Wind Farm Simulations Using an Overset *hp*-Adaptive Approach with Blade-Resolved Turbine Models

Andrew C. Kirby[*], Michael J. Brazell, Zhi Yang, Rajib Roy, Behzad R. Ahrabi,
Dimitri J. Mavriplis, Michael K. Stoellinger

Department of Mechanical Engineering, University of Wyoming, Laramie, WY 82071, USA

Jay Sitaraman

Parallel Geometric Algorithms, LLC, Sunnyvale, CA 94087, USA

Blade-resolved numerical simulations of wind energy applications using full blade and tower models are presented. The computational methodology combines solution technologies in a multi-mesh, multi-solver paradigm through a dynamic overset framework. The coupling of a finite-volume solver and a high-order, *hp*-adaptive finite-element solver is utilized. Additional technologies including in-situ visualization and atmospheric micro-scale modeling are incorporated into the analysis environment. Validation of the computational framework is performed on the NREL 5MW baseline wind turbine, the unsteady aerodynamics experimental NREL Phase VI turbine, and the Siemens SWT-2.3-93 wind turbine. The power and thrust results of all single turbine simulations agree well with low-fidelity model simulation results and field experiments when available. Scalability of the computational framework is demonstrated using 6, 12, 24, 48, and 96 wind turbine wind plant set-ups including the 48 turbine wind plant known as Lillgrund. Demonstration of the coupling of atmospheric micro-scale and CFD solvers is presented using the NCAR WRF solver and the NREL SOWFA solver. Comprehensive validation with measurements will be the focus of future research of the computational framework.

I. Introduction

Wind energy today is becoming an emergent renewable energy source throughout the United States and the world. Wind energy costs have drastically dropped over the last decade through advanced design and increased scale of turbines thus making wind energy a desirable renewable alternative to fossil fuel based energies. It is estimated that wind energy may produce as much as 20% of the total electrical energy needs by 2030 and 35% by 2050 in the United States, which will have a profound economic and societal impact [1]. The transition from fossil fuels to renewable energies will strengthen energy security and reduce greenhouse-gas emissions. Improved understanding will enable better wind turbine placement in wind farm configurations which will increase wind plant efficiency by a few percent. This can produce large economic impacts particularly for wind plants containing a few hundred multi-mega-watt turbines. Predictive simulations of wind plants in complex terrains has ushered in the need for exascale-enabled software development. Two organizations in the Department of Energy (DoE), the Office of Science and the National Nuclear Security Administration, have formed a collaborative effort to establish the Exascale Computing Project (ECP) [2]. The ECP was established to maximize the benefits of high performance computing (HPC) and accelerate the development of a capable exascale computing ecosystem. A part of the ECP, a project titled ”Exascale Predictive Wind Plant Flow Physics Modeling” has been formed to advance the understanding of wind plant flow physics such as wake dynamics, complex terrain effects, and turbine-turbine interaction effects [3]. The primary objective of this wind plant modeling project is to develop an exascale-capable system software application that will accurately simulate a wind plant containing on the order of 100 wind turbines within a 10 km by 10 km area consisting of complex terrain [4]. An estimate of 100 billion degrees of freedom will be required to simulate this problem. The goals of the current work

[*] Corresponding author: akirby@uwyo.edu

are well aligned with the objectives of the DoE ECP. The motivation of the current work is to develop and demonstrate complementary technologies that can be used at larger scale within the DoE ECP.

State-of-the-art modeling techniques of wind plants are transitioning from reduced fidelity models such as turbine parameterization techniques such as actuator lines [5–8] or actuator discs [9] to high fidelity blade-resolved models [10–24]. High fidelity simulations require the use of a full rotor model where the exact geometry of the turbine blade and tower is used. These models were computationally prohibitive until recent advancements in HPC technologies. The present day leadership class supercomputing environment includes systems containing on-the-order-of one million to ten million computing cores [25]. Wind plant simulations using full-rotor models have recently been applied using the CREATE-AV HELIOS [26–28] software. HELIOS uses a multiple-mesh, multiple-solver paradigm with an overset framework. A computational study using HELIOS for wind turbine simulation was performed by Gundling et al. [29]. In the work of Sitaraman et al. [28], HELIOS was used for a blade-resolved wind plant simulation containing 48 wind turbines under ideal and atmospheric conditions using 3,840 CPU cores. The full rotor model mesh in this work contained just under 475,000 nodes per blade and the tower mesh contained approximately 500,000 nodes. The 48 wind turbine plant equated to approximately 96 million near-body mesh points and the off-body adaptive mesh system grew from 50 million to 180 million nodes giving a grand total of nearly 280 million degrees of freedom (DOFs). Those results demonstrated the ability to simulate an entire wind plant using relatively coarse meshes for the full rotor model in an overset framework using multiple meshes and multiple flow solvers.

The goal of this work is to make advancements toward the exascale grand challenge problem of simulating wind plants using full rotor models in complex terrain environments under atmospheric inflow conditions at higher resolution. To perform this task, appropriate physics, numerical solvers, and scalability on large high performance computing systems are required. Our approach involves an analogous simulation environment to the HELIOS software through a computational overset framework using a multiple-mesh, multiple-solver paradigm. This approach of overlapping grids has been utilized in several works [26, 30–33]. Within this framework, we employ a near-body, off-body mesh philosophy. The near-body mesh system is designed to handle complex geometries by using unstructured meshes and the off-body mesh system is designed to use dynamically adaptive Cartesian meshes for enabling flow feature tracking with high levels of solution accuracy. Additionally, high-order methods on Cartesian meshes allow the potential for effectively leveraging the power of emerging computer architectures [34, 35]. In particular, we employ a high-order discontinuous Galerkin (DG) Finite Element discretization in the off-body region. DG methods and similar FEM solvers have been deployed for a variety of aerospace applications [36–46] including the use of adaptive mesh and solution refinement techniques [47–49].

The paper is outlined as follows. In Section II, the computational methodology outlining the individual software components is presented. Sections III introduces the simulation problems outlined in the subsequent results sections. Results are presented for three wind turbines including validation studies in single turbine configuration in uniform inflow conditions (Section IV), a weak scaling study using multiple wind turbine arrangements (Section V) and, lastly, demonstration of the micro-scale atmospheric coupling (Section VI). Section VII concludes the paper with a discussion on future work.

II. Computational Methodology

The computational framework deployed in this work is known as the **W**yoming **W**ind and **A**erospace **A**pplications **K**omputation **E**nvironment (W^{2}A^{2}KE3D). W^{2}A^{2}KE3D has been previously demonstrated on various aerodynamics problems such as flow over the NACA0015 wing and flow over a sphere [50]. Additionally, preliminary wind energy application results were presented using the NREL Unsteady Aerodynamics Experiment Phase VI [51]. The framework is derived to have a flexible solver and mesh system paradigm in order to perform simulations for a large class of problems in aerospace and wind energy.

W^{2}A^{2}KE3D is designed to support a dynamic overset system using multiple flow solvers and multiple computational meshes. The mesh system generally consists of a collection of *near-body* and *off-body* meshes. The near-body meshes are inherently unstructured to model the complex geometry and resolve boundary layers of wind turbine bodies. The off-body mesh is a dynamically adaptive Cartesian grid system. The use of Cartesian meshes in the off-body region allows for efficient flow solvers, efficient storage, and ease of dynamic solution-based mesh adaption. This multiple mesh paradigm allows for effective use of solver and mesh technologies in variable flow conditions. This hypothesis holds strongly for wind energy applications that require capturing boundary layer phenomenon as well as wake dynamics. Figure 1 demonstrates an overset mesh system of a traditional three bladed wind turbine with a tower and nacelle. Each blade is represented by a single mesh that is replicated, transitioned, and rotated to the correct starting position. The tower geometry is fitted with an independent unstructured near-body mesh component.

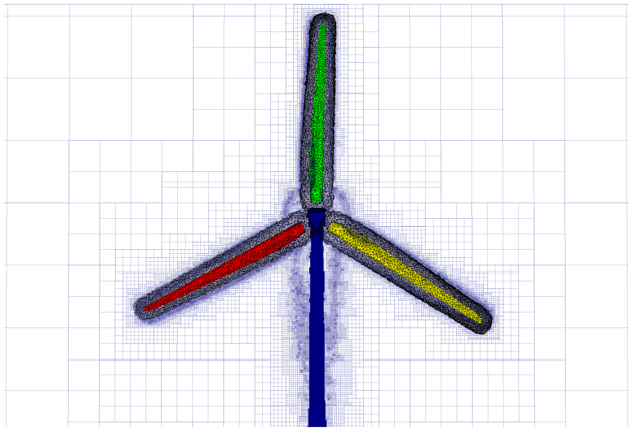


Figure 1. NREL 5MW wind turbine overset mesh system. One turbine blade unstructured mesh is replicated three times, rotated and translated to the initial positions. A fourth unstructured mesh is used to represent the tower and nacelle. The off-body adaptive mesh is visualized in the background.

The W^{2}A^{2}KE3D framework allows for multiple CFD solvers individually optimized for their respective mesh system in the multiple-mesh paradigm. The use of multiple meshes and multiple flow solvers introduces the requirement of coordination. A C-programming-language based driver choreographs all flow solvers and all mesh systems. This driver allows for solvers to run on disjoint groups of CPU cores, allowing for variable amounts of computational resources to be allocated appropriately where needed. This is particularly important for the off-body solver which uses a dynamically adaptive mesh. During the evolution of a wind turbine simulation, the flow features of interest such as the wake require additional mesh resolution. As the propagation of the wake grows over time, more computational resources are required in the off-body regions. The flow solvers present in the framework can be redistributed to different amounts of cores at the beginning of restarted simulations. This allows for long run-time simulations to be moderately load balanced. Additionally, once a solver is implemented into the framework, flow visualization and analysis is provided to the solver through the use of in-situ visualization. Pointers to the flow solver's data and mesh are held by the driver which is fed to the in-situ flow visualization software implemented the driver.

II.A. Near-Body Flow Solver

The near-body flow solver utilized in W^{2}A^{2}KE3D is NSU3D (**N**avier-**S**tokes **U**nstructured) [52,53]. NSU3D is a well-established Unsteady Reynolds Averaged Navier-Stokes (URANS) solver for unstructured meshes. The discretization is based on a vertex-centered finite-volume method with weighted least-squares gradient reconstruction providing second-order spatial accuracy. The solver uses automatic agglomeration multigrid along with line-implicit preconditioning for accelerated solution convergence [54]. NSU3D contains several turbulence models for various aerodynamics problems. This includes the Spalart-Allmaras (SA) [55], K-Omega [56], and the Delayed Detached Eddy Simulation (DDES) [57] turbulence models with rotation/curvature correction [58]. The SA and the DDES turbulence models with rotation correction are the primary methods employed in this work.

NSU3D has been demonstrated on multiple aerodynamics problems and has been a regular participant in the AIAA High Lift Prediction Workshop [59] and the AIAA Drag Prediction Workshop [60] series. The solver has been demonstrated to have good strong scalability on the NCAR supercomputer NWSC-1 Yellowstone up to 32,768 CPU cores. Additionally, NSU3D has served as a near-body flow solver in the CREATE-AV HELIOS [26] software. W^{2}A^{2}KE3D is analogous to HELIOS in that they each utilize the solution strategy composed of a near-body mesh and flow solver, an off-body mesh and flow-solver, and an overset mesh connectivity component.

II.B. Off-Body Flow Solver

W^{2}A^{2}KE3D utilizes a variable order discontinuous Galerkin (DG) finite-element method implemented in a dynamic adaptive mesh refinement (AMR) framework. In previous work, W^{2}A^{2}KE3D used an off-body solver known as SAMCartDG based on the open-source SAMRAI [61] AMR package provided by Lawrence Livermore National Laboratory with the stand-alone DG solver CartDG [62]. SAMRAI is a patch-based AMR system. For this work, the authors implemented the flow solver CartDG into an octree-based AMR system known as `p4est` [63, 64]. The `p4est` software library uses scalable adaptive algorithms for parallel

AMR on forests of octrees [65]. The p4est library was part of the winning team of then ACM Gordon Bell Prize in 2015 [66] and has been shown to scale to half a million CPU cores.

The off-body solver composed of CartDG and p4est is known herein as dg4est. In our experience, the use of p4est allows for greater ease of implementation of a finite element based solver in comparison to the patch-based AMR system used in SAMRAI. Both SAMRAI and p4est implement h-refinement, defined as the process of physically refining or coarsening mesh cells. On the other hand, neither library provides support for p-refinement, defined as the process of raising or lowering the order of discretization in individual mesh cells or patches of cells. However, the octree-based AMR system p4est has an analogous communication pattern to a traditional finite element method. That is, the parallel processing communication pattern in the finite element method, particularly for discontinuous Galerkin methods, is composed of a nearest-neighbor stencil. The AMR framework also allows for other stencils such as continuous and extended. When parallel communication is invoked, the nearest-neighbor elements on a mesh partition boundary are communicated to the respective neighboring processor. The can be viewed as an exchange of cell solutions on a mesh partition boundary. In contrast, since SAMRAI utilizes a patch-based AMR system, the native parallel communication patterns are composed of data transfers between AMR level interfaces. These data transfers generally occur in a coarse to fine level procedure. After advancing the solution on the finer levels, the solution on the fine levels are disseminated to coarser levels through a coarsening operator. This adds extraneous communication and algorithmic complexities in comparison to the direct cell solution exchanges in p4est.

The numerical kernel solver, CartDG, discretizes the compressible Navier-Stokes equations with Coriolis and gravitational forces. There are user options for a constant Smagorinsky Subgrid-Scale (SGS) Large Eddy Simulation (LES) model [67]. CartDG is designed for computational efficiency on Cartesian meshes. To maximize performance, CartDG is designed using a tensor-product, collocation based DG method making simplifications for Cartesian meshes when available. Through this approach, the numerical complexity is reduced. The details of the algorithm are outlined in Kirby et al. [62] which follows the derivation in Diosady et al. [41] and Hindenlang et al. [68]. The off-body solver dg4est has been demonstrated on various problems such as flow over the three-dimensional wing NACA0015 at moderate Reynolds number and on the Taylor-Green Vortex flow problem [69].

