Independent Research Project Plan

Shallow water flow field inference using data-driven method

Mingrui Zhang

Supervisor: Matthew Piggott, James Percival

June 2019

1 Introduction

Shallow water problems are quite common in the real world. Many fluid flows in the oceans, coastal regions, estuaries, rivers and channels could be defined as shallow water flows. The general characteristic of these flows is that the vertical dimension is much smaller than the typical horizontal scale. In this case the vertical dimension can be removed by averaging over the depth. Tidal currents represent a typical shallow water flow. It is able to produce promising renewable energy due to its high volume and periodic occurrence. Tidal energy has the potential to become a valuable part of the energy supply in the drive towards a carbon-free society [1].

Tidal turbine farms are used to collect tidal energy. One critical challenge for constructing the farm is how to choose a proper farm location and size. These usually decide the availability and capacity of tidal energy. To deal with it, coastal modelling could help analyze and predict the potential energy. Traditional coastal modelling relies on computational fluid dynamic(CFD), which solving the Navier-Stokes (NS) equations. However, due to high computational costs, CFD-based approaches are not efficient when dealing with large scale problems [2].

One possible approach to improve the efficiency is flow field inference. Given collected oceanic data (such as Satellite observations, drone taken images, limit sensor measurements etc.), a inference model could predict the related physics properties (velocity field, vortex strength etc.) of the target area. These results could be assimilated into numerical models and help conduct rapid simulations.

1.1 PIV and PTV Method

In experimental fluid mechanics field, image processing techniques are used for flow field identification and prediction such as measuring flow velocities. Particle Image Velocimetry (PIV) and Particle Tracking Velocimetry (PTV) are two popular measurement techniques. These methods are used to obtain instantaneous velocity measurements and related properties in fluids.

When conducting PIV technique, the target fluid is seeded with sufficiently small tracer particles. These particles are assumed to faithfully follow the flow dynamics. With illumination, the particles in the fluid are visible. By comparing the flow images between timestamps, the flow motion information can be calculated using image processing algorithms. On the other hand, PTV is a Lagrangian approach,which works by tracking individual particles [3].

These methods are feasible in lab scale experiments. However, when considering the large real world problem such as ocean modelling, it is difficult to track ocean currents using particles.

1.2 Data-driven Method

Data-driven or machine learning methods have much great progress in applications to many real world problems in recent years along with the huge improvement of computation ability. The deep learning techniques show an obvious advantages when tackling problems which are complex and hard to model. These problems include image processing, computer vision, natural language processing etc[4][5]. There are some highlight deep learning techniques introduced below.

Convolutional Neural Network(CNN) is a kind of deep neural network, which most commonly applied to computer vision. The CNN extracts features from objects using the convolutional kernels in multiple hidden layers simulating the visual recognition in biological processes [4]. A recurrent neural network (RNN) is a kind of neural network where connections between nodes form a directed graph along a temporal sequence. This allows it to process time series data. RNNs can use their internal memory to process sequences of inputs [6].

Deep learning technique can be regarded as a surrogate model method for inverse problems. The key idea of common deep learning technique is using neural networks substitute the physical model.The inspiration of neural networks comes from imitating the intelligent thinking of the brain[7]. There are several layers for each neural network. Each layer has a series of simple processing units, called neurons. The output value of each neuron is obtained by applying a transfer function on the weighted sum of its inputs. The neurons weights are optimized(or trained) using algorithm such as stochastic gradient descent, by minimizing a cost function on a training data set [8]. The well trained neural network is expected to extract the features and information from the underlying physics process. Then the direct mapping relationship between observed data and solutions is established. When given new data, the network is able to predict the solutions according to the learned features.

Flow field inference is a typical inverse problem, which aims to find the relationship between observations and interested physics properties. It is deserved to apply deep learning techniques to this field [9]. Some related work have been reported recently, a literature review would be showed in the next section.

2 Background

Data-driven techniques have been widely used in fluid dynamic field recently years. The main applications include flow field inference, dimension-reduction, turbulence modelling, flow control.

