Eleventh International Conference on Computational Fluid Dynamics (ICCFD11), Maui, Hi, USA, July 11-15, 2022

ICCFD11-2022-3403

An open-source incompressible-flow hybrid-solver framework for massively parallel blade-resolved wind farm simulations under atmospheric inflow

Ashesh Sharma*, Michael J Brazell*, Ganesh Vijayakumar*, Shreyas Ananthan**, Lawrence Cheung***, Marc Henry de Frahan*, Paul Mullowney*, Jon Rood*, Philip Sakievich***, and Michael A Sprague*
Corresponding authors: ashesh.sharma@nrel.gov, michael.brazell@nrel.gov

* National Renewable Energy Laboratory.
** Siemens Gamesa Renewable Energy - Digital Ventures Lab.
*** Sandia National Laboratories.

Abstract: Current open-source methods for modeling wind farm largely rely on reduced-order models using blade element momentum theory-based models which use linear superposition of sectional responses using look-up tables to model the 3D aerodynamic response of rotors, thereby restricting the operational regime in which these models can be deemed reliable. These shortcomings are alleviated by blade-resolved simulations. Given the complexity associated with developing a blade-resolved framework, only a handful of such frameworks exist in the world, none of which are open-source, thereby restricting their access within the wind energy community. Additionally the existing tools are not entirely performance portable, i.e. the same code cannot run on varying GPU and CPU architectures. Through Exawind, we have created an open-source performance-portable framework for a community-driven platform to tackle the primary challenges of blade-resolved simulations of wind turbines in atmospheric conditions. In order to achieve this, we have developed an open-source framework, Exawind, comprising of the unstructured incompressible-flow solver Nalu-Wind[1] and the block-structured adaptive mesh refinement background solver AMR-Wind[2] coupled using the overset connectivity tool, Topology Independent Overset Grid Assembler (TIOGA)[3]. Nalu-Wind is Exawind's near-body solver and models the turbulent flow around body-fitted unstructured meshes, while AMR-Wind, constructed on top of AMReX, uses block structured meshes to model the flow in the atmospheric boundary layer (ABL). Coupled through overset, the two computational fluid dynamics (CFD) solvers allow us to simulate flows across a range of length scales that are up to ten orders of magnitude apart. In this paper we provide an overview of the Exawind solver for high-fidelity large-scale wind farm simulations. We present preliminary results using Exawind's hybrid-solver strategy with application to uniform and atmospheric flow past a single and two-turbine setup. In addition, we present some scaling studies performed on U.S. DOE leadership-class systems for simulations representative of wind farm physics.

Keywords: Computational Fluid Dynamics, Incompressible Flow, Overset, Adaptive Mesh Refinement, Wind Energy, Open-source.

[1]https://github.com/Exawind/nalu-wind.git
[2]https://github.com/Exawind/amr-wind.git
[3]https://github.com/jsitaraman/tioga.git

1 Introduction

The U.S. Department of Energy recently compiled a report outlining the wind energy needs of the country [1]. In order to meet almost 35% of nation's energy demands via wind energy, some of the grand challenges in wind energy science modeling [2, 3] need to be addressed by developing tools capable of capturing otherwise under-resolved phenomena representative of wind farm physics such as fluid-structure interaction modeling blade-boundary-layer and blade-wake dynamics, deep array effects, complex-terrain impacts, and wake-atmosphere interaction [4]. With exascale computing on the horizon we have an attractive platform to overcoming the modeling barriers associated with geometry-resolved high-fidelity modeling (HFM) of wind farm physics.

Geometry-resolved CFD solvers where the full turbine geometry is represented by a body-fitted mesh are rare, primarily due to the vast temporal and spatial length scales associated with the problem. From an engineering standpoint, the foremost challenge is to balance accuracy needs with tractable computational cost. The primary approaches to achieving geometry-resolved wind farm simulations use either sliding meshes [5, 6, 7] or overset meshes [8, 9, 10] to resolve flow past the turbine geometry. Sliding meshes cannot model out-of-plane turbine-body displacements, and hence are unable to capture pitch or yaw. Overset meshes on the other hand use overlapping grids to capture the relative motion without any restriction on the movement of the grids themselves. Consequently, CFD solvers based on overset mesh frameworks have represented the state-of-art in geometry-resolved wind physics modeling, and it is also the strategy we adopt in this paper. Apart from the commercial code Simcenter STAR-CCM+, only a couple of research groups develop overset capabilities capable of resolving atmospheric flow in wind farms, largely due to the complexity associated with building such a framework. The first CFD code to pioneer use of overset grids for wind turbine simulations was EllipSys3D, a collaboration between Technical University of Denmark and the Risø National Laboratory, Denmark. The code is MPI-enabled, but lacks support for hybrid CPU/GPU architectures. For more details, the reader is referred to [8]. The only other research group that actively innovate in overset technology for wind energy applications are the developers of W^2A^2KE3D [10] at the University of Wyoming. Like EllipSys3D, W^2A^2KE3D too is designed to run on massive number of MPI ranks but lacks support for GPUs.

Motivated by creating a framework capable of predicting the highest fidelity of wind farm physics possible with accommodations for future exascale systems, and in the spirit of creating a community-driven platform, the U.S. Department of Energy has invested in the development of the open-source HFM framework, Exawind[1]. The Exawind solver utilizes a multi-solver strategy comprising of Nalu-Wind[2], a fully implicit node-centered finite volume unstructured-grid near-body solver, and AMR-Wind[3], an explicit semi-staggered finite volume block-structured off-body solver built on top of AMReX [11]. Both CFD solvers model acoustically incompressible flow. In our hybrid-solver setup, 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 meshes handled by the TIOGA[4]. Both, Nalu-Wind and AMR-Wind have been designed to be exascale-enabled. We have achieved this by constructing these codes on top of programming models that support performance portability, i.e., the same codebase runs on varying GPU and CPU architectures. In this paper we present preliminary results from our hybrid-solver strategy. The remainder of the manuscript is organized as follows: Section 2 presents an overview of the CFD solvers; Section 3 presents the overset approach for coupling the near-body and off-body solvers; Section 4 presents the numerical examples to analyze and validate the Exawind solver; Finally, a summary and concluding remarks are presented in Section 5.

