Machine Learning - Project 2

Rizk Kevin, Poulain- -Auzéau Louis
EPFL, Switzerland

***Abstract*—In this paper, we explore how to reconstruct the air flow around a cylinder using only limited information. We use a new extension of the PINN (Physic Informed Neural Network), created in 2021 that incorporates Fourier modes. First, we try to recover the results of [1] and then we use new sets of data coming from the TOPO laboratory at EPFL. In particular, we try to explore the robustness of the model with respect to increasing Reynolds number, when the flow becomes turbulent. Indeed, most research paper consider laminar flows (with low Reynolds number), when in reality, flows are more often turbulent. We used EPFL clusters to train our model.**

I. Introduction

A. Problematic

One of the main challenges in the engineering world, is to be able to reconstruct a flow around an object using limited number of information for example from a limited numbers of sensors and faster than numerical solutions which can be take very long time. In particular, the problem was, given limited information on pressure and speed of a flow given around a cylinder, to use a special neural network network to reconstruct the whole air flow around the cylinder.

B. Navier stokes equation

First, we recall that the equations that governs fluid dynamics are the Navier-Stokes equations. Here is one of a 2-dimension version of the equation that we are interested in.

$$\frac{\partial \mathbf{u}}{\partial t} + \mathbf{u} \cdot (\nabla \mathbf{u}) + \nabla p - \frac{1}{Re} \nabla^2 \mathbf{u} = 0$$
$$\nabla \cdot \mathbf{u} = 0$$

with Re corresponding to the Reynolds number. It is an a dimensionless number which measures the ratio of inertial forces to viscous forces in the fluid. One specificity of this number, is that normally a low Re number will determine a laminar flow, but for a big Re number, we normally get a turbulent flow and which is more chaotic. So the Re number is really crucial.

C. Goal

The goal is to incorporate in our Neural network some physical information, or something related to our PDE (partial differential equation), so we can have a smarter Neural network for this type of applications. Two solutions to this problem are the following type of Neural networks.

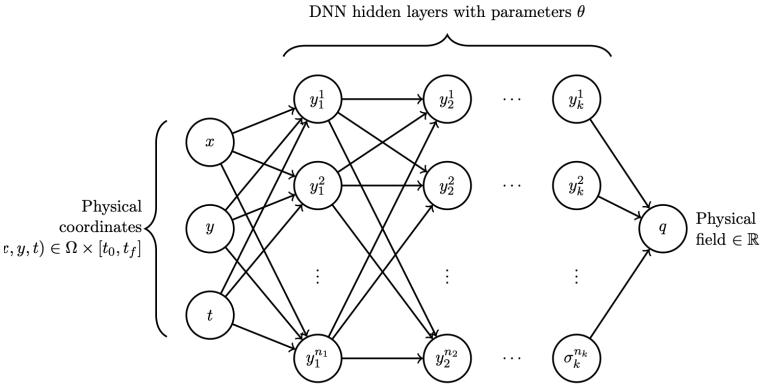
II. PINN and ModalPinn Model

A. PINN model

The Physical informed neural network is a recent neural network framework (cf [2]) that takes into account the physics of the data. Moreover it takes into account some physical information, like the derivatives or the gradient. In fact the loss function, which can be found in more details in [1] calculate the mean square error of how far we are from the boundary equations and a loss of how far we are from the from the expected PDE solution. And so by optimisation, and minimizing the error, the model should converge to something which is physically more accurate and close to solving the PDE. This is better for the purpose of this type of application.

B. The architecture of PINN model

As seen in the below figure, we can see that the PINN architecture is a fully connected neural networks which takes as input the physical inputs of space and time and output the field q(x,y,t) which can for example can be the velocity or the pressure.



