authors' knowledge, this is the first time that an overset methodology was used to perform DNS of flow past a sphere for the range of Reynolds numbers considered here.

We choose a sphere of diameter $D = 1$ m, centered at $x = y = z = 0$ in a rectangular domain with corners at $(-16D, -16D, -16D)$ and $(48D, 16D, 16D)$. The inflow velocity is set to $U_\infty = 1$ m/s. The density is set to 1 kg/m^3. The dynamic viscosity is varied to achieve the desired Reynolds number: $\mu = 0.01$, 0.00333, 0.002, 0.00133, and 0.001 kg/(m s). Because the objective of this study is DNS, special care was taken to achieve adequate grid resolutions. Benchmark grid generation guidelines previously published for sphere simulations at the Reynolds numbers of

interest were used.[83,85] The surface grid for the spherical wall is initialized as a cube, projected onto the sphere, and then smoothed using an elliptic solver until the cell sizes were uniform for equal spacing in the axial and azimuthal directions, $\Delta\theta$ and $\Delta\phi$, respectively. The near-body grid was then extruded in the wall-normal direction, denoted by s here, to $0.5D$. This wall normal extrusion uses a first cell height of $\Delta s_{\min} = 1.13/10\sqrt{Re}$, with a growth factor of 1.03 applied to subsequent extruded cells until a maximum grid size of $\Delta s_{\max} = 0.05D\sqrt{Re_{\text{ref}}/Re}$ is reached, where $Re_{\text{ref}} = 300$. The cell sizes remain constant thereafter until the extrusion reaches a distance of $0.5D$ away from the sphere surface. The resolution constraints in the axial and azimuthal directions follow Nagata el al.[85] The off-body grids for AMR-Wind have the same base grid resolution: $64 \times 32 \times 32$ cells or $\Delta x = \Delta y = \Delta z = 1D$. There are four levels of refinement for the $Re = 100$ and $Re = 300$ simulations and five levels of refinement for the higher Reynolds number flows. The first two levels of refinement span $[-4.5, -4.5, -4.5] \times [16,4.5,4.5]D$. The finer levels of refinement span $[-4.5, -4.5, -4.5] \times [4.5,4.5,4.5]D$. Table 1 summarizes the grids used for the different cases. Figure 6 presents an example of the overset grid used for this study.

For this problem, two Picard iterations were used alongside two *additional Picard iterations* for a stable evolution of the pressure field across the overlapping grids. The simulations used a fixed time-step size that targeted a Courant number of approximately 0.5 for $\Delta t = 0.01\Gamma, 0.005\Gamma, 0.004\Gamma, 0.004\Gamma$, and 0.0025Γ. All simulations were run for more than 140Γ, where $\Gamma = D/U_\infty$ is one sphere flow-through time, before data were collected for reporting. All temporally averaged data, for example, drag force, lateral force, and pressure coefficients, were averaged over the last 40Γ of the simulation. The following coefficients are analyzed in this work:

$$C_D = \frac{F_D}{\left(\frac{\rho U_\infty^2}{2}\right)\left(\frac{\pi D^2}{4}\right)}, \tag{27}$$

$$C_L = \frac{F_L}{\left(\frac{\rho U_\infty^2}{2}\right)\left(\frac{\pi D^2}{4}\right)}, \tag{28}$$

$$C_S = \frac{F_S}{\left(\frac{\rho U_\infty^2}{2}\right)\left(\frac{\pi D^2}{4}\right)}, \tag{29}$$

$$C_P = \frac{p}{\left(\frac{\rho U_\infty^2}{2}\right)}, \tag{30}$$

TABLE 1 Summary of grids used for simulations of flow over a sphere. Here θ, ϕ, and s denote the axial, azimuthal, and wall normal directions. Note, $\Delta x = \Delta y = \Delta z$ in the AMR-Wind background grid.

	Nalu-Wind near-body grid				AMR-Wind background grid			
Re	$\Delta\theta = \Delta\phi$	$\Delta s_{\min}$	$\Delta s_{\max}$	**# cells**	Δx_{base}	$\Delta x_{\min}$	**max levels**	**# cells**
100	$0.0153D$	$0.01D$	$0.085D$	300k	$1D$	$0.0625D$	4	5.5 M
300	$0.0123D$	$0.005D$	$0.05D$	720k	$1D$	$0.0625D$	4	5.5 M
500	$0.0103D$	$0.0045D$	$0.038D$	1.1 M	$1D$	$0.03125D$	5	33.5 M
750	$0.0096D$	$0.004D$	$0.031D$	1.3 M	$1D$	$0.03125D$	5	33.5 M
1000	$0.0088D$	$0.003D$	$0.027D$	1.7 M	$1D$	$0.03125D$	5	33.5 M

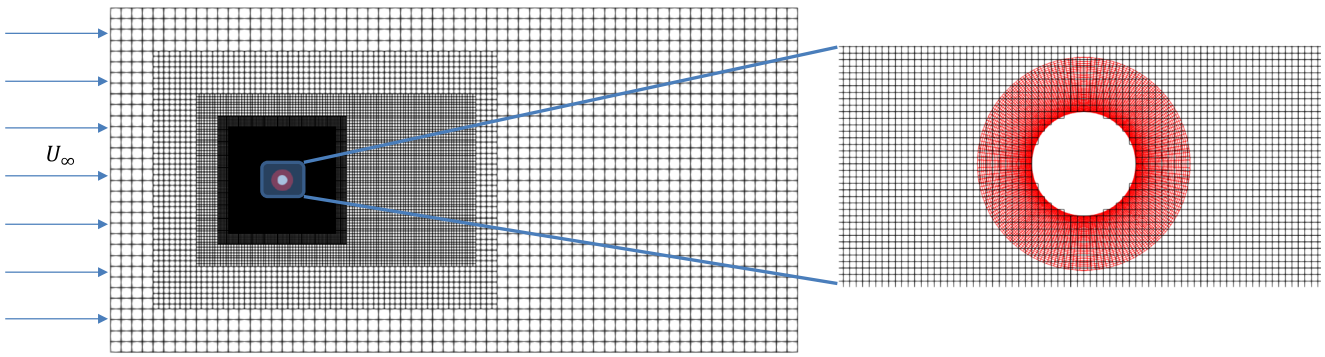

FIGURE 6 Overset grid used for $Re = 300$ flow over a sphere.

$$C_f = \frac{\sigma_{wallx}}{\left(\frac{\rho U_\infty^2}{2}\right)}, \tag{31}$$

where C_D is the drag coefficient, F_D is the drag force, C_L is the lateral force coefficient, C_S is the side force coefficient corresponding to F_S, the force normal to the mean flow symmetry plane, and F_L is the force orthogonal to F_S. The pressure coefficient is denoted by C_P. The skin friction coefficient denoted by C_f is evaluated using σ_{wallx}, the component of the skin shear stress vector, on the sphere surface, in the direction of inflow. In addition to these coefficients, we also analyze the Strouhal number which is evaluated based on the peaks identified in the C_L spectra.

Flow field visualization is provided for various Reynolds numbers in Figure 7.

As expected, the wake loses its quasi-steady nature with increasing Reynolds number in addition to the presence of hairpin vortical structures in the off-body region and wake breakdown visible in the near-body region. The interface between the two solvers presents a smooth flow field. The drag coefficient and the Strouhal number are shown in Figure 8.

The drag coefficient is the most represented quantity of interest in the literature, and excellent agreement is observed in comparison to both experimental and numerical data. For the Strouhal number, we report a maximum difference of $\approx 7\%$ with respect to the literature corresponding to $Re = 500$. Differences in Strouhal number are largely attributed to the noise in evaluating the peak in the C_L spectra which is not periodic for $Re > 300$; notwithstanding these differences, computed values follow the general trend reported in literature. At $Re = 300$, we observe an oscillation amplitude in the drag coefficient of 3.3×10^{-3}, close to results reported by other numerical simulations: 3.5×10^{-3} by Johnson and Patel[83] and 2.8×10^{-3} by Tomboulides and Orszag.[84] The lateral force coefficient oscillation amplitude at $Re = 300$ is observed to be 0.02, close to the value of 0.016 reported by Johnson and Patel.[83] The lateral force coefficient, and consequently the Strouhal number, is zero at $Re = 100$ as no

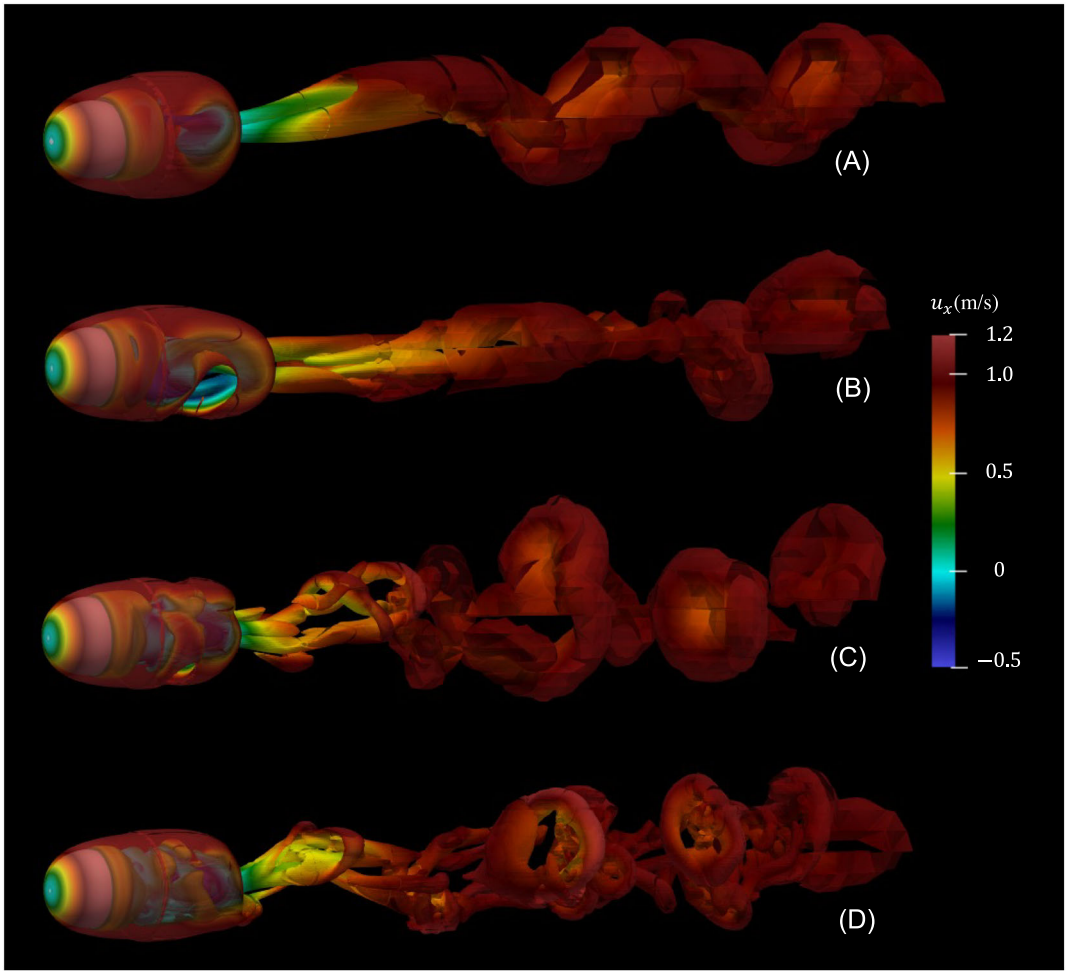

FIGURE 7 Visualization of Q-criterion isocontours at 5×10^{-4} colored by streamwise velocity u_x for flow over a sphere: (A) $Re = 300$; (B) $Re = 500$; (C) $Re = 750$; (D) $Re = 1000$

shedding is induced at such a low Reynolds number. For the sole reported value in literature, our results match the C_L value at $Re = 300$ within 6%.[83] Table 2 presents a summary of the force coefficients.

The pressure and skin friction coefficients are shown in Figure 9, for which there is little reference data.

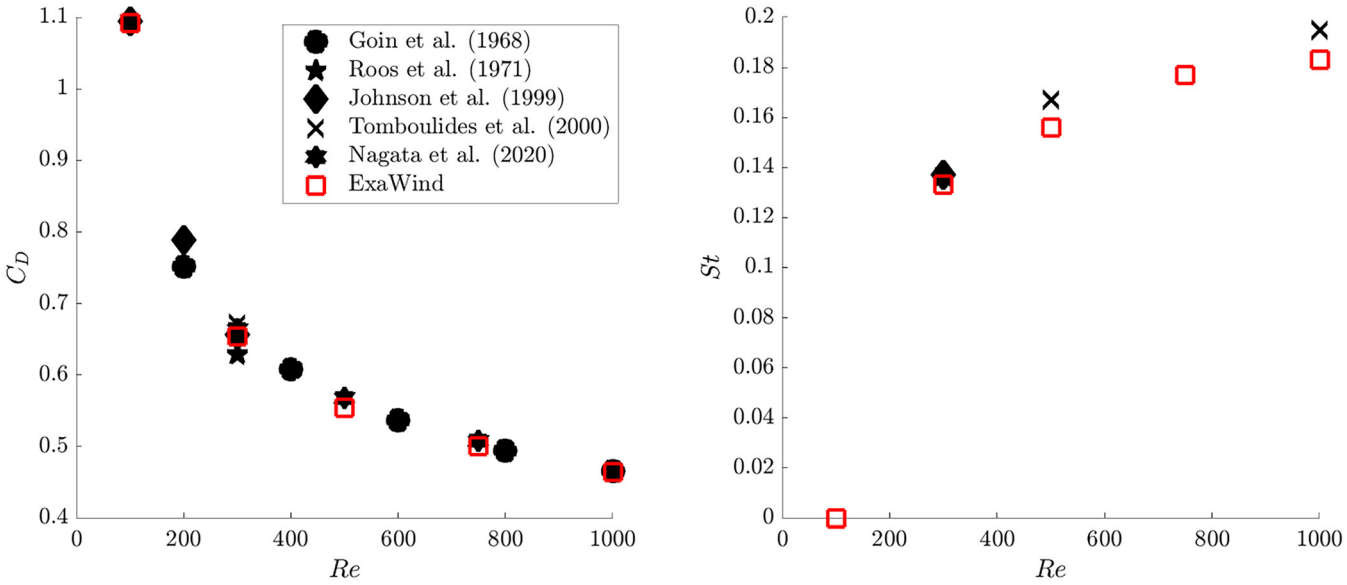

FIGURE 8 Drag coefficient and the Strouhal number (computed from the C_L spectra) as a function of Reynolds number.

