ExaWind

A software framework for predictive wind plant simulations

Shreyas Ananthan
National Renewable Energy Laboratory

14th Symposium on Overset Composite Grids and Solution Technology
October 1-4, 2018
University of Maryland College Park, MD

ExaWind team

NREL — National Renewable Energy Laboratory

- S. Ananthan
- M. Churchfield
- E. Fabiano
- K. Gruchalla
- S. Hammond
- M. Henry de Frahan
- M. Lawson
- J. Rood
- A. Sharma
- **M. Sprague (PI)**
- K. Swirydowicz
- S. Thomas
- S. Yellapantula

Sandia National Laboratories

- **P. Crozier (Co-PI)**
- C. Dohrmann
- S. Hammond
- J. Hu
- R. Knaus
- J. Overfelt
- S. Rajamanickam
- P. Sakeivich
- A. Williams

Oak Ridge National Laboratory

- J. Turner
- A. Prokopenko
- L. Capone

TEXAS — The University of Texas at Austin

- R. Moser
- J. Melvin

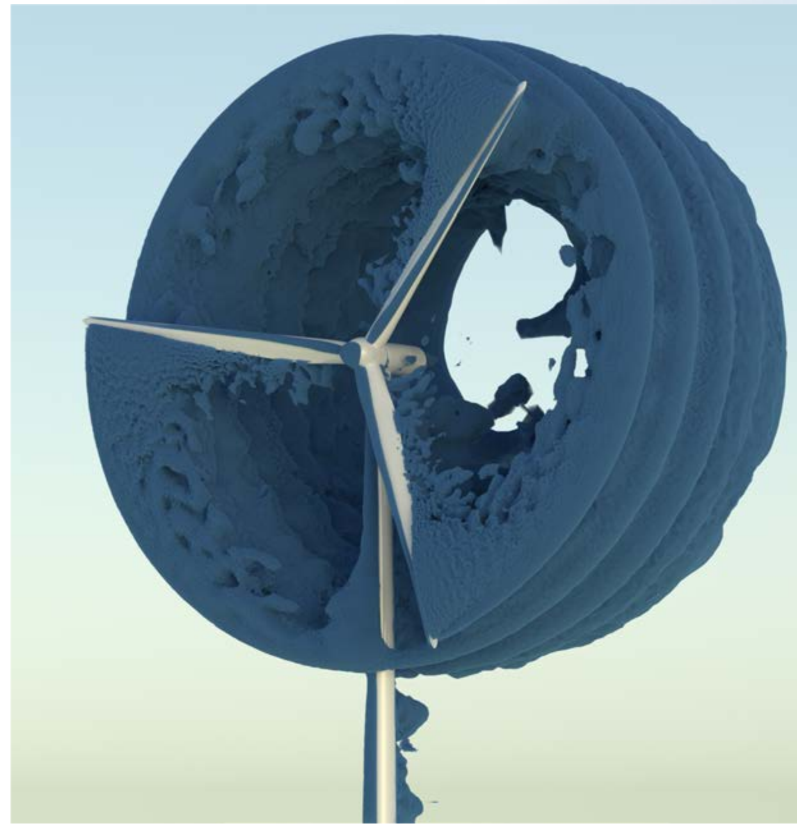
Parallel Geometric Algorithms LLC

- J. Sitaraman

ECP Exascale Computing Project

Outline

- Motivation
- ExaWind Software Stack
  - CFD Solver overview
  - Sliding/overset mesh implementation
- Recent developments
  - ABL and actuator line simulations
  - Linear solver scaling
  - Overset simulations



Motivation

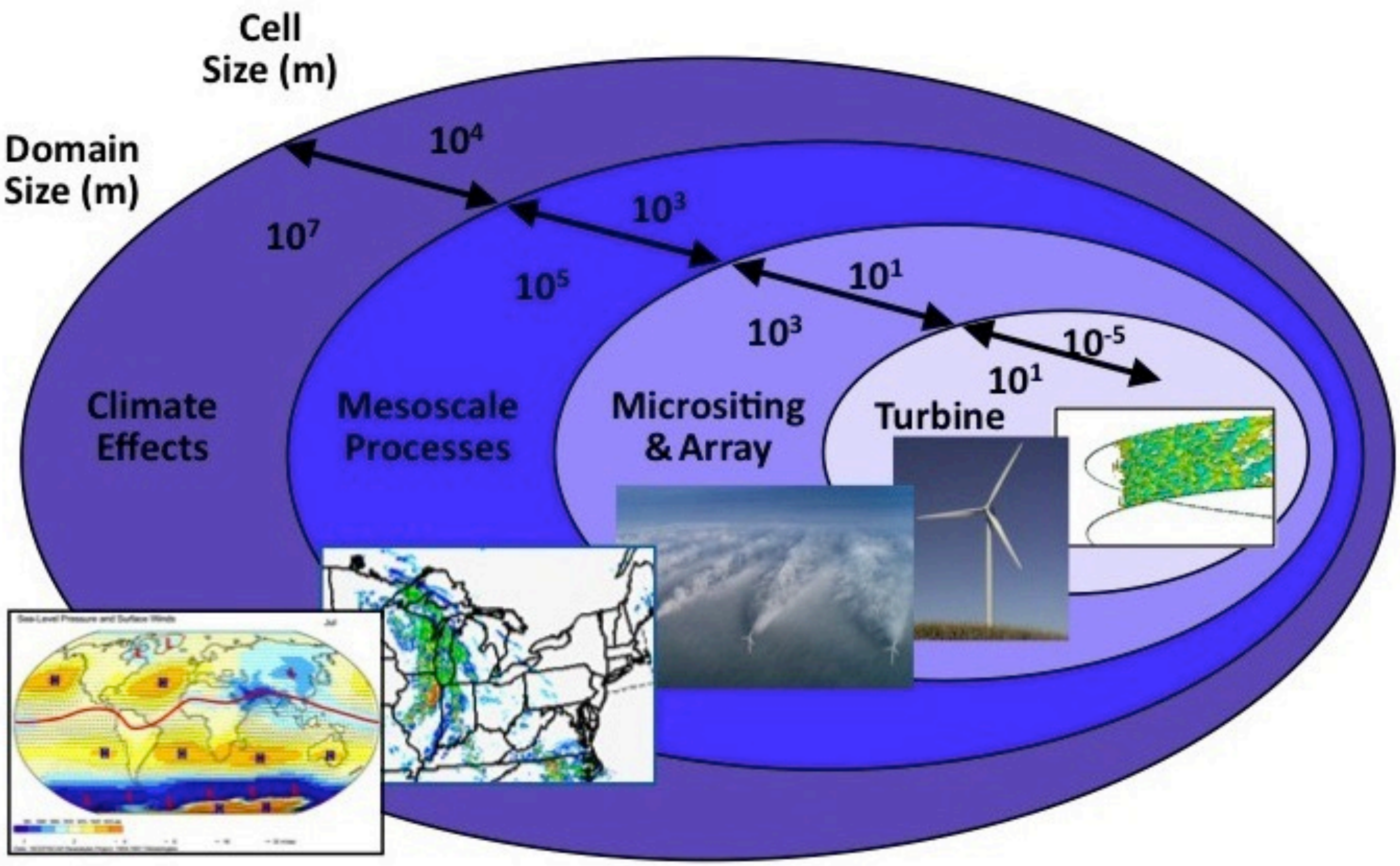
Motivation for *predictive* wind plant simulations

- Industry seems to have the *isolated-single-turbine-in-a-flat-field* problem "solved"
- Wide-scale deployment and cost competitiveness require **large wind plants** – tens to hundreds of turbines in complex terrain
- Wind plants face challenges not seen by an isolated turbine:
  - Plant-level losses are about 20%, and much higher in complex terrain; due to poorly understood turbine-wake interactions
  - Turbine failure rates are significantly higher in wind farms
  - Large uncertainty in plant performance; increases financing costs

