K01 5MW spar FOWT

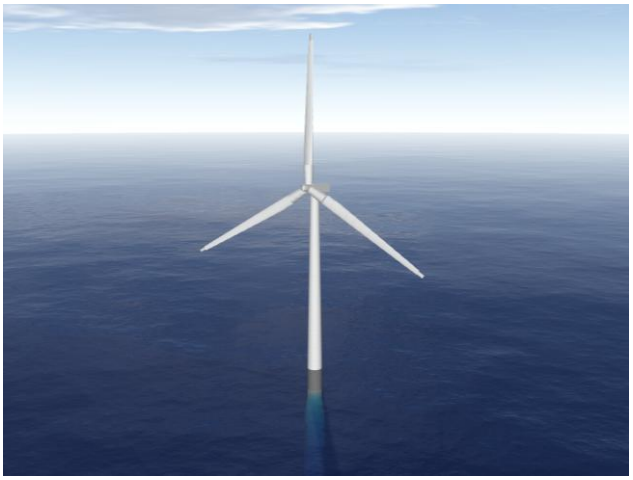
This example considers modelling of the NREL 5MW baseline turbine mounted on the OC3 Hywind spar. This mock turbine system is recognised as an industry-standard reference model which is representative of a typical utility-scale, multi-megawatt wind turbine.

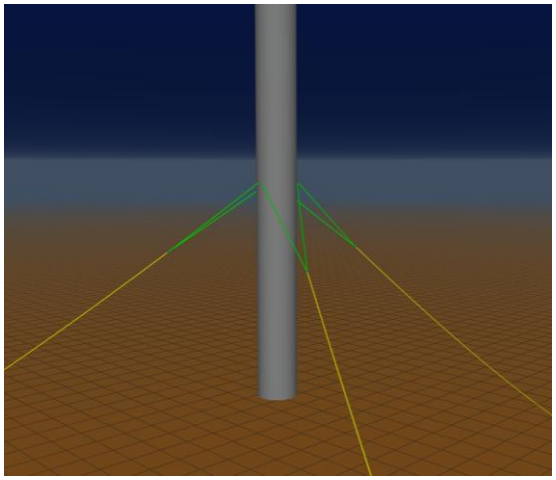
The model of the floating offshore wind turbine (FOWT) is shown below. The turbine takes the form of a conventional three-bladed rotor, with variable-speed and variable blade-pitch control capabilities. The turbine hub is fixed to the nacelle, which houses numerous components necessary for power generation, 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. The platform is moored to the seabed via three catenary lines.

The model accompanying this example document is a slightly adapted 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.

Note: the model properties considered in this example are based on our interpretation of the information and data available at the time of writing. If you use, or refer to, this model as part of your own analysis requirements then you must carry out the appropriate checks to confirm the implemented data are correct.





Python installation requirements

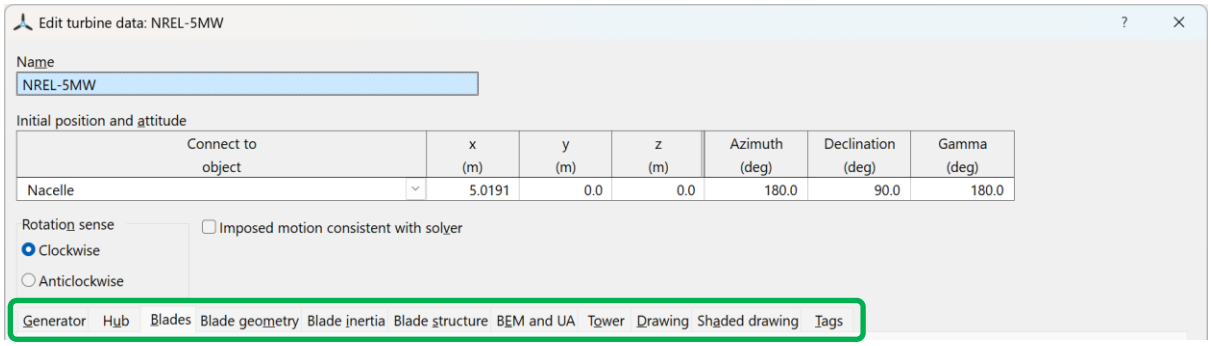
As this example uses Python scripts, Python 3 is required to view and run the simulation. It is also possible to view this model using the demo version of OrcaFlex which is available for download from the Orcina website: https://www.orcina.com/orcaflex/demo/.

Note, the OrcaFlex installation package includes the option to install a compact embedded Python distribution. If this option is selected when installing OrcaFlex, a separate installation of Python is not required for embedded Python features e.g. Python external functions and user defined results. The Python Interface: Installation help page provides further details about this subject.

Building the model

Turbine object

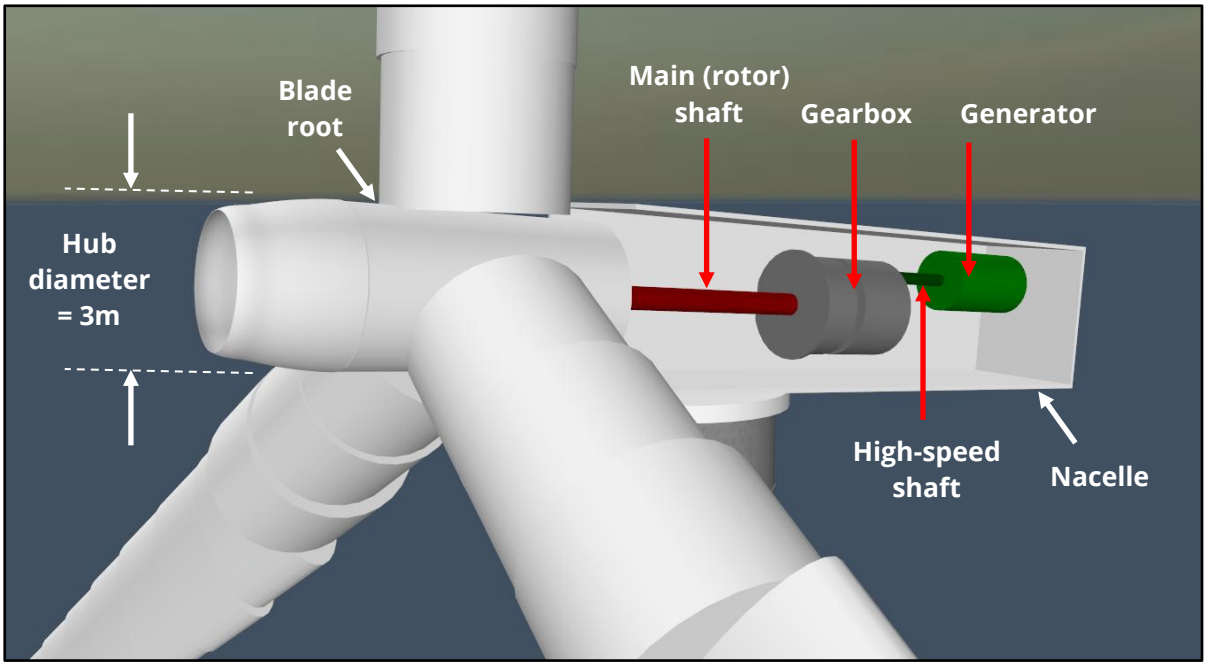
The turbine object, named *NREL-5MW*, is composite in nature and includes functionality to model the generator, gearbox, hub, blades and associated control systems. The turbine data form is split into a number of separate pages, as highlighted on the image below.



On the *generator* page, an external function named *generator torque* is nominated to model the influence of the appropriate control mechanism. A *gear ratio* of 97 is also specified here. 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.

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. The *mass*, the *axial* & *transverse moments of inertia* of the hub and the *centre of mass* are all specified here too.

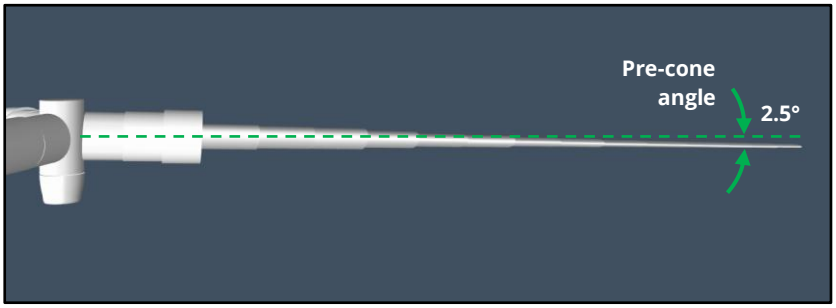
It is also possible to specify *main shaft stiffness & damping*. This allows for modelling of linear rotational stiffness and damping for the main shaft. Here, the *stiffness* is infinity, which means a rigid connection is made between the hub and gearbox's main shaft output. If a finite stiffness is specified, a degree of freedom (DOF) is introduced and the hub can then rotate, about the main shaft axis, relative to the gearbox's main shaft output.

