International Journal of
HIGH PERFORMANCE
COMPUTING APPLICATIONS

Research Paper

Wind farm simulations using an overset *hp*-adaptive approach with blade-resolved turbine models

The International Journal of High Performance Computing Applications
1–27
© The Author(s) 2019
Article reuse guidelines:
sagepub.com/journals-permissions
DOI: 10.1177/1094342019832960
journals.sagepub.com/home/hpc
SAGE

Andrew C Kirby[1], Michael J Brazell[1], Zhi Yang[1], Rajib Roy[1], Behzad R Ahrabi[1], Michael K Stoellinger[1], Jay Sitaraman[2] and Dimitri J Mavriplis[1]

Abstract

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 microscale modeling are incorporated into the analysis environment. Validation of the computational framework is performed on the National Renewable Energy Laboratory (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 setups including the 48 turbine wind plant known as Lillgrund. The largest case consisting of 96 wind turbines and a total of 385 overset grids are run on 44,928 cores at a weak scaling efficiency of 86%. Demonstration of the coupling of atmospheric microscale and Computational Fluid Dynamics (CFD) solvers is presented using the National Center for Atmospheric Research (NCAR) Weather Research and Forecasting Model (WRF) solver and the NREL Simulator fOr Wind Farm Applications (SOWFA) solver.

Keywords

High performance computing, wind energy, overset grids, high-order methods, discontinuous Galerkin

1. Introduction

Wind energy 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 (Wind Vision, 2015). The transition from fossil fuels to renewable energies will strengthen global energy security and reduce greenhouse gas emissions (Umbach, 2010). Improved understanding of wind farm dynamics will enable better wind turbine placement in wind farm configurations which will increase wind plant efficiency. 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 have 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) (Messina, 2016). The ECP was established to maximize the benefits of high performance computing (HPC) and accelerate the development of a capable exascale computing ecosystem. As part of the ECP, a project titled "Exascale Predictive Wind Plant Flow Physics Modeling" has been formed to advance the understanding of wind farm

[1] Department of Mechanical Engineering, University of Wyoming, Laramie, WY, USA
[2] Parallel Geometric Algorithms, LLC, Sunnyvale, CA, USA

Corresponding author:
Michael J Brazell, Department of Mechanical Engineering, University of Wyoming, Laramie, WY 82071, USA.
Email: mbrazell@uwyo.edu

interactions such as wake dynamics, complex terrain effects, and turbine–turbine interaction effects (Turbine Wind Plant Efficiency, 2016). 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 (National Renewable Energy Laboratory [NREL] to Lead One Exascale Computing Project, 2016). An estimate of 100 billion DOFs will be required to simulate this problem. The goals of the current work 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 turbine parameterization techniques such as actuator lines (Churchfield et al., 2012; Churchfield et al.; Fleming et al., 2014; Troldborg et al., 2007) or actuator discs (Mikkelsen, 2003) to high fidelity blade-resolved models (Bazilevs et al., 2011a, 2011b; Duque et al., 2003; Gomez-Iradi et al., 2009; Gopalan et al., 2014; Gundling et al., 2011, 2012; Li et al., 2012; Pape and Lecanu, 2004; Potsdam and Mavriplis, 2009; Rai et al., 2012; Sørensen et al., 2002; Takizawa et al., 2011; Yelmule and Vsj, 2013; Zahle et al., 2009). High fidelity simulations require the use of a full rotor model where the exact geometries of the turbine blade and tower are 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 1 million to 10 million computing cores (Top 500, 2016). Wind plant simulations using full rotor models have recently been applied with the CREATE-AV HELIOS (Sitaraman et al.,; Wissink et al., 2016; Wissink et al.,) 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. (2015). In the work of Sitaraman et al., HELIOS compared uniform velocity and atmospheric inflow conditions of a wind farm containing 48 wind turbines using 3840 CPU cores. The full rotor model mesh in that 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 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 HPC 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 (Brazell et al., 2016b; Crabill et al.; Noack; Steger, 1983; Wissink et al., 2016). 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 (Vincent et al.; Witherden et al., 2015). In particular, we employ a high-order discontinuous Galerkin (DG) finite element method (FEM) discretization in the off-body region. DG methods and similar FEM solvers have been deployed for a variety of aerospace applications (Brazell and Mavriplis, 2013; Ceze and Fidkowski, 2014; Cockburn et al., 2000; Darmofal et al., 2013; Diosady and Murman, 2014; Fidkowski, 2014; Glasby et al., 2013; Haga et al., 2011; Hartmann, 2013; Luo et al., 2008; Wang et al., 2014) including the use of adaptive mesh and solution refinement techniques (Burgess and Mavriplis, 2011; Reza Ahrabi et al., 2014; Wang and Mavriplis, 2009).

The article is outlined as follows. In section 2, the computational methodology outlining the individual software components is presented. Section 3 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 4), a weak scaling study using multiple wind turbine arrangements (section 5) and, lastly, demonstration of the microscale atmospheric coupling (section 6). Section 7 concludes the paper with a discussion on future work.

2. 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^2A^2KE3D). W^2A^2KE3D has been previously demonstrated on various aerodynamics problems such as flow over the NACA0015 wing and flow over a sphere (Brazell et al., 2016a). Additionally, preliminary wind energy application results were presented using the NREL Unsteady Aerodynamics Experiment Phase VI (Kirby et al., 2016). 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^2A^2KE3D 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

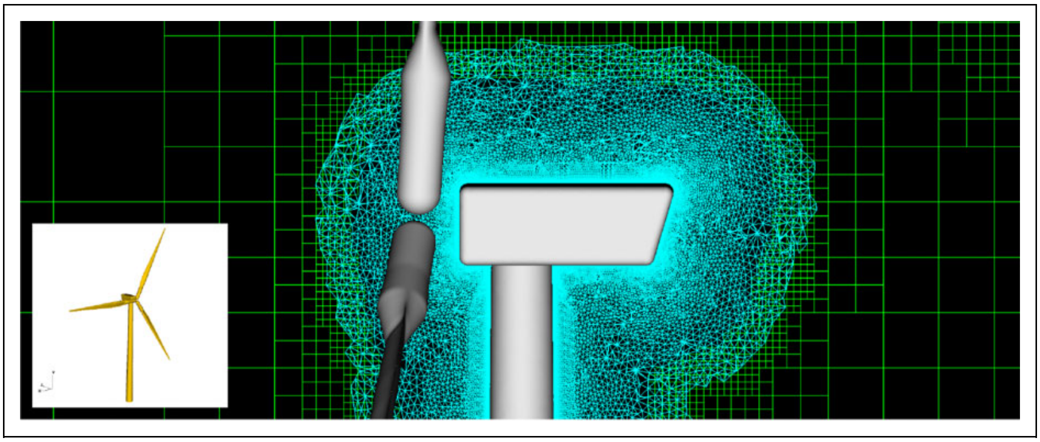

Figure 1. Siemens SWT-2.3-93 wind turbine mesh system composed of multiple overlapping meshes. The wind turbine rotor is composed of three unstructured blade meshes constructed from one mesh template which is replicated three times, rotated and translated into the starting positions. A fourth unstructured mesh is used to represent the tower and nacelle (shown in light-blue). Lastly, an off-body adaptive mesh is overlaid to provide the domain discretization away from the wind turbine (shown in green).

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, translated, and rotated to the correct starting position. The tower geometry is fitted with an independent unstructured near-body mesh component.

The W^{2}A^{2}KE3D framework allows for multiple Computational Fluid Dynamics (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 solution data and mesh are held by the driver which is fed to the in-situ flow visualization software also implemented in the driver as a dynamic library.

2.1. Near-body flow solver

The near-body flow solver utilized in W^{2}A^{2}KE3D is NSU3D (*N*avier–*S*tokes *U*nstructured) (Mavriplis, 2005; Mavriplis and Long, 2014). NSU3D is a well-established Unsteady Reynolds Averaged Navier–Stokes 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 (Mavriplis and Mani, 2014). NSU3D contains several turbulence models for various aerodynamics problems. This includes the Spalart–Allmaras (SA), K-Omega (Wilcox, 1988), and the Delayed Detached Eddy Simulation (DDES) (Spalart et al., 2006) turbulence models with rotation/curvature correction (Shur et al., 2000). 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 (Mavriplis et al., 2015) and the AIAA Drag Prediction Workshop (Park et al., 2014) series. The solver has been demonstrated to have good strong scalability on the National Center for Atmospheric Research (NCAR) supercomputer NWSC-1 Yellowstone. Figure 2 illustrates the scalability of the NSU3D unstructured mesh solver, using either an explicit (single grid) or a multigrid solver, on Yellowstone showing good scalability on a relatively coarse mesh of 75 million points up to 32,768 cores. Additionally, NSU3D has served as a near-body flow solver in the CREATE-AV HELIOS (Wissink et al., 2016) software.

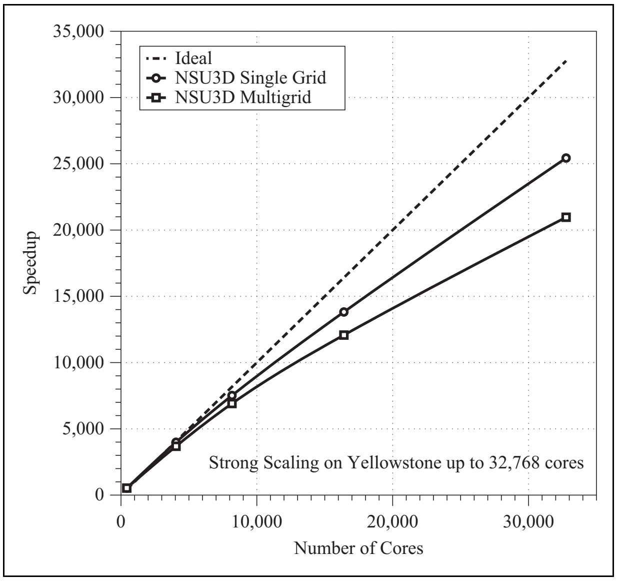

Figure 2. Scalability of near-body unstructured mesh solver NSU3D on Yellowstone using up to 32,768 cores.

2.2. Off-body flow solver

W^2A^2KE3D utilizes a variable order DG FEM implemented in a dynamic adaptive mesh refinement (AMR) framework. In previous work, W^2A^2KE3D used an off-body solver known as SAMCartDG based on the open-source SAMRAI (Wissink et al.) AMR package provided by Lawrence Livermore National Laboratory with the standalone DG solver CartDG (Kirby et al., 2015). 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` (Burstedde et al.; Burstedde et al., 2011). The `p4est` software library uses scalable adaptive algorithms for parallel AMR on forests of octrees (Isaac et al., 2015). The `p4est` library was part of the winning team of then ACM Gordon Bell Prize in 2015 (Rudi et al.) 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` (Brazell et al., 2017). 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 FEM. That is, the parallel processing communication pattern in the FEM, particularly for DG 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. Figure 3 demonstrates the octree mesh structure and the direct communication scheme where elements annotated with the letter **G** (ghost) represent copied solutions from neighboring Message Passing Interface (MPI) rank cells.

In contrast, since SAMRAI utilizes a patch-based AMR system, the native parallel communication patterns are composed of data transfers between AMR level interfaces in multiple communication phases. Figure 4 demonstrates the patch-based mesh structure and the indirect communication scheme. In the first communication phase, solution data are interpolated and communicated from coarse to fine levels. Once all ghosts are filled, solutions are advanced on all levels independently. After advancing the solution, the second phase of communication is activated. The solution on the fine levels are disseminated to coarser levels through a coarsening interpolation operator. Additionally, the flux calculated at the coarse-fine boundaries on the fine levels is communicated to the underlying coarse level where a flux correction must be performed in order to maintain conservation of the physics laws. This procedure adds extraneous communication and algorithmic complexities for FEM-based discretization. Thus, it is far more efficient to perform direct solution exchanges in as done in the `p4est` framework.

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 (Smagorinsky, 1963). 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. (2015) which follows the derivation in Diosady and Muman (2014) and Hindenlang et al. (2012). 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 (Brazell et al., 2017).

