

## K03 15MW semi-sub FOWT

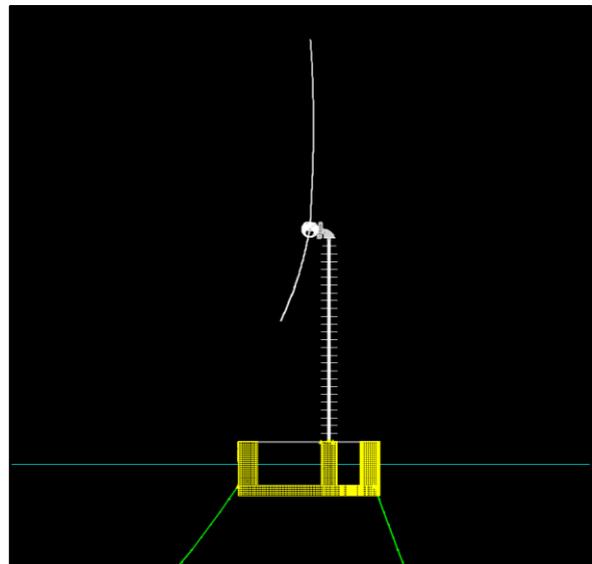
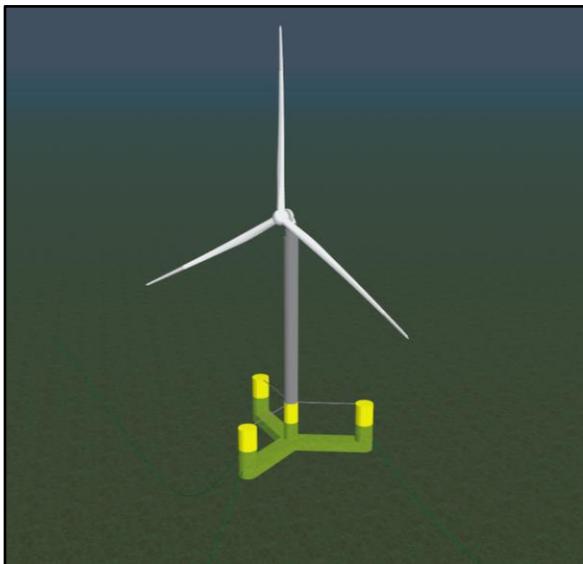
This example models a floating offshore wind turbine (FOWT). The turbine rotor represents version 1.1.3 of the 15MW reference wind turbine (RWT), developed as part of the International Energy Agency's (IEA) Wind Task 37.

The turbine takes the form of a three-bladed rotor with variable-speed and collective blade-pitch control capabilities. The turbine hub is connected to the nacelle, which houses a direct-drive synchronous generator. The nacelle is supported by the tower which is mounted upon the University of Maine (UMaine) VoltturnUS-S reference semi-submersible platform.

The semi-submersible design, chain catenary mooring and floating-specific tower are all documented by the [Definition of the UMaine VoltturnUS-S Reference Platform](#) technical report.

We have generated the OrcaFlex RNA model from the v1.1.3 RWT OpenFAST dataset, which can be downloaded from the [IEA-15-240-RWT GitHub repository](#). Further details about this can be found in the validation report summarising a [Comparison of loads from OrcaFlex and OpenFAST for the IEA Wind 15 MW RWT](#).

**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 (11.3 onwards) 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 modelling

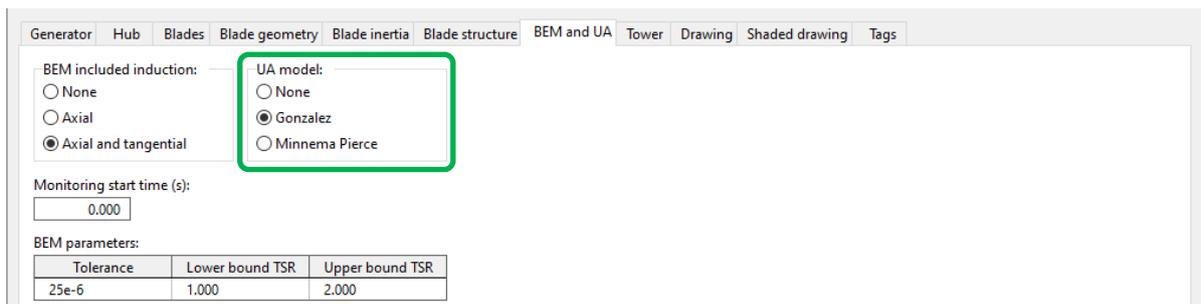
For an overview of some important wind turbine modelling aspects, please refer to our [K01 5MW spar FOWT](#) example. Further information about turbine blade pre-bend modelling can be found in the [K02 10MW fixed-bottom OWT](#) example.

### Unsteady aerodynamics

For this example, we have chosen to capture the effects of unsteady aerodynamics (UA) for the turbine blades. This allows for modelling of effects like unsteady attached flow, trailing-edge flow separation, dynamic stall, and flow reattachment.

On the [BEM and UA](#) page of the turbine data form, it is possible to choose from three different options: *None*, *González* or *Minnema Pierce*. When the option *None* is selected, the static (i.e. steady) aerofoil coefficients – specified by the *wing type* data – are used to calculate the aerodynamic loads on the blades.

However, when one of the UA models is selected, OrcaFlex will automatically modify the aerofoil coefficients to account for unsteady effects. For further details, please refer to the [Theory | Turbine theory | Aerodynamic load](#) help page.