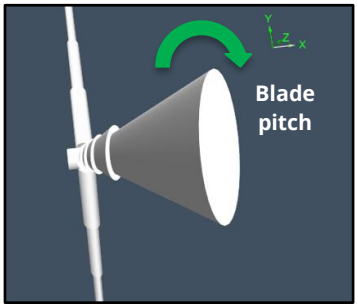


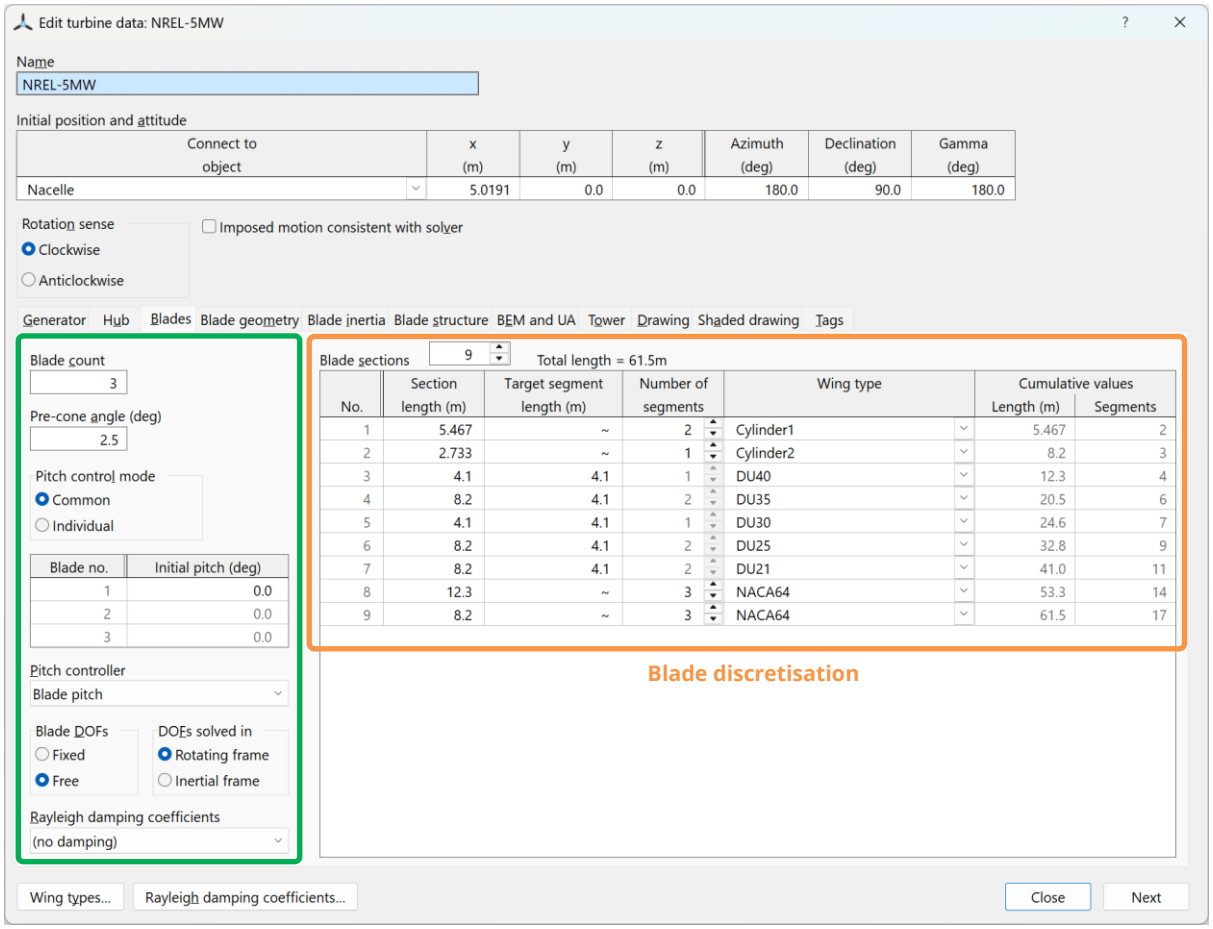
Turbine blades

Down the left-hand side of the *blades* page, the *blade count* is specified, along with the blade fitting angles (*pre-cone* and *initial pitch*). For the *common* pitch control mode, the initial pitch is specified for the first blade, and all other blades use that value. For the *individual* pitch control model, the initial pitch can be specified for each blade.

The initial pitch angle can subsequently be controlled, in the dynamic simulation, by a Python external function. In this example, the appropriate external function 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.







By default, the *blade DOFs* are set to *free*. In this scenario, the blade is akin to a line with torsion included i.e. each blade node has six calculated DOFs, three translational and three rotational, to allow each blade to deform appropriately. Alternatively, the blade DOFs can be set to *fixed* where each blade is modelled as a rigid body.

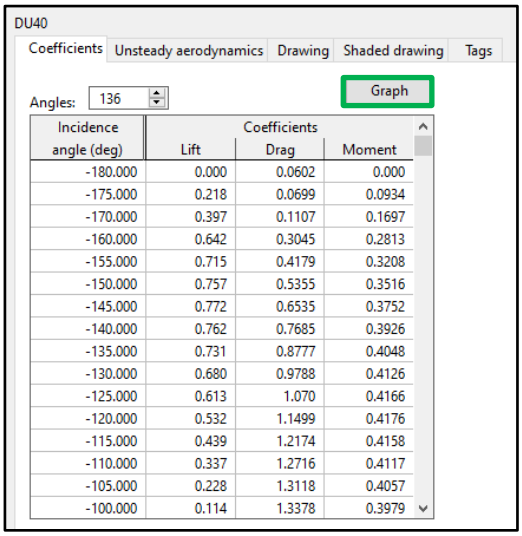
It is also possible to apply Rayleigh damping to the blades by specifying a named *Rayleigh damping coefficients* data set. However, mass proportional Rayleigh damping acts to damp out rigid body motions, and turbine blades undergo rigid body motion as the rotor turns, therefore mass proportional damping is probably inappropriate in this scenario. For this reason, it is recommended that the mass coefficient is set to zero when applying Rayleigh damping to turbine blades. In this example, no Rayleigh damping is applied.

The blade structural model is very similar in nature to the OrcaFlex line structural model i.e. each blade is divided into a series of segments, ordered from end A to end B, 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.

In this case, the blade discretisation and segment lengths have been set to match the same arrangement considered by NREL for the 5MW baseline turbine. To define the segmentation of the various blade sections, it is possible to specify a nominal *target segment length*. Alternatively, the target segment length can be set to a value of '~' and the required *number of segments* can then be specified. The blades are assigned an overall length of 61.5m and the appropriate mass & inertia properties are lumped at each node.

Wing types

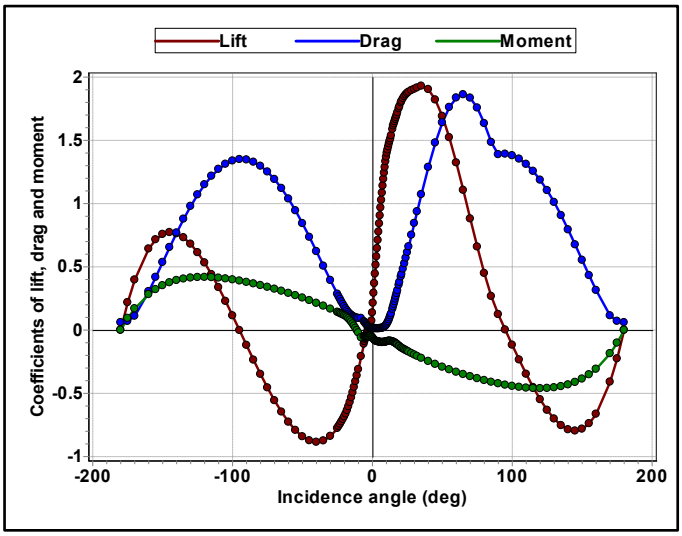
To model aerodynamic loads on each blade, each blade section is assigned a specific wing type. The wing types are specified on the *wing type* data form, which can be accessed from the bottom left-hand corner of the turbine data form.


DU40

Coefficients | Unsteady aerodynamics | Drawing | Shaded drawing | Tags

Angles: 136 Graph

Incidence angle (deg)	Coefficients Lift	Drag	Moment
-180.000	0.000	0.0602	0.000
-175.000	0.218	0.0699	0.0934
-170.000	0.397	0.1107	0.1697
-160.000	0.642	0.3045	0.2813
-155.000	0.715	0.4179	0.3208
-150.000	0.757	0.5355	0.3516
-145.000	0.772	0.6535	0.3752
-140.000	0.762	0.7685	0.3926
-135.000	0.731	0.8777	0.4048
-130.000	0.680	0.9788	0.4126
-125.000	0.613	1.070	0.4166
-120.000	0.532	1.1499	0.4176
-115.000	0.439	1.2174	0.4158
-110.000	0.337	1.2716	0.4117
-105.000	0.228	1.3118	0.4057
-100.000	0.114	1.3378	0.3979



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 (α). The *graph* button displays these coefficients graphically, allowing the input data to be checked visually.

The *unsteady aerodynamics* page of the *wing type* data form contains the data items required to control and parameterise the unsteady aerodynamics (UA) model for turbines. UA effects have not been considered for this model, but further details can be found in the OrcaFlex example K03 15MW semi-sub FOWT.