To demonstrate the computational efficiency of CartDG, several performance statistics for a wide range of orders of solution accuracy are provided. The sustained peak performance of the inviscid and viscous residual evaluations is shown in Figure 5 on a Intel Core i7-5960X Haswell-E processor with eight CPU cores, clock speed of 3 GHz and 4 GB of memory per core. For reference, the TauBench benchmark for this processor is 5.25 s with a theoretical peak performance of 384 GFLOPS. As seen from the figure, the tensor-product-based FEM formulation achieves roughly 12% sustained compute peak performance for evaluation of the residual of the compressible Navier–Stokes equations (viscous residual) at high solution

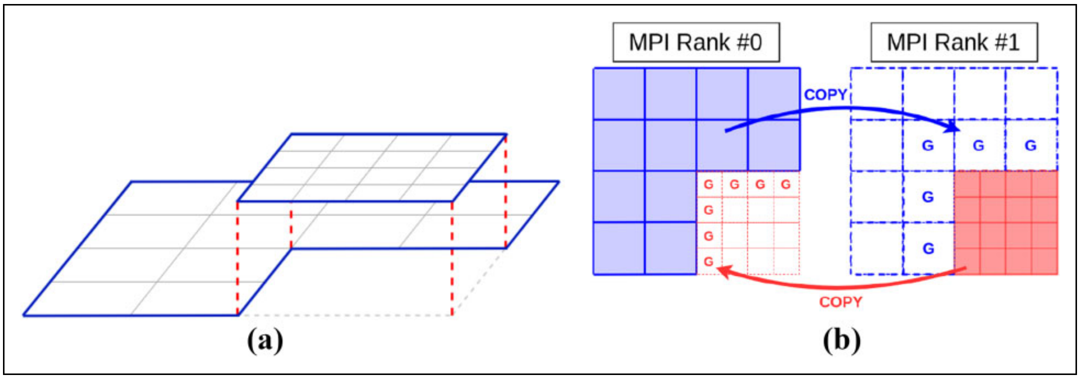

Figure 3. In octree-based AMR, coarse cells tagged for refinement are destroyed and replaced by fine cells as demonstrated in (a). The parallel communication pattern for two Message Passing Interface (MPI) ranks is demonstrated in (b). The octree-based AMR framework `p4est` performs a direct exchange of solution data whereby the ghost cells, annotated by **G**, are filled simultaneously by direct copy from the corresponding cells on neighboring MPI ranks. (a) Grid structure of an octree-based AMR system. Coarse elements that are overlapped by fine elements are replaced and destroyed. (b) MPI Communication pattern of ghosts (**G**). Colored cells are real cells with respect to each MPI rank. AMR: adaptive mesh refinement.

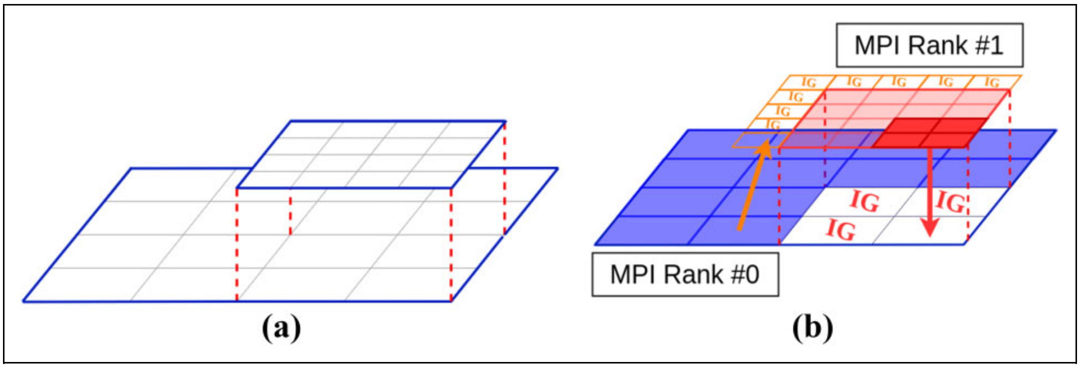

Figure 4. In patch-based AMR, coarse cells tagged for refinement are overlaid by fine cells as demonstrated in (a). The parallel communication pattern for two Message Passing Interface (MPI) ranks is demonstrated in (b). In the first communication phase, annotated by the orange arrow in (b), ghosts located on coarse-fine element boundaries on fine levels are filled from neighboring MPI ranks via solution interpolation annotated by **IG**. Once all fine-levels ghosts are filled, every level performs a solution update independent of the neighboring levels. In the second phase of communication, annotated by the red arrow, the solution is interpolated to the underlying coarse elements from the overlapping fine elements and the fine-level fluxes are transferred down for flux correction at the coarse-level coarse-fine element boundary. (a) Grid structure of a patch-based AMR system. Coarse elements are allowed to be overlapped by fine elements. For numerical discretizations based on control volumes, fluxes calculated at the coarse-fine element boundary must match on each level to preserve the physics conservation laws. (b) MPI communication patterns for patch-based AMR. Phase I (orange): coarse-to-fine ghost interpolation (**IG**) and same level solution copy communication. Phase II (red): fine-to-coarse ghost interpolation (**IG**) and fine-level flux transfer communication.

orders of accuracy. Figure 6 demonstrates the time per degree of freedom for inviscid and viscous residual evaluations. Overall, the time required to perform a viscous residual evaluation is higher than an inviscid residual evaluation due to more terms in the governing physics equations. For higher polynomial degrees, the time per degree of freedom becomes approximately constant for both inviscid and viscous residuals. The algorithm is communication bound at low polynomial degrees due to low arithmetic intensity. This severely limits the performance, effectively, to the communication bandwidth of the CPU. Figure 7 shows the sustained peak performance of the tensor-based DG method in comparison to a general FEM formulation on the NWSC-1 Yellowstone supercomputer using 128 Intel Xeon E5-2670 cores with a clock speed of 2.6Ghz and 2 GB per core memory. The general FEM implementation is able achieve approximately 80% sustained peak performance whereas the tensor-product formulation achieves 11% of peak performance. However, the general FEM formulation requires significantly more floating point operations, most of which are multiplication and addition of zeros for a nodal collocated basis. This is demonstrated in Figure 8 comparing the time per degree of freedom for a viscous residual evaluation. The cost of the general FEM formulation is nearly 10 times larger than that required by the tensor-product formulation at high polynomial degrees. The only polynomial degree for which the general FEM formulation is more efficient is $p = 0$ which corresponds

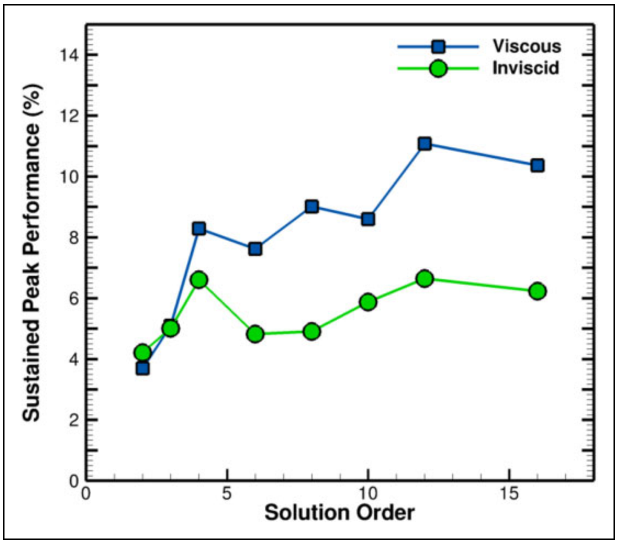

Figure 5. Sustained peak performance of CartDG, the off-body numerical solver kernel, for tensor-product basis formulations of viscous and inviscid equations for discretization orders of accuracy ranging from 2nd-order to 16th-order.

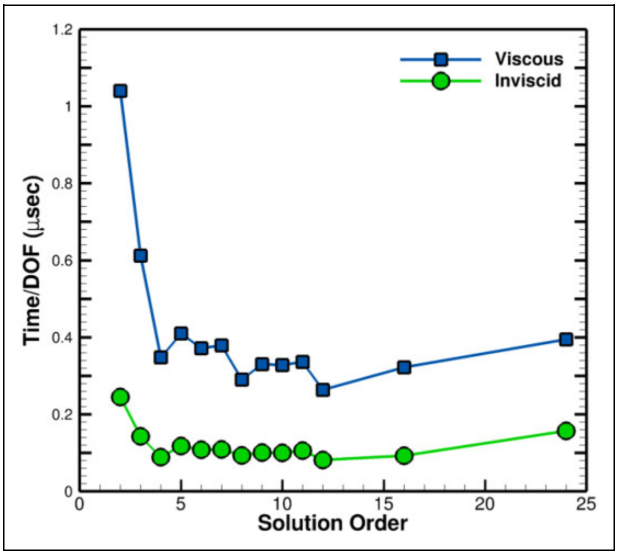

Figure 6. Time per degree-of-freedom of CartDG, the off-body numerical solver kernel, for tensor-product basis formulations of viscous and inviscid equations for discretization orders of accuracy ranging from 2nd-order to 24th-order (lower values are better).

to first-order solution accuracy. However, this polynomial degree is never used in simulations because of its severe solution inaccuracy and large numerical dissipation.

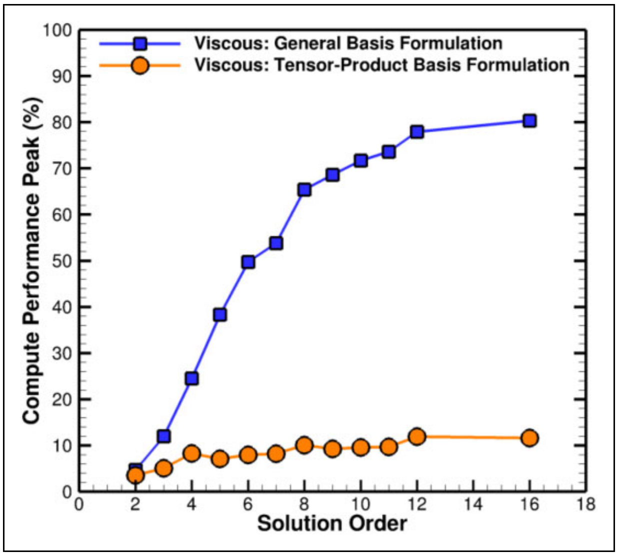

Figure 7. Sustained peak performance of CartDG, the off-body numerical solver kernel, for tensor-product basis and general basis formulations of the Compressible Navier–Stokes Equations for discretization orders of accuracy ranging from 2nd-order to 16th-order. The general basis formulation operations are casted into Level 3 BLAS matrix–matrix products resulting very high computational peak performance percent achievement.

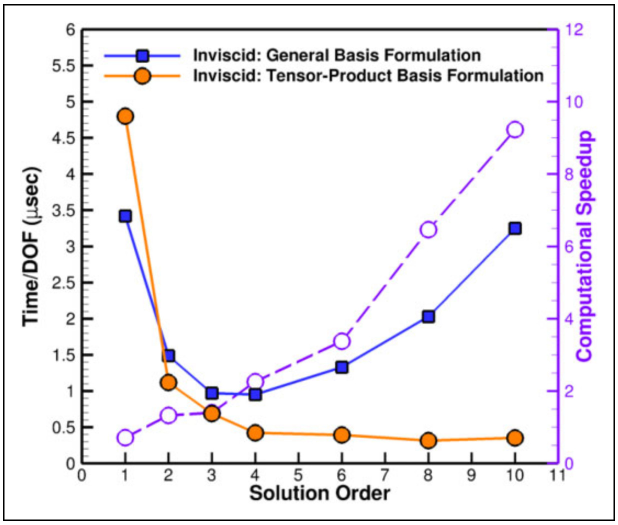

Figure 8. Time/DOF performance comparison of CartDG, the off-body numerical solver kernel, for tensor-product basis and general basis formulations of the Compressible Euler Equations for discretization orders of accuracy ranging from 1st-order to 10th-order. *Lower values of Time/DOF imply better computational efficiency*. The right axis demonstrates the computational speedup achieved by utilizing a tensor-product basis formulation (which utilizes a sum-factorization approach) compared to a general basis formulation (which utilizes matrix-matrix multiplication). Even though the general basis formulation achieves much greater computational peak performance, at 10th-order, it is nearly ten times slower than the tensor-product basis formulation. DOF: degree-of-freedom.

2.3. 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) (Brazell et al., 2016b; Crabill et al.; Roget and Sitaraman, 2014) 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 callback functions are provided in the TIOGA API (Brazell et al., 2016b). 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 Vandermonde matrix approach (Brazell et al., 2016b). 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 FEM. 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 (Brazell et al., 2016a). TIOGA has also been utilized in high-order solution techniques of intersecting Hamiltonian path and strand mesh grids (Jung et al., 2016).

2.4. 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 are 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 (Whitlock et al., 2011) library has been instrumented into the driver of W^2A^2KE3D . 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 run time. 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, iso-surfaces, 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^2A^2KE3D framework is provided with in-situ visualization and analysis automatically.

2.5. Driver

The driver 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 or (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 flow solvers to execute in parallel and each flow solver to have different numbers of CPU cores with respect to each other. In W^2A^2KE3D, 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 thereby introducing more degrees of freedom leading to a load imbalance of the simulation. To alleviate this problem, redistribution of the problem can be performed when restarting a simulation via checkpointing for more effective use of computational resources. That is, more CPU cores can be allocated to the respective flow solver requiring more computational resources as the solution evolves. Each flow solver component in the W^2A^2KE3D framework has this capability: NSU3D can be manually redistributed while `dg4est` has automatic redistribution when provided more CPU cores for a restarted solution.