Many works have been reported in flow field inference using data-driven methods. Rabault et al [3] proposed a proof-of-concept for applying artificial neural network performing particle image velocimetry and compare the results with traditional PIV methods. Erichson et al [10] implement flow field reconstruction using limit velocity sensor measurements with a shallow fully-connected neural network. Deep learning techniques are also applied to unsteady flow field predictions. Miyanawala et al [11] proposed a neural network based method to predict drag coefficients of unsteady wake flow given the geometry and velocity field as inputs. In unsteady fluid field visualization area, a Convolutional Neural network(CNN) based feature extraction frame was implemented by kim et al [12], which achieving higher accuracy compared to linear optimization method. In addition to spatial flow field predictions, a temporal-spatial unsteady flow prediction method was reported by S.Lee et al [13]. A GAN(Generative Adversarial Networks)combined with CNN was applied to process flow time series data and make predictions. As for ocean current scenarios, T.Bolton et al [14] implement a CNN algorithm to predict unresolved turbulent processes and subsurface flow fields using oceanographic observations. K.Franz et al [15] proposed ocean eddy identification and tracking method by C-LSTM(convolution long-short-term-memory) and CNN.

Dimension-reduction is another important application. Z. Wang et al [16] presented a reduced order fluid dynamics systems using deep learning. The method is combination of traditional POD(proper orthogonal decomposition) [17] and neural network, which showed a better performance compared to POD only method. D. Xiao et al [18] made a step further, applying the POD-NN to generate a ROM(Reduced Order Model)for turbulent flows in the urban environment. Dimension-reduction modeling also applied to Reynolds stress models, Z.Zhang et al design a neural network to predict the Reynolds stress of a channel flow of different Reynolds numbers [19]. The results showed a higher efficiency comparing to DNS(direct numerical simulation) and LES(large eddy simulation) method.

When making trade-off between interpretability and tractability or efficiency, data-driven method usually sacrifice the former one when the network getting deeper [20]. However, turbulence (and related complex system) modelling highly requires physical interpretablity. Common "black box" deep learning method is not suitable here. Therefore, Physics-Informed neural network is proposed [21] [22] [23], which combines some physics rules as prior knowledge. J.Ling et al [24] applied neural network to a RANS(Reynolds averaged Navier Stokes turbulence models),learning the Reynolds stress anisotropy tensor from high-fidelity simulation data. N.B. Erihson et al [25] implemented a physics-informed auto-encoder for Lyapnov-stable fluid flow prediction, using the stability of an equilibrium as prior knowledge.

As for flow control, Rabault et al [26] presented the application of an Artificial Neural Network trained through a Deep Reinforcement Learning agent to perform active flow control. J.Morton et al [27] also proposed a unsteady flow active control method using deep learning. A stable dynamical models is trained according to label from Koopman theory. The model can predict the time evolution of the cylinder system over extended time horizons.

In this project, we focus on the flow field inference problem in a shallow water flows scenario. The related methods mentioned above mainly compare the data-driven method with Computational Fluid Dynamic(CFD) simulation results. A shallow water lab experiment data would be collected as benchmark to our methods. In addition, the developed model is expected to take drone taken coastal images as test data, which would be a step further in this field for real world application.

3 Plan

3.1 Project objectives

Investigate cutting-edge data-driven flow field inference methods. Develop a data-driven(based deep learning techniques, using CNN as the start point then try to explore other methods such as LSTM and RNN etc) model which can conduct flow field inference (using velocity field as the start point). Apply the model to shallow water flows scenario (a lab scale model simulating tidal turbine farm). Evaluate the model using real world data(lab experiments data). Explore useful features which could be assimilated into numerical models and help conduct rapid simulations.

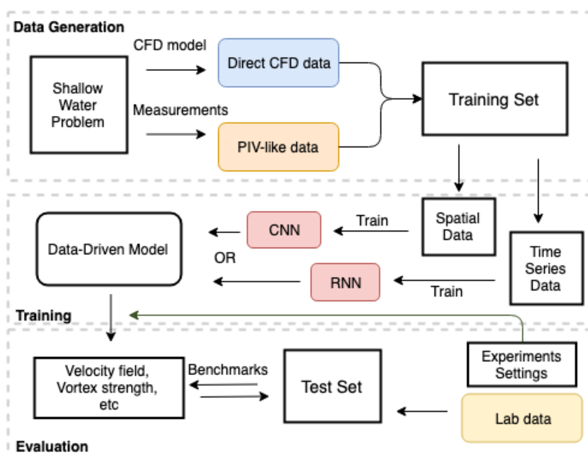


Figure 1: Workflow

3.2 Measures of Success

As for evaluations, the inference model are expected predict the properties(such as velocity field) of the flow field given input data(such as one image or limit sensor measurements). The predictions would be compared with lab experiments data as benchmarks.