BEM included induction:

None

Axial

Axial and tangential

UA model:

None

Gonzalez

Minnema Pierce

Monitoring start time (s):

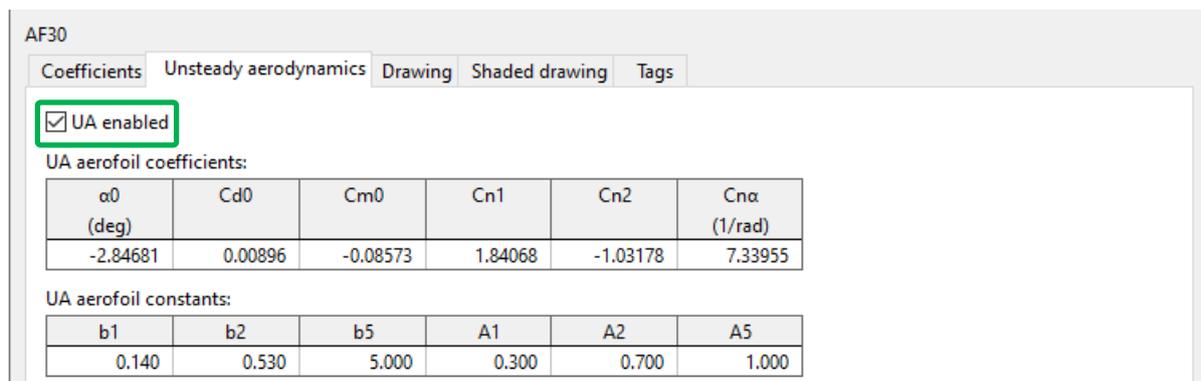
0.000

BEM parameters:

Tolerance	Lower bound TSR	Upper bound TSR
25e-6	1.000	2.000

We have chosen to apply the González UA model for this example but, for this model to apply, the wing type data must also be set appropriately. This means that the *UA enabled* check box must first be ticked, on the *Unsteady aerodynamics* page of the *Wing type* data form, for any wing types where UA effects are to be captured.

The necessary UA wing type data must then be populated. Here, we have incorporated the necessary parameters, coefficients, and constants from an equivalent OpenFAST BeamDyn blade model. For further details about the requested data, please refer to the [Modelling, data and results | Wing type data](#) page of the OrcaFlex help, and the [NREL paper](#) referenced therein.



AF30

UA enabled

UA aerofoil coefficients:

$\alpha_0$ (deg)	$C_{d0}$	$C_{m0}$	$C_{n1}$	$C_{n2}$	$C_{n\alpha}$ (1/rad)
-2.84681	0.00896	-0.08573	1.84068	-1.03178	7.33955

UA aerofoil constants:

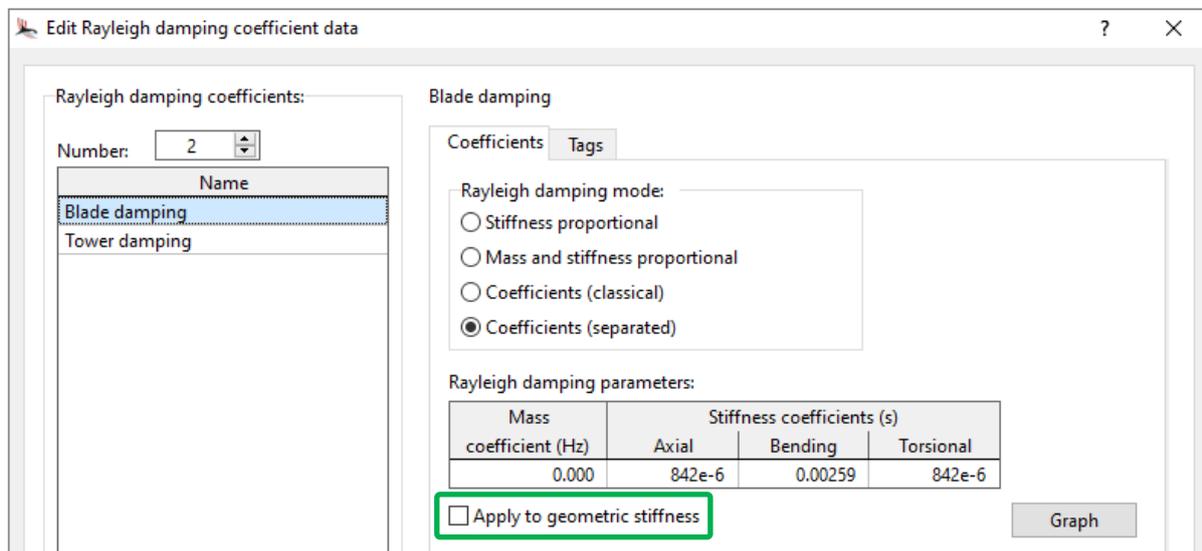
b1	b2	b5	A1	A2	A5
0.140	0.530	5.000	0.300	0.700	1.000

## Rayleigh damping

We have also included some structural damping for the turbine blades, by selecting a named set of Rayleigh damping coefficients on the *Blades* page of the *15MW RWT* data form. Again, the coefficients considered here come from an equivalent OpenFAST BeamDyn blade model.

There are a couple of important warnings to make about the implementation of Rayleigh damping for turbine blades:

- We do not recommend that mass proportional Rayleigh damping be considered. This can act to damp out rigid body modes, such as the rotor rotation, which can lead to inaccurate rotor performance.
- Ticking the option *Apply to geometric stiffness* on the *Rayleigh damping coefficient* data form is also known to affect the accuracy of the rotor response. That option has *not* been ticked for the *Blade damping* dataset used in this example.



Rayleigh damping coefficients:

Number: 2

Name
Blade damping
Tower damping

Blade damping

Coefficients Tags

Rayleigh damping mode:

Stiffness proportional  
 Mass and stiffness proportional  
 Coefficients (classical)  
 Coefficients (separated)

Rayleigh damping parameters:

Mass coefficient (Hz)	Stiffness coefficients (s)		
	Axial	Bending	Torsional
0.000	842e-6	0.00259	842e-6

Apply to geometric stiffness

Graph

The *Tower* has also been assigned some Rayleigh damping coefficients. Here, the tower damping coefficients have been matched with the coefficients in the HAWC2 dataset because they parametrise a damping model that is consistent with that employed by OrcaFlex turbine blades, i.e. Rayleigh damping.