2 CFD Solvers

The Exawind solver discretizes acoustically incompressible Navier-Stokes (NS) equations alongside forcing terms such as Coriolis, Geostrophic, buoyancy, body force, etc. Exawind contains a variety of turbulence

[1] https://github.com/Exawind
[2] https://github.com/Exawind/nalu-wind.git
[3] https://github.com/Exawind/amr-wind.git
[4] https://github.com/jsitaraman/tioga.git

models including unsteady Reynolds-averaged Navier-Stokes (RANS), large eddy simulation (LES) and hybrid RANS/LES to allow for simulation of wind turbines under different inflow conditions. Simulations of turbines where the blade boundary layer is resolved is typically handled using unsteady RANS (k-ω-SST model [12]) or hybrid RANS/LES like the IDDES model [13] that builds on the k-ω-SST model [12], e.g. Two choices are available for the sub-filter scale model for LES of ABL in the off-body solver, which are the Smagorinsky model [14] and the one-equation model from Moeng [15].

The near-body solver of the Exawind framework, Nalu-Wind, is an open-source CFD code written in C++. It is a wind-specific version of Nalu [16], which is an LES research code developed at Sandia National Laboratories. To be able to resolve complex geometries encountered in wind farms, Nalu-Wind employs an unstructured-grid finite volume method for spatial discretization, and solves the acoustically incompressible Navier-Stokes equations for which mass continuity is maintained through approximate pressure projection at every time step solving a set number of Picard iterations. For further details, the reader is referred to the derivations in [17, 18]. For unsteady RANS simulations, upwinding of the advection terms is necessary. Hence we use linear upwinding, details of which are available in the Nalu-Wind user manual.[1]. Governing equations in Nalu-Wind are discretized in space using a second-order node-centered finite volume scheme (see e.g. [19] for discretization details). For time integration, an implicit second-order backward difference formula (BDF) scheme is adopted, details of which have been discussed in [20]. Nalu-Wind has been developed with modern software engineering best practices using the Kokkos [21] programming model for a GPU-enabled performance portable code capable of running on all major high-performance computing (HPC) platforms. The various system of equations in Nalu-Wind can make use of Krylov subspace methods one or both of the *hypre* [22] and Trilinos [23] libraries for solving the challenging linear systems. In this paper, we restrict our attention to the use of *hypre* solvers only based on the most efficient and robust solver parameters determined out of experience.

Exawind's background solver, AMR-Wind is a parallel, block-structured adaptive-mesh, incompressible-flow solver specialized for efficiency and scalability. The solver is built on top of the AMReX library which provides the mesh data structures, performance portable parallel algorithms compatible with different GPU architectures, mesh adaptivity, as well as linear solvers which are a combination of geometric and algebraic multigrid solvers [24]. AMR-Wind is designed to perform LES of ABL flows when coupled to Nalu-Wind using an overset methodology (Section 3) for blade-resolved simulations of multiple wind turbines within a wind farm. The spatial discretization is a combination of the finite volume method and the finite element method. For more details the reader is referred to [25]. Velocity, scalar quantities, and gradients of pressure are located at cell centers, whereas pressure is located at nodes. Partial staggering combined with an approximate projection method yields linear systems that are well studied, have small bandwidth stencils and can be efficiently solved with standard techniques such as geometric multigrid. These discretization choices give a well balanced mix of both efficiency and accuracy. In addition to the spatial staggering there is also staggering in time similar to Crank-Nicolson time-stepping.

3 Overset approach

An overset mesh methodology is used to connect the near-body solver, Nalu-Wind, with the off-body solver, AMR-Wind. The overlapping meshes in the present study are connected using TIOGA which is an open-source automated overset-mesh-assembly library [26, 27]. TIOGA connects domains by identifying three types of nodes (or cells): field, fringe, and hole nodes. This process is known as hole cutting. Field points are regions where the governing equations are solved. Fringe points constitute the region where information is transferred between the overlapping meshes. Hole points represent the mesh nodes/cells at which solution does not exist, usually corresponding to the presence of a wall overlapping the off-body mesh. An example of a simplified hole cut is presented in Fig. 1 Recall that AMR-Wind is cell-based for velocity and scalar flow variables but node-based for the pressure variable whereas Nalu-Wind is node-based for all flow variables. Trilinear interpolation is used to interpolate variables between the two codes. To interpolate a single point from a cell-based variable the eight neighboring cells surrounding that point are used, and to interpolate a single point from a node-based variable the eight surrounding nodes are used. These interpolation strategies are identical but staggered in their interpretation.

