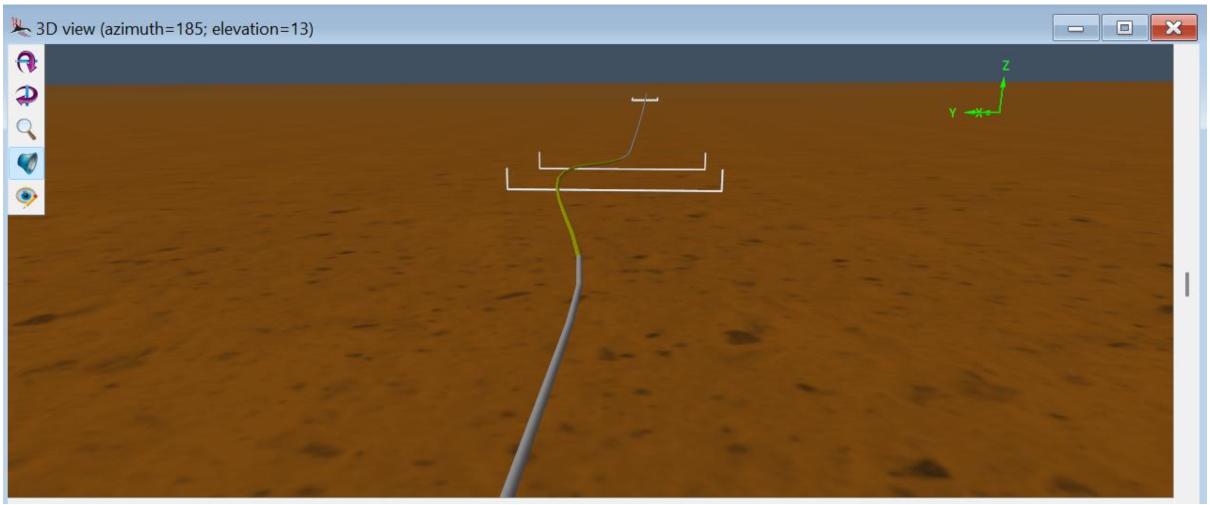
M01 Pipeline lateral buckling & walking

Introduction

This example includes two related models for the analysis of pipelines in place on the seabed. Specifically, the analysis of lateral buckling and walking that can result from changes in the pipeline contents temperature, density and/or pressure during operation.

This example does not demonstrate line cover features (buried lines), but there is a set of line cover examples embedded in the OrcaFlex help page: Theory | Line theory | Buried line examples.



Building the models

The models are of a section of subsea steel pipeline that extends along a gently sloping seabed in deep water. A slug of hot liquid travels along the pipeline until the whole pipeline is filled and this is followed by a region of gas at ambient temperature that also gradually fills the entire pipeline length. This simple pattern is repeated to check the effect of repeated heating and cooling cycles.

For the pipeline lateral buckling analysis, a short section of pipeline (1 km long) laid across two pairs of buckle initiator sleepers, is modelled for a single warm-up cycle to check the line curvature induced by the thermal expansion of the pipeline.

For the pipeline walking analysis, a 3 km pipeline with six pairs of buckle initiator sleepers is considered and the simulation is continued for two full warm-up and cool-down cycles.

Both models assume no wave action due to the deep water, but a small current is included. The current uses a *power law* vertical variation method to model an exponential reduction in current speed with depth between the still water level and the seabed.

Pipeline lateral buckling model

The model data can be viewed by opening the file *M01 Pipeline lateral buckling.sim*.

Line setup

The pipeline is anchored in place at both ends to check the maximum lateral buckle – in reality some axial movement of the ends would likely be possible, which would tend to reduce the temperature-induced buckling.

To set up the end positions initially, end A was anchored at [0, 0] with the *height above seabed* set based on the calculated seabed penetration. End B was left free and the line was laid out at a *lay azimuth* of 0 degrees with an *as laid tension* of 0 kN. The position of end B was then set to the calculated position in statics complete state (using the *use calculated positions* option under the *model* menu) and the end B connection was changed to *anchored*, which fixes end B at the calculated position.