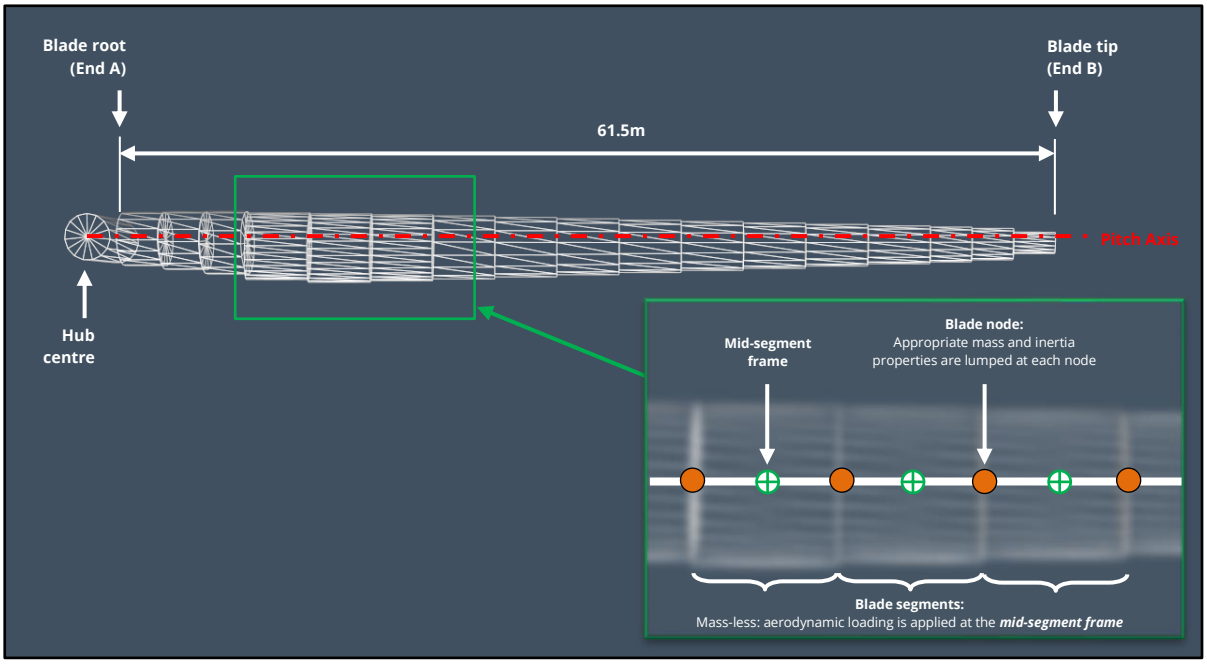
Blade profile data

The blade profile data is defined by the *blade geometry*, *blade inertia* and *blade structure* pages of the turbine data form. These pages are used to list the 'raw' blade data which define the physical properties of the blades along their length.

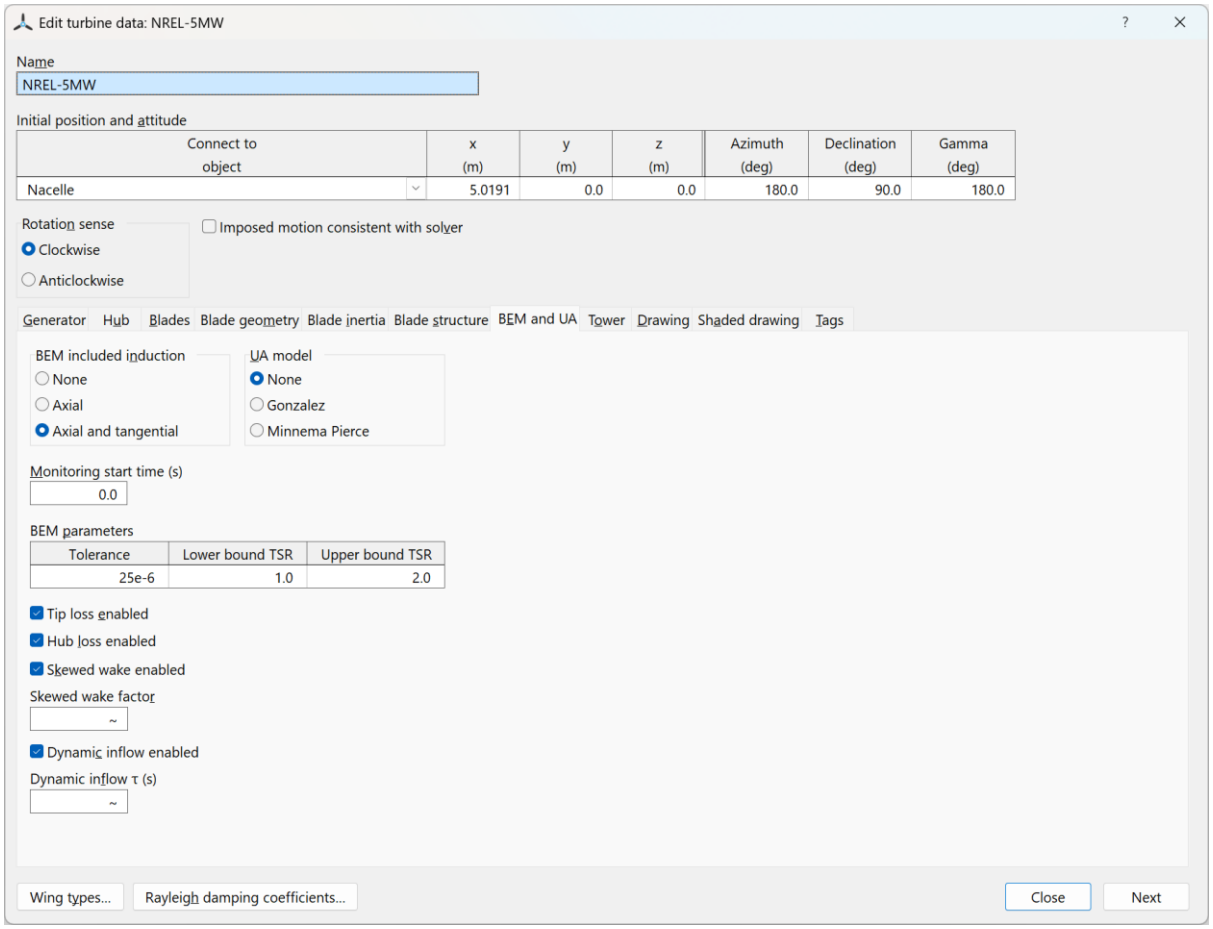
Each blade profile is given at a discrete arc length along the blade length and the specified arc lengths are completely independent of how the blade is divided into sections/segments (as specified on the *blades* page of the data form). As a result, the resolution of the blade profile data does not affect the quantity of nodes in the blade model, nor the simulation run time and file size.

The properties of each blade segment are determined from linear interpolation of the specified blade profile data, and the data are evaluated at each mid-segment frame. This means that you are free to modify the blade discretisation, on the *blades* page, without needing to modify the 'raw' blade profile data every time such a change is made.

Aerodynamic loading on the turbine blades is modelled through an implementation of the blade element momentum (BEM) theory, and the aerodynamic loads are applied at each mid-segment frame along the length of each blade. This is different to the approach taken for lines, where external loading is always applied to the line *nodes*.



The *BEM and UA* page of the turbine data form allows the BEM and unsteady aerodynamic (UA) calculation processes to be controlled and parameterised by the user. Further details about this can be found on the Modelling, data and results | Turbines | Turbine data | BEM and UA page of the OrcaFlex help.



Controller modelling

Controller modelling, for generator torque and blade pitch control, is supported through a pair of Python external functions, written to an external script named *PythonController.py*. This provides the functionality to model variable rotor speed and/or variable blade-pitch over the course of the dynamic simulation.

The controller itself is adapted from a similar baseline control system documented by NREL as part of the OC3 Hywind study.

In this case, to enhance the functionality of the *blade pitch* external function, a series of object tags are assigned to the turbine object on the *tags* page of the turbine data form. In this capacity, the object tags allow you to specify if 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.

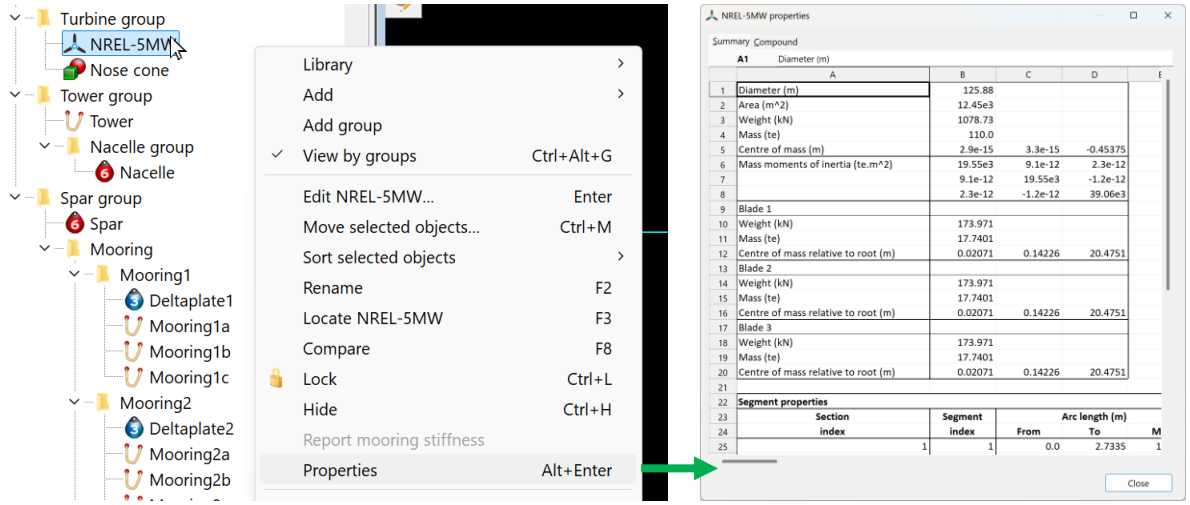
We also offer a specific guide to using external functions to model turbine controllers, which is available as a useful additional resource. The included PDF document provides a general guide to using Python external functions to model turbine controllers. This is intended to introduce the types of turbine control system that can be modelled in OrcaFlex and several example external functions, written in Python, are discussed.

The same files can also be accessed from the Modelling, data and results | Variable data | External functions help page (see the section 'External function examples').

Turbine properties report

It is also possible to access a properties report for the turbine object. This can be opened from the *model browser* by right-clicking on the turbine object and selecting the *properties* option, or by left-clicking once on the turbine object and using the *Alt+Enter* keyboard shortcut. The properties report can also be accessed from the turbine data form, using the same methods.

As well as providing some basic details relating to rotor diameter/area, weight & inertia, the report 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*. The report also provides a comprehensive summary of the blade segment properties, derived from the blade profile data.