TABLE 2 Summary of force coefficients for flow past a sphere.

Re	C_D	C_L	*St*
100	1.0843	0.0	0.0
300	0.656	−0.0649	0.133
500	0.556	−0.0465	0.156
750	0.501	−0.0146	0.177
1000	0.459	−0.0103	0.183

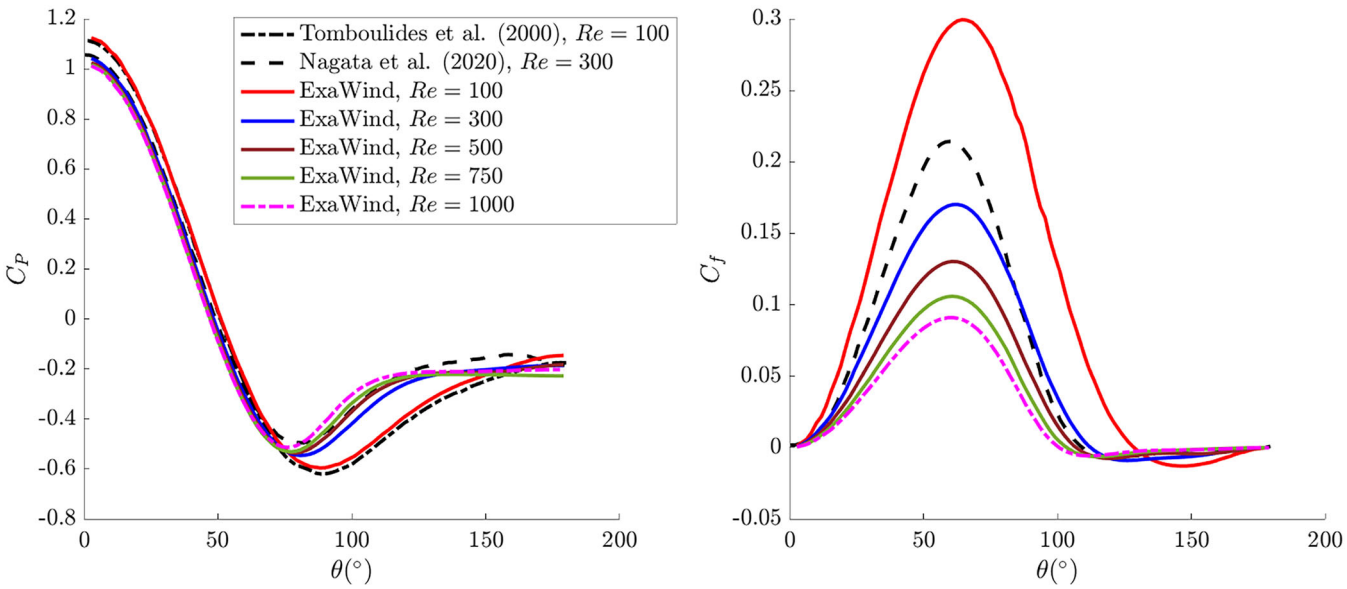

FIGURE 9 Pressure and skin friction coefficients as a function of azimuthal angle, θ.

However at $Re = 100$, the pressure coefficient matches the reference data.[84] At $Re = 300$, there is good agreement with reference pressure coefficient data.[85] For the skin friction coefficient, the data exhibit the same trend as reported in literature,[85] that is, maximum skin friction decreases as a function of *Re*. However, discrepancies with the reference data are observed in the skin friction coefficient at $Re = 300$. We hypothesize that the differences between the reference data and our results occur because the reference data were assessed at a Mach number of 0.3, allowing for compressibility effects to influence the resulting flow field. The additional data here that do not have a reference counterpart are presented to enable comparisons for future studies.

5.3 | RANS simulations of the NREL 5-MW rotor

The NREL 5-MW turbine[79] is a 126 m diameter reference turbine, designed for offshore wind research. Although a theoretical turbine, it is widely used in the wind research community for comparisons against other simulation tools published in the literature. For this study, the turbine's geometry was simplified by ignoring the tower and nacelle structures; only the three blades and the hub were modeled, all assumed to be rigid. The blade surface geometry was modeled using structured grid in spanwise and chordwise directions (Figure 10C), and the volume grid was generated using hyperbolic extrusion. The wall-normal spacing was set to 10^{-5} m for the first point off the wall.

To ensure a smooth transition to the hub, the blade surface's structured grid was constructed outboard of the 20% span. The sections inboard use unstructured grid to transition smoothly to the hub grid as shown in Figure 10. This amounts to a near-body grid of 8.67 million cells. The off-body block-structured Cartesian grid, shown in Figure 2, spans a domain of $1.9\text{km} \times 0.94\text{km} \times 0.94\text{km}$ giving the wake plenty of room to expand. The off-body grid consists of four levels of refinement with the base-level grid size set to ≈ 7.3 m. The levels of refinement, about 5*D* from the inflow boundary and centered in the spanwise directions, serve to accurately model the turbine wake, covering the wake region including the rotor geometry (positioned toward the front of the refinement grids) and span a domain of $535\text{m} \times 150\text{m} \times 150\text{m}$. All levels of refinement combine for a total of ≈ 169 million cells.

For simulating both the near-body and off-body flow, the $k-\omega$ SST model described in Section 2.3 was used to model the turbulence. The outer-domain boundary conditions imposed are slip along the lateral boundaries and zero pressure at the outflow boundary. The solution between Nalu-Wind and AMR-Wind is exchanged once, at the end of every time step, as detailed in Section 3.2 without any *additional Picard iterations*.

Simulations were performed for the NREL 5-MW rotor operating at uniform inflow of $U_\infty = 6 - 11$ m/s in a surrounding fluid with density and viscosity set to that of air at 288 K, that is, 1.23 kg/m^3 and $1.8e-5$ kg/(m s), respectively. This setup corresponds to a *Re* number of O(10^6).

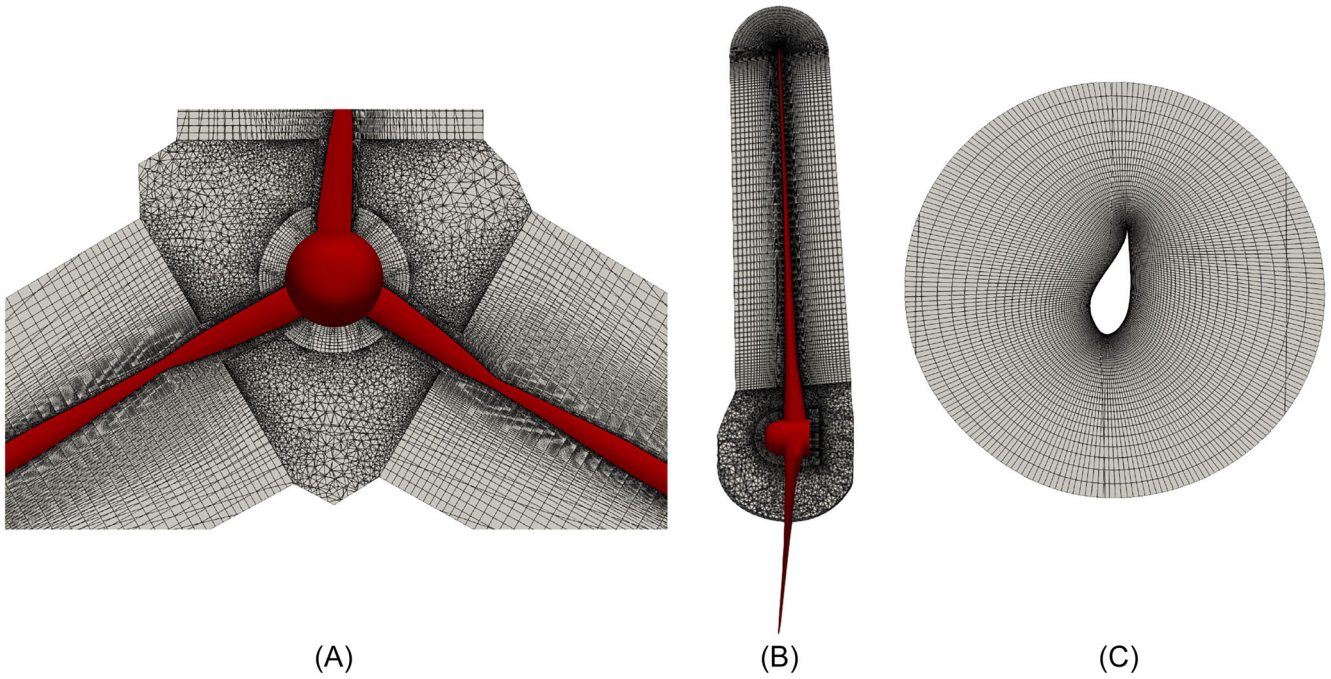

FIGURE 10 The near-body grid used to model the NREL 5-MW rotor with Nalu-Wind including the surface blade and hub grid: (A) front view; (B) side view; (C) curviliniear grid around the blades. The grid consists of a structured hyperbolically extruded grid on the rotor blades and a fully unstructured grid (consisting of pyramid, tetrahedral, and wedge element types) around the hub and the hub-blade transition region. The overset grid is shown in Figure 2.

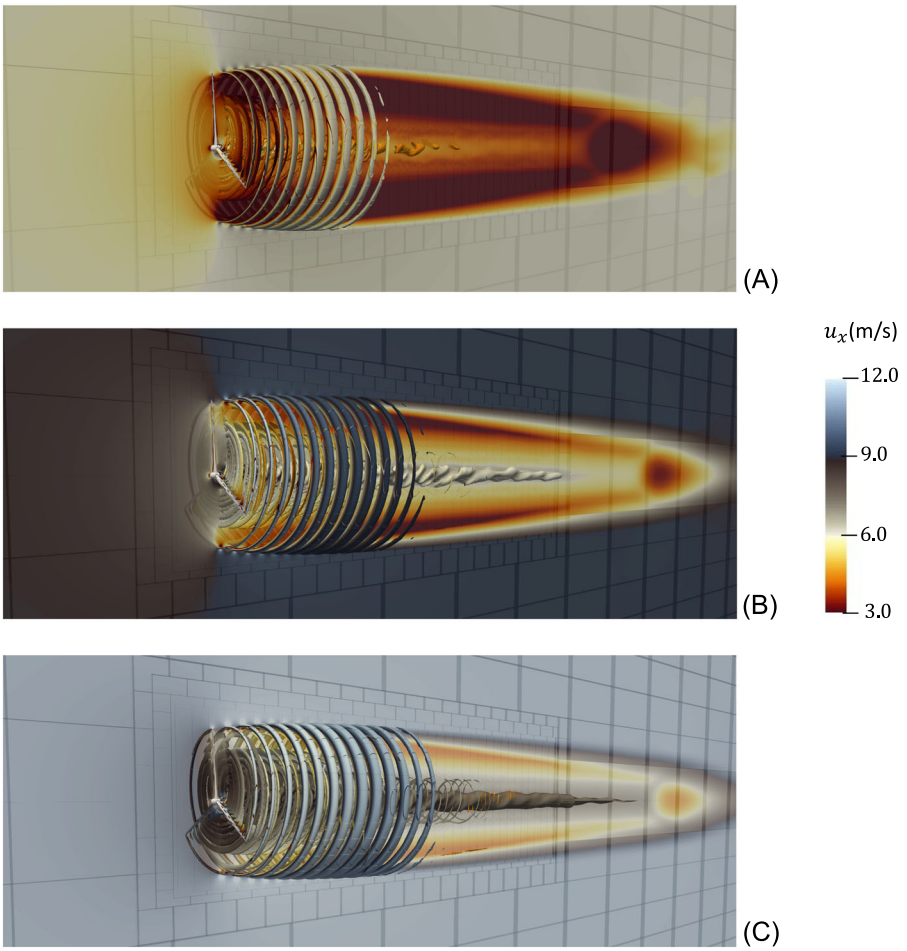

FIGURE 11 Q-criterion isocontours at 0.01, with velocity visualized in the wake for NREL 5-MW rotor operating under uniform inflow wind speeds of (A) 7 m/s, (B) 9 m/s, and (C) 11 m/s. Of note are the hub vortices which are difficult to capture using actuator parameterizations of the rotor. The background grid plot also visualizes the regions of grid refinement.

A fixed time-step size was used such that the blade rotates 0.25° each time step. Note that the time-step size varies as a function of wind speed because the RPM varies as a function of wind speed below the rated wind speed.[79] Figure 11 displays Q-criterion flow-field isocontours for the NREL 5-MW rotor at different inflow velocities.

As expected, the flow structures demonstrate longer regions with intact tip vortices at higher speeds. In contrast at slower speeds, the tip vortices disintegrate, mixing with the surrounding turbulent air quicker. Figure 12 shows the obtained power and thrust curves compared with other CFD results[37,86,87].

ExaWind simulations were only performed for wind speeds below the rated wind speed of 11.4 m/s because there was no pitch controller active for these simulations. The pitch controller is necessary for simulations above the rated wind speed and will be explored with fluid–structure coupling in future studies. The results demonstrate excellent agreement with other published CFD simulations, validating the ExaWind hybrid-solver's capability as compared with the state-of-art simulation capabilities in the space of high-fidelity modeling of wind farm physics.