2.6. Time step coordination

During a simulation, a driver global time step orchestrates solution exchange between meshes, solution updates, checkpointing, and mesh movement or adaption as

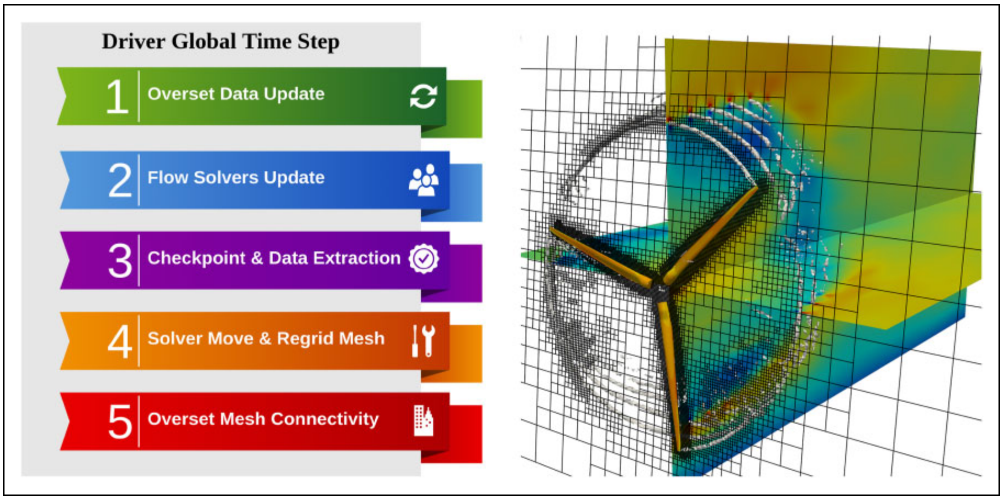
Driver Global Time Step
1 Overset Data Update
2 Flow Solvers Update
3 Checkpoint & Data Extraction
4 Solver Move & Regrid Mesh
5 Overset Mesh Connectivity

Figure 9. 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 on disjoint groups of Message Passing Interface (MPI) ranks for parallel flow solution updates.

demonstrated in Figure 9. In step 1, solution data lying at the overset grid interface, as shown in the overlapping mesh sections in Figure 1, are exchanged between overlapping meshes. When two (or more) cells overlap each other, the mesh element of finer geometric resolution serves as the solution *donor* and the other mesh element of coarser geometric resolution serves as the solution *receptor*. In the scenario where a fine cell touches its own grid boundary and is overlapped by a coarse cell, the fine element is labeled as the receptor and the coarse element as the donor in order to communicate external flow information.

Each global time step is loosely coupled. At the beginning of each global time step, the flow solutions on each mesh are exchanged via interpolation to their counterpart mesh receptor cells. When an unsteady time step is executed, the flow solvers iterate in time in an *uncoupled* manner for a time step of size Δt. The receptor cells in each mesh are held constant during the duration of the global time step therefore serving as *boundary conditions* for surrounding elements in its respective mesh. Loosely coupling each global time step results in first-order temporal accuracy at the overset grid interface. However, this allows for independent temporal discretions and reduced communication between meshes. For example, the unstructured mesh containing the aerodynamic boundary layer may elect to use an implicit solution approach to overstep the stringent time-step Courant-Friedrichs-Lewy (CFL) condition whereas the off-body mesh containing large elements may utilize an explicit solution approach for computational efficiency. This stage corresponds to step 2 in Figure 9.

In steps 3 and 4 of Figure 9, the flow solvers can perform auxiliary functions including solution checkpointing or in-situ data analysis, and mesh motion or adaption, respectively. If the latter events occur, the overset mesh connectivity is recomputed in step 5 for the next global time step.

3. Results: Overview

W^{2}A^{2}KE3D 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. Lastly, we demonstrate the microscale atmospheric and CFD coupling using the Weather Research and Forecasting Model (WRF) and Simulator fOr Wind Farm Applications (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-

Table 1. Mesh statistics used in the mesh convergence study of the NREL 5MW wind turbine blade.[a]

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

[a] Each blade mesh is replicated and placed into the correct starting position in the three-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.

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-order cells. For all simulations, all solvers are evolved using an unsteady formulation with time-accurate methods. As indicated in the previous section, the receptor cells in each mesh serve as input boundary conditions which are held constant over the entire global time step of size Δt independent of the time discretization. Even thought this approach results in first-order temporal accuracy at the overset interpolation region, this loose coupling enables autonomy in the time discretization approaches and parameters such as number of implicit linear sub-iterations or the order of the local time-stepping scheme. In this work, 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) for multiple local time steps to discretize each global time step.

4. Results: Single turbine simulations

4.1. 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 (Jonkman et al., 2009). 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 m 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.

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 10 demonstrates the time history convergence of the mesh resolution study. Figure 10(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 10(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 10(a) while the thrust prediction is only marginally effected in Figure 10(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 sub-iteration convergence study for the BDF-2 time stepping is performed in Figure 11. 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 and thrust prediction using more sub-iterations remain the same as using fewer sub-iterations.

In contrast to the sub-iteration convergence study, the time step convergence study shown in Figure 12 demonstrates a significant influence of the global time step size

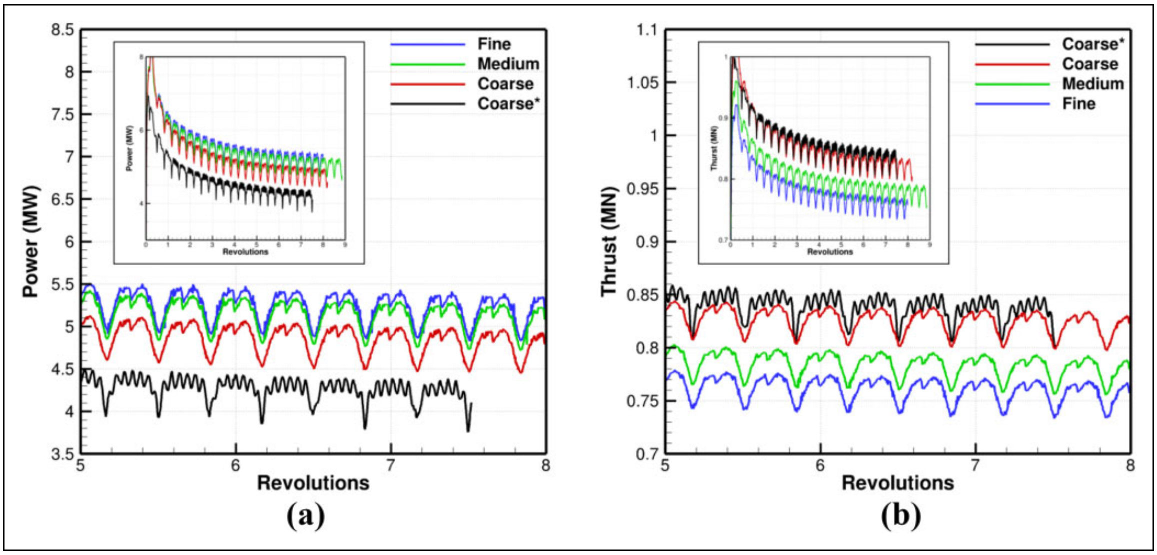

Figure 10. NREL 5MW power and thrust simulation results for the mesh resolution study. Each simulation uses a time step corresponding to a 1/4° rotation. Each time step was solved with BDF-2 using 50 sub-iterations for the near-body flow solver. (a) Power convergence history for inflow velocity 11.4 m/s for wind turbine rotor revolutions 5–8. The complete temporal power history is inlaid in the upper-left corner. (b) Thrust convergence history for inflow velocity 11.4 m/s for wind turbine rotor revolutions 5–8. The complete temporal power history is inlaid in the upper-left corner.

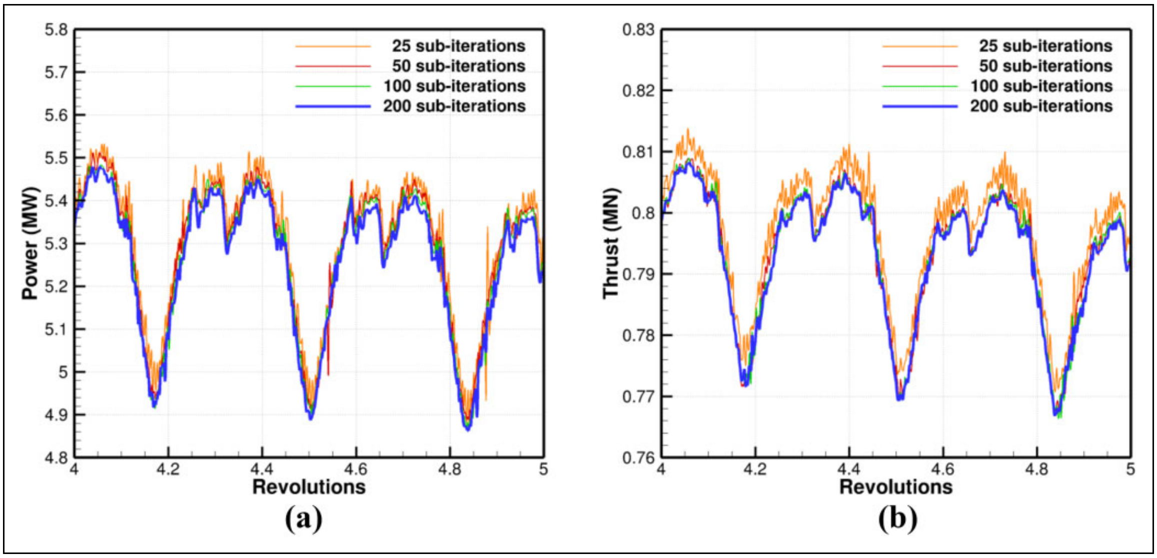

Figure 11. Simulated NREL 5MW force histories using BDF-2 time stepping for varying numbers of sub-iterations for the near-body flow solver. All results performed on the Medium-refined mesh in Table 1. (a) Power history at 4–5 revolutions with time step corresponding to 1/4° rotation. (b) Thrust history at 4–5 revolutions with time step corresponding to 1/4° rotation.

for force prediction. For this study, we choose global time steps corresponding to 1/4°, 1/2°, and 1° of rotor rotation. We can define a local blade tip CFL number as the product of blade tip speed 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°, 1/2°, and 1°, respectively. This local CFL represents the cell distances an advective wave can travel in a global time step. Thus for the 1/4° time step, an advective 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 12, 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° is used.

Figure 13 shows the power and thrust predictions for inflow velocities 6–11.4 m/s using the medium mesh compared to the NREL FAST (Jonkman et al., 2009) reference

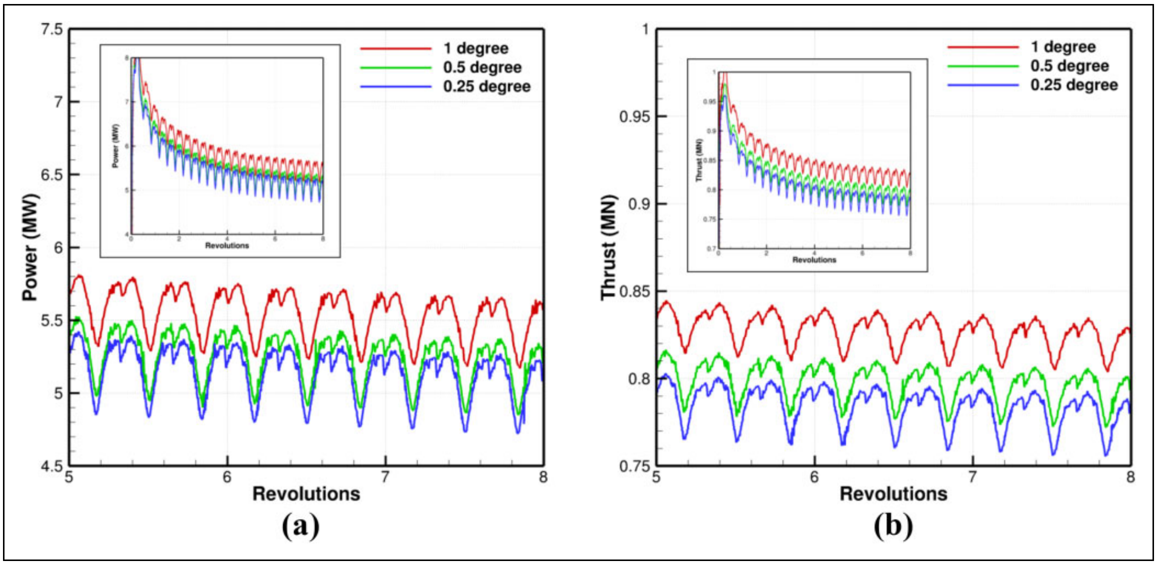

Figure 12. NREL 5MW force histories using BDF-2 time stepping for varying sizes of time step for the near-body flow solver. All results are performed on the Medium-refined mesh containing 927,701 points per blade (listed in Table 1). (a) Power history for time step sizes corresponding to 1/4°, 1/2°, and 1° of rotation using 50 sub-iterations. (b) Thrust history for time step sizes corresponding to 1/4°, 1/2°, and 1° of rotation using 50 sub-iterations.