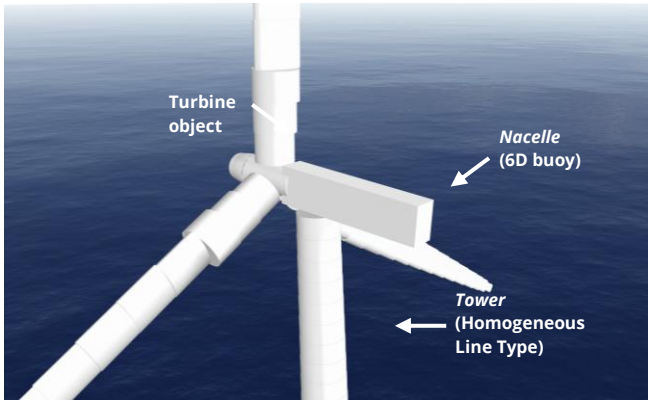


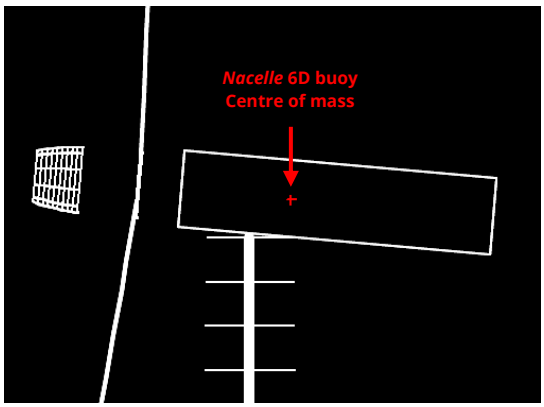
Nacelle and Tower

The nacelle serves as a connection point for both the turbine and tower objects. To increase blade-to-tower clearance, the rotor and nacelle are assigned a 5° tilt.

The *Nacelle* is modelled as a lumped 6D buoy with suitable *mass*, *mass moments of inertia* and *centre of mass* properties.

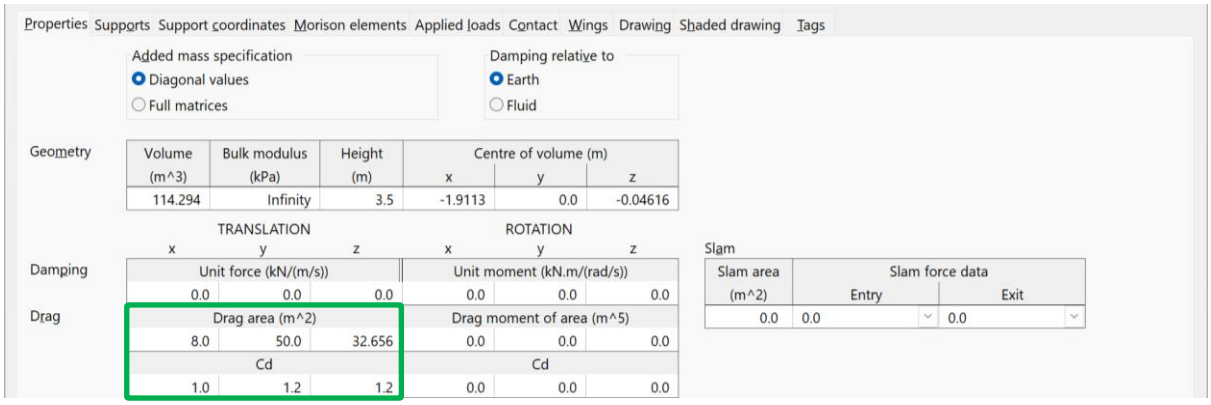
It is also possible to view the relevant origins of the *Nacelle* 6D buoy in the wire frame view. Each origin is represented by specific indicator marks. These marks are drawn optionally, determined by a preference which can be modified from the *view* menu or with the *Shift+Alt+Y* keyboard shortcut.

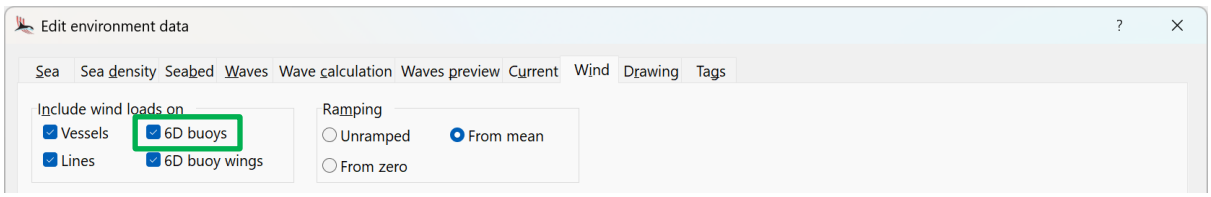




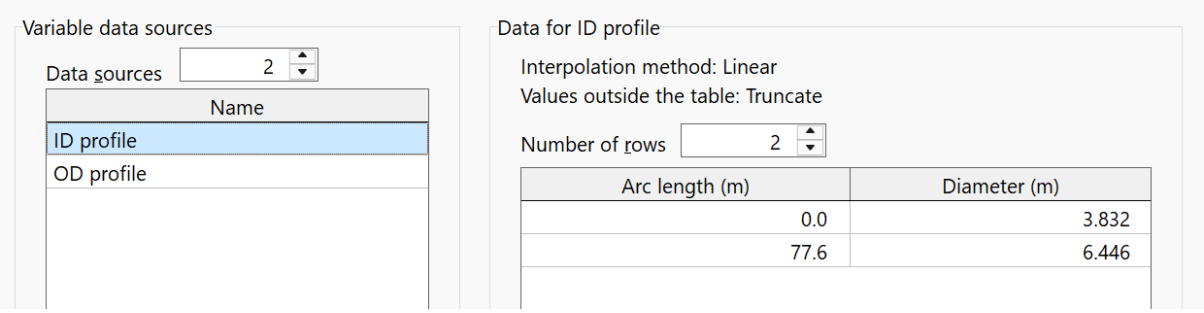
It should be noted here that the 6D buoy *centre of volume* has been set to the same position as the *centre of mass*. That's because the fluid centroid is the centre of volume, regardless of whether the 6D buoy is in air or fully submerged. Thus, the centre of volume is the point at which wind loads will be applied to the 6D buoy.

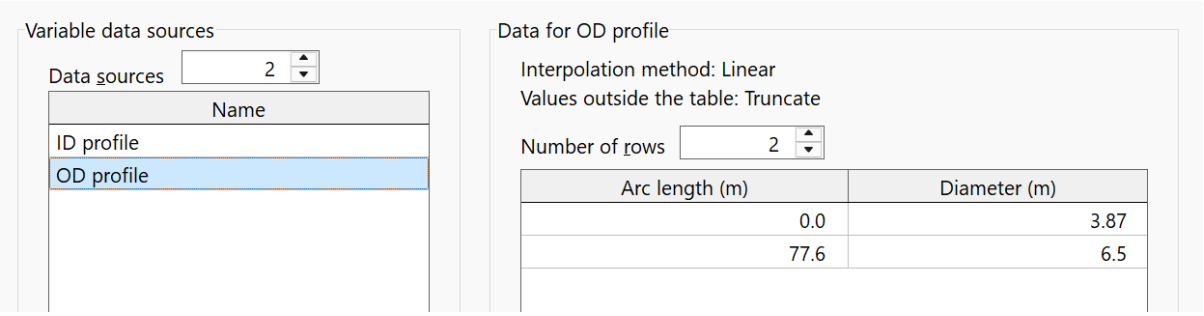
Appropriate drag areas and drag coefficients are also included on the *properties* page of the 6D buoy data form. Additionally, the *6D buoys* check box on the *wind* page of the environment data form is ticked. Altogether, this ensures that the nacelle will experience wind loading in the simulation environment.





The tower structure is represented by a line object. The *tower line type* utilises the *homogeneous 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.





The turbine can also be assigned a *tower influence* and / or *tower shadow* model, specified on the *tower* page of the turbine data form.

The available *tower influence* models allow the tower's influence on the undisturbed wind field, due to the so-called tower dam effect, to be accounted for. This can be used to account for the effect of the tower on the turbine inflow.

The available *tower shadow* models allow the wind field to be modified to account for the velocity deficit in the wake behind the tower. This can be important when modelling downwind turbines.

