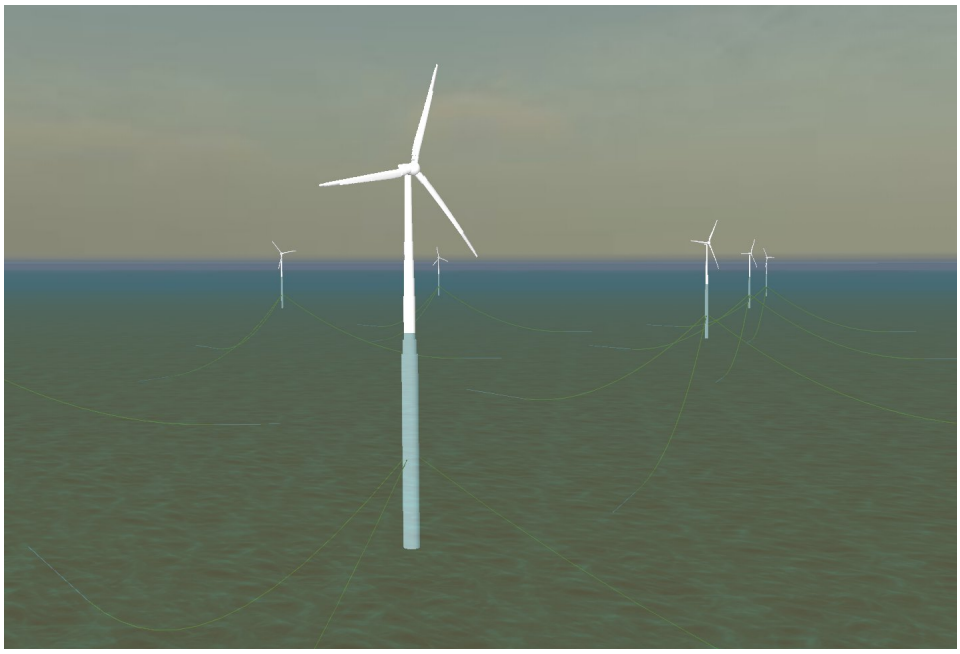
version 10.3

orcaflex newsletter



Contents

In this Newsletter

OrcaFlex 10.3 was released towards the end of 2018 and, for the first time, *not* sent out in the mail on CDs. We are now distributing by electronic download, and all customers on MUS and entitled to the new version have been emailed with instructions for downloading it.

If you think you should have it but don't, then please do contact us (details on the back page).
This newsletter reviews some of the new features in 10.3, and those to come in 10.4.

Wind turbines

OrcaFlex is already widely used for analysis work related to fixed offshore wind turbines, such as installation, power cables, CPS, which do not require modelling of the turbine. For floating wind turbines, there is a need for coupled dynamic analysis of the entire system - moorings, platform/tower, turbine, etc - and for some time now, we have offered FASTlink for this purpose.

FASTlink

FASTlink is an interface which couples OrcaFlex and NREL's FAST/Aerodyn codes, allowing OrcaFlex to model the moorings and platform while FAST/Aerodyn takes care of the tower and turbine. This allows you to take advantage of the strengths of each code, but it does still have some restrictions: working in this way is somewhat cumbersome (model building, post-processing, visualisation, etc.); the coupling between the codes is only a partial coupling, lacking a multiphysics coupling iteration; and models are limited to a single turbine.

OrcaFlex wind turbine object

To improve on this state of affairs we have now developed an OrcaFlex wind turbine object, allowing you to perform this type of analysis entirely within OrcaFlex. Without the requirement for FAST on the

Orcina Limited
+44 (0)1229 584 742
orcina@orcina.com
www.orcina.com

turbine side, there is no longer a need for FASTLink: the modelling process is easier to carry out, and simulations are quicker to run.

The OrcaFlex wind turbine uses a Blade Element Momentum (BEM) model for aerodynamic loading. Turbine blades are represented by beam elements, closely related to OrcaFlex line objects. Blade pitch can be controlled by an external function. The generator can be controlled by either rate of rotation or applied torque, each of which may take a constant value or be calculated by an external function. You can write your own external functions, as usual, or take advantage of our interface to standard Bladed-style DLLs.