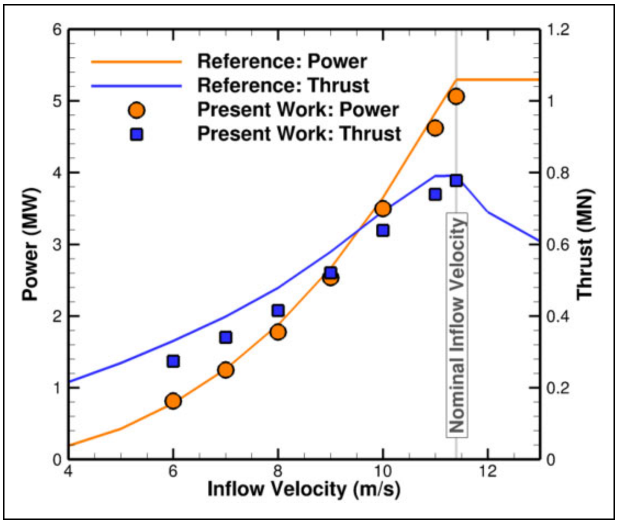

Figure 13. NREL 5MW power and thrust simulation results using a time step corresponding to a 1/4° rotation. Each time was solved with BDF-2 using 50 sub-iterations for the near-body flow solver on the medium mesh. Reference solution data provided by the NREL FAST (Jonkman et al., 2009) software.

solution. The reference solution uses blade element momentum theory with a fluid-structure interface to model structural wake effects (Jonkman et al., 2009). 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. It is important to note that the W^2A^2KE3D framework results do not yet contain blade elastic structural deflection responses.

4.2. NREL Phase VI

The NREL Unsteady Aerodynamics Experiment Phase VI is a wind turbine that has been studied experimentally (Fingersh et al., 2001; Hand et al., 2001; Schreck, 2002; Simms et al., 2001). The wind turbine was studied at NASA Ames Research Center in the 80 ft × 120 ft (24.4 m × 36.6 m) 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 (Hand et al., 2001). 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 14 shows the near-body surface mesh and Figure 15(a) depicts the near-body and off-body mesh system.

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

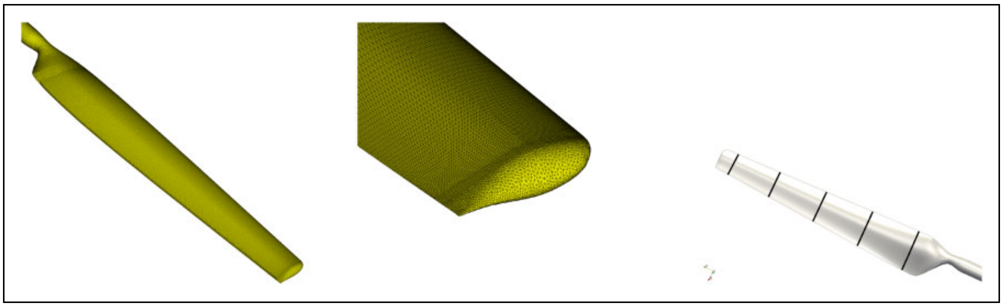

Figure 14. NREL Phase VI computational near-body surface containing three million nodes. The right figure shows the span-wise stations used for pressure coefficient measurements.

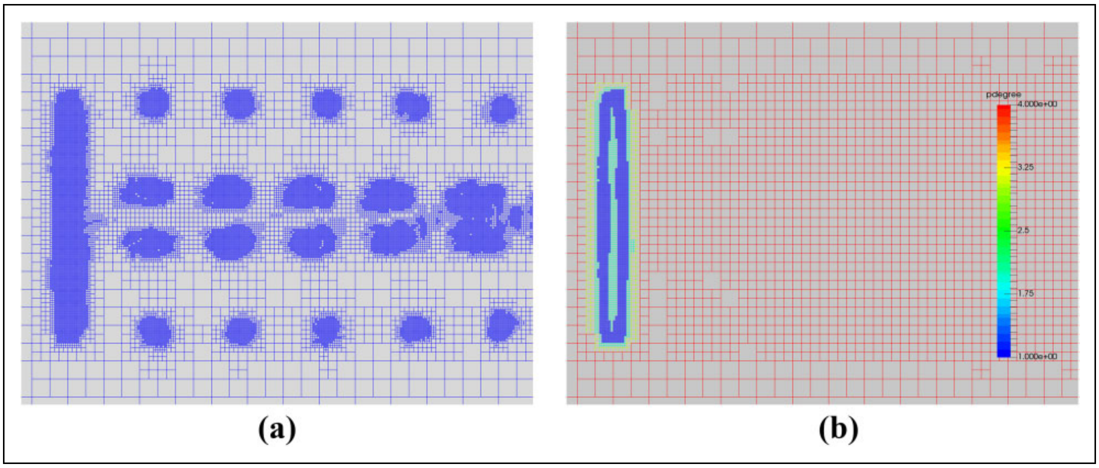

Figure 15. NREL Phase VI overset mesh systems with wake mesh adaption comparing lower solution accuracy methods (2nd-order) to higher-order solution accuracy methods (5th-order). (a) Off-body mesh adaption of $p = 1$, 2nd-order, spatial solution order of accuracy. (b) Off-body mesh adaption colored by spatial solution order of accuracy with $p = 1$, 2nd-order, near the wind turbine and growing to $p = 4$, 5th-order accuracy, in the wake region.

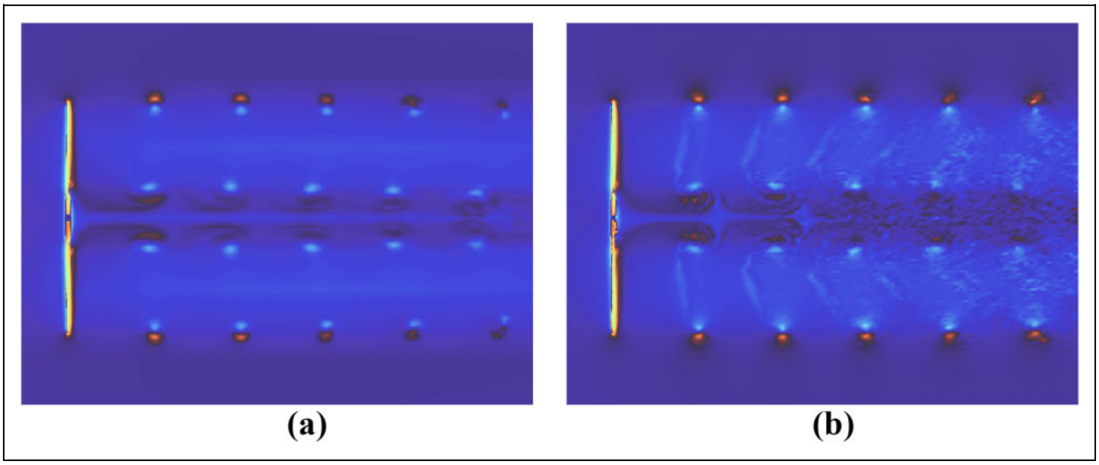

Figure 16. NREL Phase VI wake comparison of 2nd-order and 5th-order spatial accuracies. The 2nd-order solution (left) dissipates the vortex sheet that connects the tip vortex to the root vortex. (a) Velocity magnitude of $p = 1$, 2nd-order accuracy. (b) Velocity magnitude of $p = 4$, 5th-order accuracy.

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 15(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 15(c) shows the difference in AMR when only $p = 1$, second-order accurate elements are used.

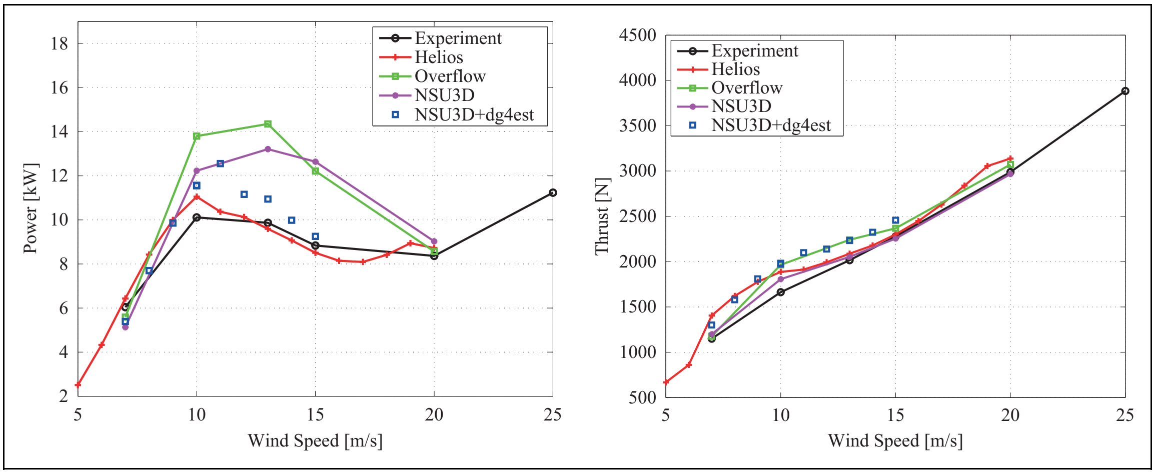

Figure 17. Simulation results for the NREL Phase VI power and thrust for uniform inflow velocities of 7–15 m/s. Results are compared to the experimental values along with other numerical simulations: NSU3D in stand-alone, CREATE-AV HELIOS (Wissink et al.), and NASA Overflow (Potsdam and Mavriplis, 2009). The W^2A^2KE3D results are represented by the blue squares and annotated as NSU3D+dg4est in the legend.

Figure 16 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.

Figure 17 shows the power and thrust predictions of W^2A^2KE3D. 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^2A^2KE3D results for power are significantly more accurate than the stand-alone OVERFLOW and NSU3D results.

Figure 18 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 19 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.

4.3. Siemens SWT-2.3-93

A generic Siemens SWT-2.3.93 turbine model using specifications from the IAE Wind Task 31-Wakebench (Moriarty et al.) 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 5 MW 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 20. Figure 21 shows a volume rendering and an iso-surface visualization of the Siemens-2.3-93 wind turbine.

5. Results: Weak scaling 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

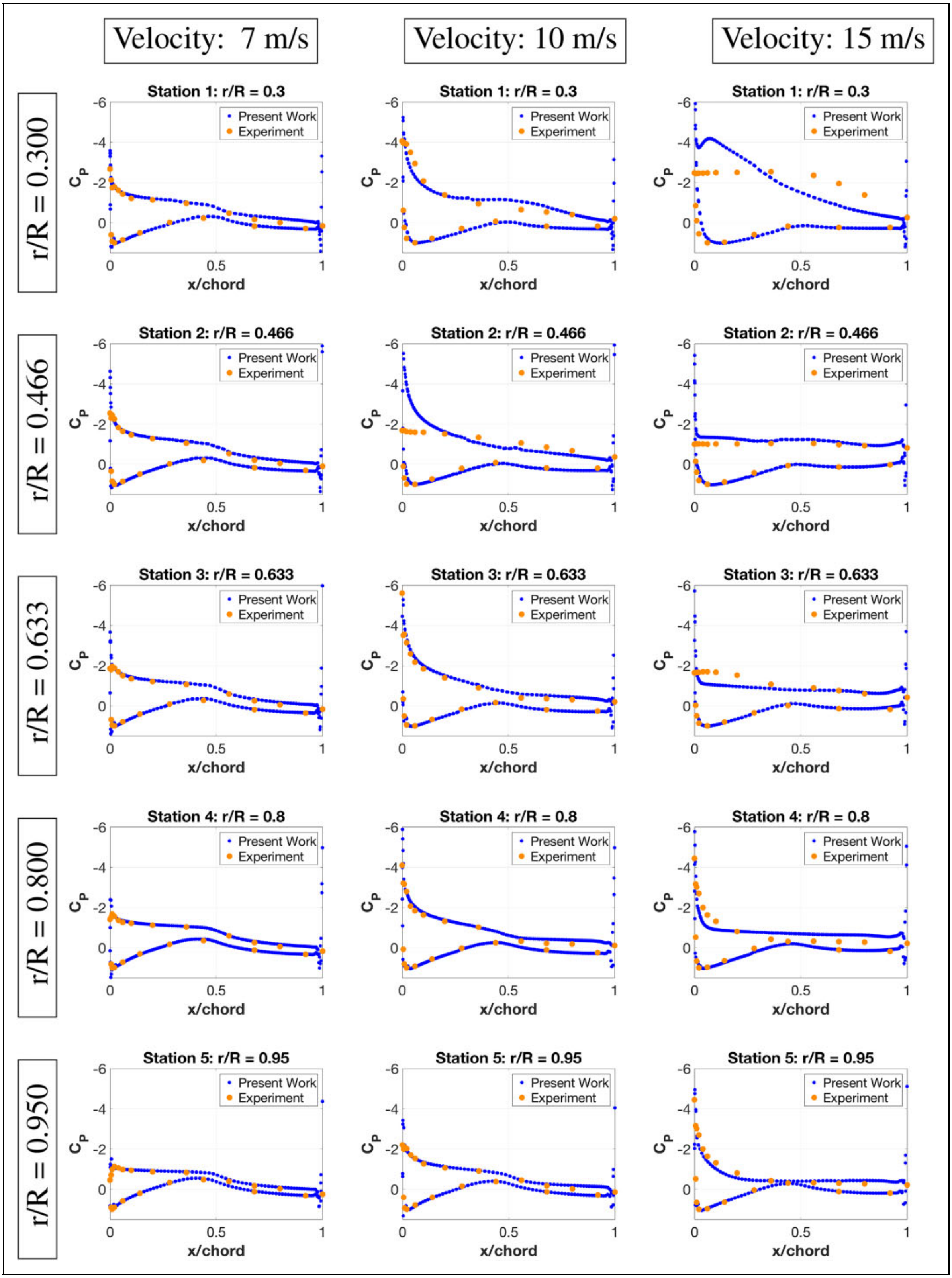

Figure 18. NREL Phase VI pressure coefficients at 30%, 46.6%, 63.3%, 80%, 95% span-wise stations for 7 m/s (column 1), 10 m/s (column 2), and 15 m/s (column 3) uniform axial inflow velocities. Predicted results of W^2A^2KE3D are plotted versus the experimental data.

