D04 J-tube pull-in

Introduction

This example shows how to model a typical J-tube pull-in operation. It includes pulling a line (a riser) out from a vessel over a chute whilst simultaneously pulling in the other end up a J-tube. It demonstrates the *line contact* model and introduces some tips for controlling the line position in the static solution.

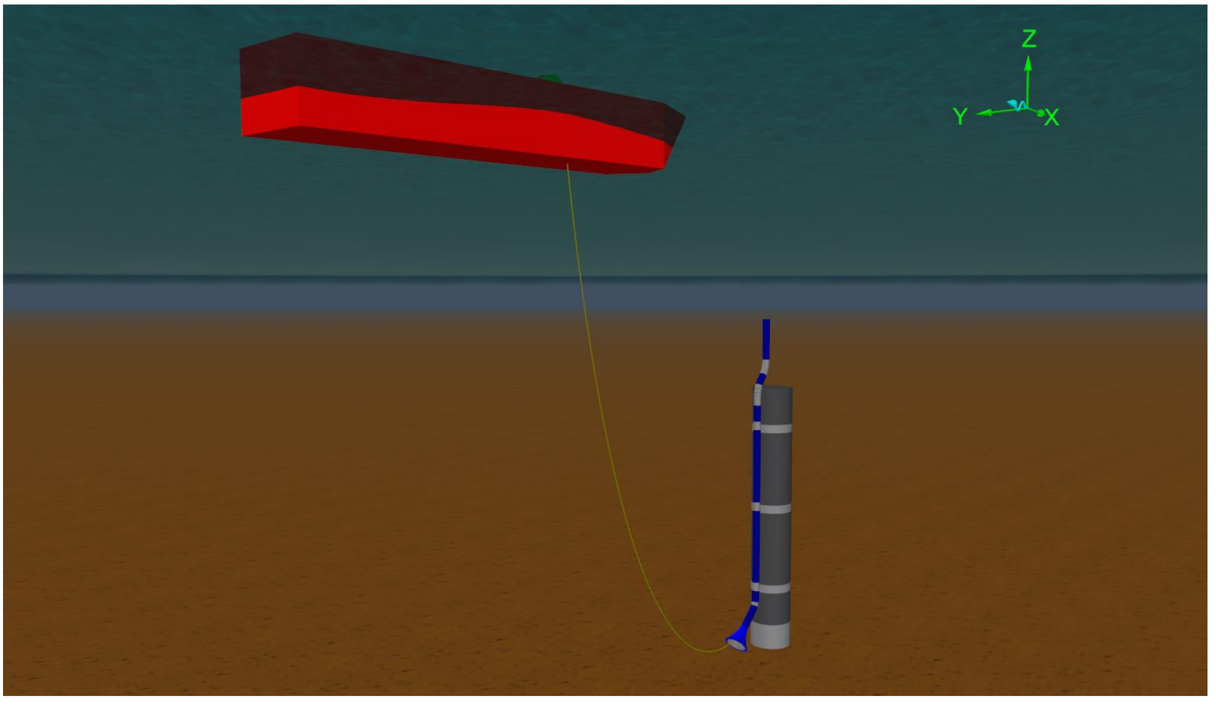
The *line feeding* feature is used at both ends of the riser to pay-out and haul-in the product, and friction is included in the contact between the riser and the J-tube wall. Note that objects have been modelled 'not to scale', for clarity.

The areas of interest in this type of analysis would most likely be the seabed clearance, possible jamming of the pull-head in the J-tube and tension in the riser. This model also captures flexing of the J-tube in response to the pull-in loads and the forces placed on the clamps holding it to the main structure.

When you open the simulation file the default workspace is opened. This automatically opens several views of the J-tube and vessel/chute arrangement. Replay the simulation (*Ctrl+R*) and you will see the riser and pull head being pulled up into the J-tube. Notice that certain parts of the riser and J-tube are highlight white, to indicate that contact has occurred.

The bottom right-hand view shows the line being deployed over the chute. Turn on the node axes (*Ctrl+Alt+Y*) to see the nodes being created at the line end on the vessel deck, and disappearing as the nodes are being hauled in at the top of the J tube. Press *Ctrl+Alt+Y* again to turn the node axes off.

In the *model browser*, right click and make sure that the *view by groups* option is ticked.

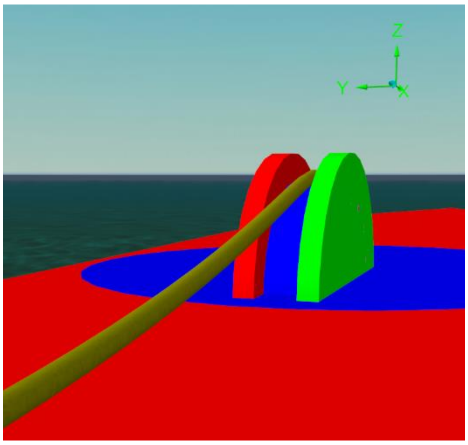


Building the model

Modelling the vessel infrastructure

Select the workspace *D04 J-Tube Pull In vessel.wrk* from the menu bar. This shows two views of the vessel and the top end of the riser passing over the chute.