The line type data form was set up with the line type inner and outer diameters based on a 10 inch NPS 250 schedule 80 steel pipe; and the other material, hydrodynamic and frictional properties all using the automatically populated values for a *homogeneous pipe* category line type. On the *coatings & linings* page, a non-structural coating of thickness 0.03m and density 0.825 te/m^3 is applied to the line type representing a Glass Syntactic Polyurethane (GSPU) thermal insulation layer.

A sensitivity study was carried out to find a suitable line segment length for this model. Segment lengths between 1m and 10m were tested. No significant change in the resulting post-buckle curvature was evident for segment lengths of 1m, 2m or 2.5m. With 5m segment lengths, the curvature was still well represented, but the resulting shape seemed slightly less smooth visually. Given the short simulation run time with the default time step (0.1 seconds), a segment length of 2.5m was chosen. Further information about the importance of line segmentation sensitivity studies can be found in the introduction to OrcaFlex training video 5: Lines.

Sleeper setup

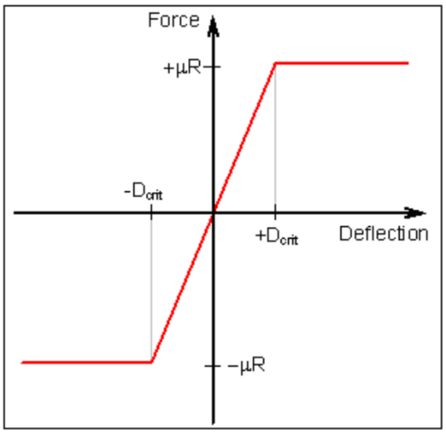
The pipeline is laid over two pairs of buckle-initiator sleepers. The sleepers are placed on the seabed every 500m starting from 250m from end A. These are modelled as a pair of U-shaped supports, 20m apart, at right angles to the pipeline. Each sleeper is 20m long with a diameter of 0.2m and includes a 2m high end post at each end. The sleepers are raised 1m above the seabed and are connected to fixed *6D lumped buoys* with negligible properties. A friction coefficient of 0.1 is defined for the line / sleeper interaction on the friction coefficients data form.

Line / seabed interaction

The seabed is modelled as a smooth slope from 1000m deep at the origin to 900m deep at X = 2800m, using a *profile* type seabed shape. The upper part of the seabed slope uses the OrcaFlex default seabed properties as specified on the *environment* data form, but in the lower part of the slope an area of softer seabed is included using data on the seabed page of the line data form.

When the *seabed friction policy* is set to *as laid*, a *friction target position* for each line node is set out along the seabed profile in the *lay azimuth* direction from the designated bottom end (which is end A in this case), with the node spacing determined from the unstretched segment length and the *as laid tension*. These friction target positions are used in the static analysis and are carried across to dynamics, where the positions are updated when the node position moves as described here: target positions in dynamics.

Solid contact friction in OrcaFlex is modelled as a modified Coulomb friction force as explained on the OrcaFlex help page [Theory | Friction theory](#).



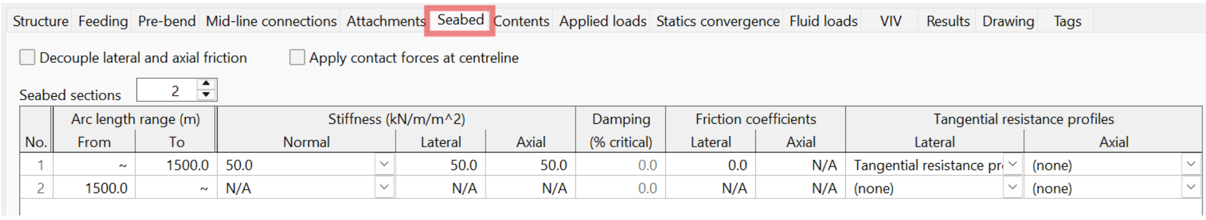
The standard Coulomb model includes a step change in the friction force when the node displacement changes direction. Mathematical models cannot handle step changes well, so OrcaFlex modifies the Coulomb friction by ramping the frictional force to its full value over a short distance D_{crit}. Once this initial ramping distance is exceeded, the friction force remains constant as the displacement increases further.

