Learning Reduced Order Mappings of Navier-Stokes

An Investigation of Generalization on the Viscosity Parameter

Amund Kiste[1]

Supervisor(s): Dr. David Tax[1]**, Mahdi Naderibeni**[1]

[1]EEMCS, Delft University of Technology, The Netherlands

A Thesis Submitted to EEMCS Faculty Delft University of Technology,
In Partial Fulfilment of the Requirements
For the Bachelor of Computer Science and Engineering
January 28, 2024

Name of the student: Amund Kiste
Final project course: CSE3000 Research Project
Thesis committee: Dr. David Tax, Mahdi Naderibeni, Nergis Tömen

Abstract

Solving Partial Differential Equations (PDEs) in engineering such as Navier-Stokes is incredibly computationally expensive and complex. Without analytical solutions, numerical solutions can take ages to simulate at great expense. In order to reduce this cost, neural networks may be used to compute approximations of the solution for use during engineering processes. PCA-net is a neural network approach that reduces the dimensionality of the input and output data for PDEs in order to allow mapping from a high-dimensional input and output function with a fully connected neural network through the use of Principal Component Analysis (PCA). In this paper, PCA-net is applied to Navier-Stokes with varying viscosities to test the generalization of PCA-net on viscosity parameters. Training is done on four discrete viscosities, while testing is done on continuous viscosities, extrapolating and interpolating around the training set. Results shows good performance on low viscosities, both with interpolation and extrapolation. Mid-to-high viscosity interpolation shows lesser performance, with high viscosity extrapolation diverging to great error. Omitting high viscosities, performance over varying viscosities is close to that shown by previous research.

1 Introduction

Computationally expensive and difficult to solve, partial differential equations (PDEs) lie at the foot of many science and engineering tasks. Oftentimes, the same PDEs have to be evaluated repeatedly during an engineering process with extreme computational cost to ensure accuracy. One of the most complex and expensive common PDEs to evaluate, the Navier-Stokes equations describe the motion of viscous fluids. They relate the changes in velocity, pressure, temperature, and density of the fluid over time, subject to a forcing function.

The Navier-Stokes equations have two properties that can be credited with making them exceedingly difficult to solve numerically or analytically. The first of these properties, non-linearity, means that, with some minor exceptions, the Navier-Stokes equations can not be solved analytically. On the other hand, numerical solutions of the Navier-Stokes equations are generally turbulent, causing general instability of the solution. In order to avoid instability due to turbulence, very fine meshes may necessary, at massive computational cost.

Due to this cost, research has been done into simplifying the process of evaluating PDEs through the use of neural networks. One method used for the evaluation of PDEs with neural networks is through dimensionality reduction with principal component analysis (PCA) [1; 2]. By reducing the dimensionality with PCA, neural nets can be trained on a lower-dimension data-set [1; 2]. Once trained, a PDE can then be evaluated with this model, instead of solving it in a computationally expensive numeric simulation [1; 2]. In theory, the major benefit of this type of evaluation is the transfer of computational workload away from being necessary every time the PDE is evaluated, and instead taking the majority of it as overhead during the training process [1; 2].

As one of the major governing set of equations for many fields such as aerodynamics and hydrodynamics the Navier-Stokes equations are fundamental in computational flow dynamics (CFD). This process requires large computational resources and must often be repeated during each iteration of a design. If a neural network proves effective at solving Navier-Stokes equations it may be viable to use this as a faster way to evaluate Navier-Stokes at each iteration. Thus, the possibility of evaluating Navier-Stokes accurately and effectively with PCA-based methods should be explored.

In this paper, the performance of PCA based neural networks will be evaluated when mapping from the forcing function and viscosity to the resultant vorticity after 5 seconds in a vorticity-stream formulation of Navier-Stokes. Training is conducted on four discrete viscosities that represent real liquids, as shown in Table 1, with testing attempting to interpolate and extrapolate data for viscosities outside the training data.

The aim of this paper is to investigate to what extent PCA-based neural-network approaches can approximate numerical solutions of the Navier-Stokes equations over different viscosities. The current methods of solving Navier-Stokes and their advantages and drawbacks will be discussed and compared to PCA-based methods.