3.3 Software and libraries

1. Firedrake (2019)

   Firedrake is an automated system for the solution of partial differential equations using the finite element method (FEM). Firedrake uses sophisticated code generation to provide mathematicians, scientists, and engineers with a very high productivity way to create sophisticated high performance simulations [28]. In this project, Firedrake is the basic tool for all CFD simulations.

2. Thetis (2019)

   Thetis is an unstructured grid coastal ocean model built using the Firedrake finite element framework. Currently Thetis consists of 2D depth averaged and full 3D baroclinic models [29].

   The shallow water flows in this project are described by Shallow Water Equations(SWE). Thetis would be used to model shallow water flows and solve SWEs.

3. OpenPIV (2019)

   OpenPIV is the community driven initiative to develop free and open-source software for Particle Image Velocimetry (PIV) image analysis and post-processing. PIV method would be used as comparison to our data-driven method.

4. Paraview (v5.6.0)

   ParaView is an open-source, multi-platform data analysis and visualization application. ParaView users can quickly build visualizations to analyze their data using qualitative and quantitative techniques. The data exploration can be done interactively in 3D or programmatically using ParaView's batch processing capabilities. The results(such as VTK files) in this project are expected to be visualized using ParaView.

Including the following libraries:

1. Pytorch (1.1.0)

   PyTorch is a Python package that provides two high-level features:Tensor computation (like NumPy) with strong GPU acceleration;Deep neural networks built on a tape-based auto-grad system [30].

   It is a widely used Python based machine learning framewrok. The machine learning related parts would be based on PyTorch.

2. SciPy (v1.3.0)

   SciPy is a free and open-source Python library used for scientific computing and technical computing. SciPy contains modules for optimization, linear algebra, integration, interpolation, special functions, FFT, signal and image processing, ODE solvers and other tasks common in science and engineering [31].

3. NumPy (v1.16.4)

   NumPy is the fundamental package for scientific computing with Python. It contains among other things: a powerful N-dimensional array object; sophisticated (broadcasting) functions; tools for integrating C/C++ and Fortran code; useful linear algebra, Fourier transform, and random number capabilities [32].

4. ParticleModule (2019)

   A python module for driving Lagrangian particles via vtk formatted data with collisions [33].

   The particle module would used to generate PIV-like data in this project. The PIV-like data would used to train the neural network, which is a comparison with that trained by direct CFD data.

3.4 Plan Schedule

Preparation: 1st Jun - 28th Jun

1. Literature review and research plan.

   Read cutting-edge papers of peer reviewed journals and conferences about machine learning techniques applied in fluid dynamics. Get an overview of the field and select sensible methods from papers. Summarize reading and thinking outcomes, finish literature review and research plan.

2. Install software and libraries

   Install all other required software and libraries. Make sure there is no technical problem from installation or platform issues.

3. Get familiar with Firedrake and Thetis

   Get familiar with Firedrake and Thetis. Start from the tutorial and run some examples. Understanding the math principles of the underlying Finite Element Methods(FEM) and shallow water equations as well as the physics scenarios. Learn how to model and solve shallow water problems using these two frameworks.

4. Simulate lab experiment setting

   Because the lab data would be used as benchmarks, it is supposed to use the same setting in the simulations. The setting include geometry scale, Reynolds number, shallowness etc.

Develop: 29th Jun - 30th Jul

1. Start to implement data-driven methods

   Generate synthetic data as the training set both spatially and temporally according to the required features and labels. Using Thetis, we can generate large amount of high-fidelity simulation data.

   As for the data, there are two options:

   - Direct CFD simulation results

   It is easy to get access to all kinds of physics properties(such as pressure field, velocity field etc.) from CFD results. But when considering real world data (such as sensor measurements, drone taken images etc), there is no simulation results but only raw data. If the neural network is trained fully on CFD results, it may not able to make expected predictions when given raw real world data.

   - PIV post-processing results

   By using Lagrange particle methods [33], the raw real world data could be invert to synthetic PIV pictures which contains flows with enough particles. Then we could use PIV post-processing methods to get the physics properties.

   It is hard to say which kind of data is better for training neural network. So it would be interesting to compare the performance and adjust the portion of these two kinds of data.

2. Training and tuning

   Train the neural network using the data set with different strategies. The possible choices can be adjusting the portion of two kinds of data in training set, using spatial data only or spatial-temporal data etc. Tune the hyper-parameters to make the model perform well enough on validation set and lab benchmark data.

