Python Automation Examples

Python automation

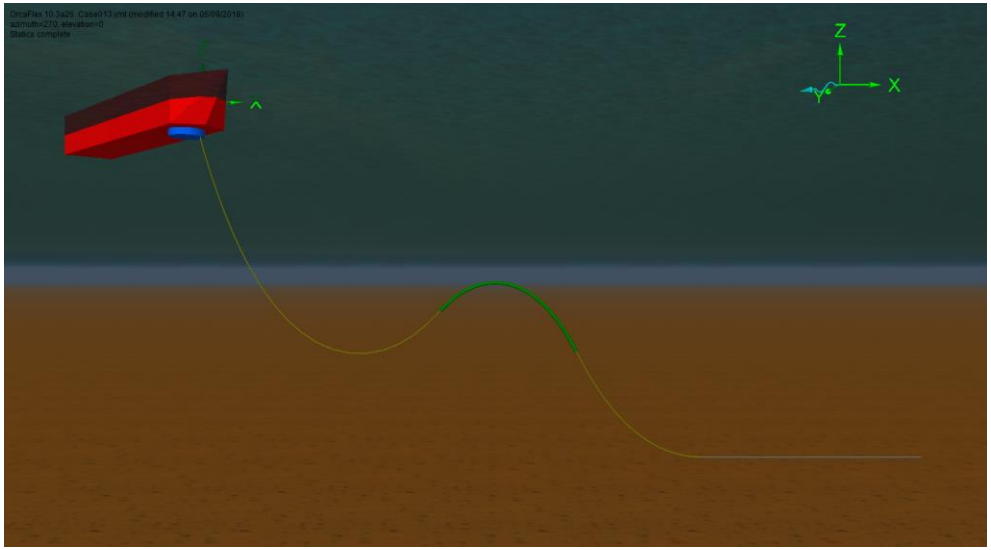
- Python = popular automation choice
- Lots of support questions / requests for examples
- Aim to answer some of those questions here, and the examples will be available on the website shortly
- Examples shown are deliberately kept short and relatively simple
- Covered Python automation in UGMs before, but most recent was 2013!
- Lots of ways to use Python scripts in your OrcaFlex work, we hope to raise awareness of the options with these examples

Python automation

4 short examples to show:

- Pre-processing YAML files
- Post-processing code check results
- Post calculation actions
- Stinger tip clearance optimisation

```yaml
BaseFile: Basecase.dat
General:
  StageDuration[1]:  8
  StageDuration[2]:  40
Environment:
  WaveDirection:  135
  WavePeriod:  8
FPSO:
  InitialX:  -26.870057685088806
  InitialY:  26.870057685088803
  InitialHeading:  -45.0
```



Python automation

DNV OS F101:

Table 4-4 Load effect factor combinations

Limit state/load combination	*Load effect combination*		*Functional loads* [1]	*Environmental load*	*Interference loads*	*Accidental loads*
			γ_F	γ_E	γ_F	γ_A
ULS	*a*	System check[2]	1.2	0.7		
	b	Local check	1.1	1.3	1.1	
FLS	*c*		1.0	1.0	1.0	
ALS	*d*		1.0	1.0	1.0	1.0

1) If the functional load effect reduces the combined load effects, γ_F shall be taken as 1/1.1.
2) This load effect factor combination shall only be checked when system effects are present, i.e. when the major part of the pipeline is exposed to the same functional load. This will typically only apply to pipeline installation.