The shear stiffness k_s and contact area a are used to determine the size of D_{crit} in the relationship:

$$D_{crit} = \frac{\mu R}{k_s a}$$

where μ is the friction coefficient and R is the contact reaction force.

In addition to the seabed properties set on the *environment* data form, this model also accounts for seabed properties that change along the length of the pipeline to model an area of softer seabed in the lower part of the seabed slope. On the *seabed* page of the line data form, the seabed stiffness and frictional properties change with line arc length. Any *stiffness* data specified here will override the seabed stiffness data on the *environment* data form, and any friction coefficients override those specified on the *line type* data form unless the special value 'N/A' is used for the data item.

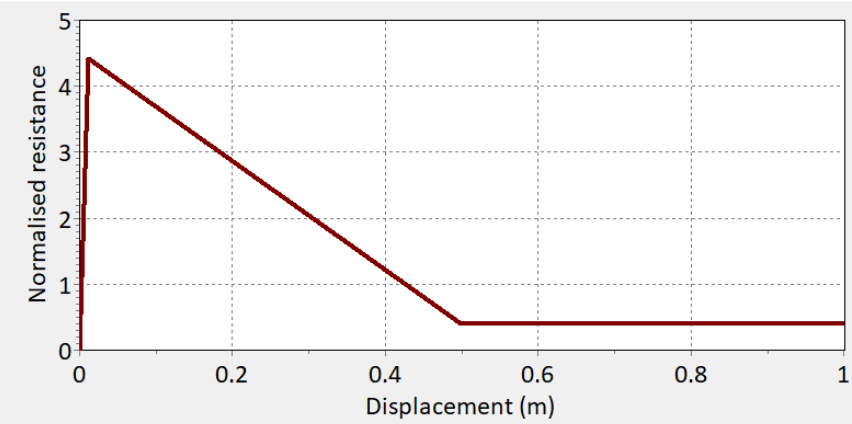


	Arc length range (m)		Stiffness (kN/m/m^2)			Damping	Friction coefficients		Tangential resistance profiles	
No.	From	To	Normal	Lateral	Axial	(% critical)	Lateral	Axial	Lateral	Axial
1	~	1500.0	50.0	50.0	50.0	0.0	0.0	N/A	Tangential resistance pr	(none)
2	1500.0	~	N/A	N/A	N/A	0.0	N/A	N/A	(none)	(none)

A lateral *tangential resistance profile* is defined for the pipeline to model lateral breakout resistance from the pipeline embedment in the softer part of the seabed. Lateral breakout takes place perpendicular to the axis of the line (in the plane of the seabed) and occurs when the lateral forces on the line are enough to overcome the initial embedment. After the line has escaped the initial embedment, it continues to push soil ahead of it, which is often modelled as a small, but constant residual resistance (in this case of similar magnitude to the seabed friction).

Note that the lateral friction coefficient is set to zero for the lower part of the line so that the only lateral resistance is from the tangential resistance profile. If the friction is not set to zero then the lateral tangential resistance will work in addition to the friction. No axial tangential resistance profile is defined in this example as the axial seabed friction is considered to be adequate to model the expected resistance to movement in the line axial direction.

Breakout resistance can be approximated by a characteristic trilinear resistance profile. Further details on seabed tangential resistance modelling can be found in the OrcaFlex help here: [Theory | Line theory | Seabed tangential resistance](#). The trilinear resistance profile used in this example is shown below. Note that the values used have been normalised by the magnitude of the *seabed normal resistance* (which is available as a [line result)](#) for use in OrcaFlex.



