

# F01 Lowering hydrodynamics

## Introduction

In these three examples, we consider a number of effects that often need to be taken into account when doing a lifting or lowering analysis through the sea surface, and show how to model them in OrcaFlex.

Firstly, a complex manifold structure is modelled using a number of objects connected together, and is lowered through the sea surface (*F01 Manifold block*). Hydrodynamic effects are applied, including slam forces.

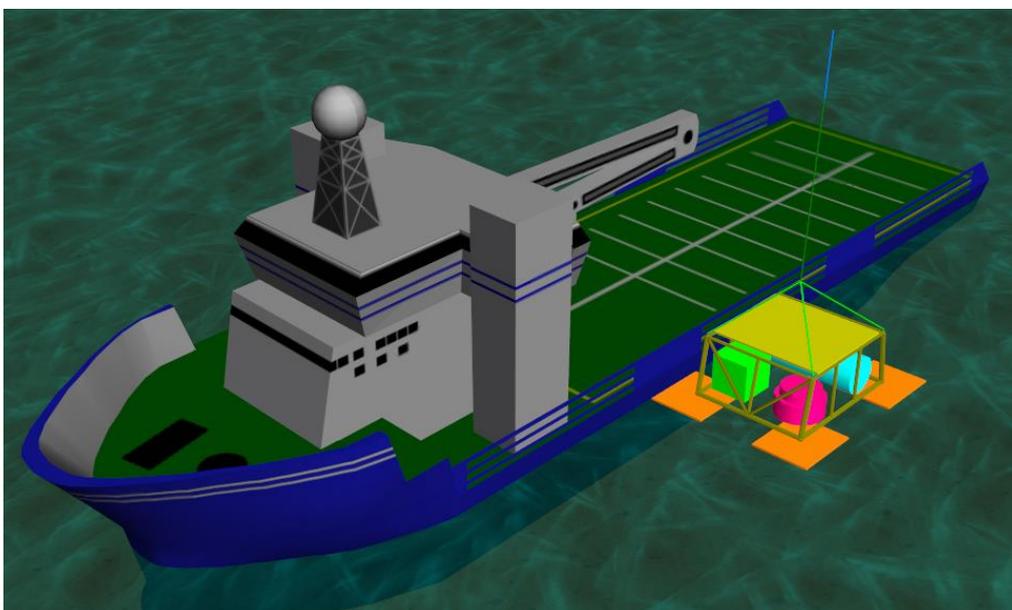
In the next model (*F01 Lowered cone*), we use a simple cone-shaped object lowered in still water to look at modelling slamming in more detail.

Finally, we look at how to model the effects of water flooding into compartments and becoming 'trapped' within the structure as it is lowered into the water (*F01 Trapped water*). Again, we use a simple object to explain the modelling methodology. For a more complex example involving trapped water, see our separate example on suction anchor installation: *F07 Suction anchor lowering*.

## Manifold block

### Building the model

On opening the simulation file, *F01 Manifold block.sim* you will see two views of the model, one in shaded mode, the other in wire frame.



**Figure 1: Manifold block**

The first thing to consider when building an OrcaFlex model of a lifting or lowering operation, is how the object's size affects the choice of OrcaFlex objects used to model it.

Crane lifting operations are often divided into two categories – light lifts and heavy lifts. DNV-RP-H103 gives guidance on this, but typically, if the lifted object is very small compared to the vessel

(< 1-2% of the displacement of the vessel) then the motion characteristics of the vessel are not affected by the lifted object. This is classed as a light lift.

If the lifted object is more than 1-2% of the vessel displacement, then this is classed as a heavy lift, and coupled dynamics of the vessel and the lifted object must be considered.

In OrcaFlex terms, this means that a light lift is modelled using a vessel whose motion is fully described by displacement RAOs. A heavy lift requires a calculated vessel, and therefore the *calculated (6DOF)* option is required, along with appropriate choices of *included effects* and *primary motion is treated as:* options (see OrcaFlex Help *Modelling, data and results | Vessels | Vessel data | Calculation data* for further details).

