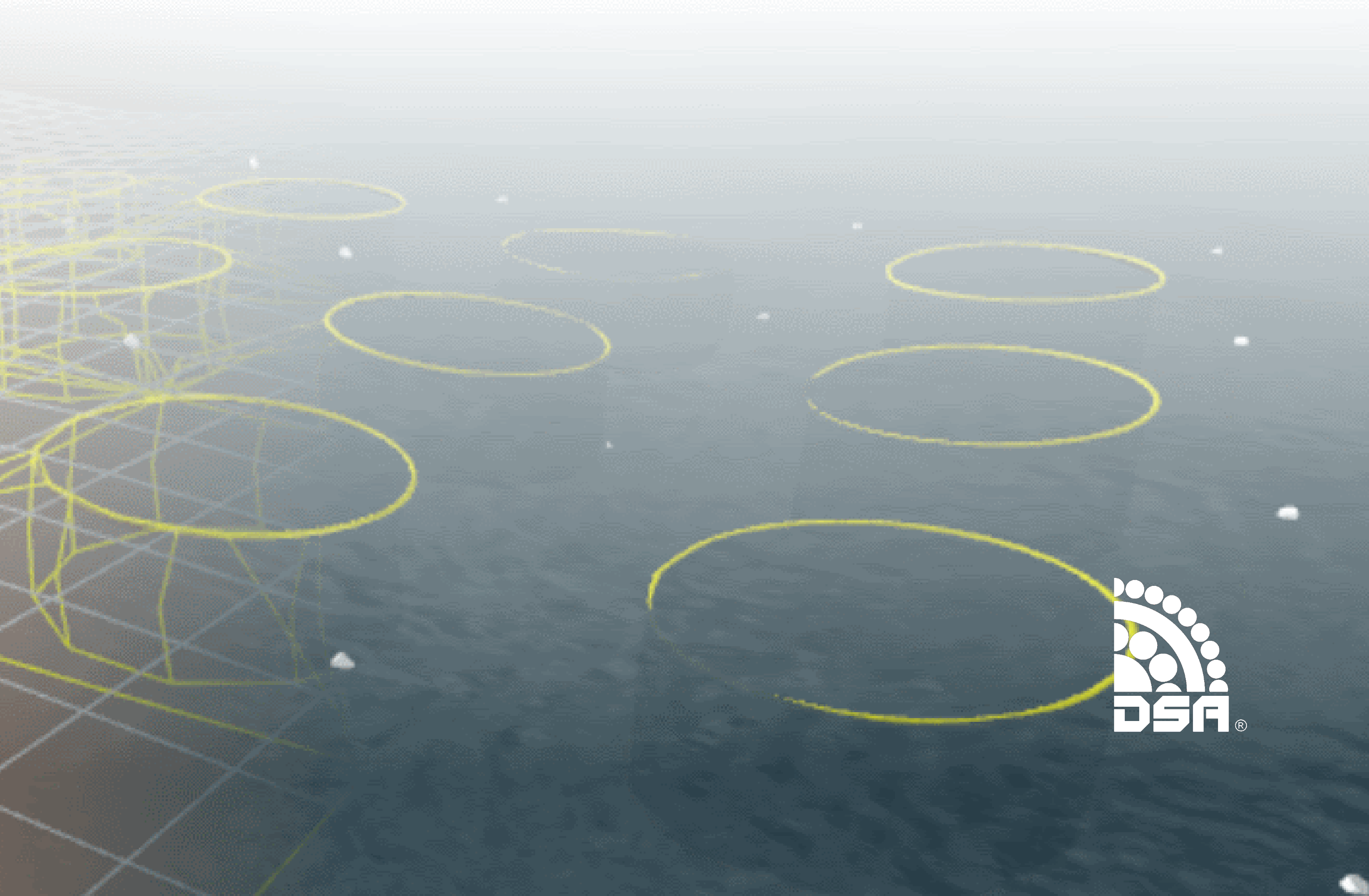
PROTEUS DS

ProteusDS 2015 Tutorials

December 15, 2014



Dynamic Systems Analysis Ltd.
101 - 19 Dallas Road
Victoria, BC, Canada
V8V5A6
phone: +1.250.483.7207

Dynamic Systems Analysis Ltd. (Halifax office)
201 - 3600 Kempt Road
Halifax, NS, Canada
B3K4X8
phone: +1.902.407.3722

ProteusDS support: support@dsa-ltd.ca

Contents

Introduction

This document steps through several tutorials to provide a basic understanding of how to simulate mechanical systems in the ocean with ProteusDS. These tutorials demonstrate the required process to create a dynamic object, or DObject, initialize it with key parameters, create connections between DObjects, and then execute and visualize a simulation.

The tutorials presented herein are designed to be a primer on general ProteusDS usage and are not intended to cover all possible models or capabilities. Users looking for further information on features and capabilities are referred to the ProteusDS User Manual. DSA also offers training and support services to cover more topics and assist with specific needs.

Each tutorial assumes that all prior tutorials have been completed by the user, as some tutorials build upon previous ones. While each tutorial focuses on a different function, they are designed to be easily modified to explore the influence of different parameters and functions available.

Tutorial 1 - Creating a project in PST

1 Tutorial overview

This tutorial covers:

- Creating a simulation project using the ProteusDS Simulation Toolbox (PST)
- Navigating the PST window
- Introducing input files and properties

2 Creating a new project

☞ To begin creating and editing a simulation, a project must be created.

✪ Open ProteusDS by clicking on the ProteusDS Simulation Toolbox (PST) icon from the Windows Start Menu. The ProteusDS core applications are found in the ProteusDS 2015 folder in the Windows Start Menu.

☞ ProteusDS relies on an input folder to contain the files associated with a simulation. A new folder should be created for each new project

☞ When PST is opened, a new project is automatically generated. Using the 'Save Project' button to save in a specific folder location

☞ Multiple copies of PST can be opened to manage different projects simultaneously if desired.

☞ As an alternative, a copy of an existing project can be made using the 'Save Project As' button and navigating to the location that serves as the new simulation project folder.

☞ PST will automatically create all the input files required to run a ProteusDS simulation.

☞ There can be only one project placed in a folder.

☞ Each simulation relies on simulation files in the project folder. Every simulation must have a simulation ('sim.ini'), library ('lib.ini'), and environment ('env.ini') input file.

☞ As new DObjects are added to a project, new *.ini files are created.

3 Navigating the PST window

☞ The upper left side of the PST window is the Project Explorer. This displays all input files in the current simulation.

✪ Click 'Environment' to view the current environment input file ('env.ini').

☞ The right side of the PST window is the Input Panel. This displays the current input file with all the active properties in blue font.

☞ The lower left side of the PST window is the Property Description Panel. This displays the documentation for the property that is currently selected. Information such as: property description, maximum and minimum values, and units are displayed in this window.

☞ Property descriptions appear in the Property Description Panel when the cursor is on the same line as an active property in the Input Panel.

☞ For some input files, the Input Panel moves to the bottom right and the top right side of the PST window is the Section Editor Panel. This displays groups of properties that can be selected, such as Features.

✪ Click 'Library' in the Project Explorer to see an example of this. Features in the library input file will be explained in later tutorials.

4 Input and data files

☞ Every object in a simulation requires a unique input file. The input files contain the properties that describe the object for the simulation.

☞ The default environment and simulation properties have automatically been populated by PST. All active properties have a '$' character before the property name and each property is followed by the property value.

☞ Property values are either numbers or strings. Sometimes vectors or matrices of numbers are required. All numbers can be written as integers or using scientific notation with the 'E' or 'e' character (eg. 1e3 = 1000). Vector and matrix entries are separated by white space.

☞ ProteusDS presently uses only a point '.' as the decimal radix separator between integer and fractional numbers (e.g. 3.14159). In countries where a comma ',' is used (e.g. 3,14159) as the decimal radix separator, within ProteusDS you must use the point convention instead.

☞ Information on the specific parameters can be found in the ProteusDS Manual.

☞ As a simulation executes, the results are written in data files in a results subfolder located within the project folder.

5 Opening the pre-visualizer window

☞ The pre-visualizer window allows the user to view the orientation all the dynamic objects (DObjects), such as cables and rigid bodies, in 3D space before the simulation is run.

☞ Visualizing the simulation will help to eliminate errors in the inital position of the DObjects in the simulation.

✪ Click 'View' on the menu bar at the top of the PST window and then click 'Open Visualizer'.

☞ An empty 3D visualizer window will appear. This window will populate with DObjects as they are added into the simulation. Adding DObjects and navigating the visualizer window will be explained further in the following tutorial.

Tutorial 2 - Running a simulation

1 Tutorial overview

This tutorial covers:

- Setting up the environment with waves
- Resolving follower properties in PST
- Creating a PointMass DObject
- Setting PointMass initial conditions (IC)
- Changing the simulation run time
- Running the simulation

2 Adding waves to an Environment

☞ Create a new PST project. Remember that each new PST project should be created in a new folder.

☞ Dynamic objects (DObjects) model dynamics within the simulation. All DObjects in a simulation are listed in the Project Explorer in PST.

☞ Most DObjects have a state that evolves through simulation. An initial condition (IC) is required to specify the state at the start of a simulation.

☞ The PointMass state is position and velocity. The IC is the position and velocity relative to the global reference frame.

☞ The global reference frame is a right hand system and is fixed at the mean sea surface of 0m. The principle axis directions can be seen in Figure 1.

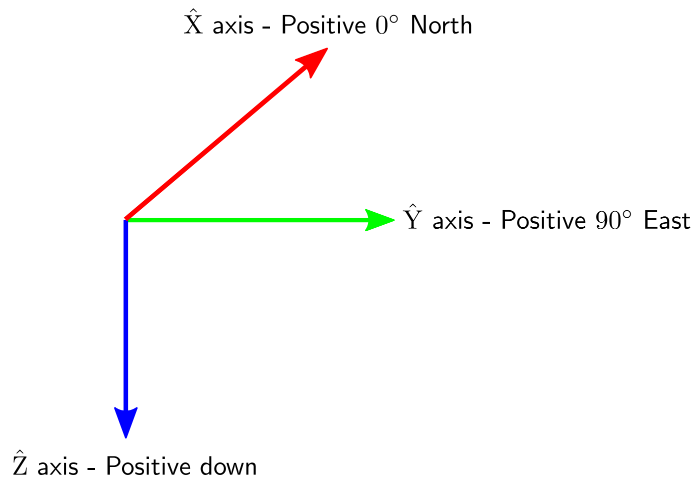


Figure 1: Global coordinate system

- ☞ ProteusDS also has the environment input file, the simulation input file, and the library input file. All these files are listed in the Project Explorer in PST. Connections between DObject are also listed.
- ✪ Click on 'Environment' in the Project Explorer.
- ✪ The property declarations shown below show how to set an Airy wave profile with 2.5m (trough to crest) wave height, 5s period, and with a heading of 0deg. Set these properties in the environment input file.

Airy wave profile setup

```
// Wave
$WaveType 1
$WaveHeading 0
$WaveHeight 2.5
$WavePeriod 5
$TRamp 2
```

- ☞ '$Wavetype 1' is used to specify a single Airy wave.
- ☞ '$WaveHeight 2.5' is used to set the wave height to 2.5m.
- ☞ '$WaveHeading 0' is used to set the wave heading to 0deg (along the global reference frame X axis or North).
- ☞ '$WavePeriod 5' is used to set the wave period as 5s.
- ☞ $TRamp specifies the time over which environmental effects are ramped up to specified values at the start of the simulation. This eliminates any large artificial discontinuities that may occur at the start of the simulation due sudden changes to the environment.

3 Resolving follower properties in PST

- ☞ Certain properties in ProteusDS are master properties and require follower properties to be initialized. This can be seen with the $WaveType property or the $CurrentProfile property.
- ☞ PST has the ability to input all the required follower properties for a specific master property.
- ☞ The Property Description Panel in the PST window will specify whether a property is a master property or not. Master properties will display '[Master]' to the right of the property name.

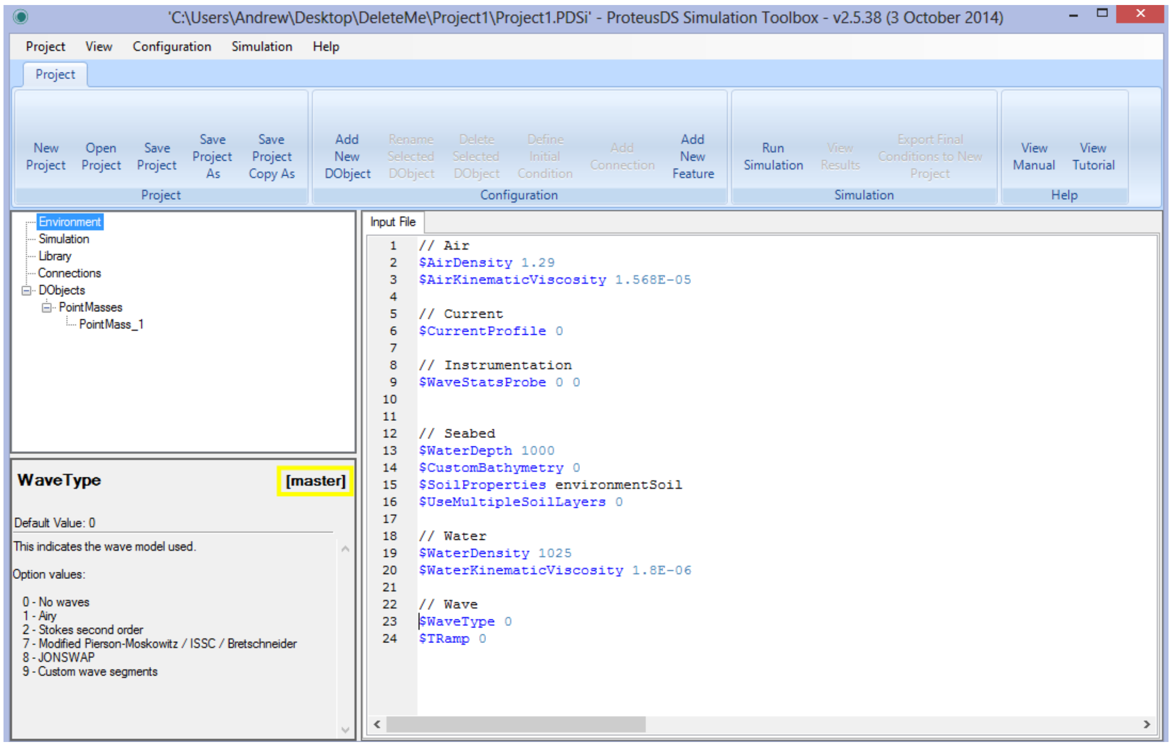


Figure 2: PST window displaying master property

- ☞ To insert all the required follower properties, right click on the master property in the input panel and select 'Resolve All Required Follower Properties', or use the hotkey, 'ctrl-r'. The required follower properties will initialize with their default values.
- ✪ Change the $CurrentProfile property to 1 and resolve all the follower properties. $CurrentSpeed and $CurrentHeading should initialize.
- ✪ Revert the $CurrentProfile back to it's default value (no current) by right clicking on the property and selecting 'Revert Property to Default Value'. The property will change to '$CurrentProfile 0'.
- ✪ When a master property is changed, the follower properties must either be added or removed. Right click on the $CurrentProfile property and select 'Resolve All Required Follower Properties'. The follower properties are no longer required and they will automatically be commented out so that they do not interfere with the simulation.

4 Creating a PointMass

- ✪ Create a new DObject by clicking the 'Add New DObject' button under the configuration tab.

✪ Select 'PointMass' from the drop down menu and provide the DObject with a name.

☞ The new dynamic object has been created and its input file added to the list of DObjects in the Project Explorer. The required properties have been populated with default values.

☞ A PointMass uses a reference sphere of uniform density to compute mass, weight, buoyancy, and hydrodynamic loads. It has 3 degrees of freedom and moves in the global X, Y, and Z directions. It has no rotational state.

☞ The PointMass hydrodynamic loads are governed by a drag and added mass coefficients and the diameter specified.

✪ Leave the PointMass properties at default settings.

5 Viewing the PointMass in the 3D Visualizer

☞ When a new PointMass DObject is created, it is placed by default at the simulation origin (global coordinates (0,0,0)).

✪ Navigate to the visualizer window. The camera is centered on the global reference frame initially but can be freely moved around the environment.

✪ Click and hold down the right mouse button on the visualization window and move the mouse to pan the camera.

✪ Click and hold down the mouse scroll button on the visualization window and move the mouse to rotate the camera.

✪ Use the scroll wheel to zoom in and out. Once zoomed in, the PointMass DObject can be seen.

✪ To reset the camera view to be centered on the PointMass, go to the PST window and select the PointMass DObject from the DObject tree on the left side of the screen. Once the DObject is selected, click the target button on the top left of the visualizer window. The camera will move to center on the PointMass Dobject. This functionality can be used to quickly navigate large simulations in the 3D window.

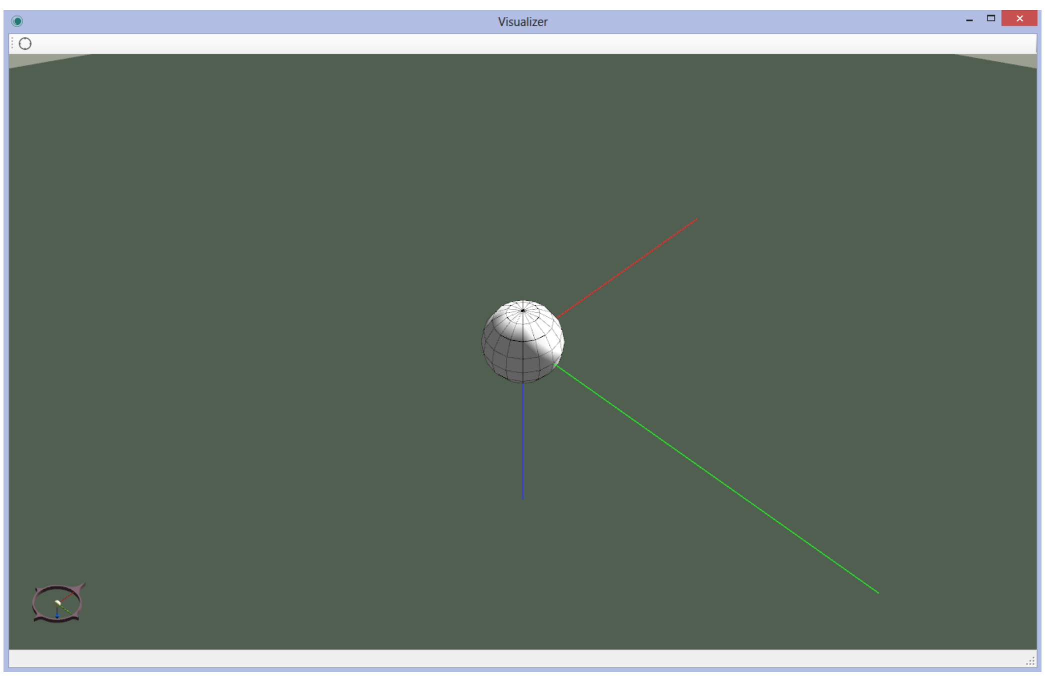


Figure 3: PointMass DObject seen in the visualizer window

6 Setting point mass initial conditions

☞ Remember the IC file specifies the state of a DObject at the start of a simulation and the IC of a PointMass is position and velocity relative to the global reference frame.

✪ Define the IC for the PointMass by clicking on the name of the PointMass in the Project Explorer and then clicking on the 'Define Initial Conditions' button under the configuration tab.

✪ The 'Define Initial Condition' window will appear that will allow you to choose the starting position and velocity of the PointMass.

✪ Choose a location of (0,0,0)m in the global X,Y,Z frame and choose a velocity of (0,0,0)m/s. This will place the PointMass on the global origin with an initial velocity of 0 in all directions.

✪ Click 'Generate' to close the window. The data file corresponding to the initial condition data can be seen in the 'Data File' tab of the PointMass input file. The format for the file can be seen below.

PointMass input file format

```
<state>
0.0 // X velocity (m/s)
0.0 // X position (m)
0.0 // Y velocity (m/s)
0.0 // Y position (m)
0.0 // Z velocity (m/s)
0.0 // Z position (m)
</state>
```

7 Setting the simulation run time

✪ To adjust properties associated with the simulation execution select 'Simulator' in the Project Explorer.

☞ In the simulation input file, the simulation start time, end time, and simulation data interval output is defined.

✪ Change the end time to 10 seconds by setting '$EndTime 10'. The simulation will now simulate 10 seconds.

✪ Leave all other values at default settings.

8 Running the simulation using the ProteusDS Solver

☞ The ProteusDS Solver reads the input files created in PST and processes them to produce the simulation results.

✪ In order to simulate the latest changes in PST, the project must be saved. Select 'Save Project' or use CTRL+S.

✪ Launch the solver by clicking the 'Run Immediately' button in the Simulation tab.

☞ The solver console window will appear on the screen. The console will write a line whenever the simulation output is written. In this case, it is every 0.1s of simulation as set in the simulation input file.

☞ The console will also print a simulation update every 10s of real time or wall clock time of simulation.

☞ When the console window is visible, the simulation can be interrupted with a simulation menu by pressing the spacebar. The simulation can be prematurely terminated or continued as needed.

☞ When the ProteusDS solver has finished, it will automatically close. The simulation is now complete and can be visualized using the PostPDS software. Visualizing the simulation results are discussed in the next tutorial.

Tutorial 3 - Visualizing a simulation and PostPDS basics

1 Tutorial overview

This tutorial covers:

- Navigating the PostPDS visualization tool window
- Selecting DObjects in PostPDS
- Using the time slider and viewing the simulation in PostPDS

2 Opening simulation data in PostPDS

✪ In PST, click the button 'View Results in PostPDS'.

✪ As an alternative, use Windows Explorer to navigate to the 'Results' subfolder of your project and double click on the 'Results.pdso' file contained within.

☞ The PostPDS window appears.

✪ The Specify Load Options window will automatically appear that allows the user to specify the range of simulated time to load, minimum visual cable diameter, and other properties. Leave the default options and click 'OK' to continue.

☞ The simulation will load and the visualization window will show the simulation at the earliest time point requested.

3 Navigating the visualization window

☞ The camera is centered on the global reference frame initially but can be freely moved around the environment.

✪ Click and hold down the right mouse button on the visualization window and move the mouse to pan the camera.

✪ Click and hold down the mouse scroll button on the visualization window and move the mouse to rotate the camera.

✪ Use the scroll wheel to zoom in and out.

✪ The heading, pitch, and zoom of the camera can be seen in the top left of the Home tab. These values will change by rotating and zooming the camera. These values can also be manually changed.

☞ The camera will rotate around and zoom in to a center of rotation location, which can be seen under the Home tab.

☞ In order to change the center of rotation, pan the camera with the right mouse button or use the keyboard keys W, A, S, and D to move the center of rotation forward, left, backwards, and right. Values for a center of rotation can also be changed manually.

4 DObject and Environment lists

☞ To get a better view of a specific DObject in the simulation, it may be necessary to rotate or zoom in on the DObject of interest.

☞ Any DObject in the simulation can be selected from the object tree on the left side panel.

✪ Click the arrow next to 'Default' to expand the DObject list. Double click on a DObject to move the camera's center of rotation onto that DObject frame. If the DObject is a cable, the camera will center onto Node 0.

✪ Right click on a DObject name show options for that particular object such as opening a text file containing the state vector or change the appearance of DObject in the visualizer.

✪ Toggle the checkbox next to a specific DObject name in the DObject list to hide or show it in the visualization window.

✪ Toggle the checkbox next to the water or sea floor in the Environment list to hide or show them in the visualization window.

5 Simulation playback

✪ Begin simulation playback visualization by clicking on the play icon at the top of the screen under the Home tab.

☞ As the simulation plays, the time slider will progress along the bottom of the screen in increments of $IntervalOutput. PostPDS attempts to playback the simulation results as close to real time as possible.

✪ Click and drag the slider to manually select any point in time in the simulation history.

☞ The buttons on either side of play move the simulation playback one data point in either direction in time. The lower buttons in the playback section move the simulation time to the start or the end.

☞ The circular button on the right of the playback section loops playback.

Tutorial 4 - Simulating a pendulum

1 Tutorial overview

This tutorial covers:

- Creating a Cable DObject
- Generating a Cable IC
- Adding a features to a Cable

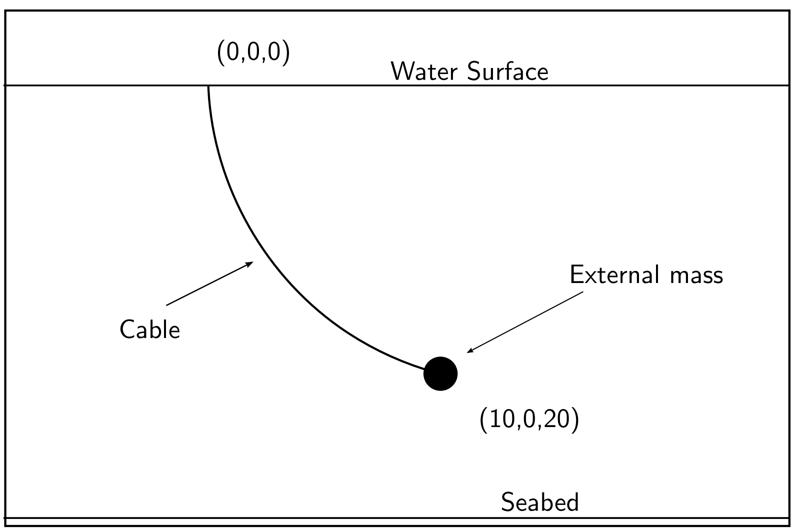


Figure 4: Pendulum layout

2 Creating a Cable

✪ Create a new project.

☞ New projects start with a default environment and numerical integrator feature called *EnvironmentSoil* and *RK45*, respectively.

✪ Add a new Cable DObject and use the default name *Cable_1*.

✪ By design, node 0 is the top of the pendulum. The top node of the cable must be pinned in place. This is done by setting '$Node0Static 1' in the Cable input file.

☞ A pinned cable node can freely rotate about the pinned location but will not move. As an alternative, the boundary of the cable may be clamped. If clamped, the principle vectors P1 and P2 must be defined, which define the plane of the cross section at the boundary, which the tangent vector is perpendicular to. More information regarding pinned and clamped constraints can be found in the ProteusDS Manual.

3 Setting Cable IC

✪ Define the Cable IC by selecting *Cable_1* in the Project Explorer and clicking 'Define Initial Condition'.

✪ Specify start point (node 0) of (0,0,0) and end point (node N) of (10,0,20).

✪ Change the number of elements to 10 and increase the length of the cable to 23m.

☞ As a reference, the straight line distance between node 0 and node N is computed automatically.

☞ Care must be taken when choosing the number of elements. If elements are too long, the Cable may not be accurately represented. If they are too short, the simulation can take significantly longer to execute.

✪ Once the initial conditions are set, view the cable in the 3D visualizer window to ensure that the cable is in the correct location. The simulation origin (0,0,0) is marked by the global coordinate axes.

4 Defining Cable material properties

✪ To define the Cable's physical properties, a library feature must be created and referenced. Click 'Add New Library Feature'.

☞ The properties listed represent generic 2 inch wire rope.

✪ Create a DCableSegment feature called *wire_rope_2in*.

✪ Change the properties to the following values:

Cable feature properties

```
// Fluid loading
$CDc 1.5
$CDt 0.01
$CAc 1

// Mechanical
$EA 2e8
$EI1 1e3
$EI2 1e3
$GJ 1e3
$Diameter 0.05
$Density 5000
$CID 1e4
$BCID 0
$TCID 0
$CE 0
```

☞ Each cable feature property definitions can be found in the Property Description Panel when the property is selected and in the ProteusDS manual.

✪ Reference the new wire rope feature in the cable input file by adding '$CableSegment wire_rope_2in 23'.

☞ This specifies that the cable is entirely composed of wire rope with a length of 23m.

☞ If only one material is indicated, it is used for the entire Cable length regardless of span indicated.

☞ The 'Segmented Cable' tutorial demonstrates the use of multiple materials in a single Cable.

5 Adding an ExtMass feature

☞ An ExtMass feature acts like a PointMass, but it is lumped in directly with the Cable and not treated as a separate DObject.

✪ Create an ExtMass feature to the library called *Weight*.

✪ To represent a steel sphere with a diameter of 1m, give the ExtMass feature the following properties:

ExtMass feature properties

```
// Fluid loading
$CD 1
$CA 0.5

// Mechanical
$Diameter 1
$Density 7000
```

✪ Add '$ExtMass Weight 23' to the Cable input file.

☞ This applies the lumped mass at the bottom of the Cable. The number following the feature name represents the arcspan along the Cable from node 0 where the mass is located.

☞ The ExtMass is automatically incorporated in the Cable as it a feature, not an external DObject.

6 Running the simulation

✪ Extend the simulation run time to 30s in order to capture the pendulum swing.

✪ Click the 'Run Immediately' button under the simulation tab to execute the simulation.

✪ Post processing using PostPDS will be completed in the next tutorial.

Tutorial 5 - PostPDS and plotting

1 Tutorial overview

This tutorial covers:

- Visualizing simulation results from the pendulum tutorial
- Plotting tension

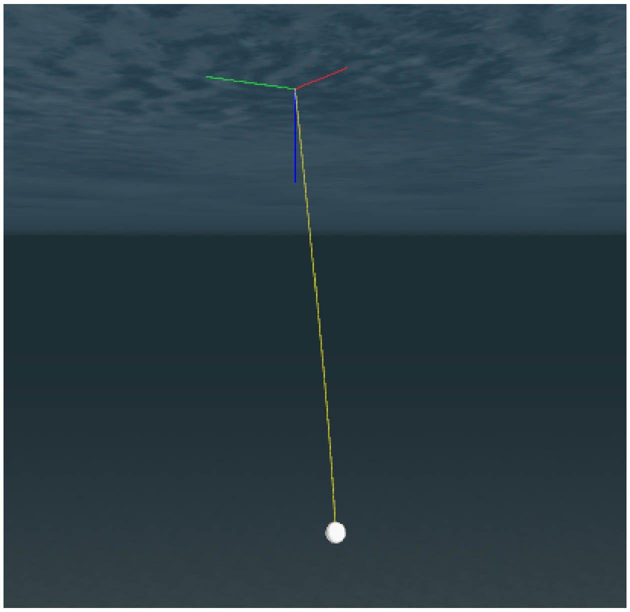


Figure 5: Cable pendulum with ExtMass feature weight

2 Loading results in PostPDS

☞ The data created from the previous pendulum simulation were automatically placed in a subfolder called ‘Results’ in the project directory.

✪ Click on the ‘View Results in PostPDS’ in PST or alternatively double-clicking the ‘Results.PDSo’ file in the pendulum simulation results folder to load simulation results in PostPDS.

✪ Use default options in the PostPDS Specify Load Options window and click ‘OK’ to continue.

✪ Move the camera such that the cable can be seen in the viewing window.

- ✪ Start simulation playback and observe the cable swing under the weight of the ExtMass.
- ☞ Note the minor snap load at the first moment of simulation. The Cable IC used produces a slack line. When it is loaded abruptly with a large weight, a snap load results. However, hydrodynamic damping due to the large diameter of the *Weight* and Cable calms the system, which reduces to pendulum oscillations.
- ☞ The display type can be changed from rendered to engineering view. In engineering view, all cables are rendered with straight lines between finite element nodes and nodes can be seen by selecting 'Show element nodes'. The red and blue nodes indicate node 0 and node N, respectively.
- ☞ Change the colour and opacity of the background water by selecting a new colour and adjusting the 'Density' slider in the 'Colours, below water (and no water)' section under the Settings tab. The same can be done for 'Colours above water'.
- ✪ Double click on 'cable_1' in the DObject tree to center the camera on node 0 of the cable.

3 Plotting data

- ☞ There are several plotting tools embedded in PostPDS that indicate tension or stress in the cable. Tension plots can be generated in a separate window.
- ✪ Right click on the cable object and select 'Plot', 'Tension', and submit element 10 to view the tensions occurring at the bottom of the cable.
- ☞ The oscillations in the tension represent the initial drop of the heavy weight and the resulting settling. The time response of tension and motion depends on many inputs, including the axial rigidity (EA) of the cable.
- ✪ Click 'Plotted Variable' in PostPDS and select 'Tension' from the drop down menu. Click the 'Show legend' check box. Now the cable in the visualization window will be coloured by tension values indicated in the legend.
- ☞ The minimum and maximum tension values used in the legend can be manually changed in their text editor boxes.
- ☞ The tension oscillations seen in the plot can now be seen during simulation playback by the change in Cable colour.

Tutorial 6 - Output files

1 Tutorial overview

This tutorial covers:

- Locating simulation results data
- Simulation results folder structure
- Data file contents created by simulation
- Additional analysis tools

2 Viewing results files

☞ Every simulation will create a 'Results' subfolder in the project directory and fill it with data as the simulation progresses.

☞ The results folder is separated into subfolders representing each DObject, the initialization files used, simulation state vectors, and the terminal state vectors of each DObject.

☞ The DObject results folder contains all the results pertaining to that specific DObject. Each file is an ACSII formatted text file with the extension '*.dat'. There is a header at the top of each output file that describes the format of the information. The first column is always the time stamp and is printed at every output interval set in the simulation input file.

☞ The 'Initial' folder contains the all the files used to run the simulation. By selecting 'Initial.PDSi', PST will open and a new simulation can be executed/edited using the same input files.

☞ The 'Restart' folder contains the state vectors of each DObject at predefined intervals throughout the simulation. These state vectors can be used to start the simulation again from some time point through the simulation.

☞ The 'TerminalIC' folder contains the state vectors of each DObject at the end of the simulation. These files are automatically created when a simulation completes. This gives the option to carry on new simulations using the final state as a starting point.

☞ All the files pertaining to the environment are located directly in the 'Results' folder and are formatted similarly to all the other output files.

3 ASCII *.dat output files

☞ All the results are written in ASCII characters for readability and the ability to import into 3rd party software such as Matlab for further analysis.

4 Experimenting with Results files

✪ Make a copy of the pendulum simulation folder or open the pendulum simulation project in PST and use the 'Save As' button to create a new project.

✪ Increase the diameter of the *Weight* to 1.5m.

✪ Run the simulation.

✪ Open the tension output file for each cable and compare the tensions.

☞ The tensions at the top of the Cable reach a steady state of roughly 30kN and 100kN for the system with 1m and 1.5m diameter weight, respectively. The tension is dominated by the weight.