1.1 Literature Survey

Multiple investigations have been made into ways of solving PDEs using reduced-order neural-networks in order to reduce computational costs.

Ohlberger and Rave [5] summarize the state of Reduced Basis Methods (RBMs) as a method for solving PDEs. RBMs use an approximation space V_N which must be found, generally using greedy methods, to reduce the size of the input set [5]. Unlike PCA based neural nets however, RBMs require prior knowledge of the PDE that is being modeled, reducing their usefulness for arbitrary PDEs [5].

Bhattacharya et al. [1] propose a reduced-order neural-network based upon PCA to learn mappings between Hilbert spaces. PCA based neural networks are established as being mesh-independent, meaning their errors do not increase as the size of the mesh upon which approximation is done is increased, a major advantage for working with large PDEs [1]. This is then further demonstrated to be capable of approximating solutions for parametric PDEs [1].

Testing this neural-network architecture on Poisson, Darcy flow and Burgers' equation show that PCA-based neural networks can adequately approximate many different PDEs [1]. Bhattacharya et al. [1] propose that these PCA-based neural-networks should further be tested on more challenging PDEs.

V. de Hoop et al. [2] describe three different neural-net implementations based on applications of PCA on the input data, in addition to a Fourier Neural Operator. The simplest of these, PCA-net describes an implementation of a network using PCA at the input and output layer to work on reduced-dimension data, the base upon which the neural-network in this paper is built upon.

In V. de Hoop et al. [2] these reduced-order neural-networks are tested on several different PDEs. First, the neural nets are tested on the mapping from the forcing function of the vorticity-stream (ω-ψ) formulation of the Navier-Stokes to the vorticity field after ten seconds [2]. All neural-networks present in the study were able to accurately predict the vorticity fields, with the Fourier Neural Operator performing the best due to the nature of the problem [2]. Further, V. de Hoop et al. [2] use three other test cases, those being Helmholtz equation, a Structural Mechanics problem, and the one-dimensional advection equation. All four networks succeed in predicting these problems as well [2]. Evaluation of each neural network in terms of accuracy versus training and evaluation cost found that the PCA-net performs best per cost of the PCA based neural networks, only beaten by the Fourier Neural Operator in some cases [2].

Kovachi et al. [3] test a series of different neural nets and operators for mapping between function spaces, as applied to PDEs. Using the PCA based neural net (PCANN) as proposed in Bhattacharya et al. [1], Kovachi et al. [3] find that PCANN performs well on Burgers' equation and Darcy flow, outperforming traditional neural nets and fully convolution networks. Kovachi et al. [3] also cite the ability of PCANN to learn only from data, without knowledge of the underlying function, as an advantage over classical reduced basis methods using PCA basis.

An alternative method of solving parametric PDEs through reduced order mappings is proposed by Li et al. [4], who introduce the Fourier Neural Operator (FNO). Working specifically on Navier-Stokes, FNO is capable of learning resolution-invariant solutions in the turbulent regime [4]. Li et al. [4] find that FNO outperforms PCANN with lower error rates on Burgers' Equation and Darcy Flow, but do not test PCANN on Navier-Stokes. Additionally, it is found that FNO scales exponentially with spatial dimension, greatly increasing its cost if used on many-dimensional data, leading to far higher costs than PCA-net [2].

Altogether, the general consensus from the reviewed literature places PCA based neural networks as a cheap and reliable method that is outperformed by more complex methods such as FNO [2; 3; 4]. Nevertheless, PCA-net is still found to be a desirable alternative to FNO thanks to its lower evaluation cost [2].

1.2 Our contribution

The primary contribution of this paper is testing the ability of PCA-net to handle extrapolation and interpolation of Navier-Stokes at varying viscosities.

Expanding upon the work in V. de Hoop et al. [2], this paper evaluates PCA-net on a more complex set of Navier-Stokes problems, including viscosity as an input parameter. This tests the ability of PCA-net to work on more complex problems.

2 Methodology

For evaluating the efficacy of PCA-Based methods to evaluate Navier-Stokes, the method will be based on the works of Martin v. De Hoop et. al. [2].