In this model, the lifted object falls into the light lift category, and therefore the use of displacement RAOs for the vessel is appropriate. Open the vessel's data form and look at the *calculation* page to see how this method has been selected.

A second consideration regarding the lifted object's size is whether the object should be modelled using diffraction theory or drag/inertia theory. In this example, we are assuming that the lifted object is small compared to the wavelength, and therefore modelling the lifted object with Morison objects is appropriate. In OrcaFlex, the Morison objects are lines and buoys.

If the lowered object is large enough to diffract the waves, then it should be modelled with a diffraction object i.e. a vessel, with its motion fully calculated. This method will not be considered further in this example.

So here, we are using Morison objects to model the lowered object (payload), and a vessel whose motion is controlled by displacement RAOs.

If the payload is passing through the sea surface, then you need relatively accurate modelling of the distribution of buoyancy and hydrodynamics loads about the structure. The easiest way to achieve this is to use multiple objects connected together to model the complete payload, rather than using a single 6D buoy, which may not be sufficient.

A single 6D *lumped* buoy handles changes in buoyancy and hydrodynamic loads in a simple way, and as a result may not be an accurate enough model for this type of analysis. A *spar*-type 6D buoy is better in that it is possible to discretise it axially, but there is no means to automatically discretise it in other directions. The spar buoy object is only really suitable for modelling cylindrical objects because the buoyancy is calculated from the cylindrical geometry. So for anything other than a cylindrical object the solution is to use several line and/or 6D buoy objects, connected together, to model the payload.

If the payload is deeply submerged however, and remains so throughout the simulation, then using a single buoy object can be sufficient.

When using multiple objects to model the payload, they need to be connected together to create a single rigid body. In the example model, it is convenient to use a reference buoy as a master (*Master Buoy*), and multiple lines (used to model the framework) and 6D buoys (used to model the objects within the frame) are connected to it. The mass and mass moments of inertia data can either be applied to the master buoy if you know these properties for the complete structure, or to the individual components. Note that if applying this data to the individual components, each individual mass moment of inertia is defined for the object itself *about its own, local centre of mass*. This means that additional inertia due to the offset of the object from the

overall centre of mass (parallel axis theorem) will be automatically accounted for by its offset connection.

Object volumes should be split out among the components for maximum accuracy, as should drag and added mass properties. Note that OrcaFlex does not automatically model shielding effects between components, therefore the user must take into account the effect of shielding when assigning drag areas and coefficients. It is important to ensure that the total mass, volume, fluid inertia and drag are correct for the overall structure i.e. nothing is missed and nothing is double-counted. The data we have applied in this model are purely arbitrary and the user should assess each individual case on its own basis.

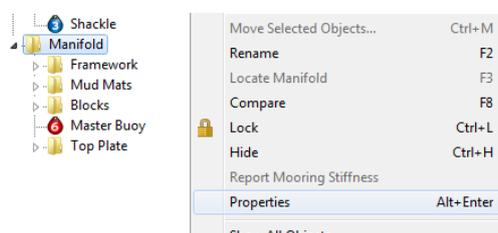
Data sources for these properties only cover basic geometries; if dealing with a particularly complex system then physical testing or CFD modelling are the only ways to get accurate values.

In the model, make sure the model browser is in *view by groups* mode, and then look at the various objects in the *Manifold* group. As this is a complex model structure, the *Manifold* group is also split into sub-groups containing different component groups, to make items easier to find. The *Mud Mats* sub-group is split down further into groups containing the objects making up each separate mud mat. Open up the data form for any of the mud mat 6D Buoys (e.g. 'MudMat Port 1') and look at the Properties page.

