

# Exawind

## Overview and Workshop

November 2, 2023

Michael Kuhn, Lawrence Cheung, Phil Sakievich



# Outline

---

- Overview of software suite
  - Individual codes and framework
  - Applications and capabilities
  
- AMR-Wind
  - Anatomy of input file
  - Tutorial
    - Installation
    - ABL
    - Actuator Disk

# Exawind software suite

- Primary target of development: geometry-resolved floating offshore wind simulations
- In the process, provide a versatile, open-source modeling tool for wind energy researchers with high-fidelity fluid mechanics and multi-fidelity turbine modeling

## AMR-Wind:

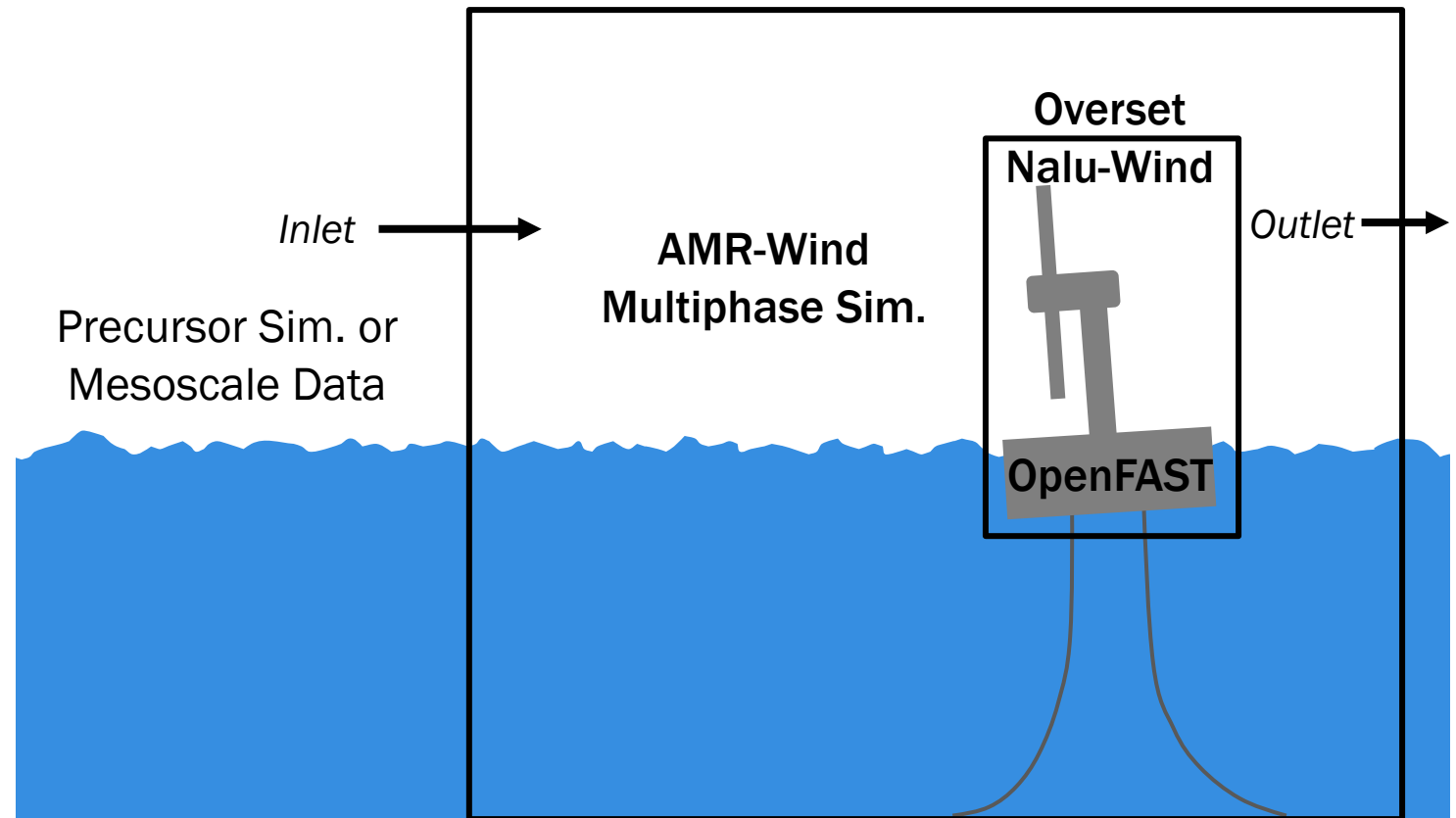
structured mesh, bulk of computational work