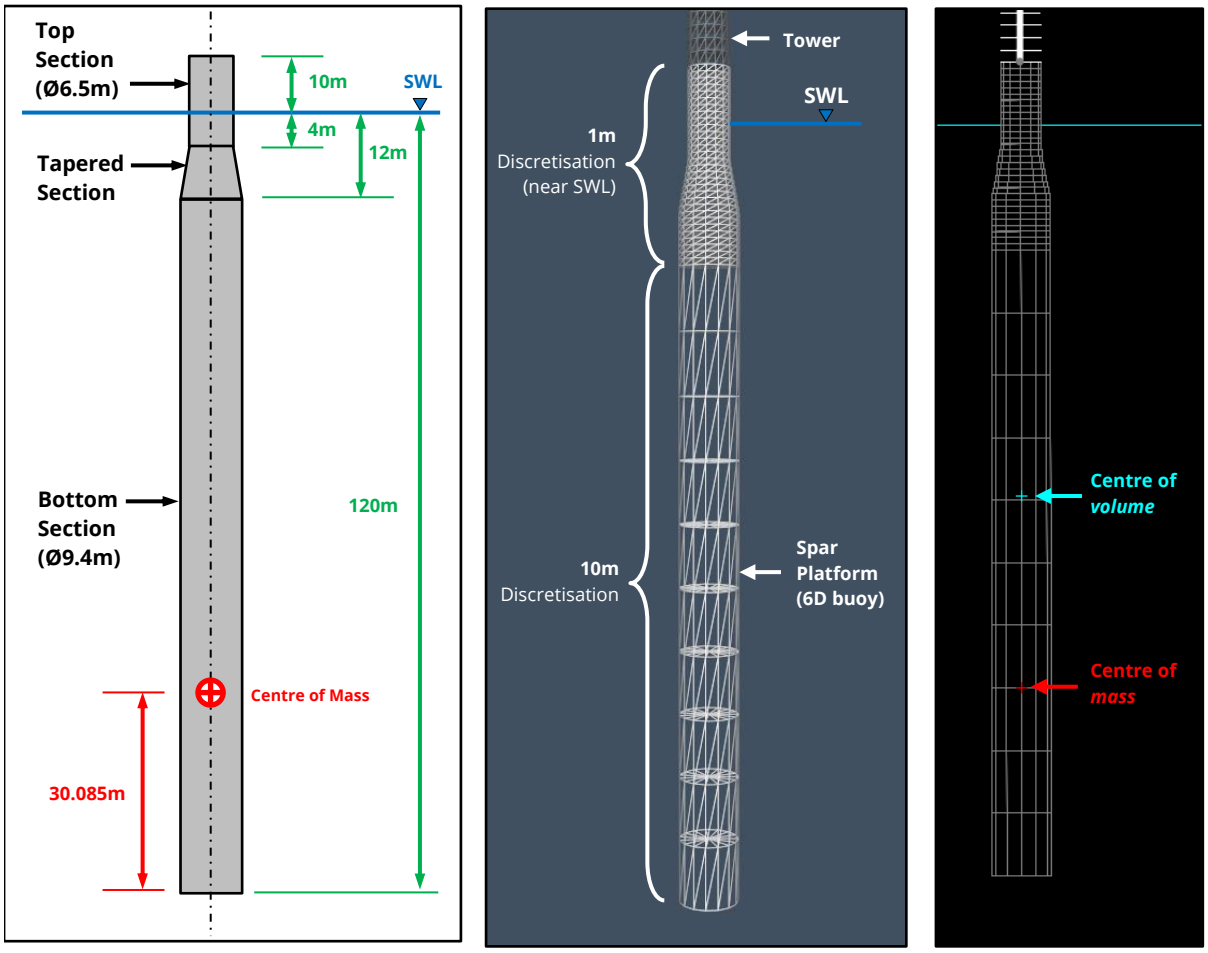
In this example, we have excluded any form of tower influence / shadow models. For further details, please see the Modelling, data and results | Turbines | Turbine data | Tower OrcaFlex help page.

Spar platform

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

The geometry of the spar platform comprises three main sections, 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 several discrete cylinders. 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.



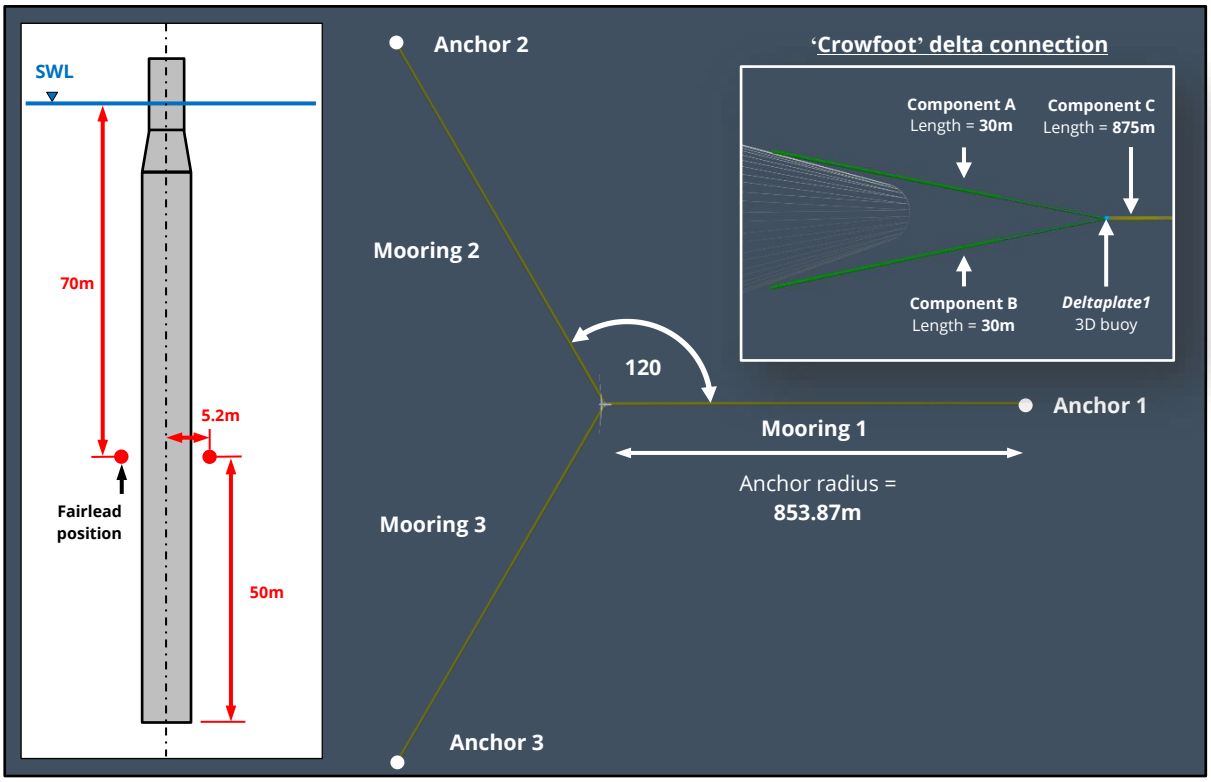
Again, it is possible to view the *Spar* origins in the wire frame view. The right-hand image above shows both the centre of mass and centre of volume origins for the 6D buoy. The centre of volume position is calculated automatically for a *spar buoy* type 6D buoy, based on the specified input data.

Note: the reported centre of volume position technically accounts for the spar being fully submerged. In this case, where the top 10m of the spar is above the water level, the true centre of volume of the submerged part of the *Spar* will differ slightly.

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 approximately 854m from the centre of the spar platform.

To introduce an element of yaw stiffness to the platform, each mooring line is connected to the spar 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.



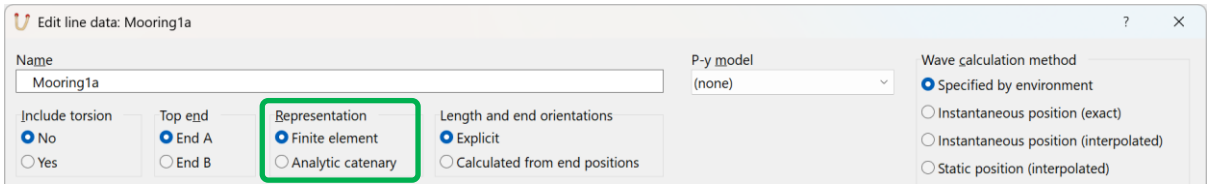
The interface between the delta connection and component C is facilitated by a delta plate, which we have modelled as a 3D buoy. The line representing component C of each mooring is assigned an overall length of 875m, with a target segment length of 10m.

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; however, an alternative line representation named *analytic catenary* can be used instead. The basis of this method is that 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, each carrying their own 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 Modelling, data and results | Lines | Analytic catenary OrcaFlex help page.



User defined result

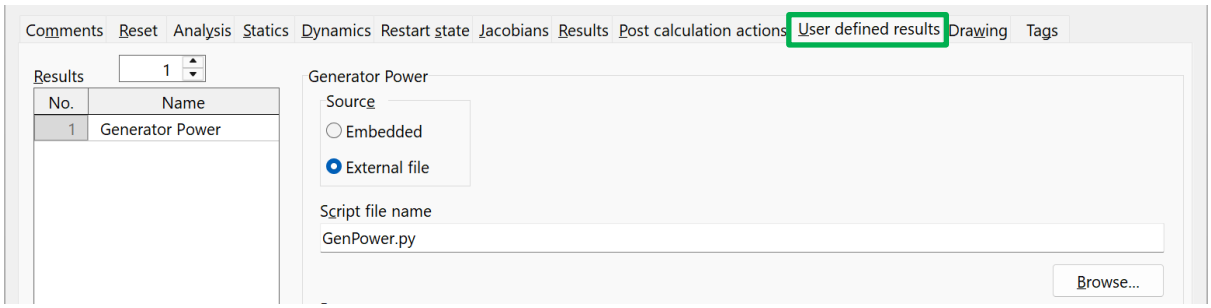
The simulation also makes use of a *user defined result* which allows us to define additional results within OrcaFlex. User defined results can be used in exactly the same way as any pre-defined results i.e. graphs of the user defined results may be plotted in the OrcaFlex GUI and results may be extracted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

When considering the generator power, the turbine object 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 implement this simple calculation in OrcaFlex.

For this example, an efficiency of 94.4% is considered. The *user defined result* is then facilitated by an external 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 specified by the script.

For further details and user defined results examples, please see the Modelling, data and results| General data | User defined results OrcaFlex help page.