Each mud mat has been assigned a Slam Area, based on the surface area of the underside of the mud mat, and Slam Force Coefficients for both entry and exit. Slamming will be discussed in detail in the following example. Note that drag and added mass data have only been assigned to translations in the buoy's z direction; this is because the mats are considered to be thin enough in this case to have negligible drag or added mass effects on motion in the other directions. The rotational terms have all been set to zero, because some drag and added mass moments will arise naturally from the fact that the mud mats are discretized, and therefore the distributed translational drag and added mass forces will also resist rotations.

The line objects used to model the frame are all single segment lines, which means that they cannot bend. Both ends of each line are connected to the master buoy, so the lines move rigidly with the rest of the structure.

Because the manifold block is made up of a number of objects, it is often useful to check the overall properties of the structure. This can be done by getting a *compound properties* report. In the model browser, right mouse click on the *Manifold* group and select *properties*.



**Figure 2: Compound properties**

The properties report shows the total properties for all the selected objects, under the assumption that they all move together as a rigid body, so in this case that's the total mass, wet weight etc. for the complete manifold structure. The centre of mass and centre of volume are reported relative to a specific object; OrcaFlex by default will select the master object if such an

object exists (in this case it's the Master Buoy to which all other manifold objects are connected), but you can choose to report these values relative to any other object in the selected set.

Line feeding is used at *end A* of the *Crane Wire* to lower the object into the water. The full length of wire required must be included in the line's structural properties, therefore on the *structural* page of the *Crane Wire* data form, the length of the wire is defined as 20m. On the *feeding* page, the *initial arc length* of *end A* is set to 8m, which means that only 12m of wire is present in the model initially, with the remaining 8m being inactive, but available to be paid out. The crane payout rate is specified in a variable data source, which ramps up the payout rate from zero to 0.125m/s during Stage 0 of the dynamic simulation, then pays out wire at a constant 0.125 m/s for the rest of the simulation.

A short time step is required in this model (0.001s) to enable the transient slam loads that occur as the different objects cross the sea surface to be accurately modelled. A sensitivity study should always be carried out on time step size to ensure that it is short enough to model the particular situation accurately. Running the same case with the explicit integration scheme is often a good way to check that the chosen implicit integration scheme time step is short enough.

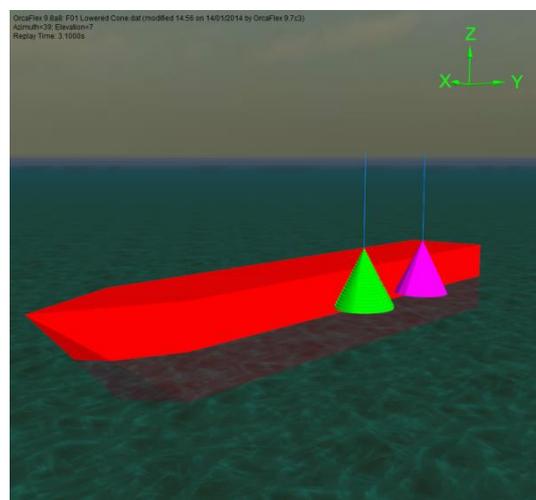
This simulation models the response of the manifold block to one specific crossing of the wave environment. A large number of realisations of any given sea state would be required to provide confidence that the variations in the environmental conditions had been adequately modelled.

## Results

Open the Workspace *F01 Manifold block results.wrk*. The results graphs show time histories of the crane wire top tension (upper right hand graph), the payload's Z position (bottom left hand graph), and the slam forces occurring on one of the mud mats and one of the roof panels (middle-right and bottom-right graphs respectively). Replay the simulation to see the manifold's motion as it passes through the sea surface.

## Lowered Cone

In this model, we use a simple cone-shaped payload to explain how water entry and exit slam forces are modelled on 6D buoys. For slamming on lines, please see example *F06 Spoolpiece Manoeuvring*.