Wind farm in Macarthur, Australia; 140 3MW turbines https://www.vestas.com/

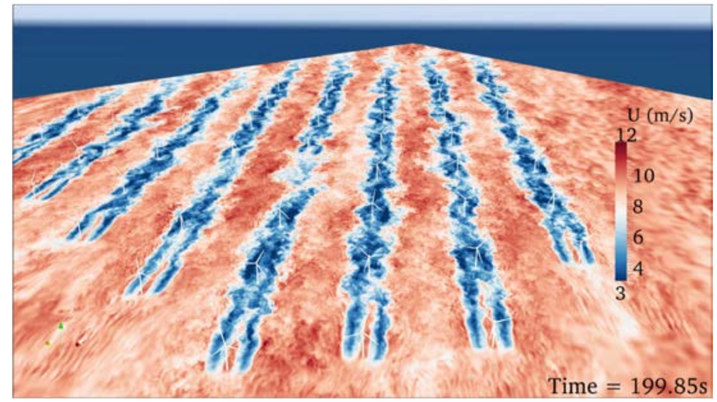
EXASCALE COMPUTING PROJECT

The wide range of spatial and temporal scales involved pose a huge challenge to predictive wind plant simulations



ECP EXASCALE COMPUTING PROJECT

Relevant physics – plant scale to turbine scale

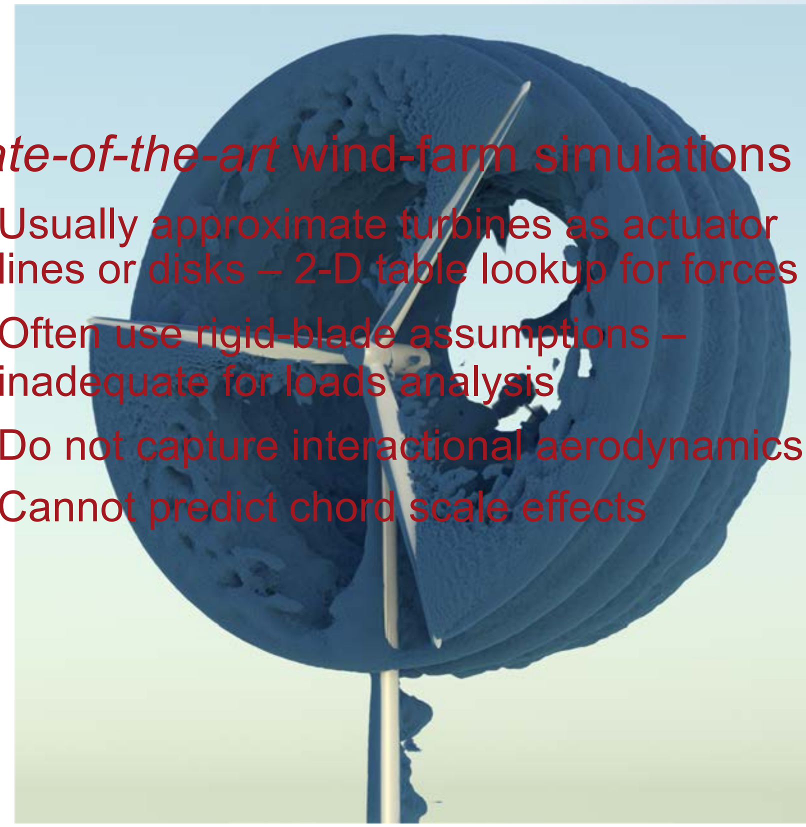


SOWFA actuator line simulation of the Lilgrund wind farm (Churchfield, NREL)

- Atmospheric Boundary Layer (ABL)
  - Stability state – neutral, stable, unstable
  - Surface conditions – roughness, heat-flux, etc.
  - Complex terrain, air-sea interface for offshore
  - Bursts, K-H waves, ramps, frontal passages, thunderstorms
- Wakes
  - Wake formation, evolution, and dissipation under ABL turbulence
  - Turbine wake interaction
- Large wind plant aerodynamics
  - Vertical momentum flux, deep array and blockage effects

Relevant physics – turbine and chord scale

- Turbine scale
  - Blade and tower deformations
  - Fluid-structure interactions (FSI)
  - Interactional aerodynamics – tower-rotor-nacelle
  - Wake formation and interaction with downstream turbines
- Chord scale
  - Unsteady inflow effects on blade boundary layer
  - Rotational augmentation and dynamic stall
  - Blade flow control
  - Icing effects



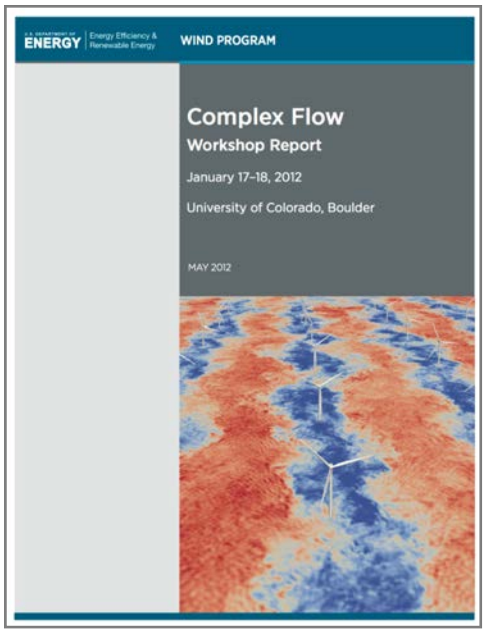
- *State-of-the-art* wind-farm simulations
  - Usually approximate turbines as actuator lines or disks – 2-D table lookup for forces
  - Often use rigid-blade assumptions – inadequate for loads analysis
  - Do not capture interactional aerodynamics
  - Cannot predict chord scale effects

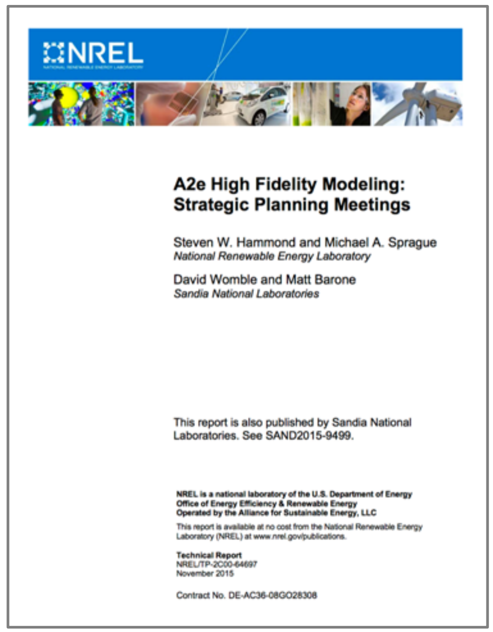
Development of R&D strategy

EXASCALE COMPUTING PROJECT

Multiple pathways to successful wind plant simulations

- R&D plan developed through stakeholder engagement in 3 workshops (2012-2015)
- Problem definition, knowledge gaps, challenges and opportunities documented