Tutorial 7 - Pendulum with a PointMass

1 Simulation overview

This tutorial covers:

- Reviewing the pendulum tutorial
- Creating a PointMass DObject
- Connecting a PointMass to a cable

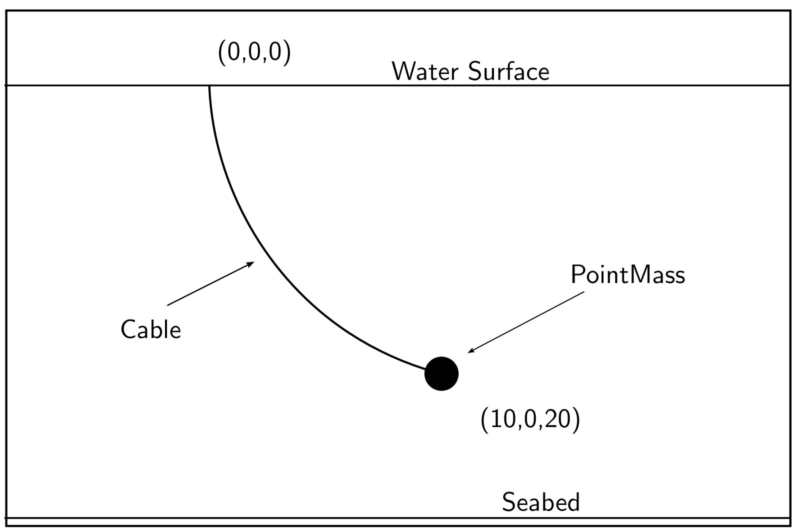


Figure 6: Pendulum with a PointMass

2 Reviewing the pendulum tutorial

✪ Make a copy of the project folder made for the 'Simulating a Pendulum' tutorial project input folder. The copy will be modified for use with this tutorial.

☞ In this tutorial, the weight modelled with an ExtMass in the previous tutorial will be replaced with a PointMass DObject.

✪ Remove the $ExtMass property from the Cable input file. The ExtMass feature can also be deleted from the library, but this is not required.

3 Creating a PointMass

- ✪ Add a PointMass DObject to the simulation. This will act as the weight at the end of the cable.
- ☞ A PointMass behaves the same as an ExtMass. However, an ExtMass is a feature that is added and embedded within each Cable while a PointMass is an independent DObject. Multiple cables can be attached to a PointMass.
- ✪ Set density and diameter of the PointMass to 7000kgm^3 and 1m, respectively, to model a solid steel sphere with a diameter of 1m.
- ✪ Initialize the PointMass with an initial position coincident with Node N of the cable and zero initial velocity.

4 Creating PointMass and Cable connections

- ☞ In order to have two DObjects interact, a connection must be made between them. Connections are listed under 'Connections' in the Project Explorer. Each DObject also lists connections associated with it under the 'Connections' tab next to the 'Input' tab.
- ☞ When connecting the PointMass to the cable, the PointMass and desired Cable node must be near each other to avoid stretching the Cable boundary element too much, which may cause a destabilization.
- ✪ Connect the Cable to the PointMass by selecting the 'Add Connection' button under the configuration tab.
- ☞ A PointMass must be the master DObject because they set the position and velocity of the boundary node of a Cable. In return, the reaction forces from the Cable are passed to the PointMass to complete the dynamic interaction.
- ☞ There are different connection types between two DObjects. In this case, a point connection type is the only connection type between a Cable and PointMass.
- ✪ View the connection properties by selecting the 'Connections' tab under either DObject used in the connection.
- ✪ Enable the $DCableFollowerNodeN property by setting the property to 1. This indicates that node N of the follower Cable is used for the connection.

5 Running the simulation and processing the results

- ✪ Set the length of simulation to 30s
- ✪ Execute the simulation.
- ✪ View the results in PostPDS. The PointMass has the same properties as the ExtMass in the previous tutorial, therefore, the dynamic motion is the same.

Tutorial 8 - Mass-spring-damper and integrator

1 Tutorial overview

This tutorial covers:

- Setting up a mass-spring-damper DObject
- Changing the integrator properties
- The consequences of under-sampling
- The difference between the non-adaptive and adaptive integrator

2 Setting up a Mass-spring-damper

☞ A mass-spring-damper (MSD) is a DObject in ProteusDS that demonstrates a simple oscillating system and has the ability to illustrate the numerical integrator performance.

✪ Create a new project and add a MSD DObject.

✪ The MSD DObject only requires two properties that control the natural frequency and damping of the system. Use the following property declarations in the MSD input file.

MSD properties

```
$NaturalFrequency 10
$DampingRatio 0.1
```

☞ This results in a mass-spring-damper with a natural frequency of 10Hz and a damping ratio of 0.1.

✪ In the simulation input file, set '$IntervalOutput 0.01'. The interval output determines how often the integrator will write results to the output files.

☞ The output interval is crucial in capturing the oscillations that will occur in the simulation. If the output interval is too large, data will be lost. If the output interval is too small, the simulation may slow due to writing too much data.

✪ Generate initial conditions for the MSD using the 'Custom' method. A MSD requires 2 initial values, the initial velocity is listed first and displacement is listed on the next line. Set the initial velocity to 0m/s and the initial displacement to 5m.

✪ The integrator is a feature that can be changed in the library input file but keep the default integrator settings.

✪ Set the simulation to run for 10 seconds.

- ✪ Run the simulation and plot the position vs time data. Notice that all the oscillations are clearly represented with ample data points present in Figure 7.

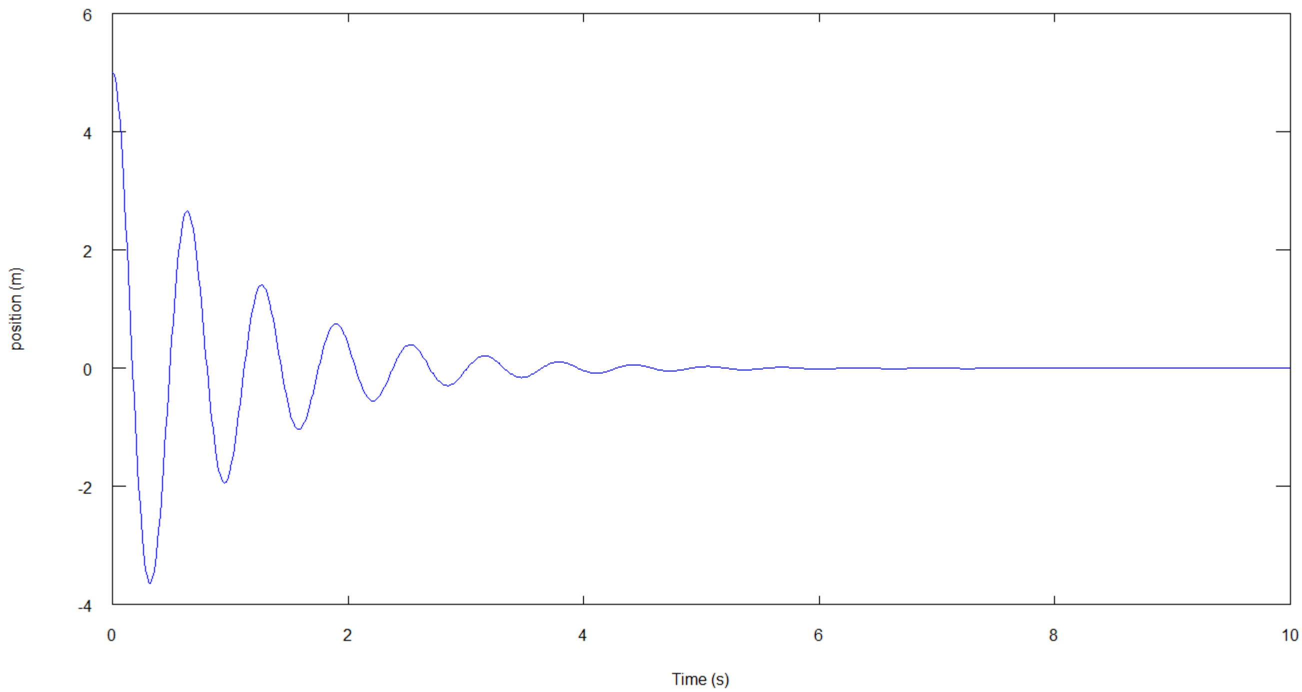


Figure 7: Position vs. Time for Mass-Spring-Damper system

3 Consequences of under-sampling

- ✪ Change the output interval to 0.4 seconds.
- ✪ Rerun the simulation and plot the results.
- ☞ Notice that several oscillations are lost and the results are incorrect in Figure 8. Erroneous data is presented due to undersampling.
- ☞ At a minimum, sampling simulation results should be twice the frequency of interest.
- ☞ In this simulation, the sampling rate is not adequate and information is lost. It is very important to ensure that proper output intervals are set to capture peak motions and loads in the dynamic system response.

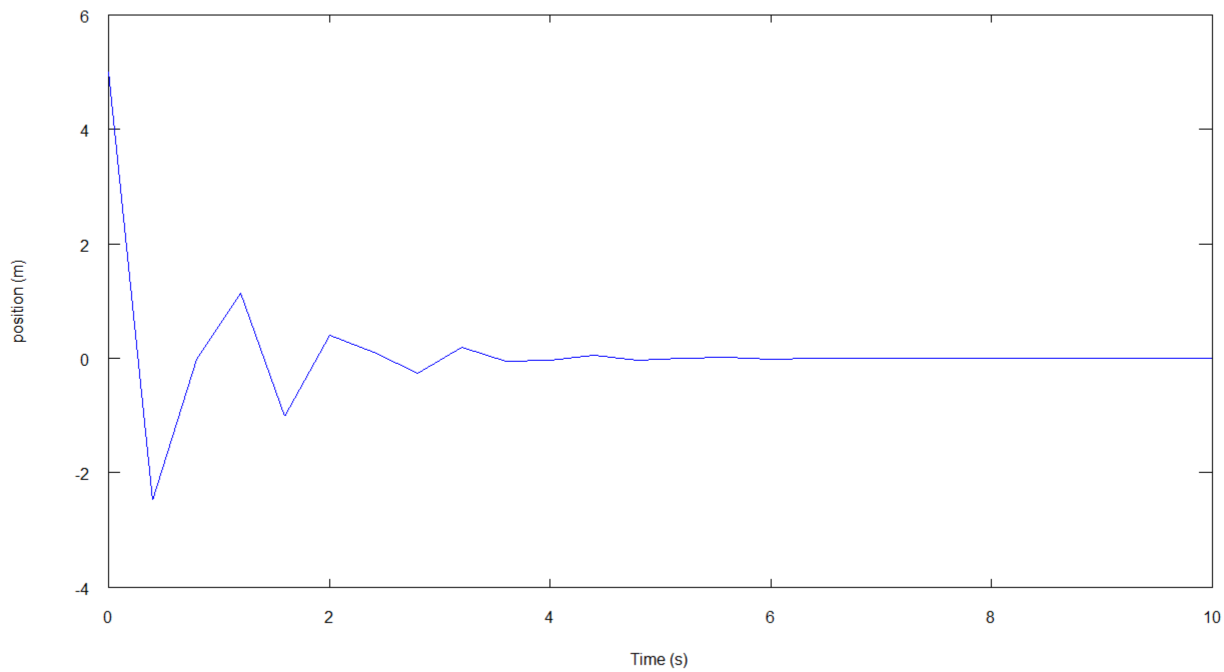


Figure 8: Position vs. Time for Mass-Spring-Damper system with loss of data

4 The difference between a non-adaptive and adaptive integrator

- ProteusDS has the ability to use a non-adaptive (4th Order RungeKutta) and adaptive integrator (4th/5th Order RungeKutta). A non-adaptive integrator runs at a constant time step, whereas the adaptive integrator adjusts the time step dynamically to account for higher frequency oscillations in the simulation.
- Loss of data can occur if the time step of the non-adaptive integrator is too large just as if the output interval is too large.
- In addition, too large of a constant time step may cause destabilization and the simulation to fail.
- If the output interval is smaller than the desired initial/constant time step, ProteusDS will adjust the initial/constant time step to match the output interval.

Tutorial 9 - RigidBody with Cylinder feature

1 Tutorial overview

This tutorial covers:

- Creating a RigidBody DObject
- Setting RigidBody initial conditions
- Creating a RigidBody Cylinder feature

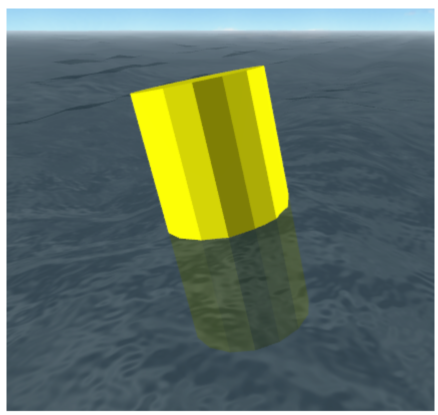


Figure 9: RigidBody DObject floating in waves

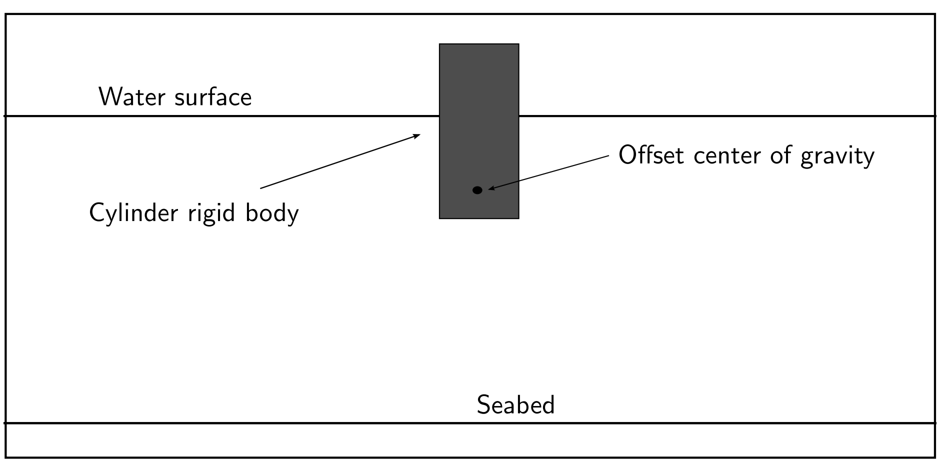


Figure 10: Rigid body with cylinder layout

2 Creating a RigidBody

☞ A RigidBody represents an inelastic structure with 6 degrees of motion freedom. It can respond to loading from current, wave, wind, soil, as well as other DObjects.

✪ Create a new project in PST and add a new RigidBody DObject.

✪ Navigate to the RigidBody input file. The core properties of a RigidBody DObject are the moments of inertia, products of inertia, and mass.

☞ Every RigidBody has its own local RigidBody coordinate frame. The position, orientation, and velocity of this frame at the start of the simulation relative to the global coordinate frame is defined by the initial conditions.

✪ Use the following mass and moments of inertia.

Cylinder mass properties

```
// Mechanical
$Ix 1e5
$Iy 1e5
$Iz 1e5
$Ixy 0
$Ixz 0
$Iyz 0
$Mass 1e5

$CGPosition 0 0 3
```

☞ By design, the RigidBody will be 50% submerged when the cylinder hull is added. The moment of inertia values are set arbitrarily.

☞ The moments and products of inertia can be defined about the RigidBody coordinate frame or about the center of gravity by using the optional flag $DefineInertiaAboutCG. By default, the values are defined about the RigidBody local coordinate frame.

✪ By design, the cylinder hull sits vertically in the water. To keep it stable in this position so it does not fall flat on the water, the center of gravity is set 3m along the Z axis of the RigidBody local frame by specifying: '$CGPosition 0 0 3'.

✪ Add the flag '$DefinteInertiaAboutCG 1'. This indicates that the mass moment of inertia values specified are about a corresponding frame located at the CG location. This will result in an automatic parallel axis computation to determine the total equivalent inertia seen at the RigidBody local frame.

3 Defining the RigidBody initial conditions

☞ Rigid bodies have 6 degrees of freedom: velocity and position in the X, Y, and Z direction as well as roll, pitch, and yaw Euler angles and Euler angle rates that orient it with respect to the global reference frame.

☞ All positions and velocities must be set in the initial conditions file prior the running the simulation.

✪ While viewing the RigidBody input file, click the 'Define Initial Conditions' button. Choose a location at 3,3,-5 and angle orientation of roll, pitch, and yaw (heading) of 45,0,0, respectively, and with all velocities set to 0. This will place the DObject 3 meters from the origin in the X and Y direction and 5m above the water surface. The DObject will also start with a roll angle of 45 degrees.

- ✪ RigidBody IC can also be generated by the custom option in the initial condition window. This allows the user to input values directly into the IC data file. Select 'Define Initial Condition' and select 'Custom' from the generation method drop down box.
- ✪ The custom IC data file follows the format $[V_x, V_y, V_z, \dot{\phi}, \dot{\theta}, \dot{\gamma}, P_x, P_y, P_z, \phi, \theta, \gamma]$. Enter the following values into the custom text window to create the same initial conditions as above.

RigidBody IC file

```
<state>
0
0
0
0
0
0
3
3
-5
45
0
0
</state>
```

- ☞ More information on the RigidBody IC data can be found in the ProteusDS Manual.

4 Creating and adding RigidBody features

- ☞ A RigidBody needs features to give it a physical presence in order to compute environmental loads from currents, waves, wind, and soil.
- ☞ Basic geometric shapes such as cylinders can be easily defined. Alternatively, a custom shaped hull mesh file can be imported. For this tutorial, a cylinder will be used.
- ✪ Add a new RigidBodyCylinder feature in the library by clicking 'Add New Library Feature' in the Configuration tab. Use the name *float*.
- ✪ The cylinder properties are defined in the cylinder library feature under the name *float*. Use a diameter of 5m and length of 10m and leave the remaining parameters at default values.
- ☞ The axial, angular, and radial segments properties are used to generate a polygonal mesh of the cylinder for drag, added mass, and buoyancy loads. More segments in the cylinder results in more polygonal faces and therefore higher accuracy for the calculations performed. However, having too many segments may result in slower simulation speed.
- ☞ The $WindLoading and $HydroLoading properties toggle wind and hydrodynamic forces acting on the feature.
- ☞ The $BuoyancyFroudeKrylov property determines whether Froude-Krylov forces will act on the feature by a surface integral of the undisturbed water pressure field over the wetted mesh surface.

Cylinder feature properties

```
// Fluid loading
$WindLoading 1
$HydroLoading 1
$BuoyancyFroudeKrylov 1
$CdAxial 1
$CdNormal 1
```

```
$CaAxial 1
$CaNormal 1

// Mechanical
$Diameter 5
$Length 10

// Numerical
$AxialSegments 10
$RadialSegments 3
$AngularSegments 8
```

☞ Features and connections are located with a fixed position and orientation with respect to the RigidBody local coordinate frame. Multiple cylinders and custom mesh features can be applied to a RigidBody.

✪ The cylinder feature will be used as the hull for the RigidBody. To add the cylinder to the RigidBody, add to the RigidBody input panel '$Cylinder float 0 0 0 0 0 0'.

☞ These 6 numbers correspond to the location of the cylinder feature with respect to the local RigidBody frame. The first three numbers correspond to distance in the local x, y, and z direction and the second three numbers correspond to the Euler angles about the RigidBody x, y, and z axes that indicate how the cylinder feature frame has rotated away from the RigidBody frame. Since all location values are 0, the cylinder feature and the RigidBody have coincident coordinate frames.

5 Running the simulation and processing results

✪ Initialize an Airy wave profile with a 3s period, 1m wave height, and a heading of 0deg.

✪ Set a uniform current with a speed of 1m/s and a heading of 90 degrees.

✪ Use a $TRamp of 10s.

✪ Set the simulation to run for 30s.

✪ Save the project and click 'Run Simulation'.

✪ Visualize the simulation using PostPDS

✪ Right click on the RigidBody DObject in the object tree and display mesh edges. This will show the polygonal mesh edges on the cylinder.

Tutorial 10 - Use of probes

1 Tutorial overview

This tutorial covers:

- Environmental probes
- DObject probes
- Probe data files in the results folder

2 Using probes to output environment data

☞ Probes are properties in ProteusDS that report specific simulation information in the form of output files. The type and location of the probe is specified in the property declaration.

☞ The two types of probes are environmental probes and DObject probes.

☞ Two environmental probes are available, a water velocity probe and a sea height probe.

☞ The $WaterVelocityProbe property is declared in the environment input file and the property must be followed by the Cartesian coordinates in the global frame of the desired location for the probe.

☞ After running the simulation an output file will appear in the results subfolder called 'WaterVelocityProbe.dat'. This file will contain the current speed and heading at the specified location for each time output during the simulation.

☞ Multiple probes can be used in the simulation by repeating the property declaration with different locations. All probe result data will appear in the same file listed in the order of declaration.

☞ The $SeaHeightProbes property is also declared in the environment input file and the property must also be followed by the Cartesian coordinate of the desired location for the probe in the X and Y direction. No Z coordinate is necessary as this probe reports the water height at a certain point on the X,Y plane.

☞ Similar to the water velocity probe, an output file will appear in the results subfolder called 'seaheight_nxm.dat' where the 'n' and 'm' values specify the X and Y location. This file will contain the sea height at the specified location for each time output during the simulation.

☞ Multiple sea height probes may also be used.

✪ Open the 'Rigid body with cylinder feature' tutorial project or any available project with waves, current, and a RigidBody DObject. Add a $SeaHeightProbe at the location (0,0). Add another $SeaHeightProbe at the location (2,0).

✪ Add a $WaterVelocityProbe at the location (0,0,10).

✪ Run the simulation as done in previous tutorials.

✪ Navigate to the project results subfolder and open the 'WaterVelocityProbe.dat', 'seaheight_0x0.dat' and 'seaheight_2x0.dat' files.

☞ Notice that the current speed at the location specified is reported through the water velocity probe and the sea height changing due to the waves is reported through the sea height probes.

3 Using probes to output DObject data

☞ DObject probes can be used to report information about a specified DObject. The available probes are: $ArcPointProbe for Cable DObjects and $AirGapProbe, $BodyAccelProbe, $GlobalAccelProbe, and $PositionProbe for RigidBody DObjects.

☞ $ArcPointProbe can be used to report the 3D Cartesian location of the point along the cable at the arc length specified.

☞ $AirGapProbe can be used to report the air gap and relative velocity between the point specified in the body coordinate frame and the water surface.

☞ $BodyAccelProbe and $GlobalAccelProbe can be used to report the accelerations of the point specified in either the body coordinate frame or the global coordinate frame.

☞ $PositionProbe can be used to report the position of the point specified in the global coordinate frame.

☞ Similar to the environmental probes, all DObject probes have files written to the DObject subfolder within the results subfolder.

✪ In PST, add the property $AirGapProbe with the location (0,0,-5) and $BodyAccelProbe with the location (0,0,2) to the RigidBody DObject input file.

✪ Run the simulation.

✪ Navigate to the DObject subfolder within the results subfolder and view the probe results.

☞ Notice the air gap and acceleration data reported.

☞ RigidBody probes are useful for determining the acceleration and position of points of interest within an advanced RigidBody structure (multiple rigid body features). The acceleration at the body coordinate frame origin is always reported in the 'results.dat' output file, however, points of interest such as rotor tips for turbines or the deck of a large vessel would benefit from the use of probes.

Tutorial 11 - Surface mooring

1 Tutorial overview

This tutorial will cover:

- Anchoring a cable to the seabed
- Cable and soil interaction
- Connecting a cable to a rigid body
- Rigid body hydrodynamic loading
- Rigid body coordinate frame

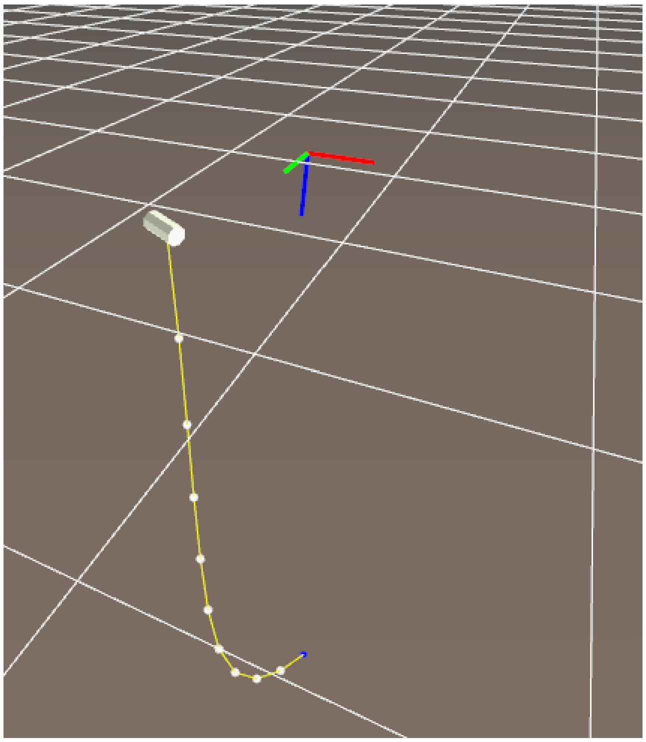


Figure 11: Surface mooring

2 Anchoring a cable to the seabed

- ✪ Create a new project.
- ✪ Create a cable.
- ✪ Specify the cable IC to span from Node 0 at global frame location (0,0,0.5) m to Node N at a location of (5,0,50) m with 10 elements and a length of 55m using a catenary IC.
- ✪ Node N of the cable, at the seabed, can be fixed in place by enabling the $NodeNStatic property.
- ✪ Create a $CableSegment feature. Leave the cable material properties for the feature as the default settings.
- ☞ If a catenary IC is specified and additional cable length is added, some of the cable nodes may be initially penetrated in the soil. Due to the soil stiffness, defined in the default $EnvironmentSoil feature, a seabed contact force will push the cable out of the seabed at the start of the simulation. Seabed contact is discussed in further detail in the 'Cable Catenary' tutorial.

3 Creating a floating RigidBody

- ✪ Create a RigidBody DObject.
- ✪ By design, the cylinder is 50% submerged in calm conditions. Set the $Mass to 800kg and $Ix, $Iy, and $Iz arbitrarily to 1e2.
- ✪ Create a RigidBodyCylinder feature called *float* with a diameter of 1m and a length of 2m.
- ✪ The RigidBodyCylinder feature should be placed at the DObject local origin but lie horizontal on the water. This can be realized by rotating the feature by 90 degrees about the y axis. Do this by adding the following to the RigidBody input file: '$Cylinder float 0 0 0 0 90 0'.
- ✪ For simplicity, use the default IC of the RigidBody at the global frame.

4 Connecting a Cable to a RigidBody

- ✪ Create a new connection and select the RigidBody as the master and the Cable as the follower. Select point connection.
- ☞ Just as a PointMass, a RigidBody is the master because it sets the position and velocity state of the boundary node of the cable. In turn, the cable automatically passes reaction forces to the RigidBody during simulation.
- ☞ Once a connection between the cable and cylinder is created, the connection location relative to the rigid body coordinate frame must be defined. This connection location in terms of the rigid body frame is constant throughout any simulation.
- ☞ In the connection properties section, $DCableFollowerLocation is a vector that locates the cable boundary node attachment location relative to the rigid body frame.
- ✪ Set the $DCableFollowerLocation to (0,0,0.5) m so the cable attachment point is on the outer surface of the cylinder on the bottom.
- ✪ Use $DCableFollowerNodeN 0 to specify the connection applies to the end of the cable at node 0.
- ✪ Be sure that the $DCableFollowerLocation on the RigidBody frame matches the cable IC defined in global space. In this case the location corresponds to (0,0,0.5) in the global coordinate frame and therefore no destabilization from excessive cable boundary element strain will occur.

5 Running the solver

✪ Specify an environment with 50m water depth and a 0.5m/s uniform current with a heading of 90 degrees.

✪ Set the length of simulation to 60 seconds and run the solver.

✪ View the results in PostPDS.

6 Changing the connection location

✪ Connect the cable to one of the ends of the cylinder by changing the $DCableFollowerLocation to (0,2,0) m.

✪ Rerun the simulation.

☞ Notice the change in the connection location. View the RigidBody DObject local origin in PostPDS to determine where the location of (0,2,0) m is in relation to the RigidBody origin.

Tutorial 12 - Segmented Cable

1 Tutorial overview

This tutorial covers:

- Using multiple materials in a Cable
- Using ExtMass features to represent instruments

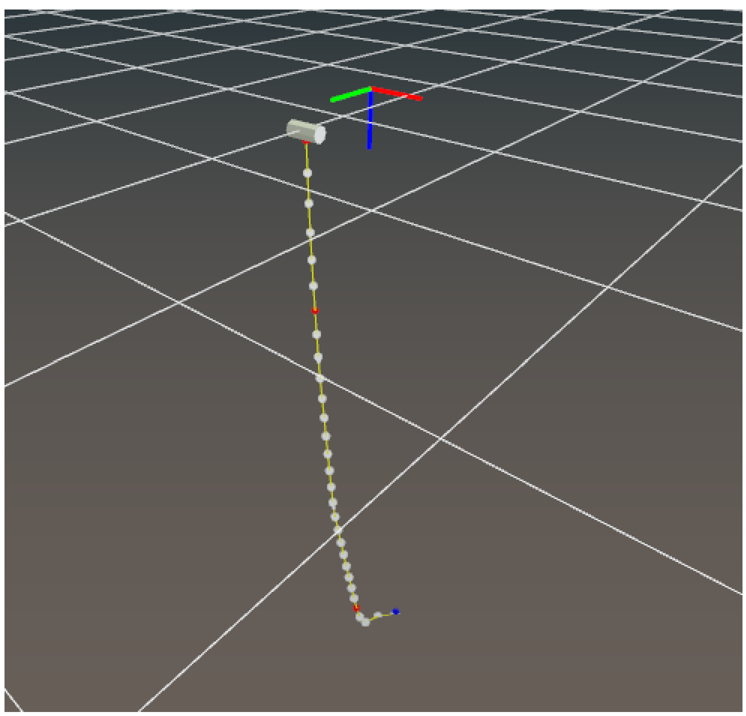


Figure 12: Surface mooring with multiple materials and instruments

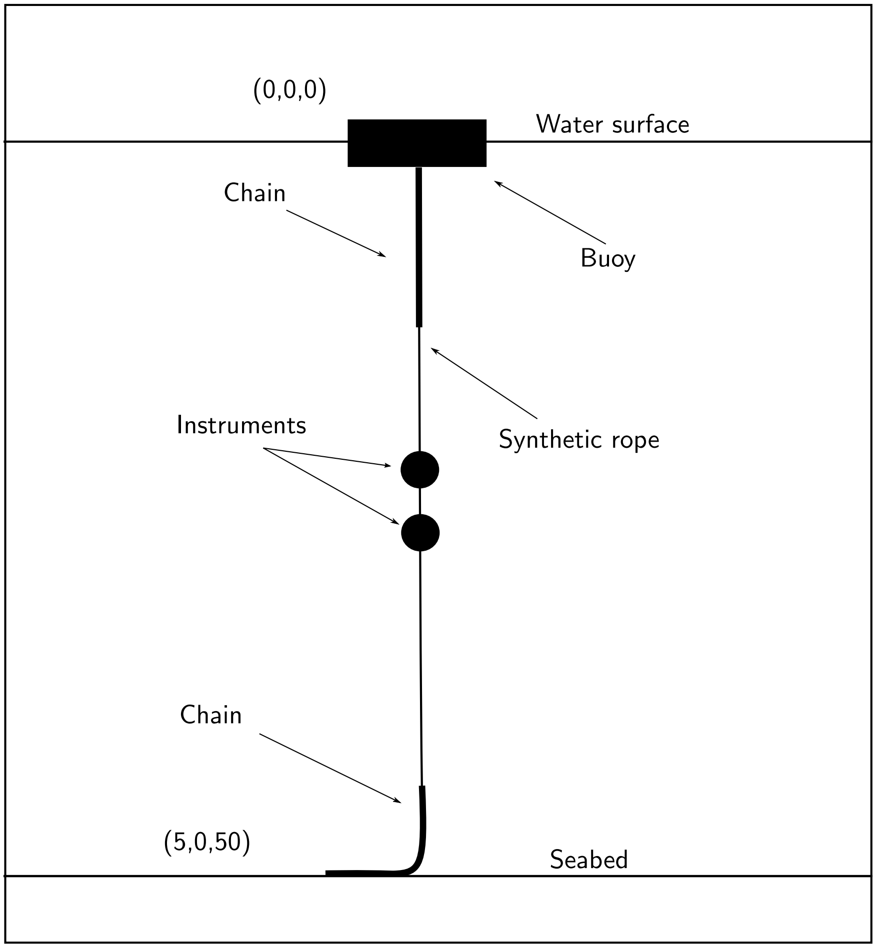


Figure 13: Segmented cable layout

2 Initialize simulation

✪ Create a copy of the previous tutorial project folder 'Surface mooring' and rename the copy 'Segmented mooring'.

3 Using multiple Cable material properties

☞ Though a single Cable is used, multiple material property segments will be specified representing chain and synthetic rope. The top and bottom portion of the mooring line will be chain with synthetic rope in the middle.

☞ In general, library features can be used multiple times and for different DObjects.

✪ Add a new DCableSegment feature to the library named *chain* with the following properties:

26mm studlink chain feature properties

```
// Fluid loading
$CDc 2
$CDt 0.1
$CAc 1
```

```
// Mechanical
$EA 6.5e6
$EI1 0
$EI2 0
$GJ 0
$Diameter 0.049
$Density 7700
$CID 500
$BCID 0
$TCID 0
$CE 0
```

✪ Add another DCableSegment feature to the library named *rope* with the following properties:

8mm synthetic rope feature properties

```
// Fluid loading
$CDc 1.5
$CDt 0.1
$CAc 1

// Mechanical
$EA 8.5e5
$EI1 4
$EI2 4
$GJ 8
$Diameter 0.008
$Density 795
$CID 500
$BCID 0
$TCID 0
$CE 0
```

☞ The material segment properties are specified in order from node 0.

☞ The first $CableSegment property will start at node 0 of the cable and run the arcspan specified. Then the second $CableSegment properties specified will be applied for the arcspan specified.

☞ The last material property specified is applied for the remainder of the Cable span.

☞ Material properties are constant for each finite element. The nearest node to the material property change is used to transition material properties.

✪ In the Cable input file, first specify: '$CableSegment chain 10'

✪ Next, specify: '$CableSegment rope 35'

✪ Finally, specify the last 10m of mooring line to the anchor to be chain by using: '$CableSegment chain 10'

4 Using an ExtMass for mooring instruments and hardware

☞ It is common to use an ExtMass to represent instruments or other significant hardware along the span of the line

☞ The ExtMass weight, buoyancy, drag, and added mass effects are dictated by a reference spherical diameter and average density. The diameter can be adjusted to approximate frontal area for drag calculations and the density can be used to ensure consistent wet weight load on the line.

✪ Add an ExtMass feature to the library and name it *instrument*. Set the diameter to 0.3m and density to 5000kg/m^3. The net weight in salt water acting on the mooring line is 550N.