Tabular line contents

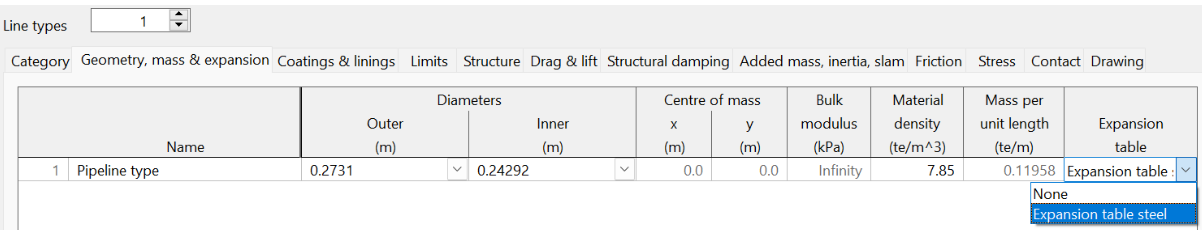
This example models a slug of hot, dense fluid travelling at constant velocity along the pipeline and so would seem ideally suited to the *slug flow* line contents method. However, we are particularly interested in the effect of contents temperature on the pipeline expansion. The slug flow method only allows for constant temperature, so we have instead used the *tabular* method for the pipeline contents. The OrcaFlex help describes the options for line contents here: Modelling, data and results | Lines | Line data | Contents.

Tabular line contents offers more flexibility than slug flow as you can define the line contents as a general function of time and/or arc length. This method works well with the output of a flow assurance package to define the contents properties table, but you can also build a contents table manually as we have in this simplified example. The table itself can be specified internally on the line data form, or through an external file. Using an external file can be advantageous for large tables because of the volume of data. In this case we chose to use an external file named *contents.txt* despite the simplicity and small size of the text file, as this means that the same external file can be used for both models in this example.

The layout of the external file must match the layout of the internal table, with all 7 columns appearing in the same order as they do in the table. Columns can be separated by spaces or tabs, with each line in the file representing a row of data. You can use the *preview* button on the data form to verify that the data is read as intended.

It is worthwhile carefully reading through the OrcaFlex help on how the tabular contents is interpolated with time and arc length. Watching a replay of an *instantaneous value* period *range graph* result (see User interface | Results | Producing results) for the interpolated property is a good way to check that the interpolation has worked as intended. In this example, we assume no temperature drop or pressure loss along the pipeline for simplicity.

The thermal expansion itself is driven by an *expansion table* that has been defined for the pipeline line type, on the *geometry, mass & expansion* page. This governs how the line type material expands and contracts due to changes in contents temperature and contents pressure.



A linear thermal expansion coefficient of 10.6e-6 m/(m°C) appropriate for mild steel has been implemented. This represents the change in length per degree of temperature change per metre of pipeline length. The variable data item *expansion table steel* is defined with a single independent variable, the temperature of the line contents; but the expansion table can also optionally include expansion with contents pressure.

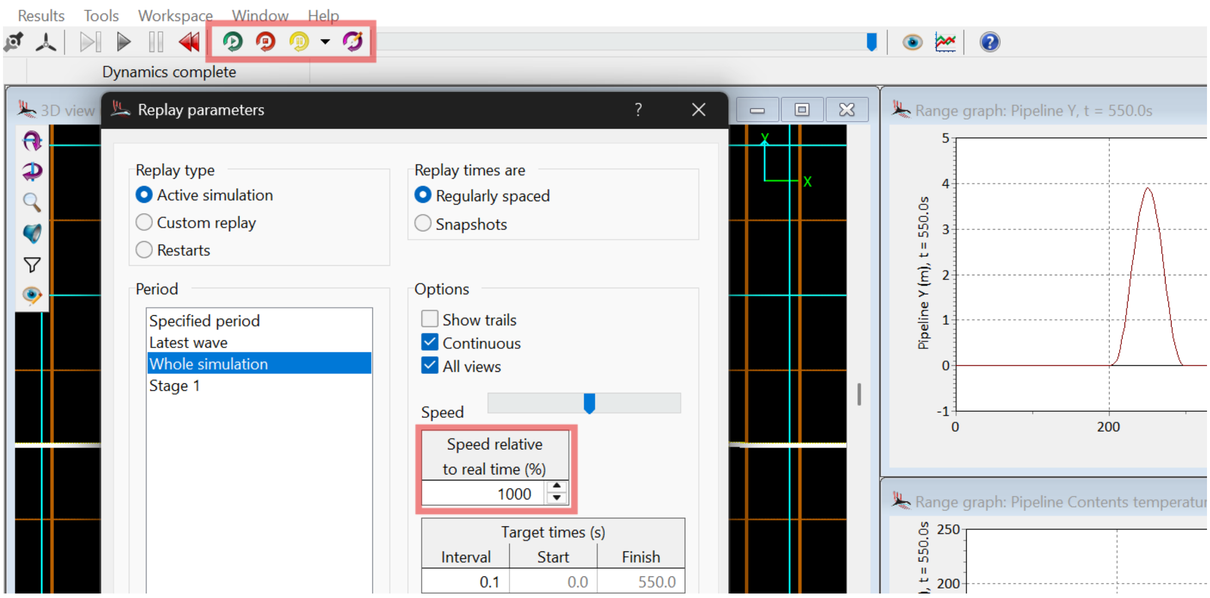
Note that thermal expansion is instantaneous with contents temperature and takes no account of the temperature of the surrounding fluid or thermal inertia in the pipeline. Note also that the expansion table is truncated, not extrapolated, outside the user defined grid of data.