II.C. Overset and Domain Connectivity Assembler

By adopting a multiple-mesh, multiple-solver paradigm in an overset framework, domain connectivity and solution interpolation is required. To fulfill this requirement, the **T**opology **I**ndependent **O**verset **G**rid **A**ssembler (TIOGA) [32,33,70] is utilized. TIOGA relies on an efficient parallel implementation of the Alternating Digit Tree (ADT) algorithm in order to handle point-in-cell inclusion tests to determine connectivity. TIOGA determines the donor-receptor patterns of overlapping mesh grids and performs the solution interpolation using the appropriate solution accuracy orders required by the respective flow solvers. To perform high-order solution interpolation, several call-back functions are provided in the TIOGA API [32]. These functions include: receptor node list generation, high-order donor inclusion test, high-order interpolation weight generation, and interpolated solution conversion to high-order solution coefficients. For high-order methods, the use of multiple points inside each cell is required. Thus the flow solver must provide a list of the node locations inside a cell to be interpolated. Additionally, for high-order methods, the use of high-order mesh geometries may be used, which include curved cells and faces, therefore requiring a high-order approach to provide a donor-inclusion test. The donor-inclusion test for high-order methods maps physical point coordinates to natural coordinates in the standard isoparametric reference space using a geometric basis function mapping. This transformation forms a system of nonlinear equations which are solved via a Newton-Rhaphson method. Once the natural coordinates are found, it is trivial to test if the point is inside the cell. Once donor cells are identified, the solution interpolation order of accuracy is required to be of the same order as the solution order of accuracy. Lastly, if the high-order numerical method is a modal-based finite element solver, then the interpolated solution needs to be converted to solution coefficients, which can be done using either a mass matrix or a Vandermode matrix approach [32]. TIOGA is agnostic to mesh element types and numerical discretizations. Therefore mixed-element meshes can be used concurrently with any combination of numerical discretizations such as a finite-volume solver with a high-order finite-element method. TIOGA's ability to perform high-order mesh assembly and high-order solution interpolation using mixed flow solvers has been demonstrated on many canonical problems such as the Ringleb flow, flow over the NACA0015 wing, and flow over a sphere [50]. TIOGA has also been utilized in high-order solution techniques of intersecting Hamiltonian path and strand mesh grids [71].

II.D. Micro-scale Atmospheric Inflow Coupler

Faithful representation of wind plants through simulation requires capturing all fluid scales and physical environments. This introduces complex terrain and atmospheric inflow conditions thus requiring meteorological micro-scale flow conditions. To achieve this, the large fluid scales are introduced through a one-way coupling between precursor atmospheric turbulence solvers and the off-body flow solver. The off-body flow solver then transfers these atmospheric conditions to the near-body CFD flow solver via the overset assembler and interpolator. The coupler incorporates a choice of two precursor atmospheric solvers: the National Center for Atmospheric Research's (NCAR) **W**eather **R**esearch and **F**orecasting (WRF) [72] model and the National Renewable Energy Laboratory's (NREL) **S**imulation f**O**r **W**ind **F**arm **A**pplications (SOWFA) [6] model. SOWFA has been extensively used in simulation of wind plant modeling in various atmospheric conditions [8] and complex terrains [73–75].

The atmospheric solvers are run as precursor simulations prior to the CFD simulation to accurately capture the atmospheric boundary layer (ABL). These precursor simulations generate the initial and boundary flow field conditions for the CFD flow solvers. An intermediary pseudo-flow solver reads in the precursor atmospheric flow mesh and data over the duration of the wind plant simulation. This pseudo-flow solver is treated in similar fashion as a regular CFD solver to the overset grid connectivity assembler. The mesh and data are registered with TIOGA as its own mesh system from which the atmospheric data is interpolated on to the CFD solver mesh system to establish the entire initial condition, and on to the CFD mesh system boundaries at each subsequent time step of the simulation. Between precursor atmospheric flow solutions, which are written in files, the pseudo-solver performs linear time interpolation of the data. TIOGA then spatially interpolates the atmospheric data, in a one-way coupling manner, to the CFD flow solver.

II.E. Flow Visualization and Post-Processing

Numerical simulations of wind plants can contain multiple billions of degrees of freedom which translates into tens to hundreds of gigabytes of data for a single time instance. Therefore, the ability to post-process flow visualization becomes intractable due to the shear amount of data. To alleviate the big-data issue, flow visualization and data analysis are no longer performed as a post-processing step by reading in data written to disk; analysis and visualization are performed while the data is being generated in the simulation. This in-situ analysis allows for real-time visualization and reduction in the amount of data written to disk.

To perform in-situ visualization and analysis, LLNL's VisIt Libsim [76] library has been instrumented into the driver of W^{2}A^{2}KE3D . Libsim is a library in the VisIt software package that enables fully-featured visualization and data analysis by feeding the data from the simulation to the VisIt algorithms during runtime. Through tight-coupling of the in-situ library, various data filters and data extraction tools are available through the VisIt interface. For example, the user has the option to output data databases, isosurfaces, and slices. Additionally VisIt can directly output image formats. Processing of data is done in parallel on the computing cores used in the simulation, thus accelerating visualization frame rates. The instrumentation of Libsim into the code has minimal invasiveness. By instrumenting the driver with Libsim, any flow solver that is coupled in the W^{2}A^{2}KE3D framework is provided with in-situ visualization and analysis automatically.

II.F. Driver

The driver software is responsible for controlling all component solvers embedded in a multiple-mesh and multiple-solver, overset framework. Inevitably, different meshes and independent flow solver speeds introduce variable amounts of computational work and efficiency. In a parallel computing environment, the software developer is presented with a few options to make computational load balancing more amenable: (i) place all flow solvers on all CPU cores, (ii) allocate disjoint groups of CPU cores to each flow solver. In the former solution, all flow solvers are partitioned across all CPU cores and execution of the flow solvers are serialized with respect to each other. The latter solution allows for flow solvers to execute in parallel and allows for each flow solver to have different numbers of CPU cores. In W^{2}A^{2}KE3D , option (ii) is chosen for the flexibility it provides regarding different solver requirements and scalability. However, this flexibility can add development and algorithmic complexity. As a simulation evolves, the off-body solver may be dynamically adapting which can introduce more overall degrees of freedom. This can lead to a load imbalance of the flow simulation. To alleviate this problem, redistribution of the problem can be performed between simulation campaign restarts for more effective use of computational resources. That is, more CPU cores are allocated to the flow solver that requires more computational resources as the solution evolves. Each flow solver component in the W^2A^2KE3D framework has this capability: NSU3D needs to be manually redistributed but `dg4est` has automatic redistribution when provided more CPU cores for a restarted solution.

A sample driver software work flow is portrayed in Figure 2 which illustrates a scenario with two near-body mesh groups and one off-body mesh group. The sample demonstrates each flow solver is independent to solve their respective time dependent problem in parallel without dependence of other solvers or meshes. During a simulation, a global time orchestrates when data is exchanged between the near-body mesh and the off-body mesh. When an unsteady time step is executed, the near-body and off-body flow solvers iterate in time in an uncoupled manner for a Δt time step which we refer to as the global time step. When the global time step is completed, data is exchanged between the near-body and the off-body solvers through the overset interface. Cells receiving data through the overset interpolation are known as receptor cells. These receptor cells zero their respective residual values and their solution is provided by the solver's counterpart mesh solution (near-body solution from off-body solution, and vice-versa). This receptor interpolated solution is then held fixed during a global time step therefore serving as a *boundary condition* for surrounding mesh elements. Thus the global time step dictates the duration of interpolated solution which serves as a fixed value boundary condition to surrounding elements. Following the time step update of the solvers, the overset assembler performs a data update in which the solutions are exchanged between meshes. The flow solvers then have the ability to perform auxiliary functions such as mesh motion or mesh adaption. If any auxiliary meshing routines are called, the new grid information is registered and processed with the overset assembler.

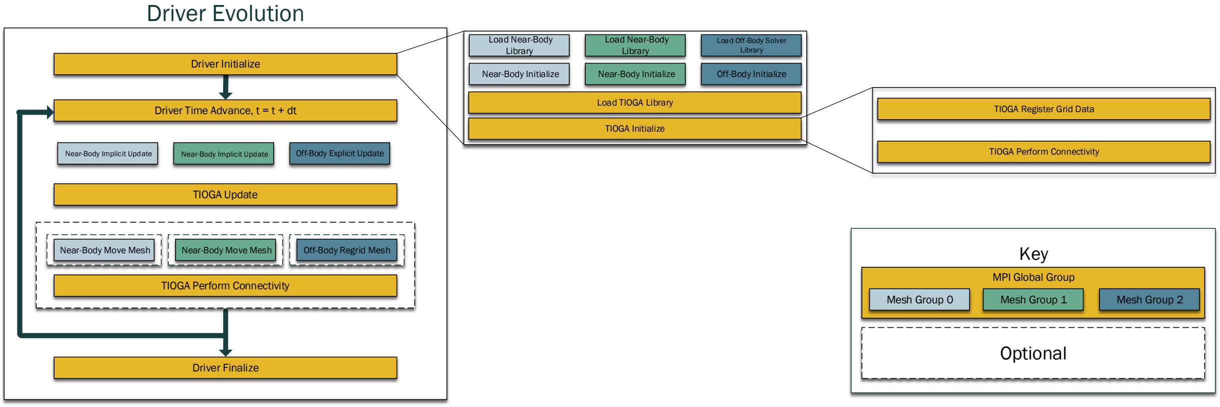


Figure 2. A driver code is used to choreograph all flow solvers, mesh movement and adaption, overset data update and grid connectivity, and in-situ visualization. All flow solvers are allocated disjoint groups of CPU cores for parallel flow solution updates.

III. Results: Overview

W^2A^2KE3D is demonstrated for three wind turbines: (i) NREL 5MW, (ii) NREL Phase VI, (iii) Siemens SWT-2.3-93. For the NREL 5MW turbine, we perform mesh resolution, time refinement, and sub-iteration studies for the near-body time-accurate solver. Analysis of single turbine performance for multiple uniform inflow velocities is performed for the NREL 5MW and for the NREL Phase VI wind turbine. Second, we perform weak scaling of the computational framework by introducing more wind turbines on comparable counts of CPU cores per turbine. Examination of a long run time simulation of the 48 wind turbine Lillgrund Wind Farm is performed and a discussion of a 144 turbine wind plant simulation is presented. Lastly, we demonstrate the micro-scale atmospheric and CFD coupling using the WRF and SOWFA solvers.

The near-body meshes are constructed for stand-alone simulations and tested for accuracy. Once the mesh is validated in a simulation using the near-body solver in stand-alone mode, the mesh is trimmed to a user specified distance from the surface of the body and overset with the background Cartesian mesh. For all overset simulations, the off-body mesh system uses as many refined mesh levels as necessary to match the grid resolution of the near-body unstructured mesh cells located at the trimmed mesh boundary. The finest level in the AMR mesh system for the off-body mesh uses $p = 1$, second-order spatial solution accuracy and all coarser levels use higher p-orders (typically $p = 2$, third-order spatial solution accuracy). Thus off-body cells that serve as donors and receptors to the near-body mesh match the second-order spatial accuracy of the near-body discretization, but away from these areas on coarser mesh levels, the solution order of accuracy is increased. This solution accuracy strategy is adopted to mitigate the explicit CFL time step restriction for the off-body solver since the time step restriction of the small p=1 cells is roughly similar to that incurred by the larger but higher p cells. For all simulations, all solvers are evolved using an unsteady formulation with time-accurate methods. The near-body solver employs the implicit BDF-2 method and the off-body solver executes the classical explicit Runge-Kutta four-stage method (RK4). Each global time step is loosely coupled; at the end of each global time step, the flow solutions on each mesh is exchanged and interpolated to their counterpart receptor cells.

IV. Results: Single Turbine Simulations

IV.A. NREL 5MW

The NREL offshore 5MW turbine is a concept design aimed at assessing offshore wind turbine technologies. The wind turbine design is a conventional three-bladed upwind variable-speed blade-pitch-to-feather-controlled turbine [77]. This model turbine is used for reference to standardize baseline offshore wind turbine specifications. The NREL 5MW turbine has a blade radius of 63.0 meters with pre-cone angle of 2.5° and a shaft angle of 5°. The simulations assume rigid blades and a rigid tower with nacelle of height 90.0 m. The rated rotor speed is 12.1 RPM at nominal conditions.

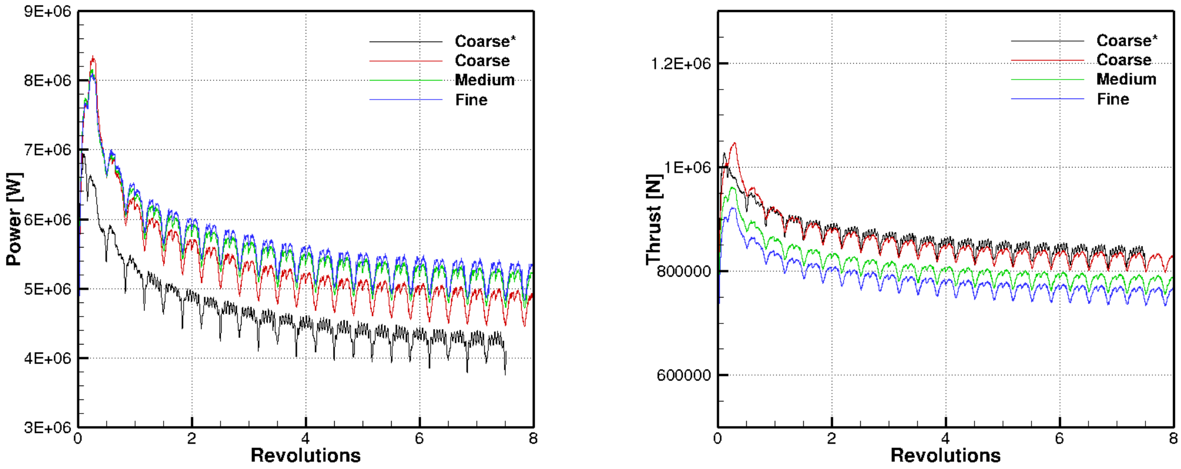
We perform a mesh convergence study using two coarse meshes, one medium mesh, and one fine mesh. Table 1 outlines the mesh statistics for each of the meshes. The coarsest mesh contains approximately 360,000 mesh nodes per blade where are the fine mesh contains nearly 2.88 million mesh nodes per blade. The tower mesh for all mesh resolution cases is fixed at just over 500,000 nodes. For a full turbine configuration, the total node count can vary from 1.58 million to nearly 9.12 million. We note that the coarse, medium, and fine meshes are representatives of a family of meshes where the coarse and fine meshes are derived from the medium mesh, while the mesh denoted by Coarse* is generated independently.

Mesh	Mesh Points	Tetrahedra	Pyramids	Prisms
Coarse*	474,383	780,283	8,926	661,274
Coarse	360,148	473,747	9,557	539,715
Medium	927,701	1,922,304	12,928	1,162,586
Fine	2,873,862	6,898,579	28,751	3,306,509

Table 1. Mesh statistics used in the mesh convergence study of the NREL 5MW wind turbine blade. Each blade mesh is replicated and placed into the correct starting position in the 3-bladed configuration at the beginning of the simulation. The coarse, medium, and fine meshes are a family of meshes; the coarse and fine meshes are derived from the medium mesh. The coarse* mesh is constructed independently.