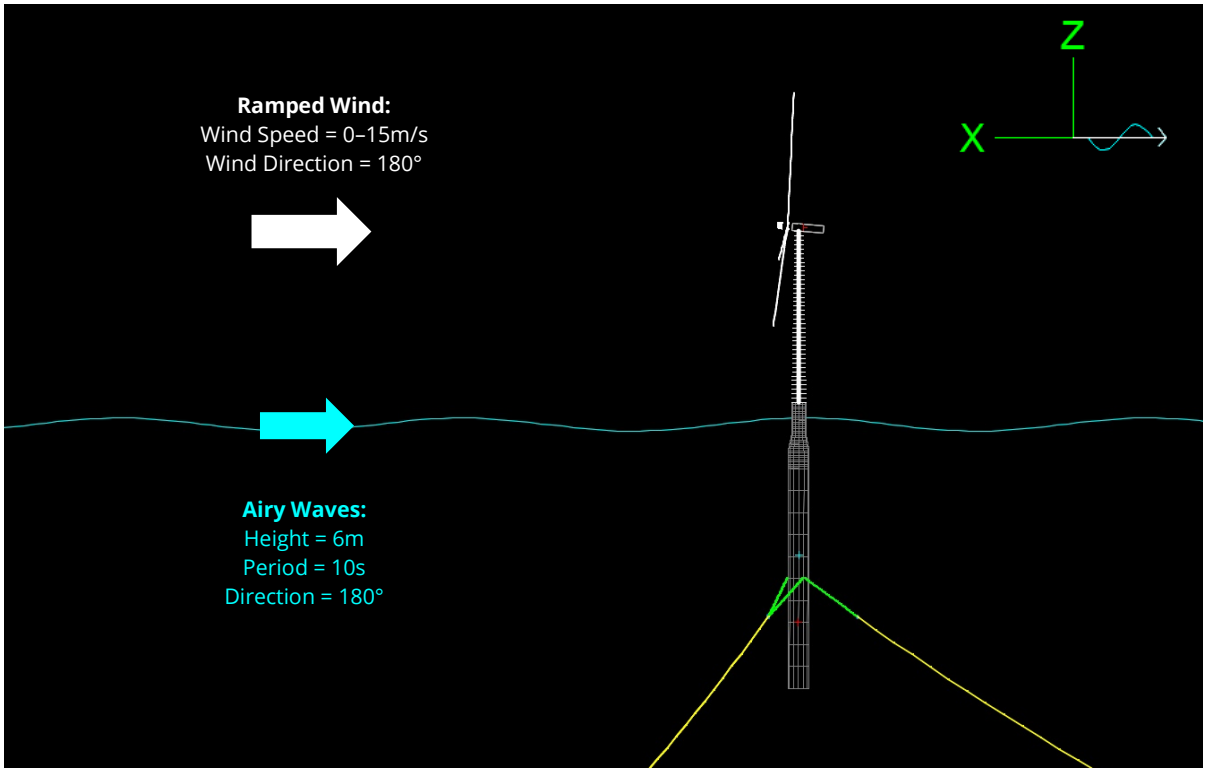

```python
def genPwr(info):
    power = info.modelObject.TimeHistory('Generator power', info.period, info.objectExtra)
    return power * 0.944

def UserDefinedResults(model):
    return (
        {
            'ObjectType': OrcFxAPI.otTurbine,
            'Name': 'Generator power (94.4%)',
            'Units': '$P',
            'TimeDomainFunction': genPwr
        },
    )
```

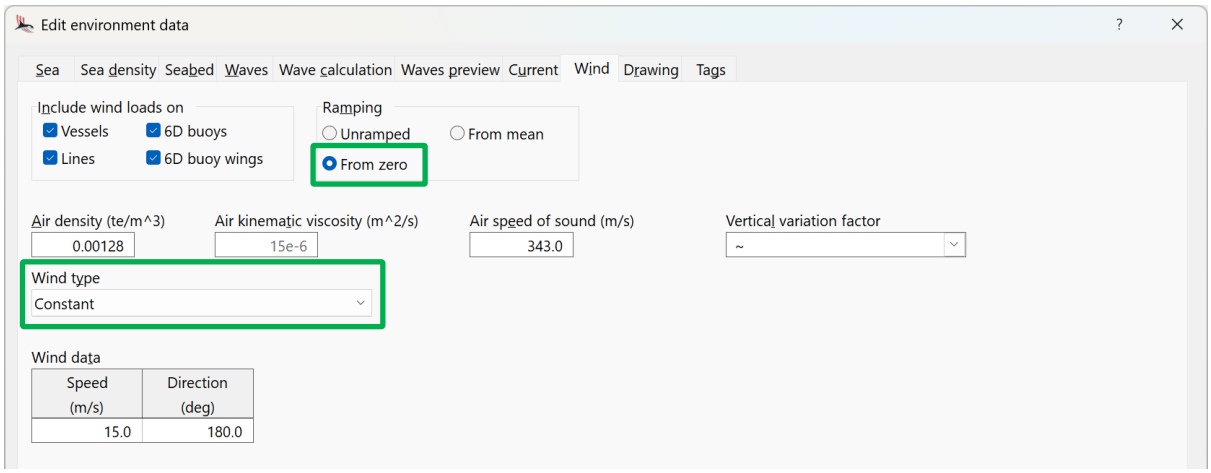
Environment

The simulation considers a build-up (*stage 0*) duration of 250s followed by a *stage 1* duration of 250s.

Airy waves, with 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 to avoid any sudden transients. 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 *stage 0*. A constant wind speed of 15m/s is then considered for the remainder of the simulation.



Here, a *constant* wind with a *speed* and *direction* of 15 m/s and 180 deg is specified on the *wind* page of the *environment* data form. The wind *ramping* is set to *ramp from zero*, which means that no wind is applied during the static analysis, and the wind velocity is ramped to its full value during the build-up period.



As well as the *constant* wind model, OrcaFlex 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, ESDU spectrum* – the wind speed varies randomly over time, using a choice of either the NPD spectrum, the API RP 2A (1993) spectrum or the ESDU spectrum. For all three wind models, the wind direction remains constant over time.
- *User defined spectrum* – defined by a table of pairs of values of frequency (*f* and *S*), the spectral energy *S(f)*.
- *User specified components* – the wind is defined as the sum of a number of given sinusoidal components. For each component, the frequency (or period), amplitude and phase lag are specified by the user.
- *Time history (speed)* – wind speed variation with time is specified by user specified time history data.
- *Time history (speed & direction)* – both the wind speed and direction variation with time are specified by user specified time history data.
- *Time history (speed, direction, shear & gust)* – this facilitates the modelling of wind fields with time varying linear shear, both horizontal and vertical.
- *Full field* – full field wind allows for variation of wind velocity in both space and time, with data specified in an external file. Two binary full field wind formats are supported: TurbSim (.bts) files; and Mann turbulence generator (.bin) files.

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

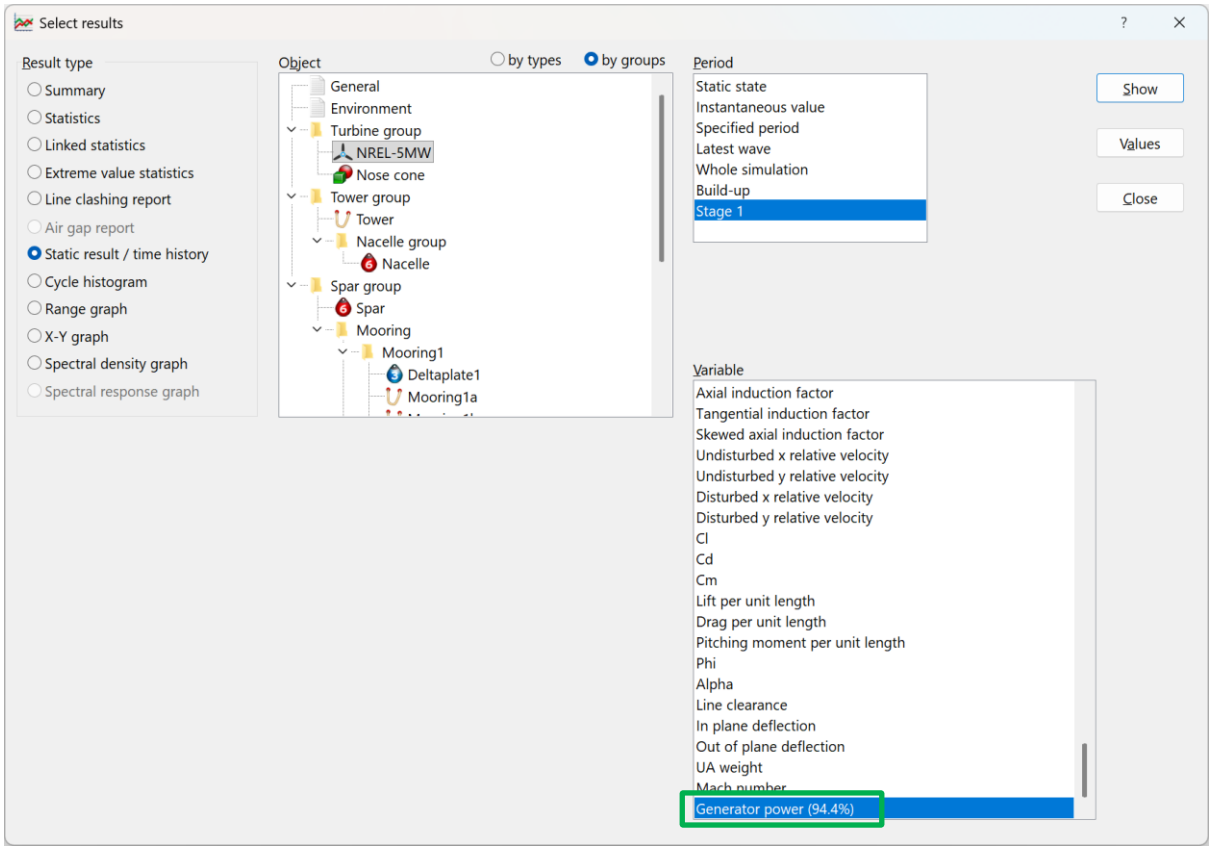
Results

Rotor and generator response

Opening the workspace file named *K01 Rotor results.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 *stage 0*.

