OpenFAST User Guide

Tailored for Departmental Use

Version 1.0

Prepared by
Aron Vogelsang
Marine Civil Engineering
NTNU

Contact:
aronv@ntnu.no

Contents

1 Introduction

OpenFAST is an open source multiphysics wind turbine simulation tool developed by the National Renewable Energy Laboratory (NREL). It is designed to model-among other aspects-the coupled dynamic response of wind turbines and floating offshore structures. This guide provides a short overview on compilation and simulation, especially tailored for the needs of the Department of Marine Civil Engineering at NTNU, Trondheim. This guide does not aim to provide an in-depth understanding of the theory or the governing equations of the respective OpenFAST modules. Detailed insights on these topics can be found in the official OpenFAST documentation [3]. Additionally, numerous papers and reports on the development of the modules and OpenFAST, as the successor to FAST v8, are available on the official NREL webpage and forum [2]. Thus, the author makes no claim to completeness.

This User Guide is written for version 4.0.1 of OpenFAST.

2 Structural Overview

The highly modular coupling structure of OpenFAST allows users to customize simulation cases to their specific needs by including or excluding various modules within the software. While OpenFAST modules can operate independently in a standalone mode, there are still dependencies and variable exchanges between them.
As a result, the primary function of OpenFAST is to manage the communication and interaction between these modules, ensuring they are coupled in a physically consistent manner. This core functionality of OpenFAST is often referred to as the glue code. Figure 1 visualizes a possible selection of modules for the simulation case of a floating offshore wind turbine. Every model communicates in some way with the glue code. Certain modules, such as HydroDyn depend on the presence of other modules. To calculate the hydrodynamic loads on turbines a sea state (wave kinematics, water depth, current etc.) need to be defined in the SeaState module.

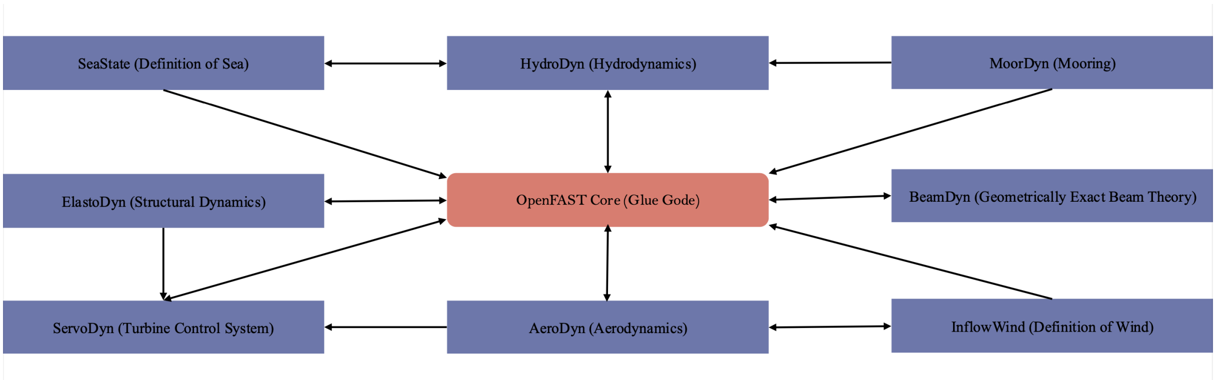

Figure 1: OpenFAST simulation example case for floating offshore wind

2.1 Overview over Modules

In this section the chosen modules in Figure 1 will be briefly defined. For insights over the other modules and an in-depth explanation see the official documentation [3].

- **AeroDyn** is the aerodynamic module, responsible for computing aerodynamic forces on wind turbine blades using Blade Element Momentum (BEM) theory and advanced wake models. It simulates rotor-wind interactions under various wind conditions and integrates with other modules for dynamic turbine analysis. To compute aerodynamic loads it is necessary for AeroDyn that InflowWind is implemented.

The main outputs are the aerodynamic forces and moments on blades, tower, and hub (e.g., lift, drag, pitching moments) and the rotor performance metrics like power, thrust, and torque.

- **InflowWind** is the wind input module, responsible for providing three dimensional wind and atmospheric flow conditions to the simulation. It supports a variety of wind profiles, from steady uniform flows to complex, turbulent wind fields, and can also incorporate user-defined wind data.

  The main outputs are wind velocity components at various points on the rotor and a three dimensional wind field.

- **ElastoDyn** is the structural dynamics module of OpenFAST, designed to simulate the flexible dynamic behavior of wind turbine components, including the blades, tower, drivetrain, and nacelle. It models how these structures respond to aerodynamic, hydrodynamic, and control system loads. Consequently, it accounts for rotational dynamics, gravity, and gyroscopic effects. ElastoDyn interacts with modules like AeroDyn, ServoDyn and HydroDyn for coupled system analysis.

  The main outputs are displacements, velocities, and accelerations of structural components, the internal forces and moments in blades and tower and the natural frequencies and mode shapes of the structure.

- **BeamDyn** is the high-fidelity structural dynamics module, designed to model blade flexibility with advanced geometrically exact beam theory. It provides more detailed and accurate simulations of complex blade deformations compared to ElastoDyn, especially applicable for large, flexible blades used in modern wind turbines. To capture nonlinear bending, torsion, and coupling effects in blades BeamDyn interacts with other modules like ElastoDyn and AeroDyn.

  The main outputs are high-fidelity and nonlinear deformations (bending, torsion) of blades and internal stresses and strains in blade cross-sections.

- **HydroDyn** is the hydrodynamic module, designed to calculate hydrodynamic forces and moments on offshore wind turbine substructures. It uses potential flow theory and strip theory to model wave-structure interactions by simulating hydrodynamic loads using linear and second-order wave theories. By inclduing HydroDyn modelling of buoyancy, added mass, and radiation damping effects will be enabled, while interacting with modules like MoorDyn for comprehensive offshore simulations. To compute hydrodynamic effects it is necessary for HydroDyn that SeaState is implemented.

  The main Outputs are hydrodynamic forces and moments on floating or fixed offshore structures, wave-induced loads and buoyancy effects and the respective added mass and damping coefficients.

- **SeaState** is used for generating wave fields and current profiles for hydrodynamic analysis. It can create regular and irregular wave conditions or incorporate externally generated wave kinematics.

  The main outputs are wave kinematics (velocities, accelerations, dynamic pressures, and wave elevation) time series at specified grid points.

- **MoorDyn** is the mooring dynamics module, designed to simulate the behavior of mooring lines and cables for floating offshore wind turbines and platforms. It models the dynamic response of mooring systems under environmental loads, considering tension, drag, and elasticity.

The main outputs are tensions in mooring lines and cables, line positions and seabed interactions, as well as the dynamic responses of mooring systems under environmental loads.

- **ServoDyn** is the control and actuator module, responsible for modeling the control systems and dynamic response of actuators in wind turbines. It manages blade pitch control, generator torque, and yaw systems, allowing for both simple and advanced control strategies. Controllers need to be pre-defined, prior to the actual simulation. An example fortran code for setting up the controller DISCON for NRELs fictive 5 MW turbine can be found in the official OpenFAST GitHub Repository.

  The main outputs are blade pitch angles, generator torque, and yaw position and the respective control system responses, including feedback signals and actuator behavior.

2.2 Code Structure of Modules

OpenFAST is written in the Fortran programming language. The user controls the glue code according to their needs through the main input file, hereinafter referred to as `Main.fst`. The `.fst` suffix is essential, as it signals to OpenFAST that this file serves as the primary configuration file. This file determines whether a module is included in the simulation.
Furthermore, each module has its own input file(s), where module-specific settings can be customized by the user. While the glue code manages the overall coupling and communication between the modules, the actual physical computations, governed by the relevant equations, are performed within the individual modules. Despite addressing different physical phenomena, the core structure of these modules remains comparable.
The `main()` program unit of each module can be found in the source code and is always named after the module itself, with a `.f90` suffix (e.g., for SeaState, `SeaState.f90`). This core function acts as a central switch: depending on the settings provided in the module's input file(s), specific routines within the code will be executed while others are skipped. Depending on the complexity of the case, functions or subroutines may be outsourced to other program units. In the Fortran programming language, these outsourced program units are also called *modules*. To avoid confusion with OpenFAST modules, the Fortran *modules* will be italicized in this guide.
The input data is handled by a Fortran *module*, named `ModuleNameInput.f90` (e.g., for SeaState, `SeaStateInput.f90`). This *module* contains a subroutine responsible for reading and parsing the user-provided input from the module's text file(s). Additionally, it transforms the input data into a format that facilitates efficient processing during simulation execution. Figure 2 illustrates the structure of the SeaState module to provide an overview of how modules function.
The role of the `...Types.f90` files is to define custom data types (known as derived data types, similar to structs or classes in object-oriented languages). These *modules* play a key role in organizing and managing complex data structures within the SeaState module and across coupled modules like HydroDyn and MoorDyn.
The actual computations in SeaState occur within the *modules* highlighted in red. Here, currents and waves are computed based on the user-selected theory. Besides calculating linear and second order waves, SeaState is able to read wavekinematics from external generated wave fields, this is integrated via the `UserWaves.f90` *module*. A description, on how to utilize this option is given in Section 4.4. `SeaSt_WaveField.f90` determines, based on user input, how the different *modules* are integrated.
Every module in OpenFAST can generate a separate summary output file. If this is desired, it must be specified by the user in the respective input file.
`SeaState.f90` is also responsible for communication with the glue code when the SeaState module is invoked during an OpenFAST simulation. If SeaState is used to simulate wave kinematics

via the standalone driver, an additional text file is required to set relevant parameters—such as time-stepping—that would typically be configured by the OpenFAST `Main.fst` file.

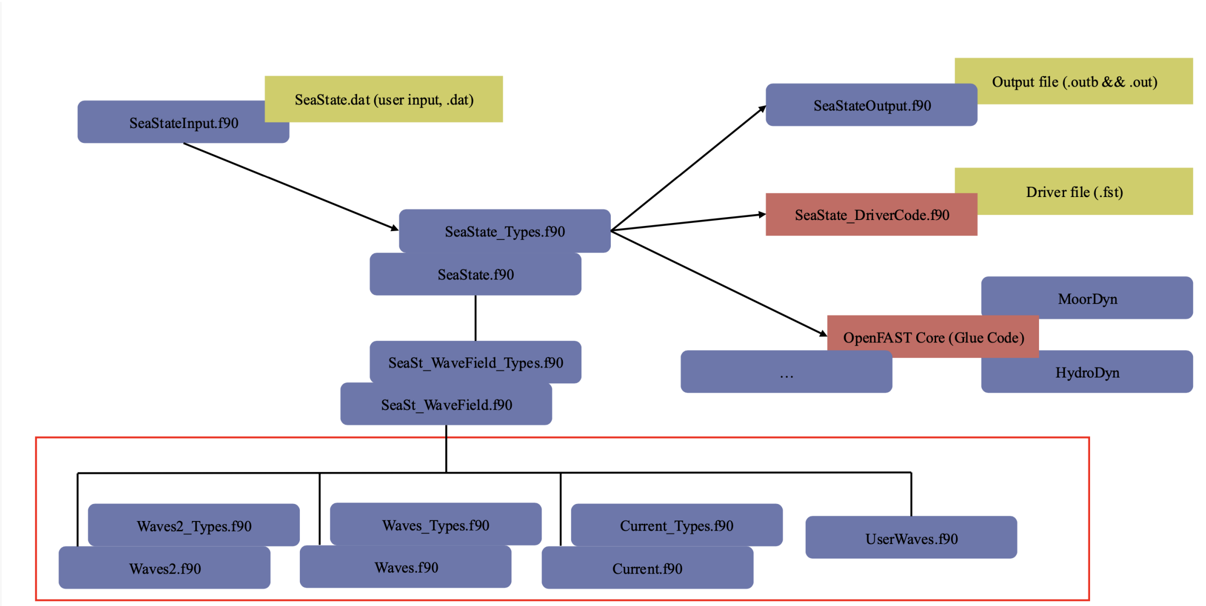

Figure 2: Pipeline model of the SeaStae module as example

3 Installation

The Department of Marine Civil Engineering and the Research Group led by Hans Bihs operate primarily on Apple macOS systems. As a result, this installation guide is specifically valid for macOS environments. For more detailed information on other installation methods, see the official OpenFAST documentation [3].