For further details about the Rayleigh damping model available in OrcaFlex, please refer to the [Modelling, data and results | Lines | Rayleigh damping](#) help page.

## Controller modelling

Controller modelling for the turbine is supported through OrcaFlex external functions, which are used as a wrapper for the NREL Reference Open Source Controller (ROSCO) dynamic-link library (DLL). This example uses a **native code** external function.

Below is a summary of the controller setup:

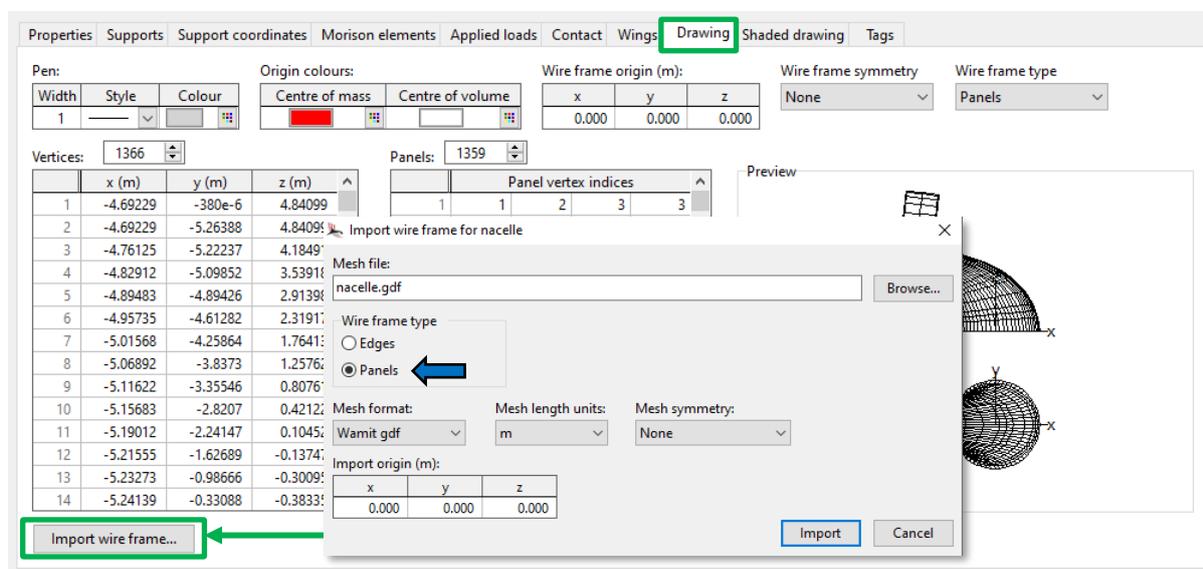
- Included amongst the example files is a DLL (*BladedControllerWrapper.dll*) that exports the *BladedController* native code OrcaFlex external function. This function is responsible for calling the ROSCO DLL and returning controller output to OrcaFlex as required, switching between generator torque and blade pitch, depending on the data name calling it.
- In the OrcaFlex model, blade pitch and generator torque are then both specified by the *Controller* external function data source which uses the *BladedController* function.
- Object tags are used to identify the DLL to be wrapped by the external function and its associated input file. Both of which are included amongst the example files. The tags have been assigned to the *15MW RWT* turbine object, on the *tags* page of the *turbine data* form.
- The *ControllerDLL* tag specifies the path, relative to the model, of the 64-bit, v2.7.0, ROSCO DLL (*libdiscon.dll*). Although this example makes use of a 64-bit version of the DLL, a 32-bit version is also available for download from the ROSCO GitHub repository.
- The *InputFile* tag specifies the path, relative to the model, of the ROSCO parameter input file (*DISCON-UMaineSemi.IN*). This can be downloaded from the [IEA-15-240-RWT GitHub repository](#). The contents of the input file can be viewed and edited using a text file editor.
- The external function is responsible for calling the DLL, once per time step. When called, the DLL updates its state and calculates all the necessary controller values. A swap array is used to communicate between the external function and the DLL. Before the DLL is called, the appropriate elements (slots) in the swap array are populated with the values to be input to the calculation. The swap array is then passed into the DLL when it is called. After it has been called, the DLL will have calculated the control values which are then read out of the swap array and returned to OrcaFlex through the external function. Please note, the wrapper only populates the swap array sufficiently to support the ROSCO control modes used by this example. If other controller modes are needed, then the wrapper may need to be further extended.
- In this example, the ROSCO wind speed estimator is not used, i.e. *WE\_Mode* is "0" in the parameter input file. If the wind speed estimator is enabled, this introduces a further DLL dependency: the rotor performance file. The *PerFFileName* variable of the ROSCO input file specifies name and location of this file, as a path relative to the input file. For completeness, it is included amongst the example files (*Cp\_Ct\_Cq.IEA15MW.txt*) and can be downloaded from the [IEA-15-240-RWT GitHub repository](#).

For more detail about using OrcaFlex external functions to wrap the ROSCO DLL, please see our [guide to using external functions to model turbine controllers](#). This includes documentation of the swap array, information on further modifying behaviour with object tags, a Python external function wrapper, and some simple guidance on debugging issues.