C. Modal PINN model

1) Idea: One thing that can sometimes be found in the physical world is the periodicity of some some quantities. It would be good if we could take into account this periodicity and incorporate it in our Neural network architecture, to have better and faster results. In more details for some field q depending on time , if it is periodic then by Fourier analysis, we have that:

$$q(x, y, t) = \sum_{k=0}^{\infty} \hat{q}_k(x, y) e^{ik\omega_0 t} + constant \tag{1}$$

with ω_0 the fundamental frequency and i the imaginary number. So now the idea of [1], is to have for the output of the neural network, not directly q but some of the modes $\hat{q}_k$. From this modes, we then want to reconstruct the field using a truncated sum of (1) to reconstruct all the field.

So now we have a neural network that will output the modes. Note that $\hat{q}_k$ is independent of the time and this can be crucial, as we can gain time for training and force the periodicity condition in our solution.

Note that, in our case and in the paper data, we indeed have periodicity in our physical fields.

2) The two types of Modal PINN model: As $\hat{q}_k$ is independent of time, we can define two loss functions which changes the way the model is trained and how $\hat{q}_k$ is found. Here are the different loss equations that can be trained with the ModalPINN Model. For more details for the exact formula of the two loss function, they can be found in the paper [1].

(a) Physical equations : Here we will use the time for calculating the physical loss and so being able to learn the modes of the modalPINN. This method is normally more accurate than Modal equations as we need a sampling of time in addition to the sampling of space. As the dimension of the space augments, more data points are needed for having a good result, which can

become computationally heavier. The architecture can be found in Figure1 which correspond to (a)

(b) Modal equations : As seen before, as the modes are independent from time, we can choose the loss equation for training the modes such that it is independent of time. For the mathematics behind it (Fourier analysis decomposition) they can be found in the paper [1]. We omit from the training the time sampling for training, which concretely means that our space has less dimensions and so we can have faster training and convergence. But normally it will be less accurate. The architecture of the neural network correspond to the (b) of Figure1

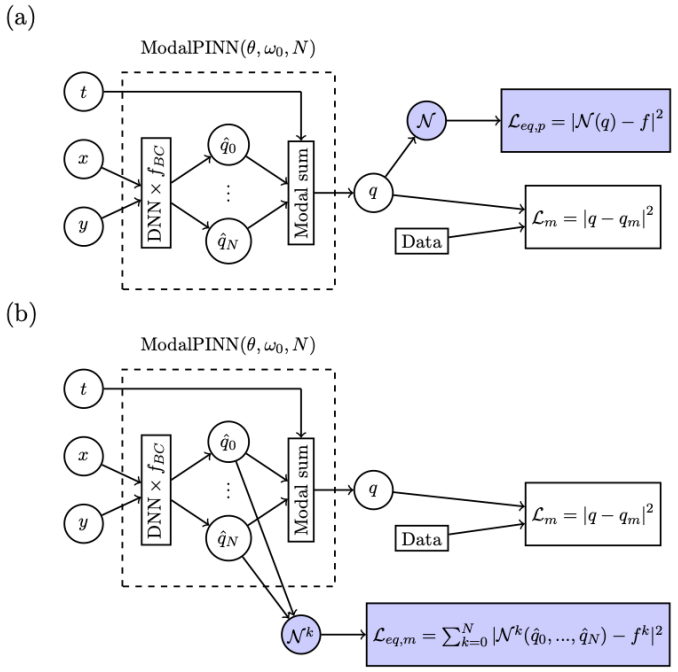


Figure 1. The 2 architecture of Modal PINN models

III. Training setup and optimization

Here we present what was used for all our training, and the optimizer used for the neural network.

1) Cluster: All our training were done in the EPFL cluster, more specifically using the Izar cluster and allocating 50 GB of RAM ,and using the available GPUs.

2) optimization and Time: For the optimization of the loss as in the paper, we used first L-BFGS which stops when we arrive at a fixed number of iterations or when the difference of the loss between two iterations became a lot smaller. And then we used for the remaining time Adam optimization. We used a learning rate equal to 10^{-5}. Changing the training rate didn't have a big impact in our training, so we kept this learning rate. Adam training is also subject to a maximum number of iterations of 10^5 and to a time limit. Indeed, after 10 hours of overall optimisation, the program was set to stop the training and move on to the validation. All the presented here were run for about 10 hours. We ran some jobs with less time for testing and understanding purpose.