The rated power of this wind turbine, as the name suggests, is 5 mega-watts at the nominal inflow rate of 11.4 m/s. We perform the mesh convergence study using the nominal inflow velocity with a time step corresponding to 1/4° of turbine rotation. The near-body solver uses 50 sub-iterations for the BDF-2 time step. The explicit method used by the off-body solver is limited by the CFL number so it performs sub-cycles using its maximal stable time step allowable until it reaches to the global time step corresponding to 1/4° of rotation which typically is less than 100 sub-cycles.

Figure 3 demonstrates the time history convergence of the mesh resolution study. Figure 3(a) shows the power time history convergence at inflow velocity 11.4 m/s for each mesh over the evolution of rotor revolutions and Figure 3(b) shows their respective thrust convergence histories. The target thrust value at 11.4 m/s is 730,000 Newtons. As demonstrated in the mesh convergence study figures, the need for at least medium refined meshes is required at nearly one million nodes per blade to capture the power correctly. The thrust convergence shows slightly more variation between the medium and fine meshes, although all values are clearly converging with additional mesh resolution. This provides an estimate that over two million mesh points per blade are required to accurately capture the aerodynamic forces on the wind turbine. Results for the Medium and the Fine meshes are close in both power and thrust forces in comparison to the two coarse meshes. Even though the mesh denoted Coarse* has more mesh elements and nodes in comparison to the standard Coarse mesh, the mesh nodes in the Coarse mesh are more appropriately placed along the blade edges and tips leading to better results. A noticeable difference in the power prediction between the two coarse meshes is shown in Figure 3(a) while the thrust prediction is only marginally effected in Figure 3(b). Highly oscillatory convergence features are also noticed in the Coarse* mesh compared to the Coarse, Medium, and Fine meshes. Recalling the Coarse, Medium, and Fine meshes are a family, we see the high frequency content in both the power and thrust curves have the same characteristics in contrast to the Coarse* mesh. The low frequency dips in the force histories are caused by the effect of the wind turbine blade passing by the tower on the downswing of rotation resulting in three dips per revolution.



(a) Power convergence history for inflow velocity 11.4 m/s.

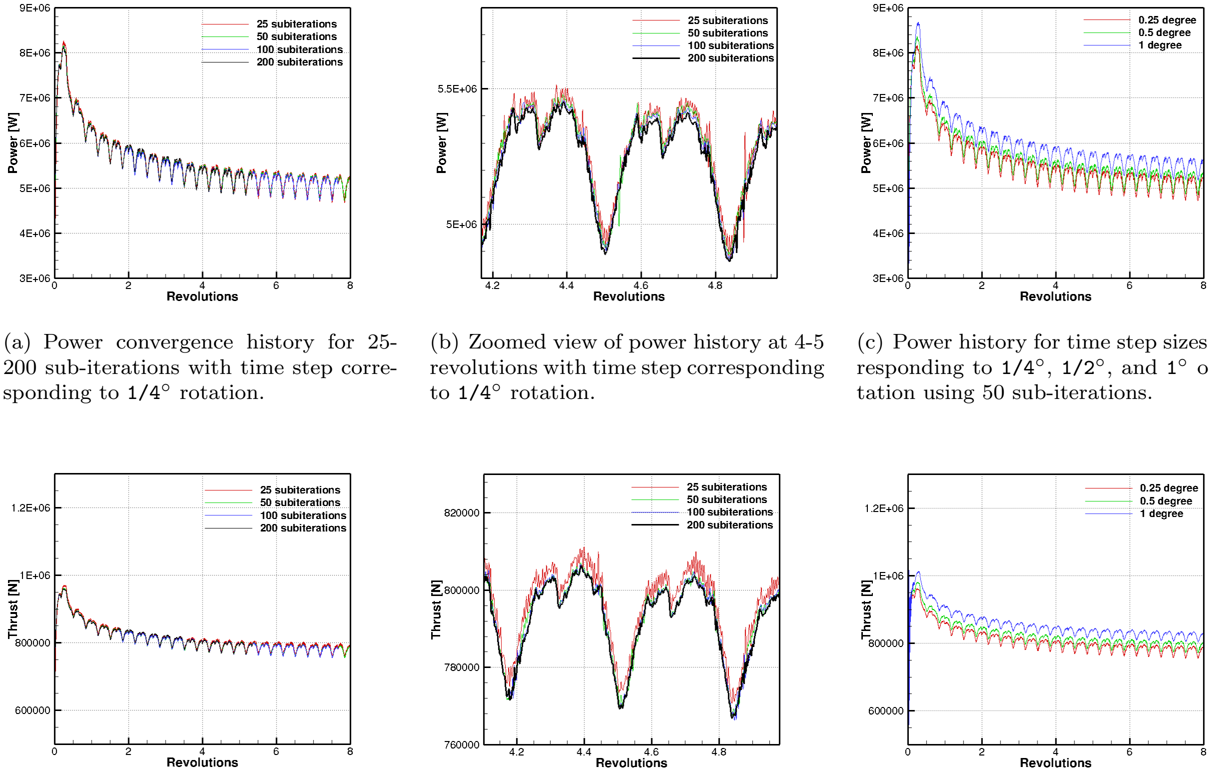
(b) Thrust convergence history for inflow velocity 11.4 m/s.

Figure 3. NREL 5MW power and thrust simulation results for the mesh resolution study. Each simulation uses a time step corresponding to a $1/4^\circ$ rotation. Each time step was solved with BDF-2 using 50 sub-iterations for the near-body flow solver. Reference solution data provided by the NREL FAST [77] software.

A time-step and numerical convergence study is performed in Figure 4. Row 1 shows the power prediction histories and row 2 shows the thrust prediction histories. Figure 4(a) demonstrates the result of using more sub-iterations for the near-body solver in the BDF-2 time stepping scheme and Figure 4(b) shows the detailed high frequency content by zooming into the 4-5 revolution time frame. The results indicate using more sub-iterations in the time step smooths the highly oscillatory content in the simulation. Overall the mean values of power prediction using more sub-iterations remain the same as using fewer sub-iterations. Figure 4(d) and (e) show similar results for the thrust forces.

In contrast to the sub-iteration convergence study, the time step study demonstrates a significant influence of the global time step size for force prediction. Figure 4(c) and (f) show the power and thrust time histories for three sizes of the global time step, respectively. For this study, we choose global time steps corresponding to $1/4^\circ$, $1/2^\circ$, and 1° of rotor rotation. We can define a local blade tip CFL number as the product of blade tip speed of sound and the global time step divided by the finest mesh element size in the off-body mesh system. The local CFL numbers are 1.02, 2.05, and 4.09 for global time steps corresponding to $1/4^\circ$, $1/2^\circ$, and 1°, respectively. This local CFL represents the cell distances an acoustic wave can travel in a global time step. Thus for the $1/4^\circ$ time step, an acoustic wave may travel an entire cell width before the near-body and off-body solutions are updated between the two mesh systems. We see that when we choose a large time step corresponding to a large local CFL, the initial solution transients are higher than using a smaller time step. As the step size is decreased, the solution converges to a refined time-step solution. From Figure 4(c) and (f), it is suggested that the values of power and thrust will be over predicted unless a sufficiently small global time step of the order of $1/4^\circ$ is used.

Figure 5 shows the power and thrust predictions for inflow velocities 6-11.4 m/s using the medium mesh compared to the NREL FAST [77] reference solution. The reference solution uses blade element momentum theory with a fluid-structure interface to model structural wake effects [77]. The power predicted from W^2A^2KE3D agrees well with FAST and the thrust is slightly under predicted for inflow velocities less than the nominal inflow velocity of 11.4 m/s. We note as the velocity increases, the power becomes slightly under predicted using the medium refined mesh but notice the power is improved when using the fine mesh. We also note that the W^2A^2KE3D framework results do not contain blade elastic structural deflection responses.



(a) Power convergence history for 25-200 sub-iterations with time step corresponding to 1/4° rotation.

(b) Zoomed view of power history at 4-5 revolutions with time step corresponding to 1/4° rotation.

(c) Power history for time step sizes corresponding to 1/4°, 1/2°, and 1° of rotation using 50 sub-iterations.

(d) Thrust convergence history for 25-200 sub-iterations with time step corresponding to 1/4° rotation.

(e) Zoomed view of thrust history at 4-5 revolutions with time step corresponding to 1/4° rotation.

(f) Thrust history for time step sizes corresponding to 1/4°, 1/2°, and 1° of rotation using 50 sub-iterations.

Figure 4. NREL 5MW force histories using BDF-2 time stepping for the near-body flow solver. All results performed on the Medium refined mesh in Table 1.

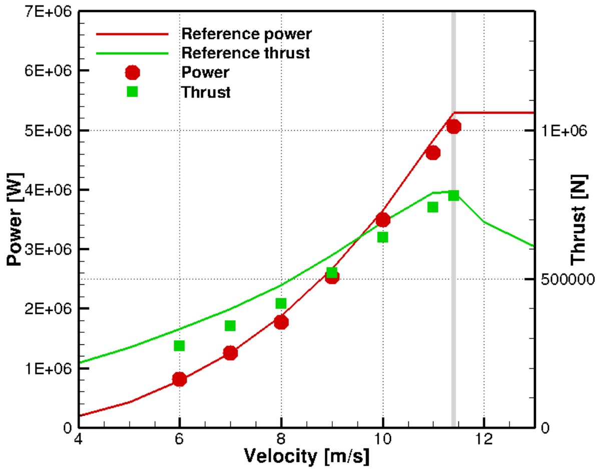


Figure 5. NREL 5MW power and thrust simulation results using a time step corresponding to a 1/4° rotation. Each time was solved with BDF-2 using 50 sub-iterations for the near-body flow solver on the Medium mesh. Reference solution data provided by the NREL FAST [77] software.

IV.B. NREL Phase VI

The NREL Unsteady Aerodynamics Experiment Phase VI is a wind turbine that has been studied experimentally [78–81]. The wind turbine was studied at NASA Ames Research Center in the 80 ft x 120 ft (24.4 m x 36.6m) wind tunnel. The experiment of the Phase VI wind turbine is regarded as one of the most extensive studies performed for a wind turbine.

The Phase VI turbine has a blade radius of 5.029 m and the rotor is assumed to be rigid with a blade pitch angle of 3°, a yaw angle of 0°, and a 0° cone angle for this computational study. The blade geometry is constructed from a single NREL S809 airfoil [78]. The rotation rate is prescribed at 72 RPM. This tower and nacelle are excluded for this case. The near-body mesh used in this simulation contains approximately seven million elements and three million nodes which extend one chord length from the surface of the blade. Figure 6 shows the near-body surface mesh and Figure 7(a) depicts the near-body and off-body mesh system.

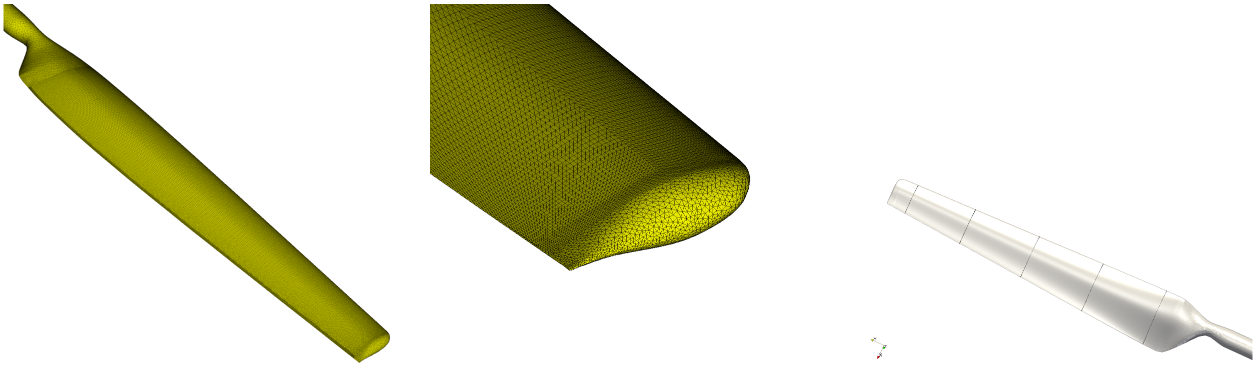
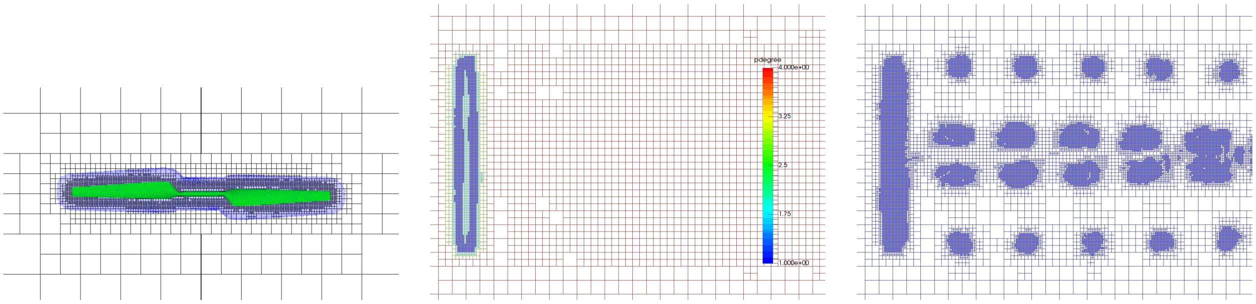


Figure 6. NREL Phase VI computational near-body mesh containing 7 million elements and 3 million nodes. The right figure shows the span-wise stations used for pressure coefficient measurements.



(a) Near-body mesh trimmed to one chord length shown in blue with the surface of the blade shown in green.

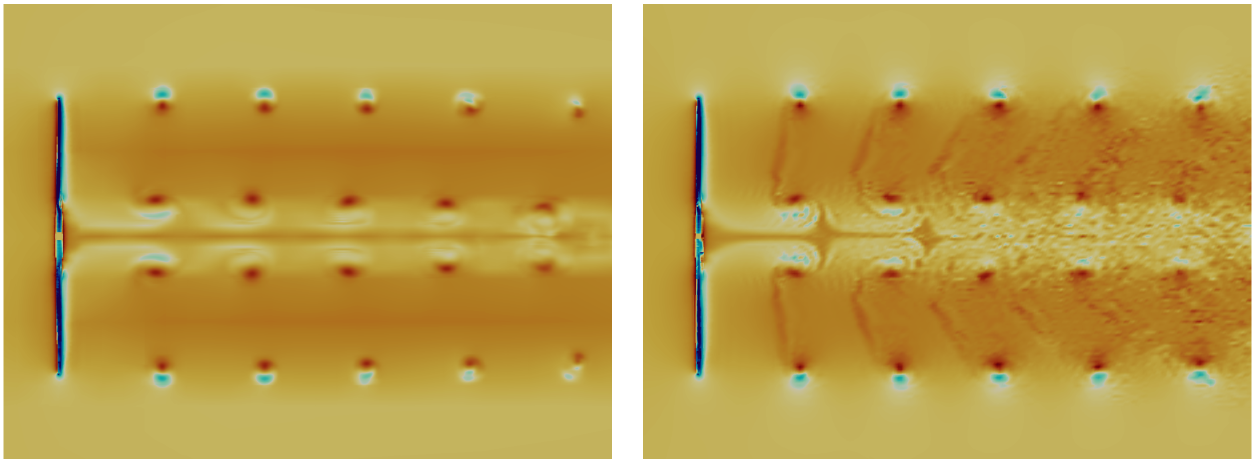
(b) Off-body mesh adaption colored by spatial solution order of accuracy with $p = 1$, second-order, near the wind turbine and growing to $p = 4$, fifth-order accuracy, in the wake region.