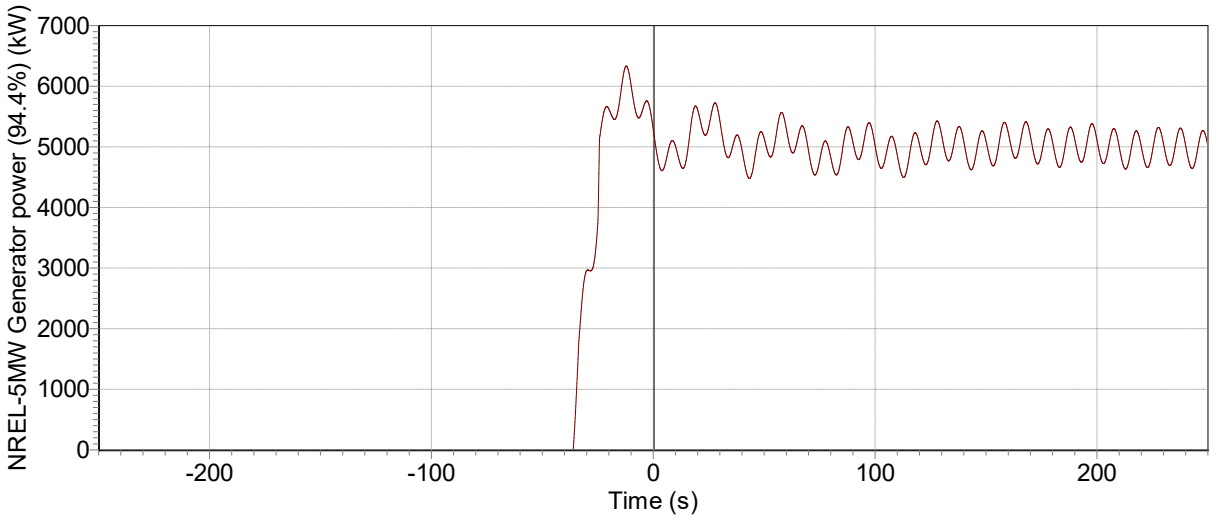
The upper right-hand graph displays the *main shaft torque* and the lower graphs display the *main shaft angular velocity* and *generator angular velocity*, both of which are measured in radians per second. The difference in the angular velocity profiles, between the main shaft and the generator shaft,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

Loading the workspace file named *K01 Generator power.wrk*, the top graph shows the *generator power* results over *stage 1*. The bottom graph displays the *generator power (94.4%)* results, facilitated by the *user defined result* script. The script has automatically added the specified results variable to the *select results* form and assigned it to the *NREL-5MW* turbine object.



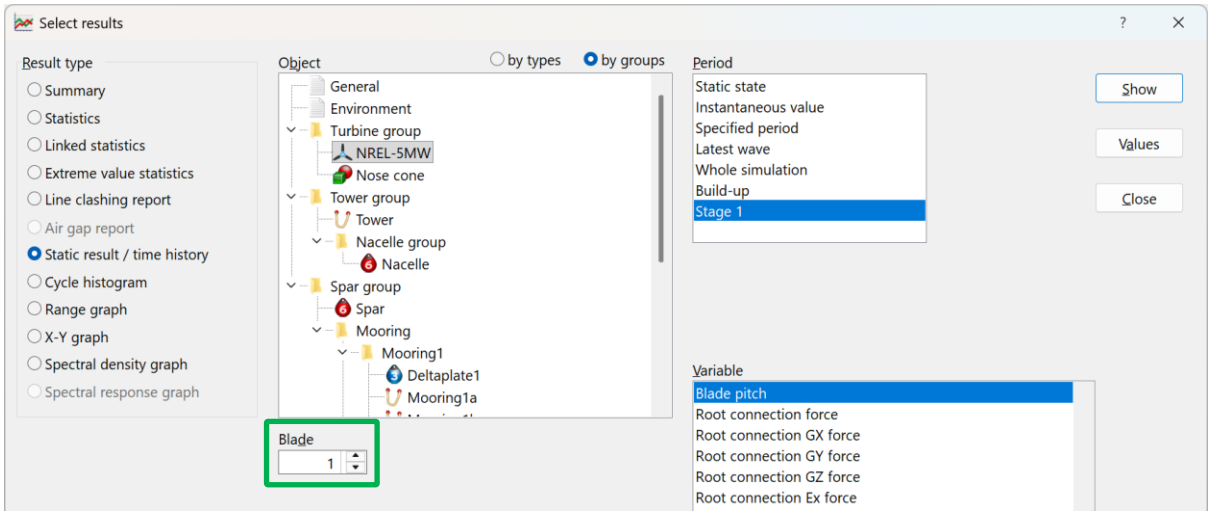
During stage 1, where the turbine is exhibiting steady-state behaviour, note that the adjusted generator power settles to a mean value of 5000kW (5MW) – which is the power rating for the considered NREL 5MW turbine system. This power regulation is achieved by the blade pitch controller, which angles the blades to alter the aerodynamic forces experienced by the rotor, thus regulating the power generated by the system.

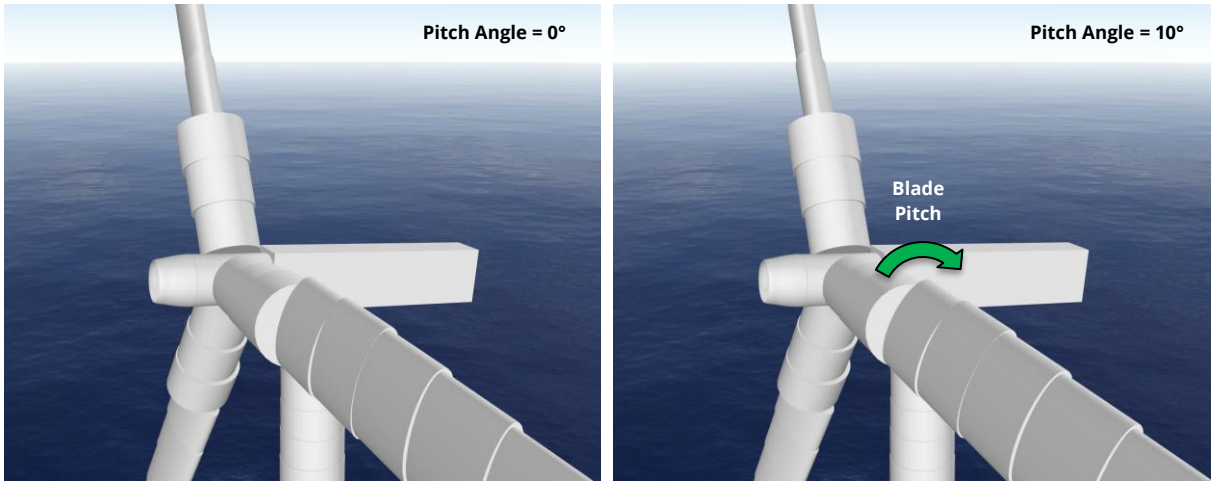
Note: *if viewing this model using the OrcaFlex demo version, it is not possible to view the user defined result output in this example. For information, a time history graph of the generator power (94.4%) user defined result is presented below.*



Blade response

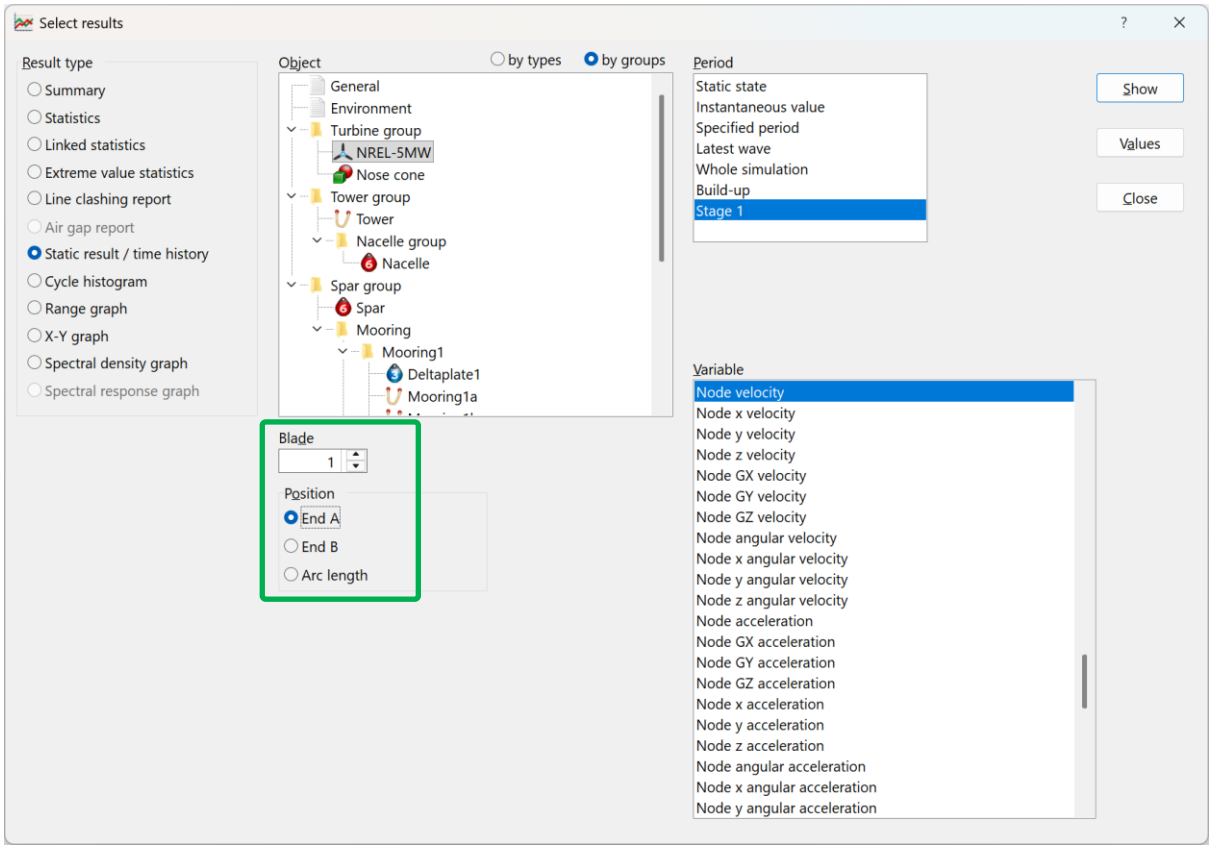
Opening the workspace file named *K01 Blade 1 response.wrk*, the graphs display some basic *time history* results of the *blade 1* response. Note that when selecting these results, the *blade number* must be specified on the *select results* form, as shown below. The upper right-hand graph displays the *blade pitch*, which eventually reaches a mean angle of approximately 10°. 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.