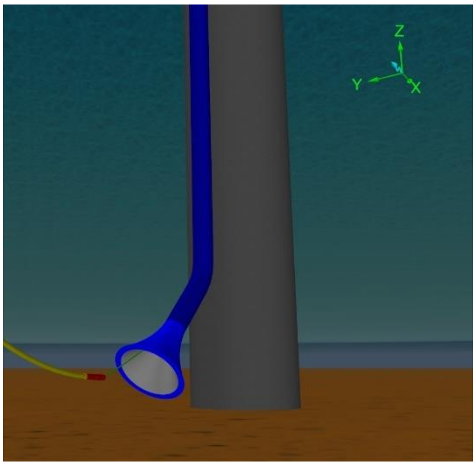
An *elastic solid shape* is used to model the deck surface; this is named *Deck* and is currently hidden in the model. This elastic solid isn't actually necessary because the line isn't in contact with it, but if you wanted to model a longer length of line, that was lying on the vessel deck, then this shape would provide the necessary contact surface.



The riser passes through a moonpool and will be shielded from the environmental loads in this region. This is modelled using a *trapped water* type shape – named *Moonpool Shielding* – which is connected to the vessel. This modifies the fluid motion within the shape's boundary to remove the environmental contribution.

The chute is modelled using *elastic solid shapes*, which form the contact surfaces for the riser to pass over. In this model we have also included some *supports* to help find the desired static solution. In the wire frame view, the supports are visible as four pink rollers, which are positioned so that they are inside the chute. Here, the support's contact surface is below the chute's contact surface so, when the static calculation starts, the supports do the initial work of picking up the riser and placing it on top of the supports. At this point the *Riser* line's nodes are in contact with the elastic solid shapes representing the chute surfaces, and therefore the node-to-shape contact takes over. In other words, the supports are only used to get the riser on top of the chute before the elastic solid shapes take over to model the chute's contact surfaces. This is a good solution because real-world chutes often have a smooth continuous surface.

The supports are part of the vessel object, meaning they are defined on the *supports* page of the *Vessel1* data form. Four *flat* type supports are used, and we have positioned them using the *simple geometry specification*. This allows for easy placement of the supports around a specified radius.



J-tube modelling

The J-tube itself is modelled as a line with pre-bend. Open the *J tube* line data form and look at the *pre-bend* page to see how this has been modelled. Note that bending is given relative to the nodal x and y directions; making the node axes visible (*Ctrl+Alt+Y*) can help to visualise this. Here, positive direction follows the right-hand rule.

Note that the pre-bent shape can be previewed using the *view pre-bend shape…* button. The top end of the line is built-in (it has infinite *end connection stiffness*) and therefore has its *end orientation* angles set appropriately (see the Modelling, data and results | Lines | Line data | End connections page of the OrcaFlex help for further details).

End B of the J-tube is left 'Free' so it is able to deflect in response to loads arising from the pull in operation.

Line contact modelling

The *line contact* model has been implemented here to allow the riser to pass through the bore of the J-tube. The J-tube is modelled as a line, so it is able to flex under loading as the riser is pulled through it. A second line contact relationship is used to model the clamps holding the J-tube to the structure (represented by the *Tower* line).

Open the *line contact data* form via the *model browser*. On the *relationships* page there are two relationships defined, one between the *Riser* and *J tube* lines and another between the *Tower* and *J tube* lines.

In the first relationship, the riser is the *penetrating line* and the J-tube is the *splined line*. The *inside* relationship type means that the *J tube* line is fitted with a smooth spline surface and the riser is fitted with penetrators that will contact it. The penetrators are positioned at the nodes of the *Riser* line.

Making the J-tube the splined line means that a spline tube is created which has the inner diameter of the line type as the bore diameter, and that has infinite wall thickness. This means that the static solution is found easily, as the section of the riser that is detected to be between the ends of the J-tube is pushed into the centre of the spline tube.

Select *D04 J-Tube pull in j-tube.wrk* from the menu bar. Reset the model (*F12*) and then run statics again (*F9*) and watch how the lines behave. Note that during *full statics* the lines do not interact with each other; the *line contact* model only comes into play during *whole system statics* and this can be seen clearly when the riser is suddenly pulled inside the J-tube.

The *containment enabled* option has been ticked for this relationship. This means that the inner line (the riser) will be shielded from the environmental fluid forces by the outer line (the J-tube) wherever the riser is positioned inside it. The contents of the J-tube line are set to *free flooding* therefore the external pressure and buoyancy force acting on the inner line are the same as for a normal (non-contained) line. An *axial friction coefficient* has also been assigned to apply axial friction between the riser and the J-tube.