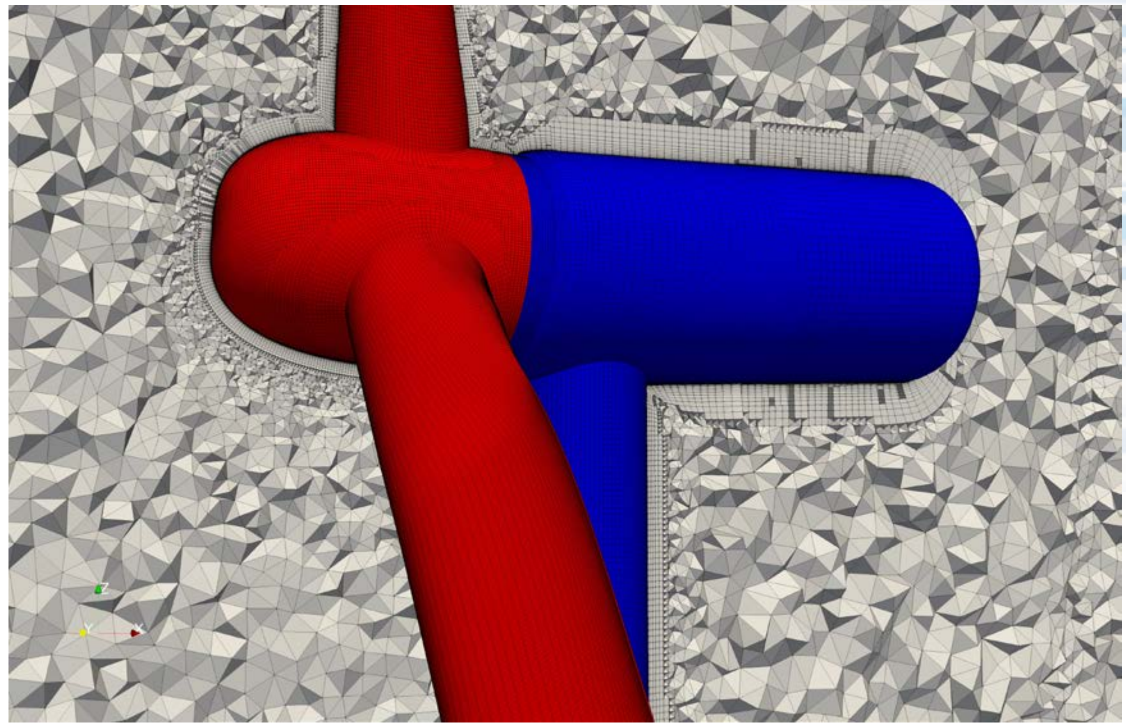




- https://www.energy.gov/sites/prod/files/2013/12/f5/complex_flow_workshop_report.pdf
- https://www.nrel.gov/docs/fy16osti/64697.pdf
- https://www.nrel.gov/docs/fy17osti/67648.pdf

Requirements for *predictive* wind plant simulations

- Resolve turbines and wakes
  - O(10)B grid points
  - Possible need for high-order methods
- Arbitrary mesh motion
  - Overset and/or sliding meshes
  - Mesh deformation (FSI)
- Appropriate turbulence models
  - DNS is impossible
  - Hybrid RANS/LES required
- Coupling to mesoscale models (e.g., WRF)



M. Lawson, NREL (2018)

Other requirements

- *Develop* in an open-source community model whose contributors and users will be from national labs, university, and industry
- *Codify* state-of-the-art numerical algorithms for predictive simulations
- *Capable* of high-fidelity, performance, and scalability on DOE machines
- *Portable* across wide range of systems – especially future exascale machines
- *Leverage* existing assets where appropriate develop new resources as needed
- *Demonstrate* credibility via standard benchmark problems in an open benchmarking exercise through community engagement

Predictive plant simulations – 10 year challenge problem

Predictive simulation of the complex flow physics within a wind plant composed of O(100) MW-scale wind turbines sited within a 10 km x 10 km area with complex terrain, involving simulations with O(100) billion grid points.

Goals/motivation for predictive simulations:

- Advance our fundamental understanding of the flow physics governing whole wind plant performance
- Create modeling and simulation capability that will revolutionize the design and control of wind farms
- Advance our ability to predict the response of wind farms to a wide range of atmospheric conditions

Proposed pathway:

- Acoustically incompressible (low-Mach) flow equations
- Hybrid RANS/LES turbulence model
- Unstructured sliding/overset mesh

Towards predictivity – A2e HFM and ExaWind projects

DOE is funding two closely coordinated projects directed towards predictive wind farm simulations:

DOE Wind Energy Technologies Office (WETO) A2e High-Fidelity Modeling (HFM) Project:

- Create a modeling capability, with multiple levels of fidelity, that is backed by rigorous V&V (with UQ)

ExaWind DOE Exascale Computing Project:

- Ensure that the HFM capability is highly scalable on next-generation architectures and exascale systems

ExaWind – Simulation Framework

EXASCALE COMPUTING PROJECT

ExaWind – primary application codes

- Nalu-Wind (https://github.com/ExaWind/nalu-wind)
  - Wind-specific fork of NaluCFD/Nalu
  - Low-Mach Navier-Stokes solver
  - Unstructured finite-volume discretization
  - C++11 code built on Trilinos
  - Blade-resolved and actuator line (AL) models for turbines
  - *k-ω SST, SST-DES, k-SGS, WALE and Smagorinsky models*

- OpenFAST (https://github.com/OpenFAST/openfast)
  - Full-turbine FEM simulation code
  - Onshore and offshore turbine dynamics
  - Third-party controller integration
  - Fortran 90 code with C++ API for CFD-CSD coupling

Nalu-Wind – Software libraries used

- **Trilinos** (https://github.com/trilinos/trilinos)
  - **Sierra ToolKit (STK)** – Unstructured mesh database and domain decomposition
  - **Belos/Ifpack2** – Linear solvers and preconditioners
  - **Tpetra** – Sparse matrix and vector interface
  - **MueLu** – Smoothed-aggregated Algebraic Multigrid (SA-AMG)
  - **Kokkos** – Next-gen platform (NGP) abstraction library for portability
- **HYPRE** (https://github.com/LLNL/hypre)
  - BoomerAMG (classic Ruge-Stüben AMG), krylov solvers, and matrix data
- **TIOGA** (https://github.com/jsitaraman/tioga)
  - Library for overset grid assembly on parallel distributed systems
- **Paraview Catalyst** (https://www.paraview.org/in-situ/)
  - In-situ visualization library

ExaWind – software engineering

- Code development coordinated via Git SCM – feature branches and pull requests
- Automated build using Spack (https://github.com/LLNL/spack)
  – Reproducible consistent builds (including dependencies) on multiple machines
  – NREL Peregrine and Rhodes, NERSC Cori, ANL Mira, ORNL SummitDev, Macbooks
  – Build scripts available on Github – https://github.com/exawind/build-test
- Automated nightly regression testing and unit testing (GoogleTest, Ctest & CDash)
  – Test matrix – [Linux, OS X] $\times$ [GCC, Clang, Intel]
  – Memory sanitizer check on Linux with Clang compiler build
  – Updated nightly here – https://my.cdash.org/index.php?project=Nalu-Wind
- Continuously updated code documentation using Sphinx (http://www.sphinx-doc.org)
  – Available online at ReadTheDocs – https://nalu-wind.readthedocs.io
  – Documentation tracked in Git alongside the source code – matches latest code version

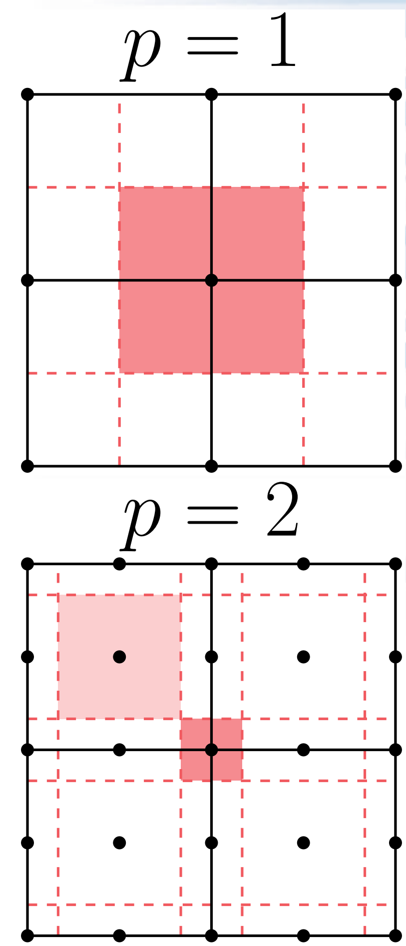
Nalu-Wind – Atmospheric Boundary Layer (ABL) model

$$\underbrace{\frac{\partial}{\partial t}(\bar{\rho}\,\widetilde{u}_i)}_{\mathbf{I}} + \underbrace{\frac{\partial}{\partial x_j}(\bar{\rho}\,\widetilde{u}_i\widetilde{u}_j)}_{\mathbf{II}} = -\underbrace{\frac{\partial p'}{\partial x_j}\delta_{ij}}_{\mathbf{III}} - \underbrace{\frac{\partial \tau_{ij}}{\partial x_j}}_{\mathbf{IV}} - \underbrace{2\bar{\rho}\,\epsilon_{ijk}\,\Omega_j u_k}_{\mathbf{V}} + \underbrace{(\bar{\rho}-\rho_\circ)\,g_i}_{\mathbf{VI}} + \underbrace{S_i^u}_{\mathbf{VII}} + \underbrace{f_i^T}_{\mathbf{VIII}}$$

