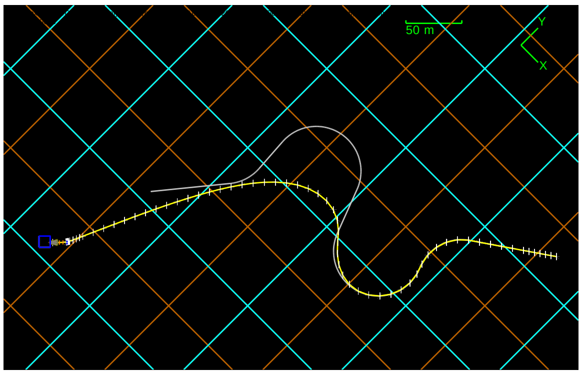
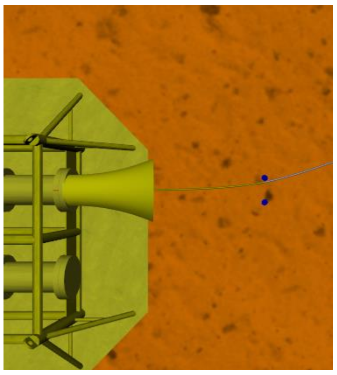
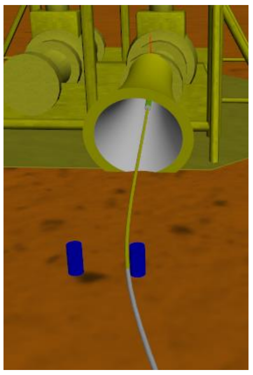
D02 Pull in analysis







Introduction

For this example, a line has been laid on the seabed along a specified route. It is to be pulled into a wellhead funnel by a winch. A gateway ahead of the funnel controls the pipe entry so it approaches at an angle that will not cause the pull head to jam.

On opening the simulation file, the default view shows a shaded view of the wellhead and a wireframe plan view of the lay route.

Building the model

Pipe modelling

Look at the *Pipe* line data form. Here, the line is laid along a prescribed path, enabled by the *prescribed* method for step 1 of line statics.

The prescribed path has been set up on the *prescribed starting shape* page. The track can be seen in the wire frame view, represented by the grey line, and takes a similar form to a child's railway track: lengths of straight and curved pieces are specified, each beginning where the last ended. You can input these pieces by specifying the *length* and *turn*, or the location (*X*,*Y*) that each point ends. OrcaFlex will then automatically determine the other settings. Note the *radius* of each curved piece is constant and the initial direction of the first piece also needs to be specified (it is common for this to match the *azimuth* angle at end A).

Further information about setting this data can be found on the Modelling, data and results | Lines | Line data | Prescribed starting shape page of the OrcaFlex help.

Generally, the *prescribed* shape option is intended for lines that have *all* their length in contact with the seabed in statics. OrcaFlex will then lay the line out on the seabed, with each nodal position determined from the distance along the track. Note that OrcaFlex also accounts for any axial strain from the *as laid tension*.

Note, in a situation where end A of the line is off the seabed, OrcaFlex would specify the nodal X and Y coordinates first, and then add Z. The line height above the seabed would then vary linearly from the raised end A down to touchdown. This would not represent a realistic situation, as the line ought to hang in a catenary shape to touchdown.

It is also important for the line segmentation to be refined enough that it can follow any curves in the track. For example, a segment length of 5 m will not be able to follow a bend radius of 1 m.

Running statics

During statics, OrcaFlex will examine the out-of-balance forces for the positions of the line nodes in the model, before trying a new set of node positions that will give smaller out-of-balance forces. This process continues, allowing OrcaFlex to gradually iterate towards a static equilibrium for the line.

In this model, we know about the line's as-laid position, meaning we do not want OrcaFlex to shift the line nodes far when finding equilibrium positions for them. This is accomplished by using strongly restrained *step 2* statics. Here, the *full statics convergence control method* has been set to *mag of std. error / change* on the *statics convergence* page. The *mag of std. change* data item at the end of the table has been reduced from the default value of 0.2, to 0.01. Reducing this number restricts the solver to making very small changes in the line node positions at each static iteration. Additionally, some extra *min* & *max damping* and an increased number of *max iterations* assist the solver in reaching a solution.

This method of solving statics, where a line is forced to remain close to a known equilibrium position, is quite widely applicable. For example, it could also be used for lines passing over chutes or through guides, where the line must stay close to a spline fit starting position.

Gateway modelling

The gateway is modelled as two lines of the required diameter and the *line clashing* model has been enabled to allow contact to take place between the pipe and the gateway itself. This models segment-to-segment contact between lines.

We have chosen to use line clashing here, rather than line-to-shape contact, because contact between a line and a shape takes place at the *nodes* of the line only (not the segments). This would demand very fine segmentation for the *Pipe* line, resulting in a bigger matrix to solve and hence increased calculation time.

To activate the line clashing routine, the *clash check* box needs to be ticked on the *structure* page of the respective data forms for the *Pipe*, *Gateway 1* and *Gateway 2* lines. Note that activating the *clash check* option will result in slower simulation run time, so it should only be applied where required. For the *Pipe* line, it has only been selected for the first 30 m, which is the length expected to contact the gateway during the pull-in.