It is also important to note the second relationship, which uses the *around* contact type. This means that penetrators on the *Tower* line are placed around the *J tube* line at the positions / offsets defined on the *penetrator locations* page.

For further details about modelling line contact, please refer to the Modelling, data and results | Lines | Line contact | Data and Modelling, data and results | Lines | Line contact | Modelling pages of the OrcaFlex help.

Modelling the bell mouth

To represent the bell mouth, an *elastic solid shape* has been connected to the bottom end of the J-tube. As the real-world bell mouth will have a relatively thin wall thickness, a *drawing* type shape has been used to show the bell mouth visually. A *drawing* type shape has no stiffness and therefore does not interact with other objects.

A second bell mouth shape (named *Bell mouth guide*) has been created with a much thicker wall to act as the structural boundary. This shape is an *elastic solid* type, therefore is has a stiffness and will interact with the line nodes that contact it. This object is hidden in the model.

Nodes are always pushed to the nearest contact surface of a shape. So, if line nodes are able to penetrate past the mid-point of a shape's thickness (which can happen if a shape is too thin), it increases the likelihood of the line nodes being pushed to the outside surface of a contact shape, rather than the inside surface. Making this boundary shape thicker than the physical structure reduces the likelihood of this happening.

Line feeding

Open the *Riser* line data form and look at the *feeding* page: both ends of the line have been given a non-zero payout rate. End A has a negative *payout rate*, meaning the line will be hauled in. Conversely, the positive *payout rate* at end B means that the line will be paid out.

Both ends have *apply ramp* selected, which means that the payout will be ramped up during the simulation build-up (stage 0). This helps to avoid discontinuities between statics and dynamics. An alternative would be to set up a variable payout rate, which can include a gradual transition between payout rates.

Note that the *initial arc length* of end B is 255m. The total riser line length is defined on the *structure* page as 297m: 88m of wire, a 2m long pull head and 207m of riser. So, by setting the initial arc length to 255m, there is 42m of "inactive" line.

For further details about line feeding, and the associated settings, please refer to the Modelling, Data and Results | Lines | Line Data | Feeding of the OrcaFlex help.

Running statics

When running statics, the solver needs to find a solution that places the *Riser* on top of the vessel deck and over the chute, as well as being inside the J-tube. Earlier it was mentioned that some supports have been used to lift the line up onto the vessel, however we can make the calculation even easier by using the *spline* option as the *step 1 statics method*. This can be used to give the line a much better starting position than the default *catenary* method.

If you reset the model again (*F12*) you will see the spline visible as a white line. The spline's shape can be configured by manually moving the control points on the screen (represented by the white-coloured crosshairs) using your mouse. Alternatively, it is possible to specify the global positions of the control points on the *spline starting shape* page of the *Riser* line data form.

OrcaFlex needs to know some information about the 'sense' of the *Riser* line, because some of the calculations need to know which end of the line (end A or end B) is at the top, and which end is at the bottom. For example, if a line contacts the seabed, the touchdown point results are determined by starting at the top end and moving towards the bottom end until the first node in contact with the seabed is found. In this example, both ends of the line are lifted up, so there would be two touchdown points if the line were to contact the seabed.

To indicate that we want the touchdown point to be reported as the point closest to end B, we need to set the *top end* data item on the line data form to *end B* (rather than the default setting of *end A*). Full details of the calculations that use this data can be found in the help file section Modelling, data and results | Lines | Line data.

The model also needs some damping applied during *whole system statics* to assist with finding the static solution. This is applied on the *statics* page of the *general* data form.

Results

Re-open the simulation file and select *D04 J-tube pull in results.wrk*.

A *range graph* for the tension in the *Riser* line is displayed (bottom right). Notice how range graph results are not available for the parts of the line that are inactive in the model for the entire duration of the specified results period (in this case that's end A to arc length 29.5m and 253.5m through to end B).

The middle right-hand graph shows the *time history* results for *effective tension* at end A of the *Riser* line. The tension results get a little noisier once the rigid pull head enters the J tube. In some cases, using the *around* line contact relationship instead of the *inside* relationship will give a smoother pull-in, and therefore a smoother tension plot. That's because with the *around* relationship you are pulling the smooth spline surface over the penetrator ring at the mouth of the J-tube, rather than the other way round. Sensitivity checks have shown that both relationships lead to similar results so, in this case, the *inside* option has been used because it's slightly more robust in statics.

The result shown in the top-right window display the instantaneous *effective tension* at end A of the *Riser* line. If you run the simulation replay, notice that the result changes with time i.e. it shows the instantaneous value, as expected.

Finally, the bottom left-hand graph shows the *line contact force* for the *J Tube* line. This provides some visibility on the force exerted on the clamps which hold the J-tube to the structure.