The future

There are a number of important features missing from the wind turbine object in OrcaFlex 10.3, simply because we ran out of time. These include

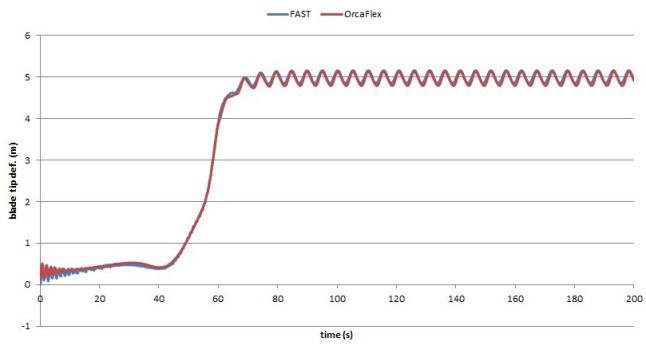
- blade / tower interaction
- dynamic stall
- dynamic inflow
- more sophisticated powertrain model (losses, flexibility, etc.)

Nevertheless, we feel that the turbine object is sufficiently far advanced to include it in 10.3: we are actively developing it so that it will be comprehensive in 10.4, and interim releases (10.3b,c, etc) will add some of the missing features in the meantime.

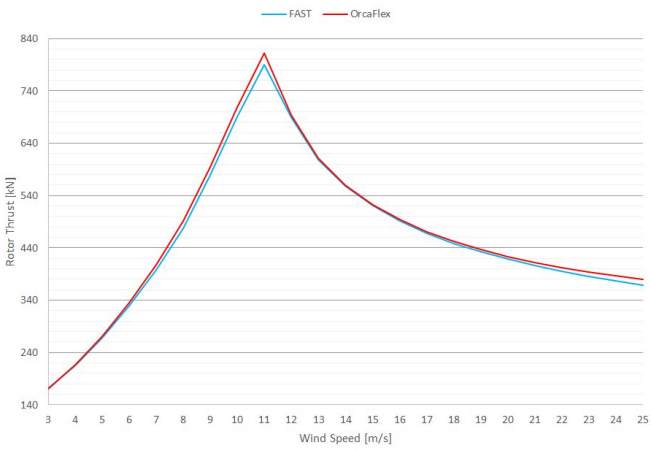
Comparison study

As always, we validate the new features as we add them. We have carried out a comparison exercise to give us confidence in the turbine features we have added so far, against a combination of FAST + AeroDyn (BEM) + BeamDyn (blade dynamics) + OrcaFlex (moorings) + FASTlink (coupling) – in fact, the length of this list alone demonstrates the advantage of being able to do it all in OrcaFlex! Our comparison model is the NREL 5MW turbine on the OC3 Hywind spar in a constant 10 m/s wind, both fixed (i.e. no spar) and floating cases.

Here are some typical results from our comparison, all of which show good agreement between the two methods:



Tip deflection (fixed turbine)



Rotor thrust v wind speed

User-defined results

You're the one using OrcaFlex: you probably have a better idea of what results you want out of it than we do. And up to now, if your idea differed from ours, well you had to take 'our' standard results and post-process them to get 'yours'. Well, no longer. Now you can define your own OrcaFlex results, instead of post-processing to get them, and thereby take advantage of all the standard OrcaFlex results facilities: plotting graphs, linked statistics, extreme value statistics, Excel post-processing, Python/MATLAB automation,...

Why?

This idea has been brewing here at Orcina for quite a few years now, not least because the list of standard OrcaFlex results has grown steadily longer. The trigger for finally implementing this feature was a question asked at a UGM meeting: could we add a new result to report the horizontal offset of a vessel from its static position?

Well, yes, we could. But the requester would have to wait for the next OrcaFlex release for, essentially, a trivial development. And it would also be inflicted on all users, many of whom might have no interest in it.

So, instead, we have developed a framework for you to define your own results, avoiding the need to either wait for them or clutter up the results form unnecessarily.

How?

