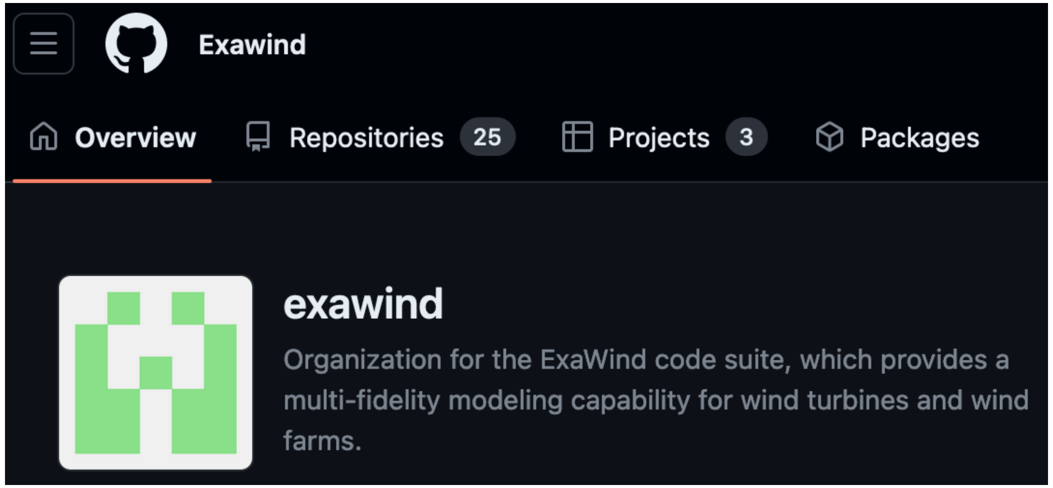
Exawind

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

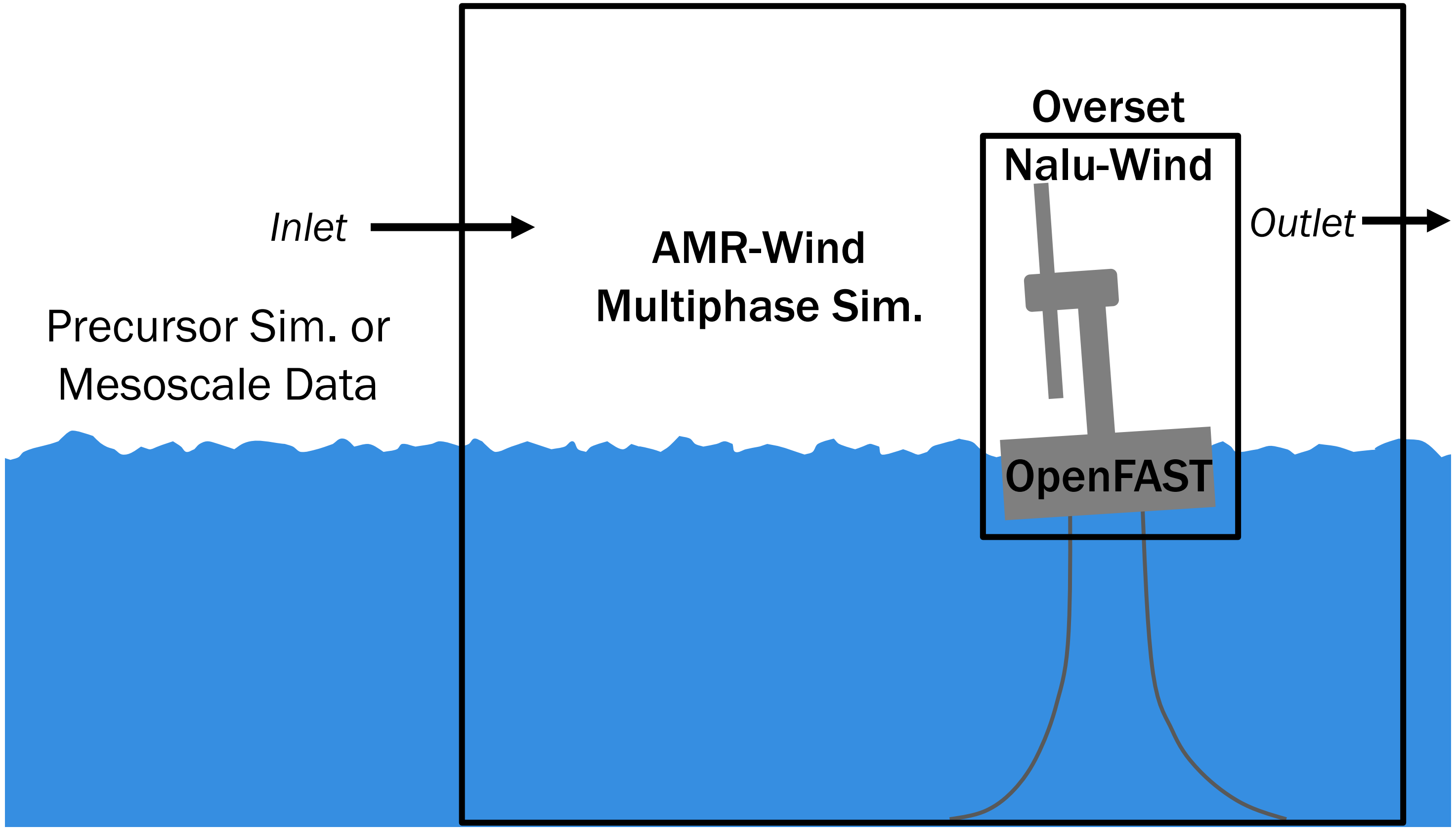
Exawind software suite

