12th Overset Grid Symposium, Atlanta, GA – October 6-9, 2014

Multiple Body Overset Connectivity Method with Application to Wind Farm Simulations

Ezequiel Martin, Pablo Carrica,

IIHR—Hydroscience & Engineering, University of Iowa

Ralph Noack, Celeritas Simulation Technology, LLC

Acknowledgement: Project supported by Iowa Energy Center grant 14-004-OG

THE UNIVERSITY OF IOWA

IIHR Hydroscience & Engineering

Introduction - Motivation

- 10 to 40 % of wind energy production and revenue is lost due to complex wind plant interaction
- Failure rate of mechanical systems is 2-3 times higher than design
- Turbines are controlled individually using local wind conditions

NREL, http://images.nrel.gov/

Introduction - Motivation

- Numerical simulations can yield insight on wake and terrain interaction
  - fully discretized blades are required for accurate load description
  - overset connectivity required even if blade model is used, as wake interactions depends strongly on yaw motions
- Large range of time and length scales of the problem make it demanding to solve
  - $\sim$100M grid points for a small farm, with medium refinement grid
  - time step set by blade rotation, tens to hundreds of thousands time steps required for simulations
- Overset connectivity of complete grid is far from optimal
  - many non-interacting bodies
  - very large background
  - memory limitations in clusters

Objectives

- Develop a new overset connectivity strategy that takes advantage of grid topology
- Simulate small group of turbines as demonstration of capabilities
- Explore extensions of overset strategy to other research areas

Overset Decomposition - Current

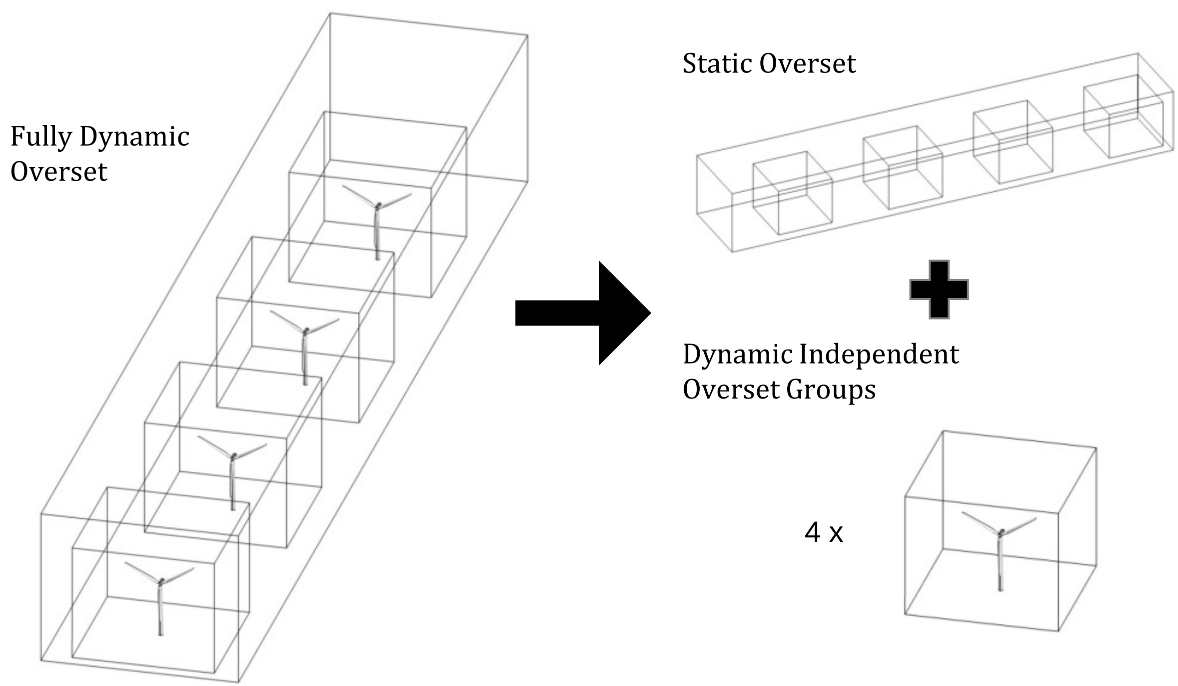
- Suggar/Suggar++ require calculation of overset connectivity over full set of grids
  - Many coefficients do not change (no grid motion)
  - Most grids do not interact with each other
  - Calculation time and memory requirements increase approximately linearly with number of grid points
- Multiple Suggar/Suggar++ instances can be run
  - Optimization of geometry
  - Multiple time lag to accommodate differences in execution time between CFD solver and connectivity solver

Overset Decomposition - Proposed

- Use multiple Suggar groups (instances) comprising subset of grids
  - Use a dci/xintout file for grids that don't move
    - About 40% of donors for considered case
  - Calculate interpolation coefficients dynamically for grids that move as independent problems
    - Some grids will be in static/dynamic groups
    - Grids should not be in more than one dynamic group
    - Rethinking of gridding strategy is required
    - Temporal multi-lagging still possible, but less likely to be necessary

Overset Decomposition – Wind Farm

- Ideal case for multiple dynamic grid groups
  - Turbines are isolated from each other
  - Large static background

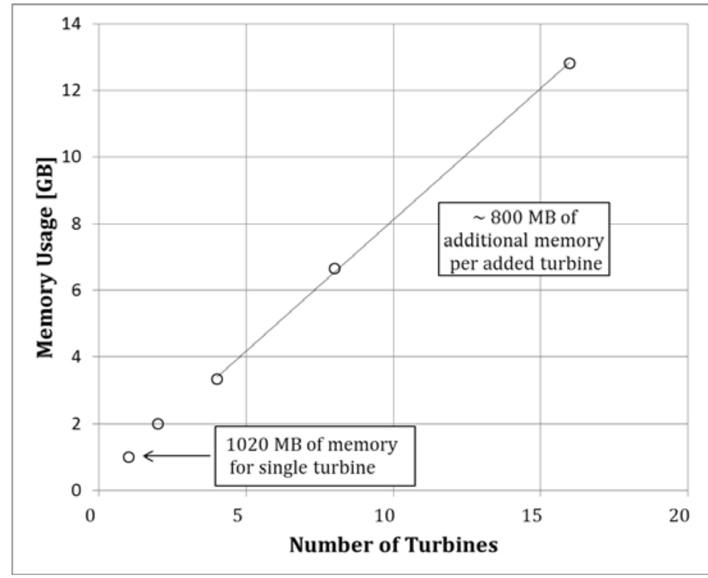


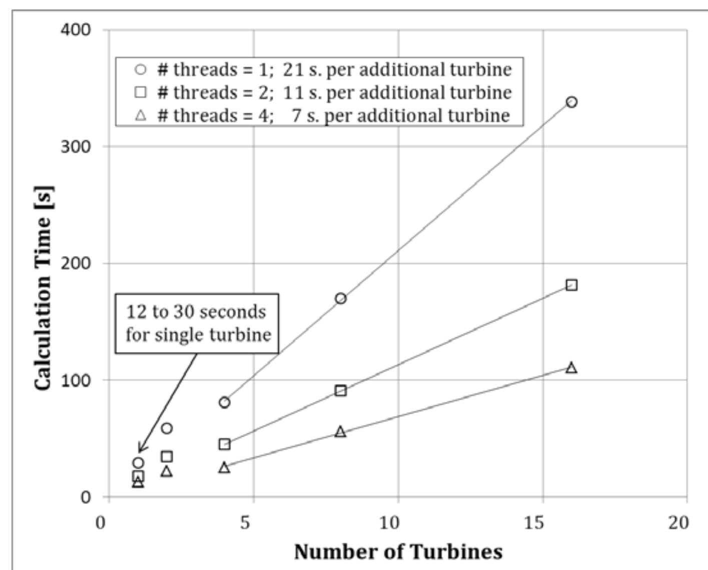
Wind Farm – Stand alone Suggar++ calculations

