26th International Congress of Mechanical Engineering
COBEM 2021
ABCM Associação Brasileira de Engenharia e Ciências Mecânicas
November 22-26, 2021, Virtual Congress, Brazil

COB-2021-0667 A Machine Learning Approach to Solve the Lid Driven Cavity Flow

Rogério Werneck Costa Rodrigues Filho, rwerneckfilho@gmail.com
Adriano Possebon Rosa, aprosa@unb.br
Universidade de Brasília, Faculdade de Tecnologia, Departamento de Engenharia Mecância, Brasília, DF. CEP: 70910-900

Abstract. *The growing production of data has made possible to use machine learning algorithms on mechanical engineering problems that were not previously viable due the lack of data. Traditional numerical simulation sometimes might have a huge computational cost, in such way that real time simulation might not be possible. Highly monitored locations usually have a lot of sensors creating a log of historical data, for example temperature inside a data center along time, in a such way that modeling the physical conditions inside this site is more viable using machine learning techniques than traditional numerical simulation. Since machine learning algorithms do not require mathematical models of the physical problem to get real results, the use of these techniques would facilitate the entire process of modelling the problem when comparing to traditional simulation. In this work, a machine learning model focused on lid driven cavity flow, is developed using feed-forward neural networks based on data results obtained by traditional numerical methods, such as first order projection method. The use of these techniques does not require the use of Navier-Stokes equation to create reliable results. This means that, the algorithm is able to create its own mathematical model based only on data. Also, machine learning algorithms, after the training process, do not require a lot of computational power to process data, in such a way that the simulation time has been drastically reduced when compared to conventional numerical simulation methods. The final results precision, using machine learning techniques, has not been significantly affected in a way that is possible to obtain reliable results through this method.*

Keywords: *machine learning, 2D lid-driven cavity flow, Artificial neural networks, CFD*

1. INTRODUCTION

Fluid mechanics experiments and simulations traditionally deal with a vast amount of data. The growth of data available nowadays permeates the most diverse scientific areas. A great confluence is taking place with the combination of growth of data, advances in computational technologies and reduction of computational costs, abundance of open sources solutions available and large investment looking for data-driven solutions. Due this, machine learning begins to make rapid advances in the context of fluid mechanics (Brunton *et al.*, 2020).

Machine learning provides fast tools to solve fluid mechanics challenges, such as experimental data processing, shape optimization, turbulence analysis, and control tools. Despite the growing impact of machine learning algorithms on fluid mechanics today, in the early 1940s, Kolmogorov envisioned the study of turbulence as one of the main fields of application of machine learning (Brunton *et al.*, 2020). In the late 1980s, at one of the great advances in machine learning models, the development of error back-propagation models allowed the advancement of multi layer neural networks (Aggarwal, 2018; Brunton *et al.*, 2020), which raised the possibility of usability of the tool. Thus, in the early 1990s, several applications of neural networks for fluid mechanics emerged, such as the identification and configuration of multiphase flows (Bishop and James, 1993; Brunton *et al.*, 2020).

In recent years there has been a new wave of advances in the field of machine learning for fluid mechanics. Part of this interest is due to the great advances in the area of deep learning in recent years. The application of fluid mechanics problem solving with the aid of machine learning algorithms is the most diverse, such as temperature control in mission critical environments such as data centers (Song *et al.*, 2014), prediction of premature arrival of injected fluids in oil reservoirs (Bai and Tahmasebi, 2020), predict aneurysm growth (Jiang *et al.*, 2020), predict thermal conductivity and dynamic viscosity of magnetic nanofluids (Esfe *et al.*, 2015).

In this work a neural network is trained to solve the classical lid driven cavity flow. In section 2 the machine learning methodology and model description is presented among a brief historical path of the classical numerical methodology. In section 3 results are shown. In section 4 a brief conclusion is presented.

2. METODOLOGY

2.1 Lid Driven Cavity Flow and Traditional Approach

Since early 1960's (Kawaguti, 1961; Burggraf, 1966), the classic problem of the lid driven cavity flow, Fig. 1, has been used as benchmark for 2D Navier-Stokes numerical solution (Ghia *et al.*, 1982; Erturk *et al.*, 2005; Marchi *et al.*, 2009).This is a two-dimensional theoretical problem where the driven force of the fluid flow is given by the movement of the top wall, due the no slip condition, the near wall fluid starts the motion creating a central vortex on the cavity (Burggraf, 1966).