Pipeline expansion due to the difference between internal and external pressure is included in the line structural model through the difference between wall tension and effective tension and through the Poisson ratio effect as detailed on the OrcaFlex help page [Theory | Line theory | Calculation stage 1 tension forces](). This means that it may not be necessary to include the contents pressure as an independent variable in the expansion table, as the relevant effects may be modelled already. For this simplified example, the contents pressure is assumed to remain constant.

Results

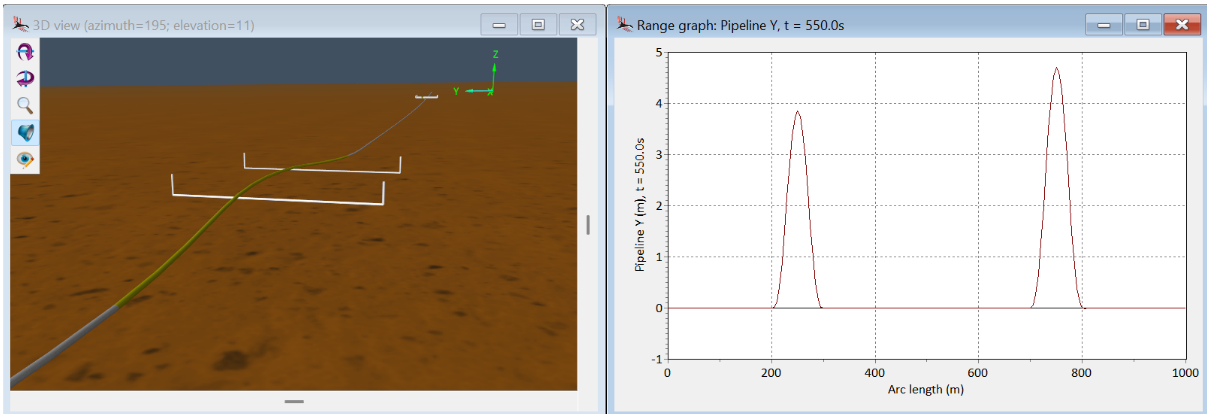
Some results of interest can be opened by loading the workspace file *M01 Pipeline lateral buckling results.wrk* from the *workspace* menu.

It might be of interest to watch a replay of the simulation to see the buckle developing as the slug of hot fluid travels along the pipeline. As the simulation duration is long, it helps to adjust the replay settings – select *edit replay parameters* under the *replay* menu (or use the toolbar icon). A *speed relative to real time* of 1000% is good choice for this particular simulation.



The results shown are [range graph]() results, so they show the results along a specified length of pipeline (in this case the full length). The *period* is also important for range graphs, for example, *instantaneous value* period range graphs show results at the replay time (so update dynamically in a replay), whereas *whole simulation* period range graphs show maximum, minimum and mean values over the whole simulation.

The pipeline buckles at the buckle initiator sleepers in response to the increase in line contents temperature. The pipeline initially bends upwards before falling/twisting to one side to give a characteristic buckle shape.



The side that the pipeline buckle falls towards can be hard to predict and small variations in the initial conditions can affect this. In this instance, the small current tends to push the buckle over to one side as the current is perpendicular to the line, but the buckle direction can be harder to predict when there is no current or when there is significant wave action present. Changing the segment length or the time step can also affect the buckle shape.