- 6.6 M grid points per turbine
  - Single turbine
    - 1GB memory usage
    - 30 seconds for single thread
  - per additional turbine
    - 800MB memory usage
    - 7 seconds (with 4 threads)

- For 16 turbines (104 M grid points) and memory limit of 2GB/proc

	Standard method		Dynamic groups
Time lag	4	6	1
Threads	4	2	1
Suggar++ procs	16	12	16
Idle procs (for mem.)	10	27	0
Mem. usage(GB)	52	78	16



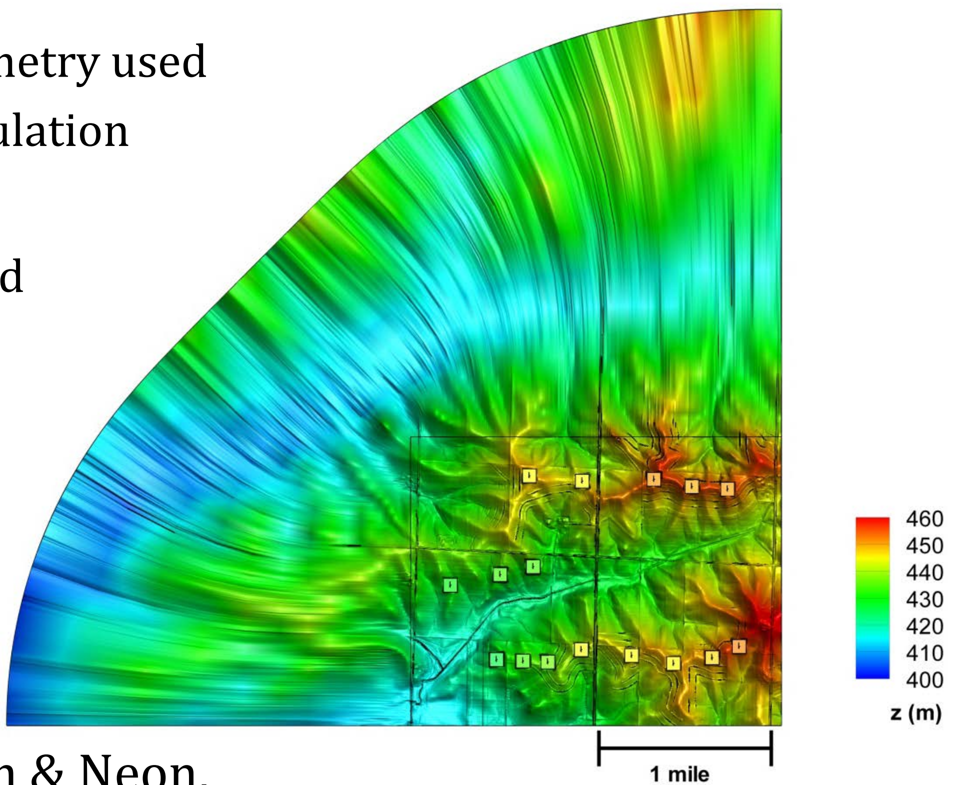


CFD Solver - Magnus

- Multiple dynamic grid groups are managed by CFD solver
- Magnus is new CFD solver from our group
  - primary use is in naval hydrodynamics
  - unsteady Reynolds-averaged Navier-Stokes, detached eddy simulation (URANS/DES)
  - structured grid solver
  - single phase level set solver for free surfaces
  - 6DOF motions solver and controllers for moving surfaces

Test Case - Intrepid Wind Farm (IA)

- 107 1.5 MW GE Turbines
  - scaled NREL 5 MW geometry used
  - only 16 turbines in calculation
- 114 M grid points
  - 40.4 M static background
  - 4.6 M per turbine
  - 16 dynamic grid groups
  - 16 Suggar++ procs.
  - 96/384 Magnus procs. (.3/1.2 M grid points each)

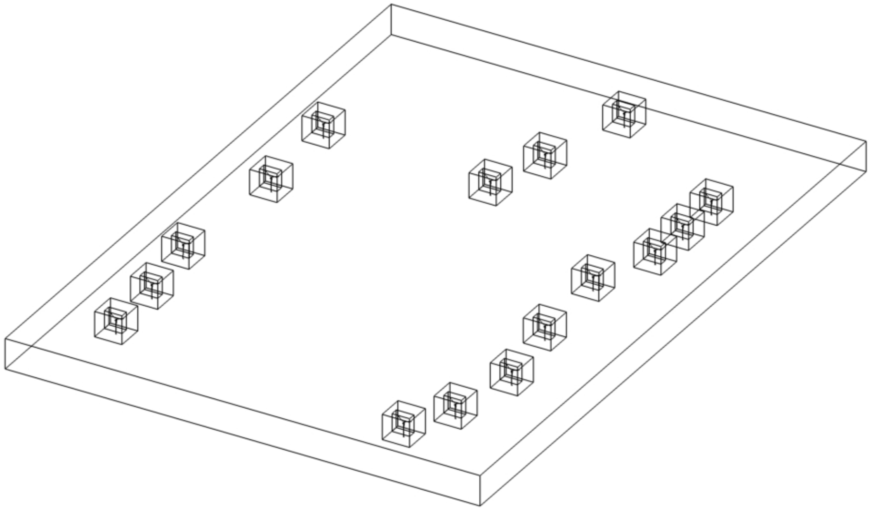


- Simulations run in Helium & Neon, high performance computing resources at University of Iowa