- 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

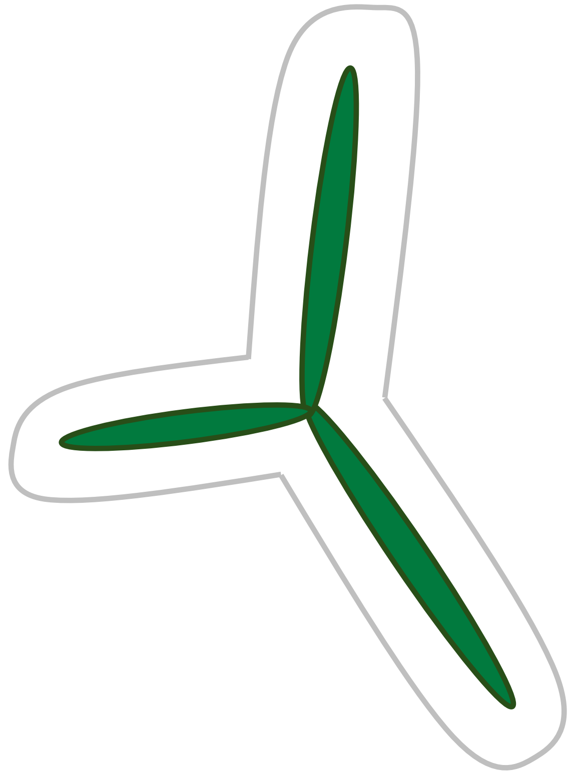
Nalu-Wind:
unstructured mesh, near-body flow dynamics