**Figure 3: Lowered cone**

## Building the model

Open the model *F01 Lowered Cone.sim*. In this example, two cone-shaped buoys are lowered into still water. In this case, the mass and hydrodynamics of the crane wires are assumed to be negligible, and therefore a simple winch object is used to perform the lowering. To demonstrate the differences between them, one of the cones is modelled with a spar-type buoy (coloured green), the other is a lumped buoy (pink).

The spar-buoy cone consists of a stack of cylinders, of gradually increasing outer diameter. This option means that the buoy has the physical geometry that you see displayed in the model. The lumped buoy however, while having the *appearance* of being a smooth cone, has no physical geometry associated with it. The cone shape simply comes from the drawing data that is applied on the drawing page of the *Lumped Cone* buoy.

The spar buoy, with its defined geometry, provides much better modelling of surface piercing objects, although multiple lumped buoys can be connected together, as described in the previous example, to create more detailed objects with better discretised properties.

When a lumped buoy pierces the surface, it is modelled as a simple vertical stick for the buoyancy calculation. The length of this stick is equal to the specified Height property of the buoy, and the centre point of the stick is placed at the buoy's *centre of volume*. In the example, the centre of volume of the *Lumped Cone* buoy is positioned 2.567m above the buoy's origin. This means that the Height which is set to 10m (the overall height of the buoy), extends vertically above and below this point by 5m. Open the workspace *F01 Lowered cone dry length.wrk* to see the impact of these choices.

The workspace shows the *dry length* results for the two cones – in other words, how much of the cone is out of the water. The top graph shows the *dry length* of the *Lumped Cone*, and the simulation is paused at 4.5s, which is the time that the Dry Length result shows that the cone first touches the sea surface. However if you look at the position of the cone in the model, it is clearly *visually* not yet touching the sea surface. This is because the 'stick', which is protruding vertically above and below the buoy's centre of volume by 5m *has* now reached the sea surface.

For a lumped buoy, the change in buoyancy and hydrodynamic load with submergence is linear. The amount of the buoy that is submerged is approximated by the proportion of the *height* 'stick' that is submerged. This *proportion wet* is used to scale the buoyancy and hydrodynamic loads as the buoy crosses the sea surface.

So a single lumped buoy provides a poor estimation of the variation with buoyancy as the cone crosses the sea surface. A better option for a surface piercing object is to use multiple lumped buoys connected together, or to use a spar buoy.

Look at the *dry length* results for the *Spar Cone* in this model (bottom right hand graph). Move the replay on to the point where the *dry length* starts to reduce, by using the dropdown arrow next to the *step replay* button to set the time to 7.5s.



**Figure 4: Select Replay Time**

Notice how the spar buoy only reports a reduction in the dry length when it actually does pass through the sea surface because its change in buoyancy is determined by its geometry.

Slamming has been applied to both cones; for the spar buoy this is done by assigning an entry slam coefficient ( $C_s$ ) and an exit slam coefficient ( $C_e$ ) to the lower-most cylinder, but these could be applied to each cylinder if deemed more appropriate. Open the [Spar Cone](#) buoy data form and look at the [drag & slam](#) page. Scroll down the page to see the slam coefficient data for cylinder 20.

OrcaFlex uses the direction of the relative velocity between the buoy and the sea surface to decide whether it is 'entering' or 'leaving' the water, and therefore which coefficient to use. The slam area used in the slam calculation is (in the case of a spar buoy) taken to be the instantaneous waterplane area, which is calculated from the buoy's geometry and the instantaneous position and orientation of each cylinder relative to the sea surface. The slam force is calculated for each cylinder of the buoy that has a slam coefficient assigned, and is applied to the instantaneous centre of the slam area of that cylinder.

For the lumped buoy, which doesn't have defined geometry, we also need to specify the [slam area](#), along with the  $C_s$  and  $C_e$ . Note that the area is a fixed value, and, unlike the spar buoy, does not change with buoy orientation. The buoy's height is used to determine the 'distance' over which the slam force is applied. So if the buoy is 10m high and it has a slam area and coefficients applied to it, then the slam force will be applied for the full duration of the buoy's height crossing the water surface, which is probably not correct because the slam force only arises when surfaces are first coming into contact with the water.