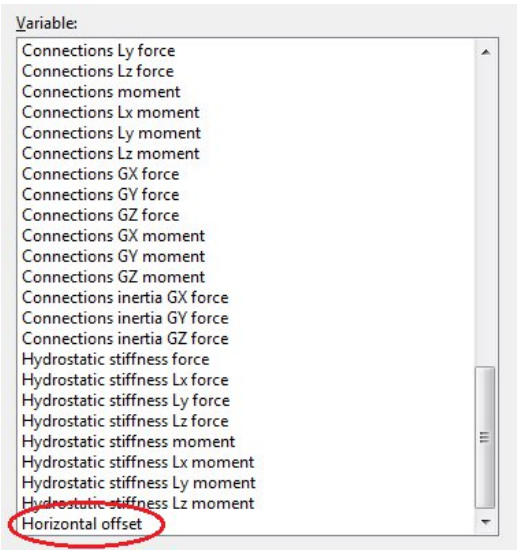
User-defined results take advantage of the Python interface to OrcaFlex. A Python script which completely defines a result has, broadly speaking, two parts: a declaration part which defines the result name, its units, etc, and a calculation part which determines the value of the result(s). Our horizontal offset example looks like this:

```
import numpy

def HorizontalOffset(info):
    obj = info.modelObject
    staticPos = obj.StaticResult(['X', 'Y'])
    pos = obj.TimeHistory(['X', 'Y'], info.period)
    return numpy.linalg.norm(pos - staticPos, axis=1)

def UserDefinedResults(model):
    return (
        {
            'ObjectType': OrcFxAPI.otVessel,
            'Name': 'Horizontal offset',
            'Units': 'LL',
            'TimeDomainFunction': HorizontalOffset
        },
    )
```

Horizontal offset code

Where?

These Python scripts are external to OrcaFlex: they are not included in .dat files, so they can be easily shared between models and users. To add a result to a model, you reference its script on the General data form's user-defined results page (if you've ever used post calculation actions you'll immediately see the similarity). The named results will then appear on the results form and can be selected in the usual way.



Horizontal offset result selection

The figure shows our horizontal offset example as it appears on the vessel results form.

What?

This is a complex feature, so we have documented it comprehensively in the help (also available online) and included a number of downloadable examples to help you get up and running. We will, of course, be happy to provide support in the usual way.

Although there is a bit of a learning curve here, there is also potential for great flexibility and power. For example, we have already seen one user write a set of user-defined results to implement a code check not currently supported by OrcaFlex.

Quasi-dynamic analysis

We are continuing to develop the abilities of OrcaFlex as a mooring analysis tool. Version 10.1 extended the frequency domain solver to the low-frequency regime, and we have now implemented *quasi-dynamic analysis*, a method popularised by BV's Ariane. Mooring analysis tends to involve large numbers of load cases and long duration simulations to capture accurate statistics, and methods like frequency domain and quasi-dynamics yield much greater efficiency and shorter runtimes than nonlinear time domain – though at the cost of some accuracy.

We have worked hard to integrate this new feature seamlessly with the existing forms of analysis in OrcaFlex, so that you can build a single model of your mooring system and easily perform frequency domain analyses, quasi-dynamic analyses and nonlinear finite element time domain analyses, all from the same base model from which you derive all your load cases.

You might, for instance, use quasi dynamics and/or frequency domain to sweep over the load case matrix to identify critical load cases, then examine these critical cases in more accuracy and detail using nonlinear time domain analysis.

This allows you to select and use the most appropriate tool for each stage of your mooring analysis without having to re-build the model in different programs.

What is it?

In a quasi-dynamic analysis the usual finite element line model is greatly simplified, and we represent the

line instead by a non-linear spring. This simplification means that we can, at the start of the simulation, calculate a lookup table relating fairlead tension to offset. This table is calculated once only (via analytic catenary equations), at the start of the simulation, so saving a large amount of calculation and, potentially, greatly reducing the simulation time.

There is, of course, a price to be paid for this simplification, and that is a reduction in accuracy. During the simulation, we determine the tension in the line – the spring's restoring force – by interpolation of the pre-calculated table. Inertia, drag and damping contributions from the line are all ignored, though the damping and drag can be accounted for (and usually are) by estimating their overall effects from the mooring system and applying these directly to the vessel as linear and quadratic other damping.