2.1 Navier-Stokes Equation

For the Navier-Stokes equation, the incompressible version of the equations, specifically the vorticity-stream (ω-ψ) formulation will be used, as shown in Equation 1 [2].

$$\begin{aligned} \frac{\delta\omega}{\delta t} + (u \cdot \nabla)\omega - v\Delta\omega &= f', \\ \omega &= -\Delta\psi, \\ \int_D \psi &= 0, \\ u &= \left(\frac{\delta\psi}{\delta x_2}, -\frac{\delta\psi}{\delta x_1}\right) \end{aligned} \tag{1}$$

As done in v. De Hoop et. al. [2], investigation will also be focused on the mapping from a forcing function f', to a corresponding vorticity field ω, at time $t = T$.

The domain is limited to a two-dimensional periodic domain from 0 to 2π [2]. In order to ensure that the numerical model stays accurate with changing viscosity, the fineness of the system will be increased by training on a grid of 100 by 100 [2].

The forcing function f' is also taken from v. De Hoop et. al. [2], using a centred Gaussian. The initial conditions is fixed and a centred Gaussian generated from the same generator as the forcing functions [2].

2.2 PCA-Net

The architecture used for the neural net for mapping between functions is based upon the PCA-net as described in V. de Hoop et al. [2].

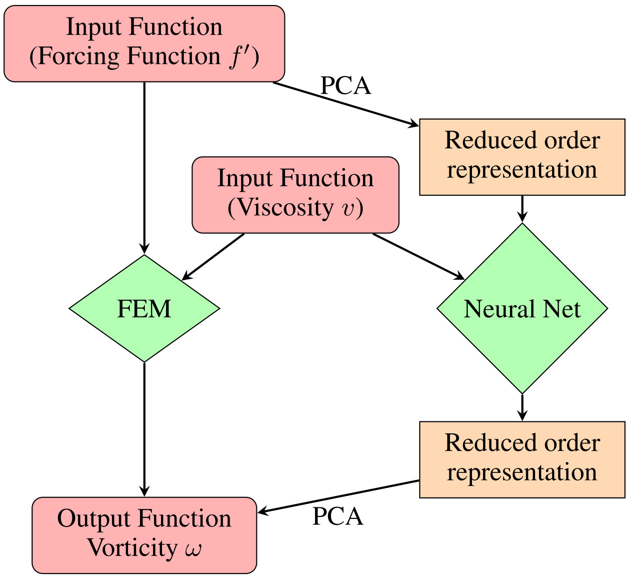
Principal component analysis is done using the singular value decomposition (SVD), in order to reduce the input basis to the neural network. This allows the network to work on a reduced data-set with features that are better correlated, simplifying the training process.

In order to decide how many ranks of the SVD should be preserved, the matrix is sorted and truncated at whichever value preserves at-least 99.99% of the variance. This then provides the basis for PCA.

Following the principal component analysis, the reduced-order data-set is passed through a fully connected neural network. As inputs, the neural network takes the output from the PCA, in addition to the viscosity. This then produces the vectors for the output state, which is then transformed using the reverse PCA to compute the loss. For the activation function, ReLU is used.

As an error metric, the relative error is used. Based on the frobenius norm, defined in Equation 2. The frobenius norm is also used to define the relative amplitude of the vorticity.

$$\text{Error} = \frac{||\text{True} - \text{Predicted}||_F}{||\text{True}||_F}$$
$$||A||_F = \sqrt{\sum_i^m \sum_j^n |a_{ij}|^2} \tag{2}$$



3 Experimental Setup and Results

3.1 Training

In order to create the test and training set, adaptions are made from the data-generation code used by v. De Hoop et. al. [2]. Instead of using a constant viscosity of 0.025, training data will have multiple different viscosities. These viscosities will be taken from the viscosities of some common fluids, with the exception of 10^{-2} for which no common fluid exists, as shown in Table 1.