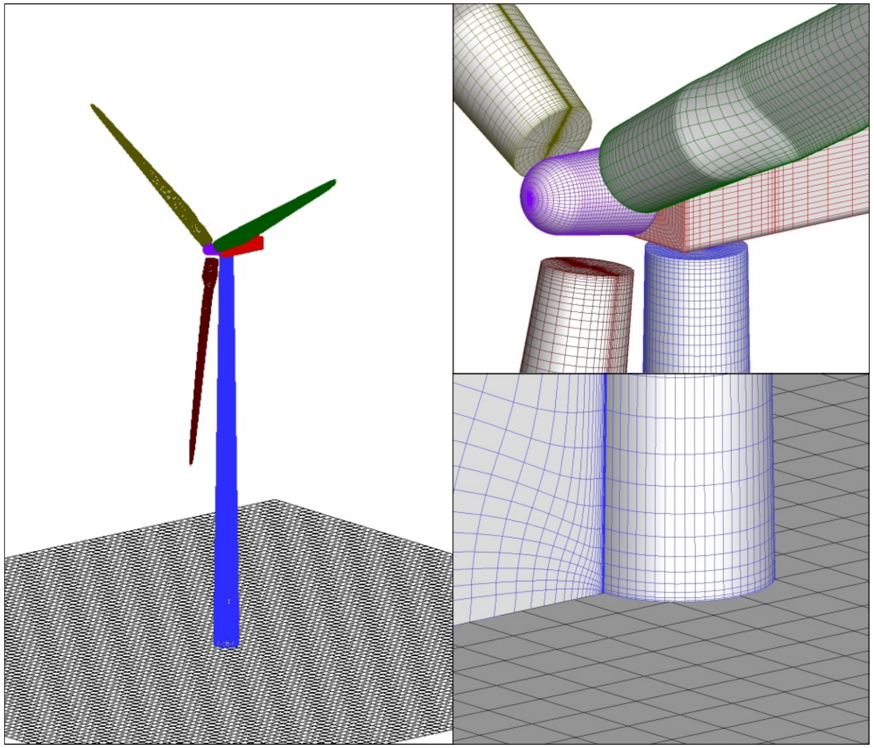
Overset Grids

Blades independently controlled and allowed to pitch

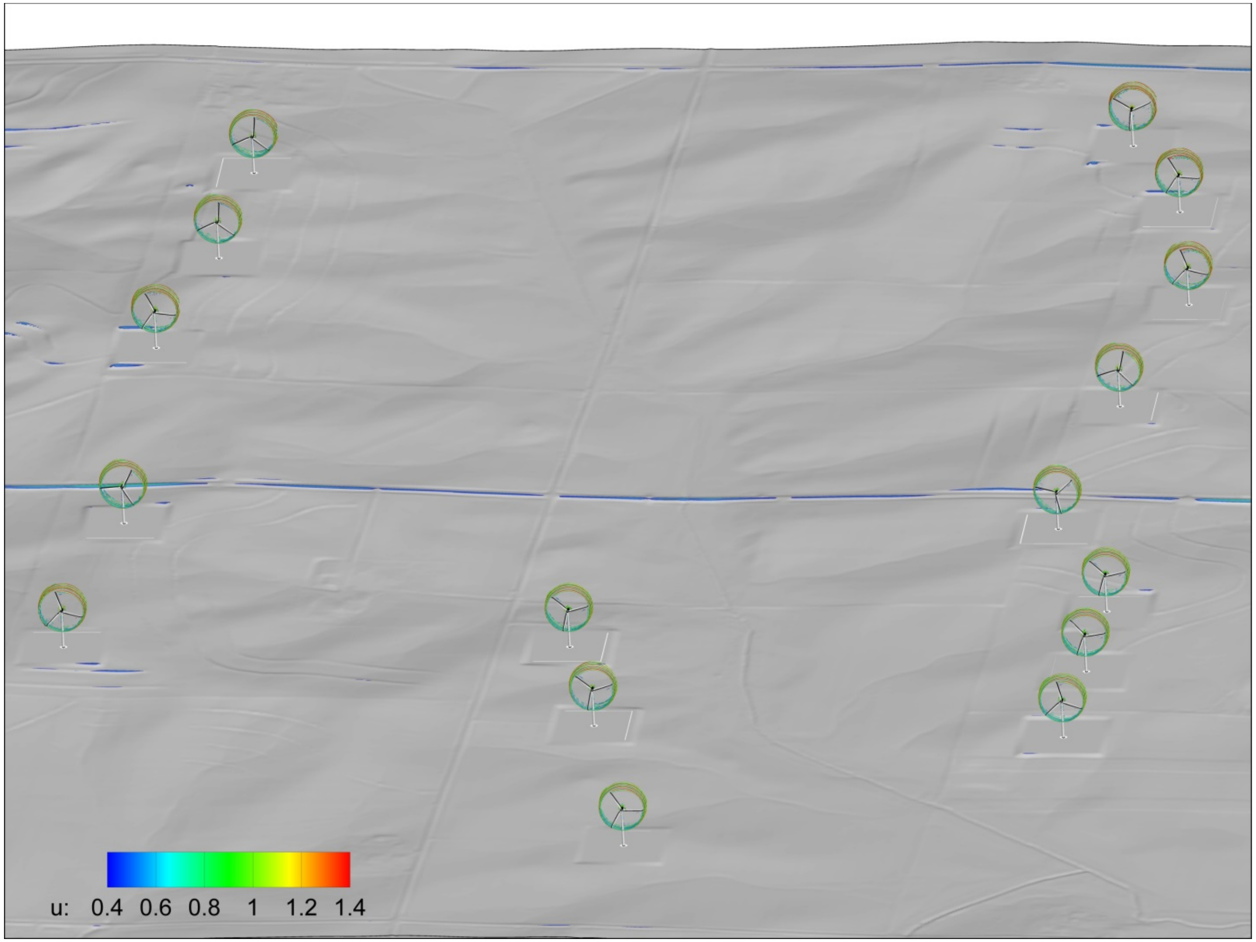
Nacelle, hub, blades and wake refinement yaw as a group



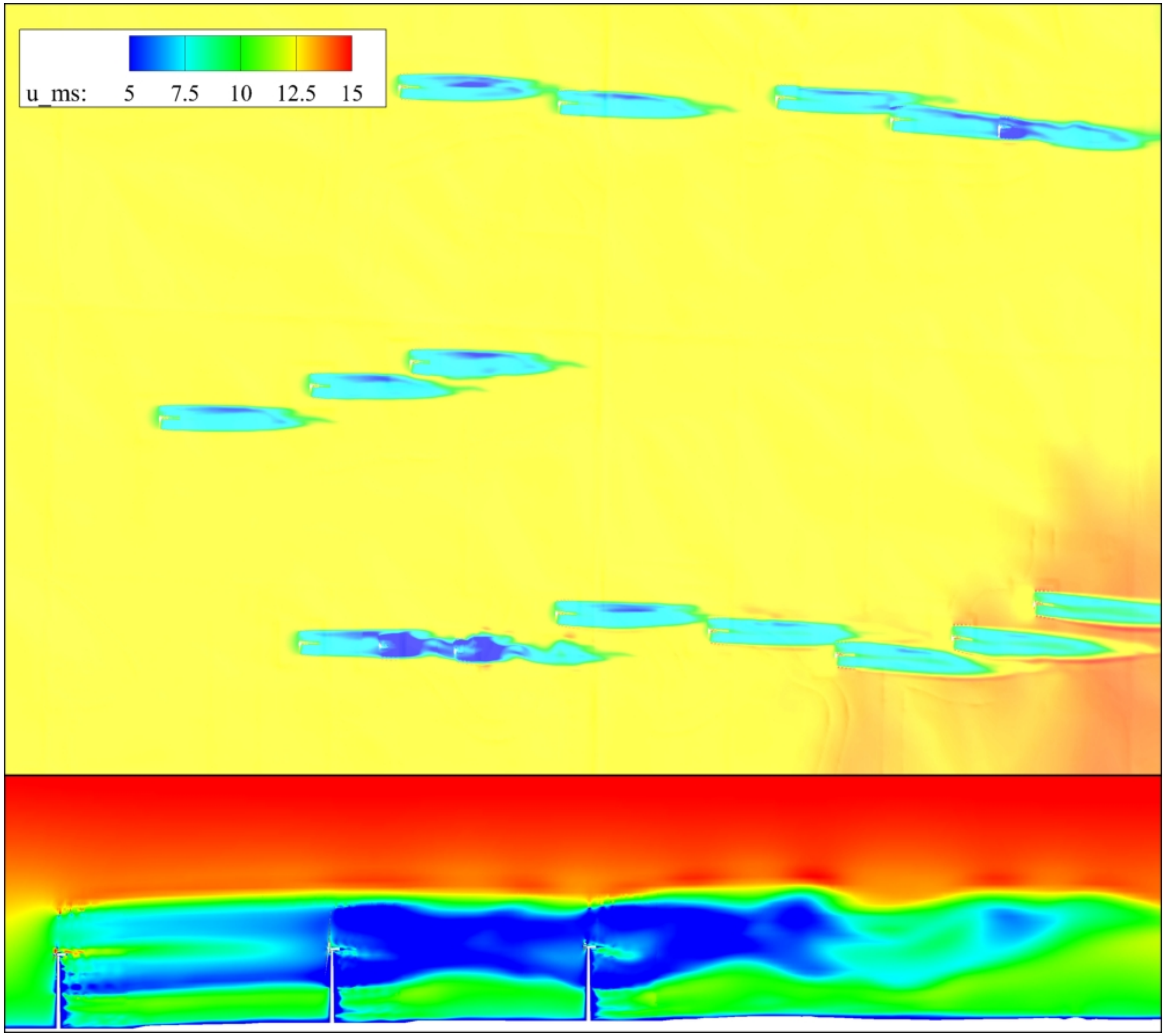
Background	Cartesian	2.5 M + 37.9M
Wrapper	Cartesian	16 x 0.4 M
Wake Ref.	Cartesian	16 x 1.8 M
Tower	O- grid, collar	16 x 0.3 M
Hub, Nacelle	Double O-grid	16 x 0.15 M 16 x 0.15 M
Blade Tip	wrapped collar	48 x 0.15 M
Blade	O-grid	48 x 0.45 M
Total		**114M**