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

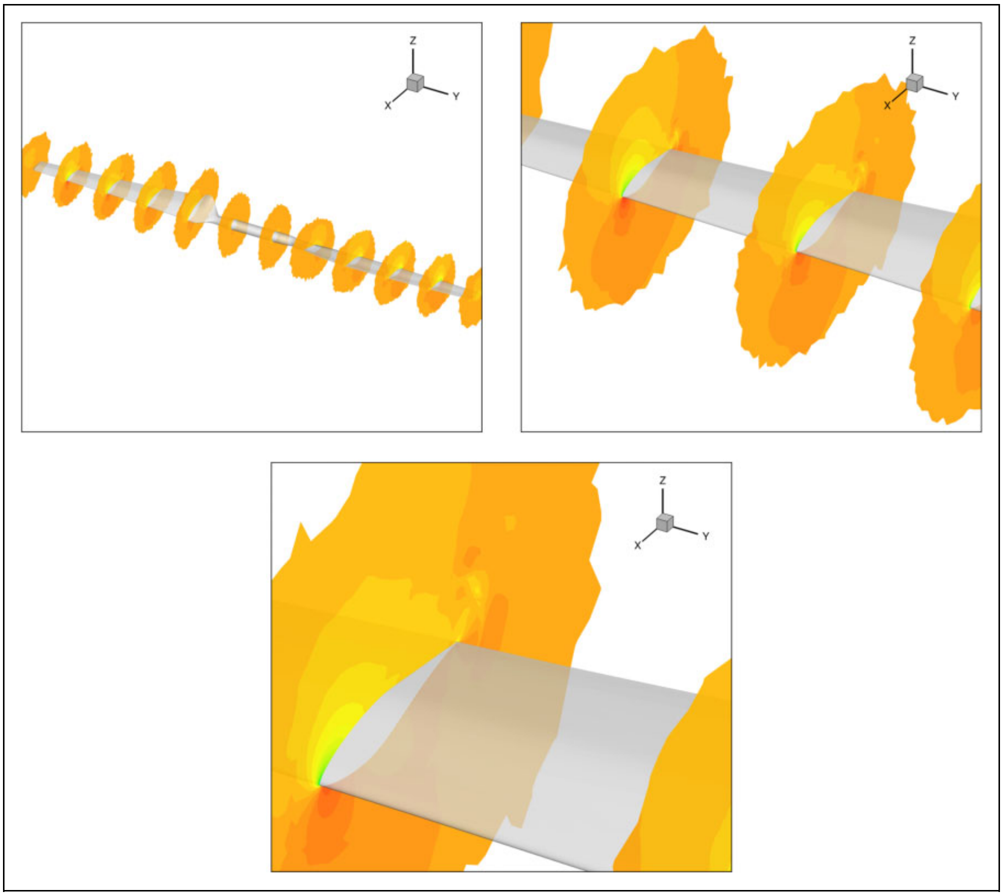

Figure 19. NREL Phase VI coefficient of pressure visualized for 11 m/s inflow velocity.

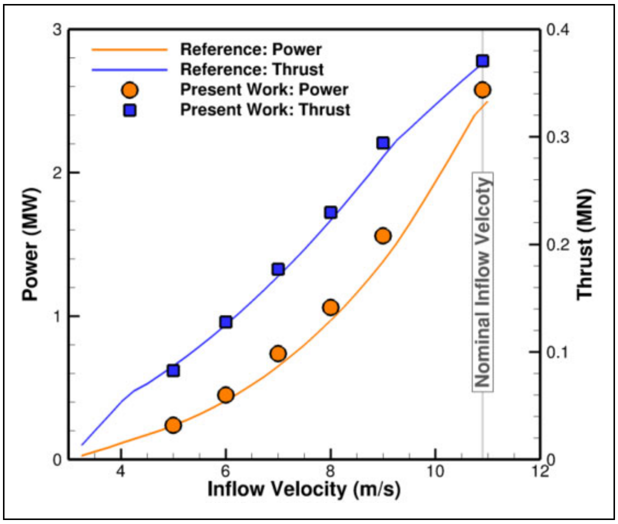

Figure 20. Siemens SWT-2.3-93 power and thrust simulation results using a time step corresponding to a 1/4° rotation. Each time was solved with BDF-2 using 25 sub-iterations for the near-body flow solver. Reference solution data provided by the NREL FAST (Jonkman et al., 2009) software.

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 (NSF NWSC-2 Cheyenne, 2016 (2017)). 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

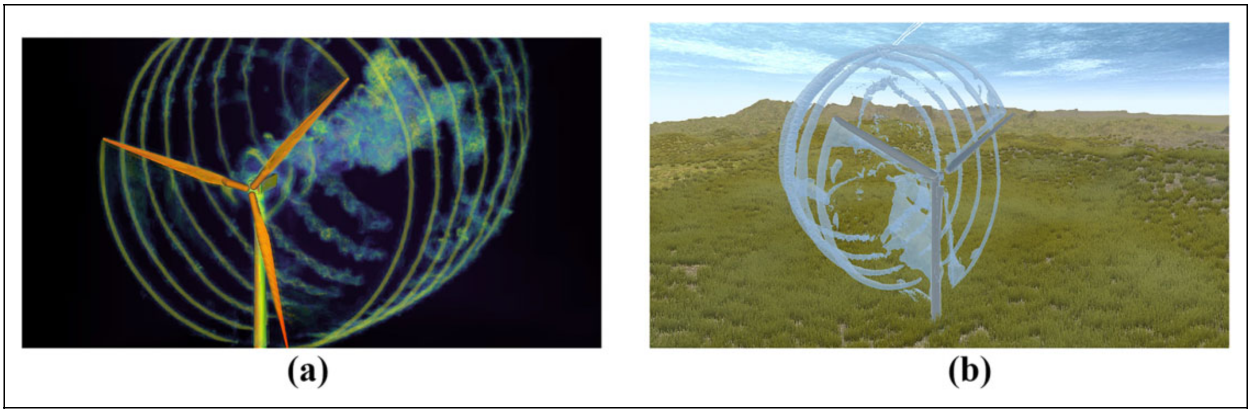

Figure 21. Siemens SWT-2.3-93 wind turbine used for the weak scaling study of the W^2A^2KE3D framework. (a) Volume rendering of vorticity. (b) Iso-surface of velocity magnitude.

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 ADT for efficient domain searching. The oriented bounding box of the ADT's are sent in an all-to-all communication to check for intersections. However, `p4est` partitions cells into non-contiguous groups via *z-order* partitioning (Burstedde et al., 2011). 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 (Burstedde et al., 2011). Using these intersection results, a processor communication map is constructed.

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. Weak scaling wind farm study: Overall performance statistics.[a]

Turbine count	Efficiency	Revs	Near-body cores	Off-body cores	Total cores
6	1.0000	1.374	2088	720	2808
12	0.9874	1.360	4176	1440	5616
24	0.9682	1.331	8352	2880	11,232
48	0.9333	1.283	16,704	5760	22,464
96	0.8686	1.194	33,408	11,520	44,928

[a]Weak scaling wind plant study performed on NWSC-2 Cheyenne (NSF NWSC-2 cheyenne, 2016 (2017)) up to 96 wind turbines for wall-clock time of 9.5 h. Six turbines are used as the perfect scaling reference.

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%. This drop in weak scalability can be attributed to the overset grid assembler as demonstrated in Figure 22.

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

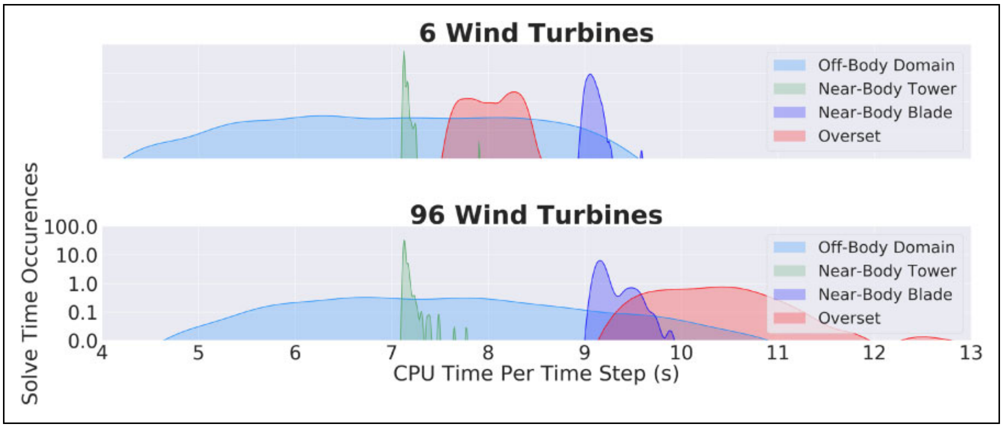

Figure 22. Time-step solve time (seconds) frequency distributions of the 6 and 96 wind turbine cases for the weak scaling study. The CFD solvers must complete the time step before the overset module can interpolate the solutions between meshes therefore placing the execution process into two serial components.

Table 3. Weak scaling wind farm study: Solver performance statistics.[a]

	Blade time (s)			Tower time (s)			Off-body time (s)			Overset time (s)		
Turbine 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

[a] Weak scaling wind plant study solver times up to 96 wind turbines.

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.

For deeper analysis of the 96 wind turbine case, Figure 22 shows the distribution of solver wall-clock times per time step. First, the near-body solve time for the blade mesh distribution (shown in purple) expands when simulating 96 wind turbines. This spread of the wall-clock time distribution from 6 wind turbines to 96 wind turbines 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. Second, the wall-clock time frequency distribution of the off-body solver (shown in blue) demonstrates a wide base due to increasing computational loads over the duration of the simulation caused by dynamic mesh adaption. As the simulation evolves, the flow features increase requiring more mesh resolution, and, thus, more degrees of freedom on the same number of computational resources. Additionally, more parallel communication is required for the 96 wind turbine simulation increasing the occurrences of solve times above 9.5 s. Lastly, the largest degradation in weak scaling is attributed to the overset grid connectivity and data update shown in red in Figure 22. A significant shift occurs in the wall-clock time for the overset grid assembler adding 1–4 s causes a significant drop in performance. The reason for this drop is due to the number of meshes the overset assembler needs to perform connectivity. The 96 wind turbine case contains a total of 385 meshes in contrast to the 6 wind turbine case containing 25 meshes. The mesh connectivity must be performed in relation to all other meshes, in particular, all unstructured meshes connecting to one off-body mesh. Thus, increasing the number of wind turbines increases the complexity of the overset connectivity problem.

Overall, 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 widens for the overset module.

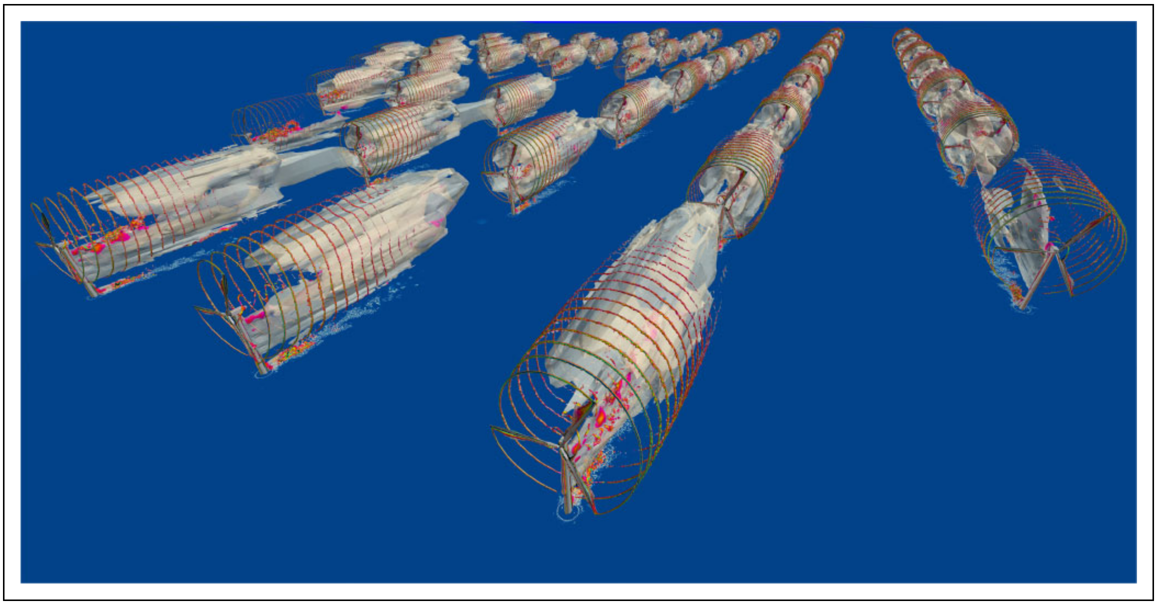

Figure 23. Iso-surfaces of velocity magnitude of the Lillgrund Wind Farm which contains 48 Siemens SWT-2.3-93 wind turbines using the W^2A^2KE3D framework.