3) neural network number of layers: As we needed to deduce the velocities (u,v) and the pressure p, 3 neural networks were constructed (one for each physical field). Each network has an architecture looking like $[2, 80, 80, 4]$. For instance this mean each node in the first layer in linked to 80 nodes and so on. 25 neurons are used per layer and per mode and we used two affine transformations and one non linear, using $\tanh$ as activation function.

A. Number of points and validation

Some of the data was used to train and other data points was used to validate. We used the same validation and cutting point as in the paper we followed [1]. In fact, we used mostly 5000 points to be used during optimisation and 8000 points for equations evaluation during optimisation. Indeed, this proved to be a good trade off between computational cost and results. For the validation, we use ten times more points for each type of point.

IV. Reproducing Paper results

We started by trying to reproduce the paper results, before doing some changes relevant to our applications.

A. Data Set

The data set corresponds to a laminar flow with a Reynolds number of 100 around a cylinder centered at 0 and of diameter $D = 1$ m. It consists in the gathering of information about speed and pressure on 82872 points in the sampling domain $[-4D, 8D] \times [-4D, 4D]$ and at 201 time steps starting at 400 s. It has been provided by [3].

This problem is very complex to solve, mainly because of the number of hyper parameters that can be used and tuned. Few of them are the mesh generation, the sparsity of data, the loss function used, the number of sensors or the location of the points from which we want to get information. We present some of them next and how we used them.

B. Physical vs Modal Equations

As told before, Modal Equations need no sampling of the time to calculate the nodes and so is trained faster, but is less accurate. And as we wanted to estimate a flow using limited amount of information, we think that the input of time is crucial, as without it, we would not have enough information to train and find good results, as the input of time is crucial for finding the solution of a PDE. And so, we tested the difference between Physical and Modal Equations. And Physical Equation performed a lot better than Modal Equation when training. For example: training using sparse data, 2 sampling zones (cd IV-C), with 30 sensors in the cylinder, and 3 modes we get a validation error of the order of 10^{-3} for the physical equations, and of order 10^{-2} for modal equations. The difference can be noted also when we do a visualisation of the physical quantities.

C. Mesh generation

There are two method to produce a mesh of the sampling domain. One is to use a uniform grid mesh and the other consists in using 80% close to the cylinder and 20% for the rest of the sampling domain. The first one is more easy to use since it doesn't require anything else than the geometry of the domain. On the other hand, it is more likely to produce less accurate results. Indeed, the physical equations prescribe the flow to have a certain behaviour around the cylinder whereas far from the cylinder, the flow won't be affected by the presence of the cylinder or not. We tried both meshes design and, as expected we obtained better results with the second option, For example 2-zone sampling gave very accurate result in Figure 4, as the visualisation with uniform sampling (not shown) is less accurate around the cylinder.

D. Number of modes

In the ModalPINN model, we can use up to 5 modes for the truncated Fourier series. We noticed that increasing the number to more than 3 was introducing more noise than precision for some runs and not much precision for others. Indeed, the higher frequencies don't bring more information as the majority is contained in lower modes. Also using only one mode was not enough to have a good approximation and so the errors were huge. In the end, we ran our jobs using 2 or 3 modes only.

E. Forces

A very important quantity to estimate the forces acting on the cylinder, the lift and drag force, in fact we have a function in our code , using Monte Carlo method to estimate an integral, and being able to estimate those forces which respect time. Those function can be plotted over time, visualize them to have a better understanding on what is happening. We obtained figure 2 showing the evolution along time for an entry speed of 0.08 m/s. This has to be compared with the real forces, that we didn't have time to plot.