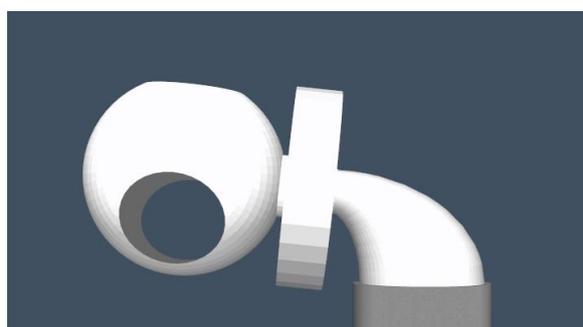
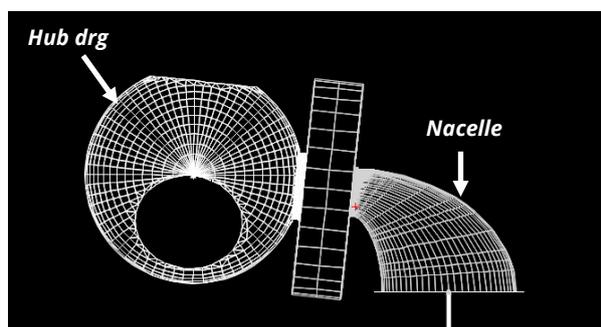
## Nacelle modelling

The *Nacelle* is modelled as a lumped 6D buoy with the necessary mass, inertia, centre of mass and aerodynamic drag properties assigned to it. To provide a useful visual representation of the structure, in the OrcaFlex model view, we have used the available SolidWorks CAD models for the RWT to create a panel mesh of the nacelle in the .gdf file format (*nacelle.gdf*). This was created using Rhino.

We have imported this file, through the *drawing* page of the *6D buoy data* form, to draw the nacelle in the model view. As part of the import process, the *wire frame type* has been specified by 'Panels' (pictured below). This is preferred over the 'Edges' option because it will usually give a better representation of the object in the shaded graphics view.



We have taken the same approach to create a panel mesh file for the turbine hub (*hub.gdf*). As the panel mesh file cannot be directly imported to the turbine object, we have attached a 'dummy' 6D buoy – with negligible properties – to the turbine object (*Hub drg*) to draw the hub in both the wireframe and shaded graphics views.



**Note:** it may be useful to import the mesh using the 'Edges' *wire frame type* if the data needs to be read by an older version of OrcaFlex which does not support wire frames specified by panels. If doing so, a panel mesh file may be imported as a shaded drawing file – separately through the *shaded drawing* page of the *6D buoy data* form – to draw the 6D buoy in the shaded graphics view.

Further details about importing wire frame drawing data can be found on the corresponding page of the OrcaFlex help: [User interface | 3D views | Importing wire frame drawing data](#).

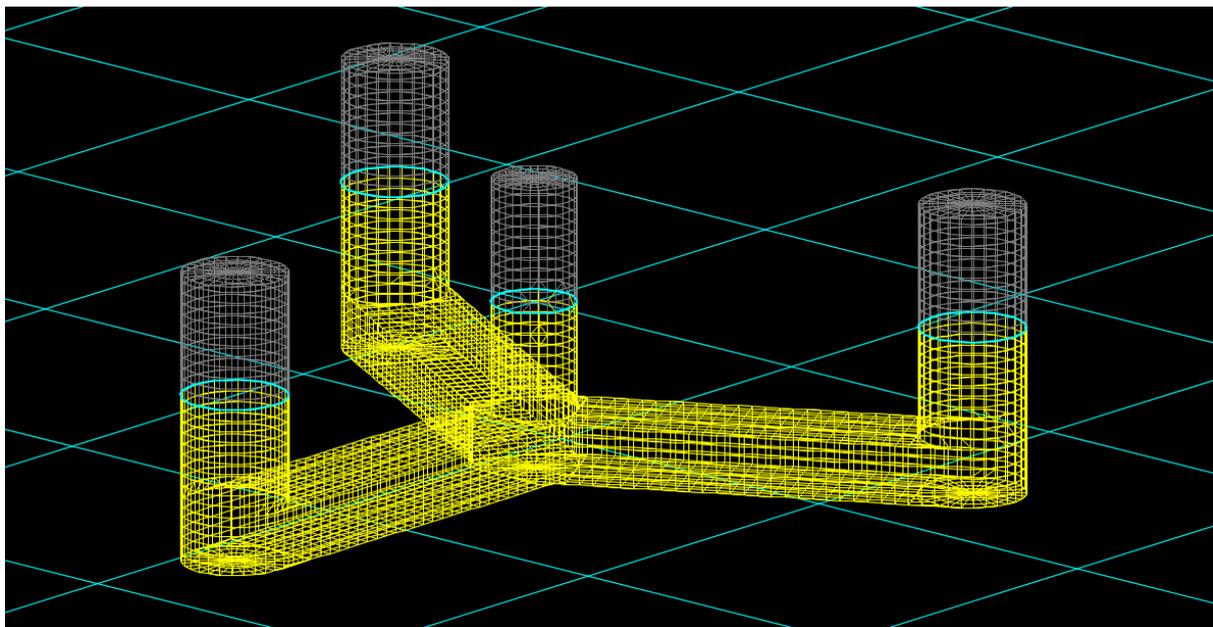
## Semi-submersible platform modelling

For this example, we have used OrcaWave to perform a diffraction analysis for the UMaine VoltturnUS-S semi-submersible. The steel platform design comprises three 12.5m-diameter outer columns connected to a 10m-diameter central column via three rectangular pontoons.

The OrcaWave results file (.owr) and the considered panel mesh file (.gdf) are included amongst the files accompanying this document.

The platform properties, considered as input to the diffraction analysis, are listed in Table 1. Those values account for the mass, centre of mass (CoM) & inertia of the hull's steel structure, the tower interface and fixed & fluid ballast; all of which are specified in the [Definition of the UMaine VoltturnUS-S Reference Platform](#) technical report.