5.4 | RANS simulations of the NM80 rotor

The NM80 turbine, manufactured by Neg Micon, is an 80 m rotor diameter turbine analyzed in the DAN-AERO MW experiments, a research project between Risø Technical University of Denmark (DTU) and industry.[80] The experiments were conducted to establish benchmark data sets for in-depth analysis of design tools for megawatt-scale turbines, primarily focusing on the differences in aerodynamic behavior measured under 2D steady conditions in a wind tunnel in contrast to unsteady 3D flow conditions of a rotor under atmospheric inflow. For the purpose of this study, the turbine geometry was simplified by ignoring the tower, hub, and nacelle structures, and only the three blades were modeled. We use the grid generated by DTU for this turbine as part of the International Energy Agency (IEA) Wind Task 29[10]. Figure 13 presents the near-body grid for the NM80 rotor.

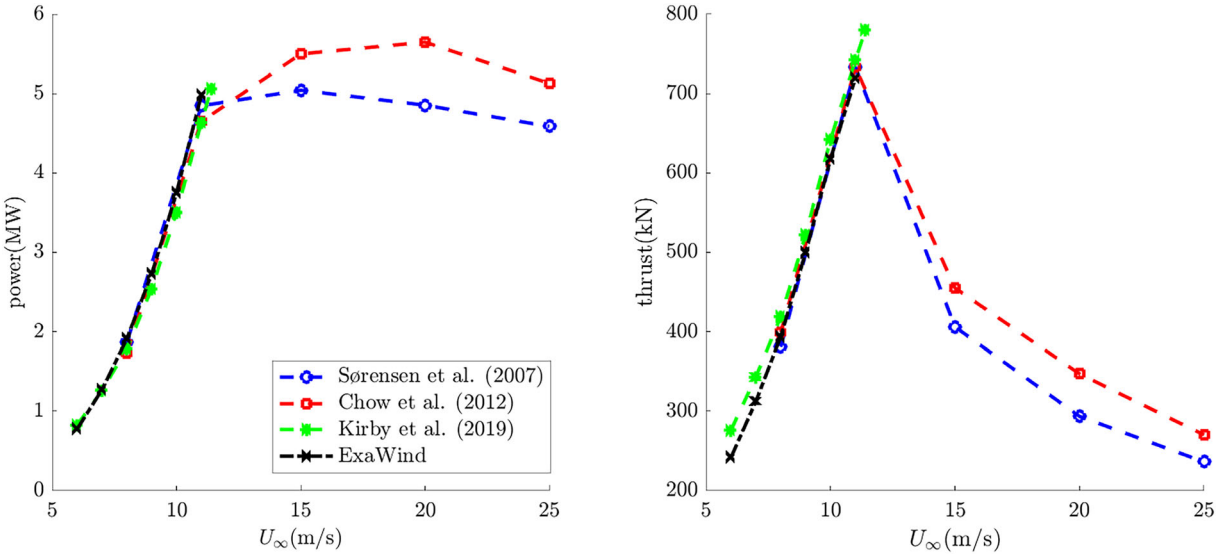

FIGURE 12 Power and thrust as a function of the inflow speed U_∞ as evaluated using the ExaWind solver. Results are compared against other high-fidelity CFD results from the literature.[37,86,87]

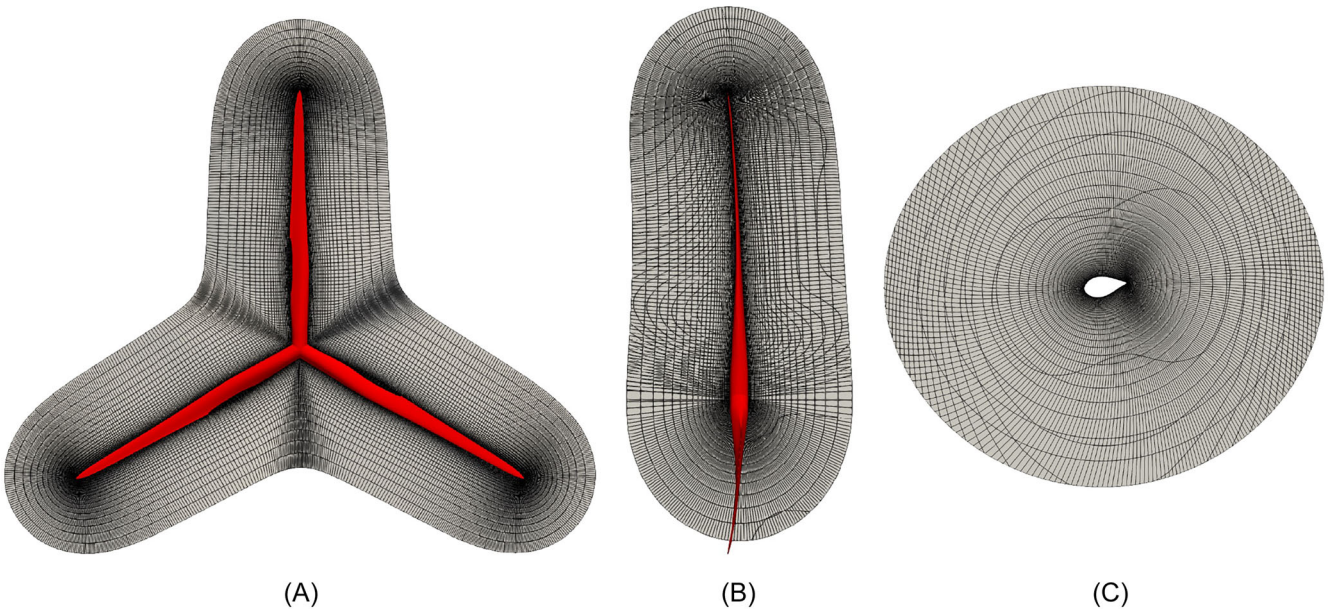

FIGURE 13 The Nalu-Wind near-body grid for the 2-MW NM80 rotor including the surface blade grid: (A) front view; (B) side view; (C) grid around the blades.

The near-body grid uses hexahedral elements and incorporates a quadrilateral surface representation of the blade, as well as a hyperbolic grid that extends away from it.[33] The wall-normal spacing for the first point off the wall was set to 10^{-6} m resulting in $y^+ < 1$. This amounts to a near-body curvilinear grid of 10.6 million cells. The off-body block-structured Cartesian grid spans a domain of $2\text{km} \times 1\text{km} \times 1\text{km}$, and consists of five levels of refinement with the base-level grid size set to ≈ 16 m. The first four levels of refinement, placed about $9.5D$ from the inflow boundary and centered in the spanwise directions, serve to accurately model the turbine wake, covering the wake region including the turbine, and span a domain of $340\text{m} \times 96\text{m} \times 96\text{m}$. The finest level of refinement, positioned to align with the front of the other refinement zones, spans a domain of $40\text{m} \times 96\text{m} \times 96\text{m}$ encompassing the near-body turbine grid for effective overset interpolation. All levels of refinement combine for a total of ≈ 10.3 million cells. The ground/terrain boundary is not resolved, instead, domain boundaries are set far from the rotor to prevent the aerodynamic forces from being affected. The outer-domain boundary conditions imposed are slip along the lateral boundaries, and zero pressure at the outflow boundary.

Two different configurations are studied. First, an idealized axisymmetric case under uniform inflow with no tilt, yaw, or shear is considered. The second case considered is that of dominating shear where a power-law profile with a shear exponent of 0.249 was used as the inflow, corresponding to the experiments.[33] For both cases, *Re* is O(10^6). Other relevant problem setup details for the two cases are summarized in Table 3.

The turbulence modeling approach and surrounding fluid properties are similar to that described in Section 5.3. A fixed time-step size was used for rotor rotation of 0.25° each time step.

A few assumptions have been made in this comparison study. The rotor blades are considered rigid, and the effects from the tower, nacelle, and hub are not modeled. The rotor geometry is assumed to be smooth and free of irregularities usually present on any turbine in the field. Also as mentioned above, no ground/terrain effects on the flow are considered. Finally, a power law is used to represent the atmospheric turbulent sheared flow profile measured by a met mast located 200 m (2.5*D*) from the turbine.

For the axisymmetric case, the experimental data used for comparison were averaged over any revolution that resembles axisymmetric flow conditions around the rotor, i.e., no yaw and very low shear. Comparisons were performed for any one of the blades. The pressure distribution across the blade cross-section is shown in Figure 14 at various locations along the blade length. Given that absolute pressure in incompressible flow is physically irrelevant, the pressure distribution obtained using ExaWind is shifted by some constant value to be inline with the field data.

Very good agreement is observed between the ExaWind solver and the experimental data as well as CFD results obtained by Ellipsys3D.[33] Compared with the experimental measurements, some discrepancy is observed, primarily near the trailing edge of the blade cross-section, although the results from the two CFD solvers match well. This discrepancy could be due to a multitude of assumptions described above including assumptions on axisymmetric and uniform inflow, but in all practicality there may be some degree of azimuthal variation in the experimental data.

TABLE 3 Setup details for the NM80 rotor simulations.

Case	Axisymmetric	Dominating shear
U_∞(m/s) at hub height	6.1	9.792
Tilt(°)	-	5
Yaw(°)	-	6.02
Pitch(°)	−0.15	4.75
Rotor speed (RPM)	12.3	16.2

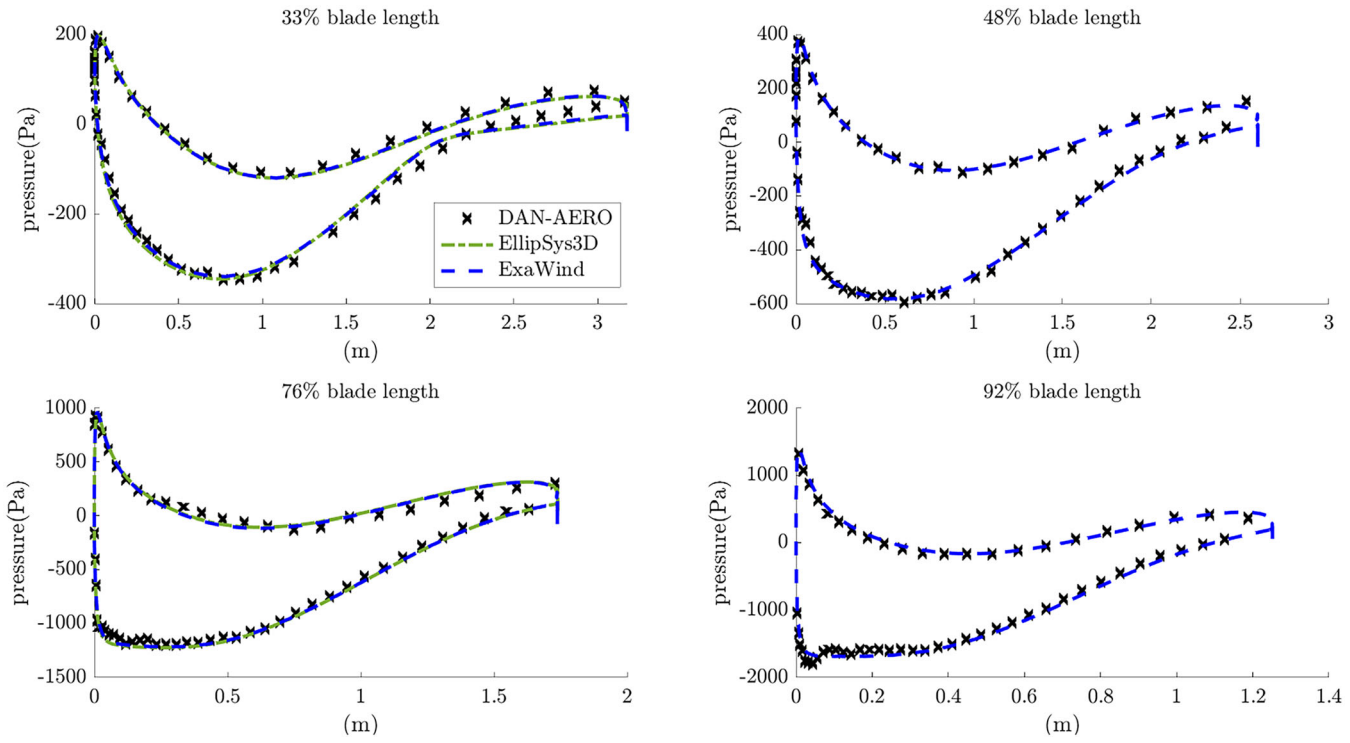

FIGURE 14 Pressure distribution across the blade cross-section at various locations along the blade length for the axisymmetric case. Results are compared against DAN-AERO experimental data[80] and results obtained using EllipSys3D.[33]

Considering the host of assumptions, the results match well with the experimental data, depicting that the ExaWind framework performs very well in this simple flow scenario. Note that at the trailing edge, a vertical line is present in the CFD results obtained from both ExaWind and Ellipsys3D. This vertical line is not considered a discrepancy or a numerical artifact and exists because the airfoils have a blunt trailing edge that is resolved in the near-body rotor grid using multiple cells, and these points are included in Figure 14 and all subsequent comparison plots presented in this numerical example. In comparison, as is evident with the experimental data, no measurements were recorded on the trailing edge which is standard of field measurements due to insufficient room for sensor placement. Conversion of this 3D blunt trailing edge data into a line plot gives the appearance of multivalued behavior of pressure at the trailing edge.

In the high-shear, tilt, and low-yaw case, turbine blades experience changing wind speeds as they rotate which creates large dissimilarities in free stream wind resulting in an increase around 90° and 180° azimuthal angles, and a decrease around 0° and 270° azimuthal angles as compared to the axisymmetric case. Consequently, pressure is plotted for one of the blades at azimuthal angles of $\theta = 0°, 180°$ as shown in Figures 15 – 16.