(c) Off-body mesh adaption of $p = 1$, second-order, spatial solution order of accuracy.

Figure 7. NREL Phase VI overset mesh system with wake mesh adaption.

The inflow conditions vary with velocities ranging from 7-15 m/s and a Reynolds number of 2.5 million based on the chord length of the wind turbine blade. The global time step is set to 1/4°of rotation and the near-body flow solver uses 25 sub-iterations per BDF-2 time step. The off-body mesh domain is 1000 m with the mesh system composed of 10 octrees in each coordinate direction and 11 levels of refinement. The finest AMR level uses $p = 1$, second-order spatial accuracy and the coarser levels transition to a $p = 4$, fifth-order spatial accuracy. To do this transition, each subsequent level from the finest mesh level increases its spatial order of accuracy by one degree until the maximal polynomial degree of $p = 4$ is achieved. For this particular study, level 11 uses $p = 1$, level 10 uses $p = 2$, level 9 uses $p = 3$, and all coarser levels use $p = 4$. Thus higher order spacial accuracy is used in the regions away from the unstructured mesh particularly in the wake region. Figure 7(b) demonstrates the spatial order of accuracy in different regions of the off-body mesh showing fewer elements are needed in the wake region when higher p-degrees are used. Figure 7(c) shows the difference in adaptive mesh refinement when only $p = 1$, second-order accurate elements are used.

Figure 8 demonstrates the use of second-order spatial accuracy in the wake region in comparison to fifth-order spatial accuracy. The fifth-order accurate solution is able to capture finer turbulence scales whereas the second-order accurate solution smears the details of the wake structure. The fifth-order accurate solution requires three less AMR levels in the wake region therefore reducing the overall element count.



(a) Velocity magnitude of $p = 1$, second-order accuracy.

(b) Velocity magnitude of $p = 4$, fifth-order accuracy.

Figure 8. NREL Phase VI wake comparison of second-order and fifth-order spatial accuracies.

Figure 9 shows the power and thrust predictions of W^{2}A^{2}KE3D. Good agreement for velocities of 7-10 m/s is demonstrated for all flow solvers in comparison to experimental data. For velocities 11-15 m/s, delayed blade stalling is present in the overset simulations in comparison to the HELIOS solver and experimental data resulting in over prediction of the power and thrust. However, the W^{2}A^{2}KE3D results for power are significantly more accurate than the stand-alone OVERFLOW and NSU3D results.

Figure 10 shows the pressure coefficient at $30\%, 46.6\%, 63.3\%, 80\%$ and 95% span-wise stations of the blade for 7, 10, and 15 m/s uniform inflow velocities. In all inflow velocity cases at all sectional locations, the computed pressure coefficient values on the pressure side of the blade compare well with experimental data, as expected. On the suction suction side of the blade, good agreement with experimental data is observed at 7 m/s. However for higher inflow velocity cases, some of the experimental values show flat profiles indicative of blade stalling, while the computational results show higher suction peaks suggesting the flow remains attached. The discrepancy is most pronounced at 46.6% span for the 10 m/s inflow velocity case, and at the 30% span location for the 15 m/s inflow velocity case. Figure 11 shows span-wise slides of the computed pressure coefficient for 11 m/s illustrating the fact the the flow remains mostly attached at this condition.

IV.C. Siemens SWT-2.3-93

A generic Siemens SWT-2.3.93 turbine model using specifications from the IAE Wind Task 31-Wakebench [83] is simulated. The geometry of the turbine blade is constructed from multiple cylinder and airfoil sections. The wind turbine contains three blades and a tower with a nacelle for a total of four near-body meshes per wind turbine. The nominal rotor rotation speed is 16 RPM. The rated power inflow velocity is 10.9 m/s generating a rated electrical power of 2.3 MW. The Siemens blade has a radius of 46.5 m and a low-speed shaft title angle of 6°, pre-cone angle of 2.5°, and nominal blade pitch of -1°. The tower has a height of 65 m. To accurately predict the power and thrust forces, as indicated by the mesh refinement study of the NREL 5MW wind turbine, a near-body blade mesh containing 2,219,940 nodes and a tower with nacelle mesh containing 504,960 nodes are used. The global time step is set to a corresponding rotation of 1/4° and 25 non-linear sub-iterations are used to converge the BDF-2 time step for the near-body solver.

Our predicted power from the simulation framework at the nominal uniform inflow velocity of 10.9 m/s is 2.5 mega-watts and the torque is predicted at 373,000 Newtons which agrees well with the NREL FAST software (2.5 mega-watts is the aerodynamic power force before losses due to the generator which yields 2.3 mega-watts). The power and thrust convergence histories are shown in Figure 12. Figure 13 shows a volume rendering and an iso-surface visualization of the Siemens-2.3-93 wind turbine.

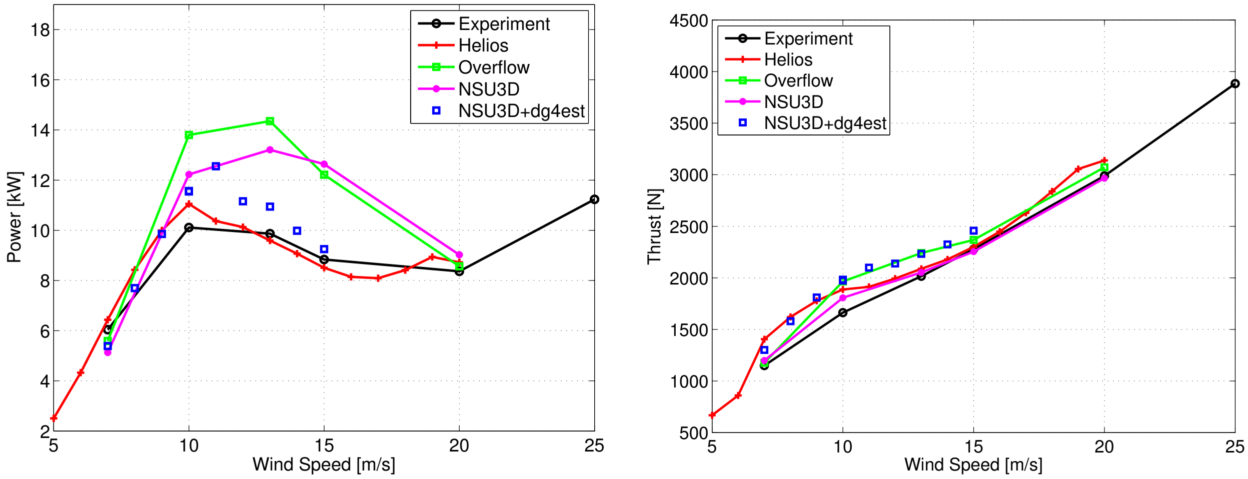


Figure 9. Simulation results for the NREL Phase VI power and thrust for uniform inflow velocities of 7-15 m/s. Results are compared to the experimental values along with other numerical simulations: NSU3D in stand-alone, CREATE-AV HELIOS [27], and NASA Overflow [82].

V. Results: Weak Scaling with Wind Plant Configurations

Parallel scalability of the computational framework is essential for enabling simulation of wind plants using full rotor models. This section is concerned with weak scalability which is defined as how the solution time varies with the number of processors for a fixed problem size *per* processor. To perform a weak scaling test in the context of a wind plant simulation, we assign a fixed number of processors per wind turbine. Then for each weak scaling sample, we increase the number of wind turbines simulated in a wind plant configuration along with the total number of cores used for the simulations. The ability to weak scale the wind plant simulation software is essential for simulating hundreds of wind turbines in a wind plant configuration.

In this study we perform the weak scaling test starting with six wind turbines using 348 CPU cores per turbine for the near-body solver and 120 CPU cores per turbine for the off-body solver. The weak scaling test evaluates the parallel weak scalability at 6, 12, 24, 48, and 96 wind turbines over a 9.5 hour wall-clock time simulation window. The total number of CPU cores used ranges from 2,808 to 44,928.

The wind turbine chosen for the weak scaling study is the Siemens SWT-2.3-93. As demonstrated from the single wind turbine performance study, the required mesh resolution to accurately capture the aerodynamic forces uses just over 2.2 million nodes per blade. Allocating 108 CPU cores per blade and 24 CPU cores per tower equates to 20,555 and 21,040 nodes per blade and tower, respectively, for a total of 348 CPU cores per wind turbine.

The weak scaling study is performed on the NSF NWSC-2 Cheyenne supercomputer [84]. Cheyenne contains 145,152 Intel Xeon E5-2697V4 processor cores rated at 2.6 GHz. The Intel Xeon E5-2697V4 processor uses the AVX-2 instruction set allowing for four double precision operations to be performed in single-instruction-multiple-data (SIMD) parallelism. Cheyenne contains 4,032 compute nodes and with two processors per node totaling 36 CPU cores per compute node. The total theoretical peak performance of Cheyenne is 5.34 petaflops. The network is a Partial 9D Enhanced Hypercube single-plane interconnect topology with Mellanox EDR InfiniBand high-speed interconnect.

Two initial challenges that limited weak parallel scalability have been addressed. The first scaling issue arose for mesh domain intersection checking. In the overset framework, all mesh partitions are assigned an alternating digit tree (ADT) for efficient domain searching. The oriented bounding box of the ADTs are sent in an all-to-all communication to check for intersections. However, `p4est` partitions cells into non-contiguous groups via *z-order* partitioning [64]. This can cause elements on opposite sides of a computational domain to be placed within the same bounding box, resulting in large bounding boxes. When the overset assembler performs an intersection check with this non-contiguous partition, large amounts of intersections are found. To address this issue, all near-body bounding boxes are communicated to all off-body processors and a bounding-box-to-cell intersection check is performed. This check uses the efficient octree search built into the AMR `p4est` framework. Using these intersection results, a processor communication map is constructed.

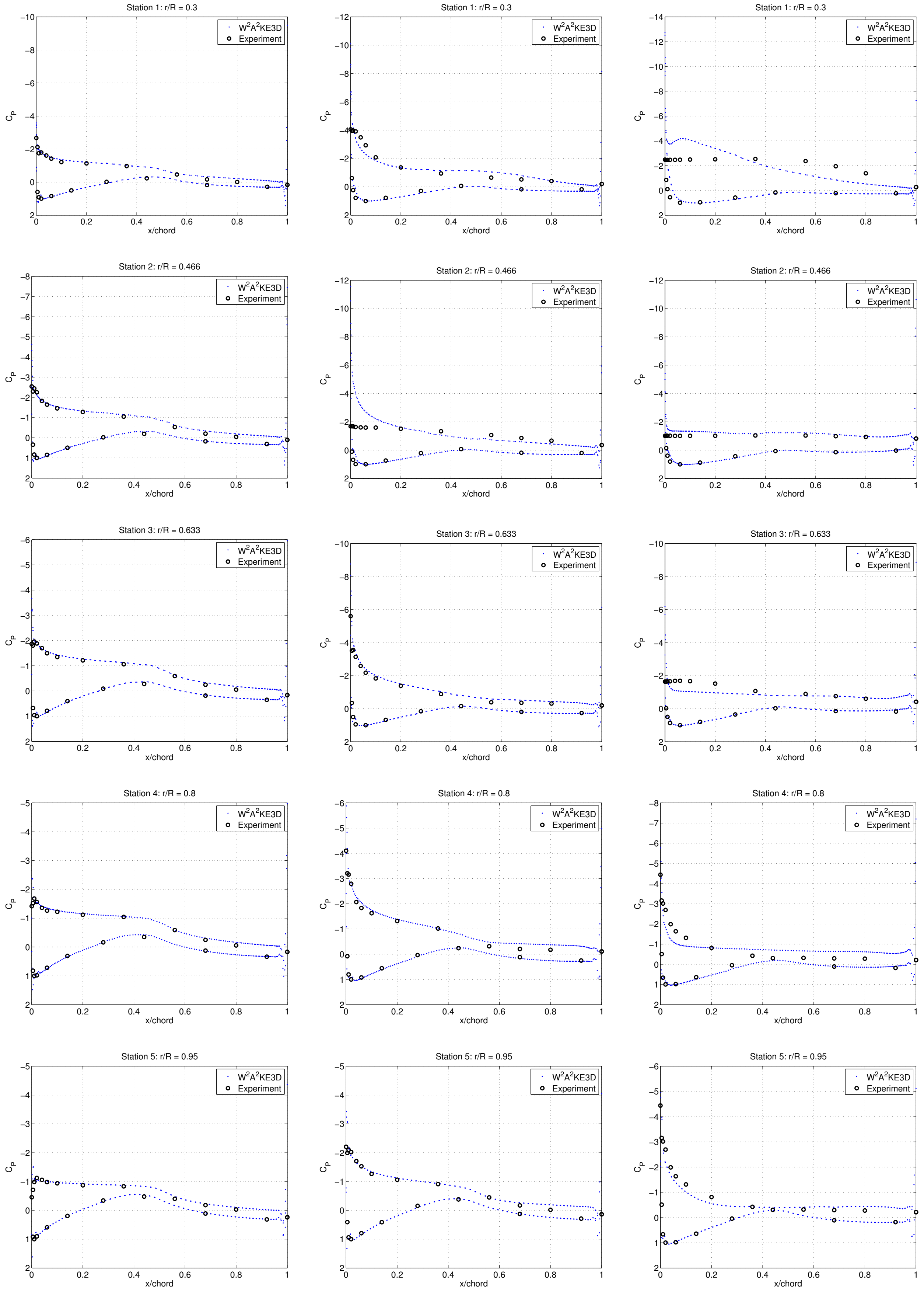


Figure 10. NREL Phase VI pressure coefficients at 30%, 46.6%, 63.3%, 80%, 95% span-wise stations for 7 m/s (column 1), 10 m/s (column 2), and 15 m/s (column 3) uniform axial inflow velocities. Predicted results of W^{2}A^{2}KE3D are plotted versus the experimental data.