✪ In the Cable input section, add an *instrument* at an arcspan location of 15m.

✪ Add another instance of an *instrument* at 30m.

☞ The Cable input section should resemble the following layout:

Cable input

```
// Boundary constraints
$Node0Static 0
$NodeNStatic 1
$NodeNRing 0

// Mechanical
$CableSegment chain 10
$CableSegment rope 35
$CableSegment chain 10

$ExtMass instrument 15
$ExtMass instrument 30
```

5 Running the simulation and processing results

✪ Use the previous environment settings with 50m water depth and a 0.5m/s uniform current with 90deg heading.

✪ Set the length of simulation to be 30s and run the simulation.

✪ View the results in PostPDS. Note in both rendered and engineering views, the ExtMasses are visible and represented by a sphere with the same diameter as the *instrument*.

☞ In engineering view with 'Show element nodes' activated, the node that demarcates the change in element material properties are indicated in red.

6 The effect of Cable finite element mesh discretization error

☞ With 10 elements used in the Cable, tension at the top of the mooring (element 1) is approximately 3000N. However, only one element represents the 10m chain section interacting with the sea floor. There may be discretization error that affects the simulation results.

✪ Click on 'Define Initial Conditions' in the Configuration tab when editing the Cable parameters and increase number of elements from 10 to 30.

✪ Run the simulation again.

✪ Note the simulation execution speed is slower. This is due to a combination of the extra computational load from more elements but also because the average timestep is smaller.

☞ Smaller finite element lengths can resolve higher frequency modes of axial vibration, which drives the time step to smaller values. Care must be used to ensure a reasonable number of elements is used to run simulations in a reasonable amount of time without comprimising the accuracy of the dynamics involved.

✪ Reload the results in PostPDS. If PostPDS is still open, try using the reload button near the top left of the window.

✪ Plot the tension at the top of the cable (element 1). The tension is approximately steady at 2600N. Further examination of the nodes shows many more elements in the chain section, which is partially lifted off the sea floor. The difference in tension is due to the discretization error of the chain section near the bottom, which pulled the surface float too far down into the water, artificially increasing tension.

Tutorial 13 - Restarting a simulation

1 Tutorial overview

This tutorial covers:

- Restarting a simulation using final states
- Importance of steady state analysis

2 Restarting a simulation

☞ After a simulation is completed or canceled, the final state of all DObjects in the simulation is saved within the results folder.

☞ A new project can be created that starts all DObjects with their state (position and velocity for all 6 degrees of freedom) as the final position from a previously run simulation.

✪ Open a previously completed simulation in PST and click the 'Export Final Condition to New Project' button on the Simulation tab.

✪ Choose a new folder to save the new project to.

☞ The IC files of the DObjects will be the final states from the previous simulation.

☞ Note that when using a restart state vector, the start and end time within the 'sim.ini' file must be updated accordingly to preserve the environmental conditions. For example, if a simulation is being restarted at 10 seconds simulation time and the user desires the simulation to continue with the exact environmental conditions present at 10 seconds, set '$StartTime 10'. This is particularly important using irregular sea states such as JONSWAP wave spectrums.

✪ A simulation restart will be performed in tutorial 'Cable catenary' demonstrating the benefit of this feature.

3 Importance of steady state analysis

☞ When completing various simulations, it is important to realize the difference between the transient state and steady state of the system.

☞ It is sometimes beneficial to start a system in a steady state configuration to avoid initial transient unusable data. Restarting a simulation is useful for starting a simulation after the transient behavior has settled.

Tutorial 14 - Cable catenary

1 Tutorial overview

This tutorial covers:

- Creating a 'false bottom' cable configuration
- Using ExtMass as floats
- Using PointMass as a surface float

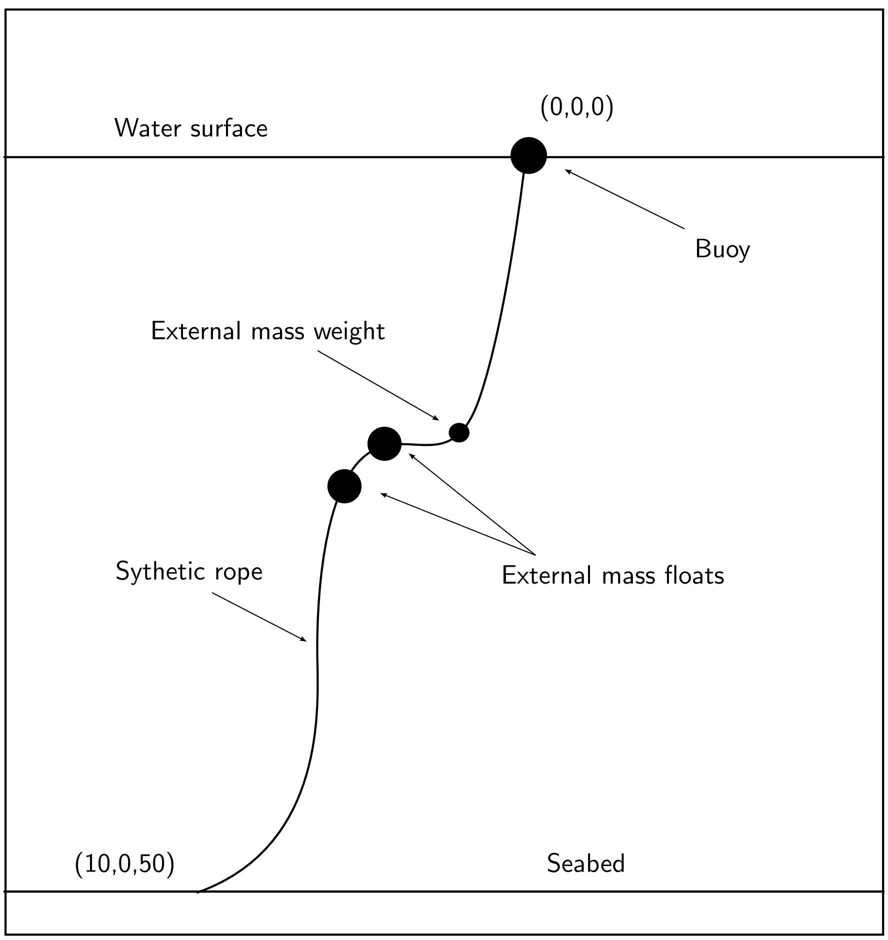


Figure 14: Cable Catenary layout

2 Creating a catenary cable

- Create a project that contains a single cable with node 0 at (0,0,0)m and node N at (10,0,50)m. Give the cable a length of 60m with 15 elements. The extra cable length will initially create a catenary that penetrates the seabed and therefore the simulation will run slowly at the beginning as the soil forces the cable out.
- Use a uniform cable material of the synthetic rope with the following properties:

8mm Dyneema feature properties

```
// Fluid loading
$CDc 1.5
$CDt 0.01
$CAc 1

// Mechanical
$EA 8.5e5
$EI1 4
$EI2 4
$GJ 8
$Diameter 0.008
$Density 850
$CID 500
$BCID 0
$TCID 0
$CE 0
```

- Set the water depth to 50m and anchor the cable at the seabed.
- Create a 0.25m/s uniform current with a heading of 0 degrees.

3 Seabed contact dynamics

- A seabed contact model is used to apply forces on Cables, RigidBodies, and PointMasses to prevent them from passing through the seabed. The seabed is comprised of contact stiffness, damping, and tangential motion friction. The soil properties are constant regardless of the object interacting with it. A figure demonstrating the forces acting on a PointMass beneath the polygonal seabed can be seen in figure 15.
- When the cable is in contact with the seabed, contact forces appear. All the forces combine to a normal and tangential contact force. More on the seabed contact model can be found in the ProteusDS manual.

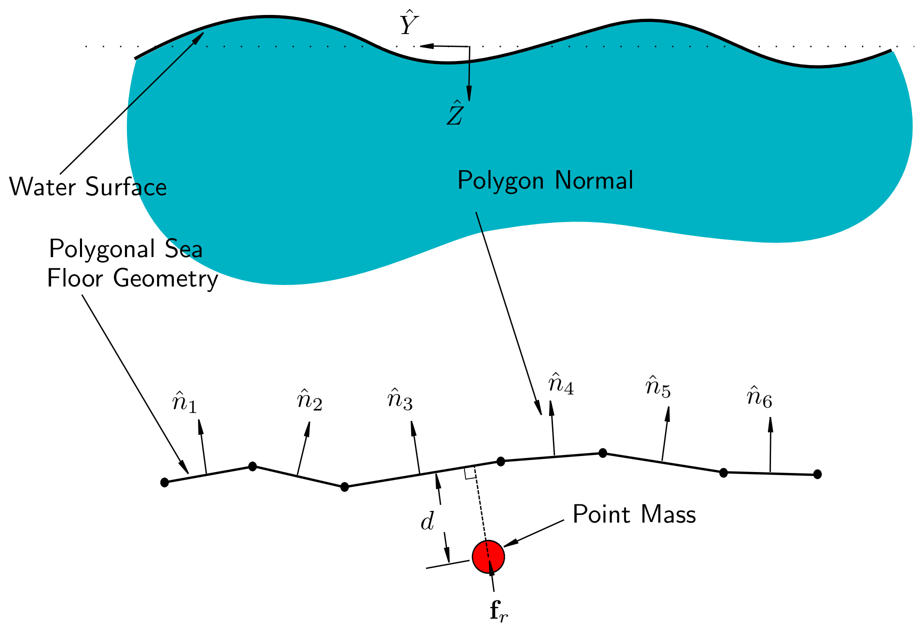


Figure 15: Seabed contact model

- ✪ Run the simulation for 10 seconds. View the results in PostPDS.
- ✪ In this time, the section of cable that was initialized beneath the seabed should have been displaced.
- ✪ Restart the simulation by clicking the 'Export Final Conditions to New Project' button in PST.
- ✪ Create a new folder and save the new project.
- ☞ This will start the new simulation with an initial orientation with the entire cable above the seabed.

4 Adding ExtMass floats and weight

- ☞ To create a lazy-s shape or false bottom in the cable, additional floatation and weight must be added to the cable at specific locations.
- ✪ Create 2 ExtMass features. One representing the float and the other representing the weight.
- ✪ Give the float ExtMass an added mass and drag coefficient of 1, a diameter of 0.28m, and a density of 450kg/m^3.
- ✪ Give the weight ExtMass the same added mass and drag coefficients used for the float but use a diameter of 0.125m and the density of 7000kg/m^3.
- ✪ Add 3 $ExtMass properties to the cable input file to represent the floats and weight. Create a float Extmass at a cable arclength of 25m and 30m and create a weight ExtMass at a cable arclength of 20m.
- ☞ The external masses will create enough upward force to lift the lower portion of the cable and to generate a slack region in the cable.
- ☞ The amount of floatation required can be calculated by taking the net wet weight of the lower cable and matching that to the total ExtMass wet weight.

5 Creating a PointMass buoy

✪ Create a PointMass with a 1m diameter and a density of 250kg/m^3 connected to node 0 of the cable. The PointMass IC will be at the global origin with no velocity.

☞ The PointMass will act as a surface buoy designed to follow the surface of the water.

6 Benefits of false bottom

☞ The false bottom decouples the surface movement from the lower portion of the cable. In a strong sea state, the surface buoy will follow the heave of the waves but the lower portion of the cable will have reduced motion due to the displacement of the cable being absorbed by the false bottom.

☞ The false bottom has a similar effect on high frequency tensions. The tensions occurring at the top of the cable will be absorbed by displacing the false bottom, therefore the tensions will not be transferred down the cable.

7 Running the solver and viewing the results

✪ Set the length of simulation to be 60 seconds and run the solver.

✪ View the results in PostPDS.

Tutorial 15 - Current profiles

1 Tutorial overview

This tutorial covers:

- Using custom current profiles
- Adding power law boundary layer current profile modifier

2 Creating a custom current profile

☞ Custom current profiles are used to represent current profiles with complex variation through the water column.

☞ Time varying current profiles and depths can be specified.

☞ Data must be provided indicating the current magnitude at discrete time and depth points. The current values are interpolated linearly over depth and time for a smooth transition.

☞ Custom current profile data files are formatted as: [Time(s), Depth(m), Velocity(m/s), Heading(deg)].

☞ Custom current profiles are created in separate *.dat files, which are referenced in the environment input file.

✪ Create a new file and name it 'CustomProfile.dat'.

✪ Add the following to 'CustomProfile.dat':

Custom Current Data

```
0        50       1.0      50
0        45       1.0      50
0        40       1.0      50
0        35       1.0      50
0        30       1.1      50
0        25       1.1      50
0        20       1.1      50
0        15       1.2      50
0        10       1.2      50
0        5        1.2      50
0        0        1.2      50
10       50       3        100
10       45       3        100
10       40       3        100
10       35       3        100
10       30       3        100
10       25       3        100
```

```
10        20        3        100
10        15        3        100
10        10        3        100
10        5         3        100
10        0         3        100
```

☞ This custom current profile specifies one current profile at a time of 0s with a heading of 50deg and a free stream velocity of 1.2m/s tapering down to 1m/s at depth. The current profile transitions such that at 10s it will have a heading of 100deg and a uniform free stream velocity of 3m/s.

✪ Save this file in the project directory.

✪ In PST, in the environment input file, change the $CurrentProfile property value to 9.

✪ Set the property '$CurrentCustomProfileFile CustomProfile.dat' to use the custom profile data generated.

3 Using a boundary layer current profile

☞ Boundary layers form where the current interacts with the sea floor.

☞ The boundary layer using a spatially varying scale factor to transition the current profile to zero at the seabed over a specified boundary layer thickness.

☞ The boundary layer uses the following formula:

$$U(z) = \left(\alpha \left(\frac{z}{Z_0}\right)^{\beta} \bar{U}\right) \qquad \text{for } 0 \leq z \leq 0.5 Z_0$$

where U(z) is the current velocity at height z in m/s, $\bar{U}$ is the depth averaged free-stream velocity in m/s, z is the height from the seabed in m, and Z_0 is the total water depth in m. α and β are dimensionless constants. The standard values for α and β are 1.18 and 0.143, respectively.

✪ To use a boundary layer current profile set the $CurrentProfile propery value to 8.

✪ Set the $CurrentHeading and $CurrentSpeed properties. The $CurrentSpeed property represents the free-stream velocity of the current.

✪ Set the $CurrentPowerLawAlpha and $CurrentPowerLawBeta properties. The default values can be used for most standard boundary layers.

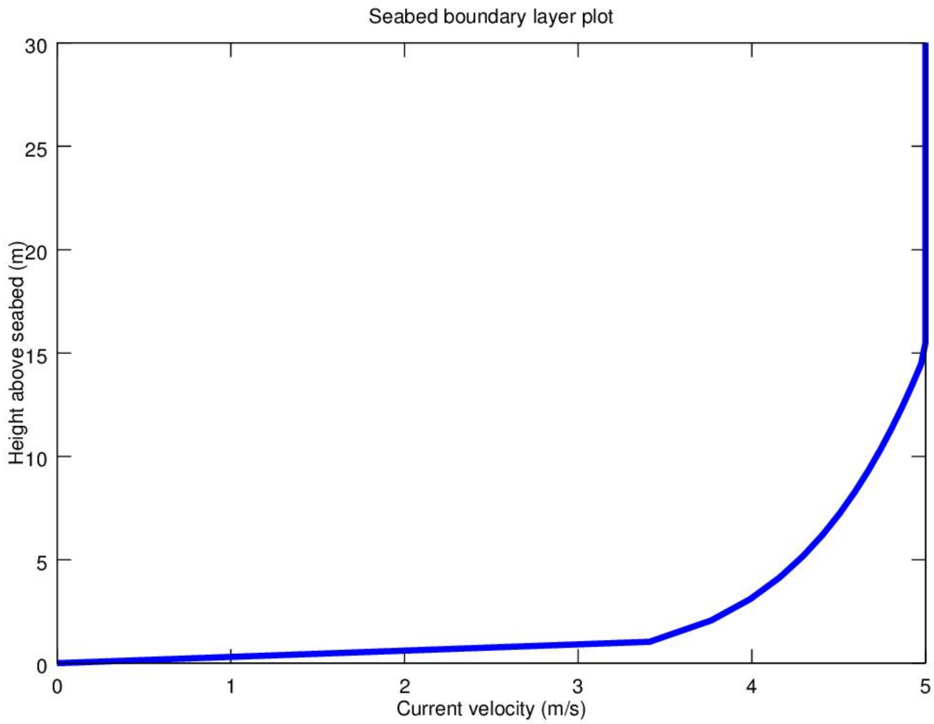
✪ The property $CurrentPowerLawPercentDepth is required to specify the percentage of the total current profile that will be scaled with a boundary layer. The equation above specifies that 50% of the profile is scaled, however this may change in different situations.

☞ The boundary layer profile will automatically calculate a $\bar{U}$ value based on the specified alpha, beta, percent depth value and water depth value. This ensures that a smooth profile is used.

✪ Run a simulation with a boundary layer current profile and water velocity probe.

✪ Plot the probe output files to view the boundary layer current profile.

☞ A current profile of 5m/s in 30m of water with a 50% boundary layer using the standard alpha and beta values will look like this:


Seabed boundary layer plot
Height above seabed (m)
Current velocity (m/s)
0
5
10
15
20
25
30
0
1
2
3
4
5

Tutorial 16 - Wave models

1 Tutorial overview

This tutorial will cover:

- Wave models
- Using the JONSWAP wave spectrum
- Using the $WaveSeed property
- The spreading function

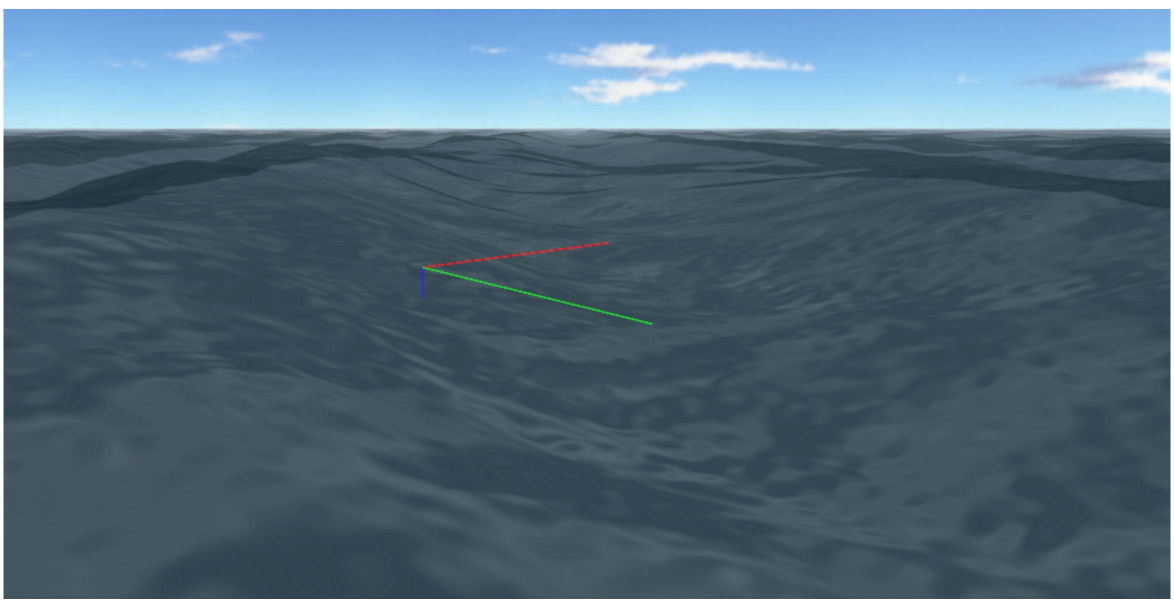


Figure 16: Waves in PostPDS in rendered mode

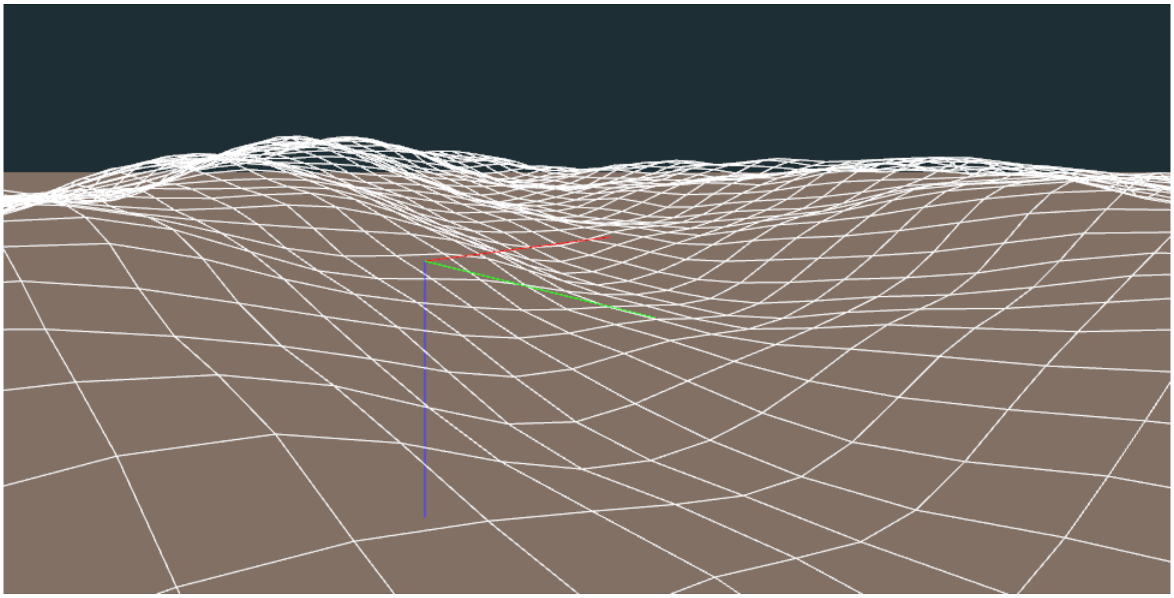


Figure 17: Waves in PostPDS in engineering mode

2 Using advanced wave profiles

- ☞ ProteusDS allows the use of several types of wave profiles in simulation.
- ☞ The most basic wave profile is the Airy wave (Stokes first order). The free surface elevation of one wave component is sinusoidal. This approximation is accurate for small ratios of wave height to water depth (shallow water waves) and wave height to wave period (deep water waves).
- ☞ A Stokes second order wave is also available. This wave profile requires an average depth to determine the surface elevation and velocity potential.
- ☞ The Pierson-Moskowitz wave spectrum represents a fully developed wind seas when the waves reach equilibrium with the wind.
- ☞ Finally, the JONSWAP spectrum is also available. The JONSWAP is an extension of Pierson-Moskowitz to account for fetch-limited, or developing seas.

3 Applying a JONSWAP wave spectrum

- ✪ Open the 'Rigid body with cylinder feature' tutorial project.
- ✪ In the environment input file, use a JONSWAP wave spectrum by setting '$WaveType 8'.
- ☞ When using a wave spectrum, certain follower properties are required. Use the following property declarations in the environment input file.

JONSWAP wave properties

```
$WaveType 8
$WaveHeight 2
$WaveHeading 0
$WavePeriod  5
```

```
$WaveSeed 12345
$WaveSegments 100
```

☞ The $WaveSegment property determines the number of discrete waves for the simulation to use. By taking the number of wave segments and wave period, the amount of time before the sea state repeats can be calculated. The formula used to determine the length of non-repeating waves is provided in the ProteusDS Manual.

☞ The $WaveSeed property is used as the seed for a random number generator. This property allows for consistent consecutive random numbers to be generated and therefore repeatable simulation conditions.

✪ Run the simulation and view the results.

4 Using the wave seed property

☞ Changing the wave seed value will provide a new set of randomly generated waves.

✪ Set '$WaveSeed 54321'.

✪ Rerun the simulation and view the results

5 Adding a spreading function

☞ When the wave spreading function is enabled, the waves are spread in headings between $-\frac{\pi}{2}$ and $\frac{\pi}{2}$ radians around the mean wave heading.

✪ Enable wave spreading by setting '$WaveSpreadingFunction 1' in the environment input file.

✪ The $SpreadingFunctionConstant is the exponent used to calculate the spreading. Set '$SpreadingFunctionConstant 2'.

✪ Rerun the simulation and view the results

Tutorial 17 - Wind models

1 Tutorial overview

This tutorial will cover:

- Using wind models
- Creating a cylinder rigid body feature
- Setting up the environment with wind

2 Creating a RigidBody DObject

✪ Initialize a vertical RigidBody using a RigidBodyCylinder feature. Give the rigid body 0 initial displacement and rotation about the global reference frame.

✪ Set the cylinder feature coordinate frame to be coincident with the RigidBody frame.

✪ Use the following properties in the RigidBody DObject input file:

Rigid body properties

```
// Mechanical
$Ix 134
$Iy 134
$Iz 1
$Ixy 0
$Ixz 0
$Iyz 0
$Mass 100

$CGPosition 0 0 2
$DefineInertiaAboutCG 1
```

☞ The center of gravity was moved relative to the RigidBody frame by 2m in the z direction. This places the center of gravity below the center of buoyancy, therefore making the cylinder more stable as it floats in the water.

✪ Use the following properties for the RigidBodyCylinder:

Cylinder feature properties

```
// Fluid loading
$WindLoading 1
$HydroLoading 1
```

```
$BuoyancyFroudeKrylov 1
$CdAxial 1
$CdNormal 1
$CaAxial 1
$CaNormal 1
$CDt 1 //fluid tangential drag is enabled

// Mechanical
$Diameter 0.25
$Length 4

// Numerical
$AxialSegments 10
$RadialSegments 3
$AngularSegments 8
```

☞ Note the optional property $CDt set to 1. This enables the fluid tangential drag.

✪ Set the simulation end time to 30 seconds and run the simulation.

✪ View the results in PostPDS. Notice the cylinder bobbing in the water with zero horizontal displacement. This is due to the absence of current, waves, or wind acting on the RigidBody.

3 Applying wind to the simulation

☞ To enable wind in the simulation, several properties associated with wind must be declared.

☞ A time varying wind is produced by selecting a specific spectrum.

☞ Variation in the wind speed with height is produced by selecting a specific profile.

☞ More information on the available wind spectra and profiles can be found in the ProteusDS Manual.

✪ To enable a uniform profile with an Ochi-Shin spectrum, add the following properties to the environment input file: ' $WindProfile 1', '$WindSpectrum 1'.

✪ Due to the fact that the Ochi-Shin spectrum was chosen, several follower properties must be declared. Use the following properties to create the Ochi-Shin spectrum:

Ochi-Shin follower properties

```
$Wind10MinMeanSpeed 5
$WindSurfaceRoughness 0.01
$WindNumFreq 50
$WindCutOffFreq 100
$WindHeading 0
```

☞ More information on the specific wind properties and their masters can be found in the ProteusDS Manual Input Files Appendix.

4 Running the simulation and visualizing in PostPDS

✪ Run the simulation again and view in PostPDS.

- ☞ Notice the cylinder being displaced horizontally due to the wind loading.
- ☞ The wind only acts on the portion of the RigidBody above the water and if the RigidBody feature has the $WindLoading property enabled.

Tutorial 18 - Custom seabed bathymetry

1 Tutorial overview

This tutorial covers:

- Using a custom bathymetry
- RigidBody interaction with the seabed
- Visualizing the bathymetry in PostPDS and plotting

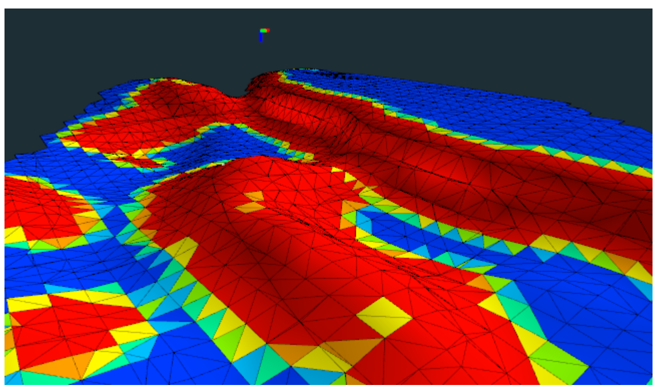


Figure 18: Exclusion zone plot of custom bathymetry

2 Using a custom bathymetry

☞ Many DObjects can interact with the sea floor.

☞ By default, a flat sea floor is used. However, custom bathymetry can be specified.

☞ Custom bathymetry data must be provided as a polygonal mesh *.obj file located in the project folder.

☞ An overview of recommended practices for generating custom bathymetry mesh data can be found in the ProteusDS Manual.

✪ Create a new project and move the custom bathymetry file provided, 'CustomBathymetryMesh.obj', into the project folder.

✪ In the environment input file, set: '$CustomBathymetry 1'

✪ Indicate the polygonal bathymetry mesh file: '$CustomBathymetryFile CustomBathymetryMesh.obj'

- ☞ All polygonal mesh files use a local reference frame. The polygonal mesh file '$CustomBathymetryMesh.obj' has its local reference frame located in the middle of the seabed.
- ✪ Specify the offset of the custom bathymetry mesh frame at 50m depth from the global reference frame by specifying: '$BathymetryOffset 0 0 50'

3 Custom bathymetry with currents and waves

- ☞ By default, currents ignore the presence of custom seabed bathymetry.
- ☞ A current boundary layer effect at the seabed can be applied to scale and align any current profile to local bathymetric variation.
- ☞ Custom bathymetry is not accounted for in wave calculations. Therefore, if waves are present in the simulation, it is important to set the $WaterDepth property to an appropriate reference depth for wave calculations.
- ✪ For this simulation, the approximate average water depth is 50m. Set $WaterDepth to 50m to ensure proper wave calculations are performed.
- ✪ Execute a 1 second simulation and visualize the results in PostPDS to verify the bathymetry data has been loaded into the simulation.

4 Create a RigidBody

- ✪ Create a new RigidBody.
- ✪ By design, the mass will sink to the seabed. Set the mass to 300kg and arbitrarily set the mass moment of inertia in the x and y direction to to 76kgm^2, and in the z direction to 150kgm^2.
- ✪ Set the RigidBody IC at a location of (-20,0,50) with no velocity.
- ✪ A physical hull is needed to interact with the soil. Add a cylinder feature to the library and set 0.2m length and 1m diameter.
- ✪ Add an instance of the cylinder to the RigidBody.

5 Simulating friction effects

- ✪ Set simulation length to 2 seconds and execute.
- ☞ Note at around 1.8 seconds, the simulation slows down and adaptive numerical integrator shrinks the time step significantly.
- ✪ When the simulation is complete, load and visualize the results in PostPDS.
- ☞ Note the bottom stiffness is soft enough to allow deep penetration of the RigidBody hull into the soil.
- ☞ The adaptive numerical integrator can be sensitive to effects like friction. Friction can involve large forces that oscillate in direction when the bodies are at low velocities.
- ✪ Set the seabed coefficient of friction $MuSoil to 0 in the soil properties listed in the library.
- ✪ Set simulation length to 10s and run the simulation. Note the simulation runs consistently faster. When visualizing the results in PostPDS, the RigidBody slides down the slope of the face.

☞ This shows friction is causing the time step to decrease and significantly increase simulation execution speed. To circumvent the issue, the constant time step integrator can be used.

6 Using the constant time step integrator with friction

✪ Add a new integrator to the library and call it *RK4*. In the new integrator, set: '$IntegrationTypeNum 0', '$InitialTimeStep 0.05'.

✪ In the simulation input section, set: '$Integrator RK4'.

✪ Set the seabed coefficient of friction back to 0.5.

✪ Run the simulation and load the results in PostPDS.

☞ In this case, friction is present and large enough to prevent the RigidBody from sliding down the slope. However, the constant time step integrator does not adaptively change the time step due to friction effects and so the simulation executes rapidly.

☞ A disadvantage to the constant time step integrator is finding an acceptable time step to use that is applicable through the entire simulation. When cables are simulated in dynamic environments, time step requirements can vary by several orders of magnitude.

7 Environmental interaction with the seabed

✪ Set a uniform current profile with heading 0deg and speed of 5m/s.

✪ Create a parabolic boundary layer with a thickness of 2m. Set $SeabedBoundaryLayerFluidVelocityScaling to 1 and $SeabedBoundaryLayerFollowSlope to 1.

✪ Place a water velocity probe at (-20,0,52), which happens to be close to the location where the RigidBody impacts the seabed for the first time.

✪ Save and run the simulation for 10 seconds.

☞ This demonstrates the effect of a custom bathymetry on the current boundary calculations as the z velocity component of the probe indicates an upwelling component at that location as the current profile follows the seabed slope in the boundary layer.

8 Viewing the custom bathymetry in PostPDS and plotting

✪ Open the simulation results in PostPDS.

✪ Hide the water surface and turn off any fog effects to better view the bathymetry.

✪ Right click on the 'Sea floor' object in the object tree to bring up the custom bathymetry plotting options.

✪ Click the 'Display mesh edges'. This option displays the polygonal mesh edges to allow polygonal mesh inspection.

✪ Right click on the 'Sea floor' object and in the Bathymetry Plotting menu select 'Plot Depth'. Accept the default maximum and minimum values. This option colours the bathymetry mesh based on depth.

☞ The slope of the bathymetry can also be plotted by colour.

☞ The exclusion zone plots areas above and below a slope threshold. Regions above the slope threshold are indicated in red. Indicating exclusion zones is useful for locating potential safe locations for placing anchors or other equipment on the sea floor.

✪ Click ‘Plot Exclusion Zones’ and use default inputs of an exclusion angle threshold of 20deg, exclusion zone range of 10m, and a buffer range of 20m. Now all regions with slopes greater than 20deg are clearly indicated.

☞ The bathymetry slope data can be exported by clicking the ‘Export Slope Data’. This function saves the slope data to a *.csv file that can be used for additional custom post processing.

Tutorial 19 - Cable internal/structural damping

1 Tutorial overview

This tutorial will cover:

- Cable internal damping
- Over-damping
- Appropriate damping values

2 Defining Cable internal and structural damping

☞ The shorter the element lengths used in the cable model, the better it is at resolving high frequency modes of motion. Usually, these high frequencies do not affect system behavior in any important way, though the adaptive numerical integrator can detect them and shrink the time step to resolve them automatically. This results in slow simulation execution speed.

☞ Often, high frequency noise can be seen in cable tension history as the cable finite element lengths decrease. One way of mitigating the noise is to use internal structural damping. This mitigates high frequency noise, allowing larger time steps to be used and faster simulation execution speeds.

☞ The properties $CID (coefficient of internal damping), $BCID (bending coefficient of internal damping), and $TCID (torsional coefficient of internal damping) are used.

☞ Care must be taken in selecting the damping coefficients because values that are too high can also reduce the time step and result in slow simulation execution.

☞ Values must be selected that do not detrimentally affect the accuracy of the time domain response of the tensions at frequencies of interest.

✪ Open the 'Surface mooring' tutorial project file.