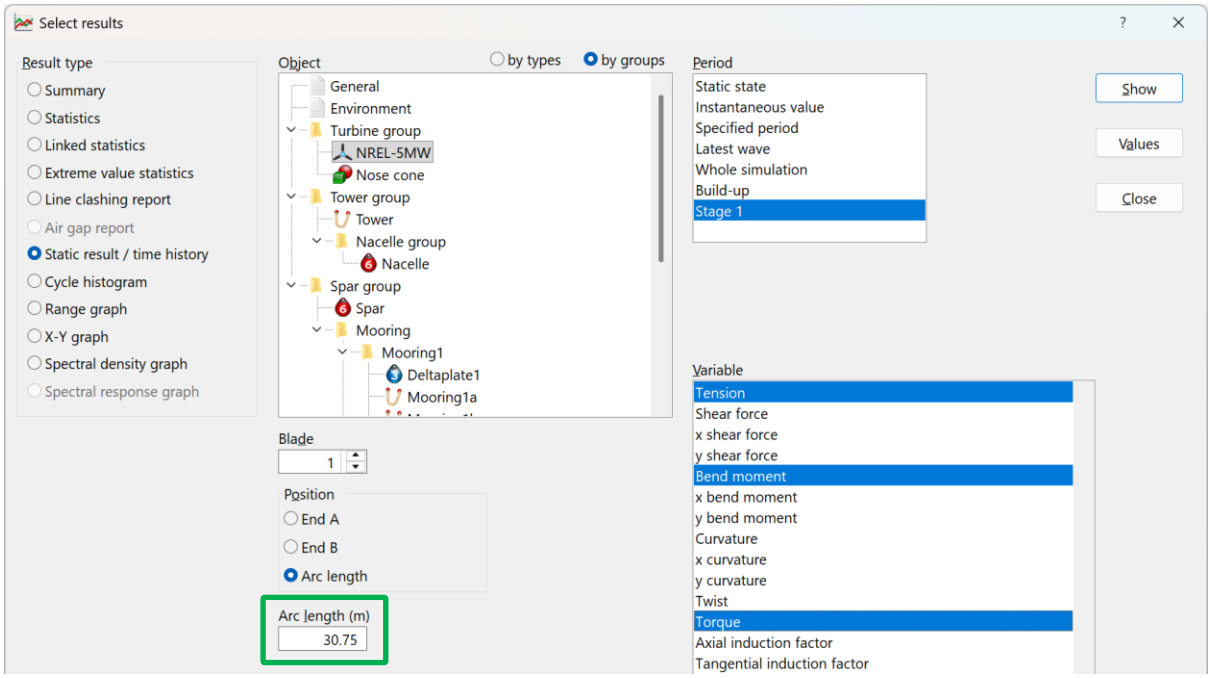


The bottom two graphs show some *node velocity* results. In this case, the *position* along the blade must be specified on the *select results* form in addition to the *blade number* (shown on the image below).

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 than the velocity experienced at the blade root.

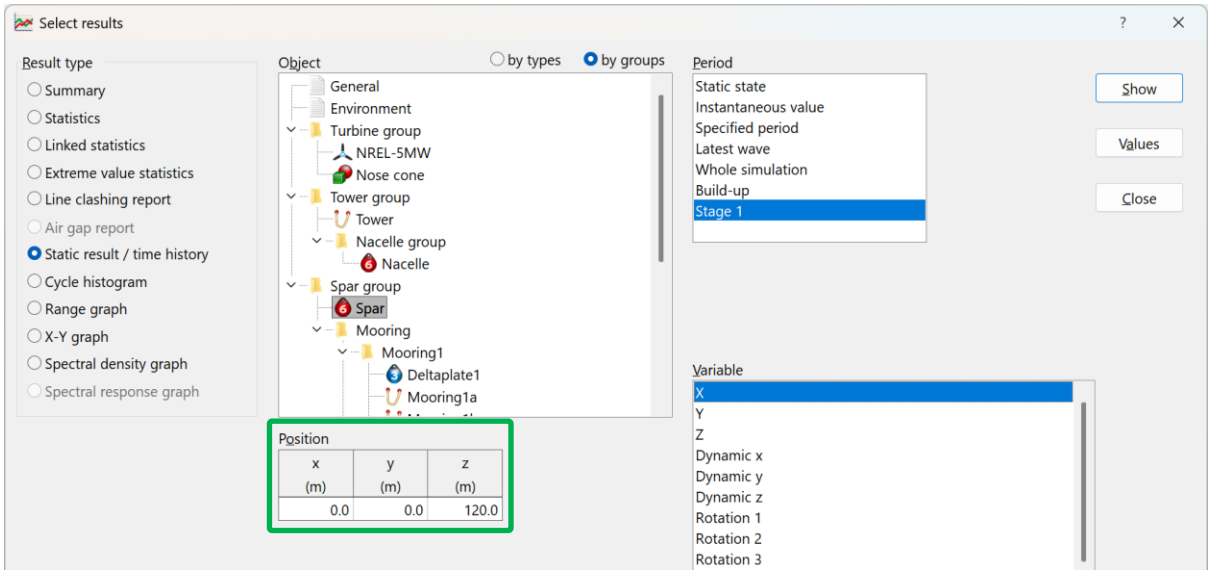


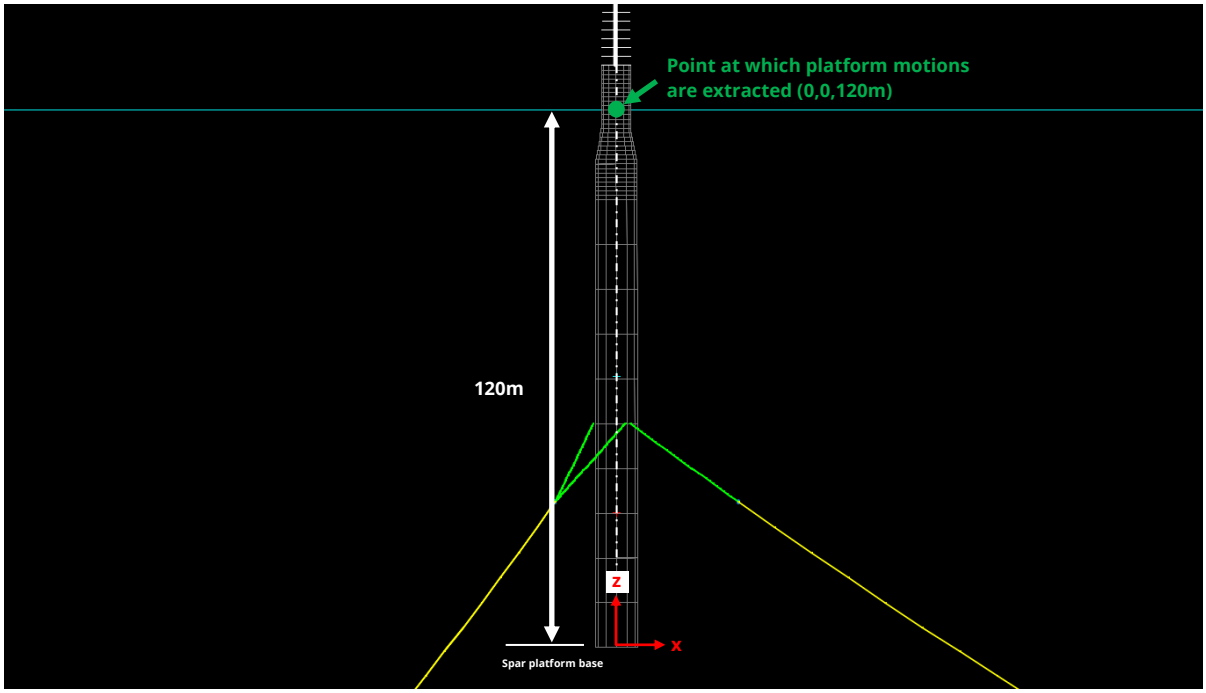
It is also possible to view results at a specific *arc length* on a given turbine blade. Opening the workspace file named *K01 Blade 2 response.wrk*, the graphs display results for the *tension*, *bend moment* and *torque* experienced by *blade 2* at a specified arc length of 30.75m (the blade mid-point).



Spar & mooring response

Lastly, opening the workspace *K01 Spar and mooring results.wrk* shows some *time history* results of the spar platform and mooring line behaviour. The upper right-hand graph shows the global *X* position of the *spar* over the *whole simulation*. The *Spar* platform motions are extracted from a fixed x, y, z position of *0, 0, 120* m relative to the platform base. This represents the point at which the centreline of the spar intersects the still water level i.e. when the system is in its static position.





Note that during stage 0, as the wind speed is ramped from 0 to 15m/s, the influence of aero- and hydrodynamic loading on the system causes the spar to drift 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 (*mooring1c* and *mooring2c*). Note that the tension experienced by *mooring1c* decreases, while the tension experienced by *mooring2c* increases.