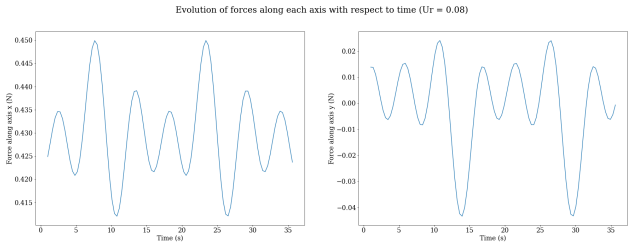


Figure 2. Evolution of forces, Ur = 0.08

F. Results

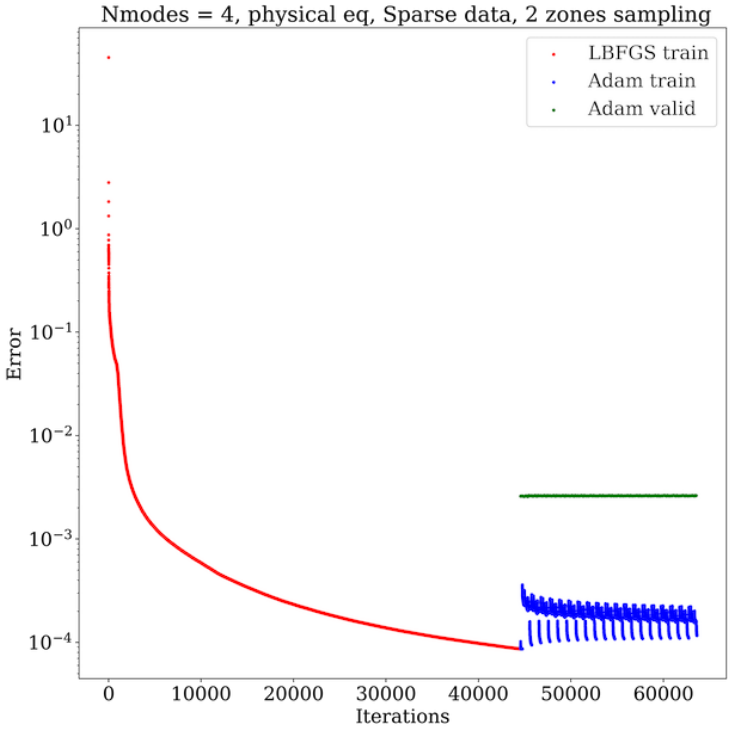


Figure 3. Losses evolution, using 4 sensors

The figure 3 presents the evolution of the two losses during the training and also the loss on the validation data. For this job we used

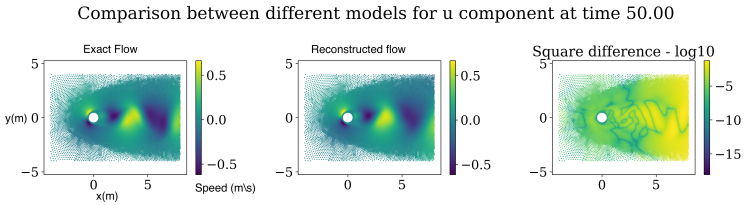


Figure 4. Visualization of velocity field when using 6 sensors , 2 zone sampling, physical equations and 3 Modes

As we can see on figure 4, the approximation is very good. There are still some errors far from the cylinder as we can see on the third image, but this is less important for real applications.

G. Tuning and experimentation Loss function

As the lab's data are different then the we want to adapt a little bit the paper code and also the conditions, to see if they still works before trying to see if it will fare well with the Lab's data.

1) Reducing the number of sensors: So in the paper, they used 30 sensors on the cylinder, as for our applications, we needed to use a smaller amount of sensors to see if we still get good results. Even if the validation loss error got a little bit smaller (from 10^-3 to about $5 * 10^-2$. For example in the Figure3 , we used 4 sensors to try and re-estimate the data Set, and we see that we still got good results, and the flow of the velocity is well approximated as seen in Figure4.

2) New loss function: both equations: Since the model is an extension of the PINN model, the neural network can be set up to optimize also on the physical equations or on the modals equations. And we also experimented and tried to use in fact both equations to find the parameters. Although it did imporve our results, so improvement were not signficant compared to using only physical equations, so we finally did not find it interesting

3) Noise: as noise is a present in the real world, it is crucial to see how the code fare with respect noise. As seen in the paper, the model fare well with normal unbiased noise, the loss error is in the range $[10^-3, 10^-1]$ depending in the standard deviation. But as noise is not always normal or unbiased in the real world, adding a baised and non normal noise could be a good idea to test the robustness of the architecure.