✪ Update the current to 3m/s.

✪ Update the cable feature using the following property values.

Cable properties without damping

```
// Fluid loading
$CDc 1
$CDt 0.01
$CAc 1

// Mechanical
$EA 1000000
```

```
$EI1 100
$EI2 100
$GJ 0
$Diameter 0.01
$Density 1100
$CID 0 //removed damping
$BCID 0
$TCID 0
$CE 1
```

3 Running and visualizing a simulation without internal damping

- ✪ Run the simulation for 30 seconds.
- ✪ Visualize the results and enable the coloured tension plot on the cable to view tension spikes that occur. Set the legend to have a minimum value of 0N and a maximum of 15000N to allow for clear viewing of the tension spikes.

4 Consequences of over-damping

- ✪ To eliminate the tension spikes occurring in the cable, set '$CID 1e7'.
- ✪ Rerun the simulation and notice that the simulation time step has decreased to such a small value that the simulation will take too long to complete.

5 Using appropriate damping values

- ✪ Lower coefficient of internal damping with '$CID 1e4'.
- ✪ Rerun the simulation and notice the speed in which the simulation is executing.
- ✪ Visualize the results in PostPDS and plot the cable tension again.
- ☞ Notice that the tension spikes have been mitigated.

Tutorial 20 - Bridled Mooring

1 Tutorial overview

This tutorial covers:

- Creating Cable to Cable connections
- Creating a bridled mooring

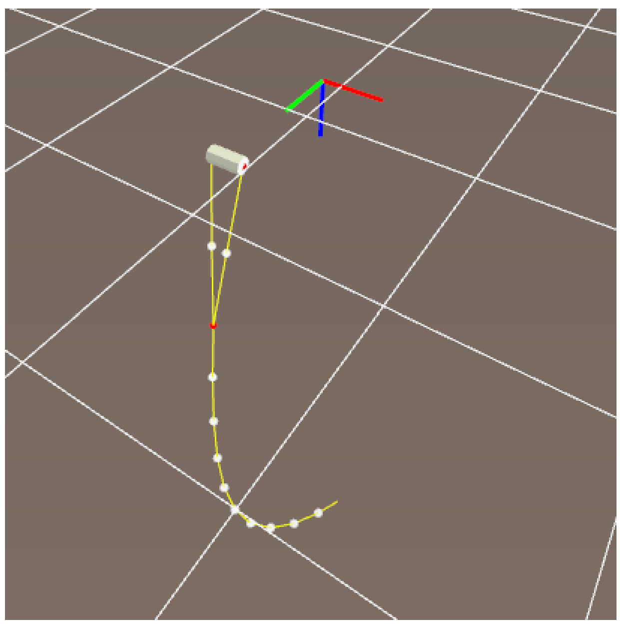


Figure 19: Bridled mooring configuration

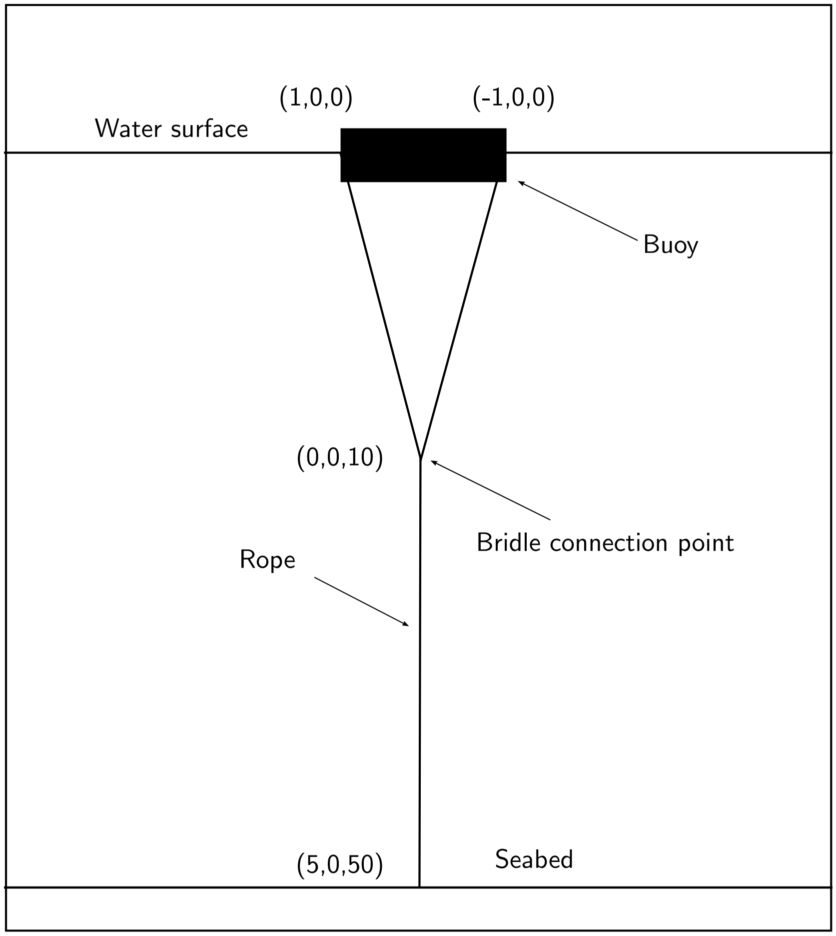


Figure 20: Bridled mooring schematic

2 Creating a RigidBody float with a bridle mooring

2.1 Create a RigidBody float

☞ A similar configuration to the surface mooring will be used.

- ✪ Create a new project.
- ✪ Create a RigidBody. Leave the IC with default values.
- ✪ By design, the rigid body will be 50% submerged in calm conditions. Specify $Mass to 800 and set moments of inertia arbitrarily to 1e2.
- ✪ Create a cylinder feature called float with diameter 1m and length 2m.
- ✪ Add a float to the rigid body such that it is lying flat on the water by specifying in the rigid body input section $Cylinder float 0 0 0 0 90 0.

2.2 Create bridle lines

☞ The bridle lines will connect to the rigid body at the ends of the cylinder.

✪ Create a new cable and name it *bridle0*.

✪ Define the IC to have node 0 at (-1,0,0) and node N at (0,0,10), 2 elements, and length of cable 11m.

✪ Create another new cable and name it *bridle1*.

✪ Define the IC to have node 0 at (1,0,0) and node N at (0,0,10), 2 elements, and length of cable 11m.

✪ Leave material properties for both cables at default values.

2.3 Create main mooring line

☞ The main mooring line will connect to the bridle lines and will be fixed at the seabed.

✪ Create a new cable and name it *mooring0*.

- Define the IC to have node 0 at (0,0,10) and node N at (5,0,50), 10 elements, and length of cable 45m.
- Fix node N in place by setting '$NodeNStatic 1'.

3 Connecting mooring and bridle

✪ Create a connection between *mooring0* and *bridle0*. Specify mooring0 as the master and *bridle0* as the follower and select point connection type.

☞ The master constrains position and velocity of the follower and in turn the follower passes reaction loads to the master during simulation.

✪ The connection properties must be defined. The arcspan location along the master Cable must be indicated where the end node of the follower Cable is connected to. Since this is at the boundary node 0 on *mooring0*, leave this as the default value of $DCableFollowerLocation 0.

✪ Set $DCableFollowerNodeN 1 to indicate that node N of *bridle0* will be connected to *mooring0*.

✪ Create a new connection between *mooring0* and *bridle1* in the same fashion.

4 Connect bridles to the float

✪ Create a new point connection between *bridle0* and the RigidBody. The connection location in terms of the rigid body local frame is (-1,0,0).

✪ Create a new point connection between bridle1 and the rigid body float. The connection location in terms of the rigid body frame is (1,0,0).

5 Simulate

✪ Specify an environment with 50m water depth and a 0.5m/s uniform current with a heading of 90 degrees.

✪ Set the length of simulation to be 30s. Run the simulation and view the results in PostPDS.

Tutorial 21 - Multi-Mesh RigidBody

1 Tutorial overview

This tutorial covers:

- Adding multiple load and mesh features to a rigid body
- Referencing global, rigid body, and feature frames

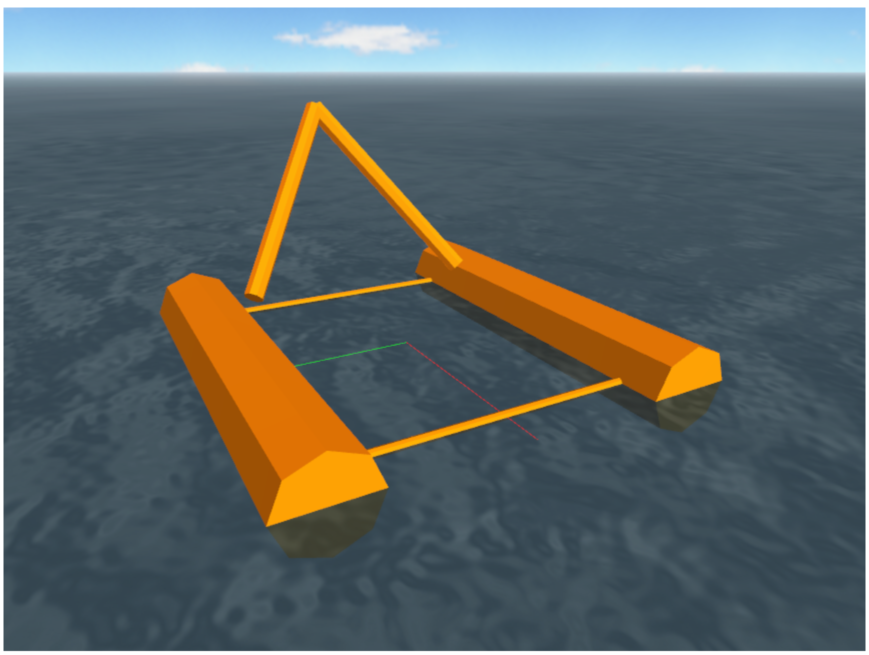


Figure 21: Simple raft with multiple load features

2 Setting a up simulation with a rigid body

- ✪ Create a new RigidBody and name it *raft*.
- ✪ By design, the raft will be 50% submerged in calm conditions due to the buoyancy of the pontoons. Set '$Mass 3e4' and set the moments of inertia arbitrarily to 1e4.
- ✪ To help keep the raft stable, set '$CGPosition 0 0 1'.

3 Creating multiple rigid body features

☞ An abstract diagram showing feature frames (f1 and f2) relative to the RigidBody frame (rb) which is relative to the global frame (O) can be seen in Figure 22.

☞ Every feature associated with a RigidBody has this type of reference frame hierarchy.

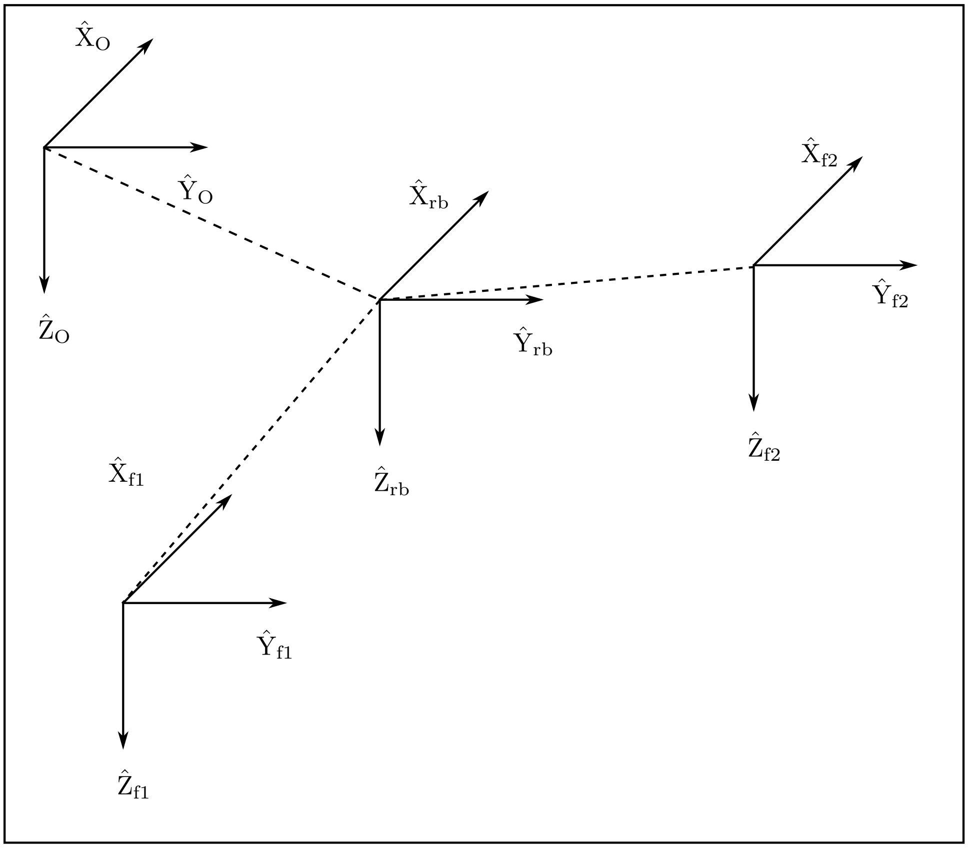


Figure 22: Reference frame layout

☞ Three different sized cylinder load features are used for the pontoon floats, the struts between them, and the A-frame booms.

✪ Add a cylinder feature to the library, name it *pontoon*, and set the diameter and length to 2 and 10, respectively.

✪ Add a cylinder feature to the library, name it *strut*, and set the diameter and length to 0.2 and 6, respectively.

✪ Add a cylinder feature to the library, name it *frame*, and set the diameter and length to 0.5 and 6, respectively.

✪ Each cylinder must be added to the rigid body. The position and orientation of the feature frame is specified relative to the rigid body frame. To simplify setup, the rigid body frame is located between the two horizontal pontoons. Position the features as shown below:

Nomad feature locations

```
$Cylinder pontoon 0 4 0 0 90 0
$Cylinder pontoon 0 -4 0 0 90 0

$Cylinder strut 4 0 0 0 90 90
$Cylinder strut -4 0 0 0 90 90
```

```
$Cylinder frame -4 1.5 -3 0 45 45
$Cylinder frame -4 -1.5 -3 0 45 -45
```

☞ Note rotation angles to position the booms in the A-frame at the rear end of the raft.

4 Running the simulation

✪ Run the short simulation.

✪ View the results in PostPDS. Right click on the RigidBody DObject and select 'Display Local Origin'. This will display the position of the RigidBody local coordinate frame origin.

☞ The large pontoon floats provide buoyancy forces to keep the raft afloat. Incorporating the A-frame geometry is necessary to ensure wind forces will be accounted for. While the struts in between the pontoons are small, they may contribute significant load from drag forces as well.

5 Observing changes in cylinder feature properties

✪ Shift the CG position aft to better represent the distribution of mass from the A-frame. Observe the change in steady position.

✪ Increase the resolution of the cylinder meshes and observe the influence in calm conditions.

✪ Create additional simple custom mesh features such as a box for a barge deck and add it to the raft.

Tutorial 22 - Custom hydrodynamic mesh

1 Tutorial overview

This tutorial will cover:

- Acceptable mesh formats
- Good meshing practices
- Using a custom hydrodynamic mesh feature
- Creating a subsurface mooring
- Difference between hydrodynamic and visualization meshes

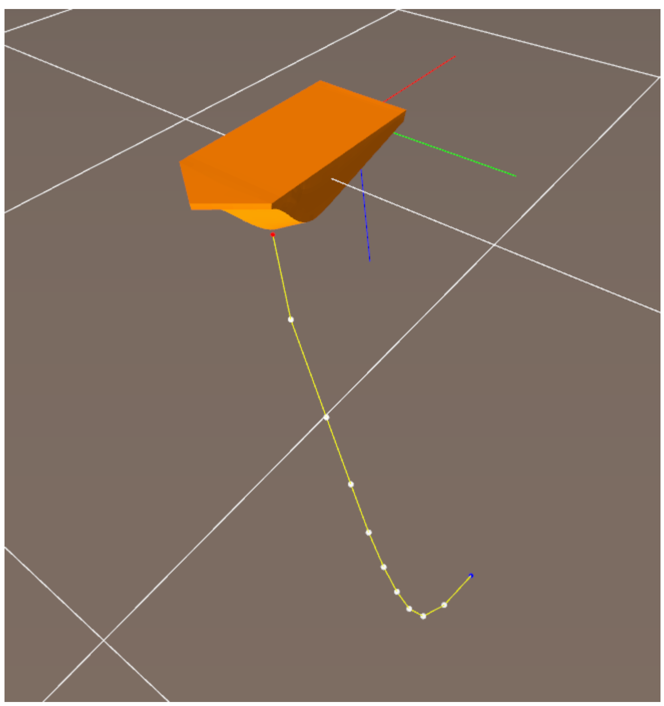


Figure 23: Custom mesh with mooring line

2 Acceptable mesh format

- ☞ The RigidBodyCustomMesh feature in ProteusDS will behave much like the cylinder feature but has the ability to model a much more complex geometry.
- ☞ All custom meshes used in ProteusDS must be in polygonal mesh format.
- ☞ ProteusDS accepts several 3D shape files. Most common are: *.3ds and *.obj. Most 3D CAD packages can import/output these types of files. Most *.obj files have an associated material (*.mtl) file that describes the visual aspects of the polygons.
- ☞ Common 3D CAD packages include Blender, Rhinoceros 3D, and Wings3D.

3 Good meshing practices

- ☞ When creating a custom mesh for use in ProteusDS it is important to have a reasonable amount of polygons on each surface. Too many polygons can cause the simulation to slow down due to the increase in computations and too few can cause unrealistic behavior due to an approximation over the large polygon.
- ☞ ProteusDS requires each polygon normal vector to point outwards from the entire mesh.
- ☞ A degenerate polygon will not cause an error but a warning will be issued and it is recommended that any degenerate polygon be fixed.
- ☞ More tips on proper meshing practices can be found in the ProteusDS Manual.

4 Using a custom mesh in PST

- ✪ Create a new RigidBody DObject and create a RigidBodyCustomMesh feature.
- ✪ Make sure the supplied hydrodynamic mesh files are in the project directory ('nomad_hydromesh.obj', 'nomad_hydromesh.mtl').
- ✪ Update the $CustomMeshFile property in the custom mesh feature to the 'nomad_hydromesh.obj' file stored in the project directory.
- ✪ Reference the custom mesh feature in the RigidBody input file by adding the $CustomMesh property. Use the name of the feature and place the feature frame coincident with the RigidBody frame.
- ✪ Set the RigidBody IC file to have a displacement of 1m along the z axis with respect to the global reference frame. By design the custom mesh will rest with the frame origin above the water surface, this displacement will prevent the RigidBody from falling into the water at the beginning of the simulation.
- ✪ Give the Nomad buoy RigidBody the following physical properties:

Nomad physical properties

```
// Mechanical
$Ix 31e3
$Iy 95e3
$Iz 72e3
$Ixy -125
$Ixz 3.3e4
$Iyz -25
$Mass 9000
```

```
$CGPosition -2.62 0 -1.57
```

☞ By adjusting the $CGPosition property, the RigidBody center of gravity can be moved relative to the RigidBody frame.

✪ Run a 1 second simulation and view the results in PostPDS to ensure the custom mesh file is correctly imported.

5 Connecting a mooring cable

✪ Create a new cable DObject with synthetic cable properties from a previous tutorial.

✪ Create a point connection for the RigidBody DObject and the Cable DObject.

✪ Connect node 0 of the cable to (-4,0,0) relative to the RigidBody local coordinate frame by using the $DCableFollowerLocation property in the RigidBody/Cable connection. This connects the cable 4m in the negative x direction away from the RigidBody local origin.

✪ The IC for the cable should be defined as (-4,0,-1) for node 0, and (10,0,50) for node N. Use a length of 55m and 10 elements.

6 Running the simulation and visualizing in PostPDS

✪ Set the length of simulation to be 30 seconds and run the solver.

✪ View the results in PostPDS. Be sure to right click on the custom mesh object and select 'Display Mesh Edges' to view the custom polygonal mesh used.

Tutorial 23 - Barge mooring

1 Tutorial overview

This tutorial will cover:

- Initialization of a four point barge mooring
- Setting up multiple IC files and connections
- DCable payout tension controllers

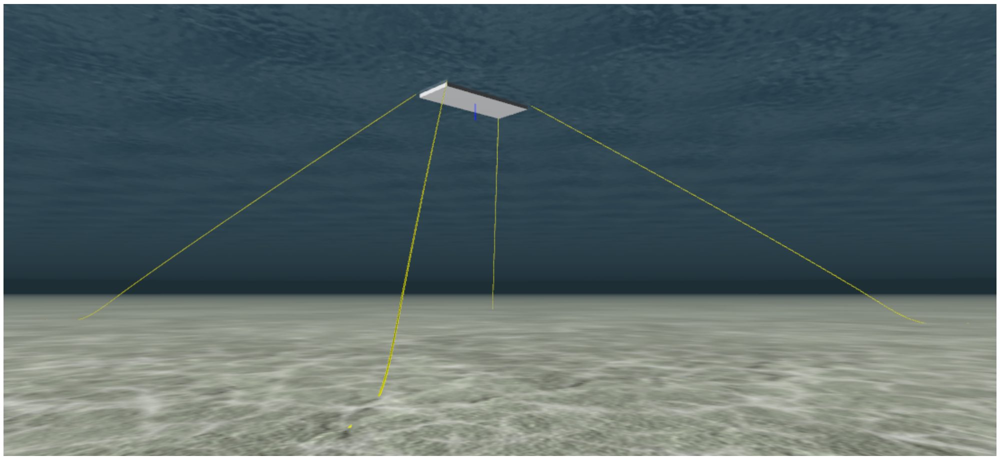


Figure 24: Moored barge

2 Setting up a barge RigidBody DObject

- ✪ Create a new project.
- ✪ Create a new RigidBody DObject that will act as the barge.
- ✪ Create the RigidBodyCuboid feature below and add it to the barge input file.

Barge cuboid feature

```
// Fluid loading
$CDt 0.01
$WindLoading 1
$HydroLoading 1
$BuoyancyFroudeKrylov 1
```

```
// Uncategorized properties
$LengthX 25
$LengthY 9
$LengthZ 1.5
$CAx 1
$CAy 1
$CAz 1
$CDx 1.1 //Approximated for a cube
$CDy 1.1
$CDz 1.1
$SegmentsX 15
$SegmentsY 8
$SegmentsZ 5
```

- ✪ Ensure that wind loading, hydrodynamic loading and Froude-Krylov buoyancy are being applied to the barge cuboid feature.
- ✪ In order to have a barge draft of 0.6m, give the barge a mass of 154 000kg.
- ✪ To provide accurate barge movement, set the moments of inertia to the following: '$Ix 13E5', '$Iy 81E5', '$Iz 96E5'.
- ✪ Set the barge to remain stationary while the cables are pretensioned by setting the $Kinematic flag to 1.

3 Setting the segmented cable properties

- ☞ The barge will be moored to the seabed by 4 mooring lines. Each line will be segmented into 15m of chain and 85m of rope.
- ☞ All 4 mooring cables will be comprised of the same materials and therefor will share a the same cable segment features.
- ✪ The cables are all going to be anchored to the seabed at node 0. Enable the property $Node0Static.
- ✪ Create a DCableSegment feature for chain with the following properties:

Chain feature properties

```
$CDc 2.4
$Cdt 0.1
$CAc 0.5
$Diameter 0.116
$Density 7850
$EA 1.1e8
$EI1 1
$EI2 1
$GJ 0
$CID 100
$BCID   0
$TCID   0
$CE 0
```

- ✪ Create a DCableSegment feature for rope with the following properties:

Rope feature properties

```
$CDc 1
$Cdt 0.1
$CAc 1
$Diameter 0.052
$Density 980
$EA 4.3e6
$EI1 1
$EI2 1
$GJ 0
$CE 1
$CID 100
$BCID 0
$TCID 0
```

- ✪ Add the chain and rope cable segment features to each cable input file.
- ✪ Create a 15m chain section and an 85m rope section. The rope feature declaration must be below the chain feature declaration to position the rope after the chain along the span of the cable.

4 Placing the four cables around the barge

- ✪ Set $WaterDepth to 50m.
- ✪ Set the barge RigidBody IC at the global origin (0,0,0)m.
- ☞ Each cable's node N will be connected to a corner of the barge. The initial location of each node N must be located at the corresponding location in the global reference frame.
- ☞ The 4 connection points represented in the global reference frame are (12,4.5,-1)m, (12,-4.5,-1)m, (-12,4.5,-1)m, and (-12,-4.5,-1)m.
- ☞ Each node 0 will be located 70m away from the barge connection point along the X and Y axis and at the seabed (50m).
- ✪ Define the initial conditions of the cables to be a straight line from the anchor position to the corner of the barge.
- ✪ The position of Cable 1, in this case defined as the north-east cable (+X, +Y quadrant) will have a Node 0 position of (70,70,50)m, and a node N position of (12,4.5,-1)m. Use the straight line length as the cable length with 20 elements.
- ✪ Repeat this process with the correct coordinates for the remaining 3 cables.
- ✪ Create a point connection for each cable node N to the barge. The connection location with respect to the rigid body frame will be the same as the global position of the node N because the barge frame is coincident to the global frame at the start of the simulation.

5 Tensioning the cables using the tension controller

- ☞ A tension controller can be used to create realistic initial loads for the cables.
- ✪ Add the property $NodeNPayoutMode to each cable input file and set the property to 2 in order to set a tension controller at node N in pretension mode.
- ☞ In pretension mode, a desired cable tension is specified and the tension controller will either pay out or pull in cable in an attempt to reach the desired tension.

✪ Set the desired cable tension to 10 000N using the property $NodeNPretension.

✪ Set the maximum pay out/pull in speed to 0.5m/s using the property $NodeNPretensionPayoutSpeed.

☞ It is important to set a reasonable maximum payout speed. If the maximum payout speed is too high, the controller will become unstable and may introduce large oscillations in the cable.

✪ After a tension controller is added to all 4 cables, save the project and run the simulation for 5 seconds.

✪ Visualize the results in PostPDS. Plot the cable tension and note the tension controller pays out cable to reduce the tension.

☞ The tension controller will stop paying out once the desired tension value is reached.

6 Simulate the barge mooring in storm conditions

☞ At the end of the simulation, the cables are now in steady state and the tension controllers have achieved an acceptable tension.

✪ Close PST and copy the files from the TerminalIC subfolder into the results subfolder. Replace the files when prompted.

✪ Reopen PST. The ending state vector for each cable is now the initial condition used for the simulation.

✪ Remove the tension controller properties from the Cable input files and the $Kinematic flag from the RigidBody input file.

✪ Introduce some current, wind or waves into the simulation to introduce some environmental loading.

✪ Run the simulation Visualize the results in PostPDS.

✪ Plot the cable tensions.

☞ Notice that the chain section resting on the seabed provides the barge with some dynamic damping.

Tutorial 24 - Cable payout and kinematic control

1 Tutorial overview

This tutorial will cover:

- DCable PID tension controller
- RigidBody Kinematic constraint

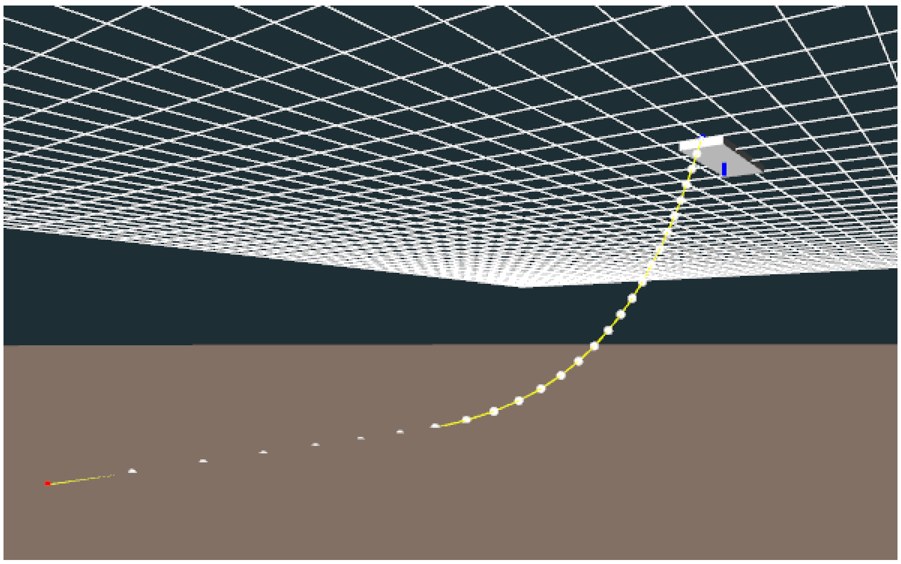


Figure 25: Cable lay operation

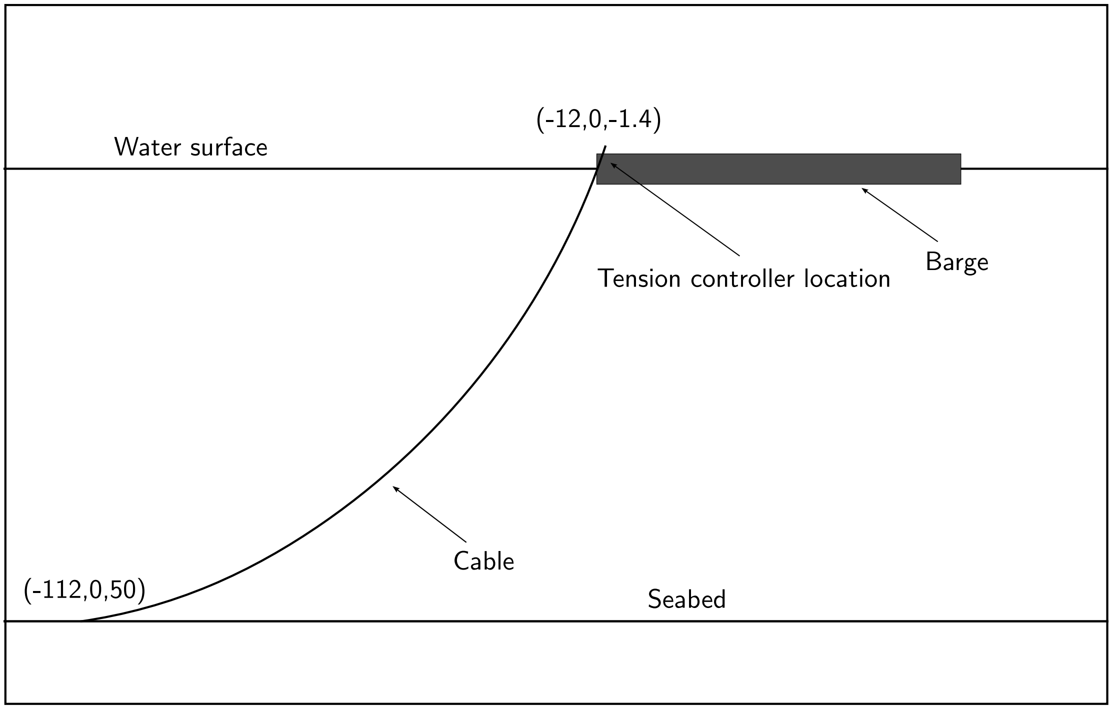


Figure 26: Cable payout layout

2 Adjusting cable connection tension

☞ PID tension controller can be used to represent winch or heave compensation behavior in simulation in various environment conditions.

✪ Create a new project.

✪ Set water depth to 50m.

2.1 Creating a barge

✪ Create a RigidBody that will act as a barge.

☞ The barge dimensions are 25m long, 10m wide, and 1.5m deep with a draft of 0.6m.

✪ Create a cuboid feature called *bargeHull* and set the $LengthX to 25m, $LengthY to 10m and $LengthZ to 1.5m. The other cuboid properties will be ignored as the kinematic flag will be set.

✪ Add the instance of the barge to the RigidBody: '$Cuboid bargeHull 0 0 0 0 0 0'

☞ Set the $Kinematic flag to 1. The barge RigidBody will be set in kinematic mode to move at a constant velocity. In this mode, all external forces are ignored. Mass and moments of inertia can be set to arbitrary values.

☞ The kinematic flag enforces constant velocity based on the declared initial conditions.

✪ In order to get the correct draft and keep the barge stationary until the cable is correctly configured, set the barge IC to a position of (0,0,-0.15)m and a velocity of (0,0,0)m/s.

2.2 Creating a subsea cable

- ✪ Create a new Cable to represent a subsea cable.
- ✪ The subsea cable will have one end fixed to the seabed and the other end connected to the barge. Set $Node0Static to 1 to pin the cable to the seabed.
- ✪ Create a DCableSegment feature for the cable with the following material properties:

Cable feature properties

```
$CDc 1.4
$CDt 0.01
$CAc 1.15
$EA 1e9
$EI1 1.5e6
$EI2 1.5e6
$GJ 2e6
$Diameter 0.13
$Density 3000
$CE 1
$CID 1e3
$BCID 1e1
$TCID 0
```

- ✪ Create a connection between the node N and the barge. The connection should occur 0.5m back from the stern of the barge and 0.5m above the deck at a location in terms of the RigidBody local frame of (-12,0,-1.25).
- ☞ As per the simulation schematic, the cables are 100m away from the winch on the barge in the X,Y plane and 50m deep in the Z direction.
- ✪ Generate the cable IC giving node 0 a position of (-112,0,50) and node N (-12,0,-1.4).
- ☞ Note this location is different from the connection location because the barge is located 0.15m in the -Z direction to account for the barge draft.
- ✪ Use the straight line cable length and set the number of cable segments to 25.
- ☞ In order to begin the simulation with the correct amount of bottom tension, a pretension controller will be used similar to the 'Barge mooring' tutorial.
- ☞ If the desired bottom tension is 10kN, a simple calculation can be performed to determine the top tension in the cable for a water depth of 50m.
- ☞ The gravitational force on the cable in the water will create a tension of 13kN. Therefore, to obtain 10kN at the seabed, the desired top tension will be approximately 23kN.
- ✪ Set a $NodeNPretension tension controller for the cable with a pretension set point of 23kN, similar to 'Barge Mooring' tutorial. Use a payout speed of 0.5m/s.

2.3 Generate pretensioned Cable ICs

- ✪ Set the simulation length of 20 seconds.
- ✪ Run the simulation and view the results. The cable will reach steady state with the desired cable tension.
- ✪ Once the cable has reached a steady state position with the desired top and bottom tension, close PST and move the terminal IC files into the project folder so that the desired cable position is saved then reopen PST.

