This example considers modelling of the NREL 5MW baseline turbine mounted on the OC3 Hywind spar. This mock turbine system is recognised as an industry-standard reference model that is representative of a typical utility-scale, multi-megawatt wind turbine. The model itself utilises the turbine object, which is new to OrcaFlex v10.3. In this example we also use additional functionality added in the interim release, v10.3b.

The model uses Python scripts, required for external function and user defined results purposes, that require installation of Python 3. Hence, viewing and running this example can only be achieved through OrcaFlex v10.3b (or later) and Python 3.

The model (K01 Floating wind turbine.sim) is illustrated below. The turbine takes the form of a conventional three-bladed rotor, with variable-speed and variable blade-pitch control capabilities. The turbine hub is fixed to the nacelle, which houses numerous components necessary for power generation; such as the electrical generator and drivetrain assembly.

The nacelle is supported by the tower, which takes on a conical tubular steel construction. The tower is cantilevered atop a floating spar-buoy platform which, for the purpose of this example, is a rigid body. To prevent the system from drifting under the influence of environmental loading, the platform is moored to the seabed via three catenary lines.

The model that accompanies this example document is a simplified version of the OC3 Hywind model considered as part of Orcina Project 1405 - Wind Turbine Validation Report; which provides further details relating to modelling and analysis of turbine systems in OrcaFlex.
Building the Model

Turbine Object

The turbine object, named NREL-5MW, is composite in nature and includes functionality to model the generator, gearbox, hub, blades and associated control systems. The turbine data form is split into a number of separate pages, as highlighted in the image below.

On the generator page, a gear ratio of 97 is specified. This represents the number of turns the generator shaft makes from one rotation of the main rotor shaft. The mass moment of inertia, about the generator shaft, is also specified here.

The rotor hub serves as an interface between the turbine blades and the main shaft; which feeds into the nacelle. On the hub page, the hub radius is specified (1.5m); which represents the offset of the blade root from the turbine reference origin. Furthermore, the axial and transverse moments of inertia (about the main shaft) may be specified here, along with the hub centre of mass.
The **blades** page allows the blade count to be specified, along with the blade fitting angles (*pre-cone* and *initial pitch*). The blades are set to have **free** degrees-of-freedom (DOFs). In this scenario, the blade is akin to a line with torsion included i.e. each node has six calculated DOFs, three translational and three rotational, thus allowing each blade to deform appropriately.

The blades are equipped with the capability to pitch. Within the dynamic simulation, the pitching mechanism is controlled by a Python external function which is selected from the *pitch controller* drop-down menu. This allows for automatic adjustment of the angle-of-attack of the blade to be turned into or out of the wind; thus, optimising power production at certain wind speeds.

The turbine blade construction is represented by beam elements that are very similar in nature to OrcaFlex line objects i.e. each blade is divided into a series of segments (ordered from end A to end B) which are then modelled by straight massless segments with a node at each segment end. End A
of the blade is positioned at the blade root (where the blade interfaces with the rotor hub), with end B located at the blade tip.

Each blade is assigned an overall length of 61.5m and the blade nodes are directed along the blade-pitch axis from the blade root. The appropriate mass and inertia properties are lumped at each node. In this case, the blade construction and segment lengths have been set to match the same arrangement considered by NREL for the 5MW baseline turbine.

In order to accurately capture the aerodynamic loading effects, each blade section is assigned a specific wing type. The wing types are specified on the wing type data form, which can be accessed via the bottom left-hand corner of the turbine data form. Each wing type represents an aerofoil with corresponding coefficients of lift ($C_L$), drag ($C_D$), and moment ($C_M$) assigned to it. These coefficients define the aerodynamic loads applied to the wing for each given incidence angle ($\alpha$).

![Blade profile data](image)

The blade profile data is defined by the blade geometry, blade inertia and blade structure pages of the turbine data form. These profiles are defined at a number of discrete arc lengths along the blade length and are completely independent of the way the length is divided into sections (as defined by the blades page of the data form).