OpenFAST:
structural modeling

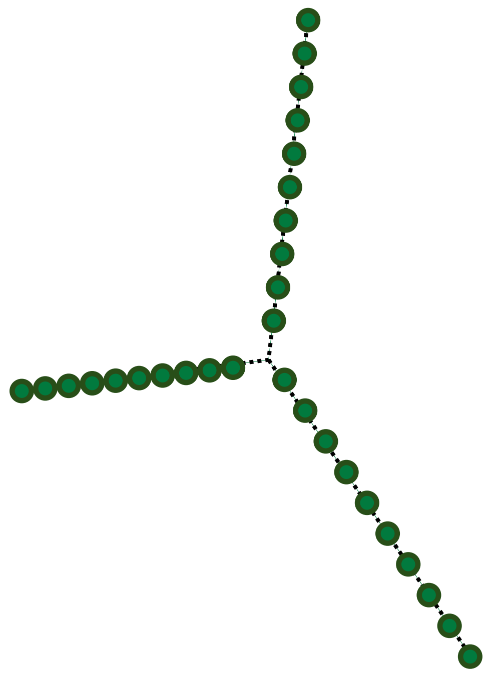


Exawind software suite

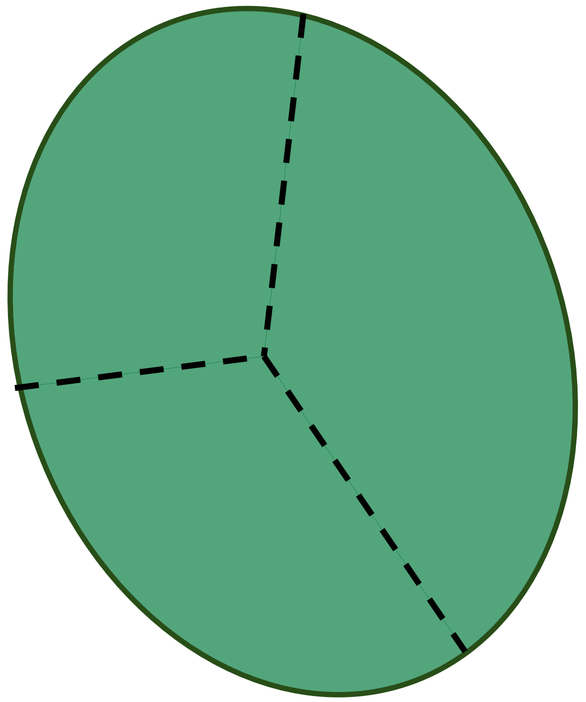
- Versatile, open-source, high-fidelity fluid mechanics and multi-fidelity turbine modeling



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 ‘#’