In the case of a lumped buoy, if you don't want the slam force to be applied over the full height of the buoy, the simplest solution is to create a second 'slam' buoy to handle the slam force only, leaving the original buoy to handle everything else. In the model, this has been done by adding a buoy called [Slam Buoy](#). Open the data form for this buoy and look at the properties. It has negligible properties for everything except the slam area, slam coefficients and the height. Negligible properties can be set by simply pressing the [give buoy negligible properties](#) button on the bottom of the buoy's data form. The height is set to match the 'distance' over which you want the slam force to be applied. Be careful however not to set the height to be too small, as this could mean that the slam event is missed completely, unless the time step used is suitably small. Also, slam force results are not spike-logged (see the OrcaFlex Help file [User interface | Results | Producing results](#) for further details on spike logging), therefore a small logging interval must also be used.

## Results

Open the workspace file [F01 Lowered Cone results.wrk](#) to see the [slam force](#) results reported for the two cones. The spar buoy version shows a slightly higher peak slam force than the lumped buoy. The difference will be due to the lumped buoy contacting the water surface first (because of the simplistic handling of proportion wet, as described above) which slows the lumped buoy's lowering speed and reduces the slam force experienced. Comparison of the [GZ-velocities](#) for the [Spar Cone](#) and [Slam Buoy](#) objects confirms that this is the case. This adds further weight to the guidance that a single lumped buoy is not recommended for modelling surface piercing objects, and that a spar buoy, or multiple lumped buoys should be used instead.

## Trapped Water

In this model (*F01 Trapped water.sim*), we consider how to model water that gets trapped within a structure when it is submerged, and hence contributes to the overall mass and inertia of the structure. It is convenient to think of the trapped water as simply being another component in the structure, which moves rigidly with the body.

For clarity, we will again consider simple objects for our payload, comprising of a single 6D buoy, but the theory can be applied to much more complex structures.

### Building the model

The easiest way of adding inertia due to trapped water is to simply add extra mass to the object. Specifically you need to:

- Increase the object's mass by the mass of trapped water.
- Mass moments of inertia and centre of mass (CoM) values should also be revised to include the trapped water mass.
- You may also need to increase the object's volume to include the volume of trapped water, if it isn't already included (as it is essentially being treated as an internal component to the structure – think of a flooded buoyancy compartment). This affects both the hydrodynamic mass and the buoyancy of the object.

An alternative, which removes the need to revise the mass moments of inertia and CoM values, is to attach an additional "point mass" to the structure. This is particularly useful if the volume of trapped water is relatively small in comparison to the overall object. Rather than increasing the mass and volume of the object itself, you can instead add an additional 6D buoy with negligible properties (apart from the mass and volume) and connect it at the centre of mass of the trapped water. You are effectively modelling the trapped water as a separate object that is rigidly connected to the main object. The mass moment of inertia and CoM effects will arise naturally from the offset connection. Multiple such buoys can be used to represent distributed trapped water or multiple locations.

The above approach works well if the object is fully submerged, and remains so during the simulation. However if the object is lowered through the sea surface during the simulation, and therefore transitions from having no trapped water to having some, then this method is not appropriate as the additional mass and inertia would be applied all the time, regardless of whether the object was submerged or not.

Also, the above method will apply the additional body inertia in all three translational directions of motion, whereas it is possible to have different amounts of water trapped in different directions, for example an open ended structure.

So how do we add the effects of trapped water to an object that is being dynamically lowered through the sea surface, and that traps water in some directions of motion, but not all?