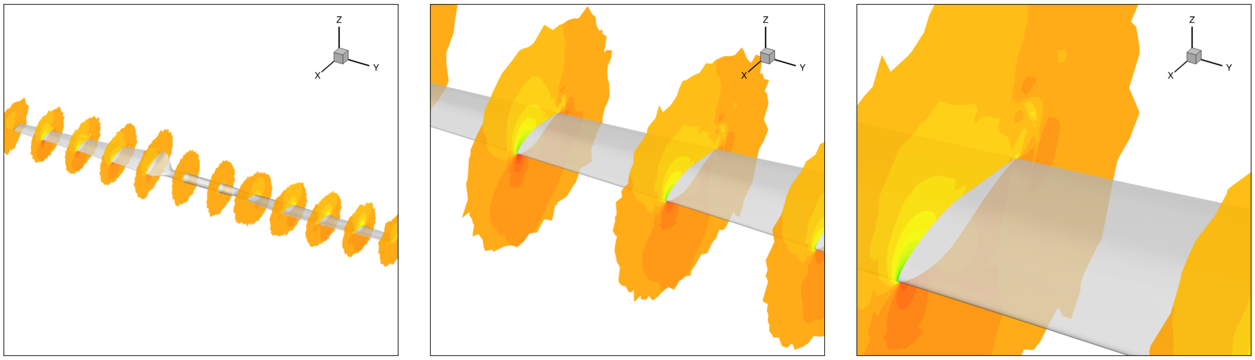


Figure 11. NREL Phase VI coefficient of pressure visualized for 11 m/s inflow velocity.

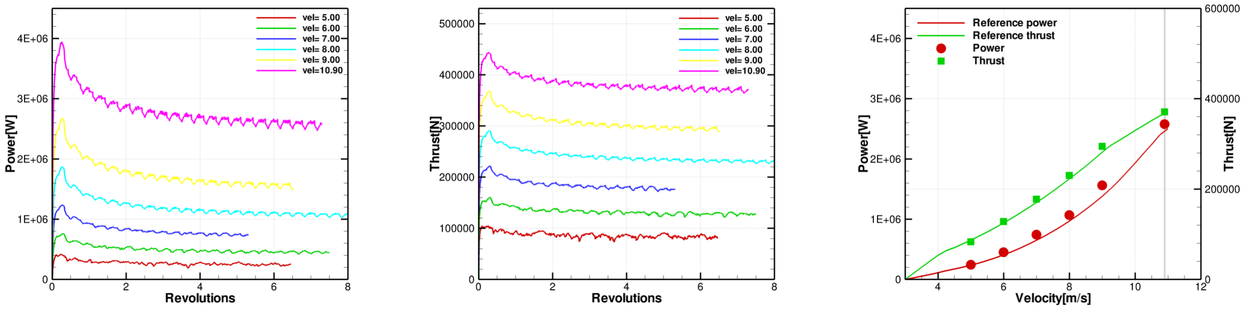
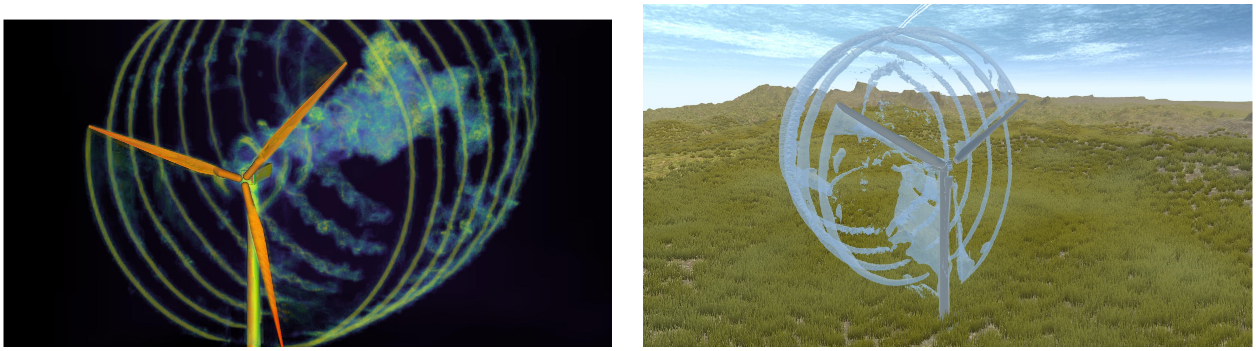


Figure 12. Siemens SWT-2.3-93 power and thrust simulation results using a time step corresponding to a 1/4° rotation. Each time was solved with BDF-2 using 25 sub-iterations for the near-body flow solver. Reference solution data provided by the NREL FAST [77] software.



(a) Volume rendering of vorticity.

(b) Iso-surface of velocity magnitude.

Figure 13. Siemens SWT-2.3-93 wind turbine used for the weak scaling study of the W^{2}A^{2}KE3D framework.

The second scaling issue arose from inter-grid boundary points (IGBPs). IGBPs are points that used to connect the off-body mesh system to the near-body meshes. These points are the outer mesh points located on or near the surface of the trimmed near-body meshes. IGBPs locations and corresponding element sizes are communicated from the near-body meshes to the off-body solver so that the off-body AMR mesh can be adaptively refined to these locations for the meshes to exchange solution data correctly. Since the near-body mesh is moving through the off-body mesh and the off-body mesh is adapting or repartitioning after every global time step, the original algorithm sent a list containing all IGBPs globally to all processors. This caused scaling issues at large core counts particularly when many meshes were used. The global list of IGPBs became substantially large, therefore increasing the communication cost. Additionally, since each list was not unique, searching of the list became costly, therefore creating a bottleneck in the regridding process of the off-body mesh. To address this issue, the same processor map used for the bounding box intersection check is used to reduce the number of IGPBs communicated and to make the IGBP list unique to each off-body processor.

Table 2 shows the present performance statistics for the weak scaling study. The results assume that the efficiency of the six wind turbine simulation is perfect as a reference value. When doubling the number of turbines successively, the parallel scalability efficiency is 98.7% for 12 turbines, 96.8% for 24 turbines, and 93.3% for 48 turbines. The weak scalability decreases slightly in performance when simulating 96 wind turbines giving an efficiency of 86.9%.

Table 3 displays solver specific timings for each of the wind plant configurations. The blade, tower, and off-body times correspond to the CFD solver times. The various component meshes of the CFD solvers are all run in parallel. However, the overset connectivity determination is performed on all processors at the end of each time step executed by the CFD solvers. Thus the total wall-clock time for each complete time step corresponds to the sum of the maximum CFD solver time for a time step and the overset connectivity time. The blade time corresponds to the near-body blade mesh and solver that is replicated three times for each wind turbine. The run times for the blade and tower meshes are on average constant for all wind plant configurations. This is expected since each near-body mesh is independent of each other and the computational work remains constant for the duration of simulations. Each near-body mesh uses a new instance of the near-body solver, thus decoupling the near-body flow meshes. The off-body solve times slightly increase from 6 to 12 turbines then to 24, 48, and 96 wind turbines. The larger wind plant configuration run times become approximately constant. The off-body solver uses only one instance of the off-body flow solver. Thus the weak scalability of the AMR framework p4est is demonstrated. Notice the average solve time at 48 wind turbines is the same at 96 wind turbines for the off-body solver.

Weak Scaling Wind Farm Study: Overall Performance Statistics

Turbine Count	Efficiency	Revs	Near-Body Cores	Off-Body Cores	Total Cores
6	1.0000	1.374	2,088	720	2,808
12	0.9874	1.360	4,176	1,440	5,616
24	0.9682	1.331	8,352	2,880	11,232
48	0.9333	1.283	16,704	5,760	22,464
96	0.8686	1.194	33,408	11,520	44,928

Table 2. Weak scaling wind plant study performed on NWSC-2 Cheyenne [84] up to 96 wind turbines for wall-clock time of 9.5 hours. Six turbines are used as the perfect scaling reference.

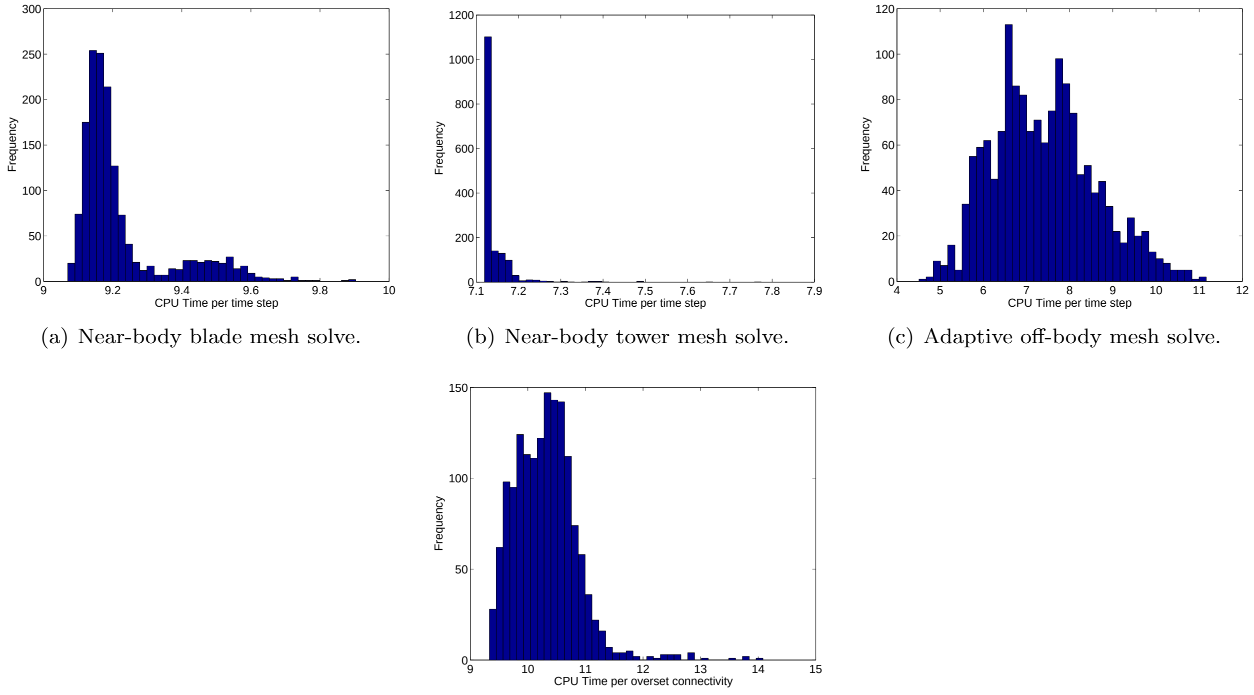
Weak Scaling Wind Farm Study: Solver Performance Statistics

Turbine	Blade Time (s)			Tower Time (s)			Off-body Time (s)			Overset Time (s)		
Count	min	max	avg	min	max	avg	min	max	avg	min	max	avg
6	8.944	9.588	9.067	7.111	7.905	7.135	4.299	9.455	7.027	7.623	8.558	8.056
12	8.949	9.347	9.075	7.126	7.632	7.148	4.310	10.11	7.152	7.684	8.632	8.141
24	8.980	9.931	9.178	7.124	7.721	7.208	4.180	11.29	7.261	7.842	9.295	8.314
48	8.996	10.00	9.224	7.147	7.974	7.243	4.203	11.29	7.428	8.056	10.89	8.613
96	9.069	9.903	9.225	7.119	7.774	7.143	4.511	11.16	7.406	9.332	14.08	10.32

Table 3. Weak scaling wind plant study solver times up to 96 wind turbines.

For deeper analysis of the 96 wind turbine case, Figure 14 shows the distribution of solver times per time step. Row 1 shows the frequency of the CFD solver execution times over the entire run. As seen in Figure 14(a), the near-body solve time for the blade mesh has a fairly wide distribution ranging from 9 seconds to 10 seconds. The wide distribution can be attributed to I/O of log files. The near-body solver, which was composed of 288 solver instances for the blades, logged large amounts of solver data to a single output log file. This causes a bottleneck in the I/O therefore slowing down the execution time even though the blade meshes are independent of each other with fixed degrees of freedom. As demonstrated in Figure 14(b), the near-body mesh solve time frequencies better demonstrate the independence of the near-body meshes where the distribution is very tight. Figure 14(c) shows the distribution of the off-body solve times. The frequency distribution is expected to have a wide base since the computation work changes throughout the duration of the simulation caused by dynamic mesh adaption. As the solution evolves, the flow features increase therefore requiring more mesh resolution which induces longer solve times. Lastly, the overset data update and grid connectivity times are shown in Figure 14(d). The time frequencies portray a skewed distribution ranging from 9.2 seconds to 11.5 seconds. However, there are a number of solve times that grow to 14 seconds.

The smaller wind plant configurations demonstrate good performance for all components of the software but at larger wind turbine counts 48 and, particularly, 96, the solver time distribution width of the overset module widens. This indicates a small number of ranks in the large scale simulations may be throttling the overall performance. In general, overset methods incur larger scalability issues than flow solvers because of the inherent imbalance in amount of searches that need to be performed. To alleviate this issue, active load balancing techniques similar to those implemented in reference [70] need to be included and is planned as part of future work. Furthermore, overall efficiency improvement of the overset grid module is also desired since the execution time for dynamic overset grid assembly are on par currently with flow solution time, while more efficient approaches [70] have demonstrated overset grid assembly to take only one order of magnitude less time compared to flow solution time.



(a) Near-body blade mesh solve.

(b) Near-body tower mesh solve.

(c) Adaptive off-body mesh solve.

(d) Overset update and connectivity.

Figure 14. Solver time (seconds) frequency histograms of the 96 wind turbine case for the weak scaling study. Row 1 shows the CFD solver times which run in parallel; row 2 shows the overset data update and connectivity times. The CFD solvers must complete the time step before the overset module can interpolate the solutions between meshes therefore placing the execution process into two serial components.

V.A. Long Run-Time and Large Scale Wind Plant Simulations

A longer physical time simulation using the 48 wind turbine Lillgrund Wind Farm is simulated to 12 revolutions. The Lillgrund wind plant uses the Siemens SWT-2.3-93 wind turbine. The Lillgrund Wind Farm contains 48 wind turbines in an arrangement with downstream spacing of 4.3 diameters of the rotor and 3.3 diameters of side spacing. Uniform inflow conditions are used with a velocity of 10.9 m/s. The rotation rate of the rotor is taken as 16 RPM. Figure 15 shows the wind plant configuration with iso-surfaces of velocity magnitude at approximately eight revolutions of rotation. Figure 16 portrays a velocity magnitude slice for a row of the Lillgrund Wind Farm and a profile of the adaptive mesh refinement pattern in the wake of a wind turbine. The wake structure is tracked well down-stream by the use of adaptive meshes.