Table 1: Training Viscosities [6]

Fluid	Absolute Viscosity ($\mathrm{N\,s\,m^{-2}}$)
Water	10^{-3}
None	10^{-2}
Olive Oil	10^{-1}
Glycerol	10^{0}

The test set however, will be a continuous set with viscosities between 10^{-4} and 10^{1}, in order to test the ability of PCA-net to extrapolate and interpolate. These will be generated as a uniform distribution from -4 to 1, and then exponentiated with base 10.

Training was tested using five different seeds to ensure that the model would converge independent of starting conditions. The loss per epoch is shown in Figure 1. This shows that convergence is achieved with a training loss of approximately 0.015 independent of starting seed when trained on discrete viscosity data.

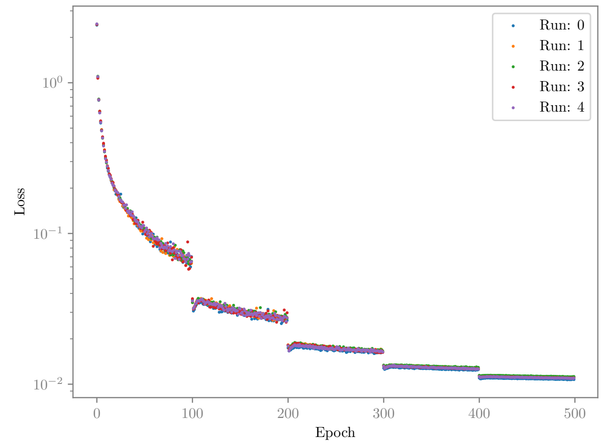


Figure 1: Training Losses dependent on Epoch

3.2 Testing

Evaluating the model performance on the test set displays some interesting properties. Attempts to extrapolate at viscosities above 10^{0} quickly display divergent properties, as shown in Figure 2. Below this threshold however, the relative error stays less than 1. At the training viscosities 10^{0}, 10^{-1}, 10^{-2}, and 10^{-3} errors are the lowest, while errors are greatest between 10^{0} and 10^{-1}, peaking at approximately $10^{-0.5}$. Interestingly enough, errors seem to stay consistently low as viscosities fall below 10^{-2}, where the further reduction of viscosities does not make much of a difference.

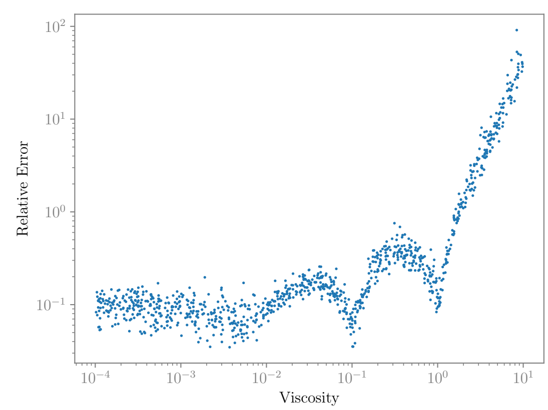


Figure 2: Error as a Function of Viscosity

These errors are shown in Figure 3, which shows the forcing function, true vorticity, predicted vorticity and relative errors. This data shown that even in the high-error interpolated case at a viscosity $10^{-0.505}$, the general shape of the ground truth is preserved, but values are over-exaggerated. This is no longer true for the extrapolated data at $10^{0.924}$, where values are greatly over-exaggerated by a factor of approximately 10, and the shape of the prediction is no longer well-correlated with the true vorticity.