Results are compared against the DAN-AERO experimental data. As seen in Figures 15 and 16, ExaWind computations are consistent with the experimental results albeit with some discrepancy in regions marked by sudden jumps in experimental data denoting faulty sensor readings. Similar to the discrepancy near the trailing edge observed in the axisymmetric case, a discrepancy near the trailing edge is observed for azimuthal angle $\theta = 0°$. Conversely, for $\theta = 180°$, some discrepancy is observed near the leading edge. Note that, although small, these pressure differences will translate to bigger differences in the forces obtained by integrating the pressure across significantly large cross-section areas of the rotor blade. A likely reason behind the discrepancy observed here stems from not capturing the inflow accurately and fitting a power law profile to resemble the turbulent sheared inflow profile as characterized by a met mast tower situated 200 m from the turbine in the field. Higher fidelity field measurements are necessary for a more consistent comparison against field data.

5.5 | Hybrid-RANS/LES wind farm simulations in turbulent atmospheric flow

Following the validation studies for various megawatt-scale turbines, in this example, we demonstrate the applicability of the ExaWind solver to a small wind farm comprising four NREL 5-MW turbines. We analyze the performance of the wind farm under neutral and strongly stable atmospheric stability states to contrast the different atmospheric conditions that turbines might experience under operation. For a consistent analysis across the two stability states, some uniformity is maintained in the overall setup of the problem as described below.

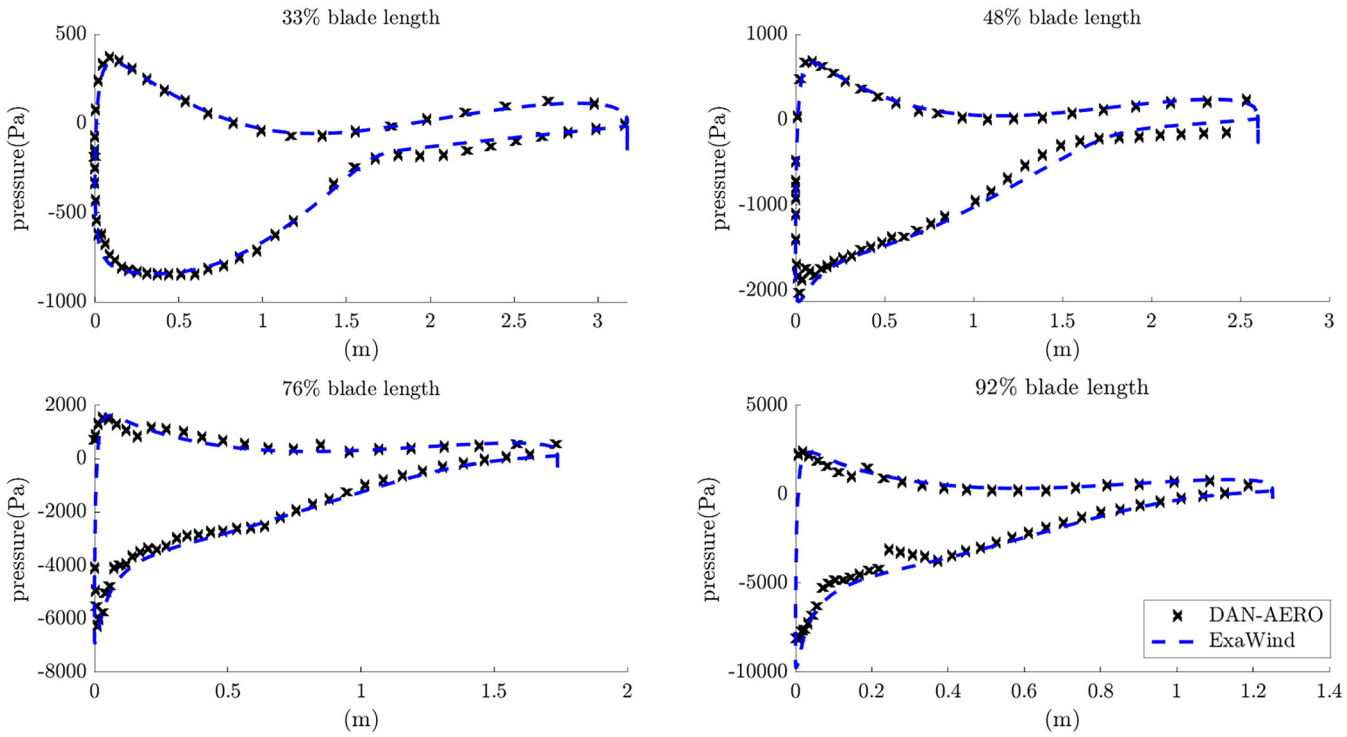

FIGURE 15 Pressure distribution across the blade cross-section at various locations along the blade length for the dominating shear case. Results are compared against DAN-AERO experimental data[80] for azimuthal angle $\theta = 0°$.

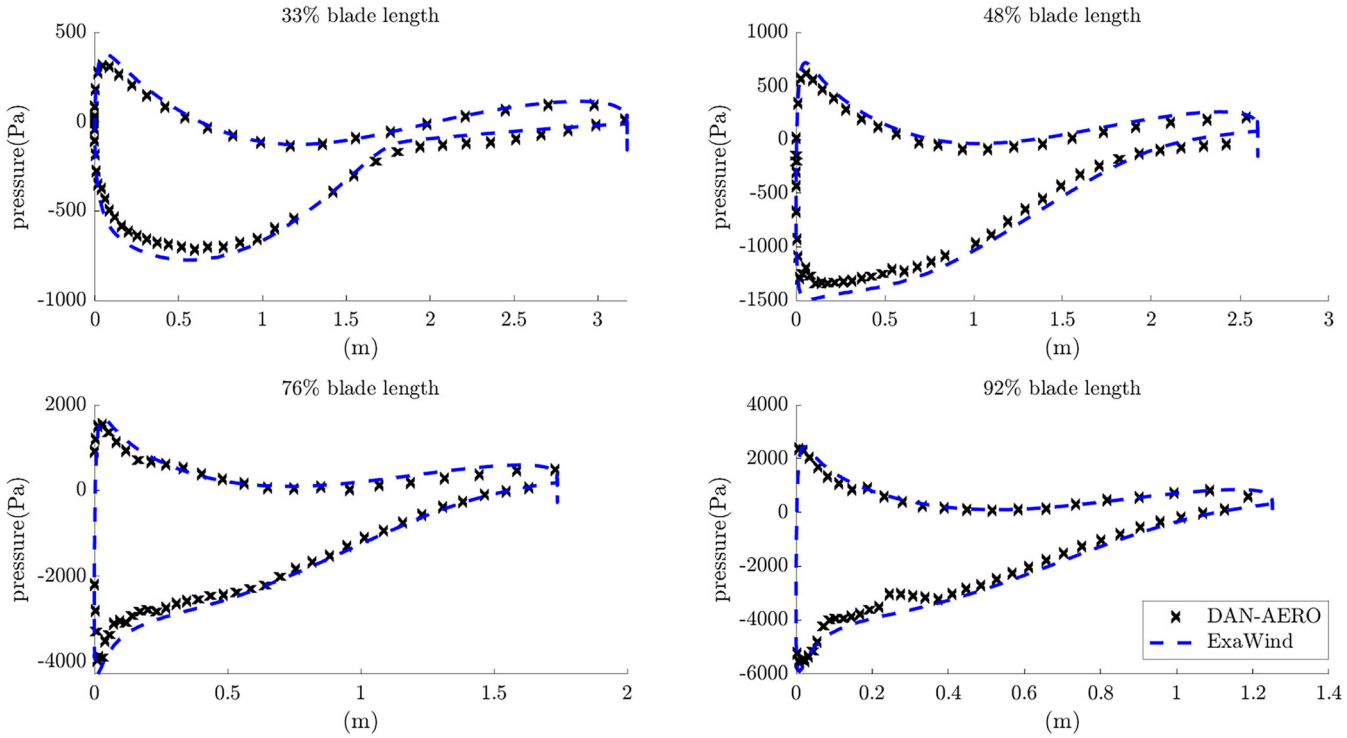

FIGURE 16 Pressure distribution across the blade cross-section at various locations along the blade length for the dominating shear case. Results are compared against DAN-AERO experimental data[80] for azimuthal angle $\theta = 180^{\circ}$.

A domain of size $1.9\text{km} \times 1.5\text{km} \times 0.94\text{km}$ is used alongside three levels of refinement as shown in Figure 17 for both stability states. All levels of refinement combine for a total of ≈ 615 million cells. The near-body grids described in Section 5.3 are used.

A mean flow of 10 m/s is enforced at the turbine hub height for both stability states simulated. Since the controller is not active, all turbines are rotated at a fixed RPM corresponding to the mean flow speed of 10 m/s.[79] This might result in nonoptimal RPMs as the wind speed varies, especially for turbine 4 which is positioned to be in the wake of turbine 1. For simulating the near-body turbine fluid dynamics, the $k-\omega$ SST model ((23) and (24)) was used to model the turbulence. The far-field turbulence is modeled using the LES model discussed in (18). The outer-domain boundary conditions imposed are slip along the top boundary, zero pressure at the outflow boundaries, and wall boundary conditions ((19) - (22)) along the bottom boundary. The solution between Nalu-Wind and AMR-Wind is exchanged once at the end of every time step, as detailed in Algorithm 1 without any *additional Picard iterations*.

We adopt a three-step approach for the wind farm simulations presented herein. First, a precursor ABL with the desired statistics is developed in a periodic domain without the turbines present. Once the ABL is developed, as deemed by converged flow statistics, the precursor is restarted to store the inflow boundary data. The simulation time upon restart corresponds to the physical time of the blade-resolved simulation. A second simulation restart includes the turbines with inflow generated using the stored boundary data.

It is important to highlight the wide range of scales applicable to wind farm simulation. The problems solved in this section involve modeling the blade-boundary layers with a wall-normal spacing of 10^{-5} m while also resolving large-scale O(km) ABLs. Altogether, ExaWind solves for flow phenomena spanning eight orders of magnitude in the ensuing simulations.

5.5.1 | Neutral ABL

For our first demonstration study, we consider a neutral ABL with statistics presented in Figure 18. The turbulent intensity for this ABL is $\approx 5.5\%$. The inversion height of the neutral ABL is capped at ≈ 600 m. Upon accounting for the Coriolis force, the wind direction at hub height corresponds to 225° by meteorological conventions which dictate that north is positive y-axis and east is positive x-axis. The height of the capping is specified by manually defining the potential temperature along the height of the domain, including a zero surface potential temperature flux. The capping height was chosen to be representative of neutral ABLs in nature. For the presented results, the boundary layer developed over the course of ≈ 34 h of physical time.

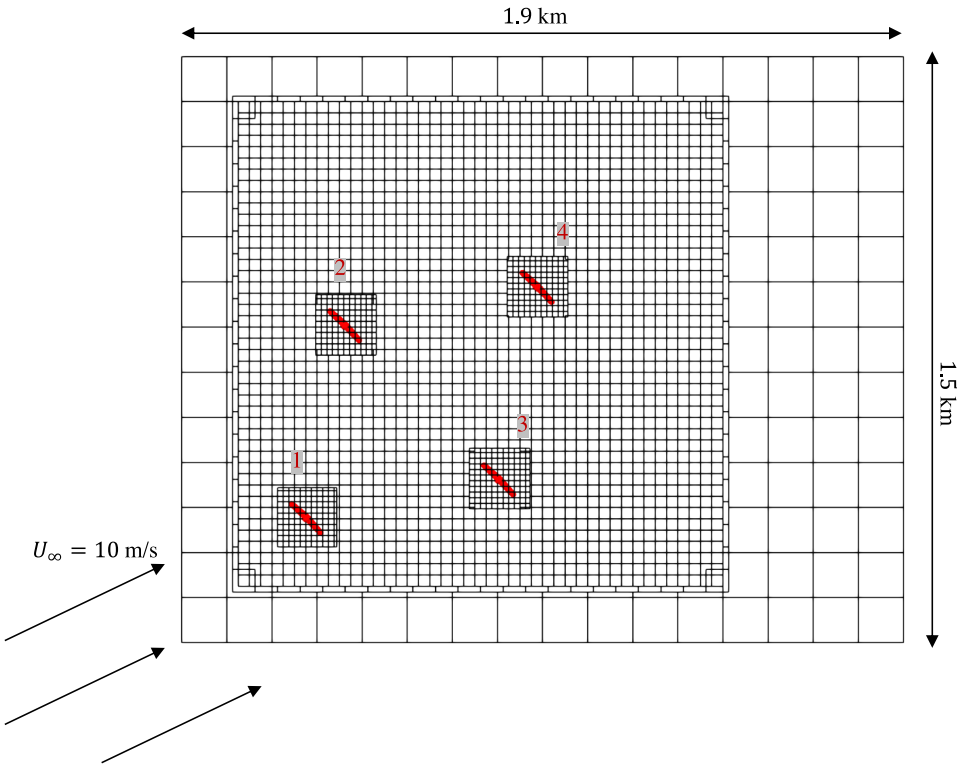

FIGURE 17 Top view of the layout of turbines for the demonstration studies, and corresponding grid refinements in the off-body mesh with domain size 1.9km × 1.5km × 0.94km. A base-level mesh resolution of ≈3.7 m is used alongside three levels of refinement. The first two levels of refinement serve to accurately model the turbine wake and cover the wake region including the turbine, and span a domain of 1250m × 1250m × 150m. The finest level of refinement spans a domain of 150 m^3 encompassing the near-body turbine grid for effective overset interpolation. The (x,y) locations of the four turbines are chosen to be (325 m, 325 m), (425 m, 825 m), (825 m, 425 m), and (925 m, 925 m) relative to the inflow boundaries.