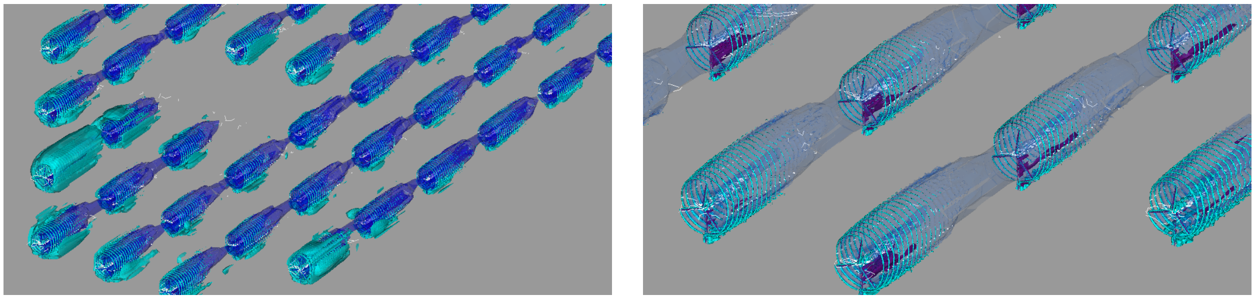
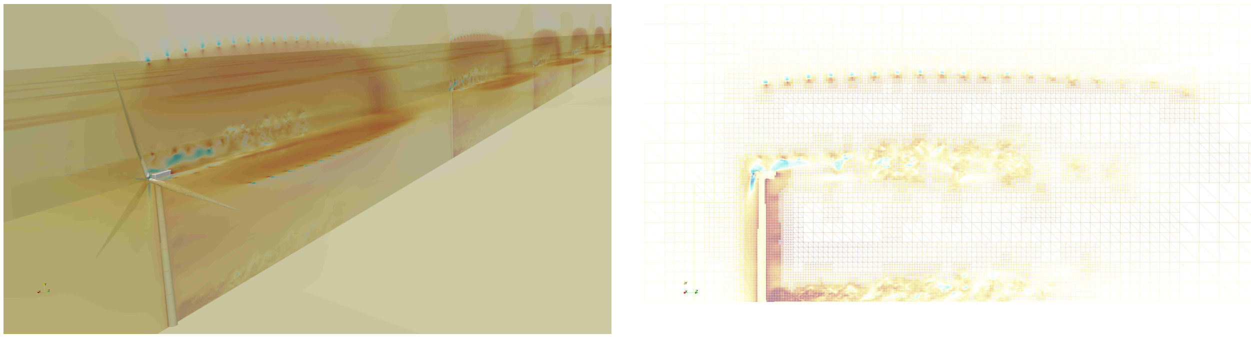


Figure 15. Iso-surfaces of velocity magnitude of the Lillgrund Wind Farm which contains 48 Siemens SWT-2.3-93 wind turbines.



(a) Velocity magnitude of a row of Lillgrund wind turbines.

(b) Adaptive mesh system of the wake of a Siemens turbine.

Figure 16. Lillgrund Wind Farm wake structures and adaptive mesh for the Siemens SWT-2.3-93 wind turbine.

Figure 17 demonstrates the evolution of the degrees of freedom for the off-body adaptive flow solver. Three linear trends are noticed in the DOFs. The initial cost of connecting the off-body mesh to all 48 near-body wind turbine meshes (four meshes per turbine) is approximately 300 million DOFs. From the start of the simulation to five revolutions, the DOFs sharply increase to approximately 1.2 billions degrees of freedom in a linear fashion representing the initial wake transients. The second linear trend represents the sustained wake growth as the simulation evolves over time. For wind turbine wake interaction to occur between upstream and downstream wind turbines with a spacing of 4.3 rotor diameters, inflow velocity of 10.9 m/s at 16 RPM, approximately 36 seconds of physical time simulation are required. This corresponds to 9.78 revolutions of rotation which is exactly the location of the peak of DOFs in Figure 17. The decreasing linear trend represents the time after which the wakes begin to interact. Under uniform inflow conditions, strong blade tip vorticies are formed invoking mesh refinement as demonstrated in Figure 16(b). Flow features reaching a user-specified threshold of Q-criterion magnitude, which is a measure of vorticity and mean-shear rate, are tagged for mesh refinement. Wake velocity deficits generated by uniform inflow conditions are much larger in comparison to turbulent inflow conditions because there is less entrainment of momentum by turbulent mixing. The reduced inflow velocities for downstream turbines generate weaker blade tip vorticies compared to wind turbines that do not have impinging wake inflow conditions resulting is less elements containing flow features that reach the refinement criterion threshold value. After the wakes impinge on the downstream wind turbines, less elements are tagged for refinement resulting in the observed decrease of DOFs.

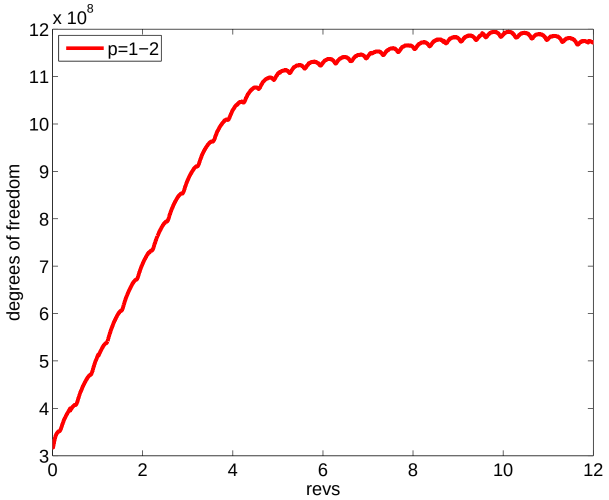


Figure 17. Degree of freedom counts for Lillgrund Wind Farm simulation. The initial linear trend corresponds to the start-up wake transients. The second linear trend corresponds to the sustained wake growth over the duration of the simulation. The last linear trend represents the interaction of the wakes between wind turbines. The peak represents the moment when the upstream wind turbine wake interacts with the downstream wind turbine.

A short run-time simulation using 144 wind turbines is also performed to stress test the parallel scalability. This simulation used the same 2.22 million node blade mesh with the 505,000 node tower mesh totaling roughly 1 billion degrees of freedom for the near-body mesh system. The off-body mesh contained 1.6 billion degrees of freedom using $p = 4$, fifth-order spacial accuracy elements giving a cumulative total of 2.6 billion DOFs across 62,208 processors. The simulation was only executed for a small number of time steps but demonstrated the ability to run 577 independent meshes in an overset environment. Future work involves improving weak and strong scalability of the framework. At such large scale with hundreds to thousands of solver instances in a single simulation, detailed concerns such as I/O of individual log files are required.

VI. Results: Atmospheric Inflow Conditions

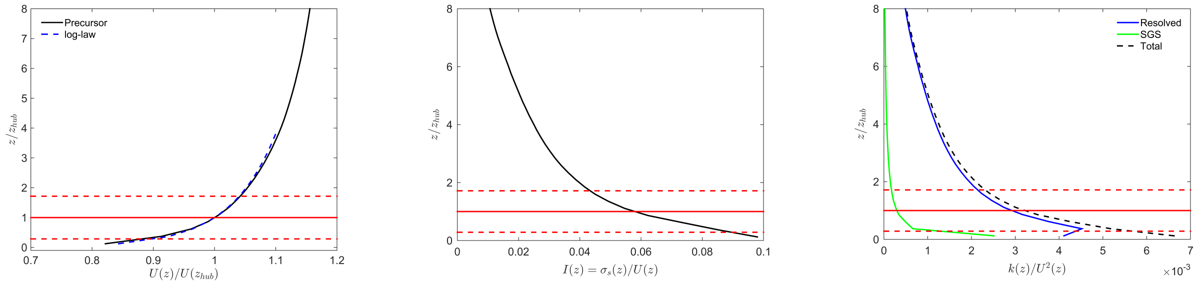
Accurate representation of the atmospheric inflow conditions are required for proper wind plant CFD analysis. To perform these tasks, the W^{2}A^{2}KE3D framework has incorporated a meteorological micro-scale flow solver interface. This interface serves as a one-way coupling from the micro-scale atmospheric solver to the CFD through the same overset domain connectivity and data transfer. A micro-scale atmospheric solution is registered as a pseudo-CFD solver with an unstructured grid to TIOGA. Prior to the CFD simulation, a precursor atmospheric simulation is performed in problem dependent situations such as specific atmospheric conditions, e.g. turbulence intensity, stable, neutral, unstable boundary layer, etc., and for specific complex terrain environments such as a specific geographical location. This precursor simulation is run until statistically converged flow statistics are achieved. When this is complete, a time history of flow solutions are written to disk for a specified duration of physical simulation time. When the CFD simulation is initialized, all initial flow variables in the near-body and off-body mesh system are filled from the atmospheric data. During a wind plant simulation, the boundary elements of the off-body mesh system are updated via linear-interpolated-time atmospheric data.

VI.A. SOWFA Precursor Results for Neutral ABL

The atmospheric inflow for the Lillgrund Wind Farm is based on the meteorological conditions described in Bergström et al. [85] and in the large eddy simulation performed by Churchfield et al. [86]. A neutral atmospheric boundary layer (ABL) is assumed with a mean hub-height velocity of 9 m/s from a direction of 221.6° and a surface aerodynamic roughness value of $Z_0 = 10^{-4}$ m is chosen to reproduce the hub-height turbulence intensity of about 6% [85].

The precursor LES domain size is 10,240 m × 4,096 m × 1,024 m with a uniform 16 m resolution in all directions resulting in a mesh consisting of 640 × 256 × 64 hexahedral cells. Periodic boundary conditions are applied in the wind-wise and cross-stream directions and a slip-wall is used at the top boundary of the domain. A capping inversion of 100 meters at 700-800 m is applied to limit the boundary layer growth. The initial potential temperature field is kept uniform at 300 K from the surface to 700 m and within the capping inversion the potential temperature rises by 8 K. Above 800 m, the potential temperature gradually increases at a rate of 0.003 K/m. This potential temperature profile is similar to that used by Churchfield et al. [86] and Moeng and Sullivan [87]. The initial velocity profile is approximated using a log-law of the wall and small perturbations are added near the surface to promote transition to a turbulent flow. The atmospheric boundary layer is simulated for 12,000 seconds to allow the initial transients to pass and achieve a quasi-equilibrium state and then run for an additional 3,000 seconds to record the velocity field at each time step which are to be coupled with W^{2}A^{2}KE3D .

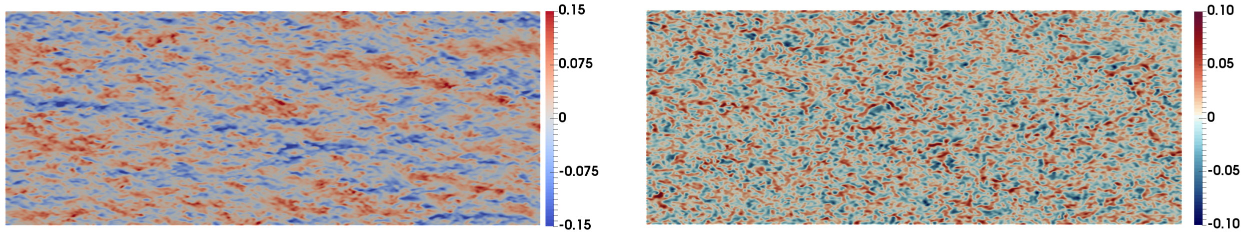
Figure 18 depicts vertical profiles of the temporally and horizontally averaged velocity, turbulence intensity and turbulence kinetic energy profile for the lower part of the ABL. The mean velocity closely follows the log-law (based on the specified rotor hub-height velocity and surface roughness) near the surface. Although this simulation contains no rotor, the target wind turbine rotor for this configuration would experience a significant mean wind shear of 1.3 m/s across the rotor diameter. The turbulence intensity is largest near the surface at approximately 10% and decreases to the desired 6% at the hub-height. In the precursor LES, the turbulence kinetic energy is mostly resolved and only a small portion is modeled by the SGS model with the exception of the first two cell levels where the modeled contribution is significant. This is a common problem of all neutral ABL LES irrespective of the resolution since the turbulence integral length scale is proportional to the distance from the surface in the log-law region and thus in the first few cells above the surface the turbulence length scale and the filter scale are comparable.



(a) Mean wind profile. (b) Turbulence intensity profile. (c) Turbulence kinetic energy profile.

Figure 18. Vertical profiles of temporally and horizontally averaged velocity, turbulence intensity and turbulence kinetic energy from the precursor LES. The solid red horizontal line represents the hub height and the two horizontal dashed lines represent the vertical extent of the wind turbine rotor.

Figure 19 illustrates contours of instantaneous normalized velocity fluctuation in a horizontal plane at the rotor hub-height. These contour plots reveal the presence of a wide range of scales in the turbulent boundary layer. The plots indicate the presence of turbulence structures that are very large (order of several kilometers) in the wind-wise direction. The existence of these large structures motivate the need for large domain sizes considered in LES of the atmospheric boundary layer. If the simulation domain is size is too small, the adopted periodic boundary conditions would artificially lock the elongated structures in place and thereby produce a spatially biased inflow condition for the wind plant CFD simulation.



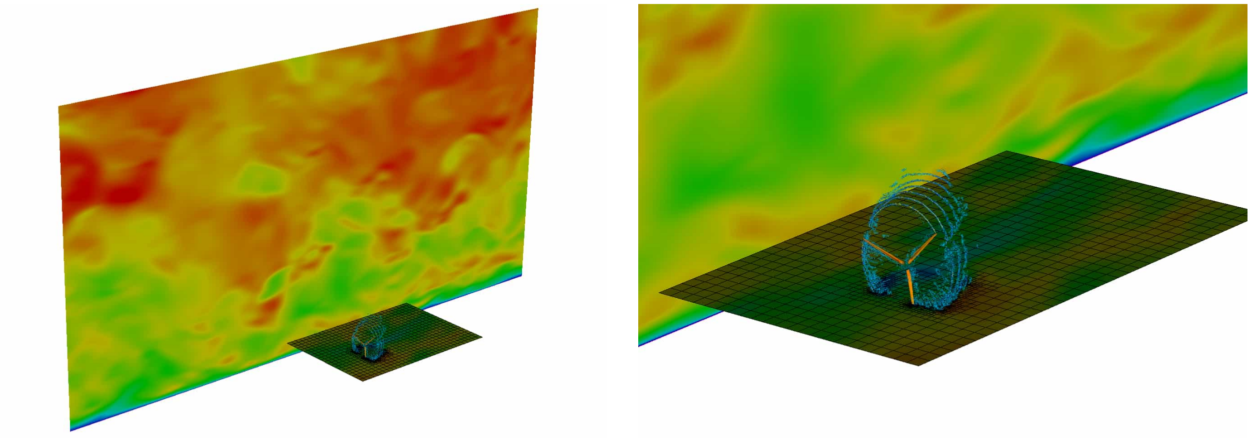
(a) Wind-wise velocity fluctuations. (b) Vertical velocity fluctuations.

