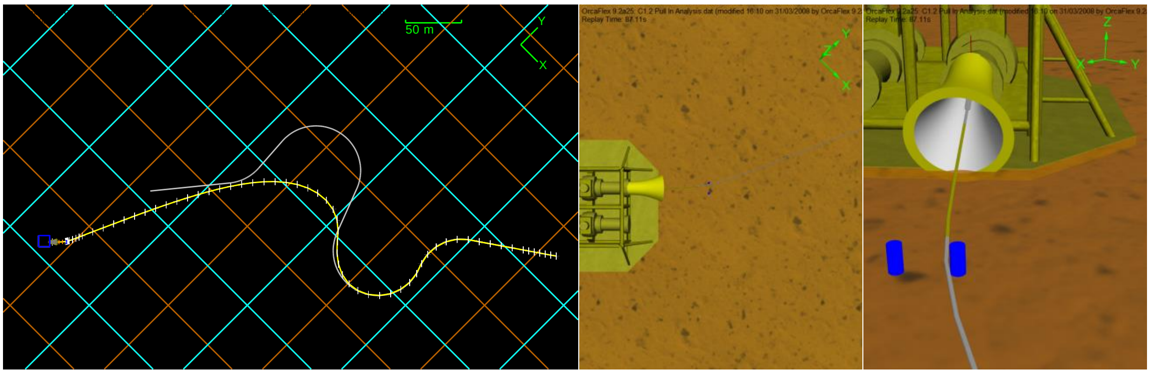
D02 Pull in analysis



Introduction

A line has been laid on the seabed in a specified route. It is to be pulled in to 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

The line is laid in a prescribed path, represented by the grey line in the wireframe view.

Look at the *prescribed starting shape* page of the Pipe's data form. The track is made up in a similar manner to a child's railway track. Lengths of straight and curved pieces are specified, each beginning where the last ended. Note the radius of each curved piece is constant.

You can input these pieces by specifying the length and radius or the location that each point ends. OrcaFlex will determine the other set from what you input. The initial direction of the first piece also needs to be specified, though it is usual to have it match the *line end azimuth* angle at end A.

During statics, OrcaFlex will lay the line out along the track. Each nodal position is specified from the distance along the track. Note that OrcaFlex also allows for any axial strain from the *as laid tension*.

In this model we know the as-laid position of the line and so we do not want OrcaFlex to shift far to find a static equilibrium position. This is accomplished by using strongly restrained *step 2* statics. The statics method has been set on the *full statics convergence* page to *mag of std. error / change*. The *mag of std. change* data item at the end of the table has been reduced from 0.2 to 0.01. The effect of this parameter is to restrict the line to making only very small motions during a single static iteration. Finally some extra damping and more iterations are added to assist in settling the system.

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

The gateway is modelled as two lines of the required diameter. They and the pipe are given contact stiffness values and the *clash check* option found on the *structure* page of the line data forms is ticked (see below for more on this). Accurate contact stiffness values are rarely known for pipes: in this case, all we require is that the gateway effectively constrains the pipe, so an arbitrary value can be applied. Look at *Theory | Line theory | Clashing* in the help file for interpretation of the clash forces.

To turn the clashing routine on, the *clash check* box needs to be ticked for the *Pipe*, *Gateway 1* and *Gateway 2* lines. Note that activating the *clash check* will result in slower runs so it should only be applied where required. For *Pipe* it has only been selected for the first 30 m of the pipe; the length expected to contact the gateway during the pull-in.

Remember also 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.

To help visualisation in the wire frame view, the two gate lines are shown with flat ends rather than hemispherical. This setting is found on the *drawing* page of the line data form. Note that for contact purposes the end nodes are still modelled as spheres, the change only affects the visual representation.

We have chosen to use line on line clashing as the contact method, rather than line to shape contact, because contact between a line and a shape takes place at the nodes of the line only, which would demand very fine segmentation of the Pipe line. This would result in a bigger matrix to solve and hence increased calculation time.

The analysis is split into multiple stages to allow the winch to perform different operations in each stage. The following table shows the actions carried out by stage:

Note that winch pull-in is denoted by negative payout – the standard OrcaFlex convention.

Stage	Time	Winch Payout		Pipe End A
		'To Gate'	'To Hub'	
Statics	n/a	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

Note that 'Relax' means the winch length varies to keep the tension at 0 kN. ‘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 pull-in length required 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 2 m 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 2 m) 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 winch *To Gate* is attached to end A of the pipe, on the seabed in the middle of the gateway, and 2 m vertically above this point. In *stage 1*, this winch pulls the pipe to the gateway; it is then released. The winch *To Hub* is attached to end A of the pipe, the far end of the funnel on the seabed, and 2 m vertically above this point. In *stage 2* this winch pulls the pipe from the gateway up into the funnel.

The pull-in model requires two winches because there is no interaction between winch wires and other structures. If only one winch was used, it would pull the line straight into the funnel, missing the gateway.

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

The prescribed shape option is intended for lines that have *all* their length in contact with the seabed. If the line has end A off the seabed then OrcaFlex will specify the nodal X and Y coordinates first, and then add Z. The line height above seabed varies linearly from the raised end A down to touchdown. Note that this does not represent a realistic situation, as the line ought to hang in a catenary shape to touchdown.

Also the line segmentation needs to be refined enough that it can follow any curves in the track. For example a segmentation length of 5 m will not be able to follow a bend radius of 1 m.

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 a computer only has a limited number of significant figures available and if you fill these up with unimportant information, then there is less available for accuracy.

For example, consider the actual field coordinate values below. Suppose the computer allowed nine significant numbers; 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 shown would be used, and this would calculate the difference between the two positions correctly.

This accuracy is important in stiff system results and achieving the solution tolerance. Imagine you needed to make a change of 0.01m to satisfy the tolerance. 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.

Results

Run the animation replay through 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* for this view have been set to *relative to* an object called *pullhead camera.* To see the view settings click on the eyeball and pencil icon:

Any buoy, vessel or shape can be used to fix the view to, so in this instance we have added 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 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 then using Ctrl+G). Note that if you simply press Ctrl+G from the default wireframe, you must then zoom in to drop below the sea surface. 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 and 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 probably take an hour or more 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.