3 Initializing the PID tension controller

- ✪ Remove the $NodeNPayout properties from the Cable input file.
- ✪ To create a PID tension controller, add a new DCableTensionController DObject.
- ✪ Connect the tension controller to the cable using the tension controller as the master.
- ✪ In the connection properties, change the $DCableFollowerNodeN property to 1, as node N is the end of the cable at which the tension controller will act by rendering or recovering cable.
- ✪ In the tension controller input file, set: '$TensionSetPoint 13000'
- ✪ Change the PID properties to have only proportional (P) gain. Set $Ki and $Kd to 0. Set $Kp to 1e3. Leave $MaxPayoutSpeed at 0.5m/s.

4 Setting barge forward speed

- ✪ In order to move the barge at a constant velocity, the $Kinematic flag will be used. Ensure that the $Kinematic flag is still enabled in the barge input file.
- ✪ To give the barge a constant velocity, set the barge IC to have a position of (0,0,-0.15)m and a velocity of (0.25,0,0)m/s. The barge will now travel at a constant velocity of 0.25m/s along the x direction, ignoring all external forces acting upon it.
- ✪ Run the simulation for 30s and visualize the results.
- ✪ Plot the tensions at element 25 using the PostPDS plotting function.

5 Adding integral (I) tension control

- ☞ Since only the P gain was used in the controller, significant noise is present. This could be mitigated by reducing the maximum payout speed. However, if the maximum payout speed is near the speed of the barge the controller may not be able to payout cable fast enough.
- ☞ The P controller is very useful for cable payout. However, whenever the cable length becomes long enough and a new cable element is created by inserting a node, the tension controller may strongly react to small fluctuations in tension. This can be avoided by adding a small amount of I gain.
- ✪ Set the $Ki property to 100. Note that too much I gain can cause severe overshoot and lag in the controller response.
- ✪ Rerun the simulation and plot the tension to see the difference that the I gain makes.
- ☞ The added I gain should reduce the amount of noise in the control system.
- ☞ PID controllers with D gain may cause instabilities due to the noise of the system. Caution should be used when adding D gain to a control system.

Tutorial 25 - ABA connection

1 Tutorial overview

This tutorial covers:

- ABA connections
- Prismatic ABA Joints
- ABA joint features

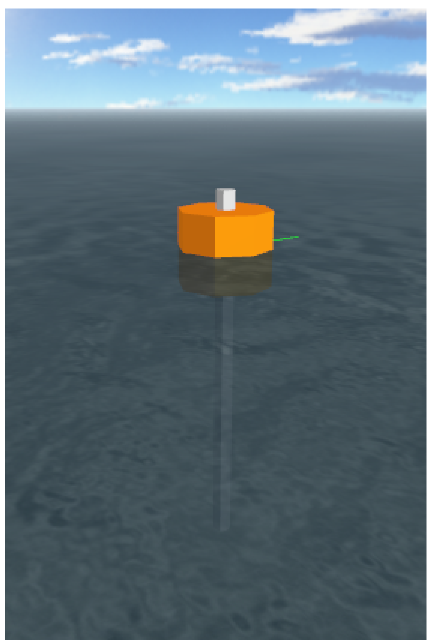


Figure 27: Monopile ABA

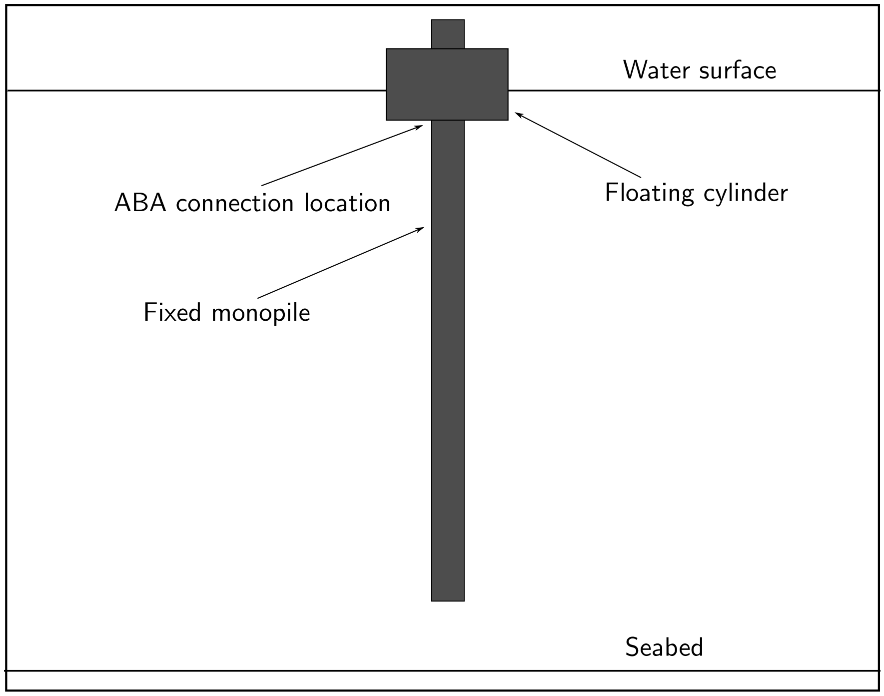


Figure 28: Monopile ABA layout

2 Setting up as monopile simulation

- ✪ Create a new project in PST.
- ✪ Ensure water depth to 17m.
- ✪ Create a new RigidBody. This object will represent a pile fixed in place.
- ✪ Create a cylinder feature for the new RigidBody. Set the cylinder length to 20m and diameter to 1m.
- ✪ To fix the rigid body in space, set '$Kinematic 1' in the input file. This setting indicates all forces will be ignored and a constant velocity will be maintained.
- ✪ Set the IC of the rigid body on the seabed at 17m depth. By design, the rigid body frame is located at the bottom of the pile. Set the initial position to (0,0,17)m and initial velocity and orientation angles to 0.
- ☞ By design, the mass properties are set such that the pile will be neutrally buoyant when 85% submerged.
- ✪ Set the *pile* properties as shown below.

Pile RigidBody properties

```
$Mass 1.4e4
$Ix 1e6
$Iy 1e6
$Iz 1e6
$Ixy 0
$Ixz 0
$Iyz 0

$CGPosition to 0 0 -3
$DefineInertiaAboutCG 1
```

- ✪ Create a second RigidBody. This is the float and will follow the motion of the ocean waves.
- ✪ Create a cylinder feature for the float rigid body. Set the cylinder length to 4m and diameter to 5m.
- ☞ By design, the float will be 50% submerged in calm conditions.

Float RigidBody properties

```
$Mass 4e4
$Ix 4e4
$Iy 4e4
$Iz 4e4
$Ixy 0
$Ixz 0
$Iyz 0
```

- ✪ Reference the appropriate cylinder feature in each rigid body input file.
- ☞ The IC of the second rigid body will be determined by the ABA joint and will be discussed in the next section.

3 Creating an ABA joint

- ☞ Rigid mechanical systems can be modeled using the Articulated Body Algorithm (ABA). This is realized in ProteusDS using the ABA connection. ABA connections can be prismatic, revolute, cylindrical, planar, spherical, or universal. Each joint type has different degrees of freedom.
- ☞ Each joint degree of freedom can have stiffness, damping, endstops, and other properties.
- ☞ The rigid bodies are kinematically constrained to each other by the joint. A joint reference frame position and orientation must be specified relative to each rigid body frame in the connection. The deflection of the joint indicates how these joint reference frames have moved apart from one another.
- ☞ In this simulation a prismatic ABA connection will be used. This linear motion joint allows the float to follow ocean surface motion in heave as the waves pass. The float will be fixed to the pile cylinder using a prismatic ABA connection.
- ✪ Create a connection with the pile cylinder as the master object and the floating cylinder as the follower object. Chose the ABA connection type.
- ✪ By design, the pile will be the master. In the connection section, the $MasterConnectionLocation flag determines the joint frame connection position relative to the master object rigid body local coordinate frame. Since the pile rigid body frame is located at the bottom of a 20m long cylinder, the connection location is set to (0,0,-17) to place the connection at the top of the pile adjacent to the float.
- ✪ The joint connection location on the float is coincident with the rigid body frame. Leave '$FollowerConnectionLocation 0'.
- ✪ Set '$Joint 0' to specify a prismatic joint.
- ✪ The $FollowerJointAxis property determines what axis of the joint reference frame is used for the joint degree of freedom. Set the property to 2 for the joint frame z axis.
- ✪ An ABA connection joint feature is automatically created called *jointProperties*. Navigate to the feature section in the Library and select the new ABA feature.
- ✪ The initial conditions of the follower float requires special ABA values. Because a prismatic joint has one degree of linear freedom, an initial velocity and position must be specified. These can both be set to 0.
- ✪ Activate ocean waves in the simulation to show the float rising and falling with the ocean surface. An Airy wave with the following properties will show the heave motion of the float clearly.

Airy wave properties

```
$WaveHeight 2
$WavePeriod 8
$WaveHeading 0
```

4 Running the simulation

- ✪ Run the simulation for 50 seconds and visualize the results in PostPDS.
- ☞ Note the float only moves in the heave degree of freedom and that the pile is fixed.
- ☞ The float may have some initial transient motion, but eventually converges to follow the ocean waves.
- ☞ No RigidBody to RigidBody contact is modeled in ProteusDS. This makes it easy to simplify advanced joints and allow modeling with basic shapes as opposed to advanced pieces.

5 Changing ABA connection properties

- ✪ Remove the $Kinematic flag from the pile, change the water depth to 50m, and re-run the simulation.
- ☞ Notice the large heave motion of the pile. This can be reduced by applying specific ABA connection properties.
- ✪ Apply an endstop location of 5m to the ABA connection by setting '$EJoint 5'.
- ✪ Apply a joint damping of 1e3 by setting '$CJoint 1e3'
- ✪ Apply a joint endstop stiffness and damping of 1e6 and 1e6 by using the $KEJoint and $CEJoint properties.
- ✪ Rerun the simulation and observe the effect these values have on the system.

Tutorial 26 - Wave energy capture device 1

1 Tutorial overview

This tutorial covers:

- Using multiple ABA connections in a chain sequence
- The revolute ABA joint

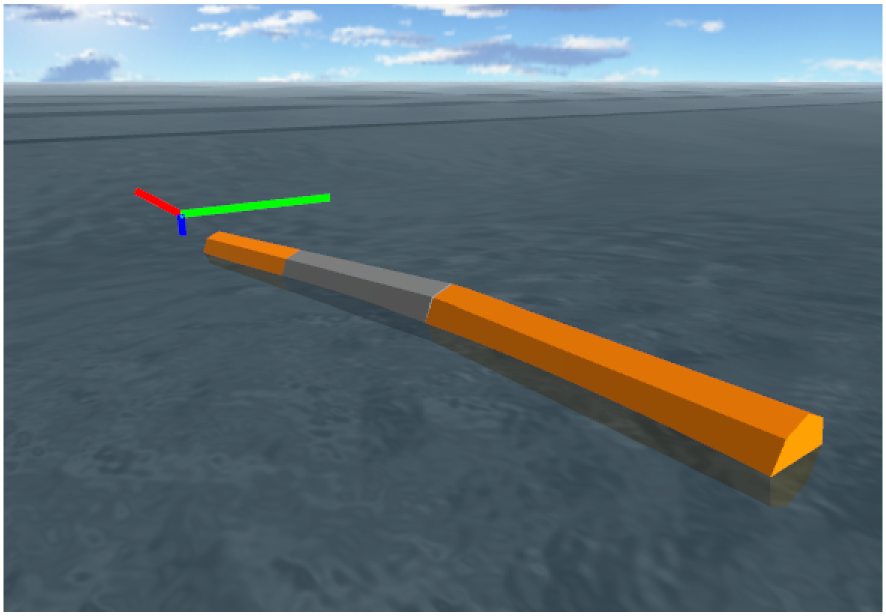


Figure 29: Wave energy converter

2 Initializing the WEC system RigidBody DObjects

✪ Create a new project.

✪ Create three RigidBody DObjects and call them *float0*, *float1*, and *float2*.

☞ The wave energy converter will be comprised of three horizontal cylinder floats of 2m diameter and 10m long connected end to end by revolute pin joints.

✪ Add a RigidBodyCylinder feature to the library and set the $Diameter property to 2m and $Length property to 10m.

- By design, the WEC cylinder floats will be 50% submerged in calm conditions. Set the $Mass property to 1.6e4kg. The moment of inertia values are set arbitrarily to 1e4kg*m^2. To prevent the cylinders from rolling, set the $CGPosition to have a 1m offset in the z direction.
- In order to have the inertia of the cylinders defined about the center of gravity, enable the $DefineInertiaAboutCG property.
- Add a cylinder to each of the RigidBodies. To make the cylinders lie flat on the water, the feature frame has to be pitched by 90 degrees in the y direction.

3 Creating a chain of ABA connections

- Create an ABA connection and make *float0* master of *float1*.
- Create a second ABA connection and make *float1* master of *float2*.
- Change the connection information for both connections to have the $Joint property set to type 1, which creates a revolute joint.
- Set the $FollowerJointAxis property to 1 for both connections. This sets the revolute axis to the y-axis.
- In the connection section, change the connection joint reference type property from the default $PrismaticJointLinear to $RevoluteJointAngular and reference the RigidBodyABAConnectionJoint that has been automatically created.
- In the RigidBodyABAConnectionJoint feature, change the $CJoint property to have a damping of 1e4N*m*s/deg. This will act as the energy capture from the relative motion of the system.
- The cylinders are located end-to-end, so in both connections, specify the $MasterConnectionLocation property to have the offset -5 0 0 m with no rotation, and the $FollowerConnectionLocation property to have the offset 5 0 0 m with no rotation. This places the connection location in between the two cylinders.
- Set the IC for *float0* to be at the origin with no velocity or rotation. Set the IC for *float1* and *float2* to have an ABA IC with one degree of freedom and a velocity of 0m/s and a angle of 0 degrees.

4 Running the simulation and visualizing in PostPDS

- Set Airy waves with height of 3m and a period of 7 seconds.
- Set the water depth to 50m.
- Run the simulation for 60 seconds.
- Visualize the results in PostPDS.

Tutorial 27 - Wave energy capture device 2

1 Tutorial overview

This tutorial covers:

- Using multiple ABA connections in a chain sequence
- The revolute ABA joint

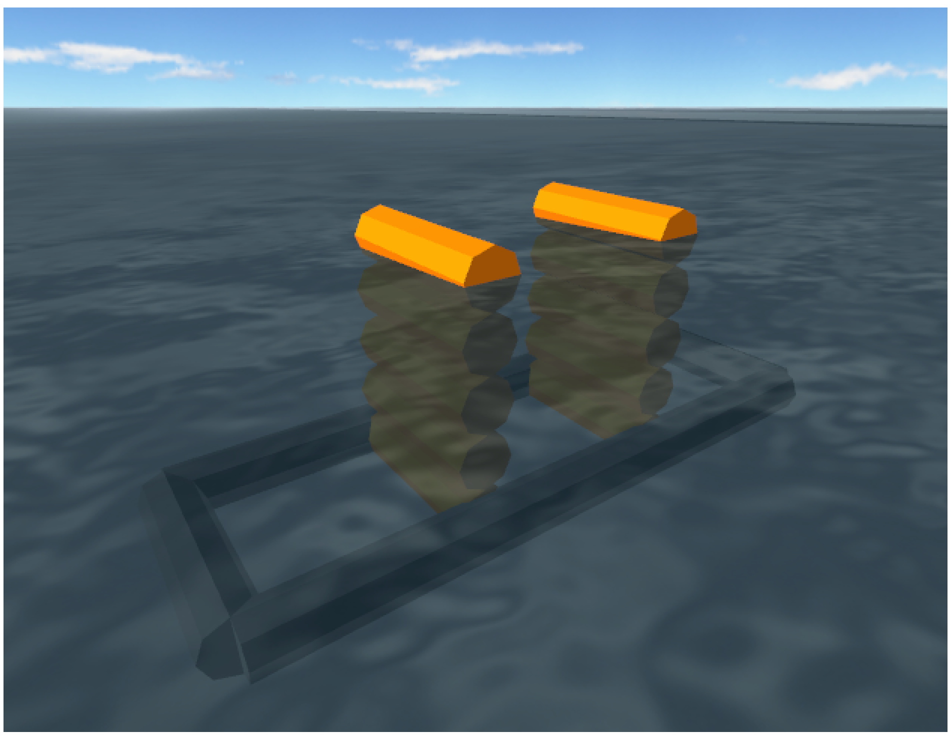


Figure 30: Wave energy converter

2 Initializing the WEC system RigidBody DObjects

- ✪ Create a new project.
- ✪ Create three RigidBody DObjects and call them *base0*, *paddle0*, and *paddle1*.

- ☞ The wave energy converter will be comprised of a submerged frame and two buoyant paddles attached with revolute pin joints.
- ☞ It is assumed the submerged frame is flooded and therefore has no buoyancy.
- ✪ Add a RigidBodyCylinder feature called *paddlePipe* to the library, set the $Diameter to 1m, $Length to 3.5m
- ✪ Add a RigidBodyCylinder feature called *longPipe* to the library, set the $Diameter to 1m, $Length to 12m, and $BuoyancyFroudeKrylov to 0.
- ✪ Add a RigidBodyCylinder feature called *shortPipe* to the library, set the $Diameter to 1m, $Length to 4m, and $BuoyancyFroudeKrylov to 0.
- ✪ Set mass moments of inertia of *base0* arbitrarily to 1e5 kg*m^2 and the mass to 2e4 kg.
- ✪ Set mass moments of inertia of *paddle0* and *paddle1* arbitrarily to 1e3 kg*m^2 and the mass to 1e3 kg.

3 Adding cylinders to base0

- ✪ Add the cylinders to *base0*. Assume the rigid body frame is located in the middle of one of the short pipes as indicated in 31.
- ✪ To add the cylinder features to the RigidBody, add the following lines to the RigidBody input file.

RigidBody properties

```
$Cylinder shortPipe 0 0 0 90 0 0
$Cylinder shortPipe -12 0 0 90 0 0
$Cylinder longPipe -6 2 0 0 90 0
$Cylinder longPipe -6 -2 0 0 90 0

$CGPosition -6 0 0 //Set the center of gravity to act in the geometric
// center of the cylinders, not at the RigidBody origin.
$DefineInertiaAboutCG 1 //Ensure that the moments of inertia are
// acting about the new center of gravity location.
```

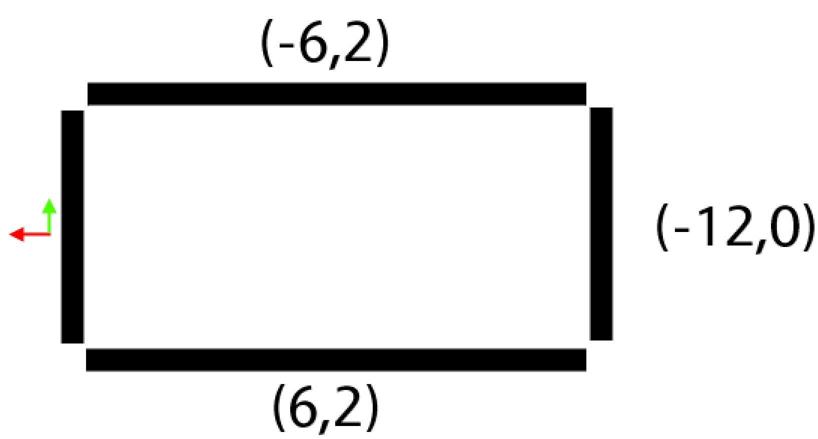


Figure 31: Wave energy converter

4 Adding cylinders to paddle0 and paddel1

☞ By design, the rigid body frame will be set at the bottom of the paddles on the rotation axis.

✪ Add the cylinders to *paddle0* and *paddle1*.

paddle0 and paddle1 properties

```
$Cylinder paddlePipe 0 0 0 90 0 0
$Cylinder paddlePipe 0 0 -1 90 0 0
$Cylinder paddlePipe 0 0 -2 90 0 0
$Cylinder paddlePipe 0 0 -3 90 0 0
$Cylinder paddlePipe 0 0 -4 90 0 0

$CGPosition -2 0 0
$DefineInertiaAboutCG 1
```

5 Creating ABA connections

✪ Create an ABA connection and make *base0* master of *paddle0*.

✪ Create a second ABA connection and make *base0* master of *paddle1*.

✪ Change the connection information for both connections to set the $Joint property as type 1. This creates a revolute joint.

✪ Set the $FollowerJointAxis property to 1 for both connections. This sets the revolute axis to the y-axis for the follower RigidBody.

✪ Create a RigidBodyABAConnectionJoint feature called *paddleConnection*.

✪ In the connection section, change the connection joint reference type property from the default $PrismaticJointLinear to $RevoluteJointAngular and reference the RigidBodyABAConnectionJoint feature created.

☞ The paddles are located 1/3 and 2/3 along the span of the *base0* hull

✪ For *paddle0*, specify the $MasterConnectionLocation property to have the offset -4 0 0 with no rotation and the $FollowerConnectionLocation property to have the offset 0 0 0 with no rotation

✪ For *paddle1*, specify the $MasterConnectionLocation property to have the offset -8 0 0 with no rotation and the $FollowerConnectionLocation property to have the offset 0 0 0 with no rotation

✪ Set the IC for *base0* to be a location of 0 0 3 m with no velocity or rotation.

✪ Set the IC for *paddle0* and *paddle1* to have an ABA IC with one degree of freedom and a velocity of 0deg/s and a angle of 0 degrees.

6 Running the simulation and visualizing in PostPDS

✪ Set Airy waves with height of 3m and a period of 7 seconds.

✪ Set the water depth to 50m.

✪ Run the simulation for 60 seconds.

- Visualize the results in PostPDS.

Tutorial 28 - Simple turbine feature

1 Tutorial overview

This tutorial covers:

- Creating a turbine feature

2 Creating the base rigid body

✪ Create a new project.

✪ Create a rigid body DObject and call it *platform*.

✪ Set '$Kinematic 2'.

✪ Define the initial condition z position to -50 (in the air).

☞ This will represent the stationary portion of the turbine.

3 Creating a turbine feature

✪ Create a rigid body turbine feature by selecting 'RigidBody Turbine' in the 'Add New Library Feature' drop down menu.

✪ Call the new turbine feature *turbine_30m*.

☞ Within the turbine feature library item the torque and thrust tables must be created.

☞ These tables will apply the specified torque or thrust at a given tip speed ratio (tsr) and linearly interpolate between them.

✪ Update the turbine properties as indicated below.

✪ Add the turbine feature to the rigid body using '$Turbine turbine_30m 0 0 0 0 0 0'.

☞ By enabling the '$ReferenceAngularVelocity' property the angular velocity is set to a constant rate and not associated with a rotating rigid body.

Turbine feature properties

```
// Mechanical
$TorqueCoefficient 0 0.1
$TorqueCoefficient 4.5 1
$TorqueCoefficient 8 0
```

```
$ThrustCoefficient 0 0
$ThrustCoefficient 4.5 0.5
$ThrustCoefficient 8 0.5

$ReferenceSweptArea 700
$ReferenceDiameter 30
$IsCrossFlowTurbine 0
$RotationAxis 0

$Mode 1
$ReferenceAngularVelocity 171.89
```

4 Simulate the turbine in constant wind

✪ In the environment input, set a constant uniform wind with speed of 10m/s as indicated below.

Constant wind

```
// Wind
$WindSpectrum 0
$WindProfile 0
$WindHeading 0
$WindSpeed 10
```

✪ Run the simulation

☞ With this wind speed and reference angular velocity, the tip speed ratio is 4.5.

☞ Since this is an axial flow turbine, the expected thrust and torque values are 23kN and in the X direction and 677kNm about the X axis, respectively.

✪ Confirm loading on the platform in the forces.dat file in the Results folder

☞ Note the turbine feature only provides loading; there is no spinning rotor to visualize.

Tutorial 29 - ABA turbine feature

1 Tutorial overview

This tutorial covers:

- Creating a turbine feature
- Revolute ABA connection
- Applying wind loads

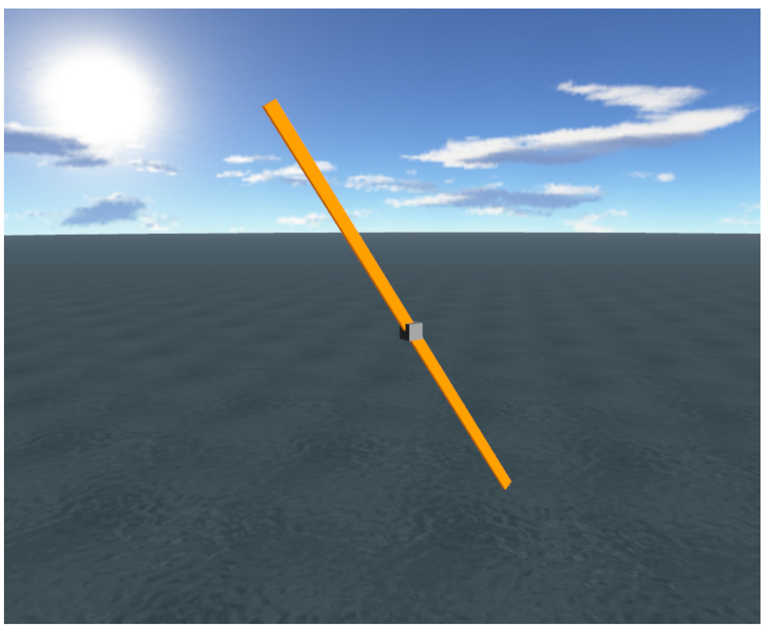


Figure 32: Simple turbine model

2 Creating the base rigid body

- ✪ Create a new project.
- ✪ Create a rigid body DObject and call it *platform*.
- ✪ Set '$Kinematic 1'.
- ✪ Set '$KinematicForceCalculations 1'.

- ✪ Define the initial condition z position to -50 (in the air).
- ☞ This will represent the stationary portion of the turbine.
- ✪ Give the base a visual block cuboid feature coincident with the rigid body frame.
- ☞ The default values in the cuboid feature will not matter because the body is kinematically constrained.

3 Creating the rotating rigid body

- ✪ Create a rigid body DObject and call it *rotor*.
- ✪ Set the mass and the moments of inertia to $1e5$.
- ☞ This will represent the rotating portion of the turbine.
- ✪ Create a cuboid feature called *blade* as a visual representation of the rotor.
- ✪ Disable all drag and added mass properties in the cuboid feature as the thrust and torque will be produced by the turbine feature.
- ✪ Since the rotor is placed in the air set '$WindLoading 0'.
- ✪ Make the blade 0.25m along the x axis, 30m along the y axis, and 0.75m along the z axis.
- ✪ Add the cuboid feature to the rotor using '$Cuboid blade 0 0 0 0 0 0'.

blade properties

```
// Fluid loading
$CDt 0

// Uncategorized properties
$LengthX 0.25
$LengthY 30
$LengthZ 0.75
$CAx 0
$CAy 0
$CAz 0
$CDx 0
$CDy 0
$CDz 0
$SegmentsX 5
$SegmentsY 5
$SegmentsZ 5
$WindLoading 0
```

4 Connecting the rigid bodies via ABA connection

- ☞ To connect the platform and rotor a revolute ABA joint is required.
- ✪ Set *rotor* as the follower to *platform* with an ABA connection.
- ✪ Define the connection under the 'Connections' heading to be revolute about the x axis.
- ☞ A default joint properties feature is automatically created when an ABA connection is made.

- ✪ In the feature library, update the joint damping coefficient to '$CJoint 2.256e5'.
- ✪ Set the initial conditions of *rotor* to ABA with an initial position and rotation rate of 0.

Revolute ABA connection about x axis

```
// Mechanical
$MasterConnectionLocation 0 0 0 0 0 0
$FollowerConnectionLocation 0 0 0 0 0 0
$Joint 1
$FollowerJointAxis 0


// Uncategorized properties
$RevoluteJointAngular jointProperties
```

5 Creating a turbine feature

- ✪ Create a rigid body turbine feature by selecting 'RigidBody Turbine' in the 'Add New Library Feature' drop down menu.
- ✪ Call the new turbine feature *turbine_30m*.
- ☞ Within the turbine feature library item the torque and thrust tables must be created.
- ☞ These tables will apply the specified torque or thrust at a given tip speed ratio (tsr) and linearly interpolate between them.
- ✪ Update the turbine properties as listed below.

Turbine feature properties

```
// Mechanical
$TorqueCoefficient 0 0.1
$TorqueCoefficient 4.5 1
$TorqueCoefficient 8 0

$ThrustCoefficient 0 0
$ThrustCoefficient 4.5 0.5
$ThrustCoefficient 8 0.5

$ReferenceSweptArea 700
$ReferenceDiameter 30
$IsCrossFlowTurbine 0
$RotationAxis 0
```

- ✪ Add the turbine feature to the *rotor* rigid body.
- ✪ Add the blade feature to the *rotor* rigid body.
- ☞ The input file to the *rotor* should look like the listing below.

Rotor input

```
// Mechanical
$Ix 1e5
$Iy 1e5
```

```
$Iz 1e5
$Ixy 0
$Ixz 0
$Iyz 0
$Mass 1e5

$Cuboid blade 0 0 0 0 0 0

$Turbine turbine_30m 0 0 0 0 0 0
```

6 Simulate the turbine in constant wind

- ✪ In the environment input, set a constant uniform wind with speed of 10m/s as indicated below.

Constant wind

```
// Wind
$WindSpectrum 0
$WindProfile 0
$WindHeading 0
$WindSpeed 10
```

- ✪ Run the simulation for 10 seconds
- ☞ The damping coefficient is calculated to match the turbine torque at an angular velocity of 171deg/s
- ☞ With this wind speed and the expected steady state angular velocity, the tip speed ratio is 4.5.
- ☞ The turbine feature uses the revolute joint angular velocity to compute the tip speed ratio
- ☞ Since this is an axial flow turbine, the expected thrust and torque values are 23kN and in the X direction and 677kNm about the X axis, respectively.
- ✪ Confirm loading on the platform in the forces.dat file in the Results folder
- ☞ Note the turbine rotor is now visualized by the spinning rotor and accelerates in the first portion of simulation.
- ✪ Run a simulation for 10 seconds and visualize the rotor spinning.
- ✪ Open the forces output data file and compare the values to expected results.
- ✪ Plot the roll velocity of the *rotor* and note the acceleration during the effect of acceleration during the first portion of the simulation.

Tutorial 30 - Turbine platform

1 Tutorial overview

This tutorial covers:

- Creating a turbine feature
- Revolute ABA connection
- Creating a moored turbine platform

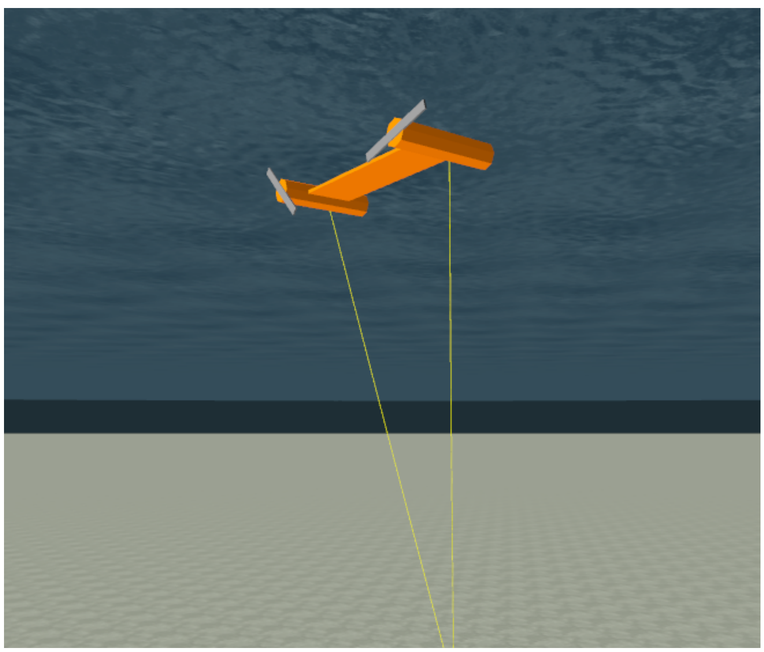


Figure 33: Simple turbine platform model

2 Creating the platform rigid body

- ✪ Create a new project.
- ✪ Create a rigid body DObject and call it *platform*.
- ✪ Set the mass and the moments of inertia to $1e4$.
- ✪ To help with roll and pitch stability, set $CGPosition001$ and also define $DefineInertiaAboutCG1$.

- ✪ Define the initial condition z position to 20 and the intital x position to 100.
- ☞ The turbine platform will consist of 3 features; 2 side cylinders and a center cuboid.
- ✪ Create a cuboid feature called *frame*.
- ✪ Set the length dimensions using '$LengthX 3', '$LengthY 12',and '$LengthZ 0.2'.
- ☞ The remaining cuboid feature properties can be left as default.
- ✪ Create a cylinder feature called *stator*.
- ✪ Set the length and diameter dimensions using '$Length 6' and '$Diameter 1.5'.
- ☞ The remaining cylinder feature properties can be left as default.
- ✪ Add the *frame* feature to the rigid body coincident to the local rigid body frame.
- ✪ Add 2 *stator* features to the rigid body $\pm$6m along the Y axis and rotate each *stator* 90 degrees about the Y axis.

3 Creating the rotors

- ✪ Create 2 rigid body DObjects and call then *rotor0* and *rotor1*.
- ✪ Set the mass and the moments of inertia to $1e3$ and $1e2$, respectively.
- ✪ Create a cuboid feature called *blade* as a visual representation of the rotor.
- ✪ Disable all loading properties in the cuboid feature as the thrust and torque will be produced by the turbine feature.
- ✪ Set '$WindLoading 0', '$BuoyancyFroudeKrylov 0', '$HyrdoLoading 0', and '$SoilLoading 0'.
- ✪ Make the blade 0.1m along the x axis, 5m along the y axis, and 0.5m along the z axis.
- ✪ Add the cuboid feature to each rotor using '$Cuboid blade 0 0 0 0 0 0'.

blade properties

```
$BuoyancyFroudeKrylov 0
$HydroLoading 0
$WindLoading 0

// Fluid loading
$CDt 0.1

// Uncategorized properties
$LengthX 0.1
$LengthY 5
$LengthZ 0.5
$CAx 1
$CAy 1
$CAz 1
$CDx 1
$CDy 1
$CDz 1
$SegmentsX 5
$SegmentsY 5
$SegmentsZ 5
```

4 Connecting the rigid bodies via ABA connection

☞ To connect the frame and each rotor, a revolute ABA joint is required.

✪ Set *rotor0* as the follower to *platform* with an ABA connection.

✪ Define the connection under the 'Connections' tab to be revolute about the x axis.

✪ Place the platform connection location at 3,6,0 with respect to the platform frame.

☞ A default joint properties feature is automatically created when an ABA connection is made.

✪ In the feature library, update the joint damping coefficient to '$CJoint 1537'.

✪ Set the initial conditions of *rotor0* to ABA with an initial position and rotation rate of 0.