Figure 19. Contours of instantaneous velocity fluctuation at rotor hub height horizontal plane of precursor LES velocity normalized by mean wind speed.

VI.B. Coupled Micro-scale Atmospheric and CFD Results

SOWFA will be the primary ABL LES solver used in this work to produce the inflow for the wind plant CFD simulations. SOWFA enables simulations of arbitrarily complex terrain through the use of unstructured grids and a terrain aligned implementation of the Schumann-Grötzbach [88, 89] wall model. First results for real complex terrain at the Sierra Madre site in south-central Wyoming and the Bolund hill [73–75] are indeed promising. To demonstrate the versatility of the multiple solver paradigm in W^2A^2KE3D , we have incorporated SOWFA inflow as well as NCAR's WRF-LES [72] inflow.

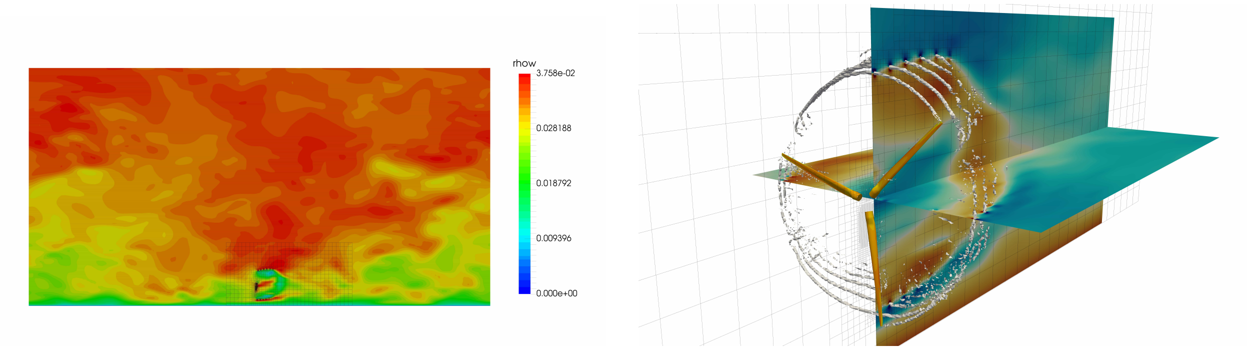
Preliminary work of coupling WRF and SOWFA with the blade resolved wind plant simulation code is shown in Figure 20 and Figure 21, respectively. In particular, Figure 21 depicts the simulation of a single Siemens turbine coupled to the precursor SOWFA calculation described in the previous section. As seen in the figure, atmospheric inflow conditions break down wake structures much faster than uniform inflow conditions. Simulations using uniform inflow significantly under predicts the energy produced by turbines that are in the wake of other turbines [28]. This is due to the inability to entrain momentum through the lack of turbulent mixing. Further work on validation for non-complex terrain will be performed in the near-future to ensure correct cumulative energy production prediction by a cluster of turbines. Once completed, complex terrain will be introduced with atmospheric inflow conditions to emulate real environments of wind farms.



(a) NCAR WRF atmospheric inflow coupling to the off-body solver.

(b) Zoomed view of the NCAR WRF inflow.

Figure 20. Micro-scale atmospheric and CFD coupling with NCAR's WRF solver to the off-body CFD solver `dg4est` for a single NREL 5MW wind turbine.



(a) NREL SOWFA atmospheric inflow coupling to the off-body solver.

(b) Zoomed view of the NREL SOWFA inflow.

Figure 21. Micro-scale atmospheric and CFD coupling with NREL's SOWFA solver to the off-body CFD solver `dg4est` for a single NREL 5MW wind turbine.

VII. Conclusion

A computational framework was presented as a potential solution to accurately simulating a wind plant containing on the order of 100 wind turbines using blade resolved models for turbine blades and towers. Validation studies were performed for three wind turbines, the NREL 5MW, NREL Phase VI, and Siemens SWT-2.3-93 wind turbines, with good agreement in power and thrust values with experimental data and other simulation software. We emphasize the importance of mesh resolution, time step size, and implicit system convergence levels required for accurately capturing the flow physics required for attaining accurate force predictions. The use of coarse meshes may provide tractability for simulating full size wind plants but suffers greatly in providing accurate power and thrust predictions. This will significantly impact the overall wind plant efficiency prediction along with prediction of structural forces on individual wind turbines.

A weak scaling study was performed using wind plant configurations comprising 6, 12, 24, 48, and 96 wind turbines. The results indicated that $\mathrm{W}^2\mathrm{A}^2\mathrm{KE3D}$ is capable of simulating approximately 100 wind turbines effectively with moderate amounts of computational resources. The framework supported longer run time simulations of the Lillgrund Wind Farm with degrees of freedom growing to over 1.3 billion for 12 revolutions. Additionally, the computational framework was shown to work on a wind plant simulation containing 144 wind turbines which equates to 577 independent meshes registered to the overset domain connectivity and data update, with over 2.6 billion degrees of freedom. Lastly, preliminary results of the atmospheric coupling with micro-scale solvers WRF and SOWFA indicate promising directions for faithfully representing the turbulent inflow conditions and complex terrain environments.

Future work involves extensive validation studies including work to improve flow regimes containing flow separation. Wake structure and wake deficit analysis will be investigated as these are essential to the power characteristics of wind plants. Research into improved computational efficiency and strong scalability of the overset framework will be conducted to enable long physical run times. Target physical run times will be on the order of minutes. Development of atmospheric inflow containing complex terrain will be implemented and validated. Validation of additional physics such as Coriolis and buoyancy forces of the off-body solver will be performed. Small scale turbulence structures will be implemented into the off-body solver to introduce fine grid atmospheric turbulence appended to large scale atmospheric structures provided by the micro-scale flow solver. Transition to higher-order solution accuracy of the off-body flow solver will be a target as higher-order solutions are better suited for newer computer architectures containing heterogeneous environments.

VIII. Acknowledgements

The authors would like to thank Earl Duque and Brad Whitlock from Intelligent Light for their support in instrumenting the in-situ visualization library Libsim into the computational framework along with providing visualizations. Additionally, we would like to thank Kyle Summerfield, Nikhil Shetty, and Emma-Jane Alexander of the Shell 3-D Visualization Center in the School of Energy Resources at the University of Wyoming for providing visualization development and support. Computer time was provided by the NCAR-Wyoming Supercomputer Center (NWSC), the University of Wyoming Advanced Research Computing Center (ARCC), and the NSF Blue Waters sustained-petascale computing project. This research was conducted with an Accelerated Science Discovery (ASD) project providing early access to the NWSC-2 Cheyenne supercomputer.

This work is supported in part by ONR Grants N00014-14-1-0045 and N00014-16-1-2737 and by the U.S. Department of Energy, Office of Science, Basic Energy Sciences, under Award DE-SC0012671. The first author was supported in part by the NSF Blue Waters Graduate Fellowship as part of the Blue Waters sustained-petascale computing project, which is supported by the National Science Foundation (awards OCI-0725070 and ACI-1238993).

References

[1] "Wind Vision: A New Era for Wind Power in the United States," Tech. rep., Technical report, US Department of Energy, Washington, DC, 2015.

[2] Messina, P., "Exascale Computing Project," 2016, Exascale Computing Project (17-SC-20-SC), a collaborative effort of the U.S. Department of Energy Office of Science and the National Nuclear Security Administration.

[3] "Turbine Wind Plant Efficiency," 2016.

[4] "NREL to Lead One Exascale Computing Project," 2016.

[5] Troldborg, N., Srensen, J. N., and Mikkelsen, R., "Actuator Line Simulation of Wake of Wind Turbine Operating in Turbulent Inflow," *Journal of Physics: Conference Series*, Vol. 75, No. 1, 2007, pp. 012063.

[6] Churchfield, M., Lee, S., and Moriarty, P., "Overview of the Simulator for Offshore Wind Farm Application SOWFA," 2012.

[7] Fleming, P. A., Gebraad, P. M., Lee, S., van Wingerden, J.-W., Johnson, K., Churchfield, M., Michalakes, J., Spalart, P., and Moriarty, P., "Evaluating techniques for redirecting turbine wakes using SOWFA," *Renewable Energy*, Vol. 70, 2014, pp. 211–218.

[8] Churchfield, M., Wang, Q., Scholbrock, A., Herges, T., Mikkelsen, T., and Sjöholm, M., "Using high-fidelity computational fluid dynamics to help design a wind turbine wake measurement experiment," *Journal of Physics: Conference Series*, Vol. 753, IOP Publishing, 2016, p. 032009.

[9] Mikkelsen, R., *Actuator disc methods applied to wind turbines*, Ph.D. thesis, Technical University of Denmark, 2003.

[10] Takizawa, K., Henicke, B., Tezduyar, T. E., Hsu, M.-C., and Bazilevs, Y., "Stabilized space–time computation of wind-turbine rotor aerodynamics," *Computational Mechanics*, Vol. 48, No. 3, 2011, pp. 333–344.

[11] Bazilevs, Y., Hsu, M.-C., Akkerman, I., Wright, S., Takizawa, K., Henicke, B., Spielman, T., and Tezduyar, T., "3D simulation of wind turbine rotors at full scale. Part I: Geometry modeling and aerodynamics," *International Journal for Numerical Methods in Fluids*, Vol. 65, No. 1-3, 2011, pp. 207–235.

[12] Potsdam, M. A. and Mavriplis, D. J., "Unstructured Mesh CFD Aerodynamic Analysis of the NREL Phase VI Rotor," AIAA Paper 2009-1221, 47th AIAA Aerospace Sciences Meeting, Orlando, FL, January 2009.

[13] Sørensen, N. N., Michelsen, J., and Schreck, S., "Navier-Stokes predictions of the NREL phase VI rotor in the NASA Ames 80 ft× 120 ft wind tunnel," *Wind Energy*, Vol. 5, No. 2-3, 2002, pp. 151–169.

[14] Duque, E. P., Burklund, M. D., and Johnson, W., "Navier-Stokes and comprehensive analysis performance predictions of the NREL phase VI experiment," *Journal of Solar Energy Engineering*, Vol. 125, No. 4, 2003, pp. 457–467.

[15] Pape, A. L. and Lecanu, J., "3D Navier–Stokes computations of a stall-regulated wind turbine," *Wind Energy*, Vol. 7, No. 4, 2004, pp. 309–324.

[16] Gomez-Iradi, S., Steijl, R., and Barakos, G., "Development and Validation of a CFD Technique for the Aerodynamic Analysis of HAWT," *Journal of Solar Energy Engineering*, Vol. 131, No. 3, 2009, pp. 031009.

[17] Zahle, F., Sørensen, N. N., and Johansen, J., "Wind turbine rotor-tower interaction using an incompressible overset grid method," *Wind Energy*, Vol. 12, No. 6, 2009, pp. 594–619.

[18] Bazilevs, Y., Hsu, M.-C., Kiendl, J., Wüchner, R., and Bletzinger, K.-U., "3D simulation of wind turbine rotors at full scale. Part II: Fluid–structure interaction modeling with composite blades," *International Journal for Numerical Methods in Fluids*, Vol. 65, No. 1-3, 2011, pp. 236–253.

[19] Gundling, C., Roget, B., and Sitaraman, J., "Prediction of Wind Turbine Performance and Wake Losses using Analysis Methods of Incremental Complexity," AIAA Paper 2011-458, 49th AIAA Aerospace Sciences Meeting, Orlando, FL, January 2011.

[20] Gundling, C., Roget, B., Sitaraman, J., and Rai, R., "Comparison of Wind Turbine Wakes in Steady and Turbulent Inflow," AIAA Paper 2012-899, 50th AIAA Aerospace Sciences Meeting, Nashville, TN, January 2012.

[21] Rai, R. K., Gopalan, H., Naughton, J. W., and Heinz, S., "A Study of the Sensitivity of Wind Turbine Response to Inflow Temporal and Spatial Scales," 2012.

[22] Li, Y., Paik, K.-J., Xing, T., and Carrica, P. M., "Dynamic overset CFD simulations of wind turbine aerodynamics," *Renewable Energy*, Vol. 37, No. 1, 2012, pp. 285–298.

[23] Yelmule, M. M. and Vsj, E. A., "CFD predictions of NREL phase VI rotor experiments in NASA/AMES wind tunnel," *International Journal of Renewable Energy Research (IJRER)*, Vol. 3, No. 2, 2013, pp. 261–269.

[24] Gopalan, H., Gundling, C., Brown, K., Roget, B., Sitaraman, J., Mirocha, J. D., and Miller, W. O., "A coupled mesoscale–microscale framework for wind resource estimation and farm aerodynamics," *Journal of Wind Engineering and Industrial Aerodynamics*, Vol. 132, 2014, pp. 13–26.

[25] "Top 500, The List: November 2016," 2016, https://www.top500.org/lists/2016/11/.

[26] Wissink, A. M., Potsdam, M., Sankaran, V., Sitaraman, J., and Mavriplis, D., "A Dual-Mesh Unstructured Adaptive Cartesian Computational Fluid Dynamics Approach for Hover Prediction," *Journal of the American Helicopter Society*, Vol. 61, No. 1, 2016, pp. 1–19.

[27] Wissink, A. M., Jayaraman, B., and Sitaraman, J., "An Assessment of the Dual Mesh Paradigm Using Different Near-Body Solvers in Helios," *55th AIAA Aerospace Sciences Meeting*, 2017, p. 0287.

[28] Sitaraman, J., Mavriplis, D. J., and Duque, E. P., "Wind farm simulations using a full rotor model for wind turbines," *32nd ASME Wind Energy Symposium*, 2014, p. 1086.

[29] Gundling, C., Sitaraman, J., Roget, B., and Masarati, P., "Application and validation of incrementally complex models for wind turbine aerodynamics, isolated wind turbine in uniform inflow conditions," *Wind Energy*, Vol. 18, No. 11, 2015, pp. 1893–1916.

[30] Steger, J. L., Dougherty, F. C., and Benek, J. A., "A chimera grid scheme.[multiple overset body-conforming mesh system for finite difference adaptation to complex aircraft configurations]," 1983.

[31] Noack, R., "SUGGAR: a general capability for moving body overset grid assembly," *17th AIAA Computational Fluid Dynamics Conference*, 2005, p. 5117.

[32] Brazell, M. J., Sitaraman, J., and Mavriplis, D. J., "An overset mesh approach for 3D mixed element high-order discretizations," *Journal of Computational Physics*, Vol. 322, 2016, pp. 33–51.