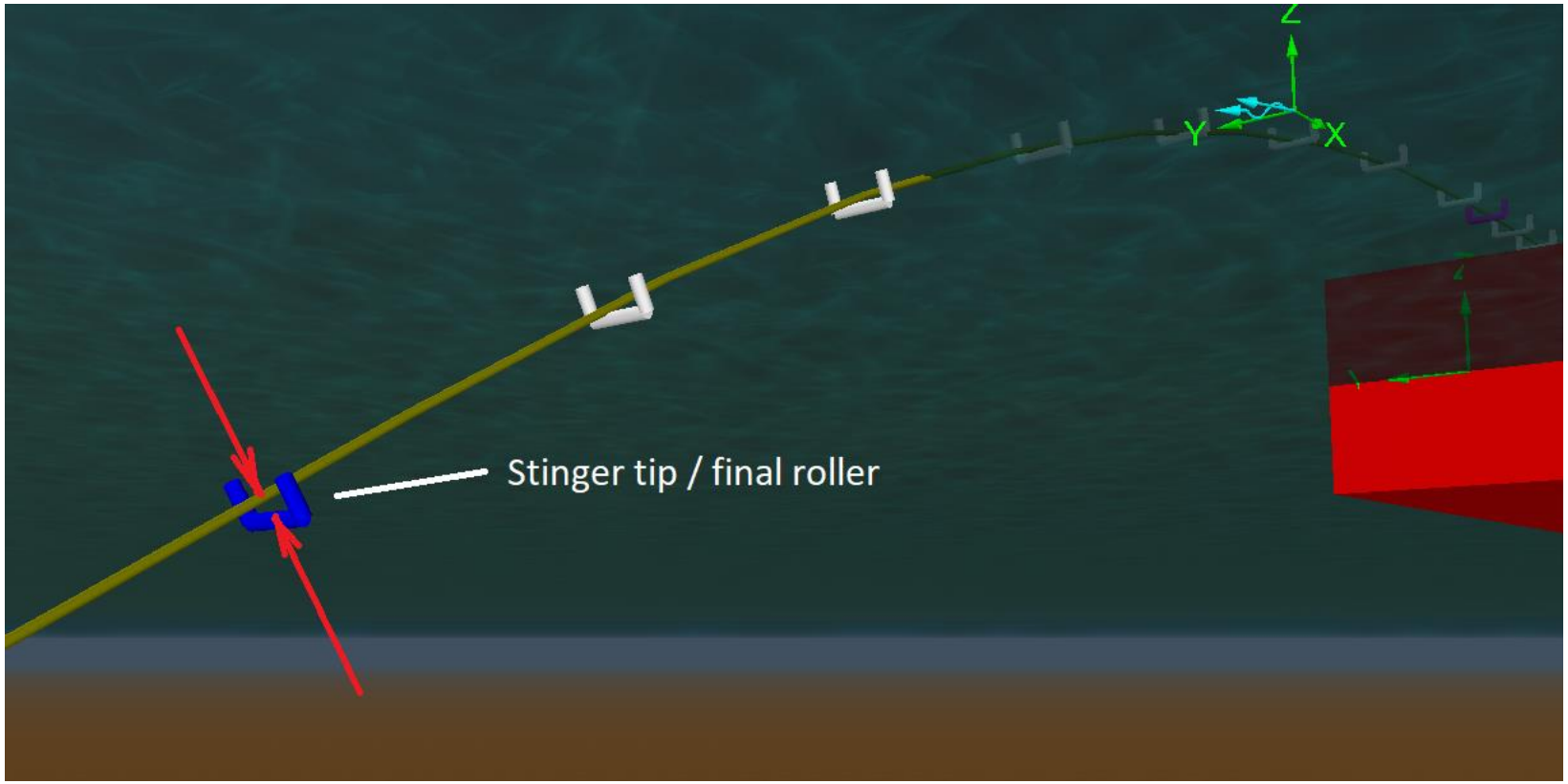
Post calculation actions:

The sequence of events that are carried out around the action's execution is as follows:

- The OrcaFlex analysis, either statics or dynamics, is performed.
- The embedded Python engine is loaded into the process.
- The action's specified script file is imported.
- The Execute function is called.
- Control returns to OrcaFlex which then saves the simulation file.

Python automation

Stinger tip clearance:



Python automation


```
import OrcFxAPI
import sys
from scipy.optimize import fsolve
```

Import the required modules

```
model = OrcFxAPI.Model('E01 Explicit Geometry Stinger.dat')
vessel = model['Vessel1']
pipe = model['Pipe']
```

Identify the model, vessel and pipe

```
targetClearance = 0.2
supportNo = 11
```

Set the target clearance and identify the last roller

Python automation


```
def calcClearance(Length):

    pipe.Length[1] = Length

    try:
        model.CalculateStatics()
    except:
        sys.exit('Static calculation has failed.')

    clearance = vessel.StaticResult('Support Contact Clearance', ……)
    return clearance - targetClearance

initialLength = pipe.Length[1]
optimisation = fsolve(calcClearance, initialLength, full_output = True)
```

Function called by fsolve

Calculation

Measure clearance

f(x) = 0

Initial estimate

Python automation

Where to find help:

- OrcFxAPI Help file: https://www.orcina.com/SoftwareProducts/OrcaFlex/Documentation/OrcFxAPIHelp/
- Introduction to the Python interface to OrcaFlex document: https://www.orcina.com/SoftwareProducts/OrcaFlex/Documentation/index.php
- Python tutorials: http://docs.python.org
- Learning Python book by Mark Lutz
- Previous UGM material: https://www.orcina.com/Support/UserGroup/
- Orcina support / training courses: orcina@orcina.com