[1] https://nalu-wind.readthedocs.io/en/latest/source/theory/advectionStabilization.html

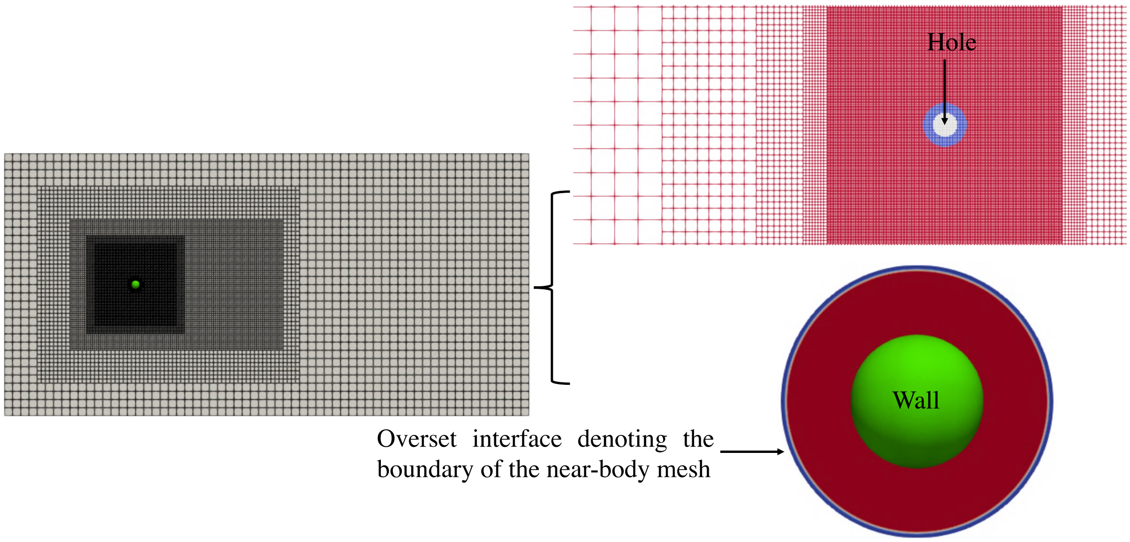

Figure 1: Hole cut associated with a sphere bpdy in a block-structured AMR mesh using TIOGA. Red regions denote field. Blue regions denote fringe, and white regions denote hole.

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 equation (PDE) at the field points for both CFD solvers, followed by update of solution at the fringe points, thus resulting in a lag in the update of solution at the fringe points. The decoupled-linear-system approach solve provides several benefits to solving massively large system of equations [28] including the use of mesh-tailored solvers, and algebraic multigrid preconditioners imperative for rapid convergence of the pressure Poisson problem [29].

Algorithm 1 Exawind driver

1: Create MPI sub-communicators near-body-comm and off-body-comm
2: **if** rank $\subset$ near-body-comm **then**
3:     Initialize near-body solver ▷ Multiple instances of near-body solver allowed
4: **end if**
5: **if** rank $\subset$ off-body-comm **then**
6:     Initialize off-body solver ▷ Only single instance allowed
7: **end if**
8: Initialize TIOGA ▷ For all ranks
9: **for** $t = 0; t < T; t \leftarrow t + 1$ **do** ▷ Where T is the total number of time steps
10:     **if** near-body mesh movement $\|$ off-body mesh adaption $\|$ $t == 0$ **then**
11:         Perform overset connectivity
12:     **end if**
13:     Exchange overset solution
14:     Perform near-body and off-body time step in parallel
15: **end for**

Algorithm 1 describes the AMR-Wind/Nalu-Wind coupling strategy. The solution at fringe points from the initial solution (or previous time step) for every overlapping mesh are exchanged in step 13. The linear solve for each CFD solver is then performed at field points, based on the updated fringe boundary conditions, in step 14 for every overlapping mesh. 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. Note, the lack of an outer Picard-like loop means that the solution at fringe points in the CFD solvers are always lagging behind the solution at field points. While this does not affect the quantities of interest as discussed in Section 4, the temporal order-of-accuracy is reduced [28]. Work is underway towards resolving this by introducing an outer coupling loop for multiple AMR-Wind and Nalu-Wind iterations at any particular time step.

4 Numerical Examples

In this section we discuss preliminary validation studies alongside some demonstration simulations to showcase the current capabilities of the Exawind solver.

4.1 Uniform flow past NREL 5-MW rotor

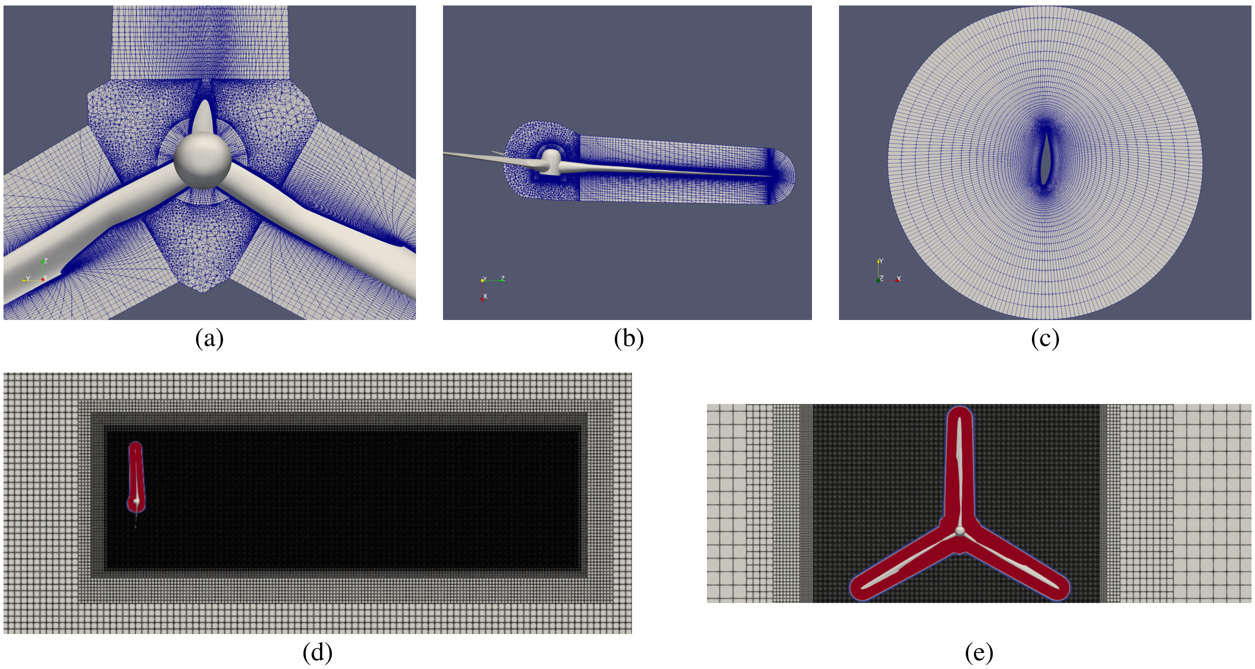

Figure 2: Unstructured near-body mesh used to model the NREL 5-MW rotor with Nalu-Wind including the surface blade and hub mesh: (a) front view; (b) side view; (c) curviliniear mesh around the blades. Overset mesh for NREL 5-MW rotor: (d) side view with refinement level for wake capture; (e) front view.