- *Boussinesq approximation* for buoyancy (Term VI); plans to explore anelastic approach
- *Coriolis forces* (Term V) implemented as source terms
- Term VII represents forcing to drive the flow to desired profile
- Term VIII represents body forces when modeling actuator lines and/or disks
- *k-SGS* and standard Smagorinsky LES turbulence models
- Wall models for surface stress and heat flux at terrain

Nalu-Wind – Spatial Discretization

- Unstructured mesh support via *Sierra ToolKit (STK)*
  - Hex, tet, wedge, and pyramid topologies
- Uses a *dual-mesh* approach
- Two finite-volume discretizations
  - Edge-based vertex centered (EBVC)
  - Control volume FEM (CVFEM)
  - Both second order accurate (for p=1)
- CVFEM more expensive, but more accurate for *poor quality meshes*
- Higher-order pathway with CVFEM
  - Need to explore *mixed-order* meshes for hybrid RANS/LES



Nalu-Wind – Next-gen Platform (NGP) Strategy

- Platform portability provided by Kokkos library
  - High-level looping constructs that generates optimized code for target architecture
  - Memory view abstraction for data management
  - Can target pthreads, OpenMP, and CUDA as backend programming models
  - Math kernels via kokkos-kernels
  - Used extensively by Trilinos solver libraries
- STK NGP and STK SIMD
  - STK NGP library provides mesh data management on GPUs
  - STK SIMD used in Nalu to target SIMD vectorization

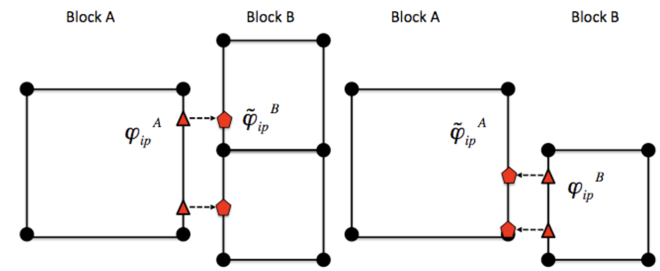
Speedup-factors for momentum-matrix assembly on Cori Haswell and KNL partitions. Speedups are calculated over baseline code without any Kokkos threading or SIMD vectorization for an open-jet benchmark problem.

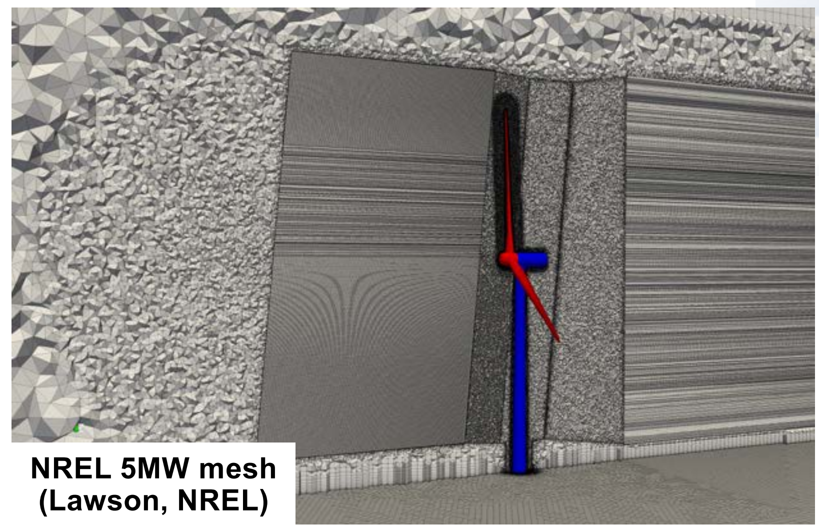
Mesh (# elements)	Haswell	KNL
Coarse P=1 (17.5M)	2.28	1.96
Fine P=1 (140M)	2.35	1.90
Coarse P=2 (17.5M)	2.47	4.60

Nalu-Wind – Sliding Mesh Algorithm

- Implemented using a non-conformal hybrid CVFEM/DG approach
- Face pairs identified using STK Search library
- All sub-grids solved in a single linear system
  - Full connectivity for element pairs appear in linear system
  - Requires linear system reinitialization for moving mesh problems

From Nalu Theory Manual (https://nalu-wind.readthedocs.io)



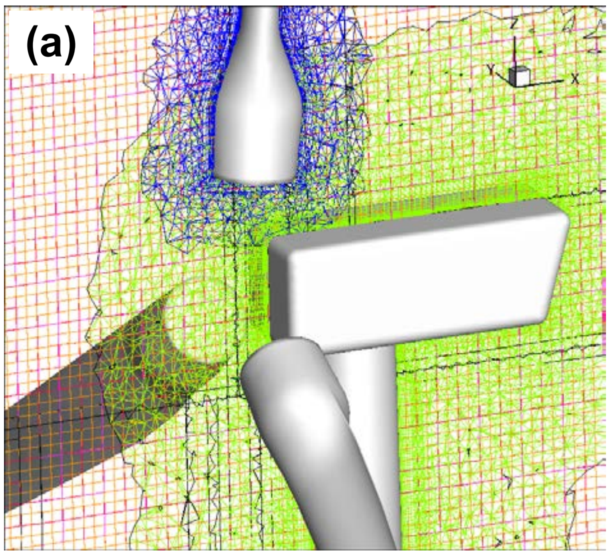


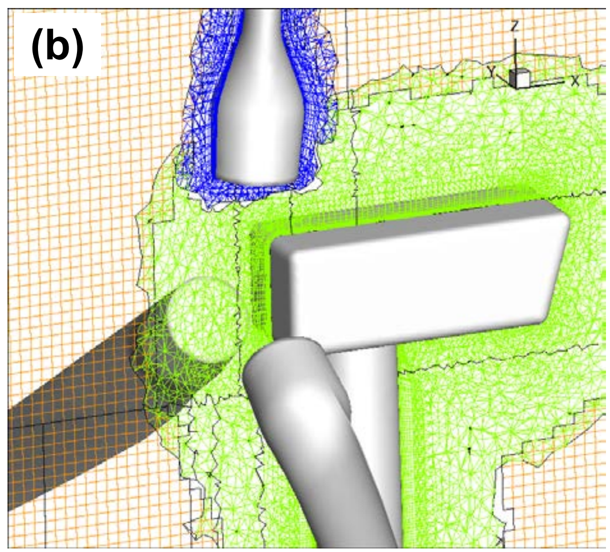
NREL 5MW mesh (Lawson, NREL)

Domino, S. P., Design-order, non-conformal low-Mach fluid algorithms using a hybrid CVFEM/DG approach, *Journal of Computational Physics*, Vol. 359, pp. 331-351, 2018.