## Nalu-Wind:

unstructured mesh, near-body flow dynamics

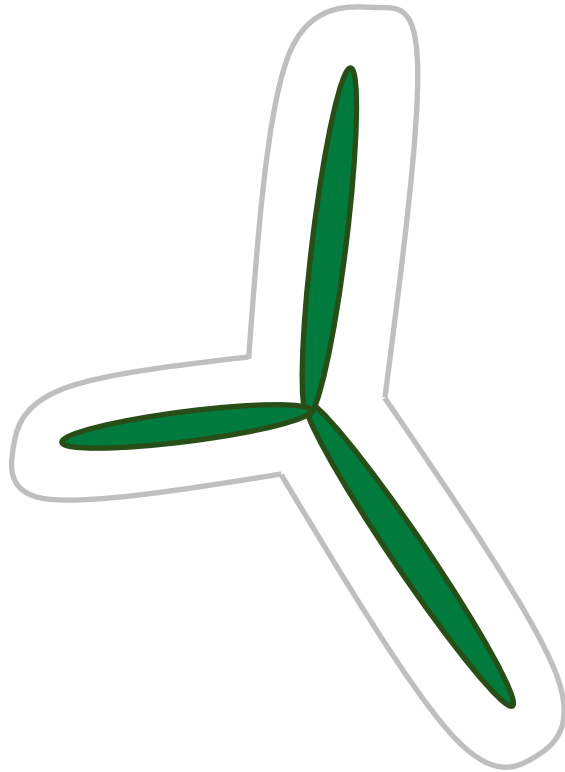
## OpenFAST:

structural modeling

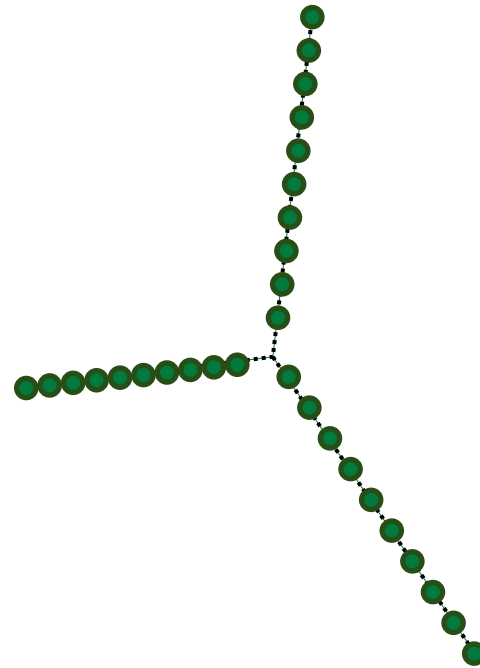


# Exawind software suite

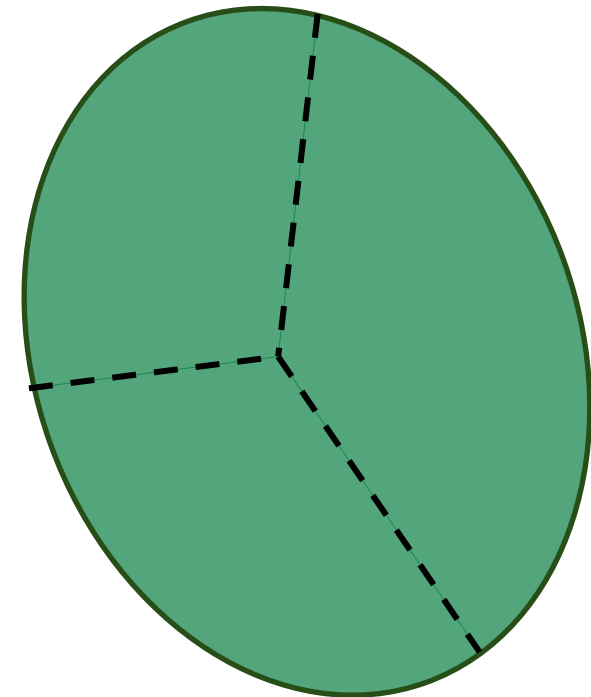
- Versatile, open-source, high-fidelity fluid mechanics and multi-fidelity turbine modeling