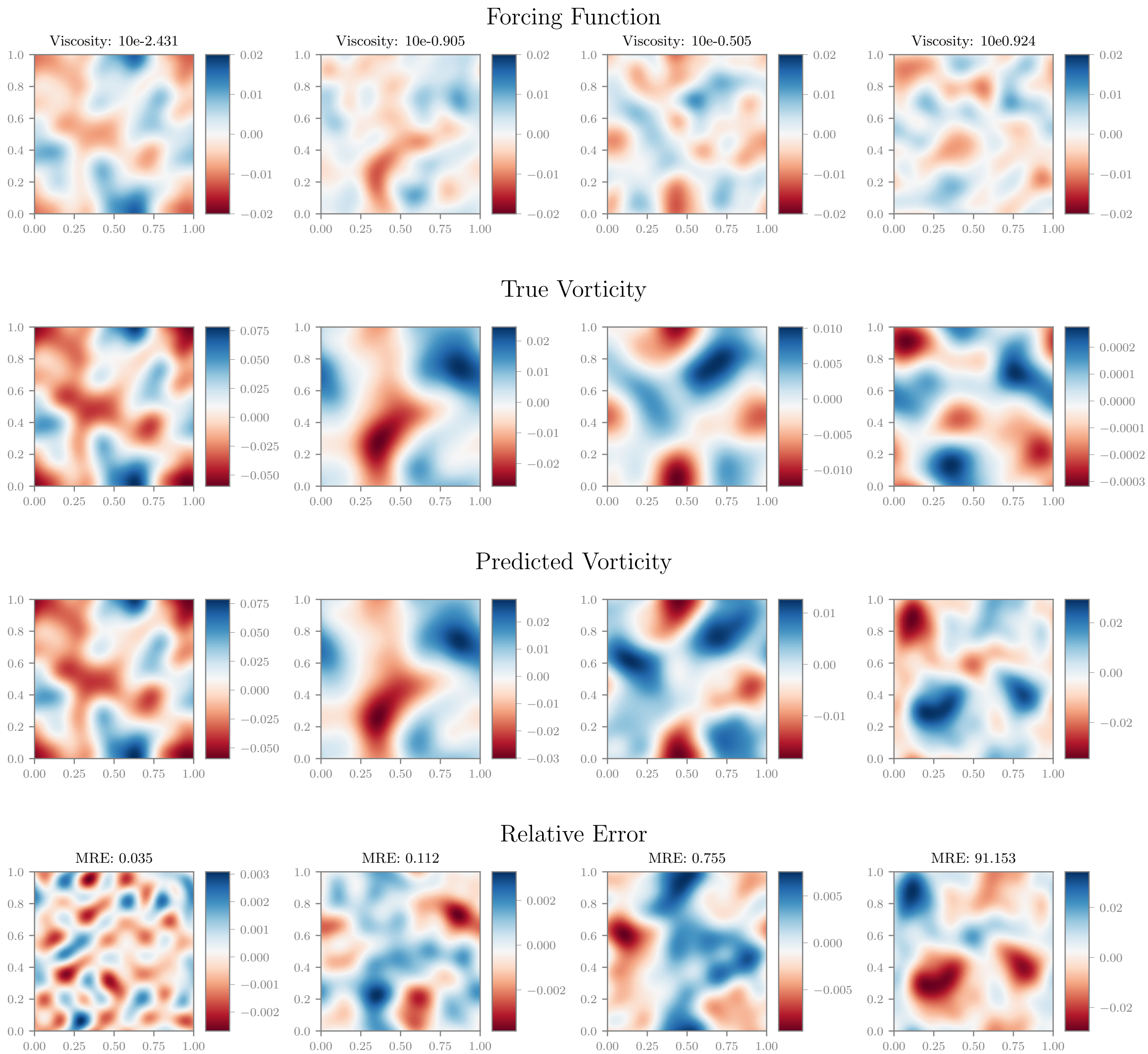


Figure 3: Forcing Function, True Vorticity, Predicted Vorticity, and Relative Error at the Lowest Error, Median Error, Highest Error Below a Viscosity of 10^0, and Highest Error Respectively in the Test-set

3.3 Explaining the Error at Greater Viscosities

In order to explain the errors that occur in higher viscosities, it can be helpful to look at how the result of the mapping varies with viscosity. This is shown in Figure 3. Observing first the result of the mapping at a low viscosity of $10^{-2.431}$, it is clearly visible that the resultant vorticity is an extremely close approximation of the forcing function with increased amplitudes. Further, looking at the highest error at a viscosity less than 10^0 it can be seen that as viscosity increases, the vorticity becomes lower as amplitudes are decreased and there is generally a greater amount of smoothing. Lastly, for the high viscosity case at $10^{0.924}$, it can be clearly seen that the resultant vorticity is of multiple magnitudes lower than the low-viscosity cases, with an even larger amount of smoothing. This is further supported by Figure 4, showing that the norm, and consequently amplitude, of the vorticity decreases as viscosity goes above 10^{-2}.