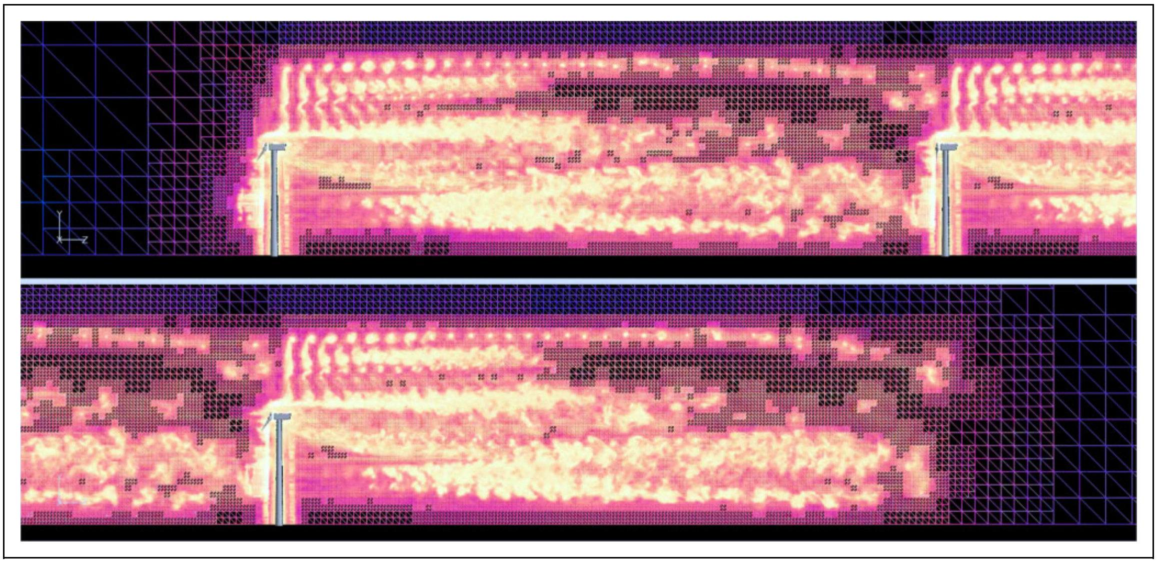

Figure 24. Lillgrund Wind Farm wake structures and adaptive mesh for the Siemens SWT-2.3-93 wind turbine simulated using the W^2A^2KE3D framework.

This indicates a small number of ranks in the large-scale simulation may be throttling the overall performance. In general, overset methods incur scalability issues due to the inherent imbalance in the amount of mesh donor searches described previously. To alleviate these issues, active load balancing techniques found in reference (Roget and Sitaraman, 2014) can be implemented. Furthermore, overall efficiency improvement of the overset grid module is desired since the execution time for dynamic overset grid assembly is of the same magnitude as the flow solver time, while more efficient approaches (Roget and Sitaraman, 2014) demonstrate overset grid assembly one magnitude faster than currently implemented.

5.1. Long run-time 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 23 shows the wind plant configuration with iso-surfaces of velocity magnitude at approximately eight revolutions of rotation. Figure 24 demonstrates a profile

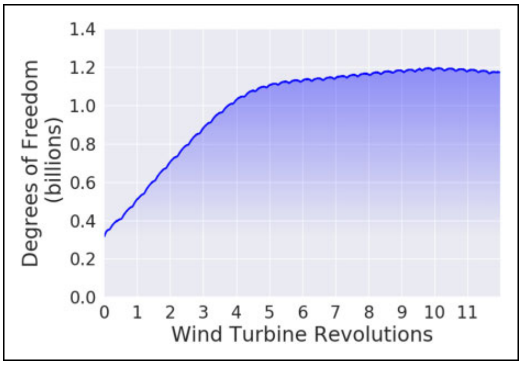

Figure 25. Degree of freedom counts for Lillgrund Wind Farm simulation of the off-body mesh. The initial linear trend from 0 to 4 revolutions corresponds to the start-up wake transients. The second linear trend from 4 to 10 revolutions corresponds to the sustained wake growth and turbulence decay captured by the off-mesh adaptive mesh. The last linear trend from 10 to 12 revolutions represents the interaction of the wakes between wind turbines. The peak at nearly 10 wind turbine revolutions represents the moment when the upstream wind turbine's wake interacts with the downstream wind turbine.

of the AMR pattern in the wake of a wind turbine. The wake structure is tracked well down-stream by the use of adaptive meshes.

Figure 25 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 25. The decreasing linear trend represents the time after which the wakes begin to interact. Under uniform inflow conditions, strong blade tip vortices are formed invoking mesh refinement as demonstrated in Figure 24. 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 vortices compared to wind turbines that do not have impinging wake inflow conditions. This results in fewer elements containing flow features that reach the refinement criterion threshold value. After the wakes impinge on the downstream wind turbines, fewer elements are tagged for refinement resulting in the observed decrease of DOFs.

6. Results: Atmospheric inflow conditions

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 microscale 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 atmospheric solvers: the National Center for Atmospheric Research's (NCAR) *W*eather *R*esearch and *F*orecasting (WRF) (Skamarock et al., 2005) model and the National Renewable Energy Laboratory's (NREL) *S*imulation f*O*r *W*ind *F*arm *A*pplications (SOWFA) (Churchfield et al., 2012) model. SOWFA has been extensively used in simulation of wind plant modeling in various atmospheric conditions (Churchfield et al.) and complex terrains (Han et al.; Han and Stoellinger; Roy and Stoellinger).

Prior to the CFD simulation containing the wind turbines, a precursor atmospheric simulation is performed in problem dependent situations such as specific atmospheric conditions, e.g. turbulence intensity, stable, neutral, and unstable boundary layer, 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, time histories 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. The atmospheric solution is registered as a pseudo-CFD solver with an unstructured grid to TIOGA. During a wind plant simulation, the boundary elements of the off-body mesh system are updated via linear-interpolated-time atmospheric data.

6.1. SOWFA precursor results for neutral ABL

The atmospheric inflow for the Lillgrund Wind Farm is based on the meteorological conditions described in Bergström (2009) and in the LES performed by Churchfield et al. 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% (Bergström, 2009).

The precursor LES domain size is 10,240 m × 4,096 m × 1024 m with a uniform 16 m resolution in all directions resulting in a mesh consisting of 640 × 256 × 64

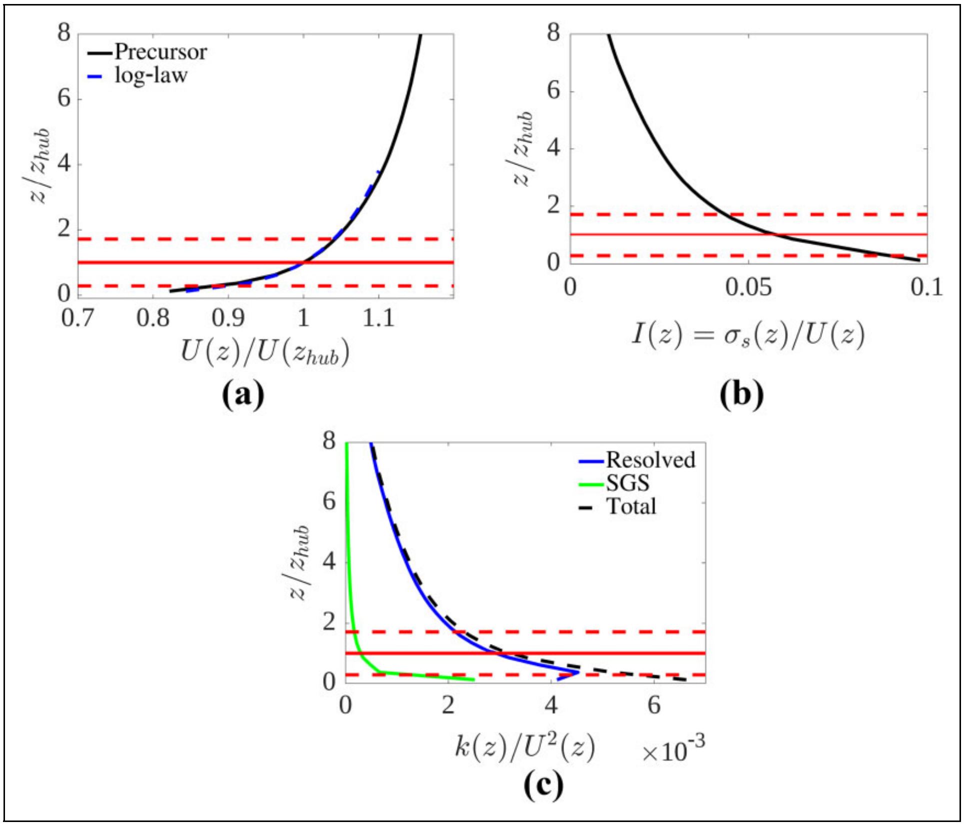

Figure 26. Vertical profiles of temporally and horizontally averaged velocity, turbulence intensity and turbulence kinetic energy from the precursor LES using NREL's SOWFA software. The solid red horizontal line represents the hub height and the two horizontal dashed lines represent the vertical extent of the wind turbine rotor.

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 m 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. and Moeng and Sullivan (1994). 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^2A^2KE3D .

Figure 26 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.

Figure 27 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.

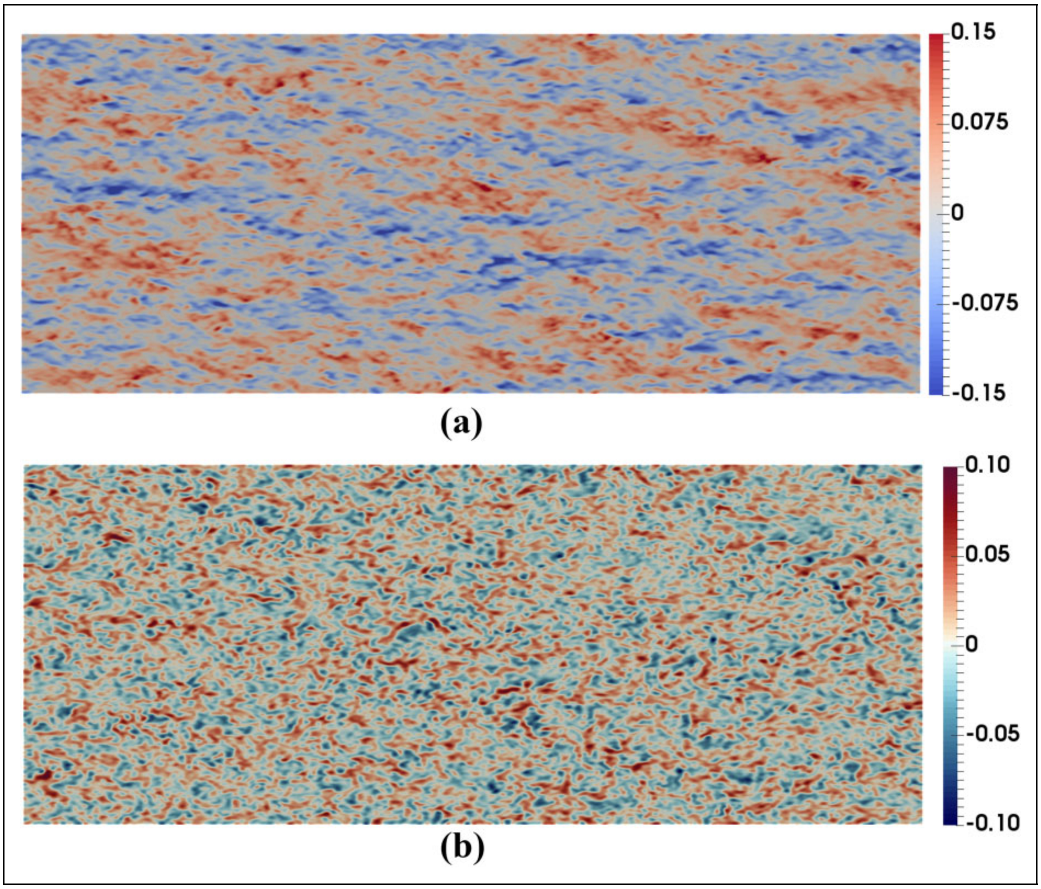

Figure 27. Contours of instantaneous velocity fluctuation at rotor hub height horizontal plane of precursor LES velocity normalized by mean wind speed using NREL's SOWFA software. (a) Wind-wise velocity fluctuations. (b) Vertical velocity fluctuations.