3.1 Required Dependencies

Compiling OpenFAST requires additional libraries and tools that are not distributed with the OpenFAST repository. Each of the following components is required for the minimum OpenFAST compilation.

- C++, C, and Fortran compiler
- BLAS and LAPACK math library
- Build system

In many cases, these tools can be installed using a system's package manager (e.g., Homebrew for macOS). Hence, the following command installs all required dependencies:

```
brew install git cmake make openblas gcc
```

The dependencies can also be downloaded directly from the vendors. Links to possible download pages:

- GNU Compiler Collection (gfortran, gcc, g++)
- BLAS/LAPACK by OpenBLAS
- CMake

3.1.1 Setting Environment Variables

This step might be negligible if the dependencies are installed via package manager. After installing the necessary dependencies, it is important to ensure that OpenFAST can locate them. This requires adding the installed libraries and tools to your systems PATH environment variable. Follow the steps below to configure your environment correctly:

1. Open your terminal.
2. Edit your shell configuration file (e.g., `.bash_profile`, `.bashrc`, or `.zshrc`) using a text editor (if you do not have the required rights to edit in the text editor, write `sudo` before `nano`):

```
nano ~/.bash_profile  # For bash users
nano ~/.zshrc         # For zsh users (default on macOS)
```

3. Add the following lines to include the paths (here example paths used) of the installed tools and libraries:

```
export PATH="/usr/local/bin:$PATH"
export LIBRARY_PATH="/usr/local/lib:$LIBRARY_PATH"
export C_INCLUDE_PATH="/usr/local/include:$C_INCLUDE_PATH"
export LD_LIBRARY_PATH="/usr/local/lib:$LD_LIBRARY_PATH"
```

4. To save the file and exit the editor press `Ctrl + O` (`O`, like in Oven, not `ZERO`) to write out and `Ctrl + X` to exit in `nano`.
5. Apply the changes by sourcing the configuration file:

```
source ~/.bash_profile  # For bash users
source ~/.zshrc         # For zsh users
```

After these steps, the system will recognize the installed dependencies, and OpenFAST should compile without issues.

3.2 Installation Steps

1. Clone the OpenFAST repository, preferably in your home directory:

```
git clone https://github.com/OpenFAST/openfast.git
```

2. Create a build directory in the OpenFAST folder:

```
cd OpenFAST
mkdir build && cd build
```

3. Run CMake:

```
cmake ..
```

Note: There are multiple `cmake ..` commands for adaption. Here the standard configuration is executed. For possible adaptions, including setting up the FastFarm driver, see CMake Options from the OpenFAST documentation.

The CMake project is configured to search for the required math libraries in default locations. However, if math libraries are not found, they can be specified directly to CMake. The two required libraries are BLAS and LAPACK, and their location can be passed to CMake with this command syntax:

```
cmake .. -DBLAS_LIBRARIES="/path/to/blas" -DLAPACK_LIBRARIES="/path/to/lapack"
```

The paths given should be to the directory which contains the libraries, not to the libraries themselves.

4. Compile the code:

```
make -jn
```

where n is the number of processor cores you want to use for compilation. This will compile OpenFAST glue code and all standalone module drivers. It is recommendable to move the OpenFAST executable out of the `OpenFAST/build/glue-codes/openfast` directory into a working directory. You can call it for example: `simulation` in the main `OpenFAST` folder.

3.3 Additional Tools

3.3.1 MATLAB and Python Toolboxes

Useful toolboxes for working with OpenFAST can be found on the OpenFAST GitHub. Various tools (e. g. libraries and code) are available for different purposes. Output data can be post-processed, and file formats can be converted between binary and text. Additionally, there are tools to assist with generating input files and modeling wind turbines.

3.3.2 Example Files

Due to the complex structure of the input files required for an OpenFAST simulation, guidelines on how to set up these files are essential. To gain an overview, various example cases of existing OpenFAST simulations can be found on the `OpenFAST/r-test` GitHub repository. It is highly recommended to clone this repository into your OpenFAST working directory. Example cases for the glue code can be found under: `r-test/glue-codes/openfast/`. Files to run specific modules with the standalone driver are located in `r-test/modules/ModuleName/`. In your work with OpenFAST, you will encounter the `NRELOffshr5MW` turbine. This is a fictitious 5-MW reference wind turbine designed for offshore system development by NREL [1]. Various cases examine this turbine, allowing for easy file replacement and adaptations between simulations.

3.3.3 Standalone Module Drivers

Standalone drivers for individual modules can be found in `OpenFAST/build/modules/ModuleName`. When compiled, in the manner previously described, no additional compilation of the drivers is necessary. If you wish to compile a specific standalone driver only, you can use the following command during compilation:
`make ModuleName`.

4 Running Simulations

4.1 General Execution of OpenFAST

To successfully run simulations, OpenFAST needs to know where it can find the required input files. To have everything organized, it is suitable to have a working directory, called for example `simulation`, from which the simulations are piloted. In the working directory, besides the OpenFAST executable, all the files will be placed, that are required for the specific simulation. To get a feeling for this process, the example case `MinimalExample` can be carried out. You can get the Minimal Example from the `r-test` glue code files introduced in Section 3.3.2. Copy all the files, that the `MinimalExample` directory contains, in the `simulation` directory. You will find following relevant files:

- `ElastoDyn.dat`: Contains main settings for the ElastoDyn module
- `ElastoDyn_Blade.dat`: Contains the definition of the Blade for ElastoDyn.
- `ElastoDyn_Tower.dat` Contains the definition of the Tower for ElastoDyn.
- `Main.fst` Main file

To run the simulation the terminal command

```
./openfast Main.fst
```

is used from the simulation directory. This is the common way for every OpenFAST simulation.

4.2 Configuring Input Files

To further enhance the understanding of working with OpenFAST input files, this section discusses another example: the 5MW OC3 Spar Variant with a turbulent wind field and irregular waves, denoted as `5MW_OC3Spar_DLL_WTurb_WavesIrr`. This example is also included in the r-test directory of OpenFAST.
The following sections contain screenshots of key portions from various input files used in this simulation. Each section explains important modifications that users can make. The discussion begins with the `Main.fst` file, specifically the case file named `5MW_OC3Spar_DLL_WTurb_WavesIrr.fst`.

4.2.1 `5MW_OC3Spar_DLL_WTurb_WavesIrr.fst`

```
---------------------- SIMULATION CONTROL --------------------------------------
False         Echo           - Echo input data to <RootName>.ech (flag)
"FATAL"       AbortLevel     - Error level when simulation should abort (string) {"WARNING", "SEVERE", "FATAL"}
          60   TMax           - Total run time (s)
      0.0125   DT             - Recommended module time step (s)
           1   InterpOrder    - Interpolation order for input/output time history (-) {1=linear, 2=quadratic}
           0   NumCrctn       - Number of correction iterations (-) {0=explicit calculation, i.e., no corrections}
         1.5   DT_UJac        - Time between calls to get Jacobians (s)
     1000000   UJacSclFact    - Scaling factor used in Jacobians (-)
```

Figure 3: Simulation control section of the `Main.fst` file. The timestep and total simulation duration can be adjusted here

```
--------------------- FEATURE SWITCHES AND FLAGS ------------------------------
          1   CompElast       - Compute structural dynamics (switch) {1=ElastoDyn; 2=ElastoDyn + BeamDyn for blades; 3=Simplified ElastoDyn}
          1   CompInflow      - Compute inflow wind velocities (switch) {0=still air; 1=InflowWind; 2=external from ExtInflow}
          2   CompAero        - Compute aerodynamic loads (switch) {0=None; 1=AeroDisk; 2=AeroDyn; 3=ExtLoads}
          1   CompServo       - Compute control and electrical-drive dynamics (switch) {0=None; 1=ServoDyn}
          1   CompSeaSt       - Compute sea state information (switch) {0=None; 1=SeaState}
          1   CompHydro       - Compute hydrodynamic loads (switch) {0=None; 1=HydroDyn}
          0   CompSub         - Compute sub-structural dynamics (switch) {0=None; 1=SubDyn; 2=External Platform MCKF}
          1   CompMooring     - Compute mooring system (switch) {0=None; 1=MAP++; 2=FEAMooring; 3=MoorDyn; 4=OrcaFlex}
          0   CompIce         - Compute ice loads (switch) {0=None; 1=IceFloe; 2=IceDyn}
          0   MHK             - MHK turbine type (switch) {0=Not an MHK turbine; 1=Fixed MHK turbine; 2=Floating MHK turbine}
```

Figure 4: Module selection section in the `Main.fst` file. The user specifies which OpenFAST modules to include in the simulation.

```
--------------------- ENVIRONMENTAL CONDITIONS --------------------------------
    9.80665   Gravity         - Gravitational acceleration (m/s^2)
      1.225   AirDens         - Air density (kg/m^3)
       1025   WtrDens         - Water density (kg/m^3)
  1.464E-05   KinVisc         - Kinematic viscosity of working fluid (m^2/s)
        335   SpdSound        - Speed of sound in working fluid (m/s)
     103500   Patm            - Atmospheric pressure (Pa) [used only for an MHK turbine cavitation check]
       1700   Pvap            - Vapour pressure of working fluid (Pa) [used only for an MHK turbine cavitation check]
        320   WtrDpth         - Water depth (m)
          0   MSL2SWL         - Offset between still-water level and mean sea level (m) [positive upward]
```

Figure 5: Environmental conditions section in `Main.fst`. The user defines key environmental parameters such as air density, gravity, and atmospheric conditions.

```
--------------------- INPUT FILES ---------------------------------------------
"NRELOffshrBsline5MW_OC3Hywind_ElastoDyn.dat"    EDFile          - Name of file containing ElastoDyn input parameters (quoted string)
"../5MW_Baseline/NRELOffshrBsline5MW_BeamDyn.dat"    BDBldFile(1)    - Name of file containing BeamDyn input parameters for blade 1 (quoted string)
"../5MW_Baseline/NRELOffshrBsline5MW_BeamDyn.dat"    BDBldFile(2)    - Name of file containing BeamDyn input parameters for blade 2 (quoted string)
"../5MW_Baseline/NRELOffshrBsline5MW_BeamDyn.dat"    BDBldFile(3)    - Name of file containing BeamDyn input parameters for blade 3 (quoted string)
"../5MW_Baseline/NRELOffshrBsline5MW_InflowWind_12mps.dat"    InflowFile      - Name of file containing inflow wind input parameters (quoted string)
"NRELOffshrBsline5MW_OC3Hywind_AeroDyn.dat"    AeroFile        - Name of file containing aerodynamic input parameters (quoted string)
"NRELOffshrBsline5MW_OC3Hywind_ServoDyn.dat"    ServoFile       - Name of file containing control and electrical-drive input parameters (quoted string)
"SeaState.dat"    SeaStFile       - Name of file containing sea state input parameters (quoted string)
"NRELOffshrBsline5MW_OC3Hywind_HydroDyn.dat"    HydroFile       - Name of file containing hydrodynamic input parameters (quoted string)
"unused"      SubFile         - Name of file containing sub-structural input parameters (quoted string)
"NRELOffshrBsline5MW_OC3Hywind_MAP.dat"    MooringFile     - Name of file containing mooring system input parameters (quoted string)
"unused"      IceFile         - Name of file containing ice input parameters (quoted string)
```

Figure 6: Input file configuration section in `Main.fst`. The user specifies the filenames for module-specific input files. If files are not located in the working directory, a relative path must be provided.

```
--------------------- VISUALIZATION ------------------------------------------
          0   WrVTK           - VTK visualization data output: (switch) {0=none; 1=initialization data only; 2=animation; 3=mode shapes}
          2   VTK_type        - Type of VTK visualization data: (switch) {1=surfaces; 2=basic meshes (lines/points); 3=all meshes (debug)} [unused if WrVTK=0]
false         VTK_fields      - Write mesh fields to VTK data files? (flag) {true/false} [unused if WrVTK=0]
         15   VTK_fps         - Frame rate for VTK output (frames per second){will use closest integer multiple of DT} [used only if WrVTK=2 or WrVTK=3]
```

Figure 7: Visualization settings in `Main.fst`. The user can enable output of `.vtp` files for post-processing in ParaView.