Revolute ABA connection about x axis

```
// Mechanical
$MasterConnectionLocation 3 6 0 0 0 0
$FollowerConnectionLocation 0 0 0 0 0 0
$Joint 1
$FollowerJointAxis 0


// Uncategorized properties
$RevoluteJointAngular jointProperties
```

✪ Repeat the above process for *rotor1* but place the connection at 3,-6,0 with respect to the platform frame.

5 Creating a turbine feature

✪ Create a rigid body turbine feature by selecting 'RigidBody Turbine' in the 'Add New Library Feature' drop down menu.

✪ Call the new turbine feature *turbine_2m*.

☞ Within the turbine feature library item the torque and thrust tables must be created.

☞ These tables will apply the specified torque or thrust at a given tip speed ratio (tsr) and linearly interpolate between them.

✪ Update the turbine properties as listed below.

Turbine feature properties

```
// Mechanical
$TorqueCoefficient 0 0.1
$TorqueCoefficient 1 0.5
$TorqueCoefficient 2 1
$TorqueCoefficient 3 1.5
$TorqueCoefficient 6 0

$ThrustCoefficient 0 0.1
$ThrustCoefficient 1 0.5
$ThrustCoefficient 2 1
$ThrustCoefficient 3 1.5
```

```
$ThrustCoefficient 4 1.5

$ReferenceSweptArea 3
$ReferenceDiameter 2
$IsCrossFlowTurbine 0
$RotationAxis 0
```

✪ Add the turbine feature to the *rotor0* rigid body.

✪ Add the turbine feature to the *rotor1* rigid body.

☞ The input file to either *rotor0* or *rotor1* should look like the listing below.

Rotor input

```
// Mechanical
$Ix 1e2
$Iy 1e2
$Iz 1e2
$Ixy 0
$Ixz 0
$Iyz 0
$Mass 1e3

$Turbine turbine_2m 0 0 0 0 0 0

$Cuboid blade 0 0 0 0 0 0
```

6 Create the mooring and bridles

✪ Add a new cable called *mooring*.

✪ Place node 0 at (100,0,50)m and node N at (0,0,500)m. Give the cable a length of 460m with 10 elements.

✪ Set node N static to simulate an ideal anchor.

☞ Node 0 of *mooring* will connect to 2 bridles which will be attached to either end of the platform.

✪ Update the library feature *segmentName* so that it matches the following listing.

```
// Fluid loading
$CDc 1
$CDt 0.01
$CAc 1

// Mechanical
$EA 1e8
$EI1 100
$EI2 100
$GJ 0
$Diameter 0.025
$Density 7700
$CID 5000
$BCID 100
$TCID 0
```

```
$CE 1
```

- ✪ Add 2 cables to the project called *bridle0* and *bridle1*.
- ✪ For *bridle0*, place node 0 at (100,6,20)m and node N at (100,0,50)m. Give the cable a length of 30 with 3 elements.
- ✪ For *bridle1*, place node 0 at (100,-6,20)m and node N at (100,0,50)m. Give the cable a length of 30 with 3 elements.
- ✪ Use the default values in the cable properties file.
- ✪ Connect each bridle node N to node 0 of *mooring*.
- ✪ Connect *bridle0* node 0 to *platform* at (-0.5,6,0.75).
- ✪ Connect *bridle1* node 0 to *platform* at (-0.5,-6,0.75).

7 Simulate the platform in constant current

- ✪ In the environment input, set a water depth of 500m and a constant uniform current with speed of 2m/s as indicated below.

Constant current

```
// Current
$CurrentProfile 1
$CurrentHeading 0
$CurrentSpeed 2

// Seabed
$WaterDepth 500
```

- ✪ Run the simulation for 20 seconds
- ☞ The turbine feature uses the revolute joint angular velocity to compute the tip speed ratio
- ☞ The damping coefficient is calculated to match the turbine torque at an angular velocity of 114deg/s
- ☞ With this current speed and the expected steady state angular velocity, the tip speed ratio is 1.0
- ☞ Since this is an axial flow turbine, ideal conditions should produce thrust and torque values are roughly 9.2kN and in the X direction and 9.2kNm about the X axis, respectively, in ideal conditions.
- ☞ Dynamic motion of the platform in rotation reduces the flow through the rotor and so turbine rotation speed and loading is less than the ideal limit
- ✪ Confirm loading on the platform in the forces.dat file in the Results folder
- ☞ The loading should match a time step near the beginning of the simulation, however, as the platform drifts the relative fluid velocity changes.
- ☞ Note ABA joint visualizes the rotation speed of the rotors.

Tutorial 31 - Net panel

1 Tutorial overview

This tutorial covers:

- Creating a net panel
- Net panel edge and node constraints

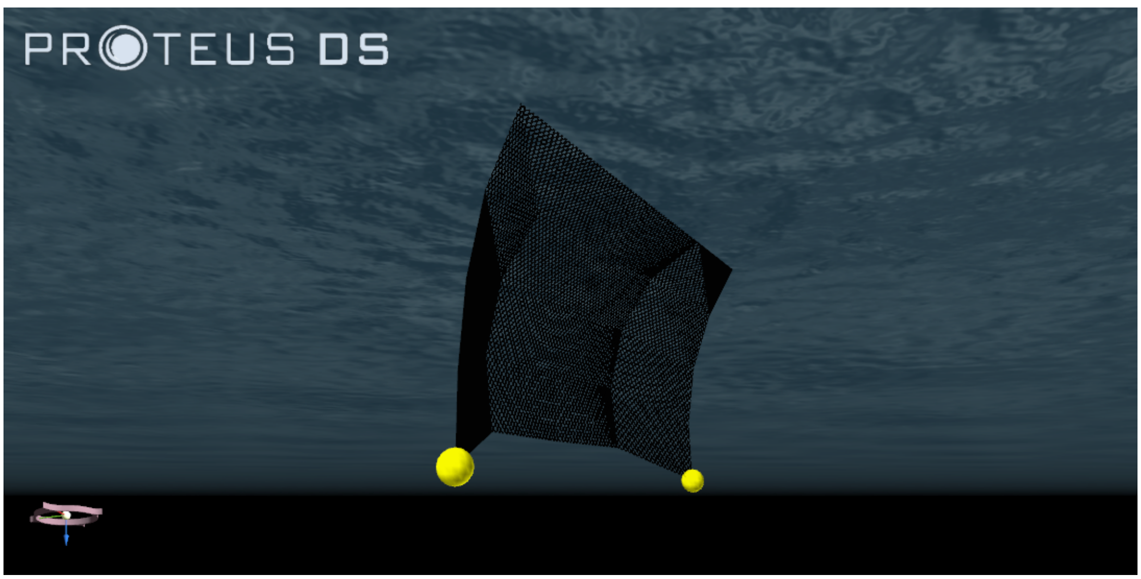


Figure 34: Net panel

2 Creating a net panel

✪ Create a new project

✪ Create a net DObject

☞ A finite-element net DObject contains nodes and elements. The nodes of a net panel are defined as shown Figure 35.

☞ Edges of a panel (edges 0,1,2,3) are defined as shown Figure 35.

✪ Create a net DObject and define its initial condition using the *panel* option according to Figure 36.

- The net properties can be adjusted using the DNetPanel feature in the feature library. For the default DNetPanel feature, change the $FilamentCount property to 50; The filament count corresponds to the inverse of the half mesh size. A filament count of 50 (number of net strands per meter) corresponds half mesh size of 20mm.

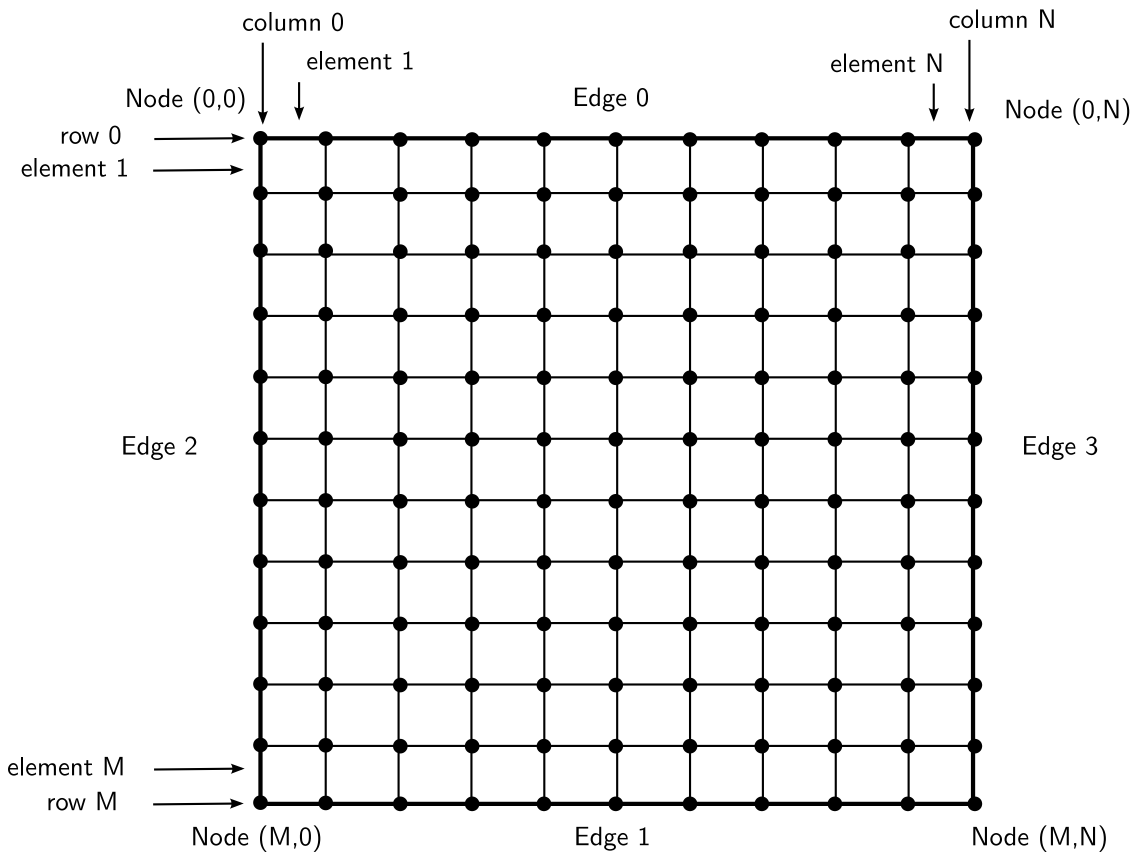


Figure 35: Net panel edge and node definition

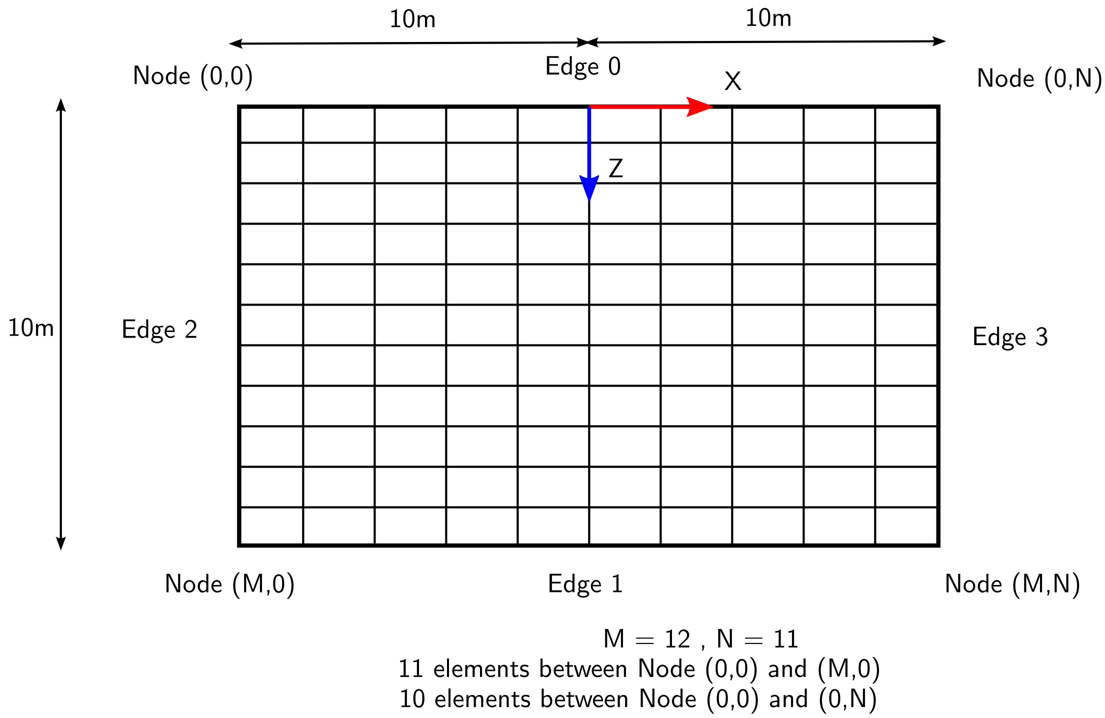


Figure 36: Tutorial net panel schematic

3 Edge constraints

- ✪ Set all of the edge constraints (e.g. $Edge0Static) to 1
- ✪ Add a current velocity in the Y direction ($CurrentHeading 90) of 1 m/s
- ✪ Run a simulation for 10 seconds and watch the deflection of the net.
- ☞ Note that the edges of the net do not move.
- ☞ View the net in engineering mode. Note that the constrained nodes (edges) are shown in in red.

4 Node constraints

- ✪ Remove the edge constraints and set all of the node constraints (e.g. $Node00Static) to 1
- ✪ Run a simulation for 10 seconds and watch the deflection of the net.
- ☞ Note that the corner nodes are constrained.

5 Combined edge and node constraints

✪ Remove all constraints on the net.

✪ Set edge 0 to be static and Node (M,0) and Node (M,N) to be static.

✪ Run a simulation for 10 seconds and note how the net responds to the combined constraints.

6 Adding an ExtMass to a net

☞ To efficiently model a clump weight attached to a net an $ExtMass can be used.

✪ Remove all constraints on the net.

✪ Set edge 0 to be static

✪ Add an ExtMass *clump* weight to the library with the following properties:

clump ExtMass feature definition

```
// Fluid loading
$CD 1
$CA 0.5

// Mechanical
$Diameter 1
$Density 2000
```

✪ Add the following $ExtMass lines to the net panel input file:

ExtMass properties in the net input file

```

$ExtMass clump 0 10
$ExtMass clump 20 10
```

✪ Run a simulation for 10 seconds and visualize the effect of the clump weights on the net.

Tutorial 32 - Cylindrical net cage

1 Tutorial overview

This tutorial covers:

- Creating a cylindrical net
- Creating a disk net
- Attaching cables to nets edges

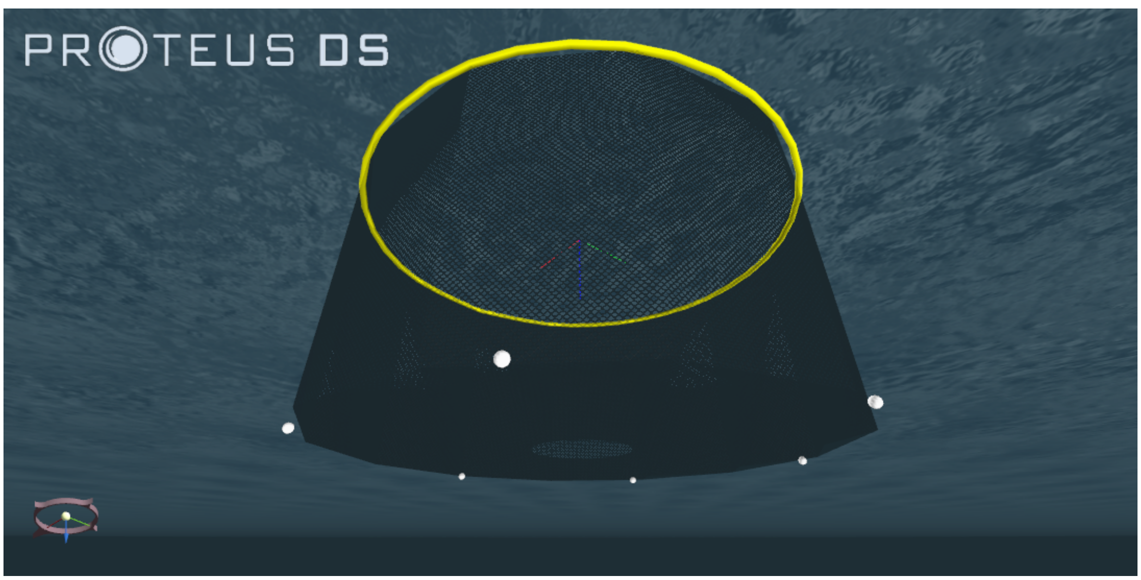


Figure 37: Cylindrical net cage with disk net bottom

2 Cylinder nets

✪ Create a new project.

✪ Create a new net DObject.

✪ Define the initial condition for the net using the the *Cylindrical* option. This initial condition dialog allows specification of the center point, radius and length of a cylinder. Set the radius of the cylinder to be 15m and the length to be 15m, and accept the default center point. Set the number of elements along the circumference to be 12 and the number of elements along the length to be 6.

- Set $Edge2Ring to be 1 and $Edge0Static to be 1 for the newly created net. The $Edge2Ring connects edge 2 and 3 together. Edge 0 is at the surface.
- Run a short simulation to visualize the cylindrical net.

3 Create a disk net

- To create a net bottom create a net DObject and define the initial condition using the *Disk* option.
- Center the disk at the center of the cylindrical net, and set the depth to 15m. The disk net has a hole at the center of it. It is recommended that the hole be a reasonable size to ensure that the element lengths used for the elements on the inner radius are not too small relative to the other elements. Small elements may result in very slow simulations.
- Choose an inner radius of 2.5m and and an outer radius of 15m. Select 12 elements around the circumference and 5 elements along the length (radial direction).
- The edge locations created with the disk net IC generator are shown in Figure 38. To prevent edge 2 and edge 3 from becoming disconnected, set $Edge2Ring to 1.
- ☞ Nets can be connected to each other along edges. A master net and follower net is specified in a net connection. The master net edge sets the position and velocity of the nodes along specified edges of a follower net.
- Connect net edge 1 of the cylinder net (master) to net edge 0 of the bottom net (follower) using the connection properties:

Connection between bottom net and cylinder net

```
// Mechanical
$DNetFollowerEdge 1 0
$DNetFollowerArcSpan 0 94.247778 //outer circumference of net = 94.2m
//specify the circumference of the net to a minimum of 6 decimal places
```

- ☞ When specifying the the circumference of the net using the $DNetFollowerArcSpan property, specify the arc span to a minimum of 6 decimal places. If this is not done, with a polar cable and cylindrical net, the finite-element cable and net models will become unstable at the beginning of a simulation. Attachment of bridle lines to the floating collar (polar cable) may hide the problem.

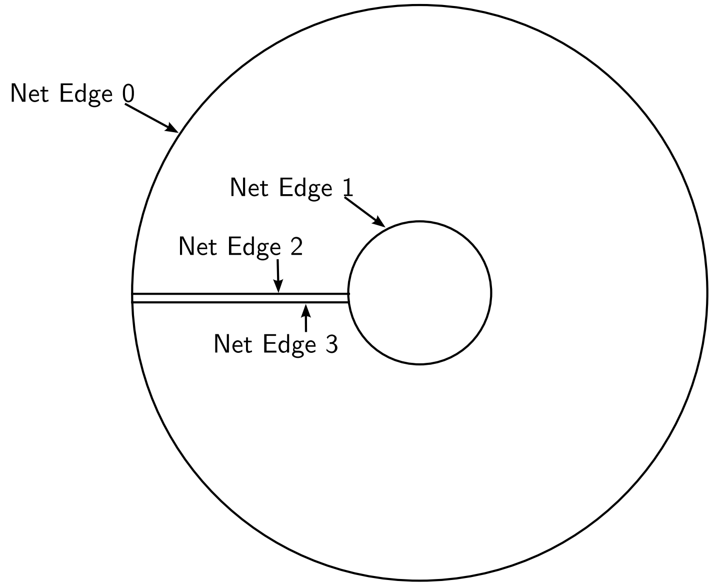


Figure 38: DiskNet1

- Run a simulation and with a steady current and watch the net cage deflect.
- To prevent the net from deflecting a clump weight can be added to the cylindrical net. Add an ExtMass feature called *clump* to the library, then add the following text to the net input file:

Net $ExtMass

```
$ExtMass clump 0 15
$ExtMass clump 15 15
$ExtMass clump 30 15
$ExtMass clump 45 15
$ExtMass clump 60 15
$ExtMass clump 75 15
```

- Run a simulation and observe the cylindrical net pen.

4 Creating a circular floating collar

- Create a cable DObject
- Define an initial condition for the cable, and select the Polar option.
- Set the circle radius to be 15m, and the number of elements to be 12.
- Set the cable (floating collar) properties in the appropriate library feature:

Floating collar properties

```
// Fluid loading
```

```
$CDc 1
$CDt 0.1
$CAc 1

// Mechanical
$EA 4.2e7
$EI1 4.24e6
$EI2 4.24e6
$GJ 0
$Diameter 0.445
$Density 317
$CID 1e5
$BCID 1e5
$TCID 1e5
$CE 1
```

- To accurately model buoyancy forces on the floating collar, add the advanced buoyancy properties in the cable input file:

Advanced cable buoyancy properties

```
$VBuoy 1
$BuoyancyAngularSubsegments 12
$BuoyancyAxialSubsegments 30
```

- To create a connection between the floating collar and the net, create a DCableDNetEdgeConnection.
- Set the connected arclength using the $DNetFollowerSpan property; the property specifies the start arclength and end arclength on the cable.

$DCableDNetEdgeConnection properties

```
// Mechanical
$DNetFollowerEdge 0
$DNetFollowerSpan 0 94.247778
```

- Run a simulation of the net cage in waves to observe how the net responds to waves.

Tutorial 33 - Net cage

1 Tutorial overview

This tutorial covers:

- Creating a net cage from net panels
- Connecting a net panel to a rigid body
- Connecting a net bottom to multiple nets

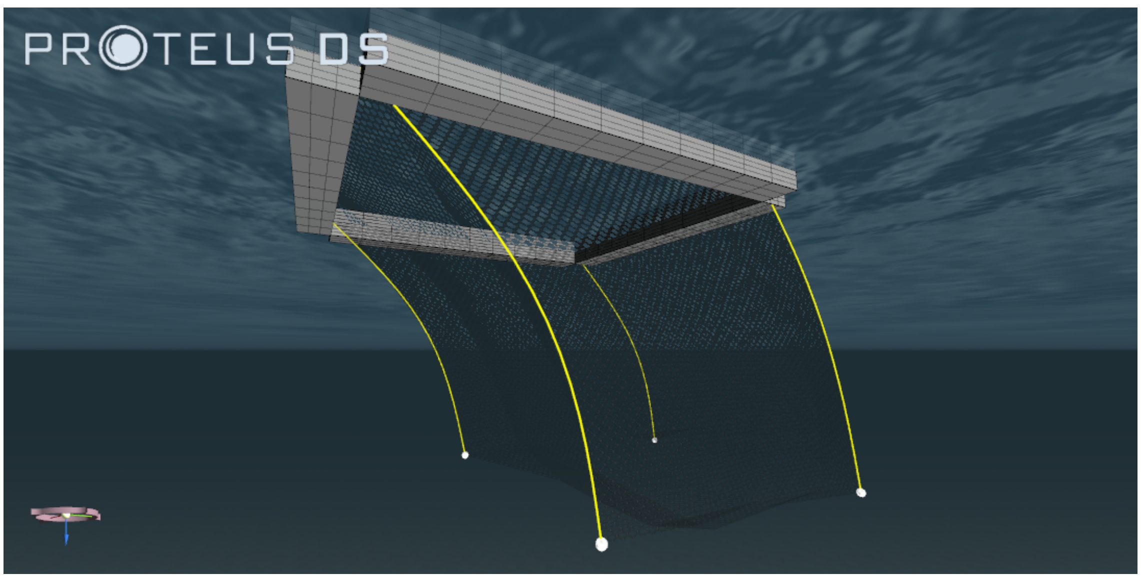


Figure 39: Net cage with rib lines

2 Creating rectangular net panels

- ✪ Create a new project.
- ✪ The net panel initial condition generator takes four planar coordinates which define the corners of a net panel. It can be used to create a cuboid shaped net cage.
- ✪ Create five (5) 10m by 10m net DObjects as shown in Figures 40-46 using the panel initial condition generator. Ensure that your node coordinate definitions match those given in the figures.

☞ A good schematic of how you have setup your net cages with net edges and net corner nodes labeled is useful to minimize connection difficulties.

✪ After all of the net panels have been created, run a short simulation and visualize the simulation results in PostPDS to ensure that all of the panel coordinates have been entered correctly.

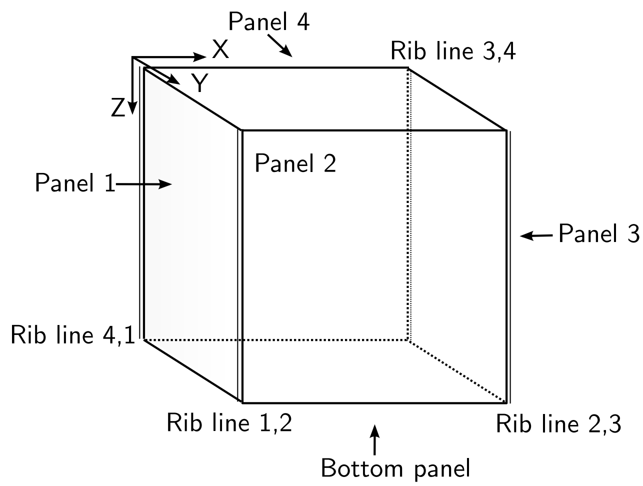


Figure 40: Tutorial net cage overview

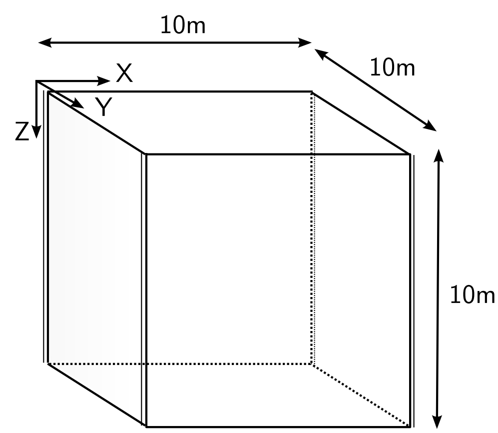


Figure 41: Net cage dimensions

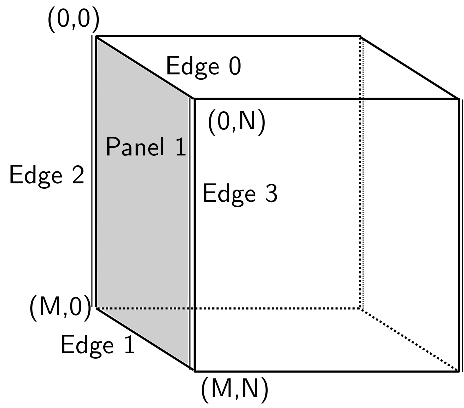


Figure 42: Net cage panel 1

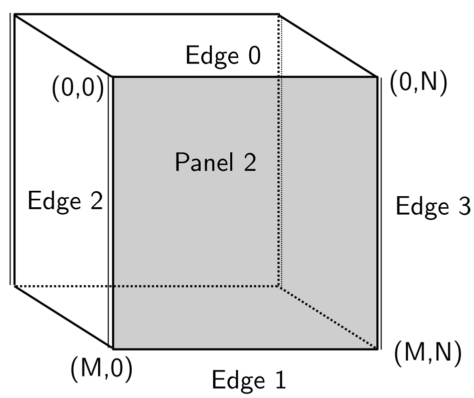


Figure 43: Net cage panel 2

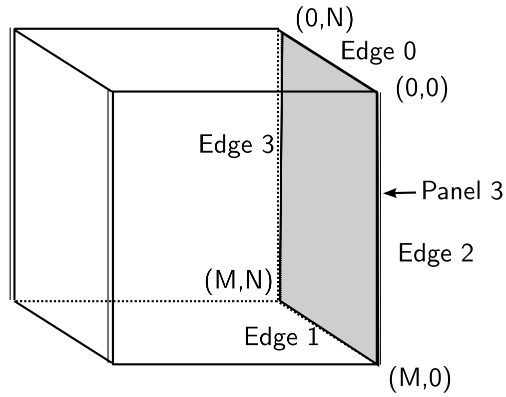


Figure 44: Net cage panel 3

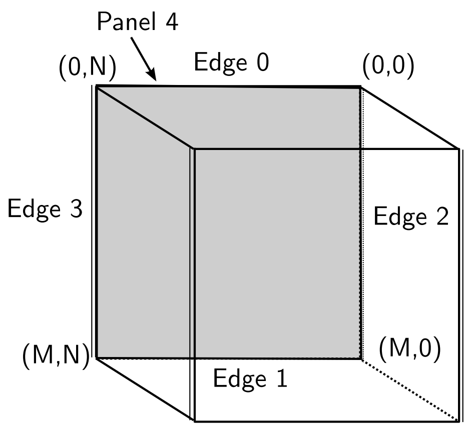


Figure 45: Net cage panel 4

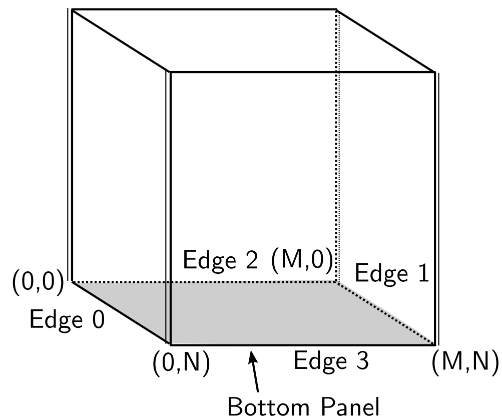


Figure 46: Net cage bottom panel

3 Creating a connection between the bottom net panel and the net cage walls

☞ To avoid circular dependencies, the bottom net panel should be the master connection with all net cage walls.

✪ Add a connection of type DNetDNetEdgeConnection between the bottom net panel and each of the four net cage wall panels (four connections in total). The bottom net panel will be the master object.

✪ The master net DObject controls all or part of a follower net DObject. To specify which arclength of the edge to control, the $DNetFollowerArcSpan property is used. The connection between the bottom panel and panel 1 is giving as:

Connection between bottom net panel and net panel 1

```
// Mechanical
$DNetFollowerEdge 0 1
$DNetFollowerArcSpan 0 10
```

☞ When making the connection between panel 4 and the bottom net panel, note that corner (0,0) on the bottom net is connected to corner (N,N) so the $DNetFollowerArcSpan must span from 10m to 0m along edge 1 of panel 4; note the change in arc span configuration given below. The same pattern applies for the connection between the bottom net and net panel 3.

Connection between bottom net panel and net panel 4

```
// Mechanical
$DNetFollowerEdge 2 1
$DNetFollowerArcSpan 10 0 //note the start and end arclength order
```

4 Adding rib lines

☞ Lines are often sewn into nets to provide structural support. These lines can be created using four cable DObjects.

✪ Create four cable DObjects as shown in Figure 47.

✪ To make a connection between a rib line and edge 2 of panel 1, create an edge connection. Specify the connection properties as:

Connection between edge 2 of panel 1 and the rib line

```
// Mechanical
$DNetFollowerEdge 2
$DNetFollowerSpan 0 10
```

✪ To make a connection between a rib line and edge 3 of panel 4, create an edge connection. Specify the connection properties as:

Connection between edge 3 of panel 4 and the rib line

```
// Mechanical
$DNetFollowerEdge 3
$DNetFollowerSpan 0 10
```

✪ Repeat the above connection method for all four rib lines.

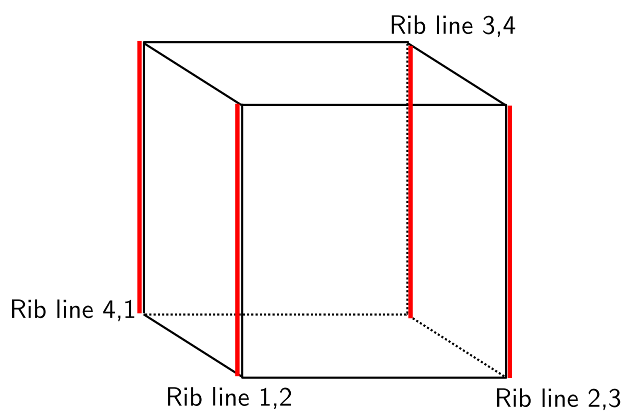


Figure 47: Rib line schematic

✪ Set edge 0 of each net to be static.

✪ Set the current profile type to 1 and add the $CurrentSpeed and $CurrentHeading properties. Set the current speed to some desired values (e.g. 0.5 m/s) and watch the net deflect.

✪ Run a simulation for 10 seconds and observe that the rib lines become disconnected from the bottom net panel.

5 Creating rib line point connections with nets

☞ In the previous section it was observed that the rib lines become disconnected from the bottom net. The end nodes of cables should be explicitly connected to nets or rigid bodies when a rib line edge connection is made, and there exists multiple nets connected to a single corner. The extra connection ensures that the rib line is fully constrained.

☞ The order that the net connections are made in ProteusDS can result in disconnections between rib lines, nets, and rigid bodies. Should the strategies presented in the tutorials not work as expected, please contact DSA for support.

✪ Between each of the rib lines, create four point connections. The point connection for rib line between panels 1 and 2 would be:

Point connections between cables and nets

```
$DNetFollowerEdge 0
$DNetFollowerArcLocation 10 //specifies the arclength location of the cable to connect
$DNetFollowerEdgeLocation 10 //specifies the arclength location on the net edge
```

✪ Once the connections between bottom net and the rib lines have been made, re-run the simulation. Notice that the top of the rib lines are still disconnected.

✪ Constrain node 0 of each rib line and re-run the simulation.

6 Add a floating collar

✪ Remove the static edge and the node 0 static constraints for all four net cage walls and the rib lines, respectively.

✪ Add a rigid body DObject to the simulation. Center the rigid body at the center of the net cage (5m,5m,10m).