If your object is a cylinder, and can therefore be modelled using a hollow spar buoy, then the flooding of the hollow bore is handled for you:

- For motion in the axial direction, the hydrodynamic mass of the buoy is calculated based on the annular volume of the buoy.
- However, for motion in the normal direction, the hydrodynamic mass is based on the overall volume of the cylinder. This assumes that the fluid inside the bore is effectively part of the structure. See the help page [Theory | 6D buoy theory | Spar buoy and towed fish added mass and damping](#) in the section [Reference fluid mass and inertia](#) for further details.
- The hydrodynamic mass gets multiplied by  $C_a$  to determine the added mass, and  $C_m$  to determine the fluid inertia, from external pressure effects on the cylinder.
- Note that this isn't the same as increasing the body inertia to reflect the mass of the water 'contents', this is also automatically added for you separately.

So with a hollow spar buoy, the effects of the trapped water are correctly accounted for, including the fact that it is an open ended structure and therefore the increase in inertia will be different in different directions. OrcaFlex also correctly calculates when the trapped water is present (i.e. object submerged) and when it is not (object dry), as well as the transition from dry to wet (or wet to dry) as the object is lifted or lowered through the sea surface.

However if your object is *not* hollow cylinder shaped, and therefore the spar buoy is not an option for you, then you need to use one or more lumped buoys to model the object. In this case OrcaFlex cannot automatically model the trapped water effect because the lumped buoy object has no geometry. This means that some care is needed when assigning properties to the object(s) to ensure that the trapped water is correctly accounted for.

This is demonstrated in the model [F01 Trapped water.sim](#).

When you first open this model, you will see two objects (modelled on top of each other) that are being lowered from a vessel. One of them (red) is a hollow spar buoy. The other (pink) is a lumped buoy. The spar buoy has been modelled with a single cylinder in this example, to provide as close a comparison as possible with an equivalent lumped buoy. Usually however this would be modelled using a stack of cylinders to give a better, discretised model of the object. The lumped buoy has been drawn as a cylindrical hollow tube, but this is for visualisation only (lumped buoys have no physical geometry). Note that if your payload really was a hollow cylinder, then the spar buoy alone would suit your needs. The aim here is to demonstrate how to get a lumped buoy to match the behaviour of the spar buoy, in other words how to include all the effects that a hollow spar buoy is automatically including.

The spar buoy and the lumped buoy have been given identical properties (i.e. the properties of a steel tube, 10m long, Ø3m OD and Ø2.5m bore), and if you replay the simulation, you'll see that they behave identically when they are in the air, but not once they come into contact with the sea surface. This is the result of the spar buoy correctly modelling the added inertia effect of the water that floods into its bore, while the lumped buoy doesn't even have a bore to fill (remember it has no geometry!) therefore it cannot automatically include this effect.

What we can do however, is to increase the lumped buoy's [hydrodynamic mass](#) to reflect the increase in fluid inertia (added mass) due to the additional 'volume' of trapped water. In the model browser, check that the [view by groups](#) option is selected (right mouse click in the model browser), and then hide the [Lumped Buoy \(Original\)](#) group by clicking on it and pressing [Ctrl+H](#). Show the group [Lumped Buoy \(HydroInertia\)](#) in the same way.

Open the data form for the *Lumped Buoy2* object and look at the *properties* page – this buoy differs from the previous one only in the values used for the translational hydrodynamic mass. The *Lumped Buoy1* object has the hydrodynamic mass set to ‘~’ in all three directions. This means that the hydrodynamic mass is taken to mean the fully-submerged displaced mass (i.e. volume x water density). *Lumped Buoy2* has had the hydrodynamic mass increased in the x and y directions to account for the mass of trapped water in the structure (in other words the hydrodynamic mass in these directions is set to the volume of both the structure and the trapped water x water density). The hydrodynamic mass is then used to calculate the hydrodynamic inertia (i.e. added mass and Froude-Krylov forces).

Now when you replay the simulation you’ll see that this lumped buoy more closely matches the spar buoy’s motion, but it is still not the same. This is because we are still missing a bit of the inertia i.e. the *body* inertia from the trapped water.