Nalu-Wind – Overset Implementation

- Overset connectivity through TIOGA integration
- Monolithic solve of the entire computational domain
  - Fringe interpolations performed using underlying FEM basis functions within Nalu
  - *Fringe* nodes appear as constraint rows in linear system
  - Requires linear system reinitialization for moving mesh applications





Overset NREL 5MW mesh.
(a) Before OGA, (b) after OGA.
(J. Sitaraman)

Motivation for overset in ExaWind

- Sliding mesh – pros and cons
  - Lower search overhead compared to overset grid assembly (2-D contact search)
  - Higher meshing overhead – ensure smooth contact especially for P=1 elements
  - Complicated nested sliding interfaces to account for complex turbine motion
  - Must be used with care with large deformations (ALE + sliding mesh)

- Overset mesh
  - Can handle large deformations and arbitrary turbine motion
    - Blade pitch, flap, and lag motions
    - Nacelle yaw – changing wind direction and novel control paradigms
    - Easier to handle blade deflections within an overset paradigm
  - Test and refine single turbine mesh and extend to full wind farm simulations

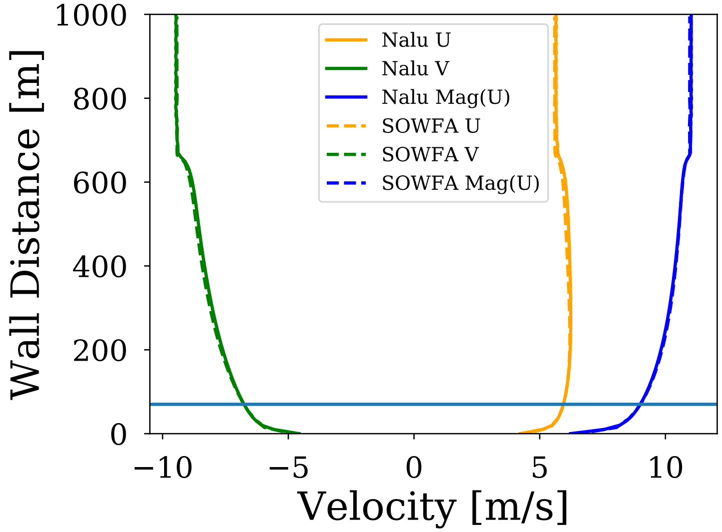
Current challenges to overset grid simulations in ExaWind

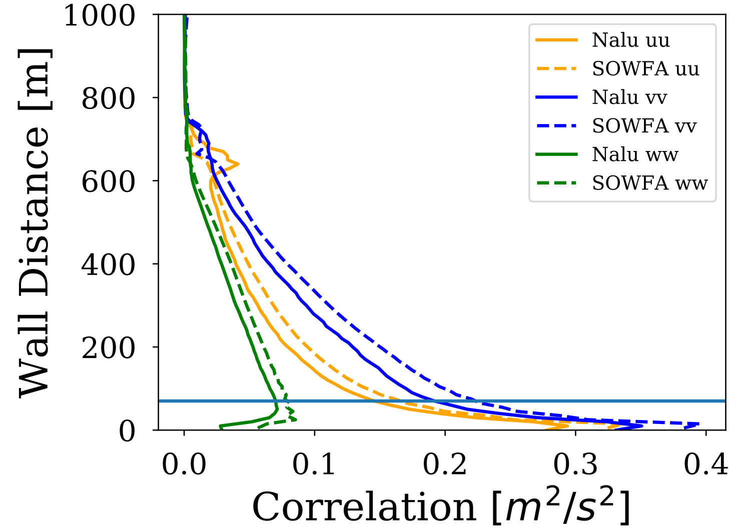
- Strong scaling of elliptic pressure Poisson solves
- AMG preconditioner in presence of overset fringe (constraint) rows
  - Both SA-AMG and C-AMG have issues when dealing with constraint rows
  - Investigating optimal constraint row elimination approaches for unstructured meshes
  - Investigating robustness of *decoupled* Poisson solves for continuity equation system
- Moving mesh applications
  - Parallel overset search connectivity costs at each timestep
  - Linear system reinitialization, preconditioner setup costs at each timestep
- High-aspect ratio meshes
  - Challenge for CVFEM discretization; exploring alternate stabilization approaches

Recent developments – ABL Simulations

EXASCALE COMPUTING PROJECT

ABL precursor simulations – 9x9x1 km domain

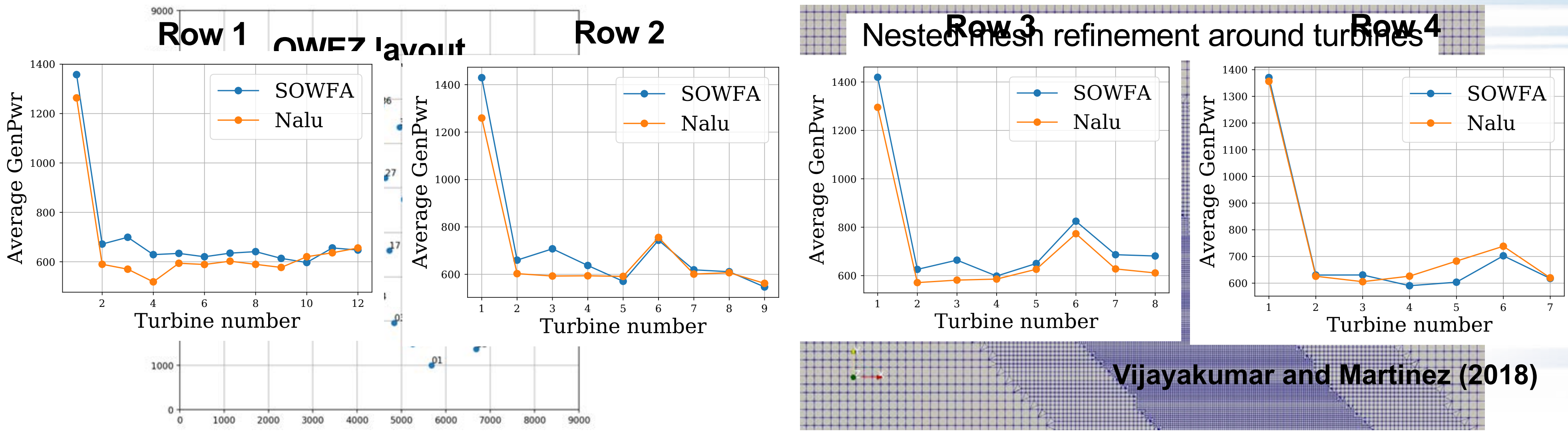




Yellapantula, Vijayakumar, Martinez (2018)

- Implemented baseline SOWFA capability in Nalu
- Differences in velocity correlations may be due to model differences
  – First grid points are different in the two codes (node vs cell centered)
  – Exploring different sampling height approach for ABL wall functions

Actuator line simulations – OWEZ wind farm



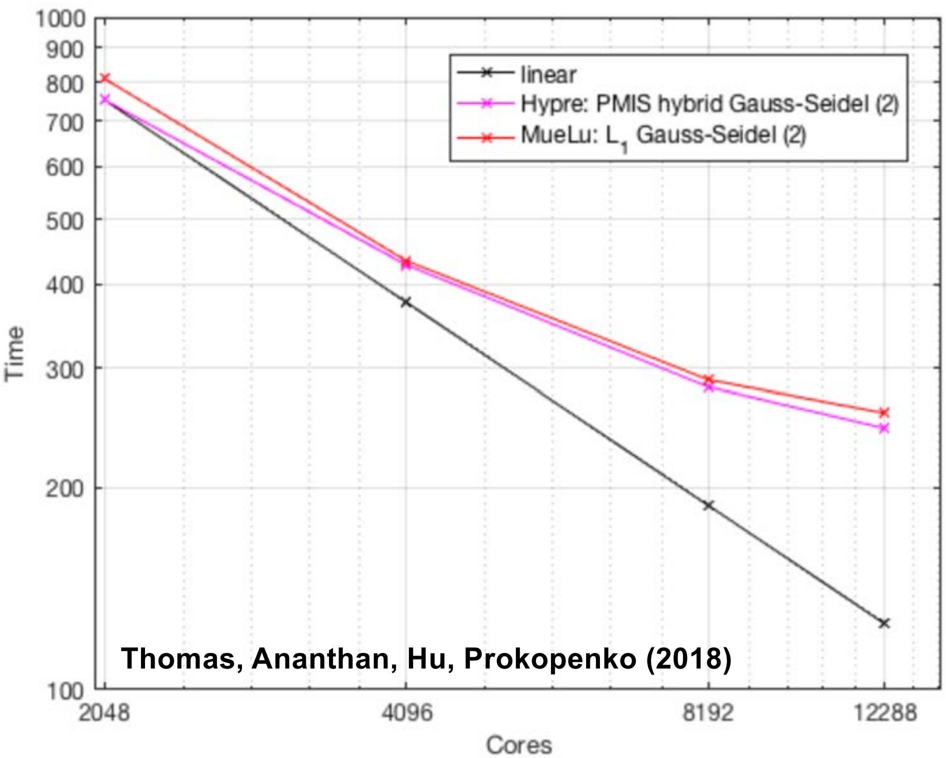
- OWEZ wind farm – 36 Vestas V90 turbines in four rows along wind direction
- Actuator line simulations performed by coupling with OpenFAST
- Conformal nested mesh refinement (total of 153M grid points) – 10m to 2.5m
- Reasonable agreement between two codes