Pipeline walking dynamic model

Model setup

To see significant walking behaviour, a longer section of pipeline is helpful. The model setup for the pipeline walking model *M01 Pipeline walking.sim* is similar to the lateral buckling model except that a longer length of pipeline (3 km) is modelled with six pairs of buckle initiator sleepers rather than just two. End B of the pipeline is connected to *free* so that the line end is free to move in response to the thermal expansion. The simulation is allowed to run for longer to model two full warm-up and cool-down cycles.

The number of line segments in the model is important for the model run time, so a segment length sensitivity study was carried out for this model also. A longer line segment length (5m) is used for this version of the model to keep the run time and file size down – as mentioned above, 5m long segments still give a good representation of the buckle shape.

The *time step* was increased to 0.2 seconds to further reduce the model run time. The *implicit solver iteration count* (which is a *time history* result under the *general* object) remains low at this time step. Readers new to OrcaFlex dynamic analysis can find further information on setting an appropriate time step and monitoring the iteration count in our introduction to OrcaFlex video series, in particular video 6 on dynamic analysis.

One possible limiting factor on segment length for pipeline analysis may be the *Euler limit*, which if exceeded means the line segments may be too long to accurately represent the post-buckle behaviour. For this particular model, segment lengths would need to be over 30m to raise Euler limit warnings, but this will not be the case for all models. For a discussion of Euler limits and modelling compression see: Modelling, data and results | Lines | Modelling compression in flexibles. There is also a tutorial video on Euler limit warnings on our website.

Results

The results of the simulation can be viewed by opening the workspace file *M01 Pipeline walking results.wrk* from the *workspace* menu. A time history plot of the global *X position* of the free end of the pipeline (end B) shows that it experiences a small residual offset in the global X direction as a result of the warming and cooling cycles.

Watching a replay of the simulation with the workspace file open shows how the effective tension in the pipeline reduces (increasing compression) as the contents' temperature increases. A *speed relative to real time* of 5000% works well for this replay. The compression is relieved by lateral buckle formation taking place at the sleepers. When the pipeline contents temperature reduces, the effective tension increases and the buckles straighten out, but the system does not quite return to the initial state. It is the small residual offset of the pipeline nodes that can cause the "walking" behaviour over multiple heating and cooling cycles.

Static restart analyses to reduce model run time

For the pipeline walking model, the simulation *duration* is 6400 seconds. The corresponding *simulation wall clock time* (which is the real time elapsed whilst performing the simulation) can be viewed in the *properties report* for the *general* object (right click the *general* object in the model browser and select *properties* from the pop-up menu). The wall clock time can vary considerably depending on the machine used to run the simulation. For real-world cases, flowlines may span many kilometres, and segment length requirements are sometimes imposed to meet design requirements. As a result, run times can be significantly longer.

It is possible to simulate a slow, gradual expansion and contraction of a pipeline using a sequential series of static *restart analyses*, and this can often reduce the overall run time, particularly for very long duration simulations. For an introduction to restart analyses the reader could watch the OrcaFlex version 11.1 user group meeting webinar on restart analyses or read through the relevant help page: Modelling, data and results | Restart analyses.

Static restart analyses could be carried out for example every 10 to 100 seconds, whereas the dynamic analysis in this example requires a time step of 0.2 seconds. This large difference in the effective 'time step' required explains why the static restarts are sometimes quicker to run than the equivalent dynamic simulation.

A chain of sequential child restart models, from a parent model with dynamics disabled, can rapidly be produced using the OrcaFlex spreadsheet (see: Automation | Text data files | Automating generation). Alternatively the OrcaFlex programming interface *OrcFxAPI*, run through Python or MATLAB, allows for programmatically creating models, running analyses and extracting results among other functionality (see the dedicated API documentation available through the help menu or here: Python interface | Introduction).

Static analysis can be sensitive to the statics convergence parameters, so when setting up a new chain of static restart models, it can take some time to find a combination of statics convergence parameters for *line statics* and/or *whole system statics* to achieve reliable convergence for all the restart models. For further details on statics convergence parameters, there is a useful "statics convergence guide" in the technical notes section of our website.