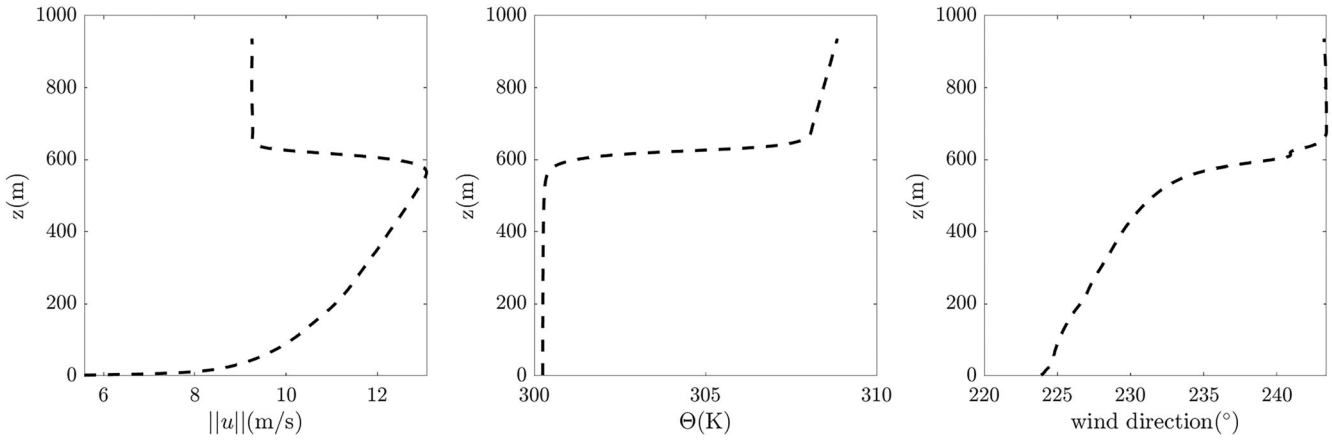

FIGURE 18 Time-averaged and horizontally averaged vertical profile statistics for a neutral ABL, including horizontal velocity profile $||u||$ (left), temperature profile Θ (middle), and wind direction (right).

We simulate the problem for ≈500 s of physical time which corresponds to about two flow-through times comprising 95 rotor rotations at an RPM consistent with that of $U_\infty = 10$ m/s.[79] Figure 19 presents the evolution of wake, turbine-wake interaction, and ABL-wake interaction within the wind farm.

Of note are the large turbulent ABL structures. Figure 20 presents the corresponding evolution of power and thrust for every turbine.

All turbines generate widely varying power. This is primarily because of the larger flow structures prevalent in a neutral ABL where a turbulence packet is likely to interact with a turbine over several rotations, generating power and thrust proportional to the mean velocity of the

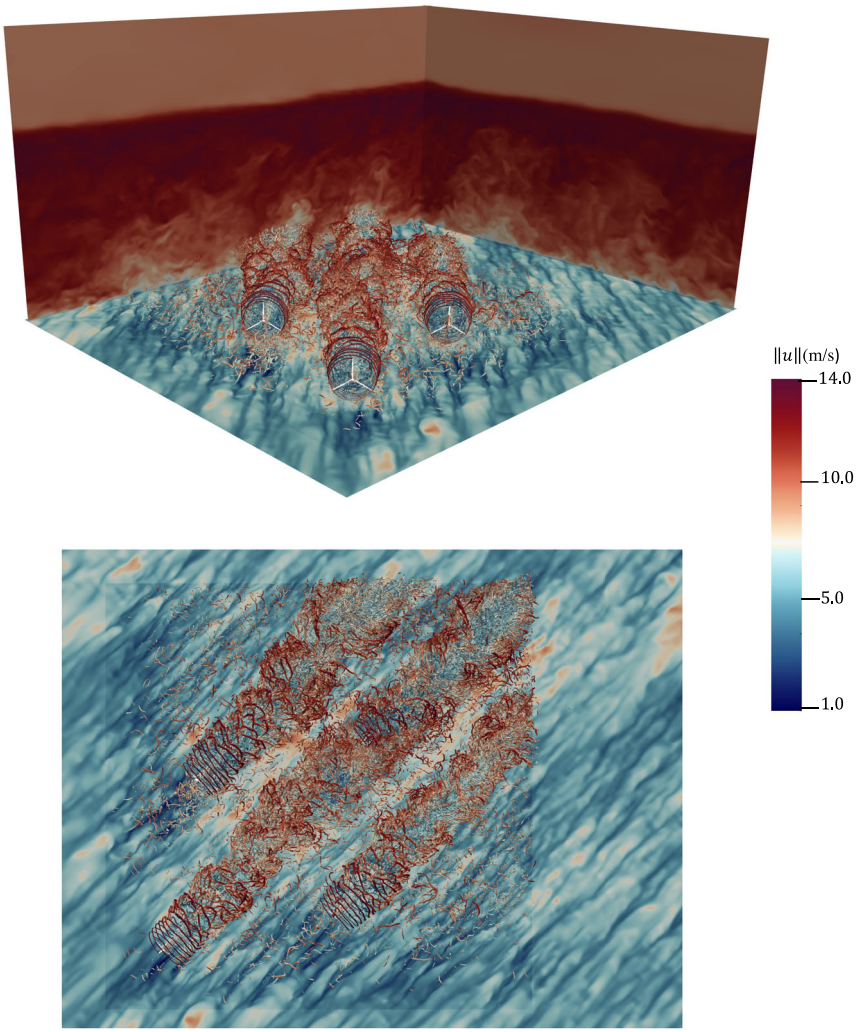

FIGURE 19 Q-criterion isocontours at 0.25 with velocity visualized for wake interaction between turbines in a small wind farm with a neutral ABL. Q-criterion of 0.25 corresponds to higher vorticity regions of the flow which need finer grid resolution. As a result, they cut off at the boundaries of the grid refinements shown in Figure 17. This does not imply that the wake only exists within the refinement zones, just that the vortical structures are not supported by the coarser grid size.

turbulent ABL structures. To minimize the influence of temporally varying turbulent ABL structures on the power and thrust calculations, the neutral case would need to be analyzed in a time-averaged manner for much larger time periods compared with the largest turbulent structures. As is also expected, turbine 4 consistently generates overall lower power given that it sits directly in the wake of the upstream turbines.

Figure 21 presents the wake profile as averaged over the last 220 s of the physical time. The velocity magnitude $||u||$ and total turbulent kinetic energy, $k = k^{SFS} + 0.5\frac{1}{T}\sum^{T}((\boldsymbol{u}-\overline{\boldsymbol{u}})\cdot(\boldsymbol{u}-\overline{\boldsymbol{u}}))$, are sampled at distances $||d|| = D, 2D, 3D$ behind rotors 1 and 4.

As expected, the average velocity in the wake of turbine 1 (Figure 21A) at $||d|| = D$ is higher than the average velocity in the wake of turbine 4 (Figure 21B) which is operating directly in the wake of turbine 1. Correspondingly, the average total turbulent kinetic energy is higher behind turbine 4 (Figure 21D) at $||d|| = D$ compared with the total turbulent kinetic energy behind turbine 1 (Figure 21C). Further downstream of the turbines at $||d|| = 2D, 3D$, the total turbulent kinetic energy increases, denoting more turbulent mixing, which results in wake recovery as observed in the average velocity plots.

It is worth noting that the wake profiles are asymmetrical which is a result of various factors. In the vertical direction, there is asymmetry due to the variation in wind direction ($\approx 2^{\circ}$) as well as wind velocity (≈ 2 m/s) across the cross-section of the rotor, as shown in Figure 18. Further asymmetry results from the fact that the wakes were not averaged over a time greater than the largest eddy length scale computed as dividing the capping height by the friction velocity.[88] For the neutral ABL presented here, this time scale is evaluated to be >1300 s. In contrast, the wake profiles in Figure 21 were averaged over a much shorter duration of 220 s as mentioned above. The shorter averaging window is in general susceptible to asymmetry due to temporally varying turbulent ABL structures and also prone to cutting off some of the larger eddies. It is also

important to note that asymmetries in the wake contours have been also been observed experimentally,[89] and some degree of asymmetry is expected as long as wind shear, veer, and wake swirl are present in the farm-level flow.

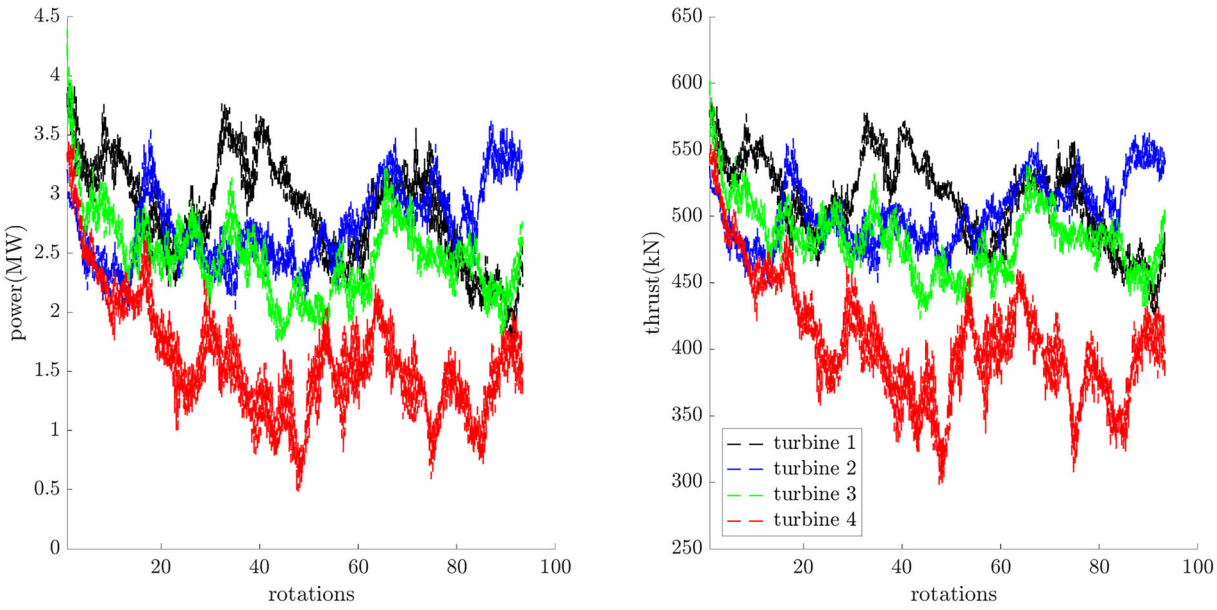

FIGURE 20 Power and thrust evolution for turbines corresponding to Figure 19.

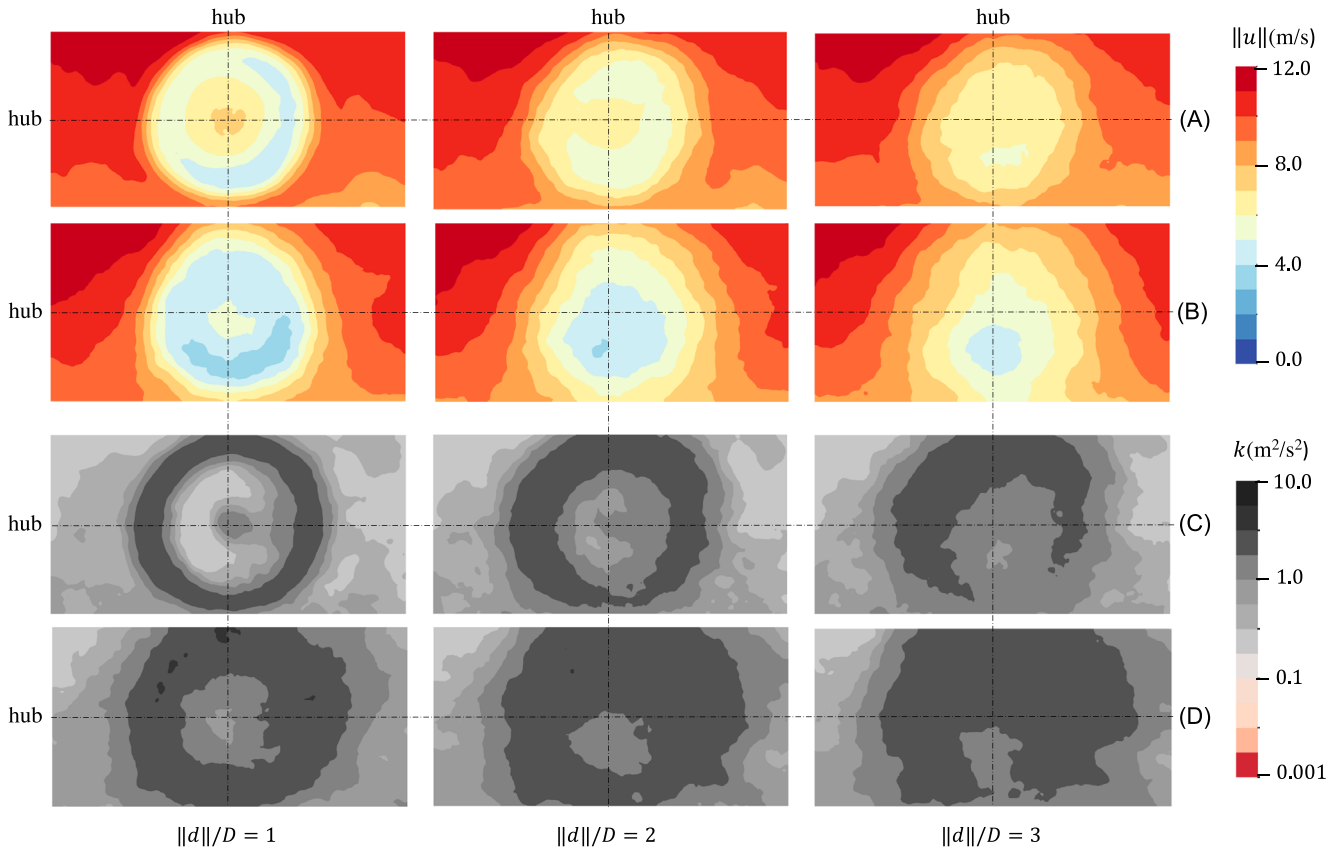

FIGURE 21 Velocity magnitude $||u||$ and total turbulent kinetic energy k sampled at distances $||d|| = D, 2D, 3D$ behind turbine 1 (A, C) and turbine 4 (B, D) for the neutral atmospheric boundary layer (ABL) case. The quantities are averaged over the last 220 s of physical time. The sampling planes are 300m × 150m centered at the turbine hubs.