The NREL 5-MW turbine is a $126m$ diameter reference turbine, designed for offshore wind research [30]. In the current study, the turbine geometry is simplified by ignoring the tower and nacelle structures. Unstructured mesh is used to model the geometry as shown in Fig. 2. There are 8.67 million cells in the near-body mesh. The off-body block-structured Cartesian mesh, also shown in Fig. 2, consists of 5 levels of refinement with the finest level mesh size set to be comparable with that of the coarsest exterior cells in the near-body mesh. All levels of refinement combine for $\approx$ 21000 grid (block-structured) patches for a total of $\approx$ 311 million cells. For simulating the near-body turbine, a $k - \omega$ SST model [31] was used to model the turbulence. We leverage *hypre*'s generalized minimal residual (GMRES) solver alongside its BoomerAMG preconditioner. For the off-body mesh, the turbulence is modeled using the one-equation model for sub-filter scale model for LES [15]. The boundary conditions imposed are slip along all boundaries except for inflow and outflow.

Simulations were performed for the NREL 5-MW rotor operating at uniform inflow of $U_\infty = 8m/s$ in a surrounding fluid with density and viscosity set to that of air. A fixed time-step size was used such that the blade rotates 0.25° each time step. Figure 3 shows the flowfield isocontours of Q-criterion. Figure 4 shows the obtained power and thrust curves compared to results obtained using the single-linear-system implemented in nalu-wind [28]. Comparison against the more strongly coupled approach of Nalu-Wind result in an average difference in power and thrust of 0.16% and 0.5%, respectively. Thus the proposed hybrid-solver strategy is demonstrated to be accurate in capturing engineering quantities of interest for high Reynolds number flows.

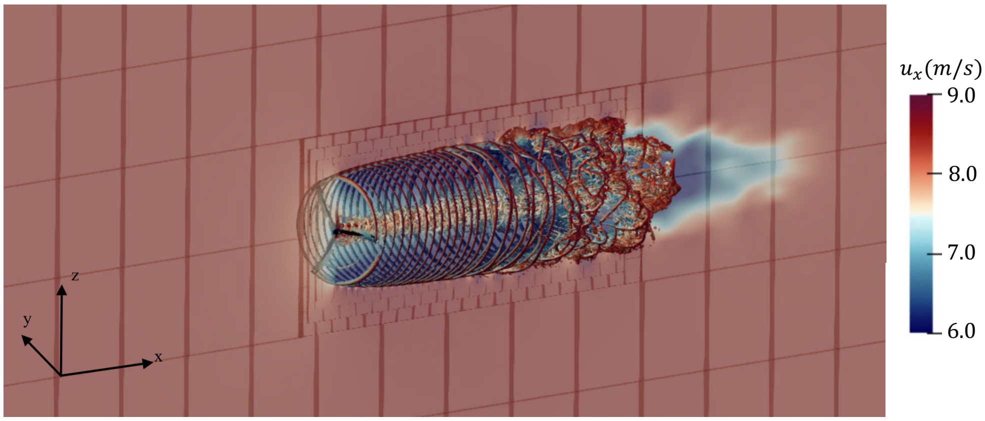

Figure 3: Isocontours of Q-criterion with velocity visualized in the wake for NREL 5-MW rotor operating under uniform inflow wind speed of $8m/s$.

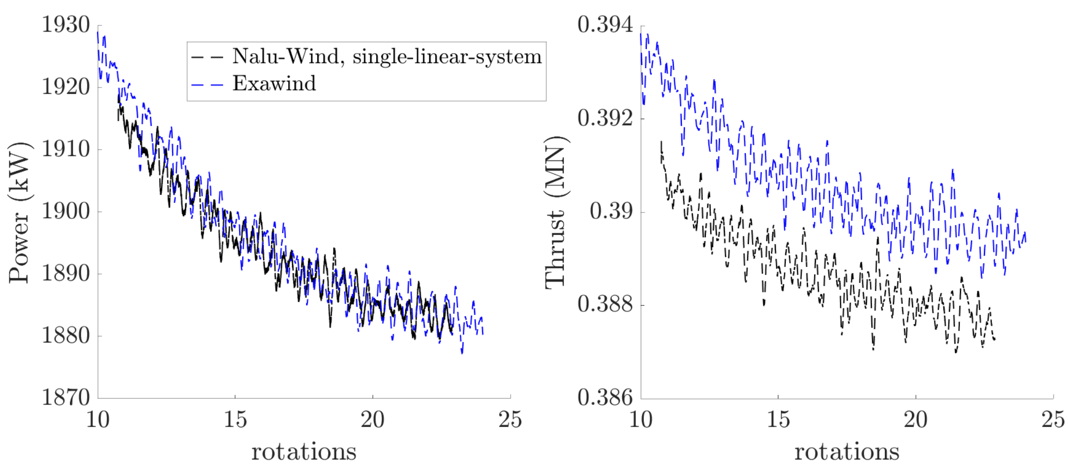

Figure 4: Power and thrust for inflow speed of $U_\infty = 8m/s$ as evaluated using the Exawind solver, and comparison against results from single-linear-system solves implemented Nalu-Wind [28].