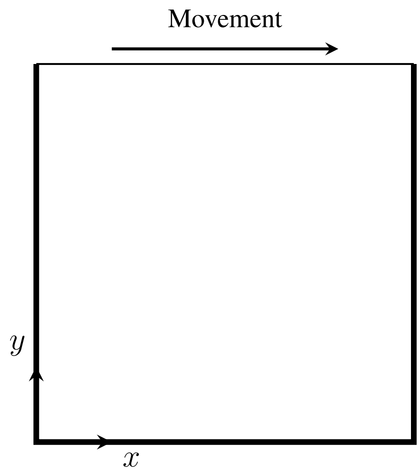


Figure 1. Ilustration of the lid driven cavity flow. Only the top wall moves while all other three are fixed.

The governing equations of this phenomena are the incompressible Navier-Stokes equations, given by

$$\nabla \cdot \mathbf{u} = 0, \tag{1}$$

and,

$$\frac{\partial \mathbf{u}}{\partial t} = -\mathbf{u} \cdot \nabla \mathbf{u} - \nabla p + \frac{1}{Re} \nabla^2 \mathbf{u}, \tag{2}$$

where ∇ is the nabla operator, $\mathbf{u}$ is the vector velocity, t represents time, p is the pressure field and

$$Re = \frac{\rho U D}{\mu}, \tag{3}$$

is Reynold's number, where ρ is the specific mass, U is the flow speed, D is the side lenght of the square cavity and μ is the kinematic viscosity of the fluid.

This classical benchmark is used to verify how accurate a solution is for the 2-D Navier-Stokes equation. There are many methods to solve unsteady Navier-Stokes equation using finite difference method. The data-set used in this paper to train the feed-forward neural network was generated using a first order projection method developed by Chorin (1968) to solve the unsteady equation. This method was applied using a Python code with Numpy and Numba libraries. This work only studies laminar flow, the data-set contains results with Reynold's number starting at 20 and increasing up to 1000 with a step of 20, i.e., $Re = 20\Gamma$ where $\Gamma = \{1, 2, 3, ..., 49, 50\}$.

To select the number of elements of the mesh it is necessary to analyse the mesh convergence increasing the number of elements up to a point where there are no significantly changes on the final result. On this work, however, this study was not made since the final result obtained by the machine learning approach will be compared only with the numerical result. It was selected a mesh shape with (26×27) elements to solve horizontal component of velocity (u), (27×26) elements to solve vertical component of velocity (v) and (27×27) elements to solve pressure field (p). The difference between the number of elements is due a staggered grid implemented (Chorin, 1968; Bell *et al.*, 1989).

2.2 Machine Learning Approach

The most basic neural network is the multi layer Perceptron, based on Rosenblatt (1958) Perceptron's. The Perceptron's conception consists on a single neuron with adjustable synaptic weights (w). A neuron, on neural networks context, is a node where all data is united by linear regression and added a bias (b). In such a way that Perceptron's might be written as:

$$z = \sum_{i=1}^{m} w_i x_i + b \tag{4}$$

Where m is the total of neurons on the layer. Let $x_0 = 1$ thus $w_0 x_0 = b$, rewriting Eq. (4):

$$z = \sum_{i=0}^{m} w_i x_i \tag{5}$$

An schematic view of Rosenblatt (1958) perceptron's might be seen on Fig. 2.

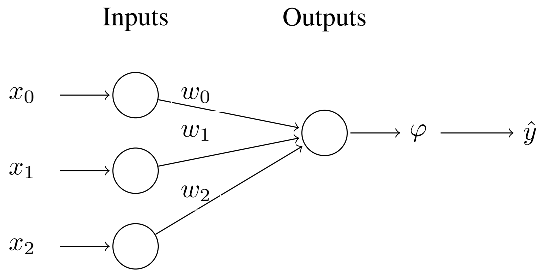


Figure 2. Schematic view of Rosenblatt's Perceptron.

The synaptic weights are updated using the stochastic gradient descend method given by (Aggarwal, 2018; Geron, 2019; Goodfellow, 2016):

$$w_{new} = w - \eta \nabla_w f(w). \tag{6}$$

Where η is the learning rate and f is the loss function used to find the optimal value of the synaptic weights. Loss functions are used to identify how far are predicted values ($\hat{y}$) from real values (y) among the total of samples (t). The loss function used in this paper is the Mean Squared Error (MSE) described by (Aggarwal, 2018; Geron, 2019; Goodfellow, 2016):

$$MSE = \frac{1}{t} \sum_{i=1}^{t} (\hat{y}_i - y_i)^2 \tag{7}$$

2.3 Model Description

To solve the lid driven cavity flow problem, three different multi-layer Perceptrons (MLP) were modeled using an open source library called TensorFlow. A MLP is only an association of Perceptrons used to gain more complexity to solve more complex problems, each of these MLPs would be responsible to solve one of the main properties evaluated on this problem:

- The first one solves horizontal velocity component (u)
- The second one solves the vertical velocity component (v)
- The third one solves the pressure field (p).