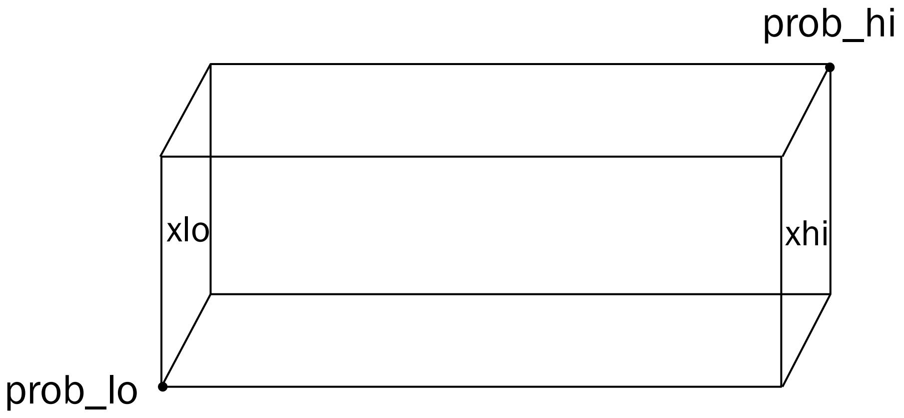
Anatomy of AMR-Wind input file

- 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/

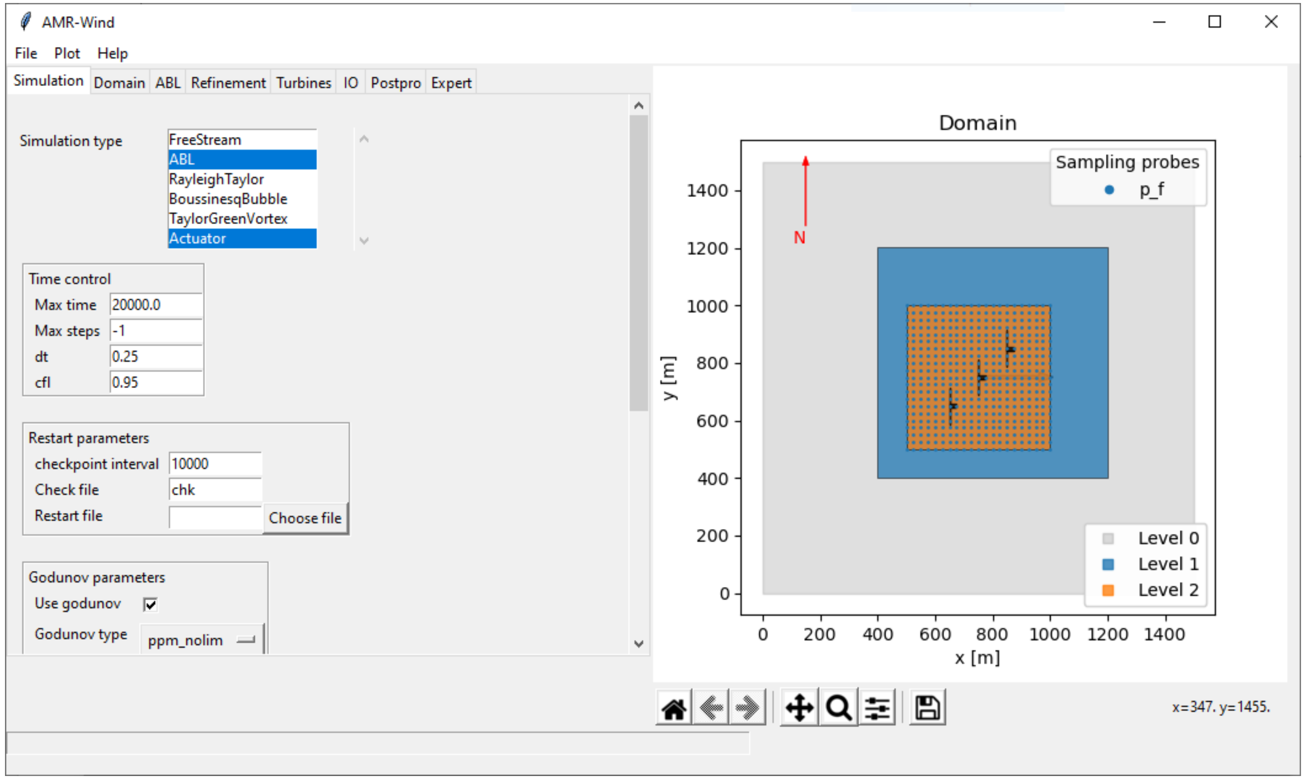
AMR-Wind: Actuator Disk example

- 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

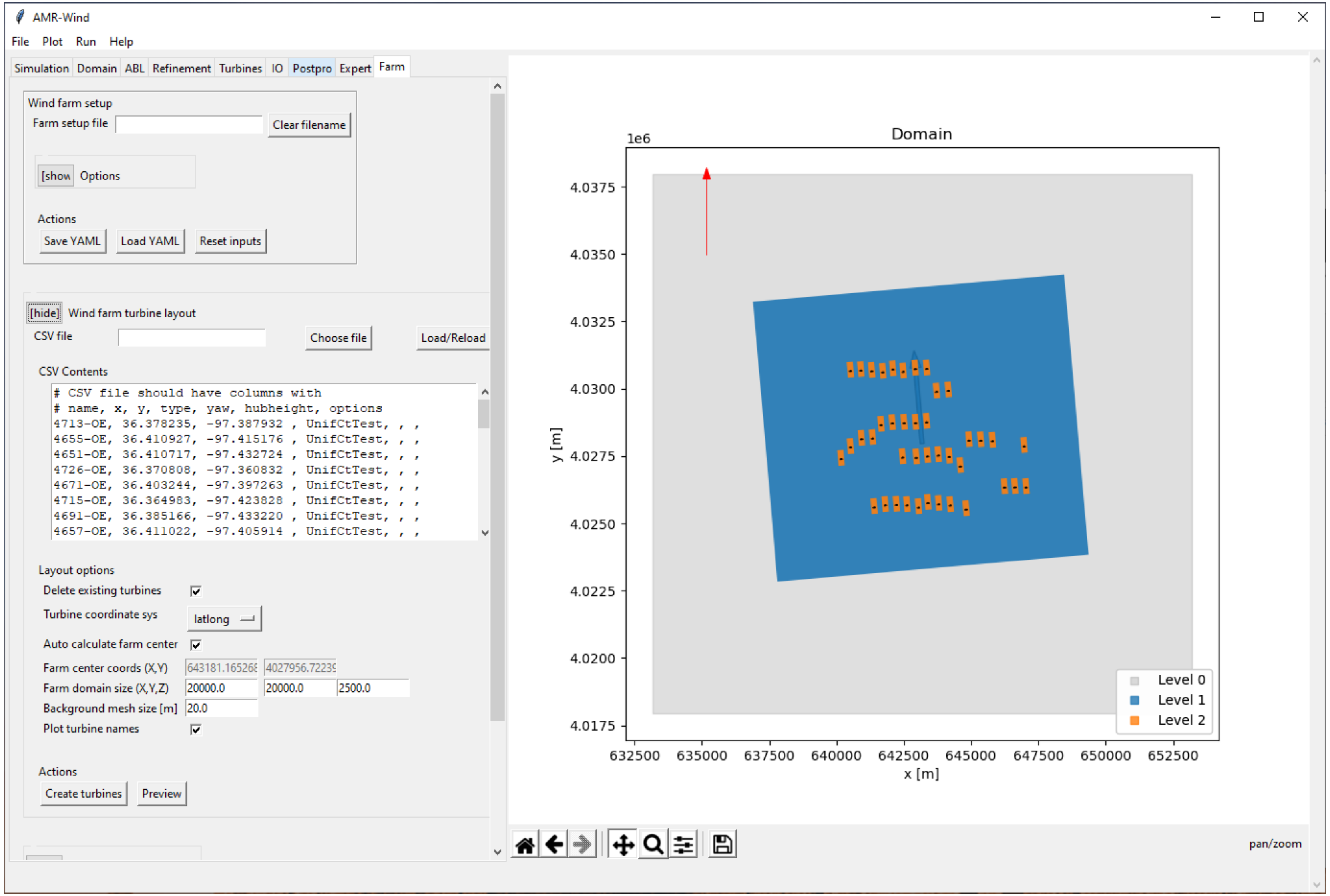