4.2 Two-turbine demonstration under atmospheric inflow

In this example we consider a small two-turbine wind farm composed of NREL 5-MW turbines is considered. The near-body mesh described in Section 4.1 is used. A netural ABL is considered in this study with a geostrophic wind (wind resulting from exact balance between the Coriolis force and the pressure gradient force) with velocity of 10 m/s used to drive the flow. The boundary layer develops over the course of 31 hours of physical time and eventually a typical neutral boundary layer forms. The statistics for the fully developed ABL are presented in Fig. 5. A domain of size of $1.875 \times 1.523 \times 0.938$ km^3 is used for a total of $\approx 268M$ cells. Figure 6 depicts the wake of the two turbines, and Fig. 7 presents the corresponding power and thrusts evaluated for each turbine. As expected, the second turbine experiences a noticeable dip in power and thrust because of the slowdown of incoming wind speed caused by the wake of the first turbine. Additionally, the power output from the first turbine occasionally dips as the instantaneous wind changes speed and directions. As follows, it should also be noted that the power and thrust depicted in Fig. 7 are only an approximation given the fact that no controller was employed in the current work, as a result of which both turbines rotated at constant RPM without any change in yaw for the duration of the simulations regardless of the local hub height wind speed and direction which is changing temporally in atmospheric inflow settings.

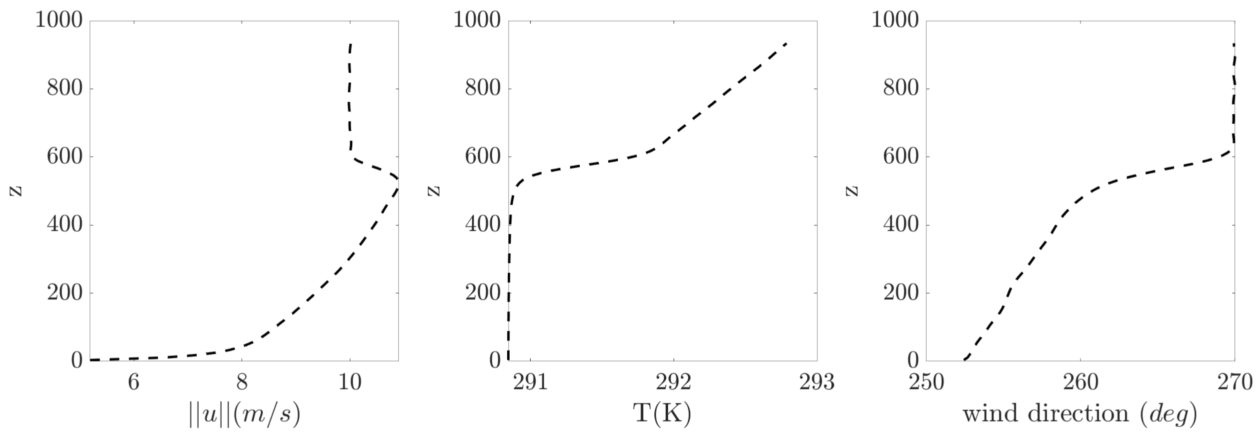

Figure 5: Vertical-distribution statistics for a neutrally stable atmospheric boundary layer.

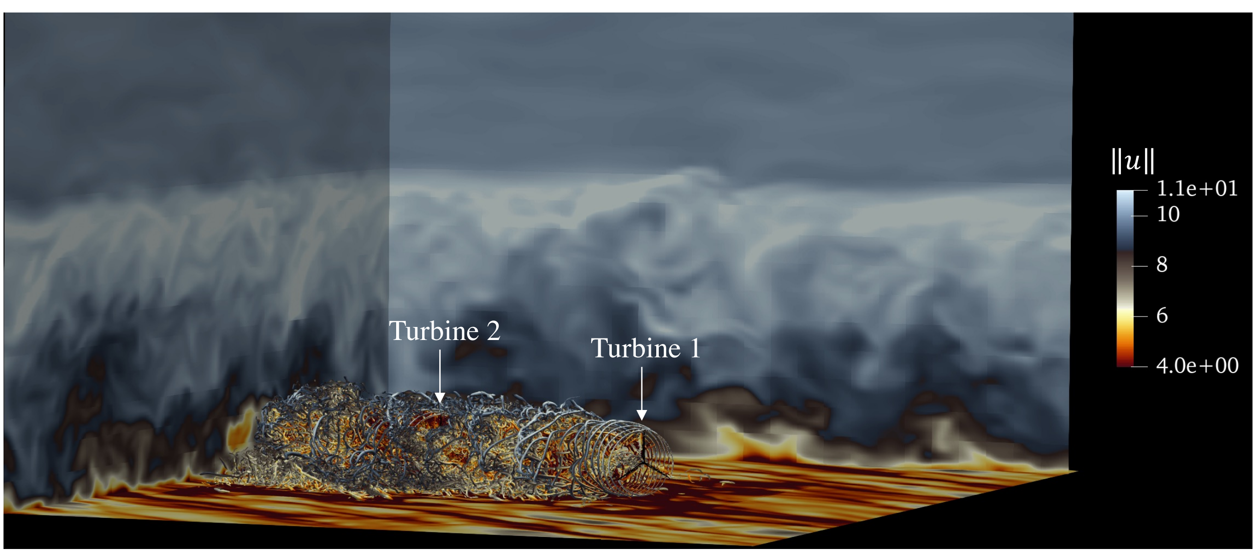

Figure 6: Wake of turbines in a neutral ABL.