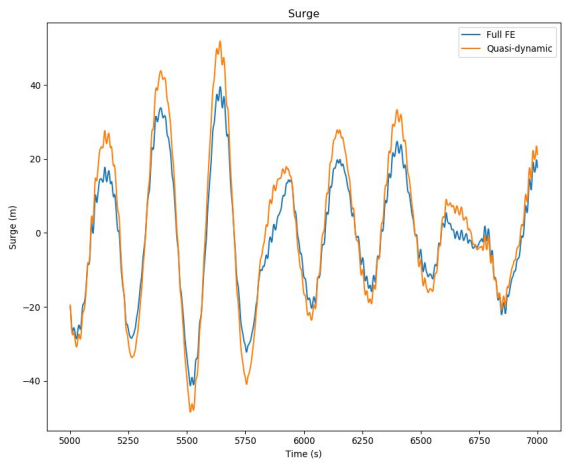
We may not be able to predict line responses and loads accurately in this way (in fact, we offer only a restricted set of results for lines modelled in this way); we do however expect this method to be reasonably accurate for the motion of the vessel.

As usual, there is much more detail in the OrcaFlex help on this.

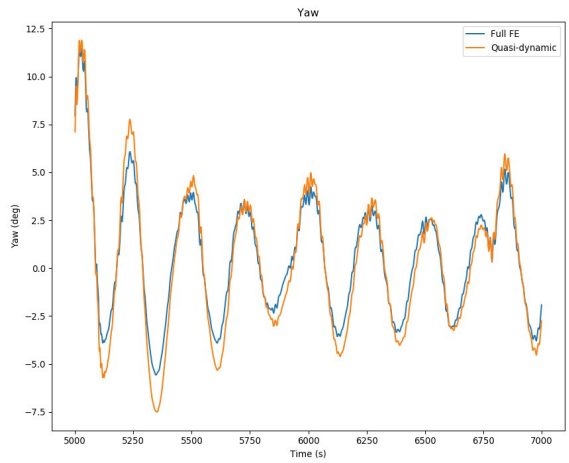
Comparison

We set up a simple model to demonstrate the potential benefits of quasi-dynamic analysis. We used our default OrcaFlex vessel to model a spread moored FPSO with chain/wire/chain mooring lines and carried out 3 hour time domain simulations, and obtained a 60x run time reduction for quasi-dynamics over the full FE.

Here are some selected results from this comparison,



Surge



Yaw

Clearly, there is a reduction in accuracy – there's no such thing as a free lunch! But the potential for use in screening studies is obvious and, for systems amenable to this form of analysis, quasi dynamics could be used for the bulk of detailed analysis too.

Other analysis

We have described quasi-dynamics here in the context of mooring analysis, but you should feel free to apply it in other settings. OrcaFlex supports a mix of quasi-dynamic analysis and finite element analysis – you can choose to have all, some or none of the lines in your model treated in this way.

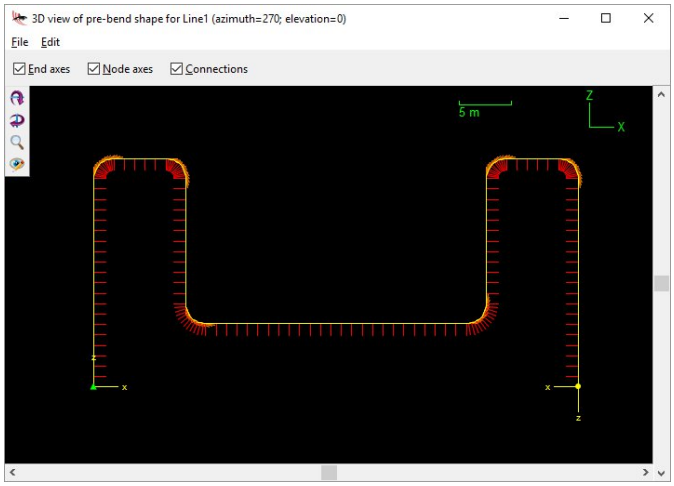
Other developments

Those are the headlines, now here are a few other improvements we've made to make your OrcaFlex-using life a bit easier.

Line pre-bend made easy

Line pre-bend was introduced in v8.5 to allow you to model lines whose unstressed state is not straight – a rigid jumper, for instance. Working out the data to define pre-bend can be tricky, so for 10.3 we have tried to make this an easier process.

The biggest difficulty was in visualising the pre-bend shape. The pre-bend data page now has a **view pre-bend shape** button, which brings up a 3D view of the unstressed line shape and node orientations. You can open as many of these windows as you like, for instance to show plan and elevation views simultaneously. Edits to the data are instantly reflected in all of the windows.