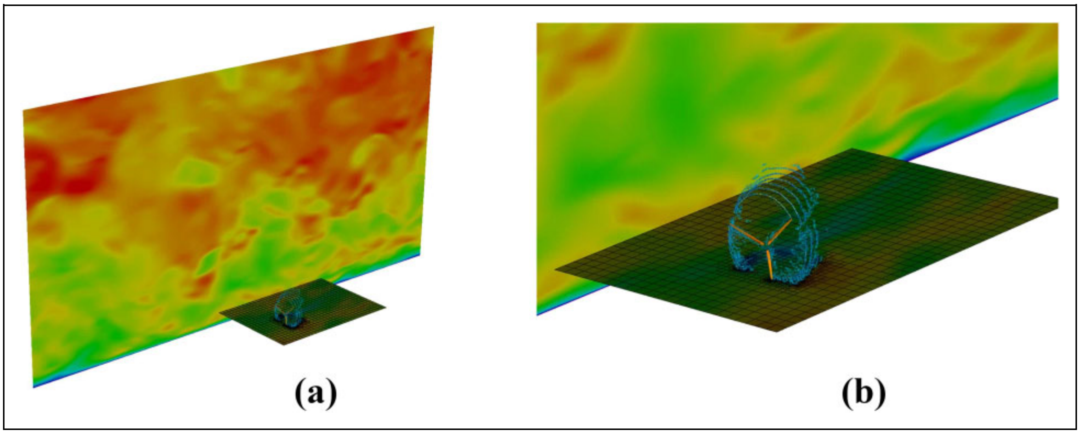

Figure 28. Microscale atmospheric and CFD coupling with NCAR's WRF solver to the off-body CFD solver `dg4est` for a single NREL 5 MW wind turbine. (a) NCAR WRF atmospheric inflow coupling to the off-body solver. (b) Zoomed view of the NCAR WRF inflow.

6.2. Coupled microscale atmospheric and CFD results

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

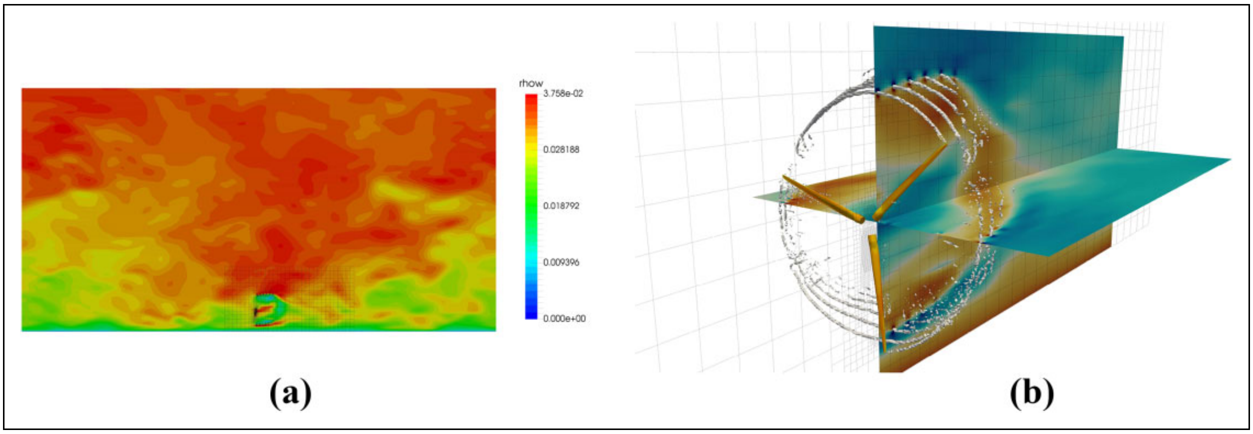

Figure 29. Microscale atmospheric and CFD coupling with NREL's SOWFA solver to the off-body CFD solver `dg4est` for a single NREL 5 MW wind turbine. (a) NREL SOWFA atmospheric inflow coupling to the off-body solver. (b) Zoomed view of the NREL SOWFA inflow.

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. SOWFA serves primary as the ABL LES solver for in this work to produce the inflow for the wind plant CFD simulations. SOWFA enables simulations of complex terrain through the use of unstructured hexahedral grids and a terrain aligned implementation of the Schumann-Grötzbach (Groetzbach and Schumann, 1977; Schumann, 1975) wall model. First results for real complex terrain at the Sierra Madre site in south-central Wyoming and the Bolund hill (Han et al.,; Han and Stoellinger,; Roy and Stoellinger) are promising.

Preliminary work of coupling WRF and SOWFA with the blade resolved wind plant simulation code is shown in Figures 28 and 29, respectively. In particular, Figure 29 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 Sitaraman et al. 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.

7. 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. This approach is forged in methods and simulation technologies that are being investigated by projects within ECP. Specifically, the use of high-order FEMs using tensor-product basis functions for computationally efficient accurate simulation is the primary focus of the Center of Efficient Exascale Discretizations (Brown et al., 2018) (CEED); this work makes use of this numerical discretization and demonstrates is suitability for large scale simulation. Second, this work utilizes the Topology Independent Overset Grid Assembler (TIOGA) which also being used for ECP's ExaWind Project (Turbine Wind Plant Efficiency, 2016). The joint efforts of both teams, herein and ExaWind, have sustained positive algorithm and software development of TIOGA through cooperative development and testing benefiting each team in individual and collaborative goals.

Validation studies were performed for three wind turbines, the NREL 5 MW, 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 W^2A^2KE3D 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. Lastly, preliminary results of the atmospheric coupling with microscale 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. Target physical simulation times will be on the order of minutes which allows the study of performance under changing atmospheric conditions.

To achieve this goal, research into improved computational efficiency and strong scalability of the framework will be conducted. At present, the framework allows for the simulation of one rotor revolution to be simulated in approximately 1.5 hours of wall-clock time at the spatial resolution required to accurately predict the power and thrust of the turbine. To simulate a wind turbine rotating at 16 rpm for a physical duration of 10 minutes, it would require 10 days of wall-clock time. Currently, the overset grid connectivity time is approximately the same duration as the flow solve times thereby effectively doubling the simulation wall-clock time (since it is executed serially with respect to the flow solvers). The present algorithm in TIOGA for locating overset donors in parallel is based on an Alternating Digital Tree search. In contrast, donor search strategies based on Exact Inverse Maps (Roget and Sitaraman, 2014) has shown up to three times speedup over ADT algorithms. By enabling a three times speed up in half of the simulation wall-clock time, a possible 33% overall speedup may be achieved which would reduce the overall simulation time from 10 days to just under 7 days. The long-term wall-clock time target to simulate one rotor revolution is 0.5 hours which corresponds to 3.3 days of wall-clock time for the 10 min physical-time simulation.

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 will be implemented into the off-body solver to append the large-scale atmospheric structures provided by the microscale 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.

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.

Author contributions

ACK code development (CartDG) and manuscript writing. MJB code development (driver and `dg4est`), simulations, analysis, and edited the manuscript. ZY code development (in-situ visualization), simulations, and analysis. RR WRF and SOWFA simulations. BRA mesh generation. MKS WRF and SOWFA simulations. JS code development (TIOGA). DJM code development (NSU3D) and group leader.

Declaration of Conflicting Interests

The author(s) declared no potential conflicts of interest with respect to the research, authorship, and/or publication of this article.

Funding

The author(s) disclosed receipt of the following financial support for the research, authorship, and/or publication of this article: This work is supported in part by ONR Grants N00014-14-1-0045 and N00014-16-1-2737 and by the US 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).

ORCID iD

Michael J Brazell https://orcid.org/0000-0001-9610-6624

References

Bazilevs Y, Hsu MC, Akkerman I, et al. (2011a) 3D simulation of wind turbine rotors at full scale. Part I: Geometry modeling and aerodynamics. *International Journal for Numerical Methods in Fluids* 65(1–3): 207–235.

Bazilevs Y, Hsu MC, Kiendl J, et al. (2011b) 3D simulation of wind turbine rotors at full scale. Part II: Fluid–structure interaction modeling with composite blades. *International Journal for Numerical Methods in Fluids* 65(1–3): 236–253.

Bergström H (2009) *Meteorological Conditions at Lillgrund. Technical Report*. Uppsala University Document. Available at: https://corporate.vattenfall.se/globalassets/sverige/om-vattenfall/om-oss/var-verksamhet/vindkraft/lillgrund/meteorological_conditions.pdf (accessed 5 May 2017).

Brazell MJ and Mavriplis DJ (2013) *3D Mixed Element Discontinuous Galerkin with Shock Capturing*. AIAA Paper 2013-

3064, 21st AIAA Computational Fluid Dynamics Conference, San Diego, CA, June 2013.

Brazell MJ, Kirby AC and Mavriplis DJ (2017) *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.

Brazell MJ, Kirby AC, Sitaraman J, et al. (2016a) *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.

Brazell MJ, Sitaraman J and Mavriplis DJ (2016b) An overset mesh approach for 3D mixed element high-order discretizations. *Journal of Computational Physics* 322: 33–51.

Brown J, Abdelfata A, Camier JS, et al. (2018). ECP Milestone Report Public release of CEED 1.0 WBS 2.2. 6.06, Milestone CEED-MS13. Available at: ceed.exascaleproject.org/docs/ceed-ms13-report.pdf (accessed 1 December 2018).

Burgess N and Mavriplis D (2011) *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, 4–7 January 2011.

Burstedde C, Ghattas O, Gurnis M, et al. (2010) Extreme-scale AMR. In: *Proceedings of the 2010 ACM/IEEE international conference for high performance computing, networking, storage and analysis*, Washington, DC, 13–19 November 2010, pp. 1–12. IEEE Computer Society.

Burstedde C, Wilcox LC and Ghattas O (2011) `p4est`: scalable algorithms for parallel adaptive mesh refinement on forests of octrees. *SIAM Journal on Scientific Computing* 33(3): 1103–1133.

Ceze M and Fidkowski KJ (2014) Drag prediction using adaptive discontinuous finite elements. *Journal of Aircraft* 51(4): 1284–1294.

Churchfield M, Lee S and Moriarty P (2012) *Overview of the Simulator for Offshore Wind Farm Application SOWFA*. Availble at: https://nwtc.nrel.gov/system/files/SOWFA_webinar_05-03-2012.pdf (accessed 5 May 2017).

Churchfield M, Lee S, Moriarty P, et al. (2012) A large-eddy simulations of wind-plant aerodynamics. In: *50th AIAA aerospace sciences meeting including the new horizons forum and aerospace exposition*, Nashville, TN, 9–12 January 2012. Reston, Virigina: American Institute of Aeronautics and Astronautics.

Churchfield M, Wang Q, Scholbrock A, et al. (2016) Using high-fidelity computational fluid dynamics to help design a wind turbine wake measurement experiment. *Journal of Physics: Conference Series* 753: 032009. IOP Publishing.

Cockburn B, Karniadakis GE and Shu CW (2000). *The Development of Discontinuous Galerkin Methods*, Vol. 11. Springer Science & Business Media.

Crabill JA, Sitaraman J and Jameson A (2016) A high-order overset method on moving and deforming grids. In: *AIAA modeling and simulation technologies conference*, Washington, DC, 13–17 June 2016, pp. 3225.

Darmofal DL, Allmaras SR, Yano M, et al. (2013) *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.

Diosady LT and Murman SM. (2014) *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.

Duque EP, Burklund MD and Johnson W. (2003) Navier-Stokes and comprehensive analysis performance predictions of the NREL Phase VI experiment. *Journal of Solar Energy Engineering* 125(4): 457–467.

Fidkowski KJ. (2014) *A high-order discontinuous Galerkin multigrid solver for aerodynamic applications*. PhD Thesis, Massachusetts Institute of Technology.

Fingersh LJ, Simms D, Hand M, et al. (2001) *Wind Tunnel Testing of NREL? Sunsteady Aerodynamics Experiment*. AIAA Paper 2001-35, 20th ASME Wind Energy Symposium and the 39th Aerospace Sciences Meeting, Reno, NV, 2001.

Fleming PA, Gebraad PM, Lee S, et al. (2014) Evaluating techniques for redirecting turbine wakes using SOWFA. *Renewable Energy* 70: 211–218.

Glasby RS, Burgess N, Anderson K, et al. (2013) *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.

Gomez-Iradi S, Steijl R and Barakos G (2009) Development and validation of a CFD technique for the aerodynamic analysis of HAWT. *Journal of Solar Energy Engineering* 131(3): 031009.

Gopalan H, Gundling C, Brown K, et al. (2014) A coupled mesoscale–microscale framework for wind resource estimation and farm aerodynamics. *Journal of Wind Engineering and Industrial Aerodynamics* 132: 13–26.

Groetzbach G and Schumann U (1977) 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. 1411–1419; 1: 14.11–14.19.

Gundling C, Roget B and Sitaraman J (2011) *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.

Gundling C, Roget B, Sitaraman J, et al. (2012) *Comparison of Wind Turbine Wakes in Steady and Turbulent Inflow*. AIAA Paper 2012-899, 50th AIAA Aerospace Sciences Meeting, Nashville, TN, January 2012.

Gundling C, Sitaraman J, Roget B, et al. (2015) Application and validation of incrementally complex models for wind turbine aerodynamics, isolated wind turbine in uniform inflow conditions. *Wind Energy* 18(11): 1893–1916.

Haga T, Gao H and Wang Z (2011) A high-order unifying discontinuous formulation for the Navier-Stokes equations on 3D mixed grids. *Mathematical Modelling of Natural Phenomena* 6(03): 28–56.