All of them were modeled using the same amount of inputs in a way that the first layer have 2 neurons. One is inputted with Reynold's number and the second one with the time step, as the data-set contains transient results. The main goal of these models is to solve the entire cavity mesh as the numerical method would do, so the output must be the entire mesh so the output layer for velocity models must have 702 neurons and for the pressure field 729 neurons. This difference is due to the staggered grid used on the projection method (Chorin, 1968; Bell *et al.*, 1989). The number of neurons on the hidden layer were chosen in such a way that all layers are linearly spaced. The Tab. 1 describes the model used.

Table 1. Model characteristics

Model	Mesh shape	Input Neurons	Hidden Neurons	Output Neurons
u	(26 × 27)	2	352	702
v	(27 × 26)	2	353	702
p	(27 × 27)	2	366	729

To obtain the correct synaptic weights, the MSE function, described in the Eq. (7), was chosen as loss function. The chosen optimizer was the stochastic gradient descent with learning rate $\eta = 0.05$, since validation loss have been monitored during training, overshooting was prevented with early stopping. For the early stop criteria, it was chosen that training should be terminated when reaching the minimum loss function given 30 new epochs, being restored the value of the synaptic weights from the moment that the lowest value was reached. Only the hidden layer has an activation function, the chosen one being the Rectified Linear Unit (ReLU).

3. RESULTS

3.1 Loss Function

The monitoring of the loss function is extremely important to identify how efficient was the training process. Figures 3, 4 and 5 illustrate the behavior of the function for the three models during training. Note that the parameter chosen for η does not exceed the convergence threshold and was not too small either, since convergence occurred in approximately 200 epochs for the three cases.