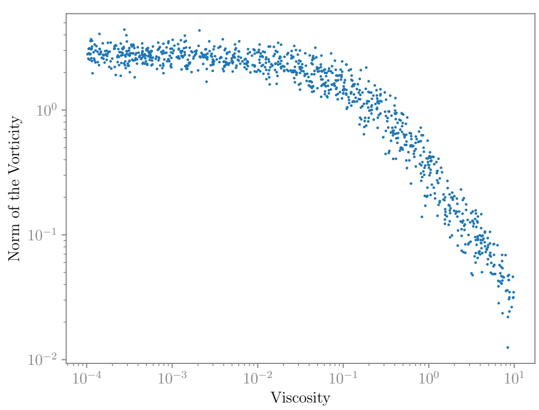


Figure 4: Norm of the True Vorticity Depending on Viscosity

Based upon this reduction in amplitude as viscosity increases, it is possible that the major increase in error above viscosities of 10^0 is due to the relative amplitude decreasing and the network being unable to cope with this change. This is supported by Figure 3 and Figure 5. As seen in Figure 5, as the norm, and consequently amplitude, reduces below $3 \cdot 10^{-1}$, the relative error quickly grows above 1, at which point the model is unable to effectively predict the resultant vorticity.

4 Responsible Research

4.1 Ethical Concerns

As with any research, it is important to reflect on the possible ethical concerns around research. For this project there are few ethical concerns to really consider.

As all the data is artificial, there is no concern that real data is leaked or that industry information is accidentally released.

In terms of the application, some concerns can be placed with how the data may be used. Ethical concerns can arise when software is used for efforts in the defense sector. Given that Navier-Stokes are commonly used in aerospace, there is always a chance that software around it is used in weapons development. However, given that this research is being done at a level that can not be directly applied to engineering tasks, this risk is incredibly low.

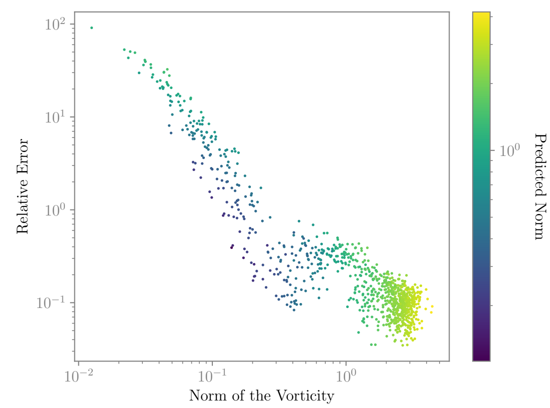


Figure 5: Relative error and Norm of the Predicted Output Depending on Norm of Ground Truth

4.2 Reproduction of Research

In order to ensure that the research done is reproducible, all code used in the research will be available online for free use. All code is forked from the Operator Learning repositor[1] by Zhengyu-Huang, as used in V. de Hoop et al. [2]. Given that this code does not have any license attached, but allows forking, the code from this repository was expanded upon. The code is available for anyone wishing to reproduce the research conducted in this paper, with all code used in the final report including the data visualization shared on github under the MIT license[2]. This should ensure easy reproduction for anyone aiming to expand upon the research committed in this paper.

5 Discussion

The aim of this paper is to investigate the ability of principal component analysis based neural nets to solve Navier-Stokes at varying viscosities.

From the results it is shown that the performance of PCA-net is highly dependent on the viscosity. When trained on the discrete viscosities shown in Table 1, PCA-net is able to extrapolate lower viscosities well, but it is unable to make good predictions for higher viscosities. This is shown to correlate with the lower norms of the resultant vorticity at higher viscosities.

There are some possible causes that lead to this weakness only manifesting at the higher viscosities rather than lower viscosities. As seen in Figure 4, three of the four training viscosities have very similar norms at the resulting vorticity, with only the training data at a viscosity of 10^0 having a much lower norm. Given a neural net only learns what its

[1] https://github.com/Zhengyu-Huang/Operator-Learning

[2] https://github.com/amundkiste/Reduced-Order-Mappings

trained on, the training data would have to include more high-viscosity cases to ensure adaptability in this region. With what information the network has been trained on, it has not been able to learn an approximation of what should happen to the amplitude of the vorticity when viscosity becomes higher. Thus, it is likely that some of the error here may be mitigated by extending the training set to include more viscosities above 10^{-1}.