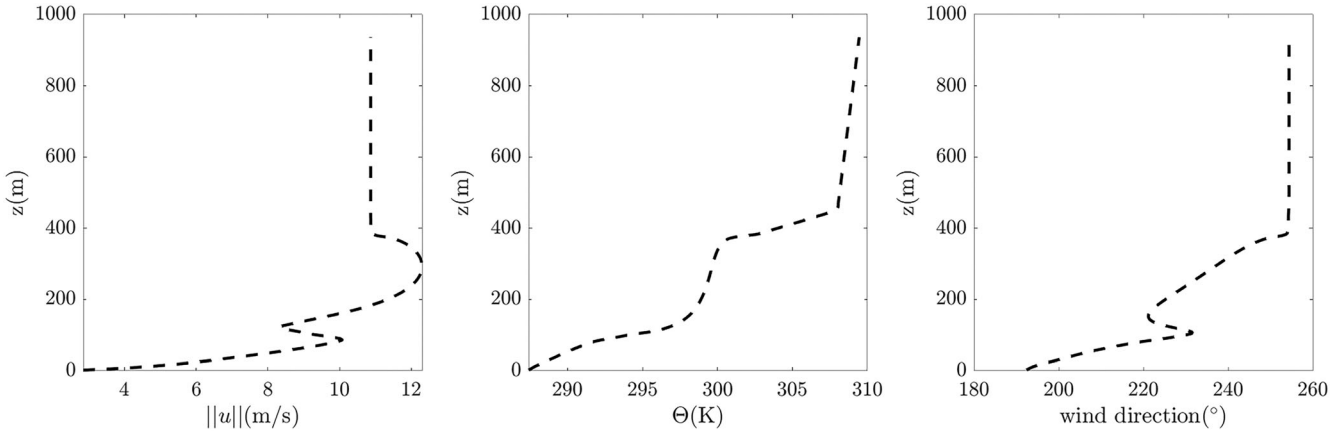

FIGURE 22 Time-averaged and horizontally averaged vertical profile statistics for a strongly stable atmospheric boundary layer (ABL) with low-level jet, including horizontal velocity profile $||u||$ (left), temperature profile Θ (middle), and wind direction (right).

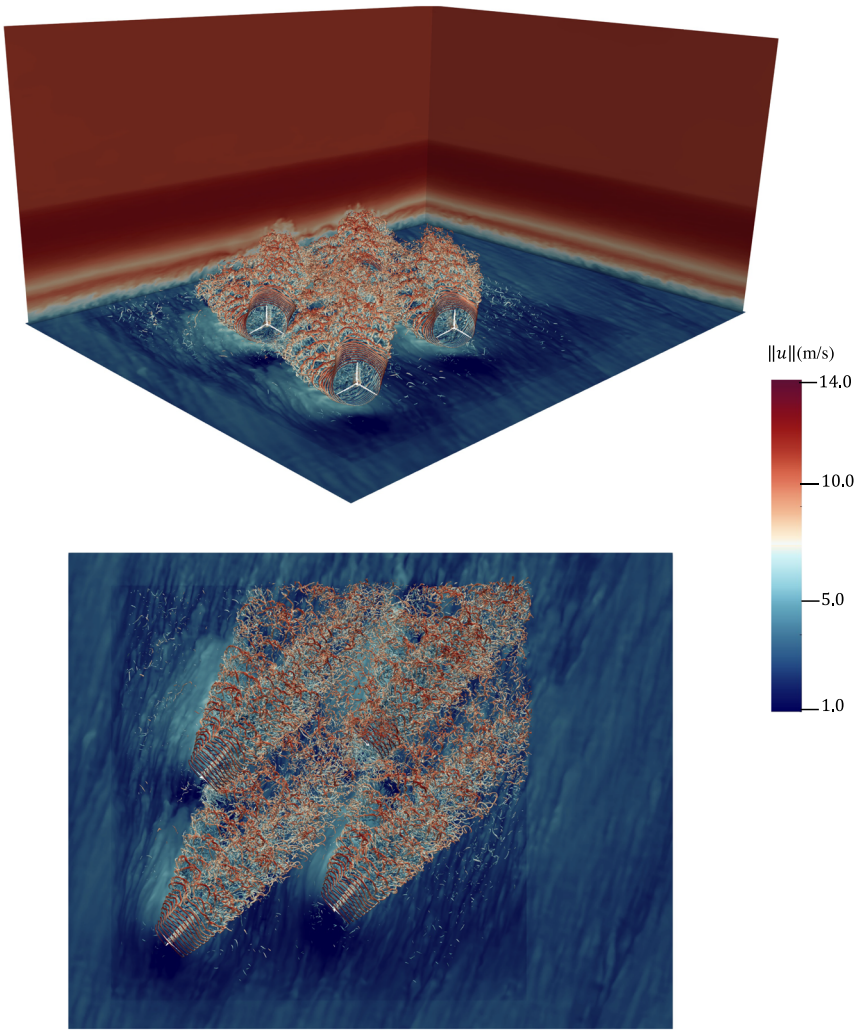

FIGURE 23 Q-criterion isocontours at 0.25 with velocity visualized for wake interaction between turbines in a small wind farm with a low-level jet. The visualization of isocontours cutting off within the computational domain is discussed in Figure 19.

5.5.2 | Strongly stable ABL with low-level jet

For our second demonstration study, we consider a strongly stable ABL with statistics presented in Figure 22.

The turbulent intensity for this ABL is $\approx 1.3\%$. Upon accounting for Coriolis force, the wind direction corresponds to 225° by meteorological conventions. This was enforced by defining the mean velocity to be consistent with the neutral ABL case. The height of the ABL is constrained by an applied inversion layer at $\approx z = 400$ m. The capping of the ABL is based on the user-specified definition of potential temperature along the height of the domain. The capping inversion height was chosen to be representative of ABLs in nature. A surface heat flux is specified which cools the air near the ground, and causes the boundary layer to become stably stratified near the ground. The surface heat flux applied is commensurate with the desired potential temperature and stability profile. Ongoing work for future simulations of ABLs involves integrating AMR-Wind with Weather Research and Forecasting (WRF) inputs, which include surface heat flux, so that the boundary conditions are consistent with mesoscale modeling inputs. For the presented results, the boundary layer developed over the course of ≈ 28 h of physical time. The long ABL development time is required due to the slower vertical transport of momentum under stable stratification conditions.

Note, the NREL-5MW turbine has a hub height of 90 m and a rotor diameter of 126 m, that is, the rotor sweeps an area with vertical domain ranging from ≈ 27 to ≈ 153 m. Analyzing the resultant velocity profile as a function of height in Figure 22, steep velocity gradients are observed in the region swept by the turbine blades, including formation of a low-level jet which has implications on the power and thrust generated by the turbines. Compared with the neutral ABL case, an identical wind speed and direction was applied at hub-height, so the ABL forcing scheme resulted in the same horizontally averaged $U_\infty = 10$ m/s from 225° at $z = 90$m. However, unlike the neutral ABL case, the presence of a strong temperature stratification, combined with a long precursor development time for the stable ABL case, leads to high shear and veer emerging below the inversion layer height. However, because the velocity and wind direction were constrained at hub height, this resulted in a very strong low-level jet appearing around the hub height, with regions of notably high shear and veer emerging both below and above.

The blade-resolved simulations are run for ≈ 500 s of physical time which corresponds to about two flow-through times comprising 95 rotor rotations at an RPM consistent with that of $U_\infty = 10$ m/s.[79] Figure 23 presents the evolution of wake, turbine–wake interaction, and ABL–wake interaction within the wind farm.

Of note are the much smaller turbulent ABL structures compared with the turbulent ABL structures observed in the neutral case, representative of stable ABLs. Also of note are the steep velocity gradients along the height of the domain including low-level jets commonly encountered in nighttime and during neutral/stable transition periods. The turbines used in this study are 153 m high. As the industry moves offshore, larger turbines are becoming prevalent with future turbines expected to get large enough to possibly impact ABLs with lower capping inversion heights. Figure 24 presents the corresponding evolution of power and thrust for every turbine.

As expected, turbines 1, 2, and 3 generate comparable amounts of power and thrust given their access to unobstructed wind. Turbine 4 generates power comparable with the other turbines for approximately 20 rotations after which the wake from the upstream turbines begin interacting with turbine 4 resulting in a steep decline in power and thrust as is observed in Figure 24. The overall power and thrust generated for the

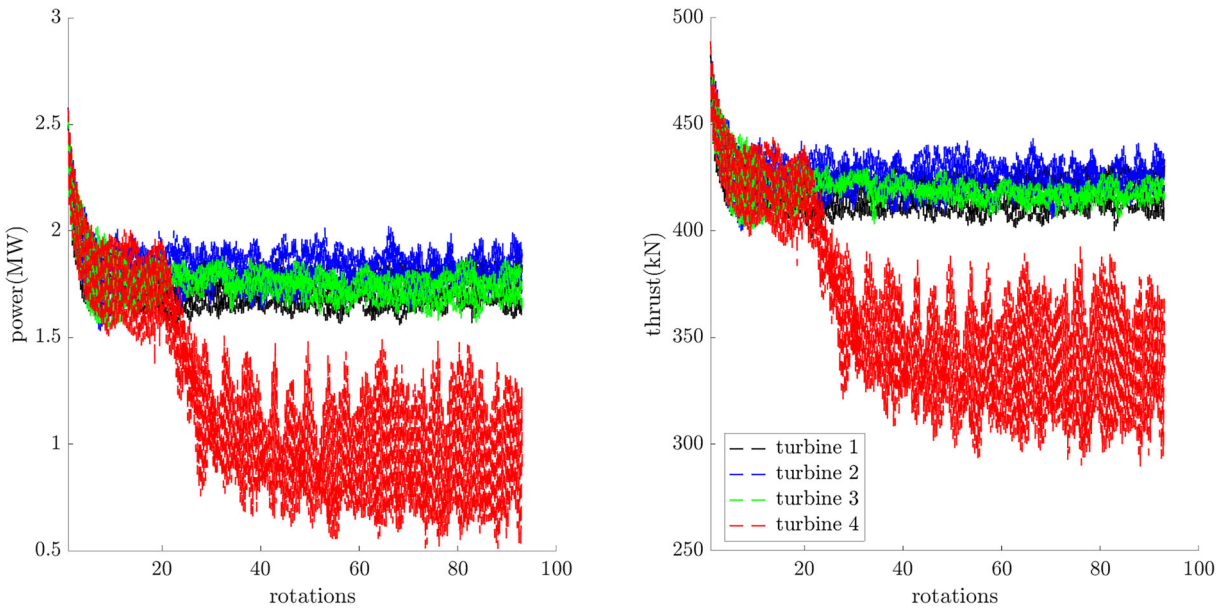

FIGURE 24 Power and thrust evolution for turbines corresponding to Figure 23.

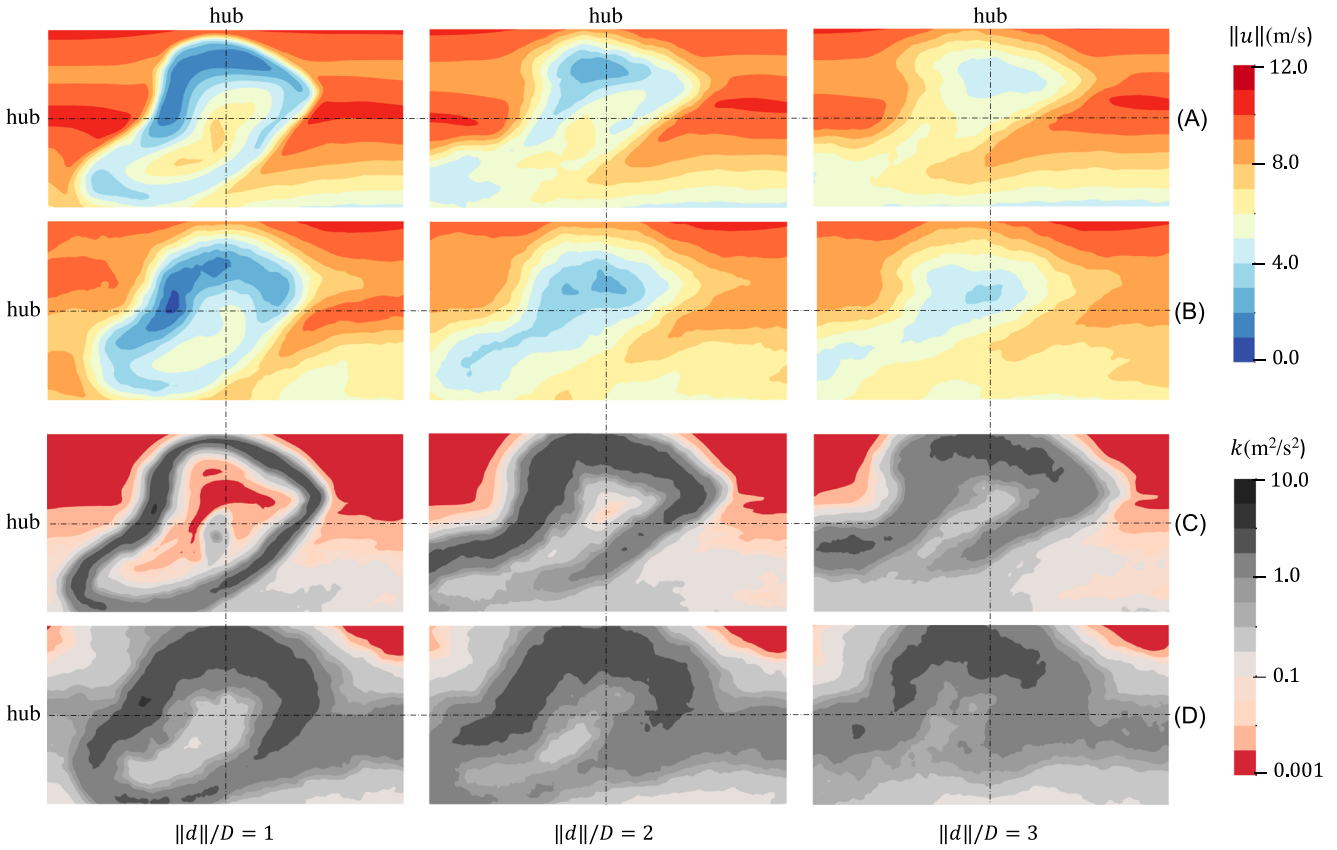

FIGURE 25 Velocity magnitude $||u||$ and total turbulent kinetic energy k sampled at distances $||d|| = D, 2D, 3D$ behind turbine 1 (A, C) and turbine 4 (B, D) for the strongly stable atmospheric boundary layer (ABL) case. The quantities are averaged over the last 220 s of physical time. The sampling planes are 300m $\times$ 150m centered at the turbine hubs.

upstream turbines is lower compared with the neutral case. This is due to the average velocity impacting the cross-section swept by the turbine blades, which is lower in the strongly stable case due to steeper gradients resulting from low-level jets as shown in Figure 22.