EXASCALE COMPUTING PROJECT

Recent developments – Linear solvers

EXASCALE COMPUTING PROJECT

Strong scaling – flat-MPI results



- V27 medium sliding mesh; 229M pressure DOF
- Performed on NERSC Cori Haswell
- Momentum: Trilinos/Belos
- Pressure: **Hypre or Trilinos/MueLu**
- Scaling shown to 18.6K nodes/core
- https://github.com/exawind/solver-performance

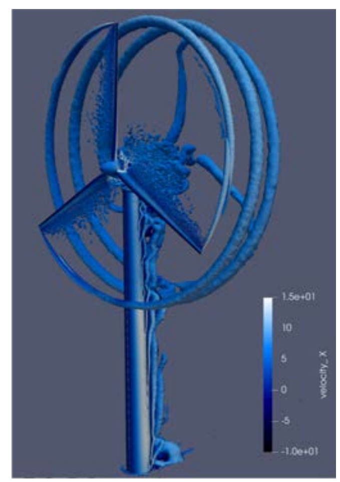
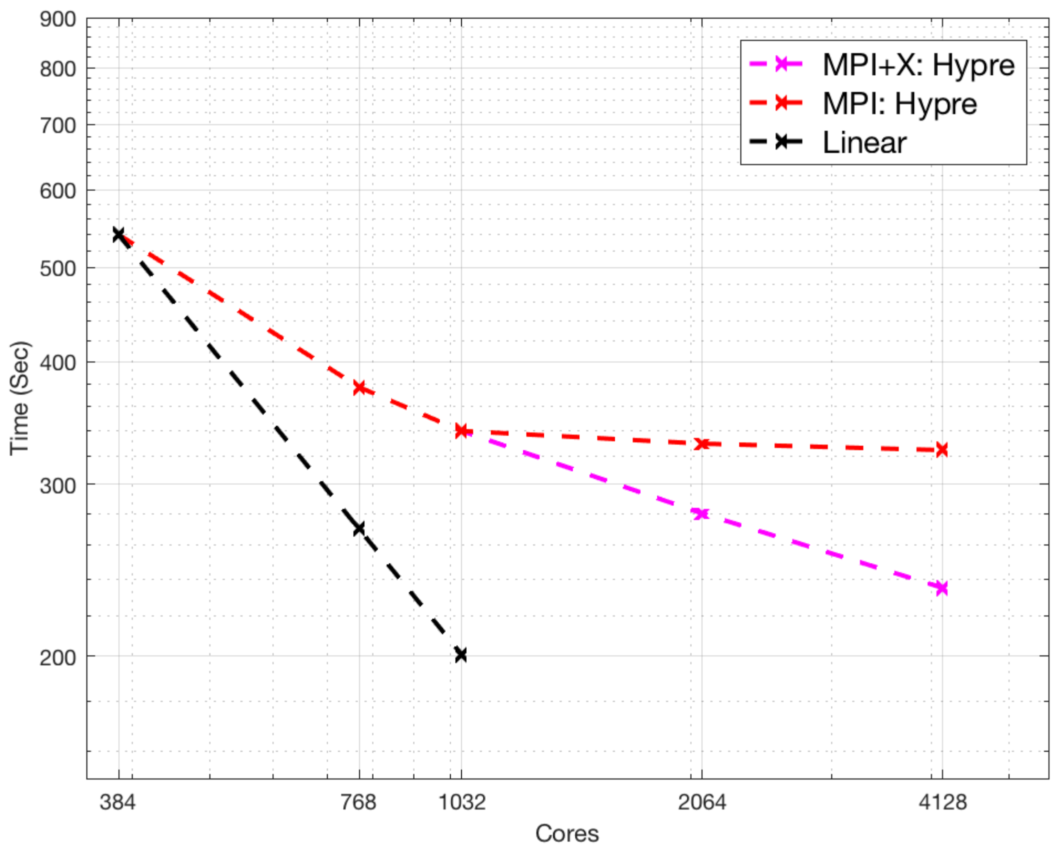


Image from Domino, Barone & Bruner; ExaWind Poster, SAND2018-1050D, Exascale Computing Project 2nd Annual Meeting, Knoxville, TN, February 6, 2018.

Thomas, et al., High-fidelity simulation of wind-turbine incompressible flows with classical and aggregation AMG preconditioners, submitted to *SIAM Journal on Scientific Computing*, Copper Mountain Conference on Iterative Methods 2018 Special Issue, on 15 June 2018.

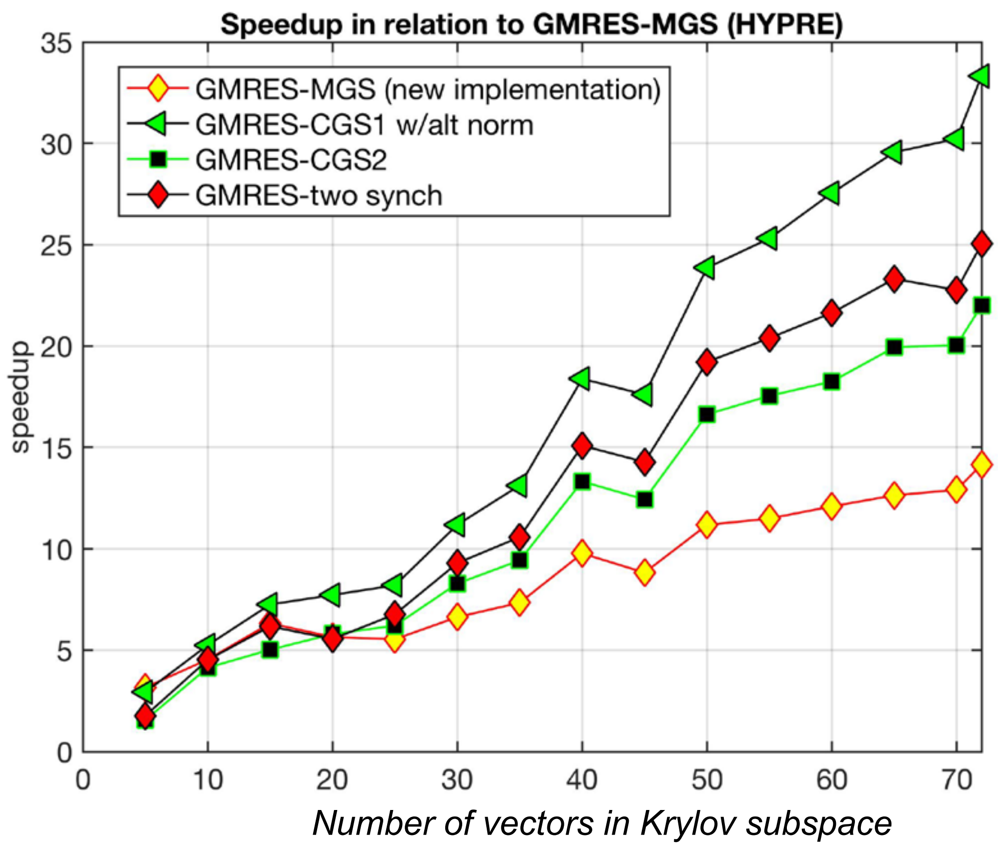
Strong scaling – OpenMP threading



Thomas, Ananthan, Swirydowicz (2018)