✪ Add a RigidBodyCuboid feature to the feature Library. Set the dimensions to be 1m x 10m x 1m:

Floater cuboid properties

```
// Fluid loading
$WindLoading 1
$HydroLoading 1
$BuoyancyFroudeKrylov 1

// Uncategorized properties
$LengthX 10
$LengthY 1
$LengthZ 1
$CAx 1
$CAy 1
$CAz 1
$CDx 1
$CDy 1
$CDz 1
$SegmentsX 10
$SegmentsY 3
$SegmentsZ 8
```

✪ Arrange the cuboid features around the outside of the net perimeter. Set the draft of the cuboids to be 0.5m and set the mass properties as per the listing below:

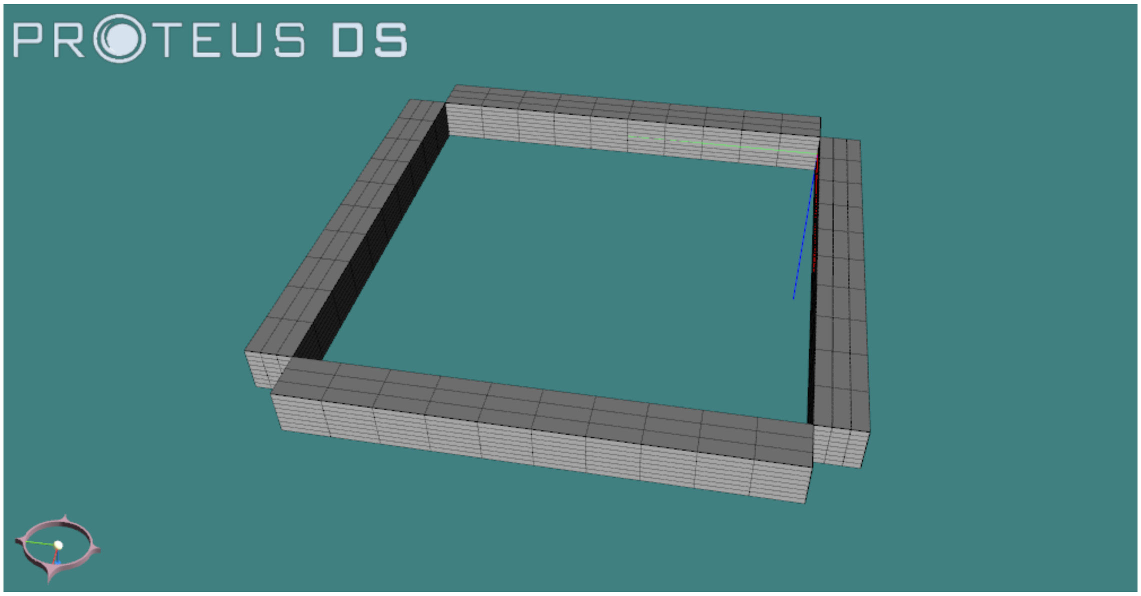


Figure 48: Net cage floater

Floater input file properties

```
// Mechanical
$Ix 172541
$Iy 172541
$Iz 341666
$Ixy 0
$Ixz 0
$Iyz 0
$Mass 20500 //kg

//panel 1 float
$Cuboid floatcollarsection -5.5 0 0 0 0 90

//panel 2 float
$Cuboid floatcollarsection 0 5.5 0 0 0 0

//panel 3 float
$Cuboid floatcollarsection 5.5 0 0 0 0 90

//panel 4 float
$Cuboid floatcollarsection 0 -5.5 0 0 0 0
```

- Make a connection between the rigid body and the net. The rigid body is the master DObject and the net cage walls are the follower DObjects. The edge of a net panel is constrained along a straight line distance between two points on the rigid body. To constrain net panel 1 to the floater, use the following connection properties:

RigidBodyDNetEdgeConnection between the floater and net panel 1

```
// Mechanical
$DNetFollowerEdge 0

\\Net panel edge start and edge location on the rigid body:
$DNetFollowerSpan -5 -5 0 -5 5 0
```

```
\\Coordinates are in terms of the rigid body local frame
```

- ✪ Make connections between the rigid body floating collar and all four net panels (along each edge 0, adjusting the $DNetFollowerSpan for each connection).
- ✪ Add a clump weight at the end of each of the rib lines using the ExtMass feature. Make the clump weight 0.25m in diameter with a density of 2000kg/m^2.
- ✪ Run the simulation again.
- ✪ Option 1: Try simulating the net cage in waves,
- ✪ Option 2: Try setting the $Kinematic flag to 1 for the floater to prevent movement of the cage in current.
- ✪ Option 3: Try adding mooring lines to the floater and simulating the response of the floater to waves and current.
- ☞ A hydrodynamic database may be required to properly estimate wave loads (radiation and diffraction effects) on the floater since floaters are often large relative to the incident wave length.

Tutorial 34 - Clamped cable connections

1 Tutorial overview

This tutorial covers:

- Creating a clamped cable connection with a rigid body

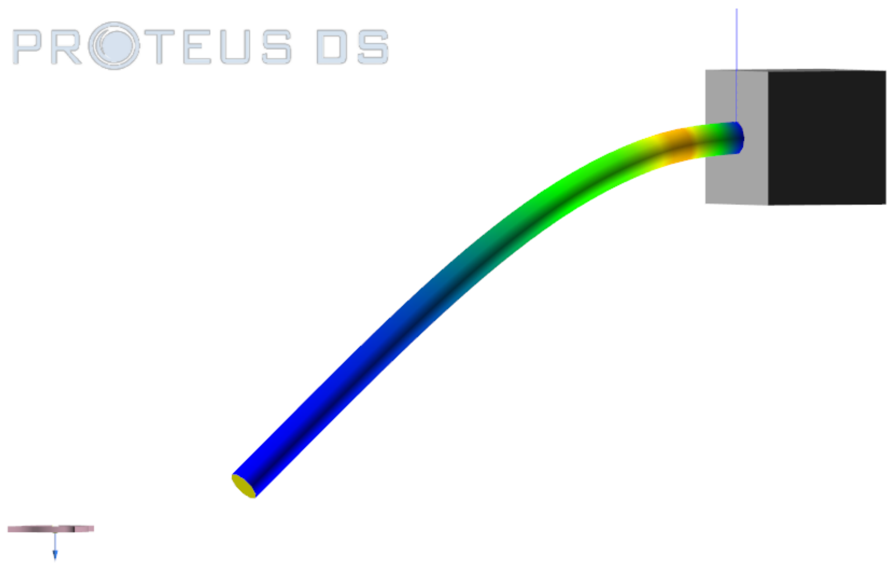


Figure 49: Clamped cable

2 Simulation setup

✪ Create a Cable DObject.

✪ Set the Cable Node 0 position to be (0,0,5) and the Cable Node N position to be (10,0,5), and the length to 10m.

✪ Adjust the default DCableSegment feature properties for the newly created cable: increase $Density to 2000kg/m^3, increase $Diameter to 0.25m and increase the bending stiffness ($EI1 and $EI2) to 10000.

☞ This will make the cable heavier and more stiff, such that it will deflect under its own selfweight. The larger cable diameter will assist with visualization.

✪ Create a RigidBody DObject.

- ✪ Set the RigidBody position to be (0,0,5)
- ✪ To fix the RigidBody in space set the $Kinematic property to 1 in the RigidBody input file
- ✪ Add a RigidBodyCuboid feature named *myCube* to the Library, use the default properties. Associate the cuboid feature with the RigidBody created using the $Cuboid property:

$Cuboid feature in RigidBody input file

```
// Mechanical
$Ix 1
$Iy 1
$Iz 1
$Ixy 0
$Ixz 0
$Iyz 0
$Mass 1

$Kinematic 1

$Cuboid myCube -0.5 0 0 0 0 0
```

3 Create a pinned and clamped connection

- ✪ Add a Point connection between RigidBody and the Cable, accepting the default connection properties
- ✪ Run a 10 second simulation and view the results. Notice that the cable swings down below the cube. Due to the point connection, the cable will act like a pendulum.
- ☞ To model a rigid connection between the stiff cable that has been created and the rigid body, a clamped connection must be used. Clamped connections are useful for simulating bend stiffeners and pipe flanges.
- ☞ The tangent direction, p_3, of a cable at node 0 and node N of a cable is defined by the p_1 and p_2 unit vectors as shown in Figure 50. To make a clamped cable connection with a rigid body, p_1 and p_2 are defined in terms of the rigid body reference frame within the cable connection. By defining these directions, the cable is constrained in torsion and bending.
- ✪ To clamp the connection between the rigid body and cable, add the $DCableFollowerClamped property to the RigidBodyDCablePointConnection that was created earlier. Additionally, one must define the p_1 and p_2 vectors:

Clamped connection properties in a RigidBody-DCable point connection

```
// Mechanical
$DCableFollowerNodeN 0
$DCableFollowerLocation 0 0 0

$DCableFollowerClamped 1
$DCableFollowerP1 0 1 0
$DCableFollowerP2 0 0 1
```

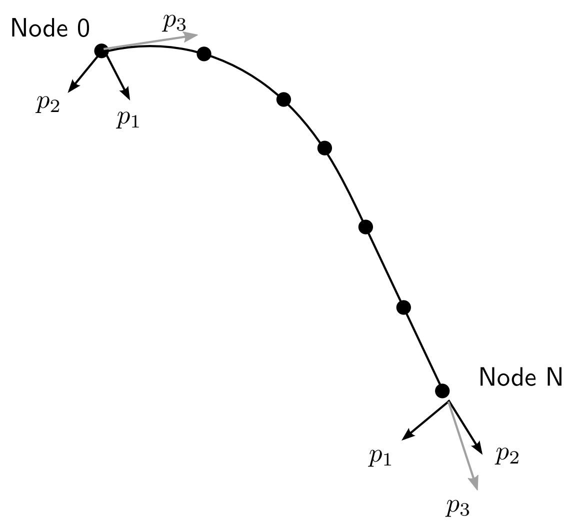


Figure 50: Relationship between p_1 and p_2 vectors and cable tangent definition at node 0 and node N

- ✪ Run the simulation again for 10 seconds and view the results.
- ✪ When loading the simulation results into PostPDS, in the 'Specify Load Options' dialog, increase the number of radial elements to display to 12 to better visualize the results; compare how the cable looks in PostPDS when this value is set to 6.
- ✪ To examine the bending radius, select 'Stress (Flexural)' as the plotted variable (as shown in Figure 49).
- ☞ For clamped connections between Node N and a RigidBody, note that the tangent vector p_3 comes out of the end of the cable, whereas at Node 0, p_3 goes into the cable.

Tutorial 35 - Fiber rope cable properties

1 Tutorial overview

This tutorial covers:

- Determining DCableSegment feature properties for fiber rope

2 Synthetic fiber rope

☞ Synthetic ropes, made from polyester, Dyneema, nylon or copolymer materials are frequently used for a variety of mooring and marine applications. For example, a high strength Dyneema rope used in mooring applications is Amsteel-Blue; the manufacturer provides properties for this rope online[1] that can be used to set the cable properties in the DCableSegment feature in ProteusDS.

☞ The following tutorial reviews how to set DCableSegment feature properties for a 3/4 inch Amsteel-Blue rope, the properties of which are given in Figure 51.

3 Size and buoyancy using specific gravity

☞ Many fiber rope do not have a uniform circular cross section, and/or they may effectively contain small air gaps that entrain water when used in subsea applications. The DCableSegment feature properties must be set in ProteusDS to ensure that the mass and buoyancy properties are set to correctly model the effective wet weight and mass in air per unit length of the rope.

☞ The $Diameter property is always used to specify the nominal diameter for ropes and cables; this diameter is used for drag and added mass calculations.

✪ For a 3/4" nominal diameter rope, the $Diameter is 0.01905m.

☞ The mass per unit length in air can be found in the rope specifications. The mass per unit length for 3/4" Amsteel-Blue rope is given as 19.8 kg/100m.

☞ The $Density property is used with the $Diameter to set the mass of the rope in ProteusDS; based on the mass per unit length the $Density can be determined using:

$$\rho = \frac{4M_l}{\pi d^2}$$

where M_l is the mass per unit length in air in kg/m and d is the nominal diameter in m.

[1] http://www.samsonrope.com/Pages/Product.aspx?ProductID=872

✪ Using this equation, the $Density is calculated to be 694.7 kg/m^3.

☞ The $BuoyancyDiameter property is used to set the distributed buoyancy force per unit length of the cable. In the case of a fiber rope which entrains water, the $BuoyancyDiameter value would be smaller than the nominal diameter. In the case of a solid pipe, the $BuoyancyDiameter would be equal to the $Diameter.

☞ If the $BuoyancyDiameter property is not specified, ProteusDS will use the nominal diameter (the $Diameter property) to calculate buoyancy.

☞ To calculate the $BuoyancyDiameter for the 3/4 inch Amsteel-Blue rope requires that either the rope mass per unit length in water, or the specific gravity of the rope material is provided. For the Dyneema fiber used in Amsteel-Blue the specific gravity is 0.98.

☞ To calculate buoyancy diameter use the equation from the ProteusDS Manual:

$$d_b = \sqrt{\frac{4M_l}{\pi \cdot 1000 \cdot SG_{cm}}}$$

✪ Using this equation, the $BuoyancyDiameter is calculated to be 0.016m.

4 Size and buoyancy using apparent mass per unit length in water

☞ It is common for manufacturers to provide the mass per unit length in air and the weight in water of a cable. For example, the 113 mm DeepRope Polyester rope data sheet can be online at: `http://www.vryhof.com/pdf/deepropemanual.pdf` on page 36, shown in Figure 52. When provided, these properties should be used to calculate the $BuoyancyDiameter of a cable.

✪ For the 113 mm DeepRope, the $Diameter property sets the nominal diameter. For a 113mm rope it is 0.113m.

✪ Using the provided mass in air (8.80kg/m) and the nominal diameter, the $Density can be calculated using the same equation from the ProteusDS Manual as used above. The $Density is calculated to be 877 kg/m^3.

☞ To calculate buoyancy diameter with known mass per unit length in air and mass per unit length in water, an equation from the ProteusDS Manual is used:

$$d_b = \sqrt{\frac{4(M_l - M_w)}{\pi \rho_w}}$$

where M_l is the mass per unit length in air in kg/m, M_w is the apparent mass per unit length in water in kg/m, d is the nominal diameter in m, and ρ_w is the density of water in kg/m^3.

✪ Using this equation, with a mass per unit length in water of 2.11 kg/m and a water density of 1025 kg/m^3 (for saltwater), the $BuoyancyDiameter is calculated to be 0.0912m.

☞ Note: Often various weights in water are provided by the manufacturer for different rope tensions. Use the value that is closest to the expected tension of the rope in the simulation. If the tension is unknown, use one value and readjust the cable properties after running some preliminary simulations.

5 Mechanical properties

☞ The axial stiffness of fiber ropes can be calculated using the elongation properties provided by the manufacturer.

☞ For the 3/4" Amsteel-Blue rope, the percent elongation is given as 0.96% at 30% of the maximum breaking load.

☞ The axial stiffness can be calculated using an equation from the ProteusDS Manual:

$$EA = \frac{100F_\zeta}{\zeta}$$

where ζ is the percent elongation at an applied load of F_ζ (in N).

✪ Taking 30% the average strength provided of 64 400 lbs, or 113.9e3 N gives a F_ζ of 34.17e3 N. Using the equation yields an axial stiffness ($EA) of 3.56e6 N.

☞ The flexural stiffness ($EI) and torsional stiffness ($GJ) are negligible in fiber rope and can be assumed to be zero for many applications.

✪ Fiber rope has low effective compressive elasticity, therefore the compressive elasticity property ($CE) may be assumed to be 0 or very low.

6 Drag and added mass properties

✪ The added mass coefficient ($CAc) for most ropes and cables can be 1.0, represented as long cylinders.

✪ The drag coefficient ($CDc) for fiber rope can be approximated as 1.5.

✪ The tangential drag coefficient ($CDt) can be left as the default value of 0.01 for all ropes.

☞ The rope properties for a 3/4 inch Amsteel-Blue rope may be set as:

Amsteel-Blue fiber rope properties

```
// 3/4" Amsteel-Blue rope properties
// Fluid loading
$CDc 1.5
$CDt 0.01
$CAc 1.0

// Mechanical
$EA 3.56E6
$EI1 0
$EI2 0
$GJ 0
$Diameter 0.01905
$BuoyancyDiameter 0.016
$Density 694.7
$CID 100
$BCID 0
$TCID 0
$CE 0
```

AMSTEEL®-BLUE

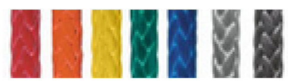
PRODUCT CODE: 872

FIBER: DYNEEMA®

SPECIFIC GRAVITY: 0.98

SPLICE: 12-STRAND CLASS II

COLOR OPTIONS:



AmSteel®-Blue is a torque-free 12-strand single braid that yields the maximum in strength-to-weight ratio and, size-for-size, is the same strength as steel—but it's so light, it floats. AmSteel®-Blue is an excellent wire rope replacement with extremely low stretch, and superior flex fatigue and wear resistance.

ELASTIC ELONGATION:

10%	20%	30%
0.46%	0.70%	0.96%

SPECIFICATIONS:

DIAM. (inch)	CIRC. (inch)	WEIGHT PER 100 FT. (lbs)	AVG. STRENGTH (lbs)	MIN. STRENGTH (lbs)	DIAM. (mm)	CIRC. (mm)	WEIGHT PER 100 M (kg)	AVG. STRENGTH (kg)	MIN. STRENGTH (kg)	ISO 2307 STRENGTH (metric tons)
7/64	5/16	0.3	1,600	1,400	2.5	7.5	0.45	730	650	0.73
1/8	3/8	0.5	2,500	2,300	3	9	0.74	1,100	1,000	1.1
5/32	15/32	0.75	4,000	3,600	4	12	1.1	1,800	1,600	1.8
3/16	9/16	1	5,400	4,900	5	15	1.5	2,400	2,200	2.4
1/4	3/4	1.6	8,600	7,700	6	18	2.4	3,900	3,500	3.9
5/16	1	2.7	13,700	12,300	8	24	4	6,200	5,600	6.2
3/8	1 1/8	3.6	19,600	17,600	9	27	5.4	8,900	8,000	8.9
7/16	1 1/4	4.2	23,900	21,500	11	33	6.2	10,800	9,800	10.8
1/2	1 1/2	6.4	34,000	30,600	12	36	9.5	15,400	13,900	15.4
9/16	1 3/4	7.9	40,500	36,500	14	42	11.8	18,400	16,500	18.4
5/8	2	10.2	52,800	47,500	16	48	15.2	24,000	21,600	24
3/4	2 1/4	13.3	64,400	58,000	18	54	19.8	29,200	26,300	29.2

Figure 51: Amsteel-Blue rope properties

DeepRope® Polyester strength table

Material: Acordis Polyester 855TN
Minimum Breaking Load in spliced condition
Total weight is in air:
Submerged weight is in seawater (_ = 1,05 kg/l)
conform ISO (@1- 2% MBL)
conform PetroBras spec. (@20% MBL)

Diam	MBL		Total weight [kg/m]		Submerged weight		Stiffness [kN]		
mm	tf	kN	@2% MBL	@20%MBL	@2%MBL	@20%MBL	EA[1]	EA[2]	EA[3]
113	380	3723	8,80	8,24	2,11	3723	7,19E+04	8,43E+04	1,10E+05
117	414	4061	9,49	8,88	2,27	4061	7,84E+04	9,20E+04	1,20E+05
126	467	4738	10,9	10,2	2,60	4738	9,15E+04	1,07E+05	1,40E+05
130	518	5077	11,5	10,8	2,76	5077	9,80E+04	1,15E+05	1,50E+05
133	552	5415	12,2	11,4	2,92	5415	1,05E+05	1,23E+05	1,60E+05
137	587	5754	12,9	12,0	3,08	5754	1,11E+05	1,30E+05	1,70E+05
141	621	6092	13,6	12,7	3,24	6092	1,18E+05	1,38E+05	1,80E+05
144	656	6430	14,2	13,3	3,40	6430	1,24E+05	1,46E+05	1,90E+05
147	690	6769	14,9	13,9	3,56	6769	1,31E+05	1,53E+05	2,00E+05
151	725	7107	15,6	14,5	3,72	7107	1,37E+05	1,61E+05	2,10E+05
154	759	7446	16,2	15,1	3,88	7446	1,44E+05	1,69E+05	2,20E+05
157	794	7784	16,9	15,8	4,04	7784	1,50E+05	1,76E+05	2,30E+05
160	828	8123	17,5	16,4	4,19	8123	1,57E+05	1,84E+05	2,40E+05
163	863	8461	18,2	17,0	4,35	8461	1,63E+05	1,92E+05	2,49E+05
166	897	8800	18,9	17,6	4,51	8800	1,70E+05	1,99E+05	2,59E+05
169	932	9138	19,5	18,2	4,67	9138	1,76E+05	2,07E+05	2,69E+05
172	966	9476	20,2	18,8	4,82	9476	1,83E+05	2,15E+05	2,79E+05
175	1001	9815	20,8	19,4	4,98	9815	1,89E+05	2,22E+05	2,89E+05
177	1035	10153	21,5	20,0	5,13	10153	1,96E+05	2,30E+05	2,99E+05
180	1070	10492	22,1	20,6	5,29	10492	2,03E+05	2,38E+05	3,09E+05
183	1104	10830	22,8	21,2	5,45	10830	2,09E+05	2,45E+05	3,19E+05
185	1139	11169	23,4	21,8	5,60	11169	2,16E+05	2,53E+05	3,29E+05
188	1173	11507	24,1	22,4	5,76	11507	2,22E+05	2,61E+05	3,39E+05
190	1208	11846	24,7	23,0	5,91	11846	2,29E+05	2,68E+05	3,49E+05
193	1242	12184	25,4	23,6	6,07	12184	2,35E+05	2,76E+05	3,59E+05
195	1277	12522	26,0	24,2	6,22	12522	2,42E+05	2,84E+05	3,69E+05
198	1311	12861	26,7	24,8	6,38	12861	2,48E+05	2,91E+05	3,79E+05
200	1346	13199	27,3	25,4	6,53	13199	2,55E+05	2,99E+05	3,89E+05
203	1380	13538	28,0	26,0	6,69	13538	2,61E+05	3,07E+05	3,99E+05
205	1415	13876	28,6	26,6	6,84	13876	2,68E+05	3,14E+05	4,09E+05
207	1449	14215	29,2	27,2	6,99	14215	2,74E+05	3,22E+05	4,19E+05
210	1484	14553	29,9	27,8	7,15	14553	2,81E+05	3,29E+05	4,29E+05
212	1518	14892	30,5	28,4	7,30	14892	2,87E+05	3,37E+05	4,39E+05
214	1553	15230	31,2	29,0	7,45	15230	2,94E+05	3,45E+05	4,49E+05
221	1656	16245	33,1	30,8	7,91	16245	3,14E+05	3,68E+05	4,79E+05
227	1760	17261	35,0	32,6	8,37	17261	3,33E+05	3,91E+05	5,09E+05
233	1863	18276	36,9	34,4	8,83	18276	3,53E+05	4,14E+05	5,39E+05
239	1967	1929	38,8	36,1	9,29	1929	3,72E+05	4,37E+05	5,69E+05
245	2070	20307	40,7	37,9	9,74	20307	3,92E+05	4,60E+05	5,99E+05

Dynamic Modulus based on type approval tests for BV and PetroBras: 1 cycling between 10-30% MBL, 2 cycling between 20-30% MBL, 3 cycling between 40-50% MBL

Figure 52: DeepRope rope properties

Tutorial 36 - Wire rope cable properties

1 Tutorial overview

This tutorial covers:

- Determining DCableSegment feature properties for wire rope

2 Wire rope

☞ Wire ropes, made from various types of steel are frequently used for a variety of mooring and marine applications. Several wire rope manufacturers provide property for their ropes online[1] that can be used to set the cable properties in the DCableSegment feature in ProteusDS.

☞ The following tutorial reviews how to set DCableSegment feature properties for a 1 inch 6x19 classification steel wire-rope, the properties of which are given in Figure 53.

3 Size and buoyancy

☞ Wire ropes do not have a uniform cross section, and they may contain small air gaps that entrain water when used in subsea applications. The DCableSegment feature properties must be set in ProteusDS to correctly model this.

☞ The $Diameter property is always used to specify the nominal diameter for ropes and cables; this diameter is used for drag and added mass calculations.

✪ For a 1 " nominal diameter wire rope the $Diameter is 0.0254m.

☞ The mass per unit length in air is provided in the rope specifications. The mass per unit length for fiber core 1" wire rope is 1.68lbs/ft, or 2.5kg/m.

✪ The $Density can be calculated from the nominal diameter and the mass in air using the equation:

$$\rho = \frac{4M_l}{\pi d^2}$$

where M_l is the mass per unit length in air in kg/m and d is the nominal diameter in m.

✪ The $Density is calculated to be 4934 kg/m^3.

☞ The $BuoyancyDiameter property is used to set the distributed buoyancy force per unit length of the cable.

☞ To calculate buoyancy diameter for a 1" wire rope with no mass per unit length in water or specific gravity provided, the specific gravity must be calculated using the material density.

[1] http://www.hanessupply.com/content/pdfs/wireRope101.pdf

☞ Assuming a material density of 7800 kg/m^3 for steel, the specific gravity can be calculated using the equation:

$$SG_{cm} = \rho_{cm}/1000$$

where ρ_{cm} is the material density.

✪ Using this equation, the specific gravity of steel wire rope is 7.8.

✪ To calculate the buoyancy diameter use the equation from the ProteusDS Manual:

$$d_b = \sqrt{\frac{4M_l}{\pi \cdot 1000 \cdot SG_{cm}}}$$

✪ Using this equation, the $BuoyancyDiameter is calculated to be 0.0202m.

4 Mechanical properties

☞ The axial stiffness of wire ropes can be calculated using elongation properties if provided by the manufacturer, similar to fiber ropes as performed in the fiber rope properties tutorial. In cases where elongation properties are not provided, the axial stiffness must be calculated using an assumed elastic modulus based on literature.

☞ For this example, no elongation properties are provided by the manufacturer and therefore an elastic modulus must be assumed.

✪ For stranded wire rope, an elastic modulus of 7.0e10 N/m^2 can be assumed (Based on properties provided in the ProteusDS Manual).

☞ The area of the wire rope can be calculated using the equation:

$$A = \pi d^2/4$$

✪ Calculating area using the nominal diameter yields an area of 5.07e-4 m^2.

✪ The axial stiffness is the product of the elastic modulus and the cross sectional area. An axial stiffness ($EA) of 3.55e7 N is calculated.

☞ Bending and torsional stiffness ($EI and $GJ) are calculated for wire rope using the area moment of inertia (I) and the polar moment of inertia (J).

☞ The area moment of inertia (I) for a wire rope can be calculated using an equation from the ProteusDS Manual:

$$I = \frac{\pi}{64}d^4$$

✪ For 1" wire rope, the area moment of inertia is calculated to be 2.04e-8 m^4.

✪ The bending stiffness is the product of the elastic modulus and the area moment of inertia. A bending stiffness ($EI) of 1430 N*m^2 is calculated.

☞ The polar moment of inertia (J) for a wire rope can be calculated using an equation from the ProteusDS Manual:

$$J = \frac{\pi}{32}d^4$$

✪ The polar moment of inertia for a 1" wire rope is calculated to be 4.08E-8 m^4.

☞ The shear modulus (G) of a material is not usually available, however, an assumption can be made to use a shear modulus that is 50% of the elastic modulus.

- ✪ Assuming a shear modulus that is 50% of the elastic modulus, the shear modulus is 3.5e10 N.
- ✪ The torsional stiffness is the product of the shear modulus and the polar moment of inertia. A torsional stiffness ($GJ) of 1430 N*m^2 is calculated.
- ✪ Wire rope has high compressive elasticity, therefore the compressive elasticity property ($CE) should be left as 1.

5 Drag and added mass properties

- ☞ The added mass coefficient ($CAc) for most ropes and cables can be 1.0, represented as long cylinders.
- ☞ The drag coefficient ($CDc) for spiral wire without sheathing is specified in the ProteusDS Manual as between 1.4 - 1.6. In this situation 1.5 will be chosen.
- ☞ The tangential drag coefficient ($CDt) can be left as the default value of 0.01 for all wire ropes.
- ☞ The rope properties for a 1 inch wire rope may be set as:

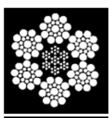
6 All properties

Wire rope properties

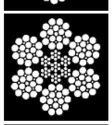
```
// 1" Wire rope properties
// Fluid loading
$CDc 1.5
$CDt 0.01
$CAc 1.0

// Mechanical
$EA 3.55E7
$EI1 1430
$EI2 1430
$GJ 1430
$Diameter 0.0254
$BuoyancyDiameter 0.0202
$Density 4934
$CID 100
$BCID 0
$TCID 0
$CE 1
```

6 x 19 Classification

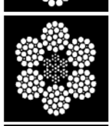
6 x 19 Classification ropes provide an excellent balance between fatigue and wear resistance. They give excellent service with sheaves and drums of moderate size. 6 x 19 Classification ropes contain 6 strands with 15 through 26 wires per strand, no more than 12 of which are outside wires.



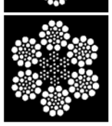
6 x 19 Seale *Characteristics:* Resistant to abrasion and crushing; medium fatigue resistance. *Typical Applications:* Haulage rope, choker rope, rotary drilling line. *IWRC shown: fiber core available*



6 x 21 Filler Wire *Characteristics:* Less abrasion resistance; more bending fatigue resistance. *Typical Applications:* Pull Ropes, load lines, backhaul ropes, draglines. *IWRC shown; fiber core available*



6 x 25 Filler Wire *Characteristics:* Most flexible rope in classification; best balance of abrasion and fatigue resistance. *Typical Applications:* Most widely used of all wire ropes - cranes hoists, skip hoists, haulage, mooring lines, conveyors, etc. *IWRC shown; fiber core available*



6 x 26 Warrington Seale *Characteristics:* Good balance of abrasion and fatigue resistance. *Typical Applications:* Boom hoists, logging and tubing lines. *IWRC shown, fiber core available*

Dia. (in)	Nominal Strength(Tons)* – Bright or Drawn Galvanized†						Approx. Wt./Ft. (lbs)	
	EEIP		EIP		IPS			
	IWRC	Fiber Core	IWRC	Fiber Core	IWRC	Fiber Core	IWRC	Fiber Core
1/4	–	–	3.40	3.02	2.94	2.74	0.116	0.105
5/16	–	–	5.27	4.69	4.58	4.26	0.18	0.164
3/8	–	–	7.55	6.71	6.56	6.10	0.26	0.236
7/16	11.2	9.90	10.2	9.09	8.89	8.27	0.35	0.32
1/2	14.6	12.9	13.3	11.8	11.5	10.7	0.46	0.42
9/16	18.5	16.2	16.8	14.9	14.5	13.5	0.59	0.53
5/8	22.7	20.0	20.6	18.3	17.9	16.7	0.72	0.66
3/4	32.4	28.6	29.4	26.2	25.6	23.8	1.04	0.95
7/8	43.8	38.6	39.8	35.4	34.6	32.2	1.42	1.29
1	57.5	50.0	51.7	46.0	44.9	41.8	1.85	1.68
1-1/8	71.5	63.0	65.0	57.9	56.5	52.6	2.34	2.13
1-1/4	87.9	77.5	79.9	71.0	69.4	64.6	2.89	2.63
1-3/8	106.0	93.0	96.0	85.4	83.5	77.7	3.50	3.18
1-1/2	125.0	111.0	114.0	101.0	98.9	92.0	4.16	3.78
1-5/8	145.0	129.0	132.0	118.0	115.0	107.0	4.88	4.44
1-3/4	168.0	149.0	153.0	136.0	133.0	124.0	5.67	5.15
1-7/8	191.0	169.0	174.0	155.0	152.0	141.0	6.50	5.91
2	218.0	192.0	198.0	176.0	172.0	160.0	7.39	6.72
2-1/8	–	–	221.0	197.0	192.0	179.0	8.35	7.59
2-1/4	–	–	247.0	220.0	215.0	200.0	9.36	8.51
2-3/8	–	–	274.0	244.0	239.0	222.0	10.4	9.48
2-1/2	–	–	302.0	269.0	262.0	244.0	11.6	10.5
2-5/8	–	–	331.0	–	288.0	268.0	12.8	11.6
2-3/4	–	–	361.0	–	314.0	292.0	14.0	12.7

Figure 53: Wire rope properties

Tutorial 37 - Chain properties

1 Tutorial overview

This tutorial covers:

- Determining DCableSegment feature properties for chain

2 Chain

☞ Chain is often used in subsea moorings and a variety of marine applications. Several chain manufacturers provide properties for their chain online [1] that can be used to set the cable properties in the DCable segment in ProteusDS.

☞ The following tutorial reviews how to set DCableSegment feature properties for a 30mm diameter R3 studless chain. The properties of which are given in Figure 55.

3 Diameter and density

☞ The characteristic diameter used in ProteusDS is different from the nominal diameter reported by the manufacturer. Nominal diameter referred to by the manufacturer is the diameter of the bar used to make the chain. This can be seen in Figure 54.

☞ The characteristic diameter ($Diameter) used in ProteusDS can be calculated using an equation from the ProteusDS Manual:

$$d = \sqrt{\frac{4M_l}{\rho_{cm}\pi}}$$

where M_l is the mass per unit length in air and ρ_{cm} is the material density.

☞ The mass per unit length in air is given as 18.18 kg/m and the material is steel, therefore a material density of 7800 kg/m^3 will be used.

✪ The characteristic diameter ($Diameter) is calculated to be 0.545m.

☞ The $Density property is the material density when used for chain. For this steel chain, the $Density property is set to 7800 kg/m^3.

☞ The $BuoyancyDiameter property should not be used for chain.

[1] `http://www.jeyco.com.au/media/files/studless/Grade%203%20Studless%20Chain%20Specs.pdf`

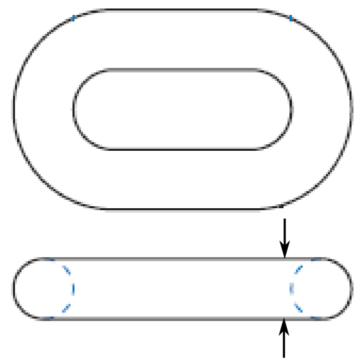


Figure 54: Nominal diameter of studless chain

4 Mechanical properties

☞ The axial stiffness of chain is the product of the elastic modulus and the cross sectional area of the chain.

☞ Elastic modulus, E, is required to calculate the axial stiffness. For R3 studless chain, the following equation from the ProteusDS Manual can be used:

$$E = (5.40 - 0.0040 * d_c) * 10^{10} N/m^2$$

where d_c is the nominal chain diameter in mm as specified by the manufacturer.

✪ Using the table, E for the R3 studless chain can be determined to be 5.28e10 N/m^2.

☞ The area of the chain can be calculated using an equation from the ProteusDS Manual:

$$A = 2\pi d_c^2/4$$

where d_c is the nominal chain diameter in m as specified by the manufacturer.

✪ The area of the chain is calculated to be 1.414e-3 m^2.

✪ The axial stiffness ($EA) property is the product of the elastic modulus and the area, therefore the $EA value is set to 7.464e7 N.

✪ The flexural stiffness ($EI) and torsional stiffness ($GJ) are negligible in chain and can be assumed to be zero.

✪ Chain has low effective compressive elasticity, therefore the compressive elasticity property ($CE) may be assumed to be 0 or very low.