**Geometry-resolved  
(Overset)**



**Actuator Line  
(ALM)**



**Actuator Disk  
(ADM)**

# Exawind software suite

---

- Applications:
  - High-res geometry-resolved turbine simulations
  - Overset wave simulations for offshore wind
  - Huge farm-scale simulations with actuators

# AMR-Wind

- After showing the bigger picture, we're going to focus on AMR-Wind. Fast, easy-to-use, actuator line and disk methods coupled to OpenFAST make it simple to apply in a variety of wind energy studies.
- Repository
  - <https://github.com/Exawind/amr-wind>
- Documentation
  - Readthedocs
    - Input file reference: <https://exawind.github.io/amr-wind/user/inputs.html>
  - Tutorial: <https://github.com/mbkuhn/amr-wind-tutorial/>
  - test/test\_files

# Anatomy of AMR-Wind input file

- File: “case\_name.inp” or “case\_name.i”
- Syntax (in general):

```
Category           =  Entry1 Entry2 Entry3  
Entry1.option1     =  Value1  
Entry2.option2     =  Value2
```

- Spacing and indentation do not matter
- Comment with ‘#’

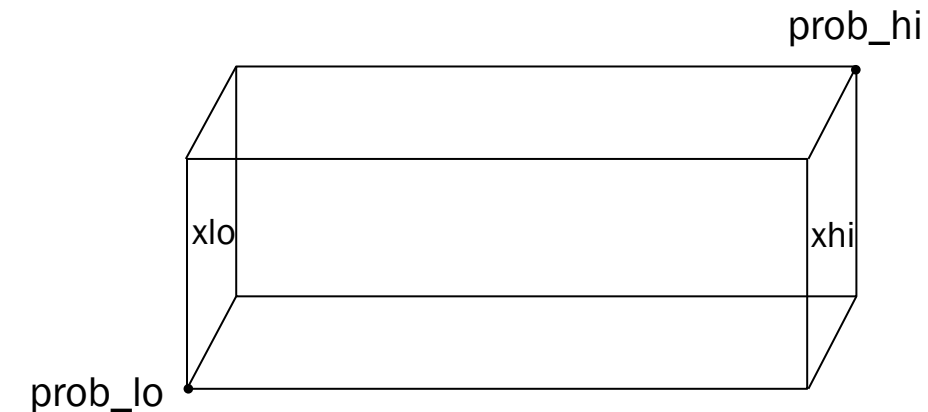
# Anatomy of AMR-Wind input file

- Physics
  - Flow initialization and changing environment during simulation
  - Can connect distinct parts of the code, like forcing and boundary conditions
  - This is the “case” that you are running; but also, can specify more than one because they can address different aspects of the simulation
  - E.g., FreeStream, ABL, Actuator, TaylorGreenVortex
- ICNS.source\_terms
  - ICNS = InCompressible Navier-Stokes, i.e., the momentum equation
  - E.g., BoussinesqBuoyancy, CoriolisForcing, ABLForcing



# Anatomy of AMR-Wind input file

- Domain
  - Due to block-structured nature, all domains are rectangular
  - Define bottom corner position, top corner position, number of cells in each direction for base resolution
  - `geometry.prob_lo`, `geometry.prob_hi`, `amr.n_cell`
- Boundary conditions
  - Specify type for each: `xlo`, `xhi`, `ylo`, etc.
  - Additional values depending on type
  - Should not be specified in periodic direction: `geometry.is_periodic`
- Time stepping
  - `time.fixed_dt`, `time.initial_dt`, `time.cfl`
  - Set negative value to make inactive