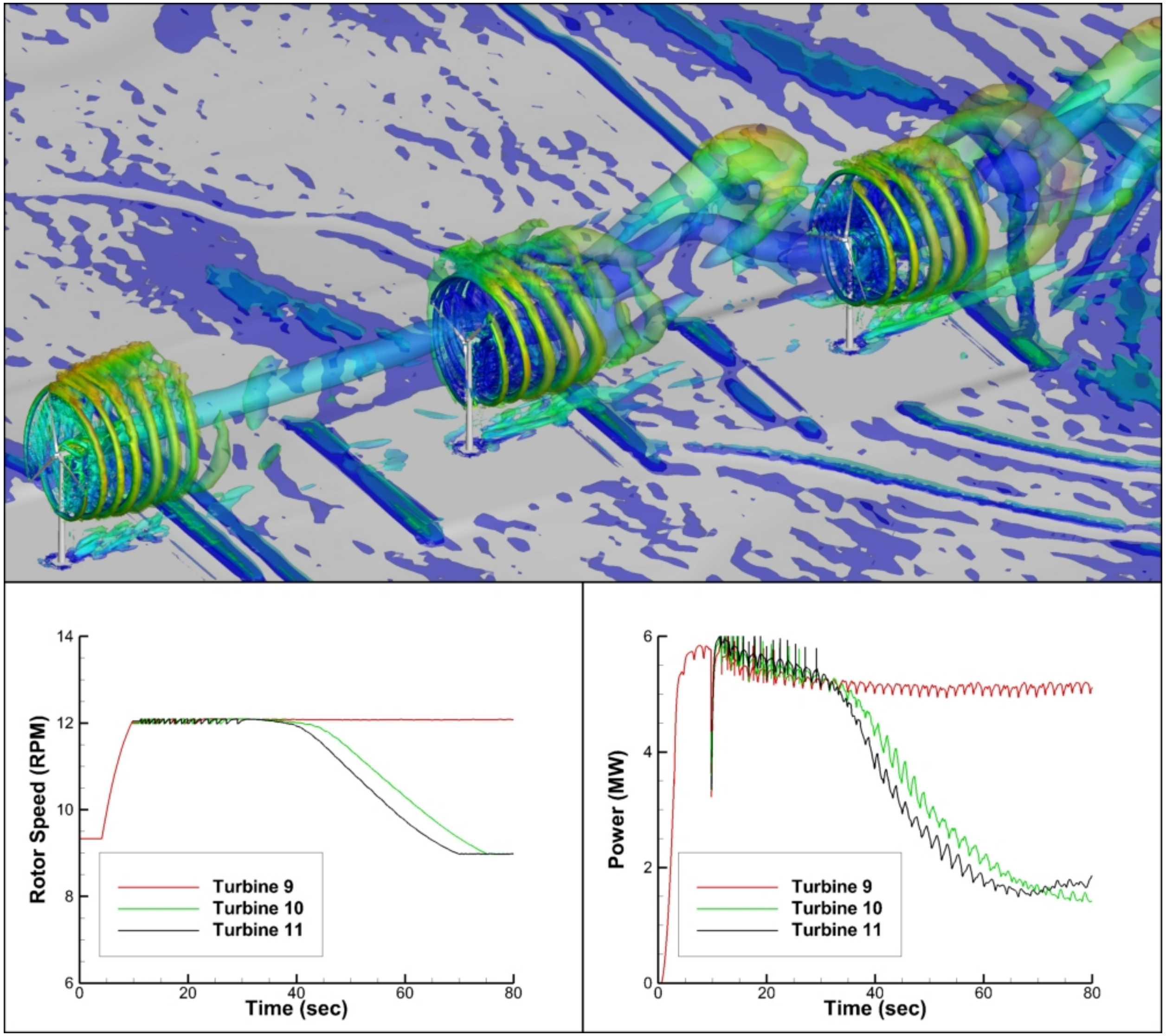
Test Case - Intrepid Wind Farm (IA)



Test Case – Wake interaction

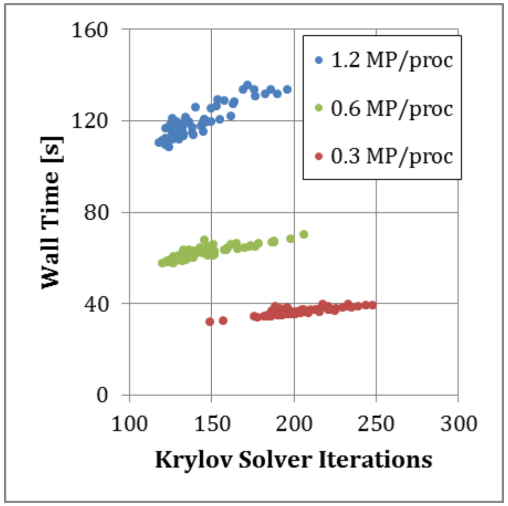


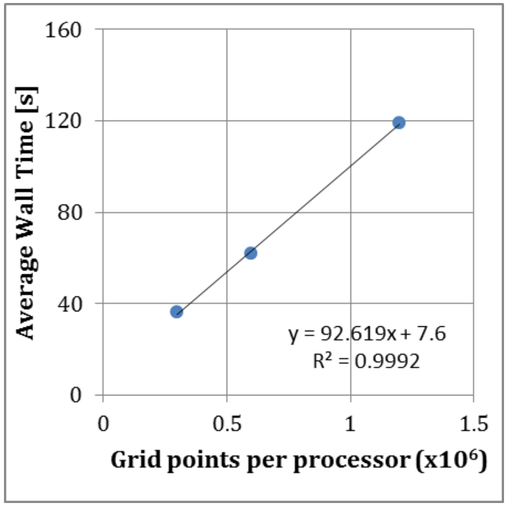
Test Case – RPM Control and Power Output