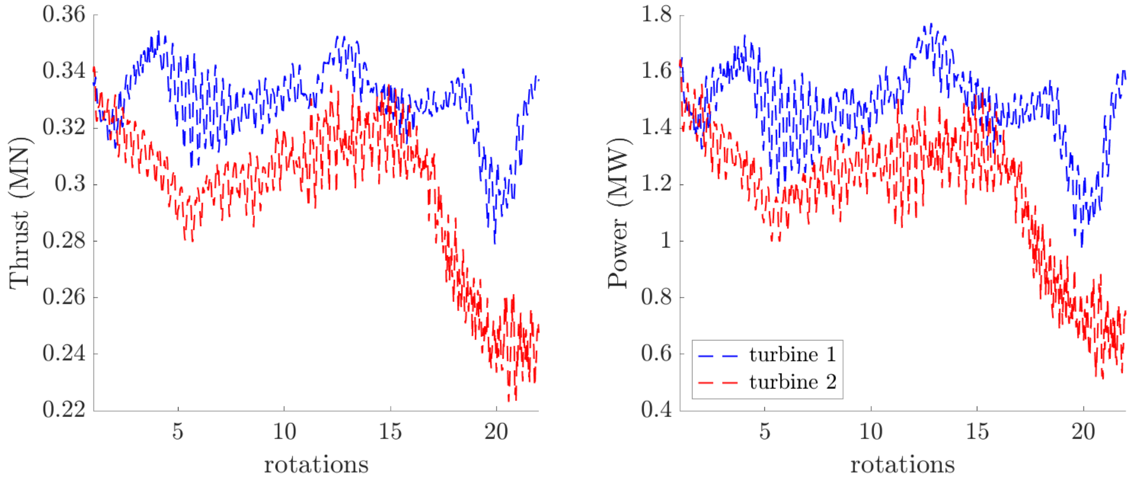

Figure 7: Power and thrust of the turbines depicted in Fig. 6.

4.3 Solver scaling

This section shows the performance results of both Nalu-Wind and AMR-Wind on a similar neutral ABL problem as discussed in Section 4.2. The domain is $5 \times 5 \times 1km^3$, the mesh for Nalu-Wind contains 2.5×10^7 nodes, and AMR-Wind uses a slightly larger mesh with 5.2×10^7 cells. The linear solver relative tolerance is same for both flow solvers and Nalu-Wind is using 2 Picard iterations per time step. Results are shown in Figure 8. We present a typical strong scaling plot, performed on U.S. DOE leadership-class systems Summit and Eagle, where the time per time step is plotted against the number of compute nodes. Note, one Summit node is 42 CPUs and 6 GPUs while one Eagle node is 36 CPUs. As expected, AMR-Wind being a structured CFD solver is approximately 10x faster compared to Nalu-Wind on Summit GPUs. Note, neither of the curves tend to flatten which is why we do not define a hard strong scaling limit for either of the solvers. We present a second plot that shows the number of grid points per second per time step that each flow solver is capable of achieving. This number represents how efficient a flow solver is and a higher number is better and AMR-Wind is capable of processing more than 20x more grid points than Nalu-Wind. Although GPUs do not appear to strong scale as well as as CPUs, GPUs are capable of processing many more grid points per rank per time step than a CPU. This strong scaling study demonstrates that AMR-Wind is more suitable for larger ABL-type background flow simulations compared to Nalu-Wind both in terms of time to solution and efficiency which reinforces our hybrid solver strategy.

5 Summary

In this paper we have presented preliminary results for the Exawind solver, a hybrid-solver framework for large-scale wind farm simulations. In contrast to other high-fidelity modeling tools for wind energy, Exawind is open-source and performance portable encouraging a community-driven platform in wind energy science. Ongoing and future work includes a more formal analysis of the Exawind solver using canonical problems in addition to validation of the framework for various turbines based on results from literature. Additionally, load balancing strategies will be investigated carefully to achieve an efficient balance of CPUs and GPUs to enable hybrid CPU-GPU runs where one of the CFD solver may operate on CPUs while the other operates on GPUs. Finally, work is underway to enable dynamic AMR (Fig. 9) to capture the turbine wakes in wind farms for more cost-effective simulations.

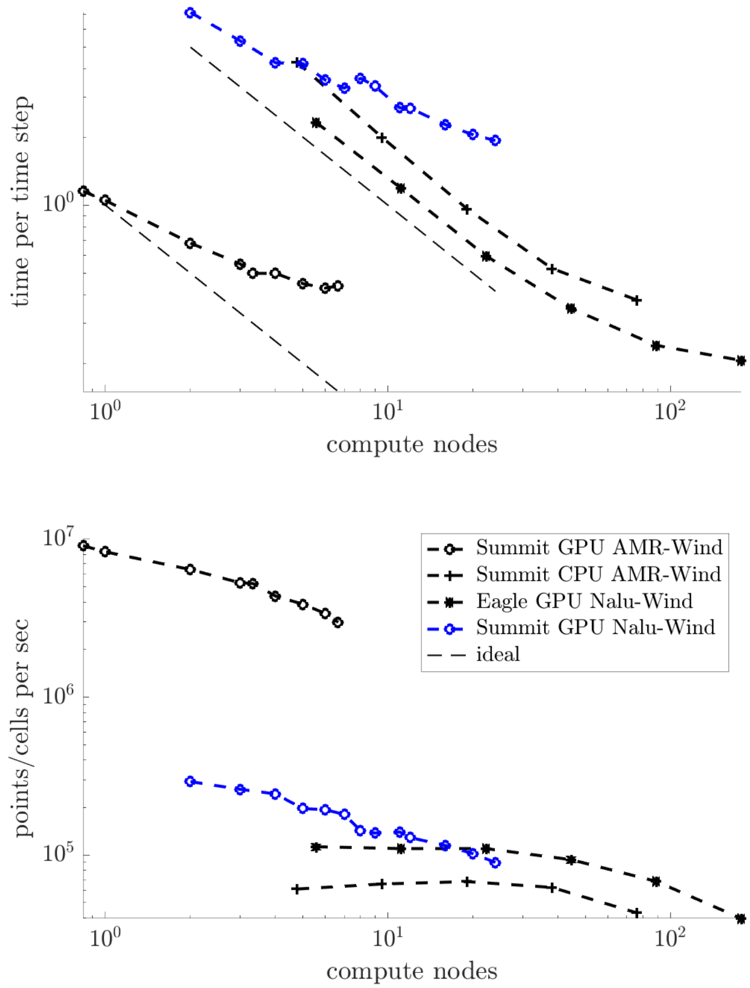

Figure 8: Scaling performance of AMR-Wind and Nalu-Wind on a neutral ABL problem. The mesh for Nalu-Wind contains 2.5×10^7 nodes, and AMR-Wind uses a mesh with 5.2×10^7 cells.