# Anatomy of AMR-Wind input file

- Checkpoints: output with `time.checkpoint_interval`, use with `io.restart_file`
- Plotfiles: output with `time.plot_interval`, can specify non-default and derived variables
- Postprocessing
  - Targeted tools to extract specific quantities and output to data files
  - E.g., ABLStats, Sampling, Averaging
- Mesh refinement
  - Add levels by specifying `amr.max_level > 0`
  - 2 primary methods for static refinement:
    - CartBoxRefinement: define inset rectangular domains
    - GeometryRefinement: define shapes (cylinders, boxes) with parameters

# AMR-Wind: installing

---

- Essential details of spack-manager: see 01\_compiling.md in amr-wind-tutorial
- <https://github.com/mbkuhn/amr-wind-tutorial/>

# AMR-Wind: ABL example

---

- see 02\_atmosphere.md in amr-wind-tutorial
- <https://github.com/mbkuhn/amr-wind-tutorial/>

# AMR-Wind: Actuator Disk example

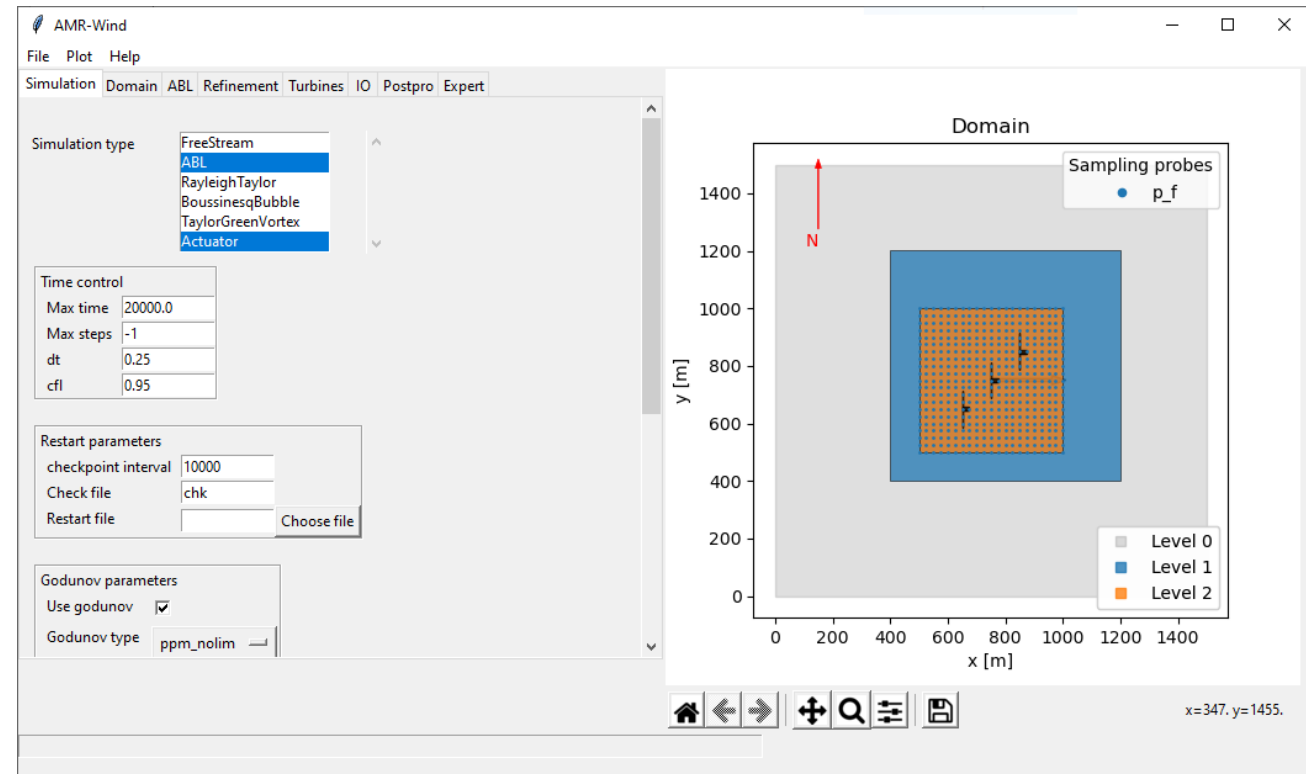
---

- see 03\_turbines.md in amr-wind-tutorial
- <https://github.com/mbkuhn/amr-wind-tutorial/>

# AMR-Wind-frontend overview

## Handy GUI & python interface to help setup complex cases

- Load an AMR-Wind input file and change parameters interactively
- Plot the simulation domain, including refinement zones and sampling probes/planes
- Set up complex wind farm configurations
- Validate AMR-Wind inputs before job submissions
- Submit jobs to a cluster
- Visualize the sampling outputs (probes, lines, and planes)
- Postprocess ABL statistics files
- Use it in Jupyter notebooks or python scripts to automate processing



# AMR-Wind frontend: Installation & Documentation

AMR-Wind frontend documentation available at

<https://github.com/Exawind/amr-wind-frontend/tree/main/docs>

Contains installation, usage, and customization instructions

- Three tutorials available:
  1. An actuator disk model in uniform flow
  2. Running an unstable ABL LES case
  3. Setting up a wind farm configuration
- Two case studies
  1. SWIFT ABL test case