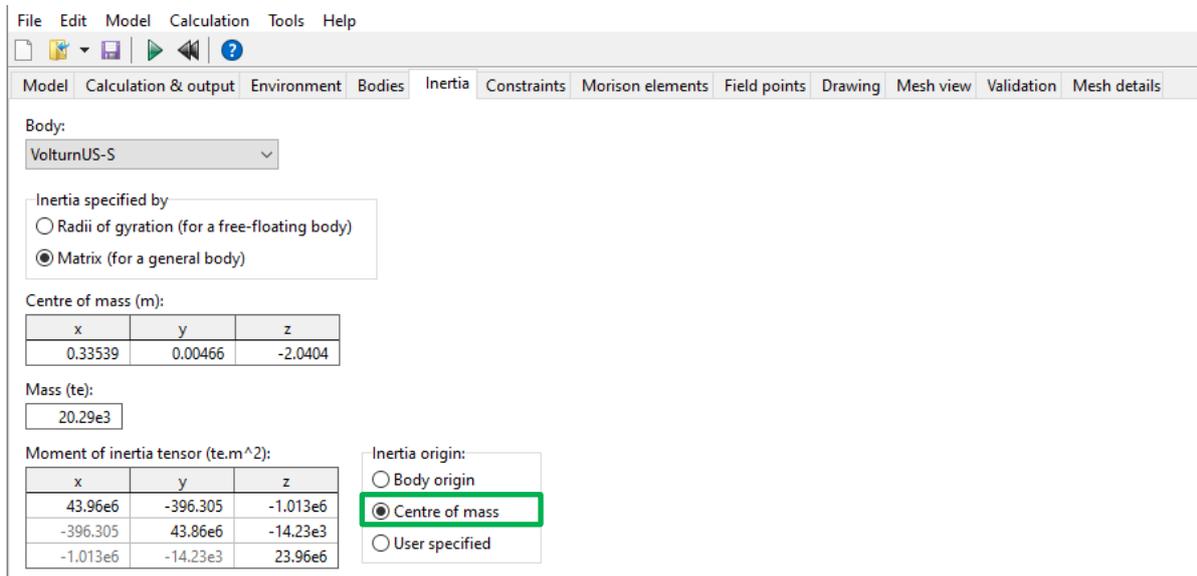
The diffraction analysis also considers the presence of the superstructure: the rotor, nacelle and tower. We have estimated the mass and inertia for those components from the corresponding OrcaFlex model objects using the [compound object properties](#) reporting feature. Those properties are listed in Table 1 along with the overall mass, CoM & inertia properties of the entire structure (platform + superstructure) considered for the OrcaWave analysis.



Item	Mass (te)	Centre of mass (CoM) relative to origin (0,0,0) (m)			Mass moments of inertia (te.m <sup>2</sup> )		
		x	y	z	lxx	lyy	lzz
Platform	17854.00	0.000	0.000	-14.940	1.251E+07	1.251E+07	2.367E+07
Tower	1483.41	0.000	0.000	56.528	1.380E+06	1.380E+06	3.650E+04
Nacelle	675.175	4.528	0.140	148.484	9.913E+03	1.086E+04	1.036E+04
Rotor (inc hub)	274.605	13.644	0.000	150.170	1.751E+05	1.751E+05	3.468E+05
<b>Total</b>	<b>20287.19</b>	<b>0.3354</b>	<b>0.0047</b>	<b>-2.040</b>	<b>4.396E+07</b>	<b>-396.305</b>	<b>-1.013E+06</b>
					<b>-396.305</b>	<b>4.386E+07</b>	<b>-14228.244</b>
					<b>-1.013E+06</b>	<b>-14228.244</b>	<b>2.396E+07</b>

**Table 1: Component properties**

The inertia matrix, shown by the red box in Table 1, is given with respect to the centre of mass. Consequently, the *inertia origin* has been set accordingly on the *Inertia* page in OrcaWave.



File Edit Model Calculation Tools Help

Model Calculation & output Environment Bodies **Inertia** Constraints Morison elements Field points Drawing Mesh view Validation Mesh details

Body: VolturnUS-S

Inertia specified by

Radii of gyration (for a free-floating body)

Matrix (for a general body)

Centre of mass (m):

x	y	z
0.33539	0.00466	-2.0404

Mass (te): 20.29e3

Moment of inertia tensor (te.m<sup>2</sup>):

x	y	z
43.96e6	-396.305	-1.013e6
-396.305	43.86e6	-14.23e3
-1.013e6	-14.23e3	23.96e6

Inertia origin:

Body origin

Centre of mass

User specified

For this example, we have chosen to consider the control surface method for the quadratic load calculations (QTFs). This method requires an additional mesh (*L03 Control surface mesh.gdf*), which has been imported via the *Bodies* page of OrcaWave.

As the mooring specification is known, a mooring stiffness matrix can be applied in OrcaWave to ensure that the displacement RAOs and QTFs for the platform are calculated correctly. This can be achieved by specifying an *external stiffness matrix* on the *constraints* page of OrcaWave.

Here, the matrix entries have been determined by a separate [vessel mooring stiffness calculation](#) in OrcaFlex. Further details about this process can be found in our [L02 OC4 Semi-sub](#) diffraction example.

**Note:** The OrcaFlex vessel mooring stiffness calculation accounts for a static trim of ~1.4 deg, experienced by the platform due to the presence of the superstructure. This is considered to have little influence on the platform motions, for this particular example, so the same static trim has not been accounted for in the subsequent OrcaWave analysis.

### Inertia compensation

After performing the required OrcaWave analyses, the results file was imported to OrcaFlex to create the vessel object and the vessel type.

Importantly, the OrcaWave analysis accounts for the mass and inertia of the superstructure: rotor, nacelle and tower. As we have modelled these items explicitly in OrcaFlex, we need to make sure the superstructure mass and inertia are not double counted.