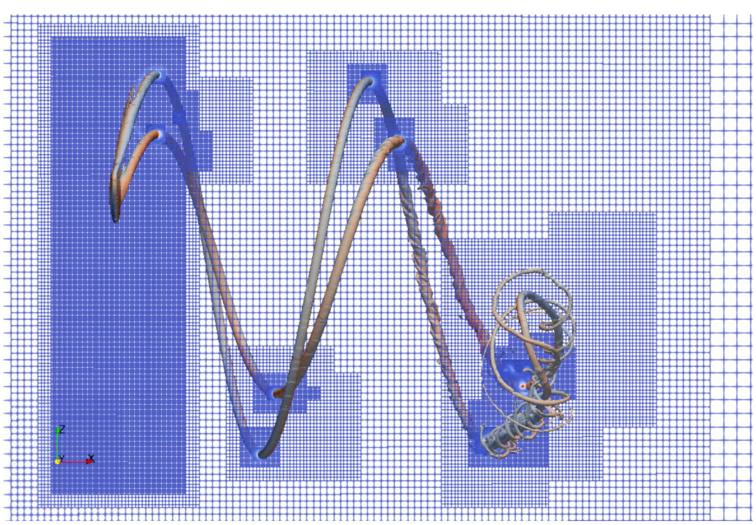

Figure 9: Dynamic AMR using the Exawind solver for a rotating ellipsoid.

Acknowledgements

This research was supported by the Exascale Computing Project (17-SC-20-SC), a joint project of the U.S. Department of Energy Office of Science and the National Nuclear Security Administration, responsible for delivering a capable exascale ecosystem, including software, applications, and hardware technology, to support the nation's exascale computing imperative. This work was authored [in part] by the National Renewable Energy Laboratory, operated by Alliance for Sustainable Energy, LLC, for the U.S. Department of Energy (DOE) under Contract No. DE-AC36-08GO28308. The research was performed using computational resources sponsored by the DOE Office of Energy Efficiency and Renewable Energy and located at the National Renewable Energy Laboratory. The views expressed in the article do not necessarily represent the views of the DOE or the U.S. Government. The U.S. Government retains and the publisher, by accepting the article for publication, acknowledges that the U.S. Government retains a nonexclusive, paid-up, irrevocable, worldwide license to publish or reproduce the published form of this work, or allow others to do so, for U.S. Government purposes. 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 U.S. Department of Energy under Contract No. DE-AC05-00OR22725.

References

[1] Wind vision detailed roadmap actions: 2017 update. March, 2019. doi:10.2172/1497756.

[2] Paul Veers, Katherine Dykes, Eric Lantz, Stephan Barth, Carlo L Bottasso, Ola Carlson, Andrew Clifton, Johney Green, Peter Green, Hannele Holttinen, et al. Grand challenges in the science of wind energy. *Science*, 366(6464):eaau2027, 2019.

[3] Paul Veers, Carlo Bottasso, Lance Manuel, Jonathan Naughton, Lucy Pao, Joshua Paquette, Amy Robertson, Michael Robinson, Shreyas Ananthan, Athanasios Barlas, et al. Grand challenges in the design, manufacture, and operation of future wind turbine systems. *Wind Energy Science Discussions*, pages 1–102, 2022.

[4] Ganesh Vijayakumar and James Brasseur. Blade-resolved modeling with fluid-structure interaction. In *Wind Energy Modelling and Simulation: Atmosphere and Plant*. Institution of Engineering and Technology, 2019.

[5] JF Van Der Auweraert. *Modeling of wind turbine wake with a sliding mesh*. PhD thesis, Delft University of Technology, 2016.

[6] Omar D Lopez Mejia, Oscar E Mejia, Karol M Escorcia, Fabian Suarez, and Santiago Laín. Comparison of sliding and overset mesh techniques in the simulation of a vertical axis turbine for hydrokinetic applications. *Processes*, 9(11):1933, 2021.

[7] Lun Ma, Pierre-Luc Delafin, Panagiotis Tsoutsanis, Antonis Antoniadis, and Takafumi Nishino. Blade-resolved cfd simulations of a periodic array of nrel 5 mw rotors with and without towers. *Wind*, 2(1):51–67, 2022.

[8] Frederik Zahle, Niels N Sørensen, and Jeppe Johansen. Wind turbine rotor-tower interaction using an incompressible overset grid method. *Wind Energy: An International Journal for Progress and Applications in Wind Power Conversion Technology*, 12(6):594–619, 2009.

[9] Jayanarayanan Sitaraman, Dimitri J Mavriplis, and Earl P Duque. Wind farm simulations using a full rotor model for wind turbines. In *32nd ASME Wind Energy Symposium*, page 1086, 2014.

[10] Andrew C Kirby, Michael J Brazell, Zhi Yang, Rajib Roy, Behzad R Ahrabi, Michael K Stoellinger, Jay Sitaraman, and Dimitri J Mavriplis. Wind farm simulations using an overset hp-adaptive approach with blade-resolved turbine models. *The International Journal of High Performance Computing Applications*, 33(5):897–923, 2019.