4.2.2 `NRELOffshrBsline5MW_OC3Hywind_ElastoDyn.dat`

In Figure 4, the modules included in this simulation are selected. Figure 6 lists the corresponding input files. The first module, ElastoDyn, is configured using its primary input file: `NRELOffshrBsline5MW_OC3Hywind_ElastoDyn.dat`.

```
---------------------- DEGREES OF FREEDOM --------------------------------------
True          FlapDOF1    - First flapwise blade mode DOF (flag)
True          FlapDOF2    - Second flapwise blade mode DOF (flag)
True          EdgeDOF     - First edgewise blade mode DOF (flag)
False         TeetDOF     - Rotor-teeter DOF (flag) [unused for 3 blades]
True          DrTrDOF     - Drivetrain rotational-flexibility DOF (flag)
True          GenDOF      - Generator DOF (flag)
True          YawDOF      - Yaw DOF (flag)
True          TwFADOF1    - First fore-aft tower bending-mode DOF (flag)
True          TwFADOF2    - Second fore-aft tower bending-mode DOF (flag)
True          TwSSDOF1    - First side-to-side tower bending-mode DOF (flag)
True          TwSSDOF2    - Second side-to-side tower bending-mode DOF (flag)
True          PtfmSgDOF   - Platform horizontal surge translation DOF (flag)
True          PtfmSwDOF   - Platform horizontal sway translation DOF (flag)
True          PtfmHvDOF   - Platform vertical heave translation DOF (flag)
True          PtfmRDOF    - Platform roll tilt rotation DOF (flag)
True          PtfmPDOF    - Platform pitch tilt rotation DOF (flag)
True          PtfmYDOF    - Platform yaw rotation DOF (flag)
```

Figure 8: Degrees of freedom section from the ElastoDyn input file. The user can impose positional constraints on specific turbine components by setting the corresponding flag to `false`.

```
---------------------- INITIAL CONDITIONS --------------------------------------
          0   OoPDefl     - Initial out-of-plane blade-tip displacement (meters)
          0   IPDefl      - Initial in-plane blade-tip deflection (meters)
          0   BlPitch(1)  - Blade 1 initial pitch (degrees)
          0   BlPitch(2)  - Blade 2 initial pitch (degrees)
          0   BlPitch(3)  - Blade 3 initial pitch (degrees) [unused for 2 blades]
          0   TeetDefl    - Initial or fixed teeter angle (degrees) [unused for 3 blades]
          0   Azimuth     - Initial azimuth angle for blade 1 (degrees)
       12.1   RotSpeed    - Initial or fixed rotor speed (rpm)
          0   NacYaw      - Initial or fixed nacelle-yaw angle (degrees)
          0   TTDspFA     - Initial fore-aft tower-top displacement (meters)
          0   TTDspSS     - Initial side-to-side tower-top displacement (meters)
          0   PtfmSurge   - Initial or fixed horizontal surge translational displacement of platform (meters)
          0   PtfmSway    - Initial or fixed horizontal sway translational displacement of platform (meters)
          0   PtfmHeave   - Initial or fixed vertical heave translational displacement of platform (meters)
          0   PtfmRoll    - Initial or fixed roll tilt rotational displacement of platform (degrees)
          0   PtfmPitch   - Initial or fixed pitch tilt rotational displacement of platform (degrees)
          0   PtfmYaw     - Initial or fixed yaw rotational displacement of platform (degrees)
```

Figure 9: Initial conditions section from the ElastoDyn input file. Initial conditions for the turbine can be set here. In this example, the initial rotor speed is configured to 12.1 rpm.

```
---------------------- TURBINE CONFIGURATION -----------------------------------
          3   NumBl       - Number of blades (-)
         63   TipRad      - The distance from the rotor apex to the blade tip (meters)
        1.5   HubRad      - The distance from the rotor apex to the blade root (meters)
       -2.5   PreCone(1)  - Blade 1 cone angle (degrees)
       -2.5   PreCone(2)  - Blade 2 cone angle (degrees)
       -2.5   PreCone(3)  - Blade 3 cone angle (degrees) [unused for 2 blades]
          0   HubCM       - Distance from rotor apex to hub mass [positive downwind] (meters)
          0   UndSling    - Undersling length [distance from teeter pin to the rotor apex] (meters) [unused for 3 blades]
          0   Delta3      - Delta-3 angle for teetering rotors (degrees) [unused for 3 blades]
          0   AzimB1Up    - Azimuth value to use for I/O when blade 1 points up (degrees)
    -5.0191   OverHang    - Distance from yaw axis to rotor apex [3 blades] or teeter pin [2 blades] (meters)
      1.912   ShftGagL    - Distance from rotor apex [3 blades] or teeter pin [2 blades] to shaft strain gages [positive for upwind rotors
         -5   ShftTilt    - Rotor shaft tilt angle (degrees)
        1.9   NacCMxn     - Downwind distance from the tower-top to the nacelle CM (meters)
          0   NacCMyn     - Lateral  distance from the tower-top to the nacelle CM (meters)
       1.75   NacCMzn     - Vertical distance from the tower-top to the nacelle CM (meters)
   -3.09528   NcIMUxn     - Downwind distance from the tower-top to the nacelle IMU (meters)
          0   NcIMUyn     - Lateral  distance from the tower-top to the nacelle IMU (meters)
    2.23336   NcIMUzn     - Vertical distance from the tower-top to the nacelle IMU (meters)
    1.96256   Twr2Shft    - Vertical distance from the tower-top to the rotor shaft (meters)
       87.6   TowerHt     - Height of tower relative to ground level [onshore], MSL [offshore wind or floating MHK], or seabed [fixed MHK]
         10   TowerBsHt   - Height of tower base relative to ground level [onshore], MSL [offshore wind or floating MHK], or seabed [fixed
          0   PtfmCMxt    - Downwind distance from the ground level [onshore], MSL [offshore wind or floating MHK], or seabed [fixed MHK]
          0   PtfmCMyt    - Lateral distance from the ground level [onshore], MSL [offshore wind or floating MHK], or seabed [fixed MHK] t
   -89.9155   PtfmCMzt    - Vertical distance from the ground level [onshore], MSL [offshore wind or floating MHK], or seabed [fixed MHK]
         -0   PtfmRefzt   - Vertical distance from the ground level [onshore], MSL [offshore wind or floating MHK], or seabed [fixed MHK]
```

Figure 10: Turbine configuration section from the ElastoDyn input file. This section defines essential turbine parameters. The user specifies the number of blades and the tower height, which are critical for the blade and tower input files discussed later in the ElastoDyn module.

```
---------------------- MASS AND INERTIA ----------------------------------------
          0   TipMass(1)  - Tip-brake mass, blade 1 (kg)
          0   TipMass(2)  - Tip-brake mass, blade 2 (kg)
          0   TipMass(3)  - Tip-brake mass, blade 3 (kg) [unused for 2 blades]
      56780   HubMass     - Hub mass (kg)
     115926   HubIner     - Hub inertia about rotor axis [3 blades] or teeter axis [2 blades] (kg m^2)
    534.116   GenIner     - Generator inertia about HSS (kg m^2)
     240000   NacMass     - Nacelle mass (kg)
    2607890   NacYIner    - Nacelle inertia about yaw axis (kg m^2)
          0   YawBrMass   - Yaw bearing mass (kg)
    7466330   PtfmMass    - Platform mass (kg)
4.22923E+09   PtfmRIner   - Platform inertia for roll tilt rotation about the platform CM (kg m^2)
4.22923E+09   PtfmPIner   - Platform inertia for pitch tilt rotation about the platform CM (kg m^2)
  164230000   PtfmYIner   - Platform inertia for yaw rotation about the platform CM (kg m^2)
          0   PtfmXYIner  - Platform xy moment of inertia about the platform CM (=-int(xydm)) (kg m^2)
          0   PtfmYZIner  - Platform yz moment of inertia about the platform CM (=-int(yzdm)) (kg m^2)
          0   PtfmXZIner  - Platform xz moment of inertia about the platform CM (=-int(xzdm)) (kg m^2)
```

Figure 11: Mass and inertia section from the ElastoDyn input file. The turbine's mass plays a key role in determining its floating behavior and response to wave loads.

```
---------------------- BLADE ---------------------------------------------------
         17   BldNodes    - Number of blade nodes (per blade) used for analysis (-)
"../5MW_Baseline/NRELOffshrBsline5MW_Blade.dat"    BldFile(1)  - Name of file containing properties for blade 1 (quoted string)
"../5MW_Baseline/NRELOffshrBsline5MW_Blade.dat"    BldFile(2)  - Name of file containing properties for blade 2 (quoted string)
"../5MW_Baseline/NRELOffshrBsline5MW_Blade.dat"    BldFile(3)  - Name of file containing properties for blade 3 (quoted string) [unused fo
```

Figure 12: Blade section from the ElastoDyn input file. The user must specify the location of the additional blade input file, which contains the blade's structural and aerodynamic properties.

```
---------------------- DRIVETRAIN ----------------------------------------------
        100   GBoxEff     - Gearbox efficiency (%)
         97   GBRatio     - Gearbox ratio (-)
  867637000   DTTorSpr    - Drivetrain torsional spring (N-m/rad)
    6215000   DTTorDmp    - Drivetrain torsional damper (N-m/(rad/s))
```

Figure 13: Drivetrain section from the ElastoDyn input file. Key drivetrain parameters can be adjusted in this section.

```
---------------------- TOWER ---------------------------------------------------
         20   TwrNodes    - Number of tower nodes used for analysis (-)
"NRELOffshrBsline5MW_OC3Hywind_ElastoDyn_Tower.dat"    TwrFile     - Name of file containing tower properties (quoted string)
```