V. Labs' data

A. The Data set

There are two types of data sets. The first one consist in eight data sets corresponding to a much more turbulent flow (Reynolds numbers or order of magnitude 10^5). Each of the 16 data sets consists in information on pressure and speed from the simulation of a flow around a cylinder of diameter $D = 0.055m$. There are 18077 points in the sampling domain $[-4D, 8D] \times [-4D, 4D]$ and 300 time steps.The second one contains eight different sets corresponding to a laminar flow (Reynolds numbers of order of magnitude 10^3 with increasing speeds ranging from 0.03 to $0.2m/s$. Next, we present our results and we compare the two types of data sets.

B. Adapting the data

In accordance with [3], the first data set had been previously normalised in the following way: distances are divided by D, speeds by U_r, the entry speed (speed of the flow at the inlet), time is divided by $\frac{U_r}{D}$ and pressure by ρU_r^2. On this neural network, variables need to

be dimensionless in order for equations to work. Hence, we applied the same normalisation.

C. Results

So the first job we ran on this data set , we used 2 zone sampling, with 6 sensor on the cylinder, using physical equations and running our code for 10 hours, the results we got can be found on this graph. As we see the validation loss is very bad , it is around of the 10^4, and the visualisation and the forces were really off than what was predicted. And even by changing some small parameters like number of modes used, sampling type, number of cylinder, none of them had an improvement on our results. And so this was some of our hypothesis.

D. Hypothesis and Tuning

As we see the results were not very good, so we had a lot of hypothesis and we tested some of our hypothesis. The main problem resided with the fact that the Re was high ans so we were no more with the laminar regime, but more in the turbulent regime.

- Hypothesis 1: One hypothesis is to use another PDE than what we used to change our loss function, but after some more consideration the Navier-Stokes equations still seemed the best choice as they are normally the most appropriate equations to use.
- Hypothesis 2 : When we are working with high RE, In fact higher frequencies are present in the Fourier decomposition, and so one idea was to to expand the range of frequencies we use in the ModalPINN to see if we have better results.But after doing this, the results did not improve at all
- Hypothesis 3 : As we have a turbulent flow, and not a laminar one as the RE is high, we can in a way loose some of the periodicity in some areas around the cylinder, which makes ModalPinn more imperfect . So some ideas that can be applied are the following 2 hypothesis or ideas.
- Hypothesis 3.a: As PINN (classical) did not assume the periodicity of the physical fields. One of our hypothesis was to assume that if we run the classical PINN on the data set , maybe this can improve the results. But this is did not at all improve our result.
- Idea 3.b: Another idea is to do a better mathematical analysis on our of turbulent flows or chaotic events. And if we can decompose a chaotic field with respect some other or additional quantities which respect the modes in the Fourier analysis. We could change the loss functions and add some intermediate outputs like in ModalPINN for the modes. But this need a careful analysis behind the mathematics of turbulent flows.

E. Second data Set

So knowing how difficult is to predict a flow when there is a high Re, we were provided with data set which has lower range of Re from 100 to 700, to see how the Modal was doing by changing the Re value. Figure 6 shows the evolution of the loss with respect to Re. We can see that the model is doing less good as the Re increases. Also a comparison of our three components to the ground truth is presented in figure 5. The associated Reynolds number is 265.8610. We can see that the results are much better compared to those obtained with high Reynolds numbers (see section V-C).