[11] Weiqun Zhang, Ann Almgren, Vince Beckner, John Bell, Johannes Blaschke, Cy Chan, Marcus Day, Brian Friesen, Kevin Gott, Daniel Graves, Max Katz, Andrew Myers, Tan Nguyen, Andrew Nonaka, Michele Rosso, Samuel Williams, and Michael Zingale. AMReX: a framework for block-structured adaptive mesh refinement. *Journal of Open Source Software*, 4(37):1370, May 2019.

[12] F. Menter, M. Kuntz, and R. Langtry. Ten years of industrial experience with the sst turbulence model. *Turbulence, Heat and Mass Transfer*, 4, 2003.

[13] Mikhail S. Gritskevich, Andrey V. Garbaruk, Jochen Schütze, and Florian R. Menter. Development of

ddes and iddes formulations for the k-ω shear stress transport model. *Flow, Turbulence and Combustion*, 88(3):431–449, 2012.

[14] Stephen B Pope and Stephen B Pope. *Turbulent flows*. Cambridge university press, 2000.

[15] A large-eddy-simulation model for the study of planetary boundary-layer turbulence. *Journal of the Atmospheric Sciences*, 41(13):2052–2062, 1984.

[16] Stefan P Domino. Sierra low mach module: Nalu theory manual 1.0, SAND2015-3107W, Sandia National Laboratories Unclassified Unlimited Release (UUR). 2015.

[17] S. Domino. 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*, pages 163–177. Center for Turbulence Research, 2006.

[18] Stephen J Thomas, Shreyas Ananthan, Shashank Yellapantula, Jonathan J Hu, Michael Lawson, and Mary Ann Sprague. A comparison of classical and aggregation-based algebraic multigrid preconditioners for high-fidelity simulation of wind turbine incompressible flows. *SIAM Journal on Scientific Computing*, 41(5):S196–S219, 2019.

[19] Boris Diskin, James L Thomas, Eric J Nielsen, Hiroaki Nishikawa, and Jeffery A White. Comparison of node-centered and cell-centered unstructured finite-volume discretizations: viscous fluxes. *AIAA journal*, 48(7):1326–1338, 2010.

[20] Shreyas Ananthan, Luigi Capone, Marc Henry de Frahan, Jonathan J. Hu, Jeremy Melvin, James R. Overfelt, Ashesh Sharma, Jay Sitaraman, Katarzyna Swirydowicz, Stephen Thomas, Ganesh Vijayakumar, Alan B. Williams, Shashank Yellapantula, and Michael Sprague. Nalu-Wind and OpenFAST: A high-fidelity modeling and simulation environment for wind energy. Milestone ECP-Q2-FY19. 2019. doi:`10.2172/1762093`.

[21] H Carter Edwards, Christian R Trott, and Daniel Sunderland. Kokkos: Enabling manycore performance portability through polymorphic memory access patterns. *Journal of parallel and distributed computing*, 74(12):3202–3216, 2014.

[22] Robert D Falgout and Ulrike Meier Yang. hypre: A library of high performance preconditioners. In *International Conference on Computational Science*, pages 632–641. Springer, 2002.

[23] Michael A Heroux, Roscoe A Bartlett, Vicki E Howle, Robert J Hoekstra, Jonathan J Hu, Tamara G Kolda, Richard B Lehoucq, Kevin R Long, Roger P Pawlowski, Eric T Phipps, et al. An overview of the trilinos project. *ACM Transactions on Mathematical Software (TOMS)*, 31(3):397–423, 2005.

[24] Weiqun Zhang, Ann Almgren, Vince Beckner, John Bell, Johannes Blaschke, Cy Chan, Marcus Day, Brian Friesen, Kevin Gott, Daniel Graves, Max Katz, Andrew Myers, Tan Nguyen, Andrew Nonaka, Michele Rosso, Samuel Williams, and Michael Zingale. AMReX: a framework for block-structured adaptive mesh refinement. *Journal of Open Source Software*, 4(37):1370, May 2019.

[25] Ann S. Almgren, John B. Bell, Phillip Colella, Louis H. Howell, and Michael L. Welcome. A Conservative Adaptive Projection Method for the Variable Density Incompressible Navier-Stokes Equations. *Journal of Computational Physics*, 142(1):1–46, 1998.

[26] Beatrice Roget and Jayanarayanan Sitaraman. Robust and efficient overset grid assembly for partitioned unstructured meshes. *Journal of Computational Physics*, 260:1–24, 2014.

[27] Michael J Brazell, Jayanarayanan Sitaraman, and Dimitri J Mavriplis. An overset mesh approach for 3D mixed element high-order discretizations. *Journal of Computational Physics*, 322:33–51, 2016.

[28] Ashesh Sharma, Shreyas Ananthan, Jayanarayanan Sitaraman, Stephen Thomas, and Michael A Sprague. Overset meshes for incompressible flows: On preserving accuracy of underlying discretizations. *Journal of Computational Physics*, 428:109987, 2021.

[29] Howard C Elman, Victoria E Howle, John N Shadid, and Ray S Tuminaro. A parallel block multi-level preconditioner for the 3D incompressible Navier–Stokes equations. *Journal of Computational Physics*, 187(2):504–523, 2003.

[30] J. Jonkman, S. Butterfield, W. Musial, and G. Scott. Definition of a 5-MW reference wind turbine for offshore system development. Technical Report NREL/TP-500-38060, National Renewable Energy Laboratory, 2009.

[31] F. R. Menter, M. Kuntz, and R. Langtry. Ten years of industrial experience with the SST turbulence model. In K. Hanjalic, Y. Nagano, and M. Tummers, editors, *Turbulence, Heat and Mass Transfer 4*, pages 625–632. Begell House, Inc., 2003.