Code performance

- Excellent scalability
- Pressure solver dominated
  - Currently using PETSc
  - HYPRE to be implemented
- Suggar++ execution time under CFD wall time
  - 0.15M grid points per proc. would require use of parallelized Suggar++ instances, but was precluded by cluster availability





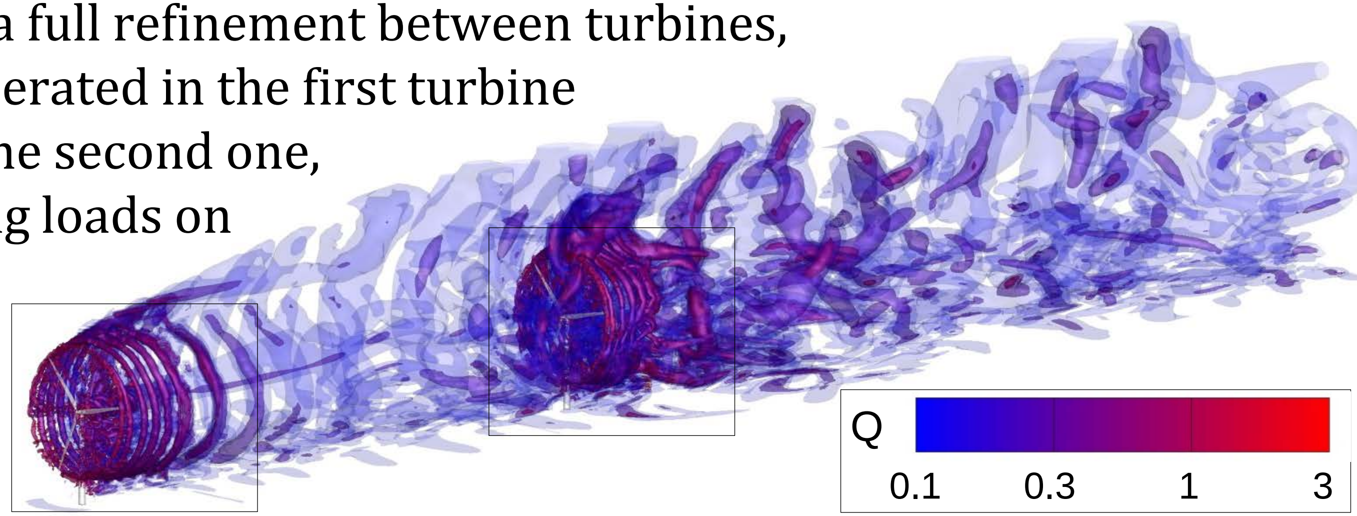
Number of points per processor	Number of processors		Average execution time per time step	Median number of Krylov solver iterations
	Magnus	Suggar		
1.2 M	96	16	119.2	70
0.6 M	192	16	61.8	71
0.3 M	384	16	36.3	101

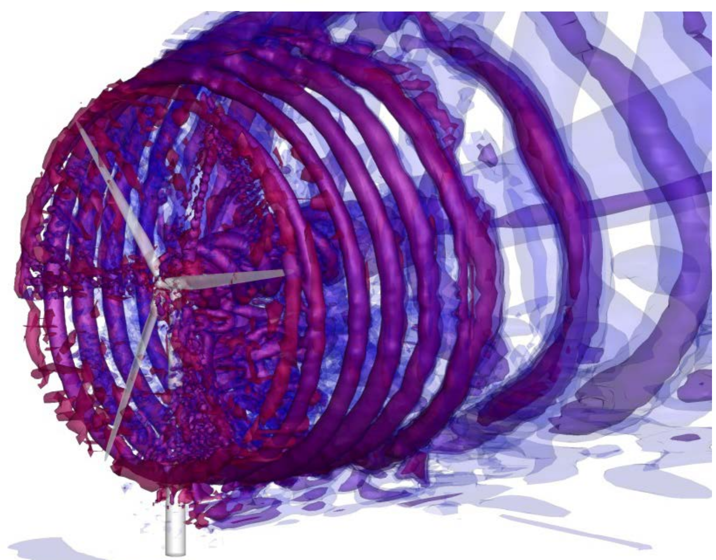
Conclusions and future work

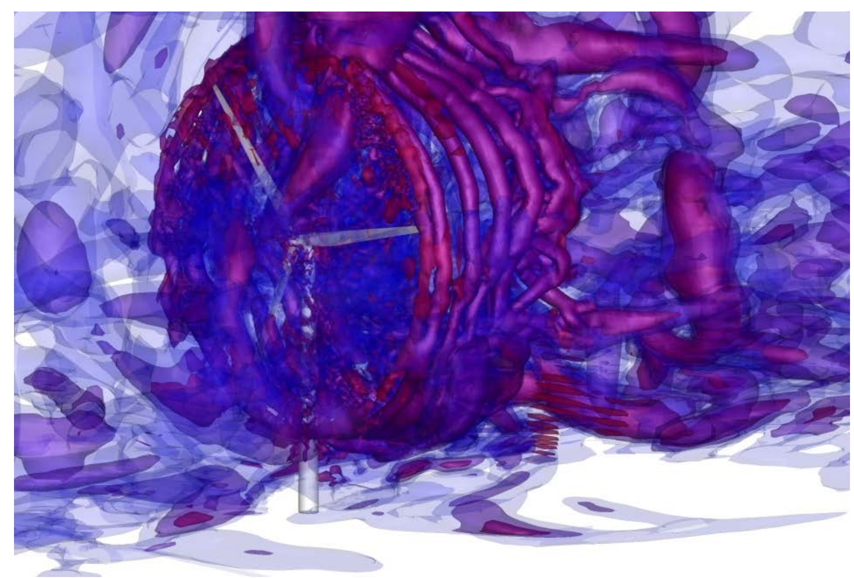
- Overset strategy greatly reduces memory usage and execution time
  - Execution time determined by CFD solver, not overset calculation
- Time step is too small for analysis of plant-wide operation of farm
  - Modeling of blades would increase time step 10-20 times
  - Fully discretized calculations are still needed to accurately described interactions between machines
- Wake interaction requires better discretization between turbines
  - Dynamic overset still required to 'channel' wakes between turbines with reasonable amount of grid points

Fully developed wake (2 Turbine case)

Even without a full refinement between turbines, structures generated in the first turbine are reaching the second one, locally affecting loads on blades