Figure 14: Tower section from the ElastoDyn input file. An additional input file is required to define the tower parameters. The user must specify the filename for this configuration.

```
---------------------- OUTPUT --------------------------------------------------
False         SumPrint    - Print summary data to "<RootName>.sum" (flag)
            1 OutFile     - Switch to determine where output will be placed: {1: in module output file only; 2: in glue code output file
True          TabDelim    - Use tab delimiters in text tabular output file? (flag) (currently unused)
"ES10.3E2"    OutFmt      - Format used for text tabular output (except time).  Resulting field should be 10 characters. (quoted string)
           30 TStart      - Time to begin tabular output (s) (currently unused)
            1 DecFact     - Decimation factor for tabular output {1: output every time step} (-) (currently unused)
            1 NTwGages    - Number of tower nodes that have strain gages for output [0 to 9] (-)
           10 TwrGagNd    - List of tower nodes that have strain gages [1 to TwrNodes] (-) [unused if NTwGages=0]
            1 NBlGages    - Number of blade nodes that have strain gages for output [0 to 9] (-)
            9 BldGagNd    - List of blade nodes that have strain gages [1 to BldNodes] (-) [unused if NBlGages=0]
              OutList     - The next line(s) contains a list of output parameters.  See OutListParameters.xlsx for a listing of available
"BldPitch1"               - Pitch angles for blades 1, 2, and 3
"BldPitch2"               - Pitch angles for blades 1, 2, and 3
"BldPitch3"               - Pitch angles for blades 1, 2, and 3
"Azimuth"                 - Blade 1 azimuth angle
"RotSpeed"                - Low-speed shaft and high-speed shaft speeds
"GenSpeed"                - Low-speed shaft and high-speed shaft speeds
"NacYaw"                  - Nacelle yaw angle and nacelle yaw error estimate
"OoPDefl1"                - Blade 1 out-of-plane and in-plane deflections and tip twist
"IPDefl1"                 - Blade 1 out-of-plane and in-plane deflections and tip twist
"TwstDefl1"               - Blade 1 out-of-plane and in-plane deflections and tip twist
"OoPDefl2"                - Blade 2 out-of-plane and in-plane deflections and tip twist
"IPDefl2"                 - Blade 2 out-of-plane and in-plane deflections and tip twist
"TwstDefl2"               - Blade 2 out-of-plane and in-plane deflections and tip twist
"OoPDefl3"                - Blade 3 out-of-plane and in-plane deflections and tip twist
"IPDefl3"                 - Blade 3 out-of-plane and in-plane deflections and tip twist
"TwstDefl3"               - Blade 3 out-of-plane and in-plane deflections and tip twist
"TwrClrnc1"               - Tip-to-tower clearance estimate for blades 1, 2, and 3
"TwrClrnc2"               - Tip-to-tower clearance estimate for blades 1, 2, and 3
"TwrClrnc3"               - Tip-to-tower clearance estimate for blades 1, 2, and 3
"NcIMUTAxs"               - Nacelle IMU translational accelerations (absolute) in the nonrotating, shaft coordinate system
"NcIMUTAys"               - Nacelle IMU translational accelerations (absolute) in the nonrotating, shaft coordinate system
```

Figure 15: Output section from the ElastoDyn input file. Each module can generate its own output. All available output channels are listed and explained at the end of the respective module input file. The output data can be extensive and detailed—ElastoDyn, for instance, provides up to 81 different output columns.

4.2.3 `NRELOffshrBsline5MW_Blade.dat`

As shown in Figure 12, ElastoDyn requires an additional input file containing blade parameters. Since this blade file is used across multiple simulations, it is stored in a separate directory. The user can either copy it into the simulation folder or specify the relative path in the input section. In this simulation, the same input file is used for all three blades. This file primarily defines material properties, spatial configurations, and computational settings for the blade.

```
------- ELASTODYN V1.00.* INDIVIDUAL BLADE INPUT FILE --------------------------
NREL 5.0 MW offshore baseline blade input properties.
---------------------- BLADE PARAMETERS ----------------------------------------
          49   NBlInpSt    - Number of blade input stations (-)
    0.477465   BldFlDmp(1) - Blade flap mode #1 structural damping in percent of critical (%)
    0.477465   BldFlDmp(2) - Blade flap mode #2 structural damping in percent of critical (%)
    0.477465   BldEdDmp(1) - Blade edge mode #1 structural damping in percent of critical (%)
---------------------- BLADE ADJUSTMENT FACTORS --------------------------------
           1   FlStTunr(1) - Blade flapwise modal stiffness tuner, 1st mode (-)
           1   FlStTunr(2) - Blade flapwise modal stiffness tuner, 2nd mode (-)
     1.04536   AdjBlMs     - Factor to adjust blade mass density (-)  !bjj: value for AD14=1.04536; value for AeroDyn=1.057344 (it would be
           1   AdjFlSt     - Factor to adjust blade flap stiffness (-)
           1   AdjEdSt     - Factor to adjust blade edge stiffness (-)
---------------------- DISTRIBUTED BLADE PROPERTIES ----------------------------
    BlFract      PitchAxis      StrcTwst       BMassDen        FlpStff        EdgStff
      (-)           (-)          (deg)          (kg/m)         (Nm^2)         (Nm^2)
0.0000000E+00  2.5000000E-01  1.3308000E+01  6.7893500E+02  1.8110000E+10  1.8113600E+10
3.2500000E-03  2.5000000E-01  1.3308000E+01  6.7893500E+02  1.8110000E+10  1.8113600E+10
1.9510000E-02  2.5049000E-01  1.3308000E+01  7.7336300E+02  1.9424900E+10  1.9558600E+10
3.5770000E-02  2.5490000E-01  1.3308000E+01  7.4055000E+02  1.7455900E+10  1.9497800E+10
5.2030000E-02  2.6716000E-01  1.3308000E+01  7.4004200E+02  1.5287400E+10  1.9788800E+10
6.8290000E-02  2.7941000E-01  1.3308000E+01  5.9249600E+02  1.0782400E+10  1.4858500E+10
8.4550000E-02  2.9167000E-01  1.3308000E+01  4.5027500E+02  7.2297200E+09  1.0220600E+10
1.0081000E-01  3.0392000E-01  1.3308000E+01  4.2405400E+02  6.3095400E+09  9.1447000E+09
1.1707000E-01  3.1618000E-01  1.3308000E+01  4.0063800E+02  5.5283600E+09  8.0631600E+09
1.3335000E-01  3.2844000E-01  1.3308000E+01  3.8206200E+02  4.9800600E+09  6.8844400E+09
1.4959000E-01  3.4069000E-01  1.3308000E+01  3.9965500E+02  4.9368400E+09  7.0091800E+09
1.6585000E-01  3.5294000E-01  1.3308000E+01  4.2632100E+02  4.6916600E+09  7.1676800E+09
1.8211000E-01  3.6519000E-01  1.3181000E+01  4.1682000E+02  3.9494600E+09  7.2716600E+09
1.9837000E-01  3.7500000E-01  1.2848000E+01  4.0618600E+02  3.3865200E+09  7.0817000E+09
2.1465000E-01  3.7500000E-01  1.2192000E+01  3.8142000E+02  2.9337400E+09  6.2445300E+09
2.3089000E-01  3.7500000E-01  1.1561000E+01  3.5282200E+02  2.5689600E+09  5.0489600E+09
2.4715000E-01  3.7500000E-01  1.1072000E+01  3.4947700E+02  2.3886500E+09  4.9484900E+09
2.6341000E-01  3.7500000E-01  1.0792000E+01  3.4653800E+02  2.2719900E+09  4.8080200E+09
2.9595000E-01  3.7500000E-01  1.0232000E+01  3.3933300E+02  2.0500500E+09  4.5014000E+09
```

Figure 16: The blade input file, containing structural and material properties.

4.2.4 NRELOffshrBsline5MW_Onshore_ElastoDyn_Tower.dat

As indicated in Figure 14, an additional input file is required to define the tower parameters. The structure of this file is similar to that of the blade input file.

```
------- ELASTODYN V1.00.* TOWER INPUT FILE -------------------------------------
NREL 5.0 MW offshore baseline tower input properties.
---------------------- TOWER PARAMETERS ----------------------------------------
          11   NTwInpSt    - Number of input stations to specify tower geometry
           1   TwrFADmp(1) - Tower 1st fore-aft mode structural damping ratio (%)
           1   TwrFADmp(2) - Tower 2nd fore-aft mode structural damping ratio (%)
           1   TwrSSDmp(1) - Tower 1st side-to-side mode structural damping ratio (%)
           1   TwrSSDmp(2) - Tower 2nd side-to-side mode structural damping ratio (%)
---------------------- TOWER ADJUSTMUNT FACTORS --------------------------------
           1   FAStTunr(1) - Tower fore-aft modal stiffness tuner, 1st mode (-)
           1   FAStTunr(2) - Tower fore-aft modal stiffness tuner, 2nd mode (-)
           1   SSStTunr(1) - Tower side-to-side stiffness tuner, 1st mode (-)
           1   SSStTunr(2) - Tower side-to-side stiffness tuner, 2nd mode (-)
           1   AdjTwMa     - Factor to adjust tower mass density (-)
           1   AdjFASt     - Factor to adjust tower fore-aft stiffness (-)
           1   AdjSSSt     - Factor to adjust tower side-to-side stiffness (-)
---------------------- DISTRIBUTED TOWER PROPERTIES ----------------------------
  HtFract       TMassDen         TwFAStif       TwSSStif
   (-)           (kg/m)           (Nm^2)         (Nm^2)
0.0000000E+00  5.5908700E+03  6.1434300E+11  6.1434300E+11
1.0000000E-01  5.2324300E+03  5.3482100E+11  5.3482100E+11
2.0000000E-01  4.8857600E+03  4.6326700E+11  4.6326700E+11
3.0000000E-01  4.5508700E+03  3.9913100E+11  3.9913100E+11
4.0000000E-01  4.2277500E+03  3.4188300E+11  3.4188300E+11
5.0000000E-01  3.9164100E+03  2.9101100E+11  2.9101100E+11
6.0000000E-01  3.6168300E+03  2.4602700E+11  2.4602700E+11
7.0000000E-01  3.3290300E+03  2.0645700E+11  2.0645700E+11
8.0000000E-01  3.0530100E+03  1.7185100E+11  1.7185100E+11
9.0000000E-01  2.7887500E+03  1.4177600E+11  1.4177600E+11
1.0000000E+00  2.5362700E+03  1.1582000E+11  1.1582000E+11
---------------------- TOWER FORE-AFT MODE SHAPES ------------------------------
      0.7004   TwFAM1Sh(2) - Mode 1, coefficient of x^2 term
      2.1963   TwFAM1Sh(3) -       , coefficient of x^3 term
     -5.6202   TwFAM1Sh(4) -       , coefficient of x^4 term
      6.2275   TwFAM1Sh(5) -       , coefficient of x^5 term
```

Figure 17: The tower input file, defining key structural and material properties.