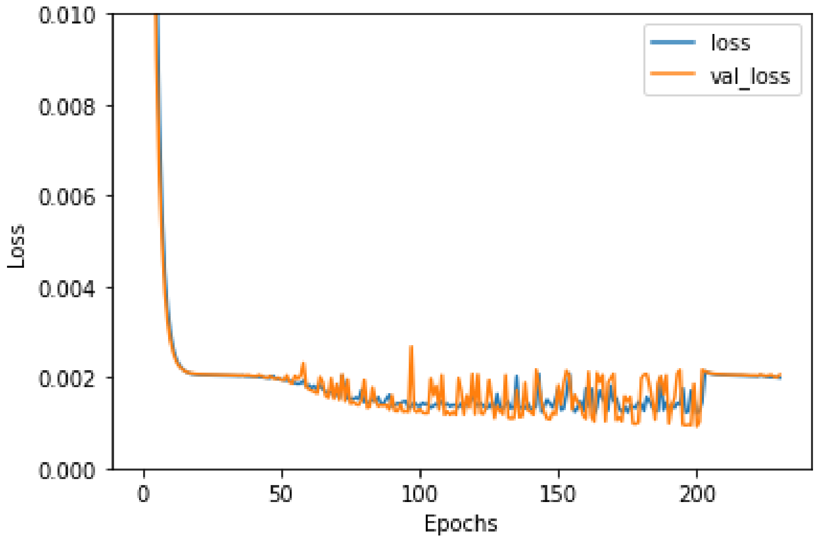


Figure 3. Loss function for Model u

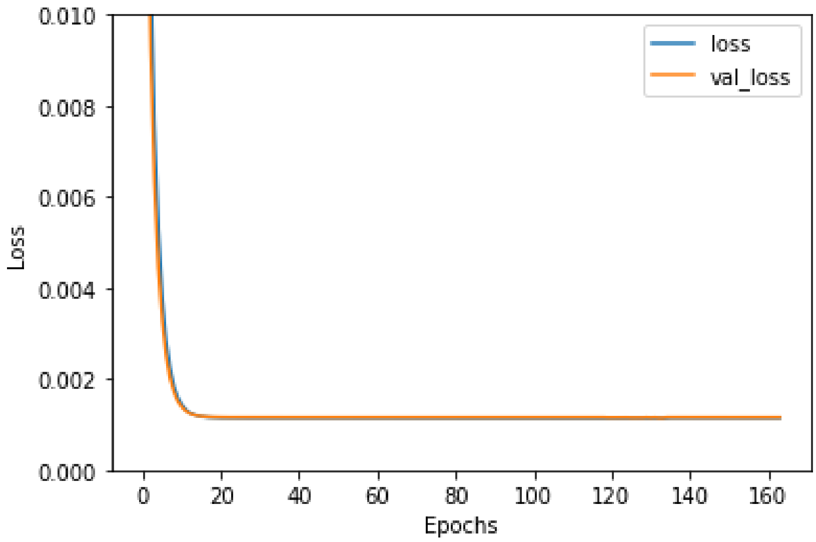


Figure 4. Loss function for Model v

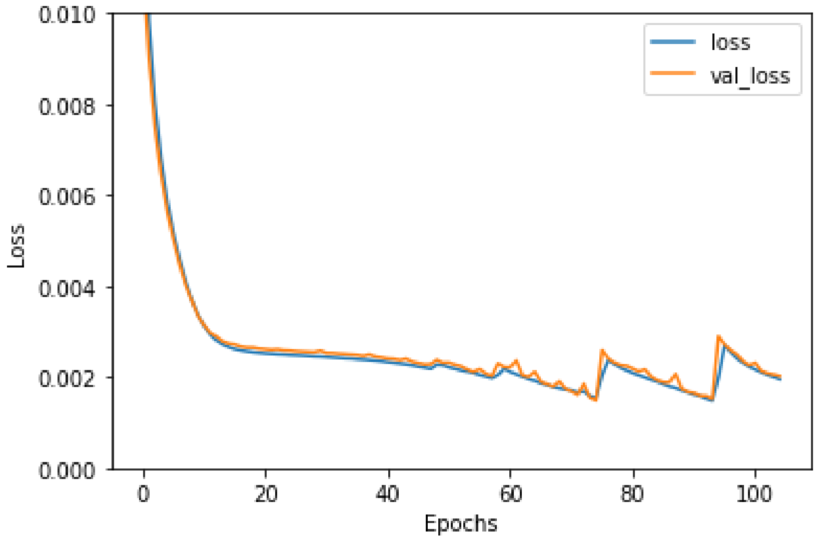


Figure 5. Loss function for Model (p)

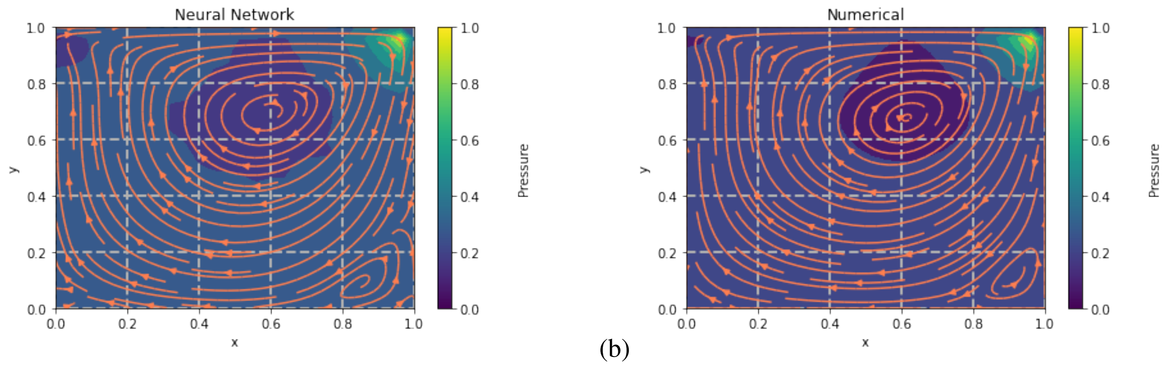
Thus, all three models were trained in a satisfactory manner, making it possible to predict the behavior of a fluid through neural networks. The final result of the loss function for each of the models is illustrated in the table 2.

Table 2. Results after train.

Model	Loss (MSE)
u	0,0009149
v	0,0011427
p	0,0014869

3.2 Comparison Between Numerical and Neural Networks

For a better view of the obtained results, an arbitrary value present in the test data-set was chosen to perform the comparison between the result obtained numerically and through the neural network, as this result comes from test data-set, it was not previously seen by the model. The chosen value has $Re = 220$ and $t = 5.78$, that is, these are the input values for the neural network.



