## NAFEMS Optimisation Working Group

**Special Interest Group Presentation on**

**'Human-Inspired Generative Design' by Dr Richard Ahlfeld, CEO and Founder of Monolith AI.**

**Below are Richard's responses to additional questions submitted for the Q&A Session**

Are you able to impose computational constraints on the use of a trained ANN?
Yes. Like in any other optimisation problem you can impose constraints on the input variables. We have seen problems were opening up the pre-defined design space to explore some more unusual options through extrapolation has been very interesting. In these cases it is really important though to have a good uncertainty model, e.g. Gaussian process to communicate how far away from known conditions you are.
If you want to learn more, feel free to contact info@monolithai.com.

In the turbine example, are the design variables that are used for optimization the parameters used within the ANN layers or are they actual practical design parameters? Are you able to determine the practical significance of ANN parameters?
The variables were latent variables found by the ANN, thus they were all between 0 and 1. Yes, it is in some cases possible to determine the practical significance of the parameters. In other cases it is even possible to force the ANN to adopt physical parameters. Happy to have a chat if you want to learn more

Are these results steady-state or transient. Do you also use time-series data to create designs? If so, How do you generate design based on transient results?
The results shown were from steady-state simulations.
We have used time-series data in various flight physics applications as well, for example: dynamic response of an aircraft after experiencing a gust.
To find optimal designs for transient results you need to define an appropriate cost function. For example for the gust scenario, we selected the maximum amplitude in the angle of attack as the QoI that we wanted to minimise.
Feel free to reach out to our simulation engineering via info@monolithai.com.

Do you have any suggestions about how you could create a large enough dataset to train the autoencorder if I currently only have a single CAD design?
We created the wind turbine use case using a parametric design made in grasshopper. If you do that the amount of work needed to use Monolith becomes the same as for other parametric optimisation approaches. The only advantage is that the deep neural networks are really robust towards large design changes. It is not necessary to use only a few variables or only model minor design changes, you can go all out.

If you want to learn more, we have a series of parametric design tools and opensource solvers like openFOAM integrated in monolith. Happy to have a chat of how the approach we used for our demo could help you.
Feel free to reach out to our simulation engineering via info@monolithai.com.

To train the models you created your own database
Yes, we ran around 600 CFDs for different geometries of wind turbines. That was only to demonstrate the tool though. We recommend you use it for a database of existing simulations for components you build regularly.

Part of your technology rely on FEA morphing capabilities ?
It does not rely on it. Not necessarily. We have used morphing to augment databases if we did not have enough designs for a use-case. It is not our favourite way of expanding the dataset though as it is a rather crude way to find more data in complex design spaces.

Does it still rely on simulation of specific configurations?
I think I am missing the original context here, but normally it does, yes.

Does the model run the simulation every time you change the input parameters or you actually use a sort of best-fitting using a database you already have?
There is no model being run or an interpolation being performed. The 3D you saw were predictions made by a deep neural network for a) what the best geometry looks like b) what the CFD result for that geometry looks like. That we can do that is the main breakthrough behind our software platform!

Is this working on unsteady / two phase CFD simulations?
I believe so. Difficult to say without knowing your use case, but we have used it for understanding cooling hole effects in gas turbine blades.

How many CFD simulations were used to train the surrogates for the wind turbine?
600 in this example. You do not necessarily need that many. Also bear in mind, that it is less about the number and more about the total computational effort needed for generative design. For a standard optimisation it sounds like a lot, but the overall computational effort is very similar to any other generative design software.

First of all: very interesting and cool ... my question is: does the approach find solutions "outside" the training set?
It could, but as it is mathematical extrapolation and not physical modelling it is rather unwise to try it unless you have means to test it later (prototype test to confirm extrapolation is correct). We recommend using this to automotive the repetitive parts of designs, e.g. brake pads on a car, sealing solutions on an engine, and use different approaches when you want to create out-of-the-box innovative designs.

How you got the Stress values? For whole training set did you have these values?
The stress values were derived from the CFD simulations. CFD --> loads --> FEA for internal stress. Yes, we performed those calculations for all CFD simulations

Is this approach also possible for FEM electromagnetic design?
As it is purely mathematical, it should apply to any area. I would be personally really interested to test it on one. Let's talk more about your use-case.

Machine learning works fine based on a large number of data. Can I ask how can it work under a condition of lack of data or no data?
Machine learning is defined as deriving laws from data without having to programme them. As such it simply does not work without data. There are methods to get results with sparse data, e.g. rely on gaussian process or linear Bayesian models. The most interesting we have learnt is this: if I optimise a wind for lift/ drag from 10 points where I got lift / drag as single points out of simulations vs if I build a 3D neural network from all individual pressure / velocity points of which I have 100,000s, then the neural network optimisation will be 20% better than the single point.