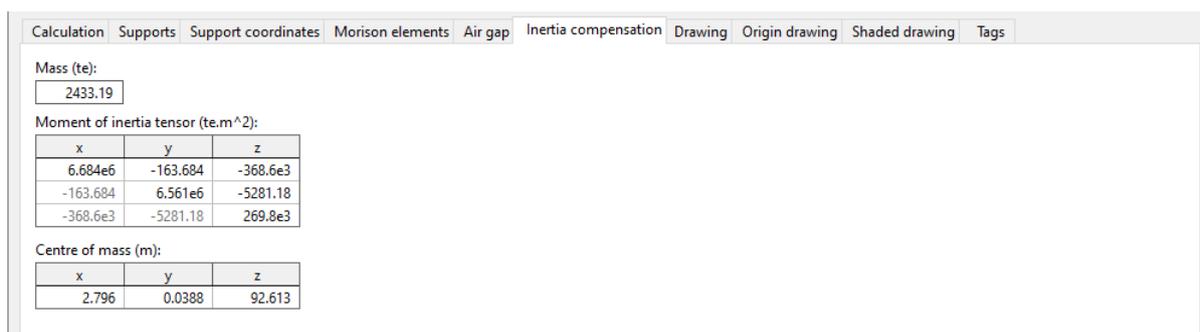
To achieve this, the *mass*, *moment of inertia tensor* and *centre of mass* of the overall superstructure – summarised in Table 2 – have been added to the *inertia compensation* page of the vessel data form in OrcaFlex. Again, we have calculated the data values from the corresponding OrcaFlex model using the [compound object properties](#) reporting feature.

Including this data ensures that the required mass and inertia of the superstructure are automatically subtracted from the vessel type (*VolturnUS-S*) assigned to the vessel. The compensation data also ensures that the correct hydrostatic stiffness is applied to the vessel.

For further details about these adjustments, please refer to our [L02 OC4 Semi-sub](#) diffraction example. Further information about the vessel inertia compensation feature can be found on the following pages of the OrcaFlex help: [Modelling, data and results | Vessels | Vessel data | Inertia compensation](#) and [Theory | Vessel theory | Inertia compensation](#).

Item	Mass (te)	Centre of mass (CoM) relative to origin (0,0,0) (m)			Mass moments of inertia (te.m <sup>2</sup> )		
		x	y	z	Ixx	Iyy	Izz
Tower	1483.41	0.000	0.000	56.5284	1.380E+06	1.380E+06	3.650E+04
Nacelle	675.175	4.528	0.140	148.484	9.913E+03	1.086E+04	1.036E+04
Rotor (including hub)	274.605	13.644	0.000	150.170	1.751E+05	1.751E+05	3.468E+05
<b>Total</b>	<b>2433.19</b>	2.796	0.0388	92.613	6.684E+06	-163.684	-3.686E+05
					-163.684	6.561E+06	-5281.179
					-3.686E+05	-5281.179	2.698E+05

**Table 2: Inertia compensation properties**



Calculation Supports Support coordinates Morison elements Air gap **Inertia compensation** Drawing Origin drawing Shaded drawing Tags

Mass (te):  
2433.19

Moment of inertia tensor (te.m<sup>2</sup>):

x	y	z
6.684e6	-163.684	-368.6e3
-163.684	6.561e6	-5281.18
-368.6e3	-5281.18	269.8e3

Centre of mass (m):

x	y	z
2.796	0.0388	92.613

### Quadratic drag

For a FOWT platform of this nature, flow-separation-induced drag is considered to make a significant contribution to the platform's total hydrodynamic damping. This drag contribution is not accounted for as part of the accompanying OrcaWave analysis, so we have added some *quadratic damping coefficients*, to the vessel type, on the *other damping* page (pictured below).

Here, we have used the diagonal entries given in Table 5 of the platform [technical report](#).

On the *calculation* page of the vessel data form, the vessel *primary motion is treated as* a mixture of 'both low and wave frequency' so we have set the *calculated from primary* option to 'Total motion' on the *other damping* page for the vessel type. This means that the quadratic damping coefficients will be applied to vessel motions arising from both wave frequency and low frequency (slow drift) effects.

Furthermore, to ensure these effects are calculated, we have ticked the 'Other damping' check box for the vessel *included effects* (on the *calculation* page of the vessel data form).

Reference origin (m):

x	y	z
0.000	0.000	0.000

Calculated from primary

- Wave frequency motion
- Low frequency motion
- Total motion

Frame of reference

- Matching calculation mode
- Primary low frequency heading
- Diffraction

Linear damping coefficients:

Surge kN/(m/s)	Sway kN/(m/s)	Heave kN/(m/s)	Roll (kN.m)/(rad/s)	Pitch (kN.m)/(rad/s)	Yaw (kN.m)/(rad/s)
0.000	0.000	0.000	0.000	0.000	0.000

