PRE-TRAINING VARIED VASCULAR GEOMETRIES WITH A DEEP LEARNING SIDE NETWORK IN PHYSICS-INFORMED NEURAL NETWORK SIMULATIONS OF VASCULAR FLUID DYNAMICS

Hong Shen Wong[1], Binghuan Li[1], Wei Xuan Chan[1], Choon Hwai Yap[1]

1 *Department of Bioengineering, Imperial College London, UK*

Introduction

As universal function approximators, deep neural networks have the potential of being the surrogate solver of the Navier-Stokes (NS) equations. This was recently demonstrated via the Physics Informed Neural Network (PINN) on aneurysm flows [1]. However, PINNs are specific to the geometry of the flow domain and require slow training for each new geometric scenario encountered. To address this, Sun *et al.* [2] designed a simple parameterization of varied vascular geometries and pre-trained various geometric scenarios by adding the geometric parameter as an input to the PINN which allowed for the quick prediction of new geometric cases. Here, we present an alternative approach, where a deep learning (DL) side network is cascaded to a PINN domain network for the pre-training of varied geometric cases, which has the potential to enhance network robustness and decrease training complexity.

Method

The DL-PINN network architecture is shown in Fig 1. The network was tested on 2D stenosis flows, where the vascular geometry can be varied by a parameter to control stenosis severity. The DL-PINN was trained with 5 geometries with increasing stenosis levels and tested on 3 geometries with stenosis levels not yet seen by the network.

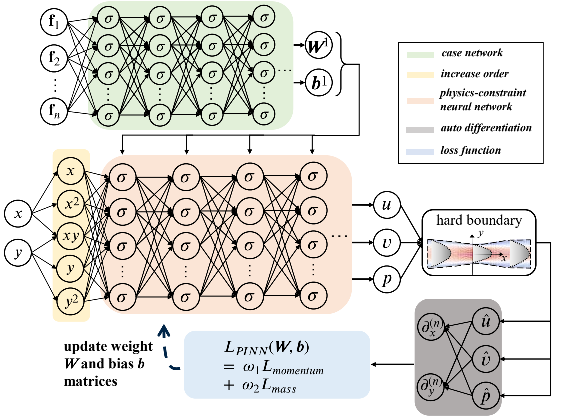

Figure 1: *The architecture of the proposed DL-PINN network. A DL network (green) using case geometric parameters ($\boldsymbol{f}$) is used to determine the weights of nodes in the PINN network (pink). A PINN pre-layer (yellow) increasing order of parameters, and a hard boundary constraint post-layer improved performance.*

Result and Discussion

Importance of hard boundary post-layer: no-slip wall velocity and inlet/outlet boundary conditions were incorporated as hard constraints via polynomial profiles across vessel diameter, and *tanh* profiles along vessel length, which were imposed onto velocity outputs from PINN. This improved convergence of velocity from a 32.2% error to a 0.2% error in a test case.

Importance of increased-order pre-layer: PINN nodes are typically modelled as tanh functions, but such networks cannot model the second-order math functions (such as x^2 and xy) well. Our inclusion of the second-order pre-layer to calculate these functions improved convergence from 32.2% velocity error to 4.7%.

Pre-training of varied geometric cases and predictions in unseen cases: Using our case network pre-trained with 5 geometric cases of varied stenosis, prediction of flow and pressure fields for the 3 new cases unseen by the network had velocity and pressure errors around 0.2-0.8% and 3.5-4.5%, respectively, comparable to errors achieved in the training cases. Fig 2 shows the comparison with CFD results, showing a good match. Predictions of new cases were almost instantaneous, compared to the 2-4 minutes needed for CFD simulation of the same cases using COMSOL.

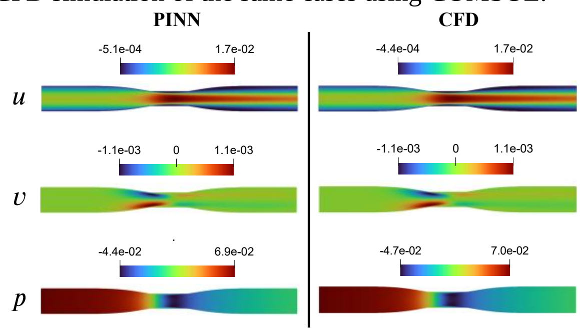

Figure 2*: Comparison of the results from a PINN-predicted case and the corresponding CFD solutions. The order of magnitude of each quantity from the predicted case is the same as that from the CFD solution. The relative L2 error of 3.16% was achieved.*

Conclusion

We demonstrated the feasibility of a DL-PINN network pre-trained to varied geometries and validated its ability to rapidly solve flow fields of geometric cases not yet seen by the network. In the future, this can be expanded to cover various vascular curvatures, cross-sectional size changes, and 3D flows. If successful, such an approach has important benefits. First, it can achieve ultra-fast, real-time fluid mechanics simulations to assist clinical evaluation. Second, it allows the network to evaluate various surgical options to find the optimal route, via parameterization of these options.

References

[1] Raissi M, et al., *Science*. 2020 Feb 28;367(6481):1026-30.
[2] Sun L, et al., *Comput Methods Appl Mech Eng*. 2020 Apr 1;361:112732.

Acknowledgements

This work was supported by the Imperial college startup fund.