In this case, the arc lengths specified on the blade geometry page are intentionally set to each mid-segment point on the blades; the rationale being that aerodynamic loading is applied to the mid-segment frame. This is different to the approach taken for lines, where external loading is always applied to the line nodes. The properties of each segment are determined from linear interpolation of the specified blade profile data; and this data is also evaluated at the mid-segment arc length.
At each of the specified blade arc lengths, the blade inertia and structural properties are specified on the respective pages.

Aerodynamic loading on the turbine blades is captured through an implementation of the blade element momentum (BEM) theory. The BEM page allows the BEM calculation process to be controlled and parameterised by the user.

Control system modelling, for generator torque control and blade pitch control, are supported through a Python external function (named PythonController.py). This controller is adapted from a similar baseline control system documented by NREL as part of the OC3 Hywind study. This provides the functionality to model variable rotor speed and/or variable blade-pitch over the course of the dynamic simulation.

To enhance the functionality of the external function, a series of object tags are assigned to the turbine object on the tags page. In this capacity, the object tags allow user specification of whether or not the turbine system is floating (FloatingSystem) and also if the blade pitch actuator (UseActuator) is active. The blade pitch actuator is an optional second order sub-system of the blade pitch controller that functions to turn a blade pitch angle demand into a physical blade pitch response within the model.

For the blade pitch actuator system, additional tags are included to specify the natural angular frequency and damping ratio of the second order system; denoted ActuatorOmega and ActuatorGamma, respectively.
A dedicated properties report is also available for the turbine object. This can be accessed from the **model browser** by right-clicking on the turbine object and selecting the **properties** option, or by clicking once on the turbine object and using the **Alt+Enter** keyboard shortcut. The properties can also be accessed by the same methods from the turbine data form. As well as providing some basic details relating to rotor diameter/area, weight & inertia properties, the report also summarises individual blade weight & centre of mass properties. Note that OrcaFlex assigns a numeric designation to each blade: **Blade 1**, **Blade 2** and **Blade 3**.
When viewing the model in the wireframe format, it is also possible to view basic blade turbine details by hovering the mouse pointer over components of interest. This is shown below.

**Nacelle and Tower**

The nacelle serves as a connection point for both the turbine and tower objects. Within the developed turbine system model, the *nacelle* is modelled as a lumped 6D buoy with suitable mass, inertia and centre of mass properties assigned to it. As OrcaFlex cannot currently capture aerodynamic effects on a 6D buoy, a mass-less drag line is attached (*nacelle drag*) to the nacelle with suitable drag and added mass coefficients assigned by a corresponding line type, named *drag line*.

To increase blade-to-tower clearance, the rotor and nacelle are assigned a 5° tilt.
The tower structure is represented by a line object. The assigned line type (*tower_line_type*) utilises the *homogenous pipe* category, which allows for modelling of the appropriate variable outer/inner diameter profiles of the conical tower construction, as well as the associated tower physical properties.

**Spar Platform**

The spar platform is an axisymmetric surface-piercing buoy and is modelled using the *spar buoy* category of 6D buoy in OrcaFlex. In this capacity the *spar* is a rigid body, having six degrees-of-freedom, with the appropriate mass, inertia, centre of gravity, geometric and hydrodynamic properties assigned to it.

The geometry of the spar platform comprises three main sections, as illustrated below. In its static position, the spar has a draft of 120m; meaning that the top of the spar, upon which the tower is mounted, extends to an elevation of 10m above the still water level (SWL).

The spar buoy construction is split into a number of discrete cylinders. In order to accurately model surface-piercing effects, a fine discretisation of 1m is assigned to the top section, tapered section, and the top 8m of the bottom section. For the remainder of the bottom section, a discretisation of 10m is considered.
Moorings

The floating system is moored to the seabed by three separate mooring lines. Each mooring line is assigned suitable physical properties via the corresponding line type (mooring_line_type). The spar mooring lines and anchors are positioned evenly around the platform, in azimuth increments of 120°, with the anchor radius set to a radius of approximately 854m from the centre of the spar platform.