3. Start writing the final report

   Write down the work records every work days. Summarize the results and start to write the final report.

Apply: 30th Jul - 19th Aug

1. Test the model using real world data

   Prepossess the new unseen real world data (such as drone taken images, and there might be some image recover work to do because the images may distort due to the different photograph angles). Test the performance of the model and evaluate the generalization ability.

2. Explore the relationship between traditional and data-driven methods

   Due to the low interpret ability of data-driven method, it could not be a fully surrogate model for traditional CFD method right now. It would be valuable to explore the relationship between the traditional numerical methods. Possibly try to propose a more robust method combing both two more sensibly (such as Physics-Informed Neural Network by adding physics laws as prior knowledge).

Summary: 20th Aug - 30th Aug

1. Collect results and draw conclusion

   Analyze the results of all the experiments and make the final conclusion of this project.

2. Finish the rest of final report

   Make the final summary and finish the rest of final report. Get suggestions from supervisor and iterate the report to a satisfied state.

3. Prepare for the final presentation

   Make PPT or poster to show the most lighten ideas of the project. Practice and mock the presentation.

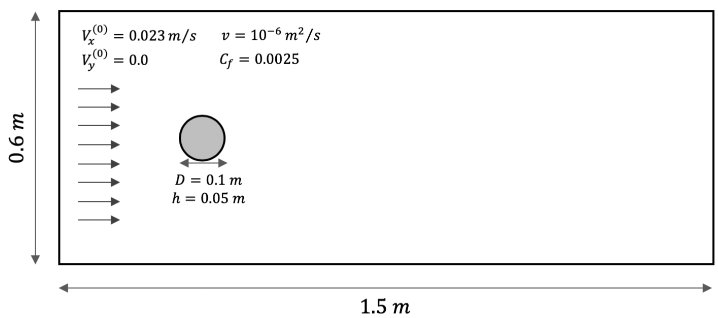


Figure 2: Experiment setting scheme

3.5 Progress to date

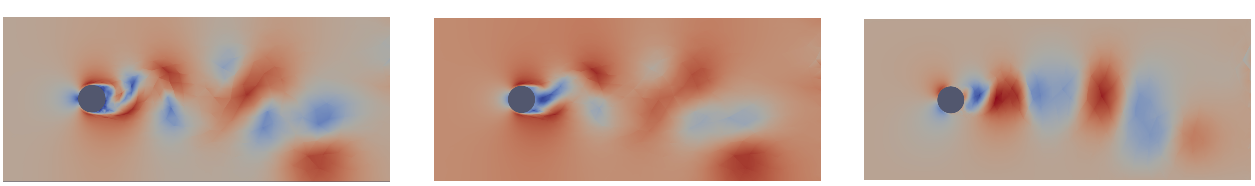
Up to now, the tasks in the Preparation part of the plan have been done. Some current results are showed below.

Shallow water flow past a cylinder

The figure 3 shows the velocity field of a shallow water flow past a cylinder problem. The results are calculated by codes based on Thetis. The parameters and geometry setting(which is showed in figure 2) approximates that of the lab experiments preparing for further comparison.

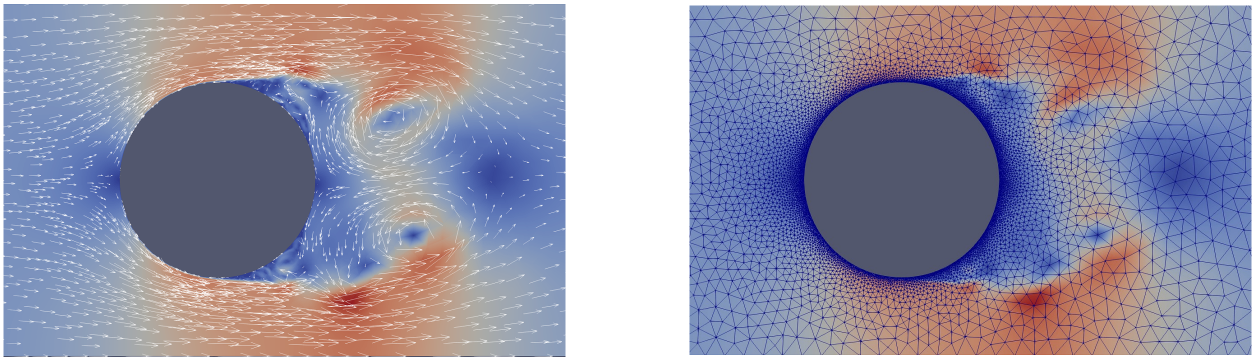
PIV-like data and direct CFD data

The figure 4 shows another high fidelity simulation result with different setting. The left image shows a PIV-like data(the data here is calculated using ParaView with direct CFD data for display purpose, the further real data would be calculated using the Particle Module). The arrows in the image are expected to represent the velocity of different particles in PIV method. The right image shows a velocity field results with mesh calculated by direct CFD simulation.