4.2.5 NRELOffshrBsline5MW_InflowWind_12mps.dat

This file is required to include a pre-generated wind field. If a turbulent wind field is used, additional pre-processing is necessary.

```
------- InflowWind INPUT FILE -------------------------------------------------------------------------
12 m/s turbulent winds on 31x31 FF grid and tower for FAST CertTests #18, #19, #21, #22, #23, and #24
---------------------------------------------------------------------------------------------------------
False         Echo           - Echo input data to <RootName>.ech (flag)
          3   WindType       - switch for wind file type (1=steady; 2=uniform; 3=binary TurbSim FF; 4=binary Bladed-style FF; 5=HAWC forma
          0   PropagationDir - Direction of wind propagation (meteorological rotation from aligned with X (positive rotates towards -Y) --
          0   VFlowAng       - Upflow angle (degrees) (not used for native Bladed format WindType=7)
False         VelInterpCubic - Use cubic interpolation for velocity in time (false=linear, true=cubic) [Used with WindType=2,3,4,5,7]
          1   NWindVel       - Number of points to output the wind velocity    (0 to 9)
          0   WindVxiList    - List of coordinates in the inertial X direction (m)
          0   WindVyiList    - List of coordinates in the inertial Y direction (m)
         90   WindVziList    - List of coordinates in the inertial Z direction (m)
```

Figure 18: In this simulation, a turbulent wind field is selected by setting `WindType = 3`. The turbulence must be pre-generated, which in this case was done using the internal TurbSim module. Additionally, the output location for the wind field is specified, along with the number of coordinate points in each direction.

```
================= Parameters for Binary TurbSim Full-Field files   [used only for WindType = 3] =============
"Wind/90m_12mps_twr.bts"    FileName_BTS   - Name of the Full field wind file to use (.bts)
```

Figure 19: The user must specify the location of the binary turbulent wind field file generated by TurbSim.

```
  -------------   Scaling parameters for turbulence   ---------------------------------------------------------
          2   ScaleMethod    - Turbulence scaling method   [0 = none, 1 = direct scaling, 2 = calculate scaling factor based on a desired
          1   SFx            - Turbulence scaling factor for the x direction (-)   [ScaleMethod=1]
          1   SFy            - Turbulence scaling factor for the y direction (-)   [ScaleMethod=1]
          1   SFz            - Turbulence scaling factor for the z direction (-)   [ScaleMethod=1]
        1.2   SigmaFx        - Turbulence standard deviation to calculate scaling from in x direction (m/s)    [ScaleMethod=2]
        0.8   SigmaFy        - Turbulence standard deviation to calculate scaling from in y direction (m/s)    [ScaleMethod=2]
        0.2   SigmaFz        - Turbulence standard deviation to calculate scaling from in z direction (m/s)    [ScaleMethod=2]
```

Figure 20: Scaling and parameters for turbulence section in the InflowWind input file. The turbulence can be scaled to a desired length in all three spatial directions. In this case, scaling is applied using a specified standard deviation.

```
  -------------   Mean wind profile parameters (added to HAWC-format files)   --------------------------------
         12   URef           - Mean u-component wind speed at the reference height (m/s)
          2   WindProfile    - Wind profile type (0=constant;1=logarithmic,2=power law)
        0.2   PLExp_Hawc     - Power law exponent (-) (used for PL wind profile type only)
       0.03   Z0             - Surface roughness length (m) (used for LG wind profile type only)
          0   XOffset        - Initial offset in +x direction (shift of wind box)
```

Figure 21: Mean wind profile parameters section in the InflowWind input file. Key wind parameters, such as mean wind speed and surface roughness, are defined in this section.

```
===================== OUTPUT ==============================================
False         SumPrint     - Print summary data to <RootName>.IfW.sum (flag)
              OutList      - The next line(s) contains a list of output parameters.  See OutListParameters.xlsx for a listing of available
"Wind1VelX"               X-direction wind velocity at point WindList(1)
"Wind1VelY"               Y-direction wind velocity at point WindList(1)
"Wind1VelZ"               Z-direction wind velocity at point WindList(1)
END of input file (the word "END" must appear in the first 3 columns of this last OutList line)
---------------------------------------------------------------------------------------
```

Figure 22: Output section in the InflowWind input file. The output consists of a three-dimensional velocity field at the specified location.

4.2.6 NRELOffshrBsline5MW_OC3Hywind_AeroDyn.dat

This file serves as the primary input for the AeroDyn module.

```
======  General Options  ============================================================================
False                  Echo        - Echo the input to "<rootname>.AD.ech"? (flag)
"default"              DTAero      - Time interval for aerodynamic calculations {or "default"} (s)
1                      Wake_Mod    - Wake/induction model (switch) {0=none, 1=BEMT, 3=OLAF} [Wake_Mod cannot be 2 or 3 when linearizing]
1                      TwrPotent   - Type tower influence on wind based on potential flow around the tower (switch) {0=none, 1=baseline po
0                      TwrShadow   - Calculate tower influence on wind based on downstream tower shadow (switch) {0=none, 1=Powles model,
True                   TwrAero     - Calculate tower aerodynamic loads? (flag)
False                  CavitCheck  - Perform cavitation check? (flag) [UA_Mod must be 0 when CavitCheck=true]
False                  Buoyancy    - Include buoyancy effects? (flag)
False                  NacelleDrag - Include Nacelle Drag effects? (flag)
False                  CompAA      - Flag to compute AeroAcoustics calculation [used only when Wake_Mod = 1 or 2]
"unused"               AA_InputFile - AeroAcoustics input file [used only when CompAA=true]
```

Figure 23: General options section in the AeroDyn input file. The user must specify whether to use a different time step for aerodynamic calculations or to retain the default interval from the main file. Additionally, wake modeling and the influence of the tower on the wind field can be configured here.

```
======  Environmental Conditions  ===================================================================
"default"              AirDens     - Air density (kg/m^3)
"default"              KinVisc     - Kinematic viscosity of working fluid (m^2/s)
"default"              SpdSound    - Speed of sound in working fluid (m/s)
"default"              Patm        - Atmospheric pressure (Pa) [used only when CavitCheck=True]
"default"              Pvap        - Vapour pressure of working fluid (Pa) [used only when CavitCheck=True]
```

Figure 24: Environmental conditions section in the AeroDyn input file. This section allows the user to define various environmental conditions. If `default` is selected, the environmental parameters from the main file will be used.

```
======  Airfoil Information =========================================================================
1                      AFTabMod    - Interpolation method for multiple airfoil tables {1=1D interpolation on AoA (first table only); 2=2D
1                      InCol_Alfa  - The column in the airfoil tables that contains the angle of attack (-)
2                      InCol_Cl    - The column in the airfoil tables that contains the lift coefficient (-)
3                      InCol_Cd    - The column in the airfoil tables that contains the drag coefficient (-)
4                      InCol_Cm    - The column in the airfoil tables that contains the pitching-moment coefficient; use zero if there is
0                      InCol_Cpmin - The column in the airfoil tables that contains the Cpmin coefficient; use zero if there is no Cpmin c
8                      NumAFfiles  - Number of airfoil files used (-)
"../5MW_Baseline/Airfoils/Cylinder1.dat" AFNames - Airfoil file names (NumAFfiles lines) (quoted strings)
"../5MW_Baseline/Airfoils/Cylinder2.dat"
"../5MW_Baseline/Airfoils/DU40_A17.dat"
"../5MW_Baseline/Airfoils/DU35_A17.dat"
"../5MW_Baseline/Airfoils/DU30_A17.dat"
"../5MW_Baseline/Airfoils/DU25_A17.dat"
"../5MW_Baseline/Airfoils/DU21_A17.dat"
"../5MW_Baseline/Airfoils/NACA64_A17.dat"
```

Figure 25: Airfoil information section in the AeroDyn input file. To compute aerodynamic forces, the user must define airfoil tables. In this case, pre-existing example files are used. The required airfoil data can be found in the `5MW_Baseline` folder.

```
======  Rotor/Blade Properties  =====================================================================
True                   UseBlCm     - Include aerodynamic pitching moment in calculations? (flag)
"../5MW_Baseline/NRELOffshrBsline5MW_AeroDyn_blade.dat" ADBlFile(1) - Name of file containing distributed aerodynamic properties for Blade
"../5MW_Baseline/NRELOffshrBsline5MW_AeroDyn_blade.dat" ADBlFile(2) - Name of file containing distributed aerodynamic properties for Blade
"../5MW_Baseline/NRELOffshrBsline5MW_AeroDyn_blade.dat" ADBlFile(3) - Name of file containing distributed aerodynamic properties for Blade
```

Figure 26: Rotor/Blade properties section in the AeroDyn input file. An additional input file is required to define the aerodynamic properties of the rotor blades. In this case, the file is sourced from `5MW_Baseline`.

```
======  Tower Influence and Aerodynamics ============================================================ [used only when TwrPotent/=0, TwrSha
11                    NumTwrNds   - Number of tower nodes used in the analysis (-) [used only when TwrPotent/=0, TwrShadow/=0, TwrAero=Tr
TwrElev        TwrDiam        TwrCd          TwrTI          TwrCb         ! TwrTI used only when TwrShadow=2; TwrCb used only when Buoyanc
(m)            (m)            (-)            (-)            (-)
1.0000000E+01  6.5000000E+00  1.0000000E+00  1.0000000E-01  0.0000000E+00
1.7760000E+01  6.2400000E+00  1.0000000E+00  1.0000000E-01  0.0000000E+00
2.5520000E+01  5.9700000E+00  1.0000000E+00  1.0000000E-01  0.0000000E+00
3.3280000E+01  5.7100000E+00  1.0000000E+00  1.0000000E-01  0.0000000E+00
4.1040000E+01  5.4500000E+00  1.0000000E+00  1.0000000E-01  0.0000000E+00
4.8800000E+01  5.1800000E+00  1.0000000E+00  1.0000000E-01  0.0000000E+00
5.6560000E+01  4.9200000E+00  1.0000000E+00  1.0000000E-01  0.0000000E+00
6.4320000E+01  4.6600000E+00  1.0000000E+00  1.0000000E-01  0.0000000E+00
7.2080000E+01  4.4000000E+00  1.0000000E+00  1.0000000E-01  0.0000000E+00
7.9840000E+01  4.1300000E+00  1.0000000E+00  1.0000000E-01  0.0000000E+00
8.7600000E+01  3.8700000E+00  1.0000000E+00  1.0000000E-01  0.0000000E+00
```

Figure 27: In this simulation, the tower's influence on the wind field is considered, as indicated by `TwrPotent = 1`. AeroDyn requires the tower's dimensions to accurately calculate its aerodynamic effects based on the selected model.

4.2.7 `NRELOffshrBsline5MW_OC3Hywind_ServoDyn.dat`

This is the input file for the ServoDyn module, which enables the wind turbine's control system. Additionally, ServoDyn models other electrical components, such as the generator.

```
------- SERVODYN INPUT FILE --------------------------------------------
NREL 5.0 MW Baseline Wind Turbine for Use in Offshore Analysis. Properties from Dutch Offshore Wind Energy Converter (DOWEC) 6MW Pre-Design (10046_009.pdf) and REpower 5
---------------------- SIMULATION CONTROL --------------------------------------
False         Echo         - Echo input data to <RootName>.ech (flag)
"default"     DT           - Communication interval for controllers (s) (or "default")
---------------------- PITCH CONTROL -------------------------------------------
          5   PCMode       - Pitch control mode {0: none, 3: user-defined from routine PitchCntrl, 4: user-defined from Simulink/Labview, 5: user-defined from Bladed-sty
          0   TPCOn        - Time to enable active pitch control (s) [unused when PCMode=0]
     9999.9   TPitManS(1)  - Time to start override pitch maneuver for blade 1 and end standard pitch control (s)
     9999.9   TPitManS(2)  - Time to start override pitch maneuver for blade 2 and end standard pitch control (s)
     9999.9   TPitManS(3)  - Time to start override pitch maneuver for blade 3 and end standard pitch control (s) [unused for 2 blades]
          2   PitManRat(1) - Pitch rate at which override pitch maneuver heads toward final pitch angle for blade 1 (deg/s)
          2   PitManRat(2) - Pitch rate at which override pitch maneuver heads toward final pitch angle for blade 2 (deg/s)
          2   PitManRat(3) - Pitch rate at which override pitch maneuver heads toward final pitch angle for blade 3 (deg/s) [unused for 2 blades]
          0   BlPitchF(1)  - Blade 1 final pitch for pitch maneuvers (degrees)
          0   BlPitchF(2)  - Blade 2 final pitch for pitch maneuvers (degrees)
          0   BlPitchF(3)  - Blade 3 final pitch for pitch maneuvers (degrees) [unused for 2 blades]
---------------------- GENERATOR AND TORQUE CONTROL ----------------------------
          5   VSContrl     - Variable-speed control mode {0: none, 1: simple VS, 3: user-defined from routine UserVSCont, 4: user-defined from Simulink/Labview, 5: user-
          2   GenModel     - Generator model {1: simple, 2: Thevenin, 3: user-defined from routine UserGen} (switch) [used only when VSContrl=0]
       94.4   GenEff       - Generator efficiency [ignored by the Thevenin and user-defined generator models] (%)
True          GenTiStr     - Method to start the generator {T: timed using TimGenOn, F: generator speed using SpdGenOn} (flag)
True          GenTiStp     - Method to stop the generator {T: timed using TimGenOf, F: when generator power = 0} (flag)
     9999.9   SpdGenOn     - Generator speed to turn on the generator for a startup (HSS speed) (rpm) [used only when GenTiStr=False]
          0   TimGenOn     - Time to turn on the generator for a startup (s) [used only when GenTiStr=True]
     9999.9   TimGenOf     - Time to turn off the generator (s) [used only when GenTiStp=True]
---------------------- SIMPLE VARIABLE-SPEED TORQUE CONTROL --------------------
     9999.9   VS_RtGnSp    - Rated generator speed for simple variable-speed generator control (HSS side) (rpm) [used only when VSContrl=1]
     9999.9   VS_RtTq      - Rated generator torque/constant generator torque in Region 3 for simple variable-speed generator control (HSS side) (N-m) [used only when VS
     9999.9   VS_Rgn2K     - Generator torque constant in Region 2 for simple variable-speed generator control (HSS side) (N-m/rpm^2) [used only when VSContrl=1]
     9999.9   VS_SlPc      - Rated generator slip percentage in Region 2 1/2 for simple variable-speed generator control (%) [used only when VSContrl=1]
```