In order to introduce an element of yaw stiffness to the moored arrangement, each mooring line is connected to the spar platform by means of a ‘crowfoot’ (delta) connection. With reference to the image below, each leg of the delta connection (component A & B) is assigned a length and target segment length of 30m & 1m, respectively. Furthermore, end A of each line is connected to the buoy at a fixed radius of 5.2m from the spar’s central axis.
The interface between the delta connection and component C is facilitated by a deltaplate, which we have modelled as a 3D buoy in this example. The line representing component C of each mooring is assigned an overall length of 875m, with a target segment length of 10m.

For the line type representing the delta components (delta_line_type) the outer diameter, mass per unit length and axial stiffness values are set to half of those used for the mooring_line_type.

The various components of the mooring lines are modelled using the standard finite element representation. An alternative line representation, named analytic catenary, has been made available in OrcaFlex v10.3. For the analytic catenary line representation, the line loads are calculated from classical analytic catenary equations. This is different to the finite element method where the line is discretised into individual nodes that each carry degrees of freedom.

Due to the presence of multiple line objects in each of the moorings, brought about by the delta connection, we have decided to consider the default finite element representation for this example.
For further details related to the analytic catenary line representation, please see the OrcaFlex help directory: *Modelling, data and results | Lines | Analytic catenary.*

**User Defined Result**

The simulation also makes use of a *user defined result*. This feature is also new to OrcaFlex v10.3, and allows the user to define additional results within OrcaFlex. User defined results can be used in exactly the same way as any pre-defined results i.e. graphs of the user defined results may be plotted in the OrcaFlex GUI and results may be extracted using any of the various post-processing interfaces available to OrcaFlex.

When considering the generator power, the *turbine* object in OrcaFlex does not automatically account for mechanical-to-electrical conversion loss, which can typically take place in a system such as this. Therefore, a user defined result can be used to add this calculation to OrcaFlex.

For the purpose of this example, an efficiency of 94.4% is considered. The *user defined result* is then facilitated by a dedicated Python script, with file name `GenPower.py`, which is specified on the *general data form* (as shown below). The script makes use of an existing pre-defined result (generator power) to which a factor of 0.944 is applied to determine a new results variable named *generator power (94.4%).*

OrcaFlex loads the Python script at the beginning of the static calculation and will register the names (and other information) internally, based on the details provided by the specified script.

For further details and examples related to user defined results, please see the OrcaFlex help: *Modelling, data and results | General data | User defined results.*
Environment

The simulation considers a build-up (*stage 0*) duration of 100s followed by a *stage 1* duration of 500s. Airy waves, assigned a height of H=6m and period T=10s, are applied in the simulation environment. During the build-up phase, the sea conditions are slowly ramped up from zero in order to avoid sudden transients when starting the simulation. The wind is applied in the same direction as the modelled waves and the wind speed is linearly ramped from 0m/s to 15m/s during the first 250s of *stage 1*. A constant wind speed of 15m/s is then considered for the remainder of the simulation.

The ramped wind speed profile is specified on the *wind* page of the *environment data form* and is facilitated by the *time history (speed)* wind model. This allows for wind speed variation with time. The mean wind speed, specified on the *mean speed*, *mean direction* page, is used during the static calculations. The same mean wind direction is used for both the static and dynamic calculations.

The time history data (defined on the *data* page) then specifies the wind speed to be considered during the dynamic simulation.
The *preview* page presents a useful graph of the data contained in the time history. Linear interpolation is used to obtain the wind speed at intermediate times. This provides visualisation of the wind speed profile applied throughout the dynamic simulation. Also included on the graph are two vertical lines, coloured green and red, which indicate the beginning and end of the simulation, respectively.
As well as the *time history (speed)* wind model, OrcaFlex also allows wind to be specified using the following models:

- **Constant** – applies a wind of constant speed and direction throughout the static and dynamic simulations.
- **NPD spectrum, API spectrum** – the wind speed varies randomly over time, using a choice of either the NPD spectrum or the API RP 2A (1993) spectrum. For both cases, the wind direction remains constant over time.
- **Time history (speed & direction)** – both the wind speed and direction variation with time are specified explicitly by user specified time history data.
- **Full field** – full field wind allows for variation of wind velocity in both space and time, with data specified in an external file.