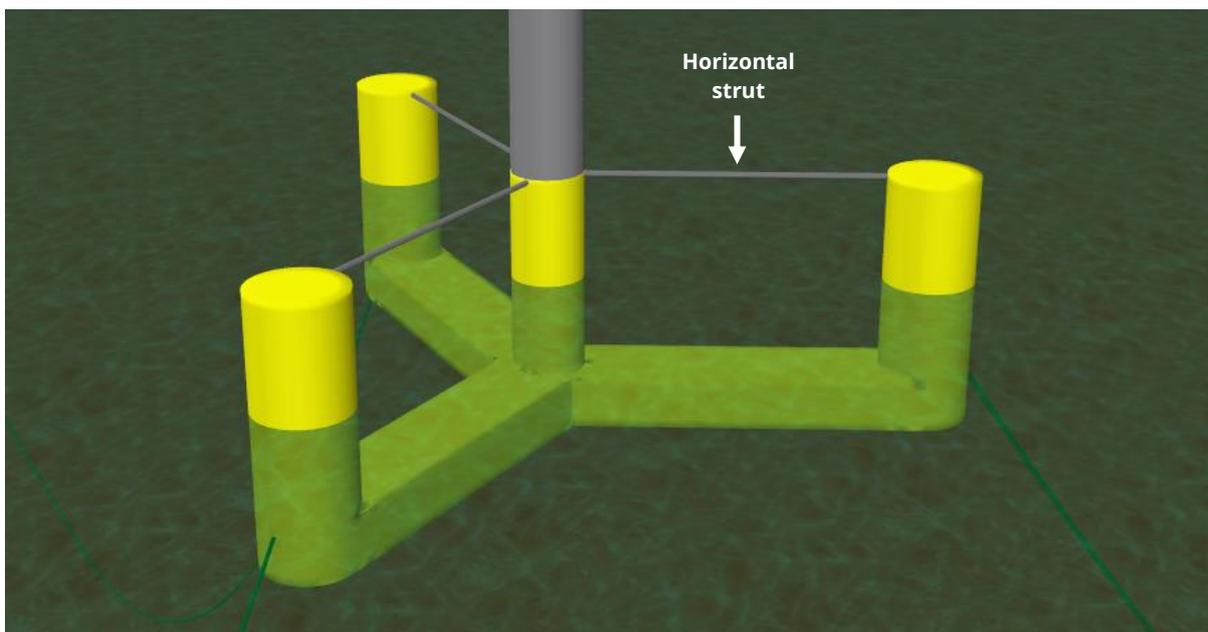
Quadratic damping coefficients:

Surge kN/(m/s) <sup>2</sup>	Sway kN/(m/s) <sup>2</sup>	Heave kN/(m/s) <sup>2</sup>	Roll (kN.m)/(rad/s) <sup>2</sup>	Pitch (kN.m)/(rad/s) <sup>2</sup>	Yaw (kN.m)/(rad/s) <sup>2</sup>
922.500	922.500	2296.00	16.76e6	16.76e6	47.98e6

### Horizontal struts

Lastly, the platform struts – i.e the three horizontal members at the top of the platform structure – are modelled using a series of *spar buoy* type 6D buoys. Their overall length is 40.5m so we have split each 6D buoy into 5 x 8.1m long cylinders. The necessary drag properties have been assigned to each buoy to ensure any aerodynamic drag, acting on the struts, is captured.

For the considered *wind direction* (0 deg), the first strut positioned upwind of the turbine is assumed to be shielded by the platform column. As OrcaFlex does not automatically account for such shielding, the drag coefficients for *Strut1* have been set to zero. We have also assigned zero mass and inertia to each strut because those properties are assumed to be built into the structure represented by the OrcaFlex vessel.



## Environment

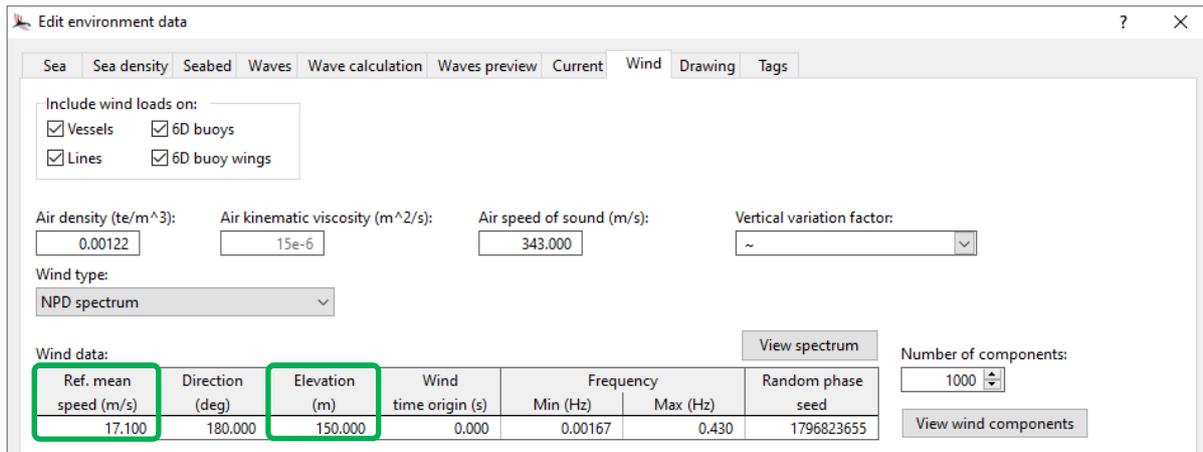
Irregular (JONSWAP) waves, with a *significant wave height* ( $H_s$ ) of 4m and *spectral peak wave period* ( $T_p$ ) of 12s are considered in the simulation environment.

We have chosen to apply the wind using the NPD spectrum, where the wind speed varies randomly over time. The NPD spectrum, as well as the ESDU and API spectra, require the *ref. mean speed* – which is the **1-hour** mean speed at an elevation of **10m** (32.8 ft) above the mean water level (MWL) – and the *elevation* above MWL at which the wind speed is to be calculated. Here we have set the *elevation* to the hub height (150m) which would typically be of interest for a wind turbine model like this.

From those data items, OrcaFlex will calculate the *mean* speed at the given *elevation* before parameterising the spectrum that determines the statistical variation about that mean.

The calculated mean wind speed is assumed to be the same at *all* heights above the MWL unless a vertical wind speed profile is specified using *vertical variation factors*. This applies to all wind types, except for *full field* wind, where the vertical variation in the wind velocity is specified directly in an external full field wind file.

Further information about wind modelling in OrcaFlex can be found on the [Modelling, data and results | Environment | Wind data](#) help page. For further details about *full field* wind modelling, please refer to our [K02 10MW fixed-bottom OWT](#) example.