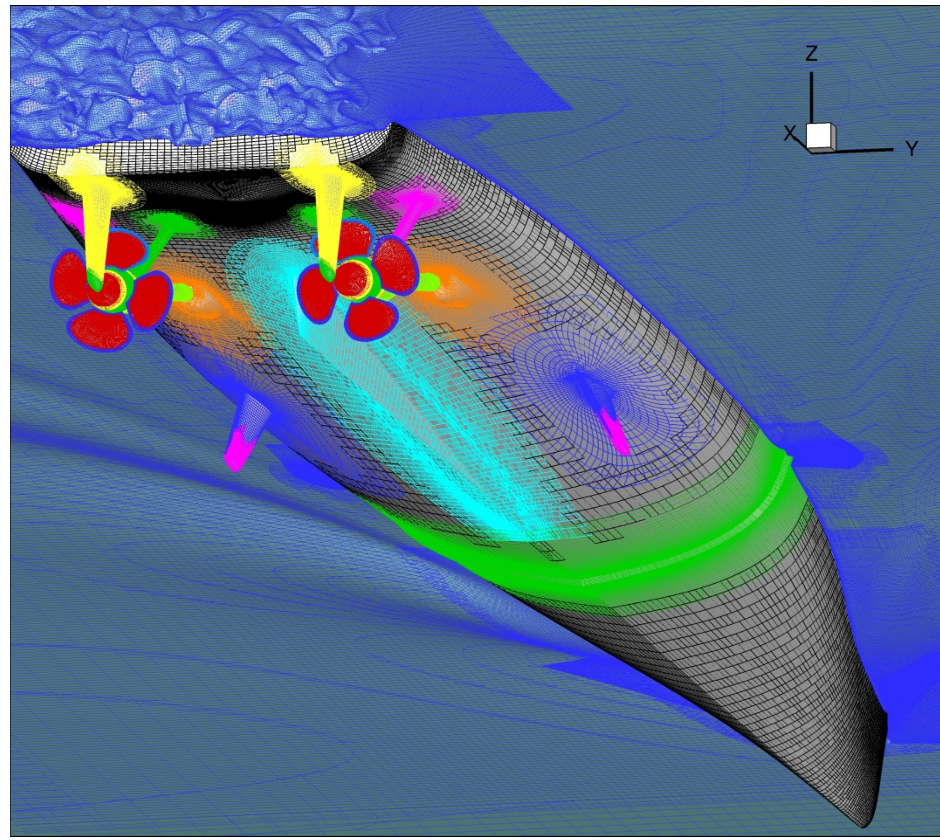




Applications to Naval Hydrodynamics

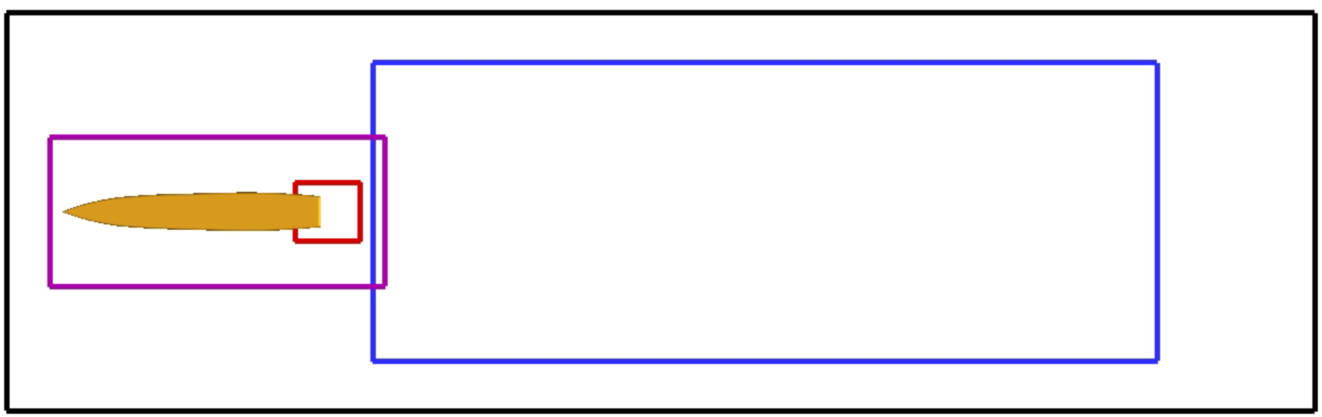
Example: Athena R/V

- 69 overset grids
- 29 M grid points
  - 82 M for fine grid
- Far wake refinement can add anywhere from 10s to 100s millions grid points
- Only truly dynamic grids
  - Rudders & blades
  - Hull (and appendages) with respect to background and refinements
  - 5-10 M grid points per dynamic group **if** grids can be separated efficiently

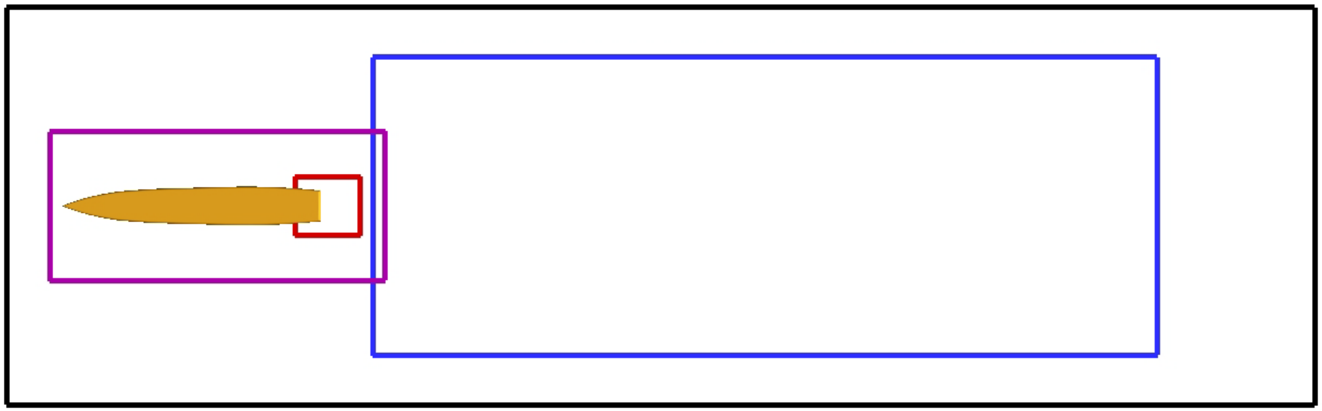


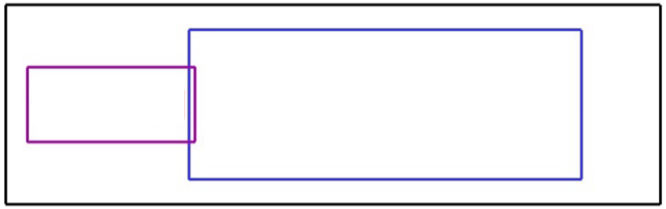
Applications to Naval Hydrodynamics

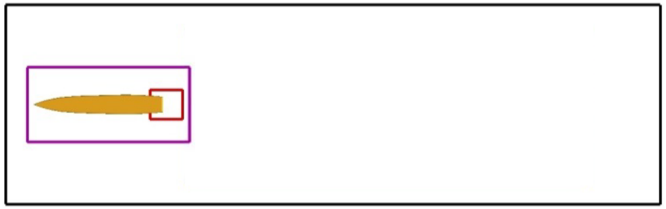
- Far wakes
  - Detection of far wakes (up to 5km from vessel) requires use of very large fine grids that are static with respect to background
  - If the effect of propellers is investigated, dynamic overset is necessary
  - Partition of the overset connectivity problem in dynamic and static parts great reduces computational cost