The strongly stable ABL case also exhibits a large degree of wind veer which skews the wake shapes and profiles, as seen in Figure 25.

Similar to the neutral case, the wake profile is averaged over the last 220 s of the physical time. The velocity magnitude $||u||$ and total turbulent kinetic energy k are sampled at distances $||d|| = D, 2D, 3D$ behind rotors 1 and 4. As also discussed above, the average velocity in the wake of the turbines is observed to be lower compared to the average velocity reported in Figure 21. Further downstream of the turbines at $||d|| = 2D, 3D$, the total turbulent kinetic energy increases denoting more turbulent mixing resulting in wake recovery as observed in the average velocity plots.

6 | CONCLUDING REMARKS AND WORK IN PROGRESS

High-fidelity modeling of wind energy physics has been established as one of the grand challenges in the widespread deployment of wind energy[1,2]. High-fidelity simulations present a promising platform for wind science discovery, as well as improvement of preexisting engineering models either through spot checking or machine-learning-based enhancements.[3] As such, there has been a rapid increase in development of tools that can serve to capture otherwise underresolved phenomena of interest pertaining to wind farms physics, including but not restricted blade–boundary layer dynamics, blade–wake, and blade–atmosphere interactions in stall- and deep-stall-inducing flows, and interactional aerodynamics between turbine components. To this end, this paper serves to present ExaWind, an open-source incompressible-flow hybrid-solver framework designed to perform large-scale geometry-resolved wind farm simulations. The hybrid-solver strategy employed by ExaWind combines Nalu-Wind, an unstructured-grid near-body solver, and AMR-Wind, a block-structured grid off-body solver. The two CFD solvers are coupled through overset grids processed by TIOGA. The majority of the ExaWind framework, including the CFD solvers, is designed to be performance portable, that is, the same source code runs on varying GPU and CPU architectures.

By way of introducing the Exawind hybrid solver, this paper introduced the governing equations solved by the CFD solvers, including the turbulence modeling schemes used in ExaWind. Turbulent simulations make use of RANS models in Nalu-Wind, whereas AMR-Wind can use RANS or LES models depending on the application. Also presented is an overview of the overset algorithm implemented in ExaWind. The multisolver strategy of ExaWind allows each CFD solver to use tailored linear-system solvers to solve the equations on the near- and off-body grids. Linear solvers form an integral piece of the ExaWind solver which relies on the Trilinos,[54] *hypre*,[55] and AMReX[77] libraries for solving the massively large sparse systems encountered in wind farm simulations. Thus, the linear solver strategy tailored to the near- and off-body grids is discussed for effective convergence of the highly nonlinear flow field associated with wind farm physics.

The numerical examples discussed in this study are aimed at formally establishing the accuracy of the ExaWind solver and at demonstrating the validity of the ExaWind solver for turbulent flow around the turbine geometries. To this end, we presented a formal order-of-accuracy study using the convecting Taylor vortex problem which has an analytical solution. The ExaWind solver is shown to be first-order accurate in time and second-order accurate in space. The first-order accuracy in time is hypothesized, in part, based on the authors' previous work on overset meshes for incompressible flows,[71] to be a combination of lack of an outer coupling loop which would enable multiple solution exchanges between Nalu-Wind and AMR-Wind and the staggered-in-time arrangement of flow variables across AMR-Wind and Nalu-Wind. Development of algorithms to enable an outer coupling loop is presently underway. Another canonical numerical example presented in this paper aimed at establishing the accuracy of the ExaWind solver is that of flow past a sphere. For the first time to the authors' knowledge, an overset-grid approach was used to analyze flow past a sphere for a varying range of Reynolds numbers ranging from 100 to 1000. Excellent agreement was reported with previously published results in the literature. Validation studies for the ExaWind solver for turbulent flow focused on flow past different megawatt-scale rotor designs corresponding to the NREL 5-MW and NM80 2-MW turbines. For flow past the NREL 5-MW rotor, excellent agreement against other simulation data was documented for both power and thrust. For flow past the NM80 2-MW rotor, results were compared against field data as well as previously published state-of-art simulation data. Excellent agreement was shown against the simulation data, while some discrepancies were observed in comparison with the experimental data, which were largely attributed to the low quality of field data available and also a host of modeling assumptions employed in the current work. Lastly, demonstration studies involving four 5-MW turbines in ABLs with different stability states were presented to illustrate the viability of the ExaWind solver for geometry-resolved wind farm simulations.

Future work is largely geared toward introducing more capabilities for improving the capture of relevant physics integral to wind farms, some of which are imminent. Of particular note are turbulence models capable of predicting stall and post-stall aerodynamic behavior, fluid–structure interaction to study vibrations and subsequent impact on aerodynamic quantities, complex terrain models for accurate modeling of wake propagation in geographical regions with varying topography, two-phase flow for modeling breaking waves and air–sea interaction crucial for modeling offshore wind farms, and mesoscale to microscale coupling for more realistic characterization of the turbulent flow into the wind farms as informed by numerical weather prediction models.

While the ExaWind software is under steady development, strides in high-fidelity modeling for wind farm physics require close collaboration with different sections of the wind energy community. ExaWind's open-source software choice stems from the need to promote transparency and community involvement across academia, research labs, and the wind industry, with ambitions of also targeting extrinsic topics such as enhanced farm inflow characterization, next-generation engineering models, and identifying gaps in wind energy standards and certification.

ACKNOWLEDGEMENTS

The authors thank Dr. Jayanarayanan Sitaraman, the lead developer of the overset assembly tool TIOGA, for their valuable insights and immense support in software development efforts toward expanding TIOGA's capabilities to accommodate the hybrid-CFD solver framework presented in this paper. This research was supported by the Exascale Computing Project (17-SC-20-SC), a joint project of the US Department of Energy (DOE) Office of Science, and the National Nuclear Security Administration (NNSA), responsible for delivering a capable exascale ecosystem, including software, applications, and hardware technology, to support the nation's exascale computing imperative. Funding was also provided by the US Department of Energy, Office of Energy Efficiency and Renewable Energy, and Wind Energy Technologies Office. This work was authored in part by the National Renewable Energy Laboratory, operated by the Alliance for Sustainable Energy, LLC, for the U.S. DOE under contract no. DE-AC36-08GO28308. The work is also authored in part by Sandia National Laboratories, a multimission laboratory managed and operated by the National Technology and Engineering Solutions of Sandia, LLC (NTESS), a wholly owned subsidiary of Honeywell International Inc., for the US DOE's NNSA under contract no. DE-NA0003525. 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. This research also used resources of the Oak Ridge Leadership Computing Facility at the Oak Ridge National Laboratory, which is supported by the Office of Science of the US Department of Energy under contract no. DE-AC05-00OR22725. The views expressed in the article do not necessarily represent the views of the DOE or the US Government. The US Government retains and the publisher, by accepting the article for publication, acknowledges that the US 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 US Government purposes.

DATA AVAILABILITY STATEMENT

The ExaWind software is open source and on GitHub (https://github.com/Exawind). The visualization data that support the findings of this study are too large to store for an extended period; however, the problem setups and the output files will be made available from the authors upon reasonable request.

REFERENCES

1. Veers P, Bottasso C, Manuel L, et al. Grand challenges in the design, manufacture, and operation of future wind turbine systems. *Wind Energy Sci Discuss*. 2022;2022:1-102.
2. Veers P, Dykes K, Lantz E, et al. Grand challenges in the science of wind energy. *Science*. 2019;366(6464):eaau2027.
3. Pawar S, Sharma A, Vijayakumar G, Bay CJ, Yellapantula S, San O. Towards multi-fidelity deep learning of wind turbine wakes. *Renew Energy*. 2022; 200:867-879.
4. Sørensen JN, Myken A. Unsteady actuator disc model for horizontal axis wind turbines. *J Wind Eng Ind Aerodyn*. 1992;39(1-3):139-149.
5. Mikkelsen R, et al. Actuator disc methods applied to wind turbines. *Ph.D. Thesis*. Lyngby, Denmark; 2003.
6. Calaf M, Meneveau C, Meyers J. Large eddy simulation study of fully developed wind-turbine array boundary layers. *Phys Fluids*. 2010;22(1):15110.
7. Sorensen JN, Shen WZ. Numerical modeling of wind turbine wakes. *J Fluids Eng*. 2002;124(2):393-399.
8. Troldborg N. Actuator line modeling of wind turbine wakes. *Ph.D. Thesis*. Lyngby, Denmark; 2009.
9. Churchfield M, Lee S, Moriarty P, Martinez L, Leonardi S, Vijayakumar G, Brasseur J. A large-eddy simulation of wind-plant aerodynamics. In: 50th AIAA Aerospace Sciences Meeting Including the New Horizons Forum and Aerospace Exposition; 2012; Nashville, Tennessee:537.
10. Boorsma K, Schepers G, Aagard Madsen H, et al. Progress in the validation of rotor aerodynamic codes using field data. *Wind Energy Sci*. 2023;8(2): 211-230. https://wes.copernicus.org/articles/8/211/2023/
11. Hodgson EL, Grinderslev C, Meyer Forsting AR, Troldborg N, Srensen NN, Srensen JN, Andersen SJ. Validation of aeroelastic actuator line for wind turbine modelling in complex flows. *Front Energy Res*. 2022;10. https://www.frontiersin.org/articles/10.3389/fenrg.2022.864645
12. Bangga G, Lutz T, Jost E, Krmer E. CFD studies on rotational augmentation at the inboard sections of a 10 MW wind turbine rotor. *J Renew Sustain Energy*. 2017;9(2):23304. https://doi.org/10.1063/1.4978681
13. Bangga G. Consistency between engineering models and high order methods. *Wind Turbine Aerodynamics Modeling Using CFD Approaches*: AIP Publishing LLC; 2022. https://doi.org/10.1063/9780735424111_008
14. Hammond SW, Sprague MA, Womble D, Barone M. A2e high fidelity modeling: strategic planning meetings. NREL/TP-2C00-64697, Golden, CO; 2015. https://www.nrel.gov/docs/fy16osti/64697.pdf
15. Robinson MC, Sprague MA. Looking forward: the promise and challenge of exascale computing. In: Veers P, ed. *Wind Energy Modeling and Simulation - Volume 1: Atmosphere and Plant*; 2019:1-22.
16. Vijayakumar G, Brasseur JG. Blade-resolved modeling with fluid structure interaction. In: Veers P, ed. *Wind Energy Modeling and Simulation - Volume 1: Atmosphere and Plant*; 2019:23-64.
17. Van Der Auweraert JF. Modeling of wind turbine wake with a sliding mesh. *Ph.D. Thesis*. Delft, Netherlands; 2016.
18. Lopez Mejia OD, Mejia OE, Escorcia KM, Suarez F, Laín S. Comparison of sliding and overset mesh techniques in the simulation of a vertical axis turbine for hydrokinetic applications. *Processes*. 2021;9(11):1933.
19. Ma L, Delafin P-L, Tsoutsanis P, Antoniadis A, Nishino T. Blade-resolved CFD simulations of a periodic array of NREL 5 mw rotors with and without towers. *Wind*. 2022;2(1):51-67.
20. Zahle F, Sørensen NN, Johansen J. Wind turbine rotor-tower interaction using an incompressible overset grid method. *Wind Energy: An Int J Progress Appl Wind Power Convers Technol*. 2009;12(6):594-619.
21. Sitaraman J, Mavriplis DJ, Duque EP. Wind farm simulations using a full rotor model for wind turbines. In: 32nd ASME Wind Energy Symposium ASME; 2014; National Harbor, Maryland:1086.
22. Kirby AC, Brazell M, Yang Z, Roy R, Reza Ahrabi B, Mavriplis D, Sitaraman J, Stoellinger MK. Wind farm simulations using an overset HP-adaptive approach with blade-resolved turbine models. In: 23rd AIAA Computational Fluid Dynamics Conference AIAA; 2017:3958.
23. Sitaraman J, Floros M, Wissink A, Potsdam M. Parallel domain connectivity algorithm for unsteady flow computations using overlapping and adaptive grids. *J Comput Phys*. 2010;229(12):4703-4723.
24. Chandar DDJ, Sitaraman J, Mavriplis DJ. A GPU-based incompressible Navier–Stokes solver on moving overset grids. *Int J Comput Fluid Dyn*. 2013; 27(6-7):268-282.
25. Crabill JA, Sitaraman J, Jameson A. A high-order overset method on moving and deforming grids. In: Aiaa Modeling and Simulation Technologies Conference AIAA; 2016:3225.
26. Hand MM, Simms DA, Fingerish LJ, Jager DW, Cotrell JR, Schreck S, Larwood SM. Unsteady aerodynamics experiment phase VI: wind tunnel test configurations and available data campaigns. NREL/TP-500-29955, Golden, Colorado, National Renewable Energy Laboratory; 2001.
27. Gundling C, Sitaraman J, Roget B, Masarati P. Application and validation of incrementally complex models for wind turbine aerodynamics, isolated wind turbine in uniform inflow conditions. *Wind Energy*. 2015;18(11):1893-1916.
28. Duque EarlPN, Burklund MD, Johnson W. Navier–Stokes and comprehensive analysis performance predictions of the NREL phase VI experiment. *J Sol Energy Eng*. 2003;125(4):457-467.
29. Potsdam M, Mavriplis D. Unstructured mesh CFD aerodynamic analysis of the NREL phase VI rotor. In: 47th AIAA Aerospace Sciences Meeting Including the New Horizons Forum and Aerospace Exposition AIAA; 2009:1221.
30. Zahle F, Johansen J, Sørensen N. Wind turbine aerodynamics using an incompressible overset grid method. *Ph.D. Thesis*. London, UK; 2007.
31. Li Y, Paik K-J, Xing T, Carrica PM. Dynamic overset CFD simulations of wind turbine aerodynamics. *Renew Energy*. 2012;37(1):285-298.
32. Zhang Y, Kim B. A fully coupled computational fluid dynamics method for analysis of semi-submersible floating offshore wind turbines under wind-wave excitation conditions based on OC5 data. *Appl Sci*. 2018;8(11):2314.
33. Grinderslev C, Vijayakumar G, Ananthan S, Sørensen NN, Zahle F, Sprague MA. Validation of blade-resolved computational fluid dynamics for a MW-scale turbine rotor in atmospheric flow. In: Journal of Physics: Conference Series, Vol. 1618 TORQUE; 2020:52049.