Figure 28: The control mode is selected in this section. In this case, the pre-compiled DISCON controller is used.

```
---------------------- SIMPLE INDUCTION GENERATOR ------------------------------
     9999.9   SIG_SlPc     - Rated generator slip percentage (%) [used only when VSContrl=0 and GenModel=1]
     9999.9   SIG_SySp     - Synchronous (zero-torque) generator speed (rpm) [used only when VSContrl=0 and GenModel=1]
     9999.9   SIG_RtTq     - Rated torque (N-m) [used only when VSContrl=0 and GenModel=1]
     9999.9   SIG_PORt     - Pull-out ratio (Tpullout/Trated) (-) [used only when VSContrl=0 and GenModel=1]
---------------------- THEVENIN-EQUIVALENT INDUCTION GENERATOR -----------------
     9999.9   TEC_Freq     - Line frequency [50 or 60] (Hz) [used only when VSContrl=0 and GenModel=2]
       9998   TEC_NPol     - Number of poles [even integer > 0] (-) [used only when VSContrl=0 and GenModel=2]
     9999.9   TEC_SRes     - Stator resistance (ohms) [used only when VSContrl=0 and GenModel=2]
     9999.9   TEC_RRes     - Rotor resistance (ohms) [used only when VSContrl=0 and GenModel=2]
     9999.9   TEC_VLL      - Line-to-line RMS voltage (volts) [used only when VSContrl=0 and GenModel=2]
     9999.9   TEC_SLR      - Stator leakage reactance (ohms) [used only when VSContrl=0 and GenModel=2]
     9999.9   TEC_RLR      - Rotor leakage reactance (ohms) [used only when VSContrl=0 and GenModel=2]
     9999.9   TEC_MR       - Magnetizing reactance (ohms) [used only when VSContrl=0 and GenModel=2]
```

Figure 29: This section allows the user to specify generator type and configuration settings.

```
--------------------- HIGH-SPEED SHAFT BRAKE --------------------------------
         0   HSSBrMode    - HSS brake model {0: none, 1: simple, 3: user-defined from routine UserHSSBr, 4: user-defined from Simulink/Labview, 5: user-defined from Bl
    9999.9   THSSBrDp     - Time to initiate deployment of the HSS brake (s)
       0.6   HSSBrDT      - Time for HSS-brake to reach full deployment once initiated (sec) [used only when HSSBrMode=1]
   28116.2   HSSBrTqF     - Fully deployed HSS-brake torque (N-m)
```

Figure 30: A brake system can be defined here to activate when cut-off wind speeds are reached.

```
--------------------- NACELLE-YAW CONTROL ------------------------------------
         0   YCMode       - Yaw control mode {0: none, 3: user-defined from routine UserYawCont, 4: user-defined from Simulink/Labview, 5: user-defined from Bladed-sty
    9999.9   TYCOn        - Time to enable active yaw control (s) [unused when YCMode=0]
         0   YawNeut      - Neutral yaw position--yaw spring force is zero at this yaw (degrees)
9.02832E+09   YawSpr      - Nacelle-yaw spring constant (N-m/rad)
  19160000   YawDamp      - Nacelle-yaw damping constant (N-m/(rad/s))
    9999.9   TYawManS     - Time to start override yaw maneuver and end standard yaw control (s)
         2   YawManRat    - Yaw maneuver rate (in absolute value) (deg/s)
         0   NacYawF      - Final yaw angle for override yaw maneuvers (degrees)
--------------------- AERODYNAMIC FLOW CONTROL --------------------------------
         0   AfCmode      - Airfoil control mode {0: none, 1: cosine wave cycle, 4: user-defined from Simulink/Labview, 5: user-defined from Bladed-style DLL} (switch)
         0   AfC_Mean     - Mean level for cosine cycling or steady value (-) [used only with AfCmode==1]
         0   AfC_Amp      - Amplitude for for cosine cycling of flap signal (-) [used only with AfCmode==1]
         0   AfC_Phase    - Phase relative to the blade azimuth (0 is vertical) for for cosine cycling of flap signal (deg) [used only with AfCmode==1]
--------------------- STRUCTURAL CONTROL -------------------------------------
         0   NumBStC      - Number of blade structural controllers (integer)
"unused"     BStCfiles    - Name of the files for blade structural controllers (quoted strings) [unused when NumBStC==0]
         0   NumNStC      - Number of nacelle structural controllers (integer)
"unused"     NStCfiles    - Name of the files for nacelle structural controllers (quoted strings) [unused when NumNStC==0]
         0   NumTStC      - Number of tower structural controllers (integer)
"unused"     TStCfiles    - Name of the files for tower structural controllers (quoted strings) [unused when NumTStC==0]
         0   NumSStC      - Number of substructure structural controllers (integer)
"unused"     SStCfiles    - Name of the files for substructure structural controllers (quoted strings) [unused when NumSStC==0]
--------------------- CABLE CONTROL ------------------------------------------
         0   CCmode       - Cable control mode {0: none, 4: user-defined from Simulink/Labview, 5: user-defined from Bladed-style DLL} (switch)
```

Figure 31: The user can define a complex control system directly within the input file. This section includes options for nacelle-yaw control.

```
--------------------- BLADED INTERFACE ---------------------------------------- [used only with Bladed Interface]
"../5MW_Baseline/ServoData/DISCON_OC3Hywind.dll"    DLL_FileName - Name/location of the dynamic library {.dll [Windows] or .so [Linux]} in the Bladed-DLL format (-) [us
"DISCON.IN"    DLL_InFile   - Name of input file sent to the DLL (-) [used only with Bladed Interface]
"DISCON"     DLL_ProcName - Name of procedure in DLL to be called (-) [case sensitive; used only with DLL Interface]
"default"    DLL_DT       - Communication interval for dynamic library (s) (or "default") [used only with Bladed Interface]
false        DLL_Ramp     - Whether a linear ramp should be used between DLL_DT time steps [introduces time shift when true] (flag) [used only with Bladed Interface]
    9999.9   BPCutoff     - Cutoff frequency for low-pass filter on blade pitch from DLL (Hz) [used only with Bladed Interface]
         0   NacYaw_North - Reference yaw angle of the nacelle when the upwind end points due North (deg) [used only with Bladed Interface]
         0   Ptch_Cntrl   - Record 28: Use individual pitch control {0: collective pitch; 1: individual pitch control} (switch) [used only with Bladed Interface]
         0   Ptch_SetPnt  - Record  5: Below-rated pitch angle set-point (deg) [used only with Bladed Interface]
         0   Ptch_Min     - Record  6: Minimum pitch angle (deg) [used only with Bladed Interface]
         0   Ptch_Max     - Record  7: Maximum pitch angle (deg) [used only with Bladed Interface]
         0   PtchRate_Min - Record  8: Minimum pitch rate (most negative value allowed) (deg/s) [used only with Bladed Interface]
         0   PtchRate_Max - Record  9: Maximum pitch rate  (deg/s) [used only with Bladed Interface]
         0   Gain_OM      - Record 16: Optimal mode gain (Nm/(rad/s)^2) [used only with Bladed Interface]
         0   GenSpd_MinOM - Record 17: Minimum generator speed (rpm) [used only with Bladed Interface]
         0   GenSpd_MaxOM - Record 18: Optimal mode maximum speed (rpm) [used only with Bladed Interface]
         0   GenSpd_Dem   - Record 19: Demanded generator speed above rated (rpm) [used only with Bladed Interface]
         0   GenTrq_Dem   - Record 22: Demanded generator torque above rated (Nm) [used only with Bladed Interface]
         0   GenPwr_Dem   - Record 13: Demanded power (W) [used only with Bladed Interface]
--------------------- BLADED INTERFACE TORQUE-SPEED LOOK-UP TABLE -------------
         0   DLL_NumTrq   - Record 26: No. of points in torque-speed look-up table {0 = none and use the optimal mode parameters; nonzero = ignore the optimal mode PAR
 GenSpd_TLU   GenTrq_TLU
 (rpm)          (Nm)
```

Figure 32: This section specifies the DLL file for integrating the DISCON controller, which is used in this case.

4.2.8 SeaState.dat

```
--------------------- SPATIAL DISCRETIZATION ---------------------------------------------------
          30   X_HalfWidth    - Half-width of the domain in the X direction (m) [>0, NOTE: X[nX] = nX*dX, where nX = {-NX+1,-NX+2,…,NX-1} and dX = X_HalfWidth/(NX-1)]
          30   Y_HalfWidth    - Half-width of the domain in the Y direction (m) [>0, NOTE: Y[nY] = nY*dY, where nY = {-NY+1,-NY+2,…,NY-1} and dY = Y_HalfWidth/(NY-1)]
         130   Z_Depth        - Depth of the domain the Z direction (m) relative to SWL [0 < Z_Depth <= WtrDpth+MSL2SWL; "default": Z_Depth = WtrDpth+MSL2SWL; Z[nZ] =
           3   NX             - Number of nodes in half of the X-direction domain (-) [>=2]
           3   NY             - Number of nodes in half of the Y-direction domain (-) [>=2]
          40   NZ             - Number of nodes in the Z direction (-) [>=2]
```