Applications to Naval Hydrodynamics -Wakes





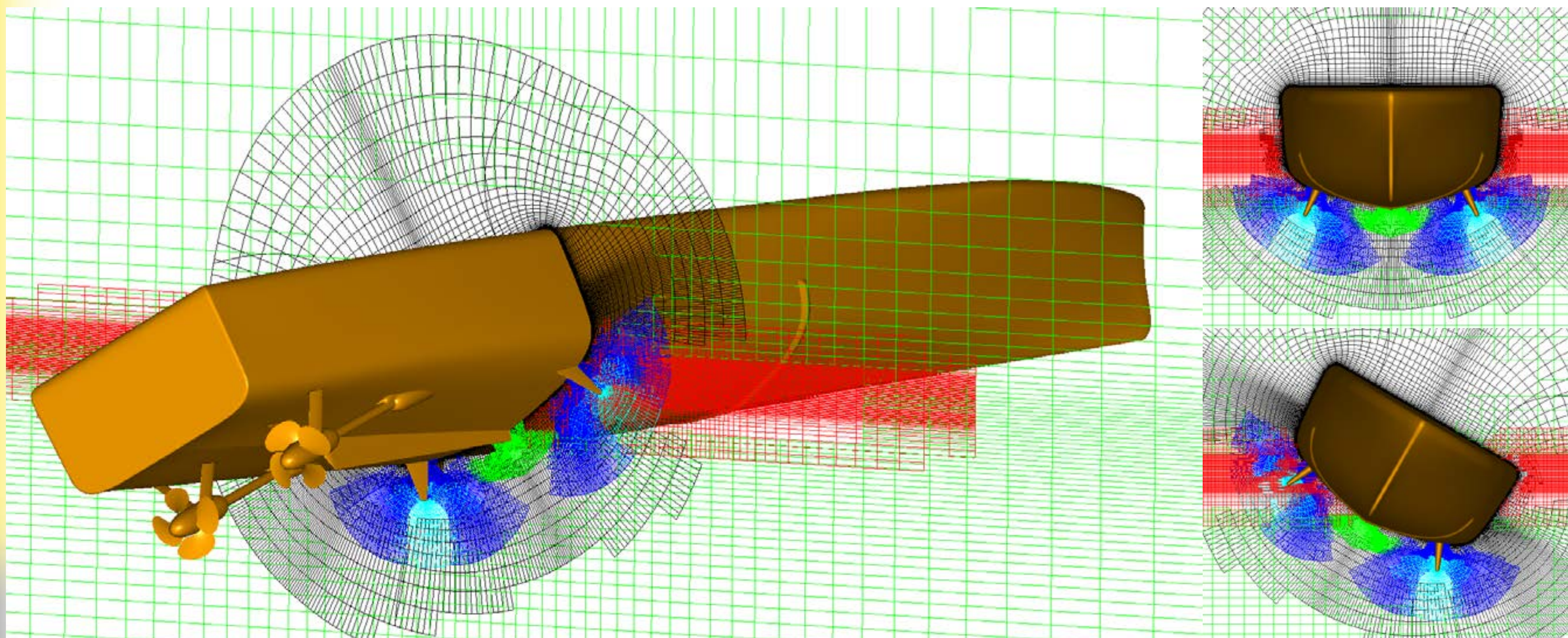


Static grids – Use dci file

Dynamic grids – Any appendage embedded in hull grids is not included in overset computation

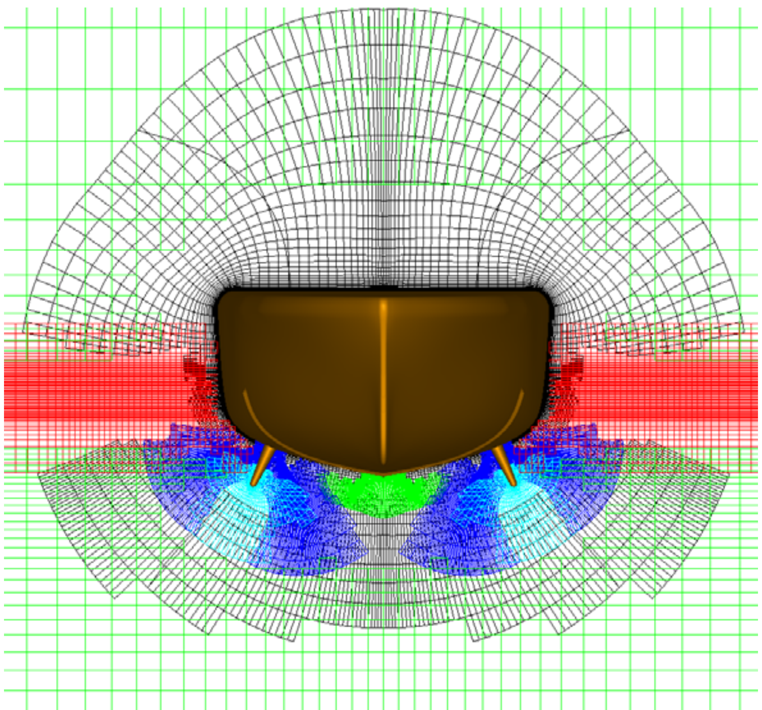
Applications to Naval Hydrodynamics

- Fixed Appendages
  - For structured grids, appendages are usually gridded separately using collar grids
  - Changes in connectivity interpolation only due to movement of parent grid