Include wind loads on:

- Vessels
- 6D buoys
- Lines
- 6D buoy wings

Air density (te/m<sup>3</sup>): 0.00122    Air kinematic viscosity (m<sup>2</sup>/s): 15e-6    Air speed of sound (m/s): 343.000    Vertical variation factor: ~

Wind type: NPD spectrum

Wind data:

Ref. mean speed (m/s)	Direction (deg)	Elevation (m)	Wind time origin (s)	Frequency		Random phase seed
				Min (Hz)	Max (Hz)	
17.100	180.000	150.000	0.000	0.00167	0.430	1796823655

Number of components: 1000

Buttons: View spectrum, View wind components

## Results

### Rotor response

Opening the workspace named *K03 Rotor response.wrk* provides some visibility on the aerodynamic response from the rotor over the first 50s of stage 1.

The right-hand graphs display time histories of the *rotor aero Cp* and *...Ct* results, which represent the aerodynamic power and thrust coefficients, respectively.

The bottom left-hand graph displays a time history of the *rotor aero Lz force*. This represents the component in the turbine's local z-axis direction (Lz) of the net aerodynamic load acting on the rotor (*excluding* structural inertia loads).

As the turbine z-axis direction corresponds to the turbine's axial direction, this result can help to provide a measure of the thrust resulting from the aerodynamic loads applied to the rotor.

Note, an important consideration related to the available *rotor aero force* results is that OrcaFlex calculates the net loading acting at the turbine frame due to the aerodynamic loading distributed over all blade segments. As part of these results calculations, OrcaFlex assumes the rotor is instantaneously a rigid body. So, if the blades are flexible – as they are here – the aerodynamic force felt by the rotor is not exactly equivalent to the *rotor aero force*.

Full details about the available turbine results can be found on the [Modelling, data and results | Turbines | Turbine results](#) page of the OrcaFlex help.

### Blade response

Loading the workspace file named *K03 Blade range graphs.wrk* displays some range graphs results as a function of arc length along blade 1.

The lower graphs show *instantaneous value* range graphs of the *bend moment* and *tension* along the blade. Those show the blade response at the active replay time so, when a replay is playing, it is possible to observe the variation in the blade response, over a chosen arc length range, as the simulation replay progresses.

The upper right-hand graph shows a range graph of *shear force* over *stage 1* of the simulation. The graph shows the minimum, mean and maximum values that the variable has taken over that time.

### Rotor shutdown

For this example, we have enabled the ROSCO shutdown routine by setting `SD_Mode=1` in the 'CONTROLLER\_FLAGS' section of the parameter input file. The shutdown mode is initiated at a user-specified blade pitch angle (`SD_MaxPit`) which, for the purpose of demonstration, we have set to 0.35 rad (20°).

When that blade pitch angle is reached, the controller feathers the blades to the maximum physical pitch limit (`PC_MaxPit` = 1.57 rad (90°)). This allows the wind to flow past the blades, reducing lift, to help slow the rotor down. The speed at which the blades feather is governed by the maximum pitch rate: `PC_MaxRat` = 0.0349 rad/s (2 °/s).

The shutdown behaviour is evident when loading the workspace file named *K03 Rotor shutdown.wrk*. The bottom left-hand graph shows the *blade pitch* (for Blade 1) exceeds 20°, around 65s into stage 1, thus initiating shutdown. The blades subsequently pitch to an angle of 90° and the *main shaft angular velocity* – shown by the top right-hand graph – drops close to zero; indicating that the rotor has slowed to a near stop.

The bottom right-hand graphs show the wind speed at elevations of 0m (MWL) and 150m (hub height). In this case, a *ref. mean wind speed* of 17.1 m/s is considered and the *elevation* above MWL, at which the wind speed is calculated, is 150m.

For the NPD spectrum, OrcaFlex calculates the **mean** wind speed using equation (3) from the [Theory | Environment theory | Wind spectra](#) page of the OrcaFlex help:

$$U_z(z) = U_{ref} \left[ 1 + 0.0573 \sqrt{1 + 0.15 U_{ref} \ln\left(\frac{z}{10}\right)} \right]$$

The graphs show that the calculated mean wind speed (22.11 m/s) is applied in statics and the wind speed then varies randomly with time in dynamics. Note, as no *vertical variation factors* are being considered, the wind speed is the same at all heights above the MWL.

### Simulation warnings

Opening the [view warnings](#) dialog, from the 'Calculation' menu of OrcaFlex, reveals that a couple of warnings have arisen in dynamics.

The first warning states that *"the unsteady aerofoil coefficients have been calculated using a Mach number in excess of 0.3, for at least one blade segment, at some point in the simulation."* This does not cause the calculations to abort but it is an indication that the UA coefficients for some parts of the blade might be invalid, due to limitations in the implemented [González](#) model.

As the warning message suggests, range graph and time histories of *Mach number* can be used to inspect where on a blade this limit has been exceeded. For example, loading the workspace named [K03 BEM and UA results.wrk](#), reveals that the Mach number exceeds 0.3 over the final 6-7m near the tip of blade 1 (shown by the bottom left-hand graph).

The second warning informs us that, at some point in the simulation, the non-dimensional tip speed ratio (TSR) drops below the *lower bound TSR*, specified on the [BEM and UA](#) page of the turbine data form. Generally, the *lower bound TSR* and *upper bound TSR* are used by the [BEM calculations](#) to smoothly scale the induction factors from zero to their full value.



As the warning message suggests, a time history of *induction weight* can be used to identify when this has occurred (shown by the top-right graph). This shows that, as the rotor shuts down midway through the simulation, the TSR (and *induction weight*) drop below a value of 1, and so, a warning is reported. When this happens, OrcaFlex will set the induction factors to zero. This is shown by the bottom right-hand graph of *Axial induction factor* over stage 1.

For further details, please refer to the [Modelling, data and results | Turbines | Turbine data | BEM and UA](#) page of the OrcaFlex help.