View pre-bend shape

Another difficulty was that the pre-bend shape was defined by x and y components of curvature, not terribly intuitive quantities to grasp. It can now be specified alternatively, by entering bend angle, bend radius (or equivalently arc length) and the angle that defines the plane of bending.

Friction for line supports

In previous versions of OrcaFlex, the line support contact model did not include any friction forces. Version 10.3 adds the ability to include friction for line supports. The friction coefficients are specified on the *friction coefficients* data form. This data form was previously known as the *solid friction coefficients* data form – the name was changed to reflect the wider applicability of the form.

There isn't much more to say about this! The friction model is the same as used for seabed contact, elastic solid contact and line contact models, so should be familiar.

Be aware, though, the normal and axial friction coefficients are defined relative to the *support's* axis, in contrast to line friction coefficients where normal and axial are defined with respect to the *line*.

Object tags

Sometimes it's necessary to associate additional data with objects in an OrcaFlex model – for instance to pass data to a PID controller, as in our external function examples.

Previously we did this in OrcaFlex with a page, on a data form, where you entered such data as text and from where they were passed to a script. This worked reasonably well, but still had a few inconveniences, so we have invented object tags to make it more convenient to specify additional data.

Object tags are *dictionaries* (much like those in Python) of pairs of names & values, which make it easier to enter and read the data, and simpler to automate their modification.

Tags are offered for every type of object. They can be used with external functions, post-calc. actions, user-defined results, etc; you are entirely free to decide if and how to use them.

Post-processing multiple time histories

One of Orcina's founders (now retired) was of the opinion that half a dozen load cases should be enough for any analysis, if they were chosen with sufficient care. Six!

Well, computers get ever faster, and acquire ever more processors and cores, and as a result ways of working change. And, when they do, they can sometimes reveal inefficiencies in OrcaFlex in areas that, previously, just didn't *need* to be efficient. Six load cases are, after all, not likely to tax many post-processing schemes.

One post-processing bottleneck we recently tackled is that which occurs when extracting time history results for lines. Lines in particular because, to keep files small, we save only a minimal set of the 250 or so line results – simplifying a little, these are *X, Y, Z* and tension for all nodes – and derive all the others from these when they are required.

This means that if you wish to extract a time history for *x*-curvature, say, then OrcaFlex needs to read *X, Y* & *Z* from the log for the 4 nodes surrounding the target segment, then perform some quite involved geometry calculations to get the value. If you then ask for *y*-curvature, OrcaFlex does it all over again. x-bend moment? Again. *y*-bend moment? Again.

Similarly, you might ask for results at multiple adjacent locations along the line. For segment 3, say, you need to read the values for nodes 2, 3, 4 and 5. Then for segment 4, you *re-read* the same values for nodes 3&4 before reading them for 5&6.

If you only have six load cases, this inefficiency doesn't really matter. If you have six hundred, however, it rapidly becomes untenable.

Well, version 10.3 introduces a new API function, *GetMultipleTimeHistories*, to remove this inefficiency. The benefit you will gain from this depends on which results you post-process, but we expect that, in many cases, post-processing will be around 10 times faster.

Full documentation, as usual, is available in the OrcFxAPI help. And don't forget that this function can be used especially effectively in the context of a post calculation action.

Diffraction analysis

Now, something of a teaser. We have begun developing diffraction analysis capability in OrcaFlex, for release in 2019 (not in 10.3, sorry!).

Our intention is to assist OrcaFlex vessel users with their workflow, especially those who have difficulty in sourcing suitable hydrodynamic data. And even if you do have the data, vessel RAO conventions are a nightmare – degrees/radians? up/down? crest/trough? leads/lags? port/starboard? – but it is crucial that they are correct.

Well, how would it be if OrcaFlex took care of all that for you? We already do so when we import AQWA and WAMIT; we will soon do it more generally by adding – at no extra cost – diffraction analysis capability for OrcaFlex, in which case all the data we will require will be the shape of your vessel.

Progress

We have reached the stage of being able to perform, for free-floating and partially fixed bodies, a complete first-order analysis: added mass and damping, load RAOs, displacement RAOs and sea state RAOs can all be computed.