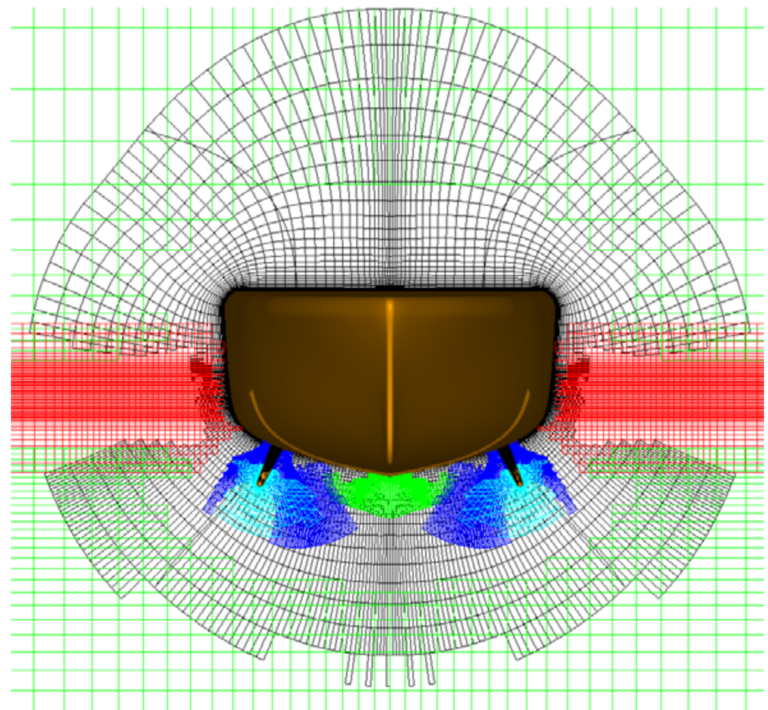


Applications to Naval Hydrodynamics-Fixed Appendages

- blanking is required to fully embed collar grids in parent grid



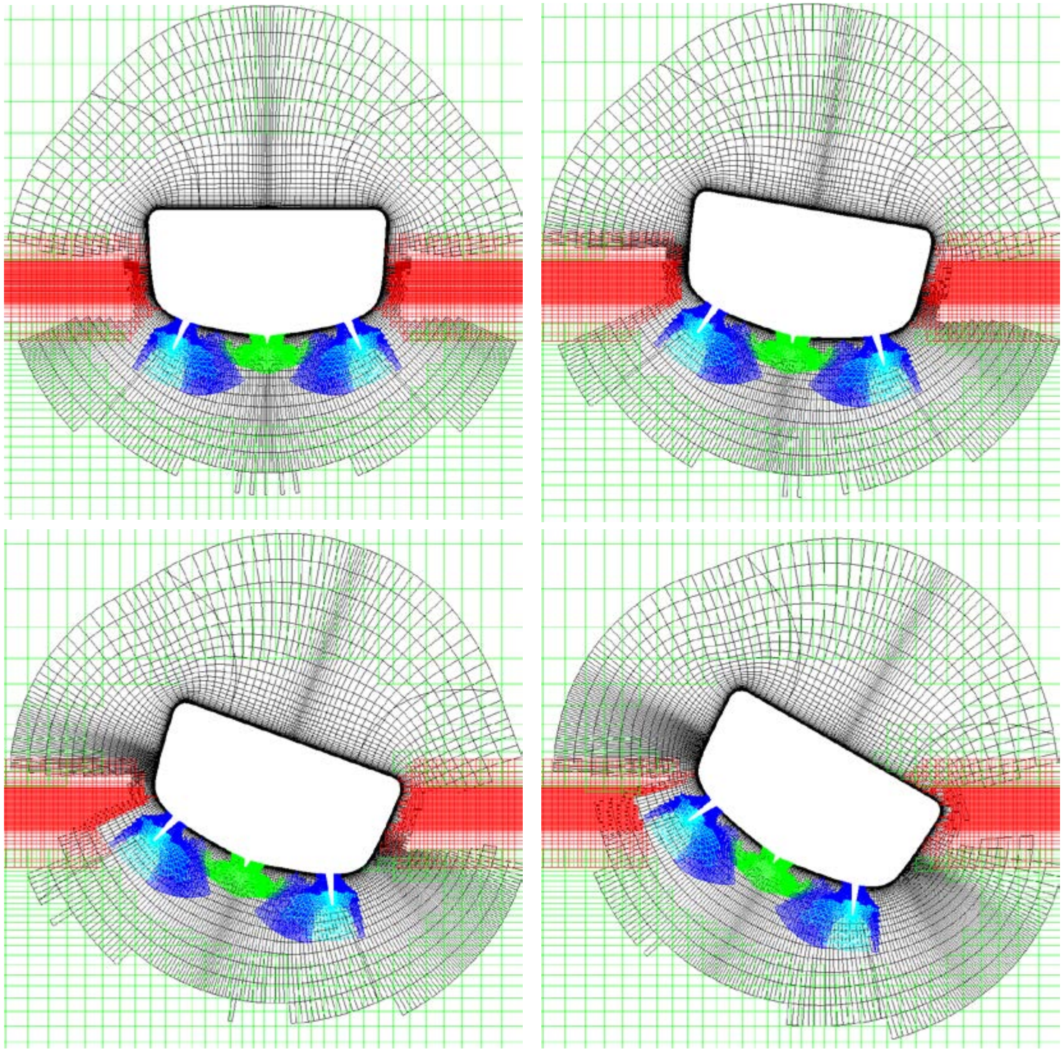
original



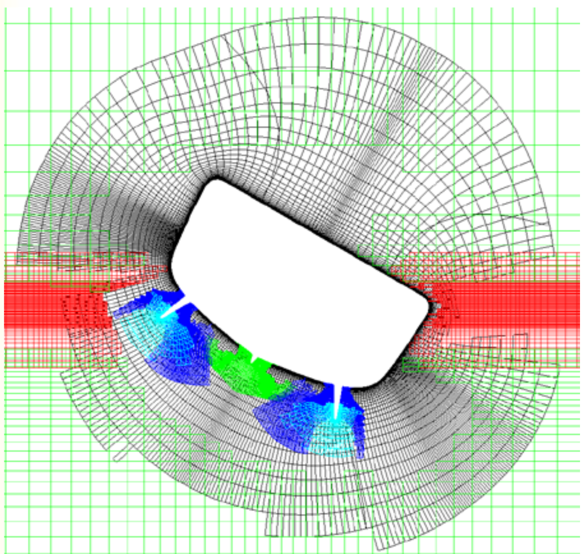
with blanking

Applications to Naval Hydrodynamics-Fixed Appendages

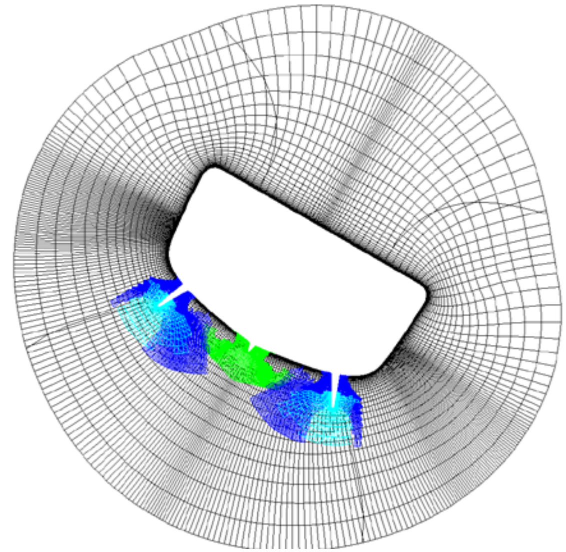
- Blanking must work for different dynamic conditions
  - Avoid overlap of appendage grid and Cartesian refinement
- Grid quality should be considered
- Re-gridding might be necessary



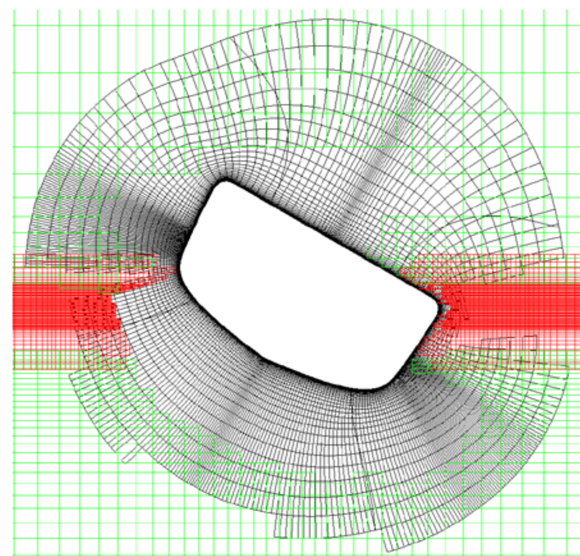
Applications to Naval Hydrodynamics-Fixed Appendages



Original, with blanking



Static group



Dynamic group

Applications to Naval Hydrodynamics – Moving Appendages

- Hardest case to implement
- Requires that grids are fully separated from hull using a static wrapper grid, or hull grid needs to be allowed in multiple dynamic groups
  - Increase in gridding complexity and cost
  - Might compromise grid quality
  - Ultimately integration in overset connectivity solver should be more efficient than through CFD solver