To allow line clashing to be registered, the corresponding line types have been given *line clashing stiffness* values (see the *contact* page of the *line type* data form). Accurate clashing stiffness values are rarely known for pipes but, in this case, we only require that the gateway effectively constrains the pipe, meaning an arbitrary value can be applied. The Theory | Line theory | Clashing page of the OrcaFlex help provides further details about interpretation of the clash forces.

It is also important to remember that clashing between lines is not active during the static calculation. This is not an issue in this example because contact occurs during the dynamic stage only.

Note, to provide better visualisation in the shaded graphics view, the two gate lines are shown with flat ends rather than hemispherical. This setting can be controlled by unticking the *draw as spheres (shaded)* option, on the *drawing* page of the line data form. Note that the end nodes are still modelled as spheres, meaning this change only affects visual representation.

Modelling the pull in

The analysis is split into multiple stages to allow the winch to perform different operations. The table below shows the actions carried out per stage. Note that winch pull-in is denoted by negative payout (the standard OrcaFlex convention), 'Relax' means the winch length varies to keep the tension at 0kN and 'Hold' means that the winch length does not change during that stage.

It is important that winch lengths in OrcaFlex never reach zero (or close to zero) as this causes convergence problems. However, it is convenient to pull-in to a connection point on the winch as this exactly defines the final pulled-in position and makes the calculation of the required pull-in length more straightforward.

In this example, each winch has three connection points. The first and second connection points are both anchored to the seabed with a (vertical) distance of 2m between them. If we pull in to the second connection point, we can easily calculate the pull-in required (total winch length at the start of the pull-in minus 2m) and avoid the problems associated with zero length winches. For the *To Gate* winch, this starting length is simply the static length of the winch with zero tension applied.

The *To Gate* winch is attached to end A of the *Pipe*, on the seabed in the middle of the gateway, and 2m vertically above this point. During *stage 1*, this winch pulls the pipe to the gateway before being released. The winch *To Hub* is attached to end A of the pipe, the far end of the funnel on the seabed, and 2m vertically above this point. During *stage 2* this winch pulls the pipe from the gateway up into the funnel.

The pull-in model requires two winches because winch objects are unable to interact with other structures. If only one winch was used, it would pull the line straight into the funnel, missing the gateway altogether.

At the end of the pull-in, the system is allowed to settle for a few seconds during stage 3.

Stage	Time	Winch payout		Pipe end A
		'To Gate'	'To Hub'	
Statics	–	Zero Tension	Zero Tension	Anchored
0	-1 s to 0 s	Hold	Relax	Settle
1	0 s to 46 s	-87.84 m	Relax	To Gateway
2	46 s to 86 s	Release	-15 m	Into Funnel
3	86 s to 91 s	–	Hold	Settle

Table 1: Winch payout settings

Model object positioning

When positioning objects in the OrcaFlex model, it is sometimes useful to enter the data in field coordinate values, which means using numbers with many significant figures. This can sometimes cause calculation problems because computers only have a limited number of significant figures available. If you fill these up with unimportant information, there are less available for calculation accuracy.

For example, consider the actual field coordinate values below. Suppose the computer allowed eight significant figures, then the actual positions would be rounded to one decimal place, as per the 'Field' figures below:

Actual	X1	6500500.6500500m	X2	6500500.6599504m
Field	X1	6500500.7m	X2	6500500.7m
Local	X1	0.6500500m	X2	0.6599504m

The actual difference between position X1 and X2 is 0.0099004m but the rounded 'Field' values will give a difference of 0.0m. If the system were moved to the model origin, then the 'Local' values would be used, and this would calculate the difference between the two positions correctly.

This is important when it comes to accurately calculating results for particularly stiff system, and for achieving the solution tolerance. Imagine you needed to make a change of 0.01m to satisfy a tolerance for a static solve. With field coordinates above, that small a change is not possible so the static or implicit search will fail.

In this situation, you can of course enter your data using field coordinates and then move the entire system closer to the global origin (using the *move selected objects* command in the *model browser*) before running the analysis so that you have maximum accuracy available for results. The User interface | Model browser | Move selected objects wizard page of the OrcaFlex help provides further details about moving objects in this way.

Results

Run the animation replay to watch the whole simulation. Note that the shaded graphics view moves with the pull head rather than staying fixed relative to global. This is because the *view parameters* have been set to *relative to* a dummy object called *pullhead camera.*

The view settings can be accessed using the keyboard shortcut *Ctrl+W* or by clicking on the eyeball and pencil icon in the top left-hand corner of the view window. Here, the view may be fixed to any buoy, vessel, shape or constraint. This can be useful if you wish to fix the view on something. In this instance we have connected a *drawing* type shape to the line and used that to control the view. Note that we have positioned it away from the pull head (at an arclength of 35m) so that the view doesn't follow the line end up into the bellmouth.

Try both wire frame and shaded graphics views (you can toggle between them using *Ctrl+G*) and watch the line being pulled first to the gateway.

Then load the results workspace *D02 Pull in analysis results.wrk*. This shows close-up views of the funnel, along with some tension time histories for the winches. Run the replay again and watch the line curve around one of the gateway lines.

Both graphs show winch tension overshoots and oscillations. These are a consequence of speeding up the operation to save processing time. The actual pull-in operation would be expected to take more than an hour to complete, compared to just over a minute here. Slowing the rate of winch pull-in will reduce the overshoots and oscillation amplitudes, but at the cost of longer run time.