Still to do

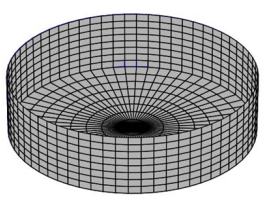
This is a long-term project, and still to be completed are: second order analysis (mean drift forces and QTFs), multi-body cases, results checking, and GUI developments such as mesh display.

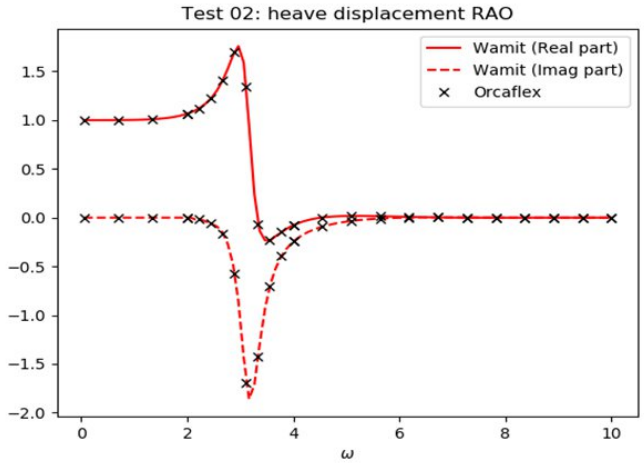
Note that we are not developing a CAD package – you will need to provide a mesh representation of your vessel to import into OrcaFlex

Validation

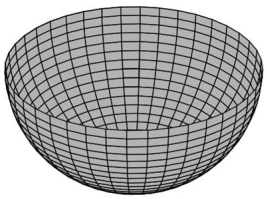
We are, of course, validating our implementation as we go. Here, for instance, is a heave displacement RAO comparison with one of the standard WAMIT examples, 02 – a vertical cylinder,

Vertical cylinder mesh
(1024 panels, Wamit test 01, 02)

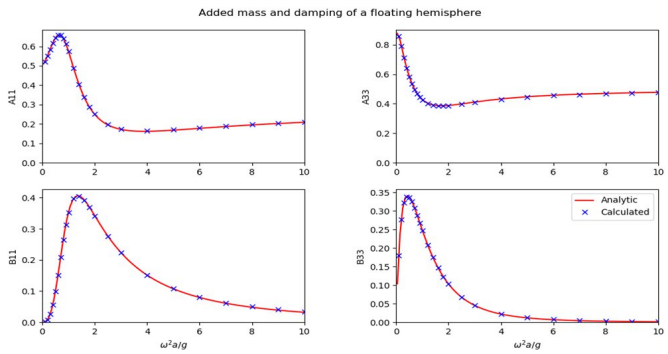




And here, added mass and damping for a floating hemisphere compared with the analytic solution,



Hemisphere mesh
(840 panels, test vs the analytic results Hulme (1982))



Feedback

We would welcome any feedback you may have on our diffraction analysis development:

- What type of analysis do you anticipate running? e.g. single or multi-body, free-floating or anchored, other geometrical features...?
- What file formats for the mesh would be most convenient for you?
- Are there any test or validation cases that would be useful for us to consider? Challenging ones, especially!

Coming up

Diffraction, though it is for the future, is clearly well underway. The wind turbine enhancements we plan are also already listed above. Other developments underway or in the pipeline include:

Mooring analysis

Implementing quasi-dynamic analysis is a big step towards making mooring analysis easier in OrcaFlex. We plan to progress further in this direction, by extending extreme statistics to include more relevant results and by offering a workflow more specifically tailored to performing mooring analyses.

Spatial wind

Wind turbine models require a full-field wind representation. This can now be done in OrcaFlex, allowing for variation of wind velocity in both space and time, with data specified in an external file.

At the moment, this only supports the TurbSim .bts binary file format. We intend to add support for other commonly used file formats, and also to extend OrcaFlex to generate turbulent wind fields directly.

Line results at nodes

Reporting line results at nodes as well as at mid-segment arc lengths has been on our development list for quite a while now – well, we really are planning to get to grips with it this time round.

In-place pipeline analysis

Including: restarts, lateral soil modelling, pipe expansion/contraction due to thermal and pressure effects.