Figure 33: Spatial discretization section in the SeaState module. Definition of the wave field mesh, specifying the spatial discretization of the computational domain.

```
---------------------- WAVES --------------------------------------------------
          2   WaveMod       - Incident wave kinematics model {0: none=still water, 1: regular (periodic), 1P#: regular with user-specified phase, 2: JONSWAP/Pierson-
          0   WaveStMod     - Model for stretching incident wave kinematics to instantaneous free surface {0: none=no stretching, 1: vertical stretching, 2: extrapol
        600   WaveTMax      - Analysis time for incident wave calculations (sec) [unused when WaveMod=0; determines WaveDOmega=2Pi/WaveTMax in the IFFT]
       0.25   WaveDT        - Time step for incident wave calculations     (sec) [unused when WaveMod=0 or 7; 0.1<=WaveDT<=1.0 recommended; determines WaveOmegaMax=P
          6   WaveHs        - Significant wave height of incident waves (meters) [used only when WaveMod=1, 2, or 3]
         10   WaveTp        - Peak-spectral period of incident waves       (sec) [used only when WaveMod=1 or 2]
"DEFAULT"     WavePkShp     - Peak-shape parameter of incident wave spectrum (-) or DEFAULT (string) [used only when WaveMod=2; use 1.0 for Pierson-Moskowitz]
          0   WvLowCOff     - Low  cut-off frequency or lower frequency limit of the wave spectrum beyond which the wave spectrum is zeroed (rad/s) [unused when Wave
        500   WvHiCOff      - High cut-off frequency or upper frequency limit of the wave spectrum beyond which the wave spectrum is zeroed (rad/s) [unused when Wave
          0   WaveDir       - Incident wave propagation heading direction                        (degrees) [unused when WaveMod=0, 6 or 7]
          0   WaveDirMod    - Directional spreading function {0: none, 1: COS2S}                 (-)       [only used when WaveMod=2,3, or 4]
          1   WaveDirSpread - Wave direction spreading coefficient ( > 0 )                       (-)       [only used when WaveMod=2,3, or 4 and WaveDirMod=1]
          1   WaveNDir      - Number of wave directions                                          (-)       [only used when WaveMod=2,3, or 4 and WaveDirMod=1; odd n
         90   WaveDirRange  - Range of wave directions (full range: WaveDir +/- 1/2*WaveDirRange) (degrees) [only used when WaveMod=2,3,or 4 and WaveDirMod=1]
  123456789   WaveSeed(1)   - First  random seed of incident waves [-2147483648 to 2147483647]    (-)       [unused when WaveMod=0, 5, or 6]
"RANLUX"      WaveSeed(2)   - Second random seed of incident waves [-2147483648 to 2147483647] for intrinsic pRNG, or an alternative pRNG: "RanLux" (-)       [unused
TRUE          WaveNDAmp     - Flag for normally distributed amplitudes                           (flag)    [only used when WaveMod=2, 3, or 4]
""            WvKinFile     - Root name of externally generated wave data file(s)        (quoted string)    [used only when WaveMod=5, 6 or 7]
```

Figure 34: Waves section in the SeaState module. Selection of general wave settings, including wave type and spectral characteristics.

```
---------------------- 2ND-ORDER WAVES ----------------------------------------- [unused with WaveMod=0 or 6]
FALSE            WvDiffQTF     - Full difference-frequency 2nd-order wave kinematics (flag)
FALSE            WvSumQTF      - Full summation-frequency  2nd-order wave kinematics (flag)
             0   WvLowCOffD    - Low  frequency cutoff used in the difference-frequencies (rad/s) [Only used with a difference-frequency method]
           3.5   WvHiCOffD     - High frequency cutoff used in the difference-frequencies (rad/s) [Only used with a difference-frequency method]
           0.1   WvLowCOffS    - Low  frequency cutoff used in the summation-frequencies  (rad/s) [Only used with a summation-frequency  method]
           3.5   WvHiCOffS     - High frequency cutoff used in the summation-frequencies  (rad/s) [Only used with a summation-frequency  method]
---------------------- CONSTRAINED WAVES ---------------------------------------
             0   ConstWaveMod  - Constrained wave model: 0=none; 1=Constrained wave with specified crest elevation, alpha; 2=Constrained wave with guaranteed peak-to-tr
             1   CrestHmax     - Crest height (2*alpha for ConstWaveMod=1 or HCrest for ConstWaveMod=2), must be larger than WaveHs (m) [unused when ConstWaveMod=0]
            60   CrestTime     - Time at which the crest appears (s) [unused when ConstWaveMod=0]
             0   CrestXi       - X-position of the crest (m) [unused when ConstWaveMod=0]
             0   CrestYi       - Y-position of the crest (m) [unused when ConstWaveMod=0]
---------------------- CURRENT ------------------------------------------------- [unused with WaveMod=6]
             0   CurrMod       - Current profile model {0: none=no current, 1: standard, 2: user-defined from routine UserCurrent} (switch)
             0   CurrSSV0      - Sub-surface current velocity at still water level  (m/s) [used only when CurrMod=1]
"DEFAULT"        CurrSSDir     - Sub-surface current heading direction (degrees) or DEFAULT (string) [used only when CurrMod=1]
            20   CurrNSRef     - Near-surface current reference depth            (meters) [used only when CurrMod=1]
             0   CurrNSV0      - Near-surface current velocity at still water level (m/s) [used only when CurrMod=1]
             0   CurrNSDir     - Near-surface current heading direction         (degrees) [used only when CurrMod=1]
             0   CurrDIV       - Depth-independent current velocity                 (m/s) [used only when CurrMod=1]
             0   CurrDIDir     - Depth-independent current heading direction    (degrees) [used only when CurrMod=1]
---------------------- MacCamy-Fuchs diffraction model -------------------------
             0   MCFD          - MacCamy-Fuchs member radius (ignored if radius <= 0) [must be 0 when WaveMod 0 or 6]
```

Figure 35: Second order Waves, constrained waves and current sections in the SeaState module. Additional options for including second-order wave effects and ocean currents.

4.2.9 `NRELOffshrBsline5MW_OC3Hywind_HydroDyn.dat`

This file serves as the input for the HydroDyn module.

```
---------------------- FLOATING PLATFORM --------------------------------------- [unused with WaveMod=6]
             1   PotMod        - Potential-flow model {0: none=no potential flow, 1: frequency-to-time-domain transforms based on WAMIT output, 2: fluid-impulse theory
             1   ExctnMod      - Wave-excitation model {0: no wave-excitation calculation, 1: DFT, 2: state-space} (switch) [only used when PotMod=1; STATE-SPACE REQUIR
             0   ExctnDisp     - Method of computing Wave Excitation {0: use undisplaced position, 1: use displaced position, 2: use low-pass filtered displaced positio
            10   ExctnCutOff   - Cutoff (corner) frequency of the low-pass time-filtered displaced position (Hz) [>0.0] [used only when PotMod=1, ExctnMod>0, and ExctnD
             0   PtfmYMod      - Model for large platform yaw offset {0: Static reference yaw offset based on PtfmRefY, 1: dynamic reference yaw offset based on low-pas
             0   PtfmRefY      - Constant (if PtfmYMod=0) or initial (if PtfmYMod=1) platform reference yaw offset (deg)
          0.01   PtfmYCutOff   - Cutoff frequency for the low-pass filtering of PRP yaw motion when PtfmYMod=1 [unused when PtfmYMod=0] (Hz)
            36   NExctnHdg     - Number of evenly distributed platform yaw/heading angles over the range of [-180, 180) deg for which the wave excitation shall be compu
             1   RdtnMod       - Radiation memory-effect model {0: no memory-effect calculation, 1: convolution, 2: state-space} (switch) [only used when PotMod=1; STAT
            60   RdtnTMax      - Analysis time for wave radiation kernel calculations (sec) [only used when PotMod=1 and RdtnMod>0; determines RdtnDOmega=Pi/RdtnTMax in
        0.0125   RdtnDT        - Time step for wave radiation kernel calculations (sec) [only used when PotMod=1 and ExctnMod>0 or RdtnMod>0; DT<=RdtnDT<=0.1 recommende
             1   NBody         - Number of WAMIT bodies to be used (-) [>=1; only used when PotMod=1. If NBodyMod=1, the WAMIT data contains a vector of size 6*NBody x
             1   NBodyMod      - Body coupling model {1: include coupling terms between each body and NBody in HydroDyn equals NBODY in WAMIT, 2: neglect coupling terms
"../5MW_Baseline/HydroData/Spar"    PotFile       - Root name of potential-flow model data; WAMIT output files containing the linear, nondimensionalized, hydrostatic res
             1   WAMITULEN     - Characteristic body length scale used to redimensionalize WAMIT output (meters) [1 to NBody if NBodyMod>1] [only used when PotMod=1]
             0   PtfmRefxt     - The xt offset of the body reference point(s) from (0,0,0) (meters) [1 to NBody] [only used when PotMod=1]
             0   PtfmRefyt     - The yt offset of the body reference point(s) from (0,0,0) (meters) [1 to NBody] [only used when PotMod=1]
             0   PtfmRefzt     - The zt offset of the body reference point(s) from (0,0,0) (meters) [1 to NBody] [only used when PotMod=1. If NBodyMod=2,PtfmRefzt=0.0]
             0   PtfmRefztRot  - The rotation about zt of the body reference frame(s) from xt/yt (degrees) [1 to NBody] [only used when PotMod=1]
       8029.21   PtfmVol0      - Displaced volume of water when the body is in its undisplaced position (m^3) [1 to NBody] [only used when PotMod=1; USE THE SAME VALUE
             0   PtfmCOBxt     - The xt offset of the center of buoyancy (COB) from (0,0) (meters) [1 to NBody] [only used when PotMod=1]
             0   PtfmCOByt     - The yt offset of the center of buoyancy (COB) from (0,0) (meters) [1 to NBody] [only used when PotMod=1]
```

Figure 36: Floating platform section in the HydroDyn module. The user must choose between the potential flow model and strip theory for calculating hydrodynamic loads.

```
--------------------- 2ND-ORDER FLOATING PLATFORM FORCES --------------------- [unused with WaveMod=0 or 6, or PotMod=0 or 2]
0   MnDrift        - Mean-drift 2nd-order forces computed                                       {0: None; [7, 8, 9, 10, 11, or 12]: WAMIT file to use} [Only
0   NewmanApp      - Mean- and slow-drift 2nd-order forces computed with Newman's approximation {0: None; [7, 8, 9, 10, 11, or 12]: WAMIT file to use} [Only
0   DiffQTF        - Full difference-frequency 2nd-order forces computed with full QTF          {0: None; [10, 11, or 12]: WAMIT file to use}          [Only
0   SumQTF         - Full summation -frequency 2nd-order forces computed with full QTF          {0: None; [10, 11, or 12]: WAMIT file to use}
```

Figure 37: Second-order forces section in the HydroDyn module. Configuration of second-order hydrodynamic forces, including difference- and sum-frequency effects.

```
--------------------- PLATFORM ADDITIONAL STIFFNESS AND DAMPING  -------------- [unused with PotMod=0 or 2]
0   AddF0    - Additional preload (N, N-m)  [If NBodyMod=1, one size 6*NBody x 1 vector; if NBodyMod>1, NBody size 6 x 1 vectors]
0
0
0
0
0
0         0         0         0         0         0   AddCLin  - Additional linear stiffness (N/m, N/rad, N-m/m, N-m/rad)  [If NBodyMod=
0         0         0         0         0         0
0         0         0         0         0         0
0         0         0         0         0         0
0         0         0         0         0         0
0         0         0         0         0  98340000
100000    0         0         0         0         0   AddBLin  - Additional linear damping(N/(m/s), N/(rad/s), N-m/(m/s), N-m/(rad/s))
0    100000         0         0         0         0
0         0    130000         0         0         0
0         0         0         0         0         0
0         0         0         0         0         0
0         0         0         0         0  13000000
0         0         0         0         0         0   AddBQuad - Additional quadratic drag(N/(m/s)^2, N/(rad/s)^2, N-m(m/s)^2, N-m/(rad/
0         0         0         0         0         0
0         0         0         0         0         0
0         0         0         0         0         0
0         0         0         0         0         0
0         0         0         0         0         0
```