  2. ADM turbine model run

# Example of using GUI to set up wind farm

AMR-Wind

File Plot Run Help

Simulation Domain ABL Refinement Turbines IO Postpro Expert Farm

Wind farm setup

Farm setup file

Actions

Wind farm turbine layout

CSV file

CSV Contents

```
# CSV file should have columns with
# name, x, y, type, yaw, hubheight, options
4713-OE, 36.378235, -97.387932, UnifCtTest, , ,
4655-OE, 36.410927, -97.415176, UnifCtTest, , ,
4651-OE, 36.410717, -97.432724, UnifCtTest, , ,
4726-OE, 36.370808, -97.360832, UnifCtTest, , ,
4671-OE, 36.403244, -97.397263, UnifCtTest, , ,
4715-OE, 36.364983, -97.423828, UnifCtTest, , ,
4691-OE, 36.385166, -97.433220, UnifCtTest, , ,
4657-OE, 36.411022, -97.405914, UnifCtTest, , ,
```

Layout options

Delete existing turbines

Turbine coordinate sys

Auto calculate farm center

Farm center coords (X,Y)

Farm domain size (X,Y,Z)

Background mesh size [m]

Plot turbine names

Actions

Domain

1e6

4.0375

4.0350

4.0325

4.0300

4.0275

4.0250

4.0225

4.0200

4.0175

y [m]

632500 635000 637500 640000 642500 645000 647500 650000 652500

x [m]

Level 0

Level 1

Level 2

pan/zoom

Plot domain

Plot settings

Choose plot view

Plot wind & N arrows

Select sample probes

Select refinement zones

Farm\_level\_0\_zone

4713-OE\_level\_1\_zone

4655-OE\_level\_1\_zone

4651-OE\_level\_1\_zone

4726-OE\_level\_1\_zone

Select turbines

4713-OE

4655-OE

4651-OE

4726-OE

4671-OE



# Python/Jupyter notebook interface

## Example of using the frontend via python interface

```
# Load the module
import amrwind_frontend as amrwind

# Start the amrwind_frontend app
tutorial1 = amrwind.MyApp.init_nogui()

# Set some parameters
tutorial1.setAMRWindInput('time_control',['const dt'])
tutorial1.setAMRWindInput('time.stop_time',100)
tutorial1.setAMRWindInput('time.fixed_dt', 0.1)
tutorial1.setAMRWindInput('incflo.physics', ['FreeStream', 'Actuator'])

# Do some other stuff here
...

# Plot the figure
tutorial1.plotDomain(ax=ax)
```