For further details related to wind modelling, please see the OrcaFlex help: *Modelling, data and results | Environment | Wind data*.

## Results

### Rotor/Generator Response

Opening the workspace file named _K01 Floating wind turbine rotor.wrk_, the graphs display *time history* results of the turbine rotor and generator response over the course of the simulation. In each of the graphs, it is clear to see how the behaviour of the system changes as the wind speed is steadily ramped from 0m/s to 15m/s over the first 250s of **stage 1**.

The upper right-hand graph displays the *main shaft torque* and the lower graphs display the *main shaft angular velocity* and *generator angular velocity*, measured in radians per second. The difference in the angular velocity profiles, between the main shaft and the generator shaft, is attributed to the gear ratio (= 97), which is specified on the *generator* page of the *turbine data form*. Recall that the gear ratio value represents the number of turns the generator shaft makes from one rotation of the main rotor shaft.

Loading the workspace file named _K01 Floating wind turbine generator_power.wrk_, the top graph shows the *generator power* results. The bottom graph displays the *generator power* (94.4%) results, which is facilitated by the *user defined result* script. The script has automatically added the specified variable to the *results menu* GUI, and assigned it to the *NREL-SMW* turbine object, in OrcaFlex (as shown below).
Over the final 200s of the simulation, where the turbine is exhibiting steady-state behaviour, note that the adjusted generator power settles to a mean value of 5.0MW – which is the power rating for the considered NREL 5MW turbine system. This power regulation is facilitated by the blade pitch controller, which functions to turn the blades in/out of the wind to alter the aerodynamic forces experienced by the rotor – and thus regulate the power generated by the system.

**Blade Response**

Opening the workspace file named *K01 Floating wind turbine blade_1.wrk*, the graphs display some basic *time history* results of the *blade 1* response. Note that when selecting these results, the *blade number* must be specified on the *results menu*, as shown below. The upper right-hand graph displays the *blade pitch* response, which eventually reaches a mean magnitude of ~9.6° once the wind has reached its steady state speed of 15m/s. The sign convention for blade pitch defines a positive value as an anti-clockwise rotation about the blade z-axis, looking from root to tip.

Note that *range graph* results are currently unavailable for turbine blades.
The bottom two graphs display node velocity results. In this case, the position along the blade must be specified on the results menu in addition to the blade number. The bottom-left graph displays the node velocity at the blade root (end A), while the graph on the bottom-right corresponds to the blade tip (end B). Intuitively, the node velocity experienced at the blade tip is far higher in magnitude than the velocity experienced at the blade root.
It is also possible to view results at a specific arc length on a given turbine blade. Opening the workspace file named **K01 Floating wind turbine blade_2.wrk**, the graphs display results for the tension, bend moment and torque experienced by blade 2 at a specified arc length of 30.75m (the mid-point of the blade).
Spar & Mooring Response

Lastly, open the K01 Floating wind turbine spar+mooring.wrk file to show some time history results pertaining to the spar and mooring line behaviour. The upper right-hand graph shows the X position of the spar over stage 1 of the simulation. The spar platform motions are extracted from a fixed x, y, z position of 0, 0, 120m relative to the platform base. This represents the point at which the centreline of the spar intersects the still water level, when the system is in its static position.
Note that over the first 250s of the simulation, as the wind speed is linearly ramped from 0 to 15m/s, the influence of aero- and hydrodynamic loading on the system causes the spar to drift significantly before being restrained by the mooring system. The influence of this spar motion is also evident in the lower graphs, which report the effective tension experienced by component C of moorings 1 and 2 (mooring\_1c and mooring\_2c). Note that the tension experienced by mooring\_1c decreases, while the tension experienced by mooring\_2c increases over the course of the simulation.