VI. Conclusion and Discussion

To conclude, modal PINN is a good neural network architecture for periodic events, as it uses the advantages of PINN, while enforcing the periodicity on the results. For example, for laminar flows, it provided

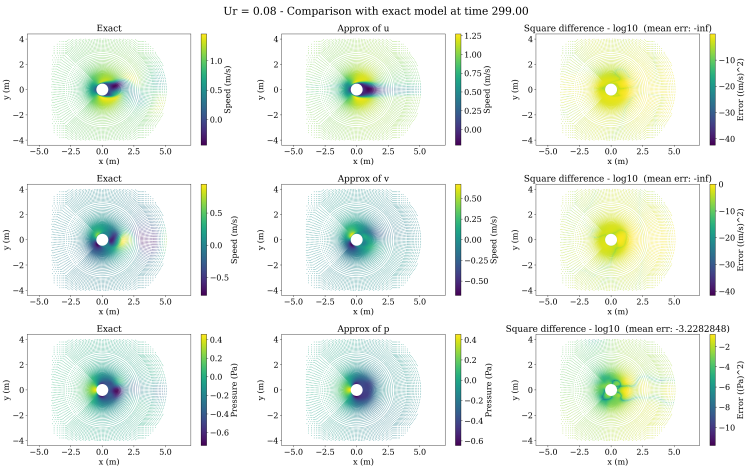


Figure 5. Visualization of the three fields for an entry speed of 0.08 m/s when using 6 sensors , 2 zone sampling, both equations and 2 modes

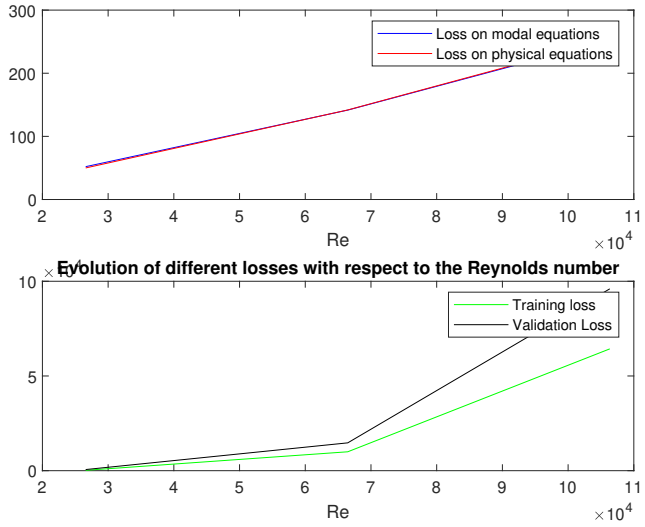


Figure 6. Evolution of loss functions with respect to increasing Re

good results and good reconstruction of flows, in the paper and in the lab's data. But as we had higher Re's number, the error got bigger and the reconstruction was worse, some hypothesis were provided. One good direction, is to adapt the modes and change a little bit the modal depending on a more careful mathematical analysis of turbulent flows. Another direction, where we can go, is to use the ideas of auto-encoders and recurrent neural networks which take advantage of sequential data and time series for example. For example, this paper [4] used those idea to reconstruct flow around a cylinder using a limited number of sensors, but the Re consider were between 30 and 300. We could also follows the lead of recent papers which try to reconstruct turbulent flows which builds up on PINN models for example [5]. Also, we could improve a bit the code so that it outputs the real forces also (this was launched but the job didn't had time to finish), for us to compare with the approximation found in Figure 2.

References

[1] Gaetan Raynaud, Sebastien Houde, and Frederick P. Gosselin. Modalpinn: an extension of physics-informed neural networks with enforced truncated fourier decomposition for periodic flow reconstruction using a limited number of imperfect sensors, 2021.

[2] Maziar Raissi, Paris Perdikaris, and George E. Karniadakis. Physics informed deep learning (part I): data-driven solutions of nonlinear partial differential equations. *CoRR*, abs/1711.10561, 2017.

[3] Mouad Boudina, Frédérick P. Gosselin, and Stéphane Étienne. Vortex-induced vibrations: a soft coral feeding strategy? *Journal of Fluid Mechanics*, 916:A50, 2021.

[4] Yash Kumar, Pranav Bahl, and Souvik Chakraborty. State estimation with limited sensors – a deep learning based approach, 2021.

[5] Chen Cheng, Peng-Fei Xu, Yong-Zheng Li, and Guang-Tao Zhang. Deep learning based on pinn for solving 2 d0f vortex induced vibration of cylinder with high reynolds number, 2021.