- V27 coarse mesh; 46M pressure DOF
- Performed on NREL Peregrine Haswell
- Scaling down to 11k DOFs/proc. unit
- Key observations
  - No benefit of threading until reaching strong MPI scaling limit
  - Can extend scaling past that limit by adding threads
  - Parallel efficiency is low; working to improve it
- Move to larger meshes; different machines (NERSC Cori; ANL Mira)

HYPRE GPU development



Swirydowicz, Thomas (2018)

- Problem: GMRES memory intensive and hard to parallelize
  - Modified Gram-Schmidt (MGS)
- Idea: Replace MGS-GMRES with cheaper yet stable alternatives
- Results on V27 R1 mesh system run on ORNL SummitDev (NVIDIA P100)
- ~35x speedup over baseline HYPRE GMRES on a single GPU

Linear solver scalability – next steps

- Strong scaling studies of NREL 5MW sliding mesh
  - ANL Mira and NERSC Cori
  - 6B nodes (pressure DOFs) and 11B element mesh
- Weak scaling studies – multiple NREL 5MW turbines
- Algorithmic improvements
  - Determine optimal MPI+OpenMP combination for pressure Poisson solves
  - Decrease matrix reinitialization and preconditioner setup costs
  - Communication-optimal GMRES (1 MPI Reduce per cycle)

NREL 5MW mesh statistics

Mesh	# Nodes (M)	# Elements (M)
gCoarse	24.84	56.98
g0	95.35	175.22
g1	761.11	1415.36
g2	608.16	11379.17

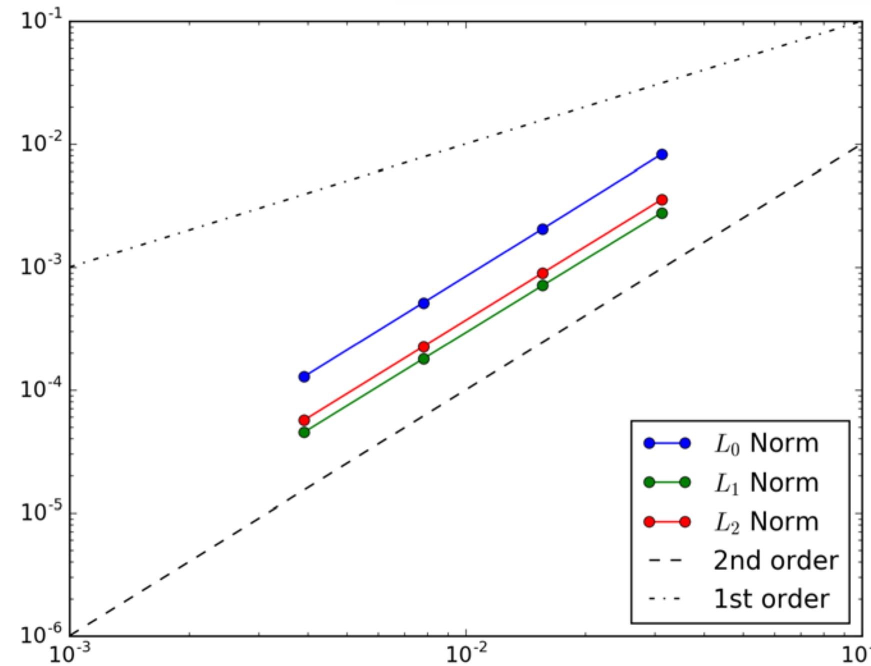
Recent developments – Overset

Overset – Solution verification using MMS

- Method of Manufactured Solutions (MMS)
- 2-D Heat conduction equation

$$T(x,y) = \frac{k}{4}\left(\cos(2a\pi x) + \cos(2a\pi y)\right)$$

- Mesh refinement study on a P=1 mesh
- Overset solution interpolation using element basis functions
- Solution converges as order of numerical scheme
- Must ensure the *same overlap* between fringe points as mesh is refined
  – Consistent with Chesshire and Henshaw (1990)
- Performing moving mesh studies – results to be presented in APS DFD 2018

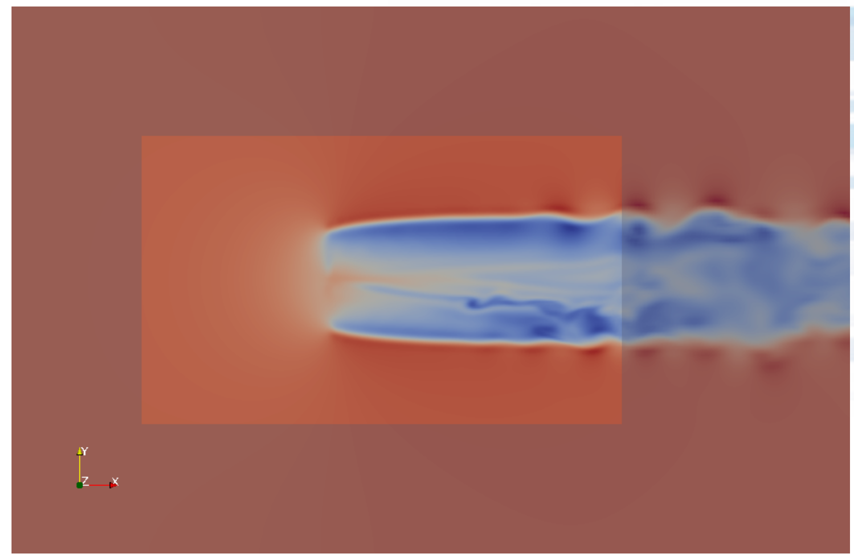


Sharma and Ananthan, 2018

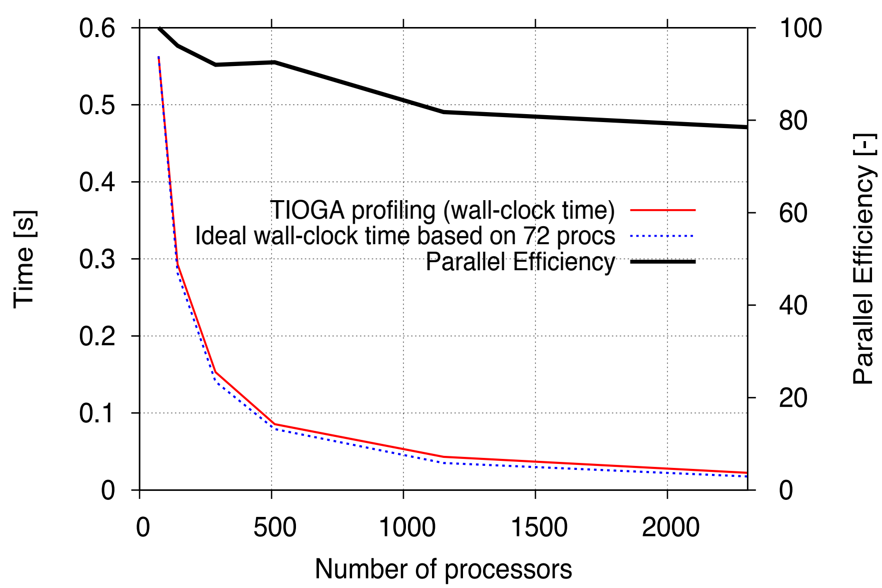
Chesshire, G, and Henshaw, W. D., Composite Overlapping Meshes for the Solution of Partial Differential Equations, Journal of Computational Physics, Vol. 90, 1990.