[33] Crabill, J. A., Sitaraman, J., and Jameson, A., "A High-Order Overset Method on Moving and Deforming Grids," *AIAA Modeling and Simulation Technologies Conference*, 2016, p. 3225.

[34] Vincent, P., Witherden, F. D., Farrington, A. M., Ntemos, G., Vermeire, B. C., Park, J. S., and Iyer, A. S., "PyFR: Next-Generation High-Order Computational Fluid Dynamics on Many-Core Hardware," *22nd AIAA Computational Fluid Dynamics Conference*, 2015, p. 3050.

[35] Witherden, F. D., Vermeire, B. C., and Vincent, P. E., "Heterogeneous computing on mixed unstructured grids with PyFR," *Computers & Fluids*, Vol. 120, 2015, pp. 173–186.

[36] Cockburn, B., Karniadakis, G. E., and Shu, C.-W., *The development of discontinuous Galerkin methods*, Springer, 2000.

[37] Fidkowski, K. J., *A high-order discontinuous Galerkin multigrid solver for aerodynamic applications*, Ph.D. thesis, Massachusetts Institute of Technology, 2004.

[38] Luo, H., Baum, J. D., and Löhner, R., "A discontinuous Galerkin method based on a Taylor basis for the compressible flows on arbitrary grids," *Journal of Computational Physics*, Vol. 227, No. 20, 2008, pp. 8875–8893.

[39] Haga, T., Gao, H., and Wang, Z., "A high-order unifying discontinuous formulation for the Navier-Stokes equations on 3D mixed grids," *Mathematical Modelling of Natural Phenomena*, Vol. 6, No. 03, 2011, pp. 28–56.

[40] Darmofal, D. L., Allmaras, S. R., Yano, M., and Kudo, J., "An adaptive, higher-order discontinuous Galerkin finite element method for aerodynamics," AIAA Paper 2013-2871, 21st AIAA Computational Fluid Dynamics Conference, San Diego, CA, June 2013.

[41] Diosady, L. T. and Murman, S. M., "Design of a Variational Multiscale Method for Turbulent Compressible Flows," AIAA Paper 2013-2870, 21st AIAA Computational Fluid Dynamics Conference, San Diego, CA, June 2014.

[42] Hartmann, R., "Higher-order and adaptive discontinuous Galerkin methods with shock-capturing applied to transonic turbulent delta wing flow," *International Journal for Numerical Methods in Fluids*, Vol. 72, No. 8, 2013, pp. 883–894.

[43] Glasby, R. S., Burgess, N., Anderson, K., Wang, L., Allmaras, S., and Mavriplis, D., "Comparison of SU/PG and DG finite-element techniques for the compressible Navier-Stokes equations on anisotropic unstructured meshes," AIAA Paper 2013-691, 51st AIAA Aerospace Sciences Meeting, Grapevine, TX, January 2013.

[44] Brazell, M. J. and Mavriplis, D. J., "3D Mixed Element Discontinuous Galerkin with Shock Capturing," AIAA Paper 2013-3064, 21st AIAA Computational Fluid Dynamics Conference, San Diego, CA., June 2013.

[45] Ceze, M. and Fidkowski, K. J., "Drag prediction using adaptive discontinuous finite elements," *Journal of Aircraft*, Vol. 51, No. 4, 2014, pp. 1284–1294.

[46] Wang, L., Anderson, W. K., Erwin, J. T., and Kapadia, S., "Discontinuous Galerkin and Petrov Galerkin methods for compressible viscous flows," *Computers & Fluids*, Vol. 100, 2014, pp. 13–29.

[47] Burgess, N. and Mavriplis, D., "An hp-adaptive discontinuous Galerkin solver for aerodynamic flows on mixed-element meshes," AIAA Paper 2011-490, 49th AIAA Aerospace Sciences Meeting and Exhibit, Orlando FL, January 4-7, 2011.

[48] Wang, L. and Mavriplis, D. J., "Adjoint-based h–p adaptive discontinuous Galerkin methods for the 2D compressible Euler equations," *Journal of Computational Physics*, Vol. 228, No. 20, 2009, pp. 7643–7661.

[49] Reza Ahrabi, B., Anderson, W. K., and Newman, J. C., "High-order finite-element method and dynamic adaptation for two-dimensional laminar and turbulent Navier-Stokes," AIAA Paper 2014-2983, 32nd AIAA Applied Aerodynamics Conference, Atlanta, GA., June 2014.

[50] Brazell, M. J., Kirby, A. C., Sitaraman, J., and Mavriplis, D. J., "A multi-solver overset mesh Approach for 3D mixed element variable order discretizations," AIAA Paper 2016-2053, 54th AIAA Aerospace Sciences Meeting, San Diego, CA., June 2016.

[51] Kirby, A. C., Brazell, M. J., Mavriplis, D. J., and Sitaraman, J., "An Overset Adaptive High-Order Approach for Blade-Resolved Wind Energy Applications," AHS Forum 72, West Palm Beach, FL., May 2016.

[52] Mavriplis, D. J., "Grid resolution study of a drag prediction workshop configuration using the NSU3D unstructured mesh solver," AIAA Paper 2005-729, 23rd AIAA Applied Aerodynamics Conference, Toronto, Ontario Canada, June 2005.

[53] Mavriplis, D. and Long, M., "NSU3D Results for the Fourth AIAA Drag Prediction Workshop," *Journal of Aircraft*, Vol. 51, No. 4, 2014, pp. 1161–1171.

[54] Mavriplis, D. J. and Mani, K., "Unstructured Mesh Solution Techniques using the NSU3D Solver," AIAA Paper 2014-0081, Presented at the 52nd AIAA Aerospace Sciences Conference, National Harbor, MD, January 2014.

[55] Spalart, P. and Allmaras, S., "A one-equation turbulence model for aerodynamic flows," *30th aerospace sciences meeting and exhibit*, 1992, p. 439.

[56] Wilcox, D. C., "Reassessment of the scale-determining equation for advanced turbulence models," *AIAA journal*, Vol. 26, No. 11, 1988, pp. 1299–1310.

[57] Spalart, P. R., Deck, S., Shur, M., Squires, K., Strelets, M. K., and Travin, A., "A new version of detached-eddy simulation, resistant to ambiguous grid densities," *Theoretical and computational fluid dynamics*, Vol. 20, No. 3, 2006, pp. 181–195.

[58] Shur, M. L., Strelets, M. K., Travin, A. K., and Spalart, P. R., "Turbulence modeling in rotating and curved channels: assessing the Spalart-Shur correction," *AIAA journal*, Vol. 38, No. 5, 2000, pp. 784–792.

[59] Mavriplis, D., Long, M., Lake, T., and Langlois, M., "NSU3D results for the second AIAA high-lift prediction workshop," *Journal of Aircraft*, Vol. 52, No. 4, 2015, pp. 1063–1081.

[60] Park, M. A., Laflin, K. R., Chaffin, M. S., Powell, N., and Levy, D. W., "CFL3D, FUN3D, and NSU3D Contributions to the Fifth Drag Prediction Workshop," *Journal of Aircraft*, Vol. 51, No. 4, 2014, pp. 1268–1283.

[61] Wissink, A. M., Hornung, R. D., Kohn, S. R., Smith, S. S., and Elliott, N., "Large scale parallel structured AMR calculations using the SAMRAI framework," *Supercomputing, ACM/IEEE 2001 Conference*, IEEE, 2001, pp. 22–22.

[62] Kirby, A. C., Mavriplis, D. J., and Wissink, A. M., "An Adaptive Explicit 3D Discontinuous Galerkin Solver for Unsteady Problems," AIAA Paper 2015-3046, 22nd AIAA Computational Fluid Dynamics Conference, Dallas, TX, June 2015.

[63] Burstedde, C., Ghattas, O., Gurnis, M., Isaac, T., Stadler, G., Warburton, T., and Wilcox, L., "Extreme-scale AMR," *Proceedings of the 2010 ACM/IEEE International Conference for High Performance Computing, Networking, Storage and Analysis*, IEEE Computer Society, 2010, pp. 1–12.

[64] Burstedde, C., Wilcox, L. C., and Ghattas, O., "p4est: Scalable algorithms for parallel adaptive mesh refinement on forests of octrees," *SIAM Journal on Scientific Computing*, Vol. 33, No. 3, 2011, pp. 1103–1133.

[65] Isaac, T., Burstedde, C., Wilcox, L. C., and Ghattas, O., "Recursive algorithms for distributed forests of octrees," *SIAM Journal on Scientific Computing*, Vol. 37, No. 5, 2015, pp. C497–C531.

[66] Rudi, J., Malossi, A. C. I., Isaac, T., Stadler, G., Gurnis, M., Staar, P. W., Ineichen, Y., Bekas, C., Curioni, A., and Ghattas, O., "An extreme-scale implicit solver for complex pdes: Highly heterogeneous flow in earth's mantle," *Proceedings of the international conference for high performance computing, networking, storage and analysis*, ACM, 2015, p. 5.

[67] Smagorinsky, J., "General circulation experiments with the primitive equations: I. The basic experiment," *Monthly weather review*, Vol. 91, No. 3, 1963, pp. 99–164.

[68] Hindenlang, F., Gassner, G. J., Altmann, C., Beck, A., Staudenmaier, M., and Munz, C.-D., "Explicit discontinuous Galerkin methods for unsteady problems," *Computers & Fluids*, Vol. 61, 2012, pp. 86–93.

[69] Brazell, M. J., Kirby, A. C., and Mavriplis, D. J., "A high-order discontinuous-Galerkin octree-based AMR solver for overset simulations," AIAA Paper 2017-3944, 23rd AIAA Computational Fluid Dynamics Conference, Denver, CO., June 2017.

[70] Roget, B. and Sitaraman, J., "Robust and efficient overset grid assembly for partitioned unstructured meshes," *Journal of Computational Physics*, Vol. 260, 2014, pp. 1–24.

[71] Jung, Y. S., Govindarajan, B., and Baeder, J. D., "Unstructured/Structured Overset Methods for Flow Solver Using Hamiltonian Paths and Strand Grids," AIAA Paper 2016-1056, 54th AIAA Aerospace Sciences Meeting, San Diego, CA., June 2016.

[72] Skamarock, W. C., Klemp, J. B., Dudhia, J., Gill, D. O., Barker, D. M., Wang, W., and Powers, J. G., "A description of the advanced research WRF version 2," Tech. rep., DTIC Document, 2005.

[73] Han, Y., Stoellinger, M., and Naughton, J., "Large eddy simulation for atmospheric boundary layer flow over flat and complex terrains," *Journal of Physics: Conference Series*, Vol. 753, IOP Publishing, 2016, p. 032044.

[74] Roy, R. and Stoellinger, M. K., "Large Eddy Simulation of Wind Flow Over Complex Terrain: The Bolund Hill Case," *35th Wind Energy Symposium*, 2017, p. 1160.

[75] Han, Y. and Stoellinger, M. K., "Large eddy simulation of atmospheric boundary layer flows over complex terrain with varying stability conditions," *35th Wind Energy Symposium*, 2017, p. 1161.

[76] Kuhlen, T., Pajarola, R., and Zhou, K., "Parallel in situ coupling of simulation with a fully featured visualization system," 2011.

[77] Jonkman, J., Butterfield, S., Musial, W., and Scott, G., "Definition of a 5-MW reference wind turbine for offshore system development," *National Renewable Energy Laboratory, Golden, CO, Technical Report No. NREL/TP-500-38060*, 2009.

[78] Hand, M. M., Simms, D., Fingersh, L., Jager, D., Cotrell, J., Schreck, S., and Larwood, S., *Unsteady aerodynamics experiment phase VI: wind tunnel test configurations and available data campaigns*, National Renewable Energy Laboratory, Golden, Colorado, USA, 2001.

[79] Simms, D. A., Schreck, S., Hand, M., and Fingersh, L., *NREL unsteady aerodynamics experiment in the NASA-Ames wind tunnel: a comparison of predictions to measurements*, National Renewable Energy Laboratory Golden, CO, USA, 2001.

[80] Schreck, S., "The NREL full-scale wind tunnel experiment Introduction to the special issue," *Wind Energy*, Vol. 5, No. 2-3, 2002, pp. 77–84.

[81] Fingersh, L. J., Simms, D., Hand, M., Jager, D., Cotrell, J., Robinson, M., Schreck, S., and Larwood, S., "Wind Tunnel Testing of NRELs Unsteady Aerodynamics Experiment," AIAA Paper 2001-35, 20th ASME Wind Energy Symposium and the 39th Aerospace Sciences Meeting, Reno, NV, 2001.

[82] Buning, P. G., Jespersen, D. C., Pulliam, T. H., Chan, W., Slotnick, J. P., Krist, S., and Renze, K. J., "Overflow users manual," *NASA Langley Research Center*, Vol. 1, 1998.

[83] Moriarty, P., Rodrigo, J. S., Gancarski, P., Chuchfield, M., Naughton, J. W., Hansen, K. S., Machefaux, E., Maguire, E., Castellani, F., Terzi, L., et al., "IEA-Task 31 WAKEBENCH: Towards a protocol for wind farm flow model evaluation. Part 2: Wind farm wake models," *Journal of Physics: Conference Series*, Vol. 524, IOP Publishing, 2014, p. 012185.

[84] "NSF NWSC-2 Cheyenne," 2016, Computational and Information Systems Laboratory. 2017. Cheyenne: SGI ICE XA System (Climate Simulation Laboratory). Boulder, CO: National Center for Atmospheric Research. doi:10.5065/D6RX99HX.

[85] Bergström, H., "Meteorological Conditions at Lillgrund," Tech. rep., Uppsala University Document, 2009.

[86] Churchfield, M., Lee, S., Moriarty, P., Martinez, L., Leonardi, S., Vijayakumar, G., and Brasseur, J., "A Large-Eddy Simulations of Wind-Plant Aerodynamics," *50th AIAA Aerospace Sciences Meeting including the New Horizons Forum and Aerospace Exposition*, American Institute of Aeronautics and Astronautics, Reston, Virigina, January 2012.

[87] Moeng, C.-H. and Sullivan, P. P., "A comparison of shear-and buoyancy-driven planetary boundary layer flows," *Journal of the Atmospheric Sciences*, Vol. 51, No. 7, 1994, pp. 999–1022.

[88] Schumann, U., "Subgrid Scale Model for Finite Difference Simulations of Turbulent Flows in Plane Channels and Annuli," *Journal of Computational Physics*, Vol. 18, 1975, pp. 376–404.

[89] Groetzbach, G. and Schumann, U., "Direct numerical simulation of turbulent velocity-, pressure-, and temperature-fields in channel flows," *In: Symposium on Turbulent Shear Flows, University Park, Pa., April 18-20, 1977, Proceedings. Volume 1. (A77-33806 15-34) University Park, Pa., Pennsylvania State University, 1977, p. 14.11-14.19.*, Vol. 1, 1977, pp. 14.11–14.19.