At lower viscosities however, the performance of the PCA-net model is quite excellent, with performance very similar to what is found by V. de Hoop et al. [2]. At the viscosity of approximately $2.5 \cdot 10^{-2}$, the relative errors hover around $2 \cdot 10^{-1}$. While V. de Hoop et al. [2] do not state their error for Navier-Stokes explicitly, when trained on a test-set of 5000, PCA-net shows a mean test-error of 10^{-1}. The neural net in this paper trained for continuous viscosities - when omitting viscosities above 10^{0} - perform quite similarly on the test set with a relative error of $1.5 \cdot 10^{-1}$.

Relative to other methods of solving Navier-Stokes, the advantages of PCA-net are sizeable. Firstly, the mesh-independence of PCA-net effectively allows it to be applied to a variety of problems, without error growing as the mesh-size increases [3; 1]. Further, in comparison to traditional reduced basis methods, knowledge is not needed of the underlying PDE to model it effectively [3; 5]. Compared with other reduced basis networks like PARA-net or DeepONet, PCA-net has lower evaluation cost for test-errors, only being beat out by the Fourier Neural Operator [2].

The greatest advantage however of PCA-net, or for that matter just about any sufficiently accurate neural net, is the cost being magnitudes lower than any numerical simulations. While data-generation for the test-set took multiple hours to generate, the evaluation of the test-set using the model took less than a second.

On the backside, the biggest drawback of PCA-net relative to more complex methods is the limits of its adaptability. Unlike spectral methods like the Fourier Neural Operator, the model can only be applied to the same mesh as has been trained upon [2; 3].

6 Conclusions and Future Work

Aiming to investigate the ability of principal component analysis based neural nets to solve Navier-Stokes at varying viscosities, it was found that PCA-net exhibits good performance at low viscosities, with accuracy lowering as viscosities are increased. PCA-net was found to be somewhat satisfactory at predicting vorticities when interpolating viscosity. When extrapolating at lower viscosities errors converged towards 10^{-1}. On the flipside, extrapolation above a viscosity of 10^{0} shows divergent properties with errors growing with increased viscosity.

In order to improve performance of the model, the training set should be expanded upon to include more viscosities above 10^{0}. This would possibly reduce the divergence shown at higher viscosities.

Another line of future research would be to model more complex versions of Navier-Stokes, such as the 2-dimensional flow over an airfoil. Continued research into this would be more applicable for real-world applications, where the cost of evaluation is a real issue.

Lastly, this research should be recreated using other mappings such as the Fourier Neural Operator. Testing similar to that conducted in V. de Hoop et al. [2] would verify the performance of PCA-net in relation to other reduced-order mappings.

References

[1] Kaushik Bhattacharya, Bamdad Hosseini, Nikola B. Kovachki, and Andrew M. Stuart. Model reduction and neural networks for parametric pdes. 2021.

[2] Maarten V. de Hoop, Daniel Zhengyu Huang, Elizabeth Qian, and Andrew M. Stuart. The cost-accuracy trade-off in operator learning with neural networks. 2022.

[3] Nikola Kovachki, Zongyi Li, Burigede Liu, Kamyar Azizzadenesheli, Kaushik Bhattacharya, Andrew Stuart, and Anima Anandkumar. Neural operator: Learning maps between function spaces, 2023.

[4] Zongyi Li, Nikola Kovachki, Kamyar Azizzadenesheli, Burigede Liu, Kaushik Bhattacharya, Andrew Stuart, and Anima Anandkumar. Fourier neural operator for parametric partial differential equations, 2021.

[5] Mario Ohlberger and Stephan Rave. Reduced basis methods: Success, limitations and future challenges, 2016.

[6] The Engineering Toolbox. Viscosity - absolute (dynamic) vs. kinematic, 2003.