Han Y and Stoellinger MK (2017) Large eddy simulation of atmospheric boundary layer flows over complex terrain with varying stability conditions. In: *35th Wind Energy Symposium*, Grapevine, Texas, 9–13 January 2017, pp. 1161.

Han Y, Stoellinger M and Naughton J (2016) Large eddy simulation for atmospheric boundary layer flow over flat and complex terrains. In: *Journal of Physics: Conference Series* 753: 032044. IOP Publishing.

Hand MM, Simms D, Fingersh L, et al. (2001) *Unsteady aerodynamics experiment phase VI: wind tunnel test configurations and available data campaigns*. National Renewable Energy Laboratory, Golden, Colorado, USA.

Hartmann R (2013) Higher-order and adaptive discontinuous galerkin methods with shock-capturing applied to transonic turbulent delta wing flow. *International Journal for Numerical Methods in Fluids* 72(8): 883–894.

Hindenlang F, Gassner GJ, Altmann C, et al. (2012) Explicit discontinuous galerkin methods for unsteady problems. *Computers & Fluids* 61: 86–93.

Isaac T, Burstedde C, Wilcox LC, et al. (2015) Recursive algorithms for distributed forests of octrees. *SIAM Journal on Scientific Computing* 37(5): C497–C531.

Jonkman J, Butterfield S, Musial W, et al. (2009) *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.

Jung YS, Govindarajan B and Baeder JD. (2016) *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.

Kirby AC, Brazell MJ, Mavriplis DJ, et al. (2016) *An Overset Adaptive High-Order Approach for Blade-Resolved Wind Energy Applications*. AHS Forum 72, West Palm Beach, FL, May 2016.

Kirby AC, Mavriplis DJ and Wissink AM (2015) *An Adaptive Explicit 3D Discontinuous Galerkin Solver for Unsteady Problems*. AIAA Paper 2015-3046, 22nd AIAA Computational Fluid Dynamics Conference, Dallas, TX, June 2015.

Li Y, Paik KJ, Xing T, et al. (2012) Dynamic overset CFD simulations of wind turbine aerodynamics. *Renewable Energy* 37(1): 285–298.

Luo H, Baum JD and Löhner R (2008) A discontinuous Galerkin method based on a Taylor basis for the compressible flows on arbitrary grids. *Journal of Computational Physics* 227(20): 8875–8893.

Mavriplis D and Long M (2014) NSU3D results for the fourth AIAA drag prediction workshop. *Journal of Aircraft* 51(4): 1161–1171.

Mavriplis D, Long M, Lake T, et al. (2015) NSU3D results for the second AIAA high-lift prediction workshop. *Journal of Aircraft* 52(4): 1063–1081.

Mavriplis DJ and Mani K (2014) 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.

Mavriplis DJ (2005) 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.

Messina P (2016) Exascale computing project. Exascale Computing Project (17-SC-20-SC), a collaborative effort of the US Department of Energy Office of Science and the National Nuclear Security Administration. Available at: https://exascaleproject.org (accessed 5 May 2017).

Mikkelsen R. (2003) *Actuator Disc Methods Applied to Wind Turbines*. PhD Thesis, Technical University of Denmark.

Moeng CH and Sullivan PP (1994) A comparison of shear- and buoyancy-driven planetary boundary layer flows. *Journal of the Atmospheric Sciences* 51(7): 999–1022.

Moriarty P, Rodrigo JS, Gancarski P, et al. (2014) IEA-task 31 wakebench: Towards a protocol for wind farm flow model evaluation. Part 2: Wind farm wake models. *Journal of Physics: Conference Series* 524: 012185. IOP Publishing.

Noack R (2005) Suggar: a general capability for moving body overset grid assembly. In: *17th AIAA computational fluid dynamics conference*, Toronto, Ontario, Canada, 6–9 June 2005, pp. 5117.

NREL to Lead One Exascale Computing Project (2016) Available at: https://phys.org/news/2016-10-nrel-exascale.html (accessed 5 May 2017).

NSF NWSC-2 Cheyenne, 2016 (2017) *Computational and Information Systems Laboratory. Cheyenne: SGI ICE XA System (Climate Simulation Laboratory)*. Boulder, CO: National Center for Atmospheric Research.

Pape AL and Lecanu J (2004) 3D navier–stokes computations of a stall-regulated wind turbine. *Wind Energy* 7(4): 309–324.

Park MA, Laflin KR, Chaffin MS, et al. (2014) CFL3D, FUN3D, and NSU3D contributions to the fifth drag prediction workshop. *Journal of Aircraft* 51(4): 1268–1283.

Potsdam M and Mavriplis DJ (2009) Unstructured mesh CFD aerodynamic analysis of the NREL phase VI rotor. In: *AIAA Paper 2009-1221, 47th AIAA Aerospace Sciences Meeting*, Orlando, FL, January 2009.

Rai R, Gopalan H, Naughton J, et al. (2012) A study of the sensitivity of wind turbine response to inflow temporal and spatial resolution. In: *50th AIAA Aerospace Sciences Meeting including the New Horizons Forum and Aerospace Exposition*, Nashville, TN, 9–12 January 2012, pp. 819.

Reza Ahrabi B, Anderson WK and Newman JC (2014) 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.

Roget B and Sitaraman J. (2014) Robust and efficient overset grid assembly for partitioned unstructured meshes. *Journal of Computational Physics* 260: 1–24.

Roy R and Stoellinger MK (2017) Large eddy simulation of wind flow over complex terrain: The Bolund Hill case. In: *35th wind energy symposium*, Grapevine, TX, 9–13 January 2017, pp. 1160.

Rudi J, Malossi ACI, Isaac T, et al. (2015) An extreme-scale implicit solver for complex pdes: highly heterogeneous flow in earth's mantle. In: *Proceedings of the international conference for high performance computing, networking, storage and analysis*, ACM, Austin, TX, 15–20 November 2015, pp. 5.

Schreck S (2002) The NREL full-scale wind tunnel experiment introduction to the special issue. *Wind Energy* 5(2–3): 77–84.

Schumann U (1975) Subgrid scale model for finite difference simulations of turbulent flows in plane channels and annuli. *Journal of Computational Physics* 18: 376–404.

Shur ML, Strelets MK, Travin AK, et al. (2000) Turbulence modeling in rotating and curved channels: assessing the Spalart-Shur correction. *AIAA Journal* 38(5): 784–792.

Simms DA, Schreck S, Hand M, et al. (2001) *NREL Unsteady Aerodynamics Experiment in the NASA-Ames Wind Tunnel: A Comparison of Predictions to Measurements*. National Renewable Energy Laboratory Golden, CO, USA, 2001.

Sitaraman J, Mavriplis DJ and Duque EP (2014) Wind farm simulations using a full rotor model for wind turbines. In: *32nd ASME wind energy symposium*, National Harbor, MD, 13–17 January 2014, pp. 1086.

Skamarock WC, Klemp JB, Dudhia J, et al. (2005) *A Description of the Advanced Research WRF Version 2*. Technical report, DTIC Document.

Smagorinsky J. (1963) General circulation experiments with the primitive equations: I. the basic experiment. *Monthly Weather Review* 91(3): 99–164.

Sørensen NN, Michelsen J and Schreck S (2002) Navier-Stokes predictions of the NREL Phase VI rotor in the NASA Ames 80 ft× 120 ft wind tunnel. *Wind Energy* 5(2–3): 151–169.

Spalart P and Allmaras S (1992) A one-equation turbulence model for aerodynamic flows. In: *30th aerospace sciences meeting and exhibit*, Reno, NV, 6–9 January 1992, pp. 439.

Spalart PR, Deck S, Shur M, et al. (2006) A new version of detached-eddy simulation, resistant to ambiguous grid densities. *Theoretical and Computational Fluid Dynamics* 20(3): 181–195.

Steger JL (1983) A chimera grid scheme; advances in grid generation. *American Society Of Mechanical Engineers Fluids Engineering Division* 5: 55–70.

Takizawa K, Henicke B, Tezduyar TE, et al. (2011) Stabilized space–time computation of wind-turbine rotor aerodynamics. *Computational Mechanics* 48(3): 333–344.

Top 500, the list: November 2016 (2016) Available at: https://www.top500.org/lists/2016/11/ (accessed 5 May 2017).

Troldborg N, Sørensen JN and Mikkelsen R (2007) Actuator line simulation of wake of wind turbine operating in turbulent inflow. *Journal of Physics: Conference Series* 75(1): 012063. Available at: http://stacks.iop.org/1742-6596/75/i=1/a=012063.

Turbine Wind Plant Efficiency (2016) Available at: http://www.nrel.gov/news/press/2016/37739 (accessed 5 May 2017).

Umbach F. (2010) Global energy security and the implications for the EU. *Energy Policy* 38(3): 1229–1240.

Vincent P, Witherden FD, Farrington AM, et al. (2015) PyFR: next-generation high-order computational fluid dynamics on many-core hardware. In: *22nd AIAA computational fluid dynamics conference*, Dallas, TX, 22–26 June 2015, pp. 3050.

Wang L and Mavriplis DJ. (2009) Adjoint-based h–p adaptive discontinuous Galerkin methods for the 2D compressible Euler equations. *Journal of Computational Physics* 228(20): 7643–7661.

Wang L, Anderson WK, Erwin JT, et al. (2014) Discontinuous Galerkin and Petrov Galerkin methods for compressible viscous flows. *Computers & Fluids* 100: 13–29.

Whitlock B, Favre JM and Meredith JS (2011) Parallel in situ coupling of simulation with a fully featured visualization system. In: *Proceedings of the 11th Eurographics conference on Parallel Graphics and Visualization*, Aire-la-Ville, Switzerland, 10–11 April 2011, pp. 101–109. Eurographics Association.

Wilcox DC (1988) Reassessment of the scale-determining equation for advanced turbulence models. *AIAA Journal* 26(11): 1299–1310.

Wind Vision (2015) *A New Era for Wind Power in the United States*. Technical report, US Department of Energy, Washington, DC, 2015.

Wissink AM, Hornung RD, Kohn SR, et al. Large scale parallel structured AMR calculations using the SAMRAI framework. In: *Supercomputing, ACM/IEEE 2001 conference*, Denver, CO, 10–16 November 2001, pp. 22–22. IEEE.

Wissink AM, Jayaraman B and Sitaraman J An assessment of the dual mesh paradigm using different near-body solvers in HELIOS. In: *55th AIAA aerospace sciences meeting*, Grapevine, TX, 9–13 January 2017, pp. 0287.

Wissink AM, Potsdam M, Sankaran V, et al. (2016) A dual-mesh unstructured adaptive cartesian computational fluid dynamics approach for Hover prediction. *Journal of the American Helicopter Society* 61(1): 1–19.

Witherden FD, Vermeire BC and Vincent PE (2015) Heterogeneous computing on mixed unstructured grids with PyFR. *Computers & Fluids* 120: 173–186.

Yelmule MM and Vsj EA. (2013) CFD predictions of NREL Phase VI rotor experiments in NASA/Ames wind tunnel. *International Journal of Renewable Energy Research (IJRER)* 3(2): 261–269.

Zahle F, Sørensen NN and Johansen J. (2009) Wind turbine rotor-tower interaction using an incompressible overset grid method. *Wind Energy* 12(6): 594–619.

Author biographies

Andrew C Kirby holds a PhD in Mechanical Engineering from the University of Wyoming and an MS in Applied Mathematics from Columbia University. He is a Postdoctoral Associate at the University of Wyoming researching high-performance numerical algorithms for the field of Computational Fluid Dynamics.

Michael J Brazell is a Postdoctoral Research Scientist in the CFD Laboratory at the University of Wyoming. He holds a PhD in Mechanical Engineering from Clarkson University. His research focuses on the development of high-order finite element methods for both continuous and discontinuous formulations. Applications of his research include indoor air-quality, aerodynamics, and wind energy.

Zhi Yang is a research scientist at the CFD Laboratory of the University of Wyoming. His research is focused on unsteady aeroelastic simulations methods.

Rajib Roy is a PhD Candidate at the University of Wyoming. His research interests include Large Eddy Simulations of atmospheric boundary layer flow in complex terrains, turbulence modeling, and kinetic energy preserving algorithms.

Behzad R Ahrabi is a research scientist at the CFD Laboratory of the University of Wyoming. Currently, his work has focused on the development of robust, scalable, and multilevel parallel algorithms for simulation of turbulent flows on unstructured grids. He is the developer of HOMA solver, a high-order finite-element framework for simulation of airflow over 3D aircraft configurations.

Michael K Stoellinger is currently an Associate Professor in the Department of Mechanical Engineering at the University of Wyoming. His research interest is in turbulence modeling including heat transfer and chemical reactions for engineering problems.

Jay Sitaraman holds a PhD in Aerospace Engineering obtained at the University of Maryland in 2003. He currently heads a research firm with contracts from DoD and DoE to develop numerical simulation software for modeling complex unsteady flows in rotorcraft and wind turbines. In the past, he held positions as a senior research scientist at the National Institute of Aerospace and Associate Professor at the University of Wyoming.

Dimitri J Mavriplis is the Max Castagne Professor in the Department of Mechanical Engineering at the University of Wyoming. He holds a PhD in Mechanical and Aerospace Engineering obtained at Princeton University. He has held previous positions at the National Institute of Aerospace and at ICASE, NASA Langley Research Center.