5 Drag and added mass properties

☞ The added mass coefficient ($CAc) for most chains can be 1.0, represented as long cylinders.

☞ The drag coefficient for studless chain is specified in the ProteusDS Manual as between 2.0 - 2.4 (based on DNV recommendations). In this situation 2.2 will be chosen.

☞ The diameter of chain to be used in the drag and added mass calculations can be specified by the $FluidDiameter property.

✪ The $FluidDiameter property is set to 0.30.

✪ the $CAc property is set to 1.0, the $CDc property us set to 2.2.

☞ The tangential drag coefficient ($CDt) can be left as the default value of 0.01 for all chains.

☞ The DCableSegment feature properties for a 30mm R3 studless chain may be set as:

6 All properties

Chain properties

```
// 30mm Chain properties
// Fluid loading
$FluidDiameter 0.30
$CDc 2.2
$CDt 0.01
$CAc 1.0

// Mechanical
$EA 7.464E7
$EI1 0
$EI2 0
$GJ 0
$Diameter 0.545
$Density 7800
$CID 100
$BCID 0
$TCID 0
$CE 0
```

JEYCO

STUDLESS CHAIN GRADE 3

Available Galvanised or Black – Larger sizes available on request.

Code	Dia. (mm)	Proof Load (tonne)	Breaking Load (tonne)	Weigh t (kg/m)
CSLESS314B/G	14	7.75	15.80	3.96
CSLESS316B/G	16	9.99	20.39	5.18
CSLESS318B/G	18	14.27	29.05	6.55
CSLESS320B/G	20	16.41	32.62	8.08
CSLESS322B/G	22	19.98	40.88	9.78
CSLESS324B/G	24	23.69	48.52	11.65
CSLESS325B/G	25	25.33	52.60	12.63
CSLESS326B/G	26	27.76	56.68	13.66
CSLESS328B/G	28	32.04	65.44	15.85
CSLESS330B/G	30	36.68	74.92	18.18
CSLESS332B/G	32	41.60	84.91	20.69
CSLESS334B/G	34	46.74	95.51	23.36
CSLESS336B/G	36	52.23	107.03	26.18
CSLESS338B/G	38	57.94	118.25	29.17
CSLESS340B/G	40	63.93	130.48	32.32
CSLESS342B/G	42	70.00	142.71	35.64
CSLESS344B/G	44	77.14	156.98	39.11
CSLESS346B/G	46	83.49	171.25	42.75
CSLESS348B/G	48	90.62	184.51	46.54
CSLESS350B/G	50	97.76	199.80	50.50

Figure 55: Chain properties

Tutorial 38 - Hydrodynamic feature

1 Tutorial overview

This tutorial covers:

- High-level overview of creating a hydrodynamic database using a potential flow solver
- Using the ProteusDS RigidBodyRadDiffHydrodynamic feature
- Replacing RigidBodyRadDiffHydrodynamic's linear hydrostatic model with a non-linear buoyancy model
- Replacing RigidBodyRadDiffHydrodynamic's viscous resistance model with a different viscous loading model
- Single point mooring

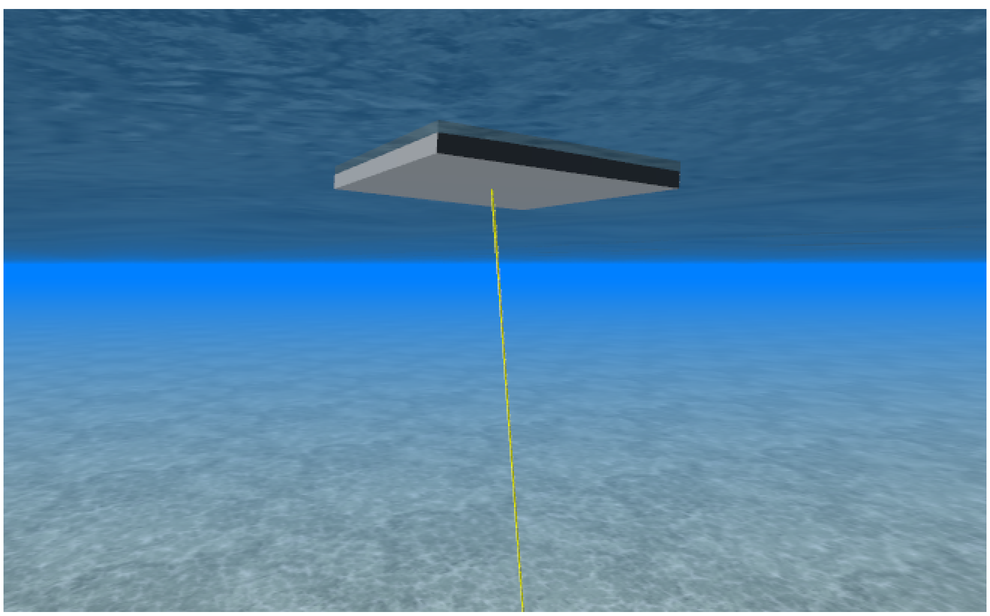


Figure 56: RigidBody DObject floating in waves

2 Using a hydrodynamic database in ProteusDS

☞ The RigidBodyRadDiffHydrodynamic feature can be used to model wave radiation forces, wave excitation forces including diffraction, hydrostatic forces and some viscous resistance forces.

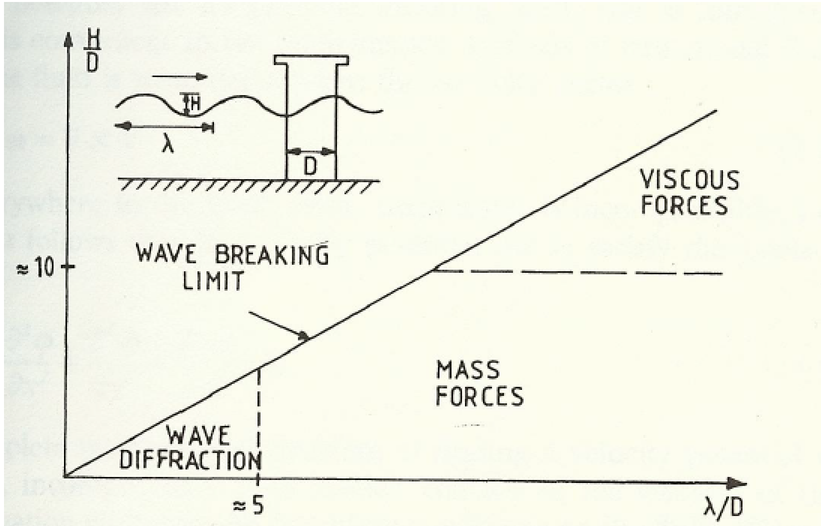


Figure 57: Relative importance of mass, viscous drag, and diffraction forces on marine structures (Faltinsen, 1990)

- ☞ It is reasonable to ignore radiation and diffraction loads when the characteristic length of the object in question can be said to be much smaller than the incident wavelengths, as described in Figure 57. This is not the case for the 20m by 20m by 2m barge, as the wavelength of the incoming waves is on the same order as the length of the barge.
- ☞ To include wave radiation and diffraction loading in PDS, a database of the added mass, damping and loading as a function of wave frequency, heading and potentially also forward speed is required. This database can be obtained using a frequency domain potential flow solving tool such as ShipMo3D or WAMIT.
- ☞ Incident wave loads, hydrostatic loads, and viscous resistance loads can be disabled and instead computed using a different RigidBody feature such as the Cuboid feature as discussed in Sections 7 and 10.
- ☞ More information on the hydrodynamic feature can be found in the ProteusDS Manual.
- ☞ If you do not have a potential flow solving tool, you may skip to Section 3 and use the pre-generated database provided with this tutorial.

3 Creating the hydrodynamic database

- ✪ Using a panel method potential flow hydrodynamics solver (e.g. ShipMo3D, WAMIT, InWave, Moses), define the floating body's geometry as a flat 20m by 20m block with a height of 2m, a draft of 1m and the CG located in the geometric center.
- ✪ Generate a mesh for the solver using around 1000 panels as shown in Figure 58.
- ✪ Compute a hydrodynamic forces database using the mesh, including radiation and diffraction effects.
- ☞ Radiation effects should be computed for a range of encounter frequencies between 0.01 rad/s to 10.01 rad/s with 0.5 rad/s increments, wave headings between 0 and 360 in 15 degree increments and forward speeds between 0 m/s and 1 m/s with 0.5 m/s increments.
- ☞ It is important that the infinite frequency added mass and damping are properly reflected by the highest frequency entries in the database. That is, the frequency dependent added mass and damping results should have converged to the infinite frequency values for the highest frequencies in the database.
- ☞ Wave excitation loads can be computed from the same headings and speeds but only require data for the frequency range that will be encountered in simulation.

✪ Compute wave excitation forces for a range between 0.2 rad/s to 4 rad/s with 0.2 rad/s increments. This will create a coarse database for computing hydrodynamic forces with a large enough range to cover any waves the block might encounter.

☞ For waves modeled using a spectrum like JONSWAP, ProteusDS will create waves of 1/4 of the peak frequency to 4x the peak frequency.

✪ Define the radius of gyration of 5.8 m, 5.8 m and 8.2 m for the body about its $\hat{\mathbf{x}}$, $\hat{\mathbf{y}}$, and $\hat{\mathbf{z}}$ axes respectively.

☞ Forward resistance coefficients based on wetted surface area can be provided to ProteusDS as a function of forward speed to model forward motion resistance for ships. Lateral drag coefficients can also be defined which themselves are based on the submerged lateral projected area. For this tutorial, these resistance models will be disabled and modelled using a Cuboid feature; this is discussed in Section 4.

✪ After creating the database, it must be exported to the input ProteusDS format. Save the file as '20x20x2Block.ini'. ShipMo3D has this export function in the BuildShip interface.

☞ The parameters required in the database are defined in the RigidBody section of the ProteusDS Input Files appendix in the manual.

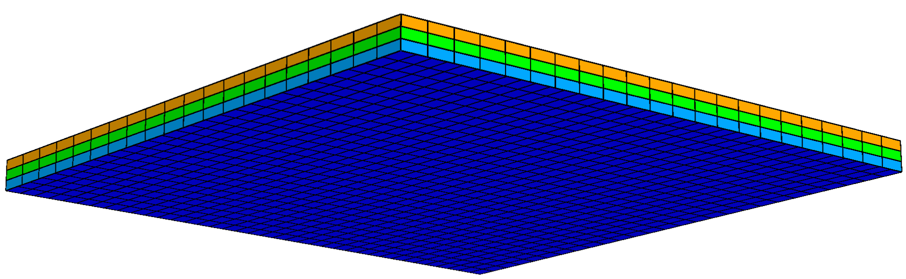


Figure 58: 20m×20m×2m box as meshed by ShipMo3D, showing wet hull only.

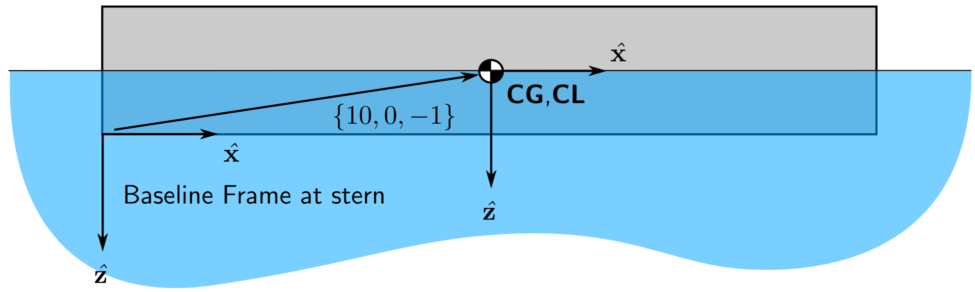


Figure 59: The baseline frame for the 20mx20mx2m box showing the location of the center of loading frame.

4 The hydrodynamic database

- ✪ View the contents of the '20x20x2Block.ini' file that was produced in the previous section (or the one was provided to you with this tutorial).
- ✪ Note how there are a large number of ProteusDS properties defined here.
- ☞ These properties once loaded into ProteusDS provide the database required to compute hydrodynamic forces as a function of forward speed, encounter frequency and heading.
- ☞ One property worth noting is $CLPositionFromBaseLine which defines the location of the center of loading for the database relative to the stern/baseline frame as shown in Figure 59. The center of loading frame is the reference frame about which all loads are defined in the database.
- ☞ The baseline frame is a frame about which the geometry was defined. In ShipMo3D, the geometry is defined about a frame located at the stern and baseline of the ship.
- ☞ In ProteusDS, the hydrodynamic database and its center of loading are located relative to the rigid body frame by specifying the location of the baseline frame.
- ☞ For more information on the various parameters defined in this file, refer to the RigidBodyRadDiffHydrodynamic feature section of the ProteusDS manual.

5 Using the hydrodynamic feature

- ☞ This section assumes that a hydrodynamic database was exported/converted from a potential flow solver to ProteusDS' hydrodynamic database input file format. This database can be created by following the previous sections, or the reader can use the database provided with this tutorial as a file named '20x20x2Block.ini'.
- ✪ In PST, create a new project and move the hydrodynamic database that was created or provided, '20x20x2Block.ini', into the project folder.
- ✪ Create a new rigid body object and call it *barge*.
- ✪ Set the *barge* rigid body's mass and moments of inertia as shown in the Rigid body properties block below.

Rigid body properties

```
// Mechanical
$Ix 1.38e7
$Iy 1.38e7
$Iz 2.7e7
$Ixy 0
$Ixz 0
```

```
$Iyz 0
$Mass 4.1e5

$RadDiffHydrodynamicModel hydroBlock -10 0 1
```

☞ All environmental loading for the *barge* will be handled by the RigidBodyRadDiffHydrodynamic feature. The feature uses the hydrodynamic database which contains frequency dependent added mass and damping data. Incident wave loads, wave diffraction loads, hydrostatic loads, wave radiation loads will be calculated and applied to the *barge*.

✪ Create a 'RigidBodyRadDiffHydrodynamic' feature and call it *hydroBlock*.

✪ Add the property $RadDiffHydrodynamicModel hydroBlock -10 0 1 to *barge*. The last 3 numbers specify the $\hat{\mathbf{x}}$, $\hat{\mathbf{y}}$ and $\hat{\mathbf{z}}$ coordinates of the hydrodynamic database's baseline frame relative to the *barge* rigid body's body-fixed frame.

☞ Note that in this case, the rigid body frame, the center of gravity and the center of loading frame are all coincident. ProteusDS locates the hydrodynamic database using the baseline frame which is not coincident.

✪ In the *hydroBlock* feature properties, ensure the $HydrodynamicDatabaseFile is referring to the correct database file (20x20x2Block.ini), and that the file is located in the simulation input directory.

✪ Disable the viscous resistance model by setting ViscousResistance to 0. Viscous force modelling will be handled using a Cuboid feature, described in Section 7.

☞ The *hydroBlock* feature's properties should look as follows:

RadDiffHydrodynamic feature properties

```
// Data
$HydrodynamicDatabaseFile 20x20x2Block.ini

// Fluid loading
$DiffractionLoading 1
$HydrostaticLoading 1
$IncidentLoading 1
$WaveRadiationLoading 1
$ResistanceLoading 0
```

6 Setting up the Environment

☞ The barge will be placed in a single Airy wave. It should follow the wave motion.

✪ Set the depth of the water to 100 meters.

✪ Initialise waves to 7 second 1.5 meter Airy waves heading North (0 degrees).

Environment properties

```
// Air
$AirDensity 1.29
$AirKinematicViscosity 1.568E-05

// Current
$CurrentProfile 0

// Seabed
$WaterDepth 100
```

```
$CustomBathymetry 0
$SoilProperties defaultSoilProperties
$UseMultipleSoilLayers 0

// Water
$WaterDensity 1025
$WaterKinematicViscosity 1.8E-06

// Wave
$WaveType 1
$WaveHeading 0
$WaveHeight 1.5
$WavePeriod 7.0
```

7 Creating CustomMesh for viscous modelling and for visualization

☞ The RadDiffHydrodynamicModel does not create a mesh for PostPDS to visualize. A separate feature can be defined with all environmental loading disabled for visualisation purposes. Here, a Cuboid feature will be used to both provide a mesh for visualisation as well as to provide viscous load modelling.

✪ Add a RigidBodyCuboid feature called *viscousVizBlock*.

Cuboid properties

```
// Fluid loading
$CDt 0.0

// Uncategorized properties
$LengthX 20
$LengthY 20
$LengthZ 2
$CAx 0
$CAy 0
$CAz 0
$CDx 1
$CDy 1
$CDz 1
$SegmentsX 5
$SegmentsY 5
$SegmentsZ 5

$HydroLoading 1
$WindLoading 0
$BuoyancyFroudeKrylov 0
```

☞ Note that BuoyancyFroudeKrylov model is disabled. This is because the RadDiffHydrodynamic feature is already modelling the hydrostatic forces. Enabling it would result in double counting of buoyancy and incident wave forces.

☞ Wind modelling is not important for this tutorial, it has also been disabled.

☞ The added mass coefficients were set to 0 since the RadDiffHydrodynamic feature is already modelling that effect.

☞ The $HydroLoading parameter enables the modelling of viscous forces. Viscous forces are affected by the drag coefficients $CDx, $CDy and $CDz for modelling drag based on frontal area as well $CDt for modelling drag based on wetted surface area. For more information on how the Cuboid feature models viscous forces, the reader is referred to the ProteusDS manual.

☞ The Cuboid feature's mesh resolution is set using SegmentX, SegmentY and SegmentZ which defines how the Cuboid is discretised in the X, Y and Z directions respectively. They are all set to 5 which creates a mesh with 150 faces. The resolution of the environmental forces will converge with increasing mesh resolution.

✪ Add the Cuboid feature to the rigid body at the rigid body reference frame with the same orientation.

Rigid body properties

```
// Mechanical
$Ix 1.38e7
$Iy 1.38e7
$Iz 2.7e7
$Ixy 0
$Ixz 0
$Iyz 0
$Mass 4.1e5

$RadDiffHydrodynamicModel hydroBlock
$Cuboid viscousVizBlock 0 0 0 0 0 0
```

8 Adding a mooring line

✪ Add a cable called *mooring*.

✪ Set the initial conditions such that node 0 is at (0, 0, 0) and node N is at (10, 0, 100).

✪ Set the cable length to 100m with 10 cable elements.

✪ Connect the cable to the barge at the center.

✪ Create the anchor point at node N by enabling the $NodeNStatic flag

9 Run the simulation

✪ Run the simulation for 20 seconds.

☞ Watch the motion of the barge and check to be sure that it is following the wave motions.

10 Modelling buoyancy and incident wave using non-linear models

☞ If non-linear buoyancy and incident wave loads are important to accurately capture the dynamics of the system, they hydrostatic and incident wave loads can be disabled in the RigidBodyRadDiffHydrodynamic feature and modelled with a CustomMesh feature instead.

☞ CustomMesh based non-linear buoyancy and incident wave loading comes at the cost of execution speed over the HydrodynamicDatabase feature's hydrostatic and incident wave loading.

✪ Disable the $IncidentLoading and $HydrostaticLoading parameters in the RigidBodyRadDiffHydrodynamic feature's properties:

RadDiffHydrodynamic feature properties

```
// Data
$HydrodynamicDatabaseFile 20x20x2Block.ini

// Fluid loading
$DiffractionLoading 1
$HydrostaticLoading 0
$IncidentLoading 0
$WaveRadiationLoading 1
$ResistanceLoading 0
```

✪ Enable $BuoyancyFroudeKrylov modelling in the Cuboid feature's properties:

Cuboid properties

```
// Fluid loading
$CDt 0.01

// Uncategorized properties
$LengthX 20
$LengthY 20
$LengthZ 2
$CAx 0
$CAy 0
$CAz 0
$CDx 1
$CDy 1
$CDz 1
$SegmentsX 5
$SegmentsY 5
$SegmentsZ 5

$HydroLoading 1
$WindLoading 0
$BuoyancyFroudeKrylov 1
```

☞ The Cuboid feature's Buoyancy and Froude-Krylov modelling component computes the same hydrostatic and incident wave loading forces as the RigidBodyRadDiffHydrodynamic feature except that can handle variable submergence more accurately for large displacements since it is a non-linear model.

11 Rerun the simulation

✪ Run the simulation for 20 seconds.

☞ Watch the motion of the barge and check to be sure that it is following the wave motions.

Tutorial 39 - Linear quadratic drag feature

1 Tutorial overview

- ☞ This tutorial reviews the use of the RigidBodyLinearQuadraticDrag feature. This feature applies linear and quadratic damping to a rigid body regardless of its level of submersion.
- ☞ This feature could be used, for example, to apply roll damping to a floating body or model the drag behaviour of an unmanned underwater vehicle (UUV) such as a towed body, AUV (autonomous underwater vehicle), or ROV (remotely operated vehicle).

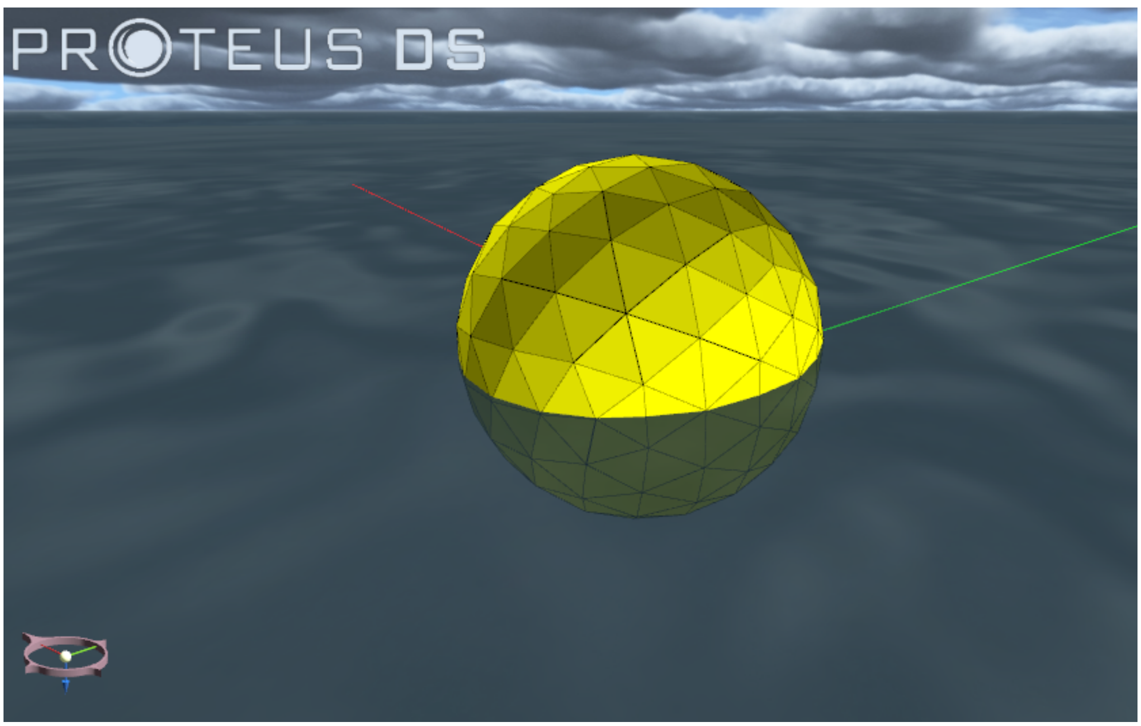


Figure 60: Sphere with linear and quadratic damping

2 Linear and quadratic damping

- ☞ In general, a rigid body obeys the Newton-Euler equations for rigid body motion. When a new RigidBody is created in ProteusDS, no drag, damping, buoyancy or control forces will apply loads to a rigid body. For example, to apply loads to a rigid body, a hydrodynamic feature (e.g. spheroid, cuboid) could be added, or a connection made with a cable, or a controller connection made.

☞ A simplified representation of the rigid body dynamics can be written as:

$$M\dot{v} + C(v)v + D(v)v + F_g = F_{env} + F_{cable} + F_{contact} + ...$$

☞ In the above equation, M is the combined added mass and inertia matrices, C accounts for coriolis and centripetal effects, and D is the total system damping. F_g is gravitational and restoring effects, F_{env} is the applied wind and wave loads, F_{cable} are the forces and moments applied to the body due to the cable connections, and $F_{contact}$ are soil contact forces. Additional forces could be added and in ProteusDS may include, joint forces or controller forces. The velocity v is the combination of the ocean currents and the body relative velocity in terms of the rigid body local coordinate frame[1].

☞ The damping or drag on a rigid body that is either deeply submerged or floating ($D(v)v$) is often conveniently represented by linear and nonlinear (or quadratic) components. Many sources of damping may exist for a hydrodynamic body including: radiation damping, skin friction, wave drift damping, lifting forces, pressure induced drag, vortex shedding damping. It is often difficult to separate these effects, so the damping on a rigid body may be adjusted or fully represented using linear and quadratic drag matrices which might be identified through experiment, or other means (e.g. experience). Consider the relative velocity v_r, such that the damping of a rigid body is represented as the combination linear and quadratic terms:

$$D(v_r) = D_{linear} + D_{quadratic}(v_r)$$

☞ In the above equation, D_{linear} and $D_{quadratic}$ are 6x6 damping matrices. Given that v is the body velocity in the local coordinate frame: $v = [u\ v\ w\ p\ q\ r]^T$ and likewise the relative velocity is $v_r = [u_r\ v_r\ w_r\ p_r\ q_r\ r_r]^T$, the linear drag (or damping) matrix takes the form:

$$D_{linear} = -\underbrace{\begin{bmatrix} X_u & X_v & X_w & X_p & X_q & X_r \\ Y_u & Y_v & Y_w & Y_p & Y_q & Y_r \\ Z_u & Z_v & Z_w & Z_p & Z_q & Z_r \\ K_u & K_v & K_w & K_p & K_q & K_r \\ M_u & M_v & M_w & M_p & M_q & M_r \\ N_u & N_v & N_w & N_p & N_q & N_r \end{bmatrix}}_{\$LinearDragCoefficients}$$

☞ The quadratic drag (or damping) matrix is:

$$D_{quadratic}(v_r) = -\underbrace{\begin{bmatrix} X_{uu} & X_{vv} & X_{ww} & X_{pp} & X_{qq} & X_{rr} \\ Y_{uu} & Y_{vv} & Y_{ww} & Y_{pp} & Y_{qq} & Y_{rr} \\ Z_{uu} & Z_{vv} & Z_{ww} & Z_{pp} & Z_{qq} & Z_{rr} \\ K_{uu} & K_{vv} & K_{ww} & K_{pp} & K_{qq} & K_{rr} \\ M_{uu} & M_{vv} & M_{ww} & M_{pp} & M_{qq} & M_{rr} \\ N_{uu} & N_{vv} & N_{ww} & N_{pp} & N_{qq} & N_{rr} \end{bmatrix}}_{\$QuadraticDragCoefficients} \begin{bmatrix} |u_r| \\ |v_r| \\ |w_r| \\ |p_r| \\ |q_r| \\ |r_r| \end{bmatrix}$$

☞ In ProteusDS, $QuadraticDragCoefficients and $LinearDragCoefficients as shown in the above equations are specified in the RigidBodyLinearQuadraticDrag feature.

3 Sphere with roll, pitch and yaw damping

✪ Create a new project in PST and add a new RigidBody

✪ Add an Ellipsoid feature to the rigid body, creating a sphere hydrodynamic object of $1m$ in diameter.

[1] For more information on the rigid body dynamics model and hydrodynamics model consider *Handbook of Marine Craft Hydrodynamics and Motion Control* by Thor I. Fossen. (2011 by John Wiley & Sons Ltd.)

- Set the CDt property for the Ellipsoid to 0.
- Set the mass of the rigid body to 268kg such that the body will float at the surface half submerged.
- Set the $Ix, $Iy, and $Iz to 26.8 kgm^2, which is mass moment inertia of a sphere with a 1m diameter.
- Define the IC of the rigid body such that the sphere has an initial heading rate of 10deg/s.
- Run a simulation for 100 seconds and view the results in PostPDS.
- Plot the yaw velocity of the sphere. Note that it takes approximately 100 seconds for the yaw velocity of the sphere to reach 0 deg/s.

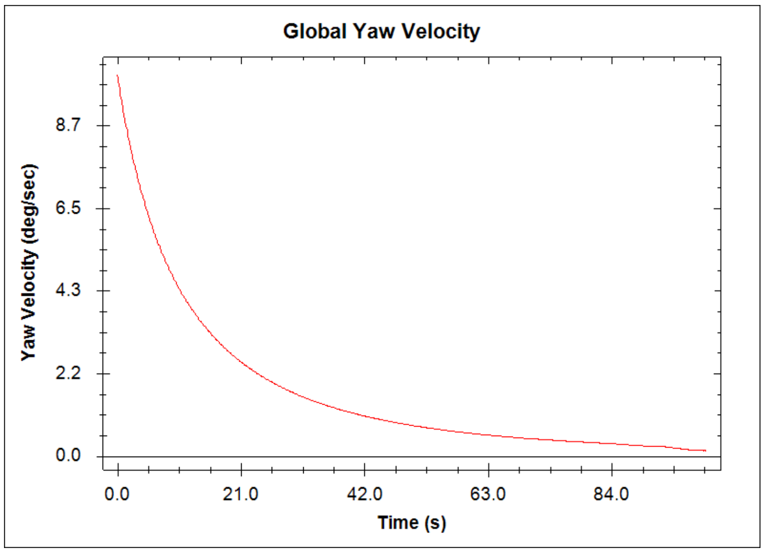


Figure 61: Yaw velocity of sphere

- To change the behavior of the sphere in yaw, it is possible to set the skin friction drag of the sphere CDt to a larger value (e.g. 0.1). However, this is not advisable because this will also increase the drag of the sphere in translational degrees of freedom (i.e. surge, sway and heave). Instead of adjusting CDt to change the yaw motion of the sphere, add a new RigidBodyLinearQuadraticDrag feature to the Library in PST; this feature will allow specification of roll, pitch and yaw damping that is decoupled from translational velocity of the body.
- Add the newly created RigidBodyLinearQuadraticDrag feature to the rigid body using the $LinearQuadraticDrag property in the rigid body input file. Set the feature location to be at the origin, aligned with the rigid body local coordinate frame.

RigidBody input file

```
// Mechanical
$Ix 26.8
$Iy 26.8
$Iz 26.8
$Ixy 0
$Ixz 0
$Iyz 0
$Mass 268 //kg

$Ellipsoid RigidBodyEllipsoid_1 0 0 0 0 0 0

$LinearQuadraticDrag RigidBodyLinearQuadraticDrag 0 0 0 0 0 0
```

- ✪ Add quadratic roll, pitch and yaw damping to the the RigidBodyLinearQuadraticDrag feature that was created: $K_{pp} = M_{qq} = N_{rr} = 100$.
- ✪ In addition, recall that in the initial simulation the rigid body heaves constantly. To increase the damping on the rigid body at low speeds in heave, add a linear heave damping of 500: $Z_w = 500$.

RigidBodyLinearQuadraticDrag feature

```
// Fluid loading
$LinearDragCoefficients 0 0 0 0 0 0
$LinearDragCoefficients 0 0 0 0 0 0
$LinearDragCoefficients 0 0 500 0 0 0 // Z_w = 500
$LinearDragCoefficients 0 0 0 0 0 0
$LinearDragCoefficients 0 0 0 0 0 0
$LinearDragCoefficients 0 0 0 0 0 0

$QuadraticDragCoefficients 0 0 0 0 0 0
$QuadraticDragCoefficients 0 0 0 0 0 0
$QuadraticDragCoefficients 0 0 0 0 0 0
$QuadraticDragCoefficients 0 0 0 100 0 0 //K_pp = 100
$QuadraticDragCoefficients 0 0 0 0 100 0 //M_qq = 100
$QuadraticDragCoefficients 0 0 0 0 0 100 //N_rr = 100
```

☞ The two figures above show the heave motion and yaw velocity of the sphere. Note that now the heave motion of the sphere is damped, and the yaw velocity reduces to zero quicker with the increased quadratic drag.

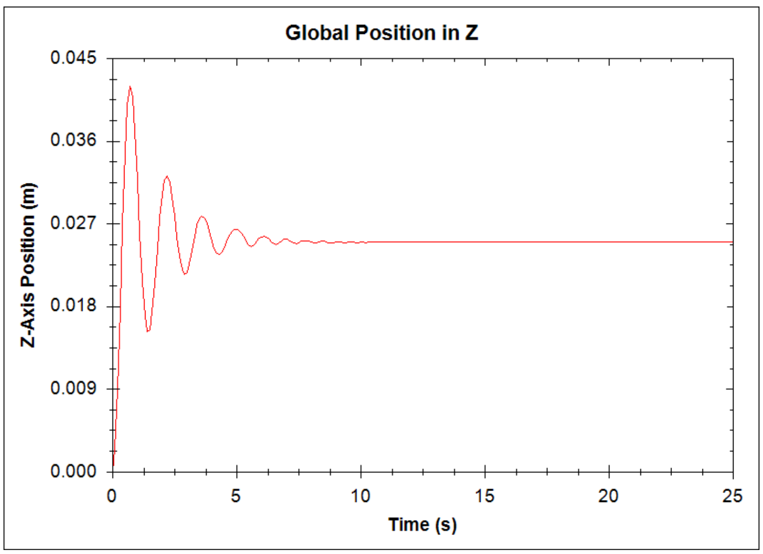


Figure 62: Heave motion of sphere with linear heave damping

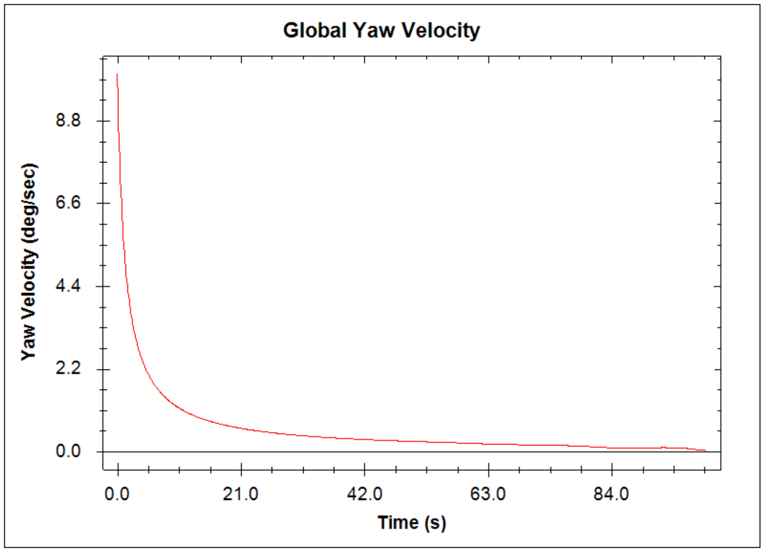


Figure 63: Yaw velocity of sphere with quadratic damping