Figure 38: Additional stiffness and damping parameters can be specified to refine the system's dynamic response in the spring-mass equation.

```
--------------------- STRIP THEORY OPTIONS --------------------------------------
0   WaveDisp       - Method of computing Wave Kinematics {0: use undisplaced position, 1: use displaced position) } (switch)
0   AMMod          - Method of computing distributed added-mass force. (0: Only and always on nodes below SWL at the undisplaced position. 2: Up to the inst
```

Figure 39: Additional settings for strip theory can be configured here if this method is selected for hydrodynamic load calculations.

```
--------------------- SIMPLE HYDRODYNAMIC COEFFICIENTS (model 1) --------------
   SimplCd    SimplCdMG    SimplCa    SimplCaMG    SimplCp    SimplCpMG    SimplAxCd  SimplAxCdMG   SimplAxCa  SimplAxCaMG  SimplAxCp   SimplAxCpMG    SimplCb    SimplC
     (-)         (-)         (-)         (-)         (-)         (-)         (-)         (-)         (-)         (-)         (-)         (-)         (-)         (-)
    0.60        0.00        0.00        0.00        1.00        1.00        0.00        0.00        1.00        1.00        1.00        1.00        1.00        1.0
--------------------- DEPTH-BASED HYDRODYNAMIC COEFFICIENTS (model 2) ---------
0   NCoefDpth       - Number of depth-dependent coefficients (-)
Dpth      DpthCd    DpthCdMG   DpthCa    DpthCaMG      DpthCp    DpthCpMG   DpthAxCd    DpthAxCdMG    DpthAxCa    DpthAxCaMG    DpthAxCp    DpthAxCpMG   DpthCb    DpthCbMG
(m)       (-)       (-)        (-)       (-)           (-)       (-)        (-)         (-)           (-)         (-)           (-)         (-)          (-)       (-)
--------------------- MEMBER-BASED HYDRODYNAMIC COEFFICIENTS (model 3) --------
0   NCoefMembers       - Number of member-based coefficients (-)
MemberID    MemberCd1     MemberCd2     MemberCdMG1   MemberCdMG2   MemberCa1     MemberCa2     MemberCaMG1   MemberCaMG2   MemberCp1     MemberCp2     MemberCpMG1   Memb
  (-)         (-)           (-)           (-)           (-)           (-)           (-)           (-)           (-)           (-)           (-)           (-)
```

Figure 40: User-defined hydrodynamic coefficients can be specified to customize the complexity of the model.

```
-------------------- OUTPUT CHANNELS --------------------------------------
HydroFxi
HydroFyi
HydroFzi
HydroMxi
HydroMyi
HydroMzi
B1Surge
B1Sway
B1Heave
B1Roll
B1Pitch
B1Yaw
B1TVxi
B1TVyi
B1TVzi
B1RVxi
B1RVyi
B1RVzi
B1TAxi
B1TAyi
B1TAzi
B1RAxi
B1RAyi
B1RAzi
B1WvsFxi
B1WvsFyi
B1WvsFzi
B1WvsMxi
B1WvsMyi
B1WvsMzi
B1HDSFxi
B1HDSFyi
B1HDSFzi
B1HDSMxi
B1HDSMyi
B1HDSMzi
B1RdtFxi
B1RdtFyi
B1RdtFzi
B1RdtMxi
B1RdtMyi
B1RdtMzi
```

Figure 41: Overview of the available HydroDyn output channels, illustrating the range of hydrodynamic computations that can be performed.

4.2.10 NRELOffshrBsline5MW_OC3Hywind_MAP.dat

```
---------------------- LINE DICTIONARY ---------------------------------------
LineType     Diam     MassDenInAir    EA          CB     CIntDamp   Ca   Cdn    Cdt
(-)          (m)      (kg/m)          (N)         (-)     (Pa-s)    (-)  (-)    (-)
Material     0.09      77.7066       384.243E6   0.001   1.0E8     0.6  -1.0   0.05
---------------------- NODE PROPERTIES ---------------------------------------
Node Type       X     Y     Z      M     B      FX    FY    FZ
(-)  (-)       (m)   (m)   (m)   (kg)  (m^3)   (N)   (N)   (N)
1    fix     853.87  0    depth   0     0      #     #     #
2    Vessel    5.2   0    -70.0   0     0      #     #     #
---------------------- LINE PROPERTIES ---------------------------------------
Line    LineType  UnstrLen    NodeAnch  NodeFair  Flags
(-)      (-)       (m)         (-)       (-)       (-)
1       Material   902.2        1         2        tension_fair tension_anch
---------------------- SOLVER OPTIONS-----------------------------------------
Option
(-)
repeat 240 120
```

Figure 42: The MAP file defines the position, dimensions, and material properties of the mooring lines.

4.3 Standalone Module Drivers

Besides the main OpenFAST executable, standalone versions of the modules are also available as executable programs in the build directory. The execution process of a standalone driver is similar to that of the OpenFAST glue code. However, since OpenFAST itself is not running, a `.fst` file is not required. Instead, a driver file is needed, typically named `modulename_driver.inp`. This driver file defines essential parameters, such as the time step and total simulation time, which would normally be controlled by the main OpenFAST input file. In addition to the driver input file, the module still requires its standard input file(s).

Once all necessary files are configured according to the user's requirements, the standalone simulation can be executed with the following command:

```
./drivername modulename_driver.inp
```

It is recommended to create a dedicated simulation folder containing the executable driver and all required input files.

4.4 External Wave Kinematics

SeaState allows the user to incorporate externally generated wave fields, expanding the range of wave modeling beyond the inherent limitations of OpenFAST's internal methods.
To enable externally generated wave kinematics, the user must set `WaveMod = 6`. In this mode, SeaState requires eight text-based input files, all formatted without headers. These files provide wave velocity, acceleration, and dynamic pressure data in the global inertial-frame coordinate system.
The first seven files contain the velocity components (`.Vxi`, `.Vyi`, `.Vzi`), acceleration components (`.Axi`, `.Ayi`, `.Azi`), and dynamic pressure time series (`.DynP`). Each of these files must start with exactly 13 header lines, which SeaState will ignore. The remaining data consists of `WaveTMax/WaveDT` rows and `N` columns, where `N` corresponds to the total number of wave grid points defined in the SeaState input file. The spatial ordering follows a structured pattern: nodes are arranged first by increasing X-position, then by Y-position, and finally by Z-position. The first node is located at `(-X_HalfWidth,-Y_HalfWidth,MSL2SWL-Z_Dpth)`. Time values are not explicitly included in the files but are assumed to progress from zero to `WaveTMax` in steps of `WaveDT`.
The eighth file, `.Elev`, contains the wave elevation time series. It follows the same row count (`WaveTMax/WaveDT`) but differs in its spatial arrangement: it includes only horizontal-plane grid nodes, structured by increasing X-position first, followed by Y-position. The first node in this file is positioned at `(-X_HalfWidth,-Y_HalfWidth)`.
To properly use externally generated wave kinematics, the user must generate complete wave data at every time step and for all grid points, as SeaState does not perform interpolation. Numerical values, including zeros, are treated as valid data points, while any non-numeric entries are converted to zero. Importantly, the data is neither post-processed (e.g., filtered) nor checked for physical plausibility.
A key restriction of `WaveMod = 6` is that it cannot be used alongside the potential-flow model. Furthermore, only vertical and Wheeler stretching methods are supported, whereas extrapolation stretching is not available.
To facilitate the generation of compatible input files for `WaveMod = 6`, users can employ the standalone SeaState driver program. By running an internal wave-generation model (e.g., `WaveMod = 2`) with `WrWvKinMod = 2` in the driver input, a valid set of template input files can be automatically generated.

5 Post-Processing

5.1 Using ParaView for Visualization

ParaView can read and visualize `.vtk` files. To enable the output of `.vtk` files, the user must navigate to the visualization section of the `Main.fst` file (see Fig. 7). Various visualization options are available, with the most commonly used being the "surface" files. These allow for rendering the surfaces of the turbine components and the wave field, enabling the creation of three-dimensional animations of the simulation. Figure 43 shows a screenshot from the ParaView animation of the simulation prepared in Section 4.2.

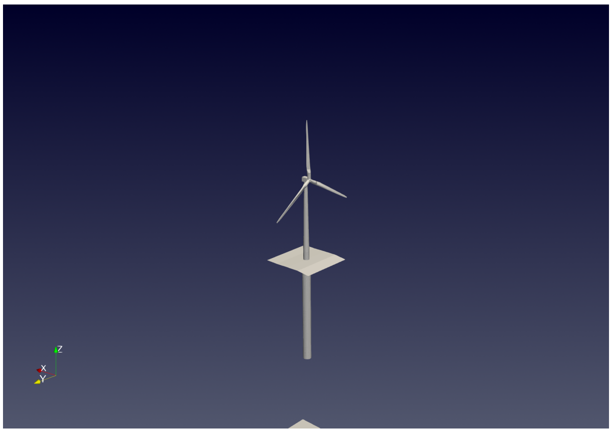

Figure 43: Screenshot from 5MW OC3 Spar FOWT with a turbulent wind field and irregular waves

It is possible to enable a filter on the wave field to visualize the surface elevation. A visualization of that can be seen in Figure 44.

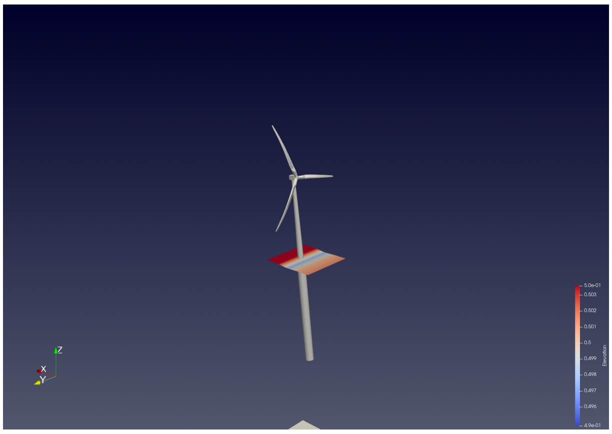

Figure 44: Screenshot from 5MW OC3 Spar FOWT with a turbulent wind field and irregular waves with surface elevation filter

5.2 Script for Output Transformation

OpenFAST often generates binary output files. This section provides a MATLAB script that I personally wrote to convert these binary files into `.csv` format for further processing. To enable MATLAB to read OpenFAST binary files using `ReadFASTbinary()`, the OpenFAST MATLAB toolbox is required.

```
% Read file
outbfile = '/Users/home/OpenFAST/path/to/your/file.outb';
[Channels, ChanName, ChanUnit, FileID, DescStr] = ReadFASTbinary(outbfile);
time = Channels(:,1);

% Create a cell array to store all the data
%       First row: ChanName
%       Second row: ChanUnit
%       Next xxxx rows: Channels data
outputData = [ChanName'; ChanUnit'; num2cell(Channels)];
```

```
% Convert the cell array into a table
outputTable = cell2table(outputData);

% Define the filename and location for the CSV file
csvFileName = '/Users/home/path/to/outputs/filename.csv';

% Write the table to a CSV file
writetable(outputTable, csvFileName, 'WriteVariableNames', false);
```

References

[1] Tony Burton, Nick Jenkins, David Sharpe, and Ervin Bossanyi. Wind energy handbook. Technical Report NREL/TP-500-38060, National Renewable Energy Laboratory (NREL), 2009. Accessed: 2024-02-09.

[2] National Renewable Energy Laboratory. Nrel forum, 2024. Accessed: 2024-02-09.

[3] National Renewable Energy Laboratory. *OpenFAST Documentation*. National Renewable Energy Laboratory, 2024. Accessed: 2024-02-09.