(a) Velocity Magnitude (b) Velocity-x (c) Velocity-y

Figure 3: Velocity field of a shallow water flow past cylinder scenario



(a) PIV-like data (b) Direct CFD data

Figure 4: Sample of PIV-like data and direct CFD data

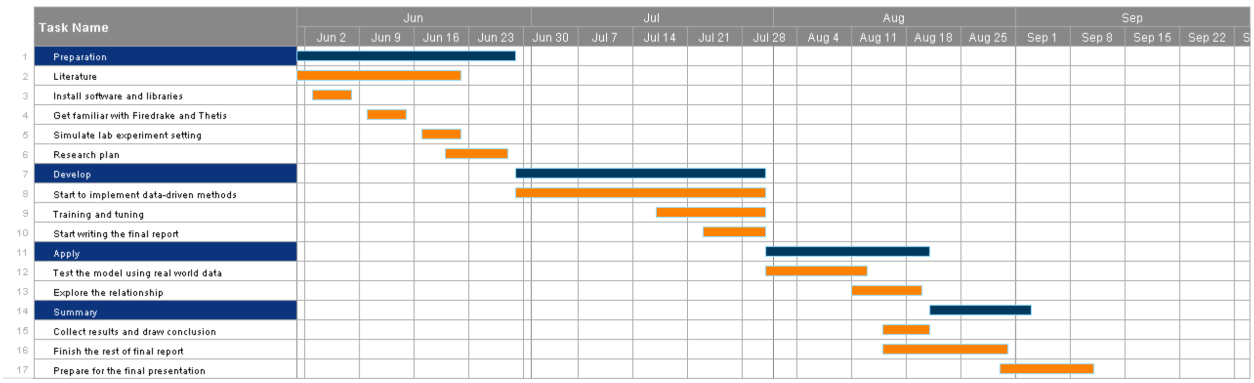


Figure 4: Research Plan Schedule

References

[1] R. du Feu, S. Funke, S. Kramer, J. Hill, and M. Piggott, "The trade-off between tidal-turbine array yield and environmental impact: a maximum entropy modelling approach," *Renewable Energy*, pp. 390–403, 2019.

[2] W. Pan, S. C. Kramer, and M. D. Piggott, "Multi-layer non-hydrostatic free surface modelling using the discontinuous galerkin method," *Ocean Modelling*, vol. 134, pp. 68–83, 2019.

[3] J. Rabault, J. Kolaas, and A. Jensen, "Performing particle image velocimetry using artificial neural networks: a proof-of-concept," *Measurement Science and Technology*, vol. 28, p. 125301, 2017.

[4] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," *Advances in neural information processing systems*, pp. 1097–1105, 2012.

[5] A. Graves, A. rahman Mohamed, and G. Hinton, "Speech recognition with deep recurrent neural networks," *IEEE international conference on acoustics, speech and signal processing*, pp. 6645–6649, 2013.

[6] H. Sak, A. Senior, and F. Beaufays, "Long short-term memory recurrent neural network architectures for large scale acoustic modeling," *Google*, 2014.

[7] Y.LeCun, Y.Bengio, and G.Hinton, "Deep learning," *Nature*, vol. 521, p. 436–444, 2015. [Online]. Available: http://dx.doi.org/10.1038/nature14539

[8] I.Goodfellow, Y.Bengio, and A.Courville, *Deep learning*. MIT Press, 2016. [Online]. Available: http://www.deeplearningbook.org

[9] S. L. Brunton, B. R. Noack, and P. Koumoutsakos, "Machine learning for fluid mechanics," *arXiv:physics*, 2019.

[10] N. Erichson, L. Mathelin, Z. Yao, S. Brunton, M. Mahoney, and J. Nathan Kutz, "Shallow learning for fluid flow reconstruction with limited sensors and limited data," *arXiv preprint*, vol. arXiv:1902.07358, 2019.