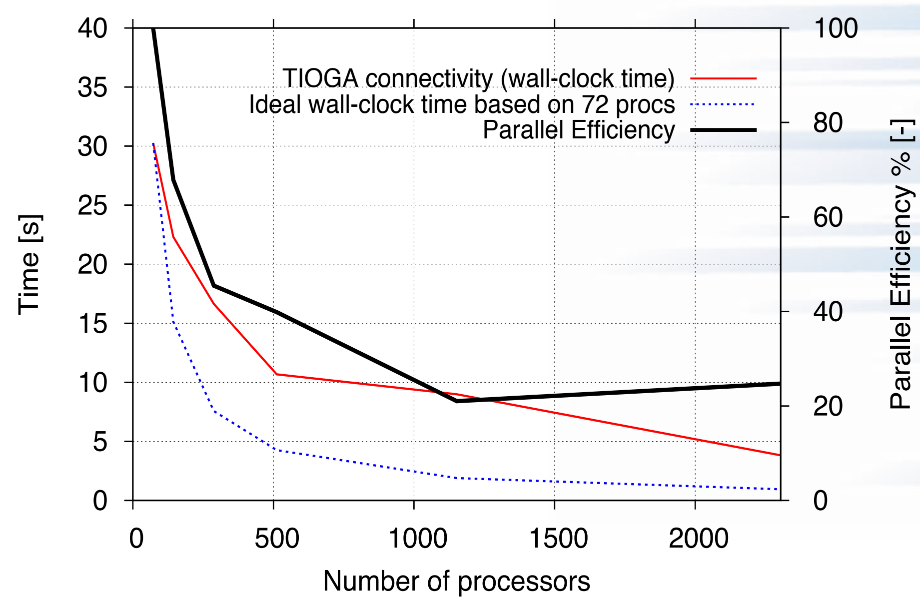
Overset Actuator Line Simulations

- NREL 5MW turbine in uniform inflow (8 m/s)
- Coupled Nalu-OpenFAST simulation
  - OpenFAST provides forces and deflections
  - Nalu provides instantaneous velocity field
- Two nested meshes
  - 2.5m resolution turbine mesh
  - 5m resolution background mesh
  - 10.7 million elements
- Results consistent with non-overset simulations
- Fringe constraint rows negatively impact convergence of pressure Poisson solve
- Implemented and currently testing fringe minimization in TIOGA to improve performance



TIOGA Parallel Performance





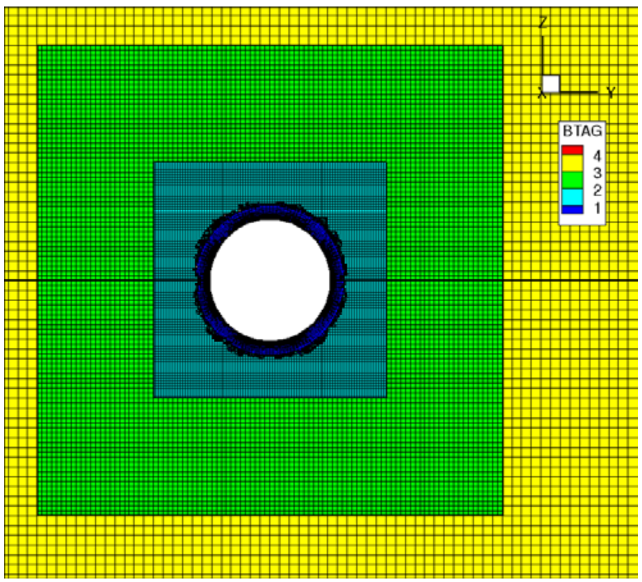
Results from J. Sitaraman (2018)

- *Profiling* – create resolution maps, tag mandatory receptors, and create auxiliary data structures
- *Connectivity* – exchange of search data, containment search and donor/receptor reduction
- Ongoing development to improve TIOGA search algorithms

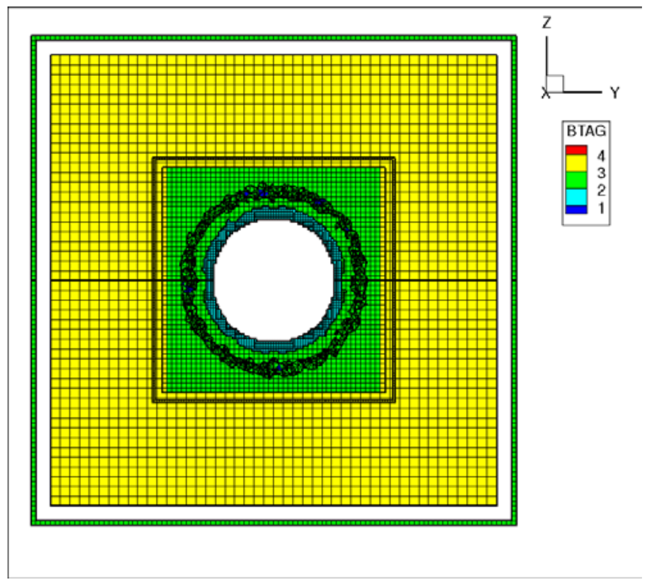
Recent algorithmic improvements to TIOGA

- Added support for multiple unstructured blocks on the same MPI partition
- Improved search to minimize fringe/field mismatches across partition interfaces
- Fringe minimization support in TIOGA

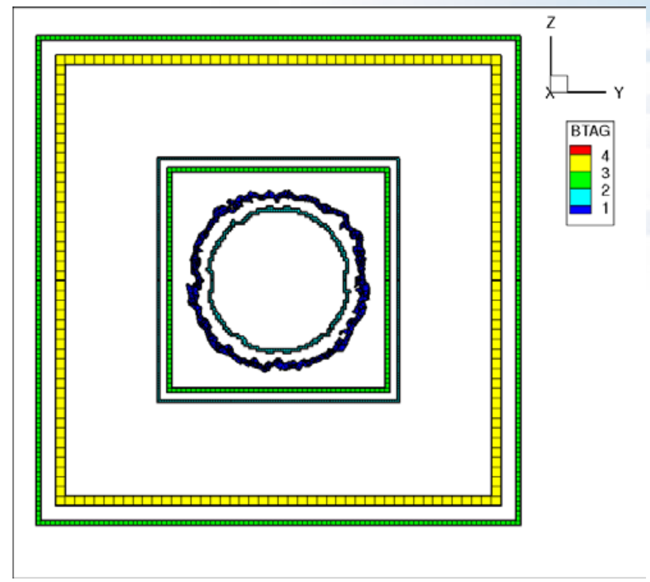
Overset mesh system



Baseline connectivity



After fringe minimization



Results from J. Sitaraman (2018)

Planned improvements to overset capability

- Search algorithm improvements in TIOGA library
  - Faster search for Cartesian type blocks
  - Faster divide and conquer algorithm using an *inverse map*
  - Active load balancing
- Demonstrate overset grid simulations on GPU architectures – native GPU support in TIOGA and interface with Nalu-Wind (via Kokkos and STK-NGP)
- Modularize search and overset connectivity interfaces in TIOGA
  - Use NGP-capable search library for other applications within Nalu-Wind (sliding mesh, post-processing, etc.)
- Demonstrate higher-order overset connectivity support in Nalu-Wind
- Updates will be released publicly at https://github.com/jsitaraman/TIOGA

EXASCALE COMPUTING PROJECT

Acknowledgements

This research was supported by the Exascale Computing Project (17-SC-20-SC), a joint project of the U.S. Department of Energy's Office of Science and National Nuclear Security Administration, responsible for delivering a capable exascale ecosystem, including software, applications, and hardware technology, to support the nation's exascale computing imperative.

This work was authored in part by the National Renewable Energy Laboratory, operated by Alliance for Sustainable Energy, LLC, for the U.S. Department of Energy (DOE) under Contract No. DE-AC36-08GO28308.

This research used resources of the National Energy Research Scientific Computing Center (NERSC), a U.S. Department of Energy Office of Science User Facility operated under Contract No. DE-AC02-05CH11231.

A portion of this research used resources of the Oak Ridge Leadership Computing Facility, which is a DOE Office of Science User Facility supported under Contract DE-AC05-00OR22725.

A portion of this research was performed using computational resources sponsored by the Department of Energy's Office of Energy Efficiency and Renewable Energy and located at the National Renewable Energy Laboratory.