Software licensing

We are now working on FlexNet software-based licensing for OrcaFlex, as an alternative to HASP dongles. We will continue to offer HASP dongles for some time yet (probably years), but moving away from physical dongles opens opportunities for cloud computing, time-limited evaluation licences, all sorts of wonderful things.

As always, suggestions and any other feedback on our proposed developments are always welcome. Much of this feedback comes from user group meetings – so thank you to all who have kindly contributed to this.

And finally...

You might have noticed our new logo at the head of this newsletter – new, but still featuring Orcy the orca. Orcy first arrived on the scene way back in 1986 when Orcina began,

At that time, OrcaFlex was distributed on a single floppy disk. Flat Orcy served us well until the arrival of 3D Orcy with OrcaFlex 9.4 in 2010, by which time we'd moved on to CDs,

Now, a further eight years on, we are pleased to introduce you to a brand new streamlined Orcy for the digital download era:

In addition to the new logo, there is a completely new website in the pipeline as well, and we hope to be launching that very soon - watch this space!

the back page

News

We aim to publish these Newsletters yearly, to complement each OrcaFlex release. Around the same time we also produce a video (available from orcina.com/SoftwareProducts/OrcaFlex/Videos) some features are best seen dynamically rather than on a static page.

We do also try to keep you up to date with developments on a more frequent basis. Our company **LinkedIn** page has been going for a while now, at linkedin.com/company/orcina-ltd. New followers are always welcome. The **OrcaFlex Blog** may be useful as a source of more detail on some of the v10.3 developments: these are all wrapped up in a recent post, orcina.com/blog/orcaflex-103. Future posts will cover, among other things, 10.4 plans and developments.

Out and About

During 2018, as well as the usual mix of training courses and UGMs (see below), we exhibited at Subsea Expo (Aberdeen), Oceanology (London), Subsea Tieback (Galveston), and attended Offshore Wind (Glasgow), Floating Offshore Wind Turbines (Marseille), Global Offshore Wind (Manchester), OPT (Amsterdam), OMAE (Madrid), and Subsea Lifting (Stavanger).

For 2019, current plans are to exhibit at Subsea Expo (Aberdeen), Oceanology (San Diego), Subsea Tieback (San Antonio), Ocean Business (Southampton), OMAE (Glasgow), Offshore Wind (London), Offshore Europe (Aberdeen), and to attend OPT (Amsterdam) and Subsea Lifting (Stavanger).

Training

OrcaFlex training courses proved increasingly popular in 2018, particularly our course on using Python with OrcaFlex.

Planned courses, as well as other events, are listed at orcina.com/UpcomingEvents. In addition, we are always open to requests for training, whether it is our standard introduction to OrcaFlex, Python automation, or more advanced and tailored to your specific requirements.

User Group Meetings

In 2018 we again held 11 User Group meetings at various locations around the world. This year these were Houston, London, Aberdeen, Stavanger, Amsterdam, Kuala Lumpur, Jakarta, Singapore, Perth, Shanghai and Rio. Content from these can be found at orcina.com/Support/UserGroup. We also held our 'Introduction to Python' workshop in Houston and Ulverston.

Thanks to all who attended and contributed to their continuing success.

2019 UGMs will run between September and December – we'll post details of these and other events on orcina.com/UpcomingEvents and on LinkedIn. It's likely that we will also hold Python workshops in 2019: these are yet to be decided, but again, we'll post details as soon as we have them.

Orcina Agents

Orcina is supported in its marketing and technical support activities by the following agents:

USA, Canada & Mexico
Jacob Technologies
Paul Jacob
pj@jtec-tx.com
+1 713 398 9595
Heron Offshore
Dongmei Chu
dchu@heronoffshore.com
+1 832 725 2438

South Korea
SACSKO OceanTech
Hyunwoo Jang
hyun.j@sacsko.com, +82 2 421 8018

South America
NSG Engenharia
Nelson Galgoul
nsg@nsg.eng.br, +55 21 99995 9212.

China
Richtech
Yujing (Jean) Chen
orcaflex@richtechcn.com
+86 10 8446 7760 / +86 1812 129 2356.

Social Media

If you use LinkedIn please follow us:
Orcina Ltd - the home of OrcaFlex

Orcina Limited
+44 (0)1229 584 742
orcina@orcina.com
www.orcina.com