[11] T. P. Miyanawalaa and R. K. Jaimana, "An efficient deep learning technique for the navier-stokes equations: Application to unsteady wake flow dynamics," *arXiv preprint*, vol. arXiv:1710.09099, 2018.

[12] B. Kim and T. Günther, "Robust reference frame extraction from unsteady 2d vector fields with convolutional neural networks," *Eurographics Conference on Visualization (EuroVis) 2019*, vol. 38, p. Number 3, 2019.

[13] S. Lee and D. You, "Data-driven prediction of unsteady flow over a circular cylinder using deep learning," *arXiv preprint*, vol. arXiv:1804.06076, 2018.

[14] T. Bolton and L. Zanna, "Applications of deep learning to ocean data inference and subgrid parameterization," *Journal of Advances in Modeling Earth Systems*, vol. 11, pp. 376–399, 2019.

[15] K. Franz, R. Roscher, A. Milioto, S. Wenzel, and J. Kusche, "Ocean eddy identification and tracking using neural networks," *Poster at IEEE International Geoscience and Remote Sensing Symposium*, 2018.

[16] Z.Wang, D. Xiao, F. Fang, R. Govindan, C. C. Pain, and Y. Guo, "Model identification of reduced order fluid dynamics systems using deep learning," *Int. J. Numer. Methods Fluids*, vol. 86(4), p. 255–268, 2017.

[17] R. Gurka, A. Liberzon, and G. Hetsroni, "Pod of vorticity fields: A method for spatial characterization of coherent structures," *International Journal of Heat and Fluid Flow*, vol. 27, p. 416–423, 2006.

[18] D. Xiao, C. Heaney, L. Mottet, F. Fang, W. Lin, I. Navon, Y. Guo, O. Matar, A. Robins, and C. Pain, "A reduced order model for turbulent flows in the urban environment using machine learning," *Build. Environ.*, vol. 148, p. 323–337, 2019.

[19] Z. Zhang, X. dong Song, S. ran Ye, Y. wei Wang, C. guang Huang, and Y. ran An andYao-song Chen, "Application of deep learning method to reynolds stress models of channel flow based on reduced-order modeling of dns data," *Journal of Hydrodynamics*, vol. 31(1), pp. 58–75, 2019.

[20] R. King, O. Hennigh, A. Mohan, and M. Chertkov, "From deep to physics-informed learning of turbulence: Diagnostics," *arXiv preprint*, vol. arXiv:1810.07785, 2018.

[21] E. J. Parish and K. Duraisamy, "A paradigm for data-driven predictive model- ing using field inversion and machine learning," *Journal of Computational Physics*, vol. 305, pp. 758–774, 2016.

[22] J. N. Kutz, "Deep learning in fluid dynamics," *Journal of Fluid Mechanics*, vol. 814, pp. 1–4, 2017.

[23] M. Raissi, P. Perdikaris, and G. Karniadakis, "Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations," *Journal of Computational Physics*, vol. 378, pp. 686–707, 2019.

[24] J. Ling, A. Kurzawski, and J. Templeton, "Reynolds averaged turbulence modelling using deep neural networks with embedded invariance," *Journal of Fluid Mechanics*, vol. 807, pp. 155–166, 2016.

[25] N. B. Erichson, M. Muehlebach, and M. W. Mahoney, "Physics-informed autoencoders for lyapunov-stable fluid flow prediction," *arXiv preprint*, vol. arXiv:905.10866, 2019.

[26] R. Jean, K. Miroslav, and J. Atle, "Artificial neural networks trained through deep reinforcement learning discover control strategies for active flow control," *arXiv preprint*, vol. arXiv:1808.07664, 2018.

[27] J. Morton, A. Jameson, M. J. Kochenderfer, and F. Witherden, "Deep dynamical modeling and control of unsteady fluid flows," *Advances in Neural Information Processing Systems*, p. 9258–9268, 2018.

[28] Firedrake-Community. (2019) Firedrake. [Online]. Available: https://www.firedrakeproject.org/

[29] Thetis-Community. (2019) Thetis. [Online]. Available: https://thetisproject.org/

[30] PyTorch-Community. (2019) Pytorch. [Online]. Available: https://pytorch.org

[31] SciPy-Community. (2019) Scipy. [Online]. Available: https://www.scipy.org

[32] Numpy-Community. (2019) Numpy. [Online]. Available: https://www.numpy.org

[33] J. Percival. (2019) Particle module. [Online]. Available: https://github.com/jrper/ParticleModule