34. Grinderslev C, González Horcas S, Sørensen NN. Fluid–structure interaction simulations of a wind turbine rotor in complex flows, validated through field experiments. *Wind Energy*. 2021;24(12):1426-1442.
35. Brazell MJ, Sitaraman J, Mavriplis DJ. An overset mesh approach for 3D mixed element high-order discretizations. *J Comput Phys*. 2016;322:33-51.
36. Kirby A, Brazell M, Sitaraman J, Mavriplis D. An overset adaptive high-order approach for blade-resolved wind energy applications. In: Ahs 72nd Annual Forum American Helicopter Society International, Inc.; 2016.
37. Kirby AC, Brazell MJ, Yang Z, Roy R, Ahrabi BR, Stoellinger MK, Sitaraman J, Mavriplis DJ. Wind farm simulations using an overset HP-adaptive approach with blade-resolved turbine models. *The Int J High Perform Comput Appl*. 2019;33(5):897-923.
38. Mavriplis D. Grid resolution study of a drag prediction workshop configuration using the NSU3D unstructured mesh solver. In: 23rd AIAA Applied Aerodynamics Conference AIAA; 2005:4729.
39. Burstedde C, Wilcox LC, Ghattas O. P4est: scalable algorithms for parallel adaptive mesh refinement on forests of octrees. *SIAM J Sci Comput*. 2011; 33(3):1103-1133.
40. Gopalan H, Gundling C, Brown K, Roget B, Sitaraman J, Mirocha JD, Miller WO. A coupled mesoscale–microscale framework for wind resource estimation and farm aerodynamics. *J Wind Eng Ind Aerodyn*. 2014;132:13-26.
41. Li Y, Castro AM, Sinokrot T, Prescott W, Carrica PM. Coupled multi-body dynamics and CFD for wind turbine simulation including explicit wind turbulence. *Renew Energy*. 2015;76:338-361.
42. Li Y, Castro AM, Martin JE, Sinokrot T, Prescott W, Carrica PM. Coupled computational fluid dynamics/multibody dynamics method for wind turbine aero-servo-elastic simulation including drivetrain dynamics. *Renew Energy*. 2017;101:1037-1051.
43. Zhang W, Almgren A, Beckner V, et al. AMReX: a framework for block-structured adaptive mesh refinement. *J Open Source Softw*. 2019;4(37):1370. https://doi.org/10.21105/joss.01370
44. Jonkman J. The new modularization framework for the fast wind turbine CAE tool. In: 51st AIAA Aerospace Sciences Meeting Including the New Horizons Forum and Aerospace Exposition; 2013; Grapevine, Texas.
45. Wang Q, Sprague MA, Jonkman J, Johnson N, Jonkman B. BeamDyn: a high-fidelity wind turbine blade solver in the fast modular framework. *Wind Energy*. 2017;20:1439-1462.
46. Sprague MA, Ananthan S, Vijayakumar G, Robinson MC. ExaWind: A multi-fidelity modeling and simulation environment for wind energy. In: J. Phys. Conf. Series NAWEA; 2020.
47. Sprague MA, Boldyrev S, Fischer P, Grout R, Gustafson Jr WI, Moser R. Turbulent flow simulation at the exascale: opportunities and challenges workshop: August 4-5, 2015, washington, dc; 2017.
48. Domino S. Sierra low mach module: NALU theory manual 1.0. SAND2015-3107W, Sandia National Laboratories Unclassified Unlimited Release (UUR) 30; 2015.
49. Diskin B, Thomas JL, Nielsen EJ, Nishikawa H, White JA. Comparison of node-centered and cell-centered unstructured finite-volume discretizations: viscous fluxes. *AIAA J*. 2010;48(7):1326-1338.
50. Domino S. Toward verification of formal time accuracy for a family of approximate projection methods using the method of manufactured solutions. In: Proceedings of the 2006 Summer Program. Center for Turbulence Research; 2006:163-177.
51. Thomas SJ, Ananthan S, Yellapantula S, Hu JJ, Lawson M, Sprague MA. A comparison of classical and aggregation-based algebraic multigrid preconditioners for high-fidelity simulation of wind turbine incompressible flows. *SIAM J Sci Comput*. 2019;41(5):S196-S219.
52. Chang W, Giraldo F, Perot B. Analysis of an exact fractional step method. *J Comput Phys*. 2002;180(1):183-199.
53. Perot JB. An analysis of the fractional step method. *J Comput Phys*. 1993;108(1):51-58.
54. Heroux MA, Bartlett RA, Howle VE, et al. An overview of the trilinos project. *ACM Trans Math Softw (TOMS)*. 2005;31(3):397-423.
55. Falgout RD, Yang UM. hypre: a library of high performance preconditioners. In: International Conference on Computational Science Springer; 2002: 632-641.
56. Edwards HC, Trott CR, Sunderland D. Kokkos: enabling many core performance portability through polymorphic memory access patterns. *J Parallel Distributed Comput*. 2014;74(12):3202-3216.
57. Mullowney P, Li R, Thomas S, Ananthan S, Sharma A, Williams A, Rood J, Sprague MA. Preparing an incompressible-flow fluid dynamics code for exascale-class wind energy simulations. In: Proceedings of the ACM/IEEE Supercomputing 2021 Conference (Accepted). ACM; 2021.
58. Zhang W, Myers A, Gott K, Almgren A, Bell J. AMReX: block-structured adaptive mesh refinement for multiphysics applications. *The Int J High Perform Comput Appl*. 2021;35(6):508-526.
59. Almgren AS, Bell JB, Szymczak WG. A numerical method for the incompressible Navier–Stokes equations based on an approximate projection. *SIAM J Sci Comput*. 1996;17(2):358-369.
60. Almgren AS, Bell JB, Colella P, Howell LH, Welcome ML. A conservative adaptive projection method for the variable density incompressible Navier–Stokes equations. *J Comput Phys*. 1998;142(1):1-46.
61. Van Leer B. Towards the ultimate conservative difference scheme. IV. A new approach to numerical convection. *J Comput Phys*. 1977;23(3):276-299. https://www.sciencedirect.com/science/article/pii/002199917790095X
62. Colella P, Woodward PR. The piecewise parabolic method (PPM) for gas-dynamical simulations. *J Comput Phys*. 1984;54(1):174-201.
63. Borges R, Carmona M, Costa B, Don WS. An improved weighted essentially non-oscillatory scheme for hyperbolic conservation laws. *J Comput Phys*. 2008;227(6):3191-3211.
64. Jiang G-S, Shu C-W. Efficient implementation of weighted ENO schemes. *J Comput Phys*. 1996;126(1):202-228.
65. Almgren AS, Bell JB, Crutchfield WY. Approximate projection methods: Part I. Inviscid analysis. *SIAM J Sci Comput*. 2000;22(4):1139-1159. https://doi.org/10.1137/S1064827599357024
66. Moeng C-H. A large-eddy-simulation model for the study of planetary boundary-layer turbulence. *J Atmosph Sci*. 1984;41(13):2052-2062.
67. Etling D. Modelling the vertical ABL structure. In: Lalas DP, Ratto CF, eds. *Modelling of Atmospheric Flow Fields*; 1996:45-86.
68. Gritskevich MS, Garbaruk AV, Schütze J, Menter FR. Development of DDES and IDDES formulations for the k-ω shear stress transport model. *Flow, Turbulence Combust*. 2012;88(3):431-449. https://doi.org/10.1007/s10494-011-9378-4
69. Bidadi S, Vijayakumar G, Sharma A, Sprague MA. Mesh and model requirements for capturing deep-stall aerodynamics in low-mach-number flows. *J Turbulence*. 2023;2023:1-26.
70. Roget B, Sitaraman J. Robust and efficient overset grid assembly for partitioned unstructured meshes. *J Comput Phys*. 2014;260:1-24.

71. Sharma A, Ananthan S, Sitaraman J, Thomas S, Sprague MA. Overset meshes for incompressible flows: on preserving accuracy of underlying discretizations. *J Comput Phys*. 2021;428:109987.
72. Henshaw WD. A fourth-order accurate method for the incompressible Navier–Stokes equations on overlapping grids. *J Comput Phys*. 1994;113(1):13-25.
73. Hadzic H. Development and application of finite volume method for the computation of flows around moving bodies on unstructured, overlapping grids. *Ph.D. Thesis*. Hamburg, Germany; 2006.
74. Jude D, Sitaraman J, Lakshminarayan V, Baeder J. An overset generalised minimal residual method for the multi-solver paradigm. *Int J Comput Fluid Dyn*. 2020;34(1):61-74.
75. Elman HC, Howle VE, Shadid JN, Tuminaro RS. A parallel block multi-level preconditioner for the 3D incompressible Navier–Stokes equations. *J Comput Phys*. 2003;187(2):504-523.
76. Tang HS, Jones SC, Sotiropoulos F. An overset-grid method for 3D unsteady incompressible flows. *J Comput Phys*. 2003;191(2):567-600.
77. Zhang W, Almgren A, Beckner V, et al. AMReX: a framework for block-structured adaptive mesh refinement. *J Open Source Softw*. 2019;4(37):1370. https://doi.org/10.21105/joss.01370
78. Bidadi S, Brazell M, Brunhart-Lupo N, et al. Demonstrate multi-turbine simulation with hybrid-structured / unstructured-moving-grid software stack running primarily on GPUS and propose improvements for successful KPP-2, Albuquerque, NM, Sandia National Laboratories; 2019. https://www.osti.gov/biblio/1891592
79. Jonkman J, Butterfield S, Musial W, Scott G. Definition of a 5-MW reference wind turbine for offshore system development. NREL/TP-500-38060, Golden, Colorado, National Renewable Energy Laboratory; 2009.
80. Madsen H, Fuglsang P, Romblad J, et al. The DAN-AERO MW experiments. In: 48th AIAA Aerospace Sciences Meeting Including the New Horizons Forum and Aerospace Exposition; 2010:645.
81. Goin KL, Lawrence WR. Subsonic drag of spheres at Reynolds numbers from 200 to 10,000. *AIAA J*. 1968;6(5):961-962.
82. Roos FW, Willmarth WW. Some experimental results on sphere and disk drag. *AIAA J*. 1971;9(2):285-291.
83. Johnson TA, Patel VC. Flow past a sphere up to a Reynolds number of 300. *J Fluid Mech*. 1999;378:19-70.
84. Tomboulides AG, Orszag SA. Numerical investigation of transitional and weak turbulent flow past a sphere. *J Fluid Mech*. 2000;416:45-73.
85. Nagata T, Nonomura T, Takahashi S, Fukuda K. Direct numerical simulation of subsonic, transonic and supersonic flow over an isolated sphere up to a Reynolds number of 1000. *j Fluid Mech*. 2020;904.
86. Sørensen NN, Johansen J. Upwind, aerodynamics and aero-elasticity, rotor aerodynamics in atmospheric shear flow. In: 2007 European Wind Energy Conference and Exhibition; 2008; Milan, Italy.
87. Chow R, Van Dam CP. Verification of computational simulations of the NREL 5 mw rotor with a focus on inboard flow separation. *Wind Energy*. 2012;15(8):967-981.
88. Jayaraman B, Brasseur JG. The surprising transition in atmospheric boundary layer turbulence structure from neutral to moderately convective stability states and mechanisms underlying large-scale rolls. arXiv preprint arXiv:180703336; 2018.
89. Cheung L, Blaylock ML, Brown K, deVelder N, Herges TG, Hsieh A, Maniaci DC, Cutler J. Comparison of simulated and measured wake behavior in stable and neutral atmospheric conditions. In: AIAA SCITECH 2022 Forum; 2022:1923.

How to cite this article: Sharma A, Brazell MJ, Vijayakumar G, et al. ExaWind: Open-source CFD for hybrid-RANS/LES geometry-resolved wind turbine simulations in atmospheric flows. *Wind Energy*. 2024;27(3):225-257. doi:10.1002/we.2886