If we were to increase the mass of the buoy to try to include this extra body inertia, we would increase it in all directions, all of the time (whether submerged or not), which would clearly not be correct. The best option is to add a separate dummy buoy, which has negligible properties for everything except for the hydrodynamic mass and Ca. In this case the hydrodynamic mass is set to be the trapped water mass (but only in the directions that the water is trapped, so x and y in this case) and Ca is set to 1.0. Cm must also be set to zero to avoid additional (Froude-Krylov) fluid inertia being included. This now includes the additional body inertia that results from the extra mass of the water being added to the structure in such a way as to allow different effects in different directions, and to only apply the additional inertia when the object is partially or fully submerged.

Open the *Trapped Water* buoy’s data form (located in the *Lumped Buoy (Body+HydroInertia)* group) to see the settings that have been given to this object.

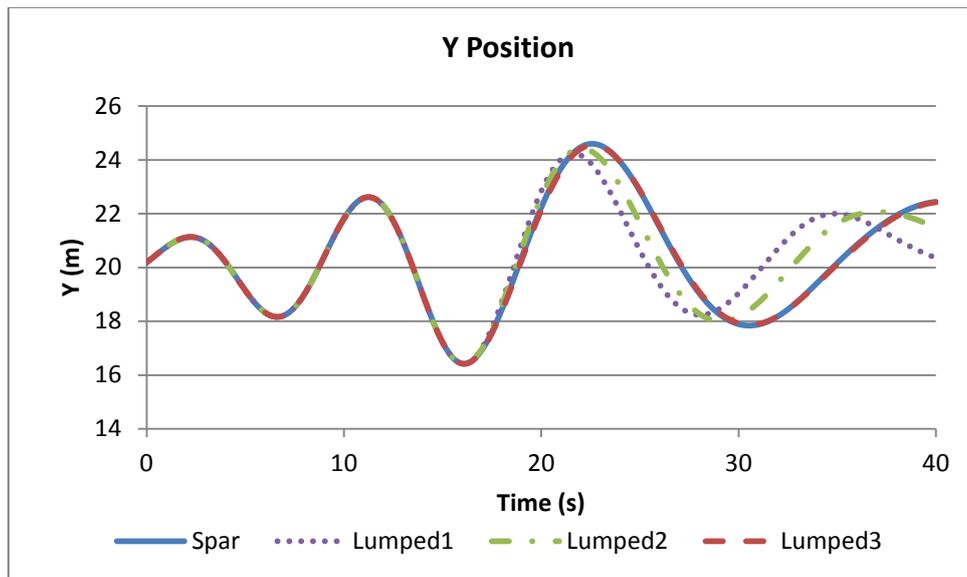
Hide the *Lumped Buoy (HydroInertia)* group and show the *Lumped Buoy (Body+HydroInertia)* group and run the replay. The motion of this lumped buoy closely matches the motion of the spar buoy, as they are both now modelling the effect of the trapped water fully.

## Results

Open the workspace *F01 Trapped water results.wrk*. This displays graphs of the payload’s Y position during the simulation and the crane wire tension. The payload results are presented for the *Hollow Spar Buoy* and the *Lumped Buoy3*. The results are clearly very similar, but will never be identical. The small differences between them come from how the change in proportion wet with submergence is calculated (linear for the lumped buoy, non-linear for the spar).

Further differences would appear if the wave and / or current direction were not perpendicular to the cylinders, due to slight differences in the drag formulation for the two different types of buoy.

The graph below shows the Y position of all four buoys plotted together, so that the differences in motion can be seen.



**Figure 5: Payload Y results**

In summary, if the object you are considering is a simple hollow cylinder then you should use a hollow spar buoy to model it, as all of the trapped water effects are automatically accounted for. If the flooding object is anything *other* than a hollow cylinder, then a more complex approach is needed, as demonstrated by the [Lumped Buoy3](#) method, to model the same effects. In this example we model a hollow cylinder in this way, simply to show how the same effects are achieved.