AMR-Wind frontend documentation available at
https://github.com/Exawind/amr-wind-frontend/tree/main/docs

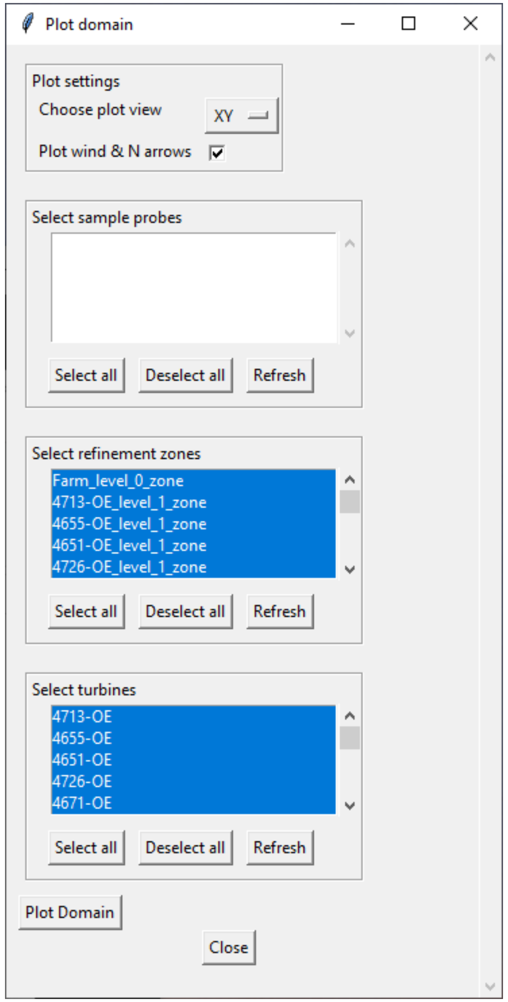
Contains installation, usage, and customization instructions

- 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





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)
```