(a) (b)

Figure 6. Streamlines and pressure field for $Re = 220$ and $t = 5.78$. It is shown results obtained by Neural Networks (a) and Numerical (b)

Note that the results presented in Fig. 6 are very similar. The possibility of using the trained neural network as a way of predicting the behavior of the fluid confined in a cavity becomes clear. There are some regions that show some divergences, such as the vortex located in the lower right corner of the cavity. In the numerical case it is more defined while the case obtained via neural networks presents discrepancies. Furthermore, observing the pressure field, it is noted that in the central region the shape of the drop present in the numerical result is more defined when compared to the result obtained via neural networks.

To improve the comparison of results, the values obtained for u in the region of the vertical center-line ($x = 0.5$) were observed, displayed in the Fig. 7.

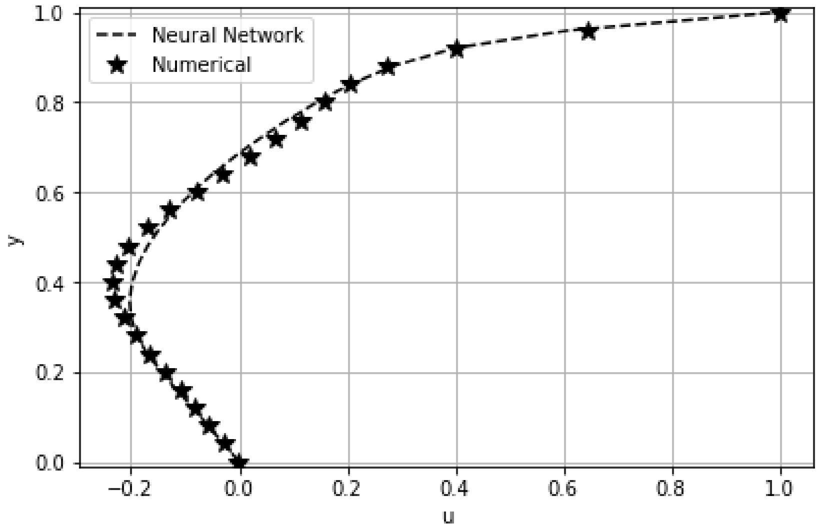


Figure 7. Comparison of velocity component (u) given $x = 0.5$ for $Re = 220$ and $t = 5.78$

There are few differences when comparing the two results, showing once again the possibility of using neural networks with multi layer perceptron architecture to solve the proposed problem.

Following the same procedure, this time for velocity (v) and horizontal centerline ($y = 0.5$), the results presented in Fig. 8 were obtained.

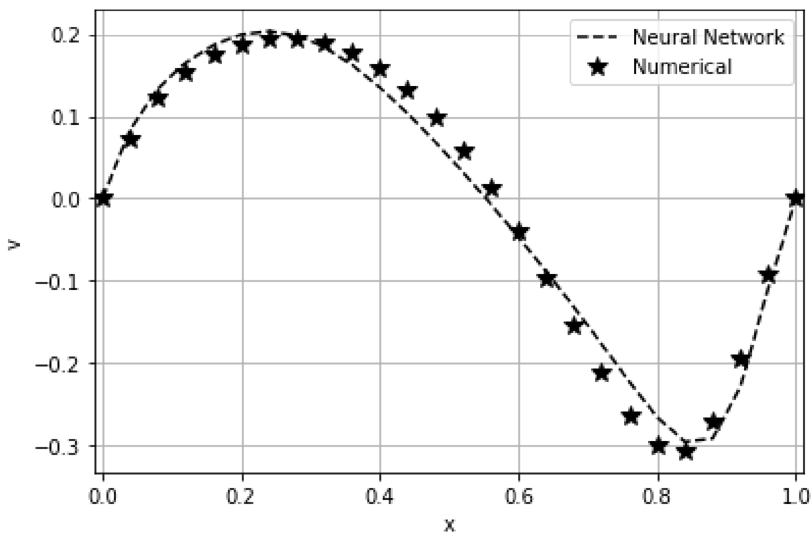


Figure 8. Comparison of velocity component (v) given $y = 0.5$ for $Re = 220$ and $t = 5.78$

Once again, it is shown that the results obtained through neural networks show little divergence when compared to the numerical solution.

To visualize the last trained model, the results obtained for the pressure along the lines ($x = 0.5$) and ($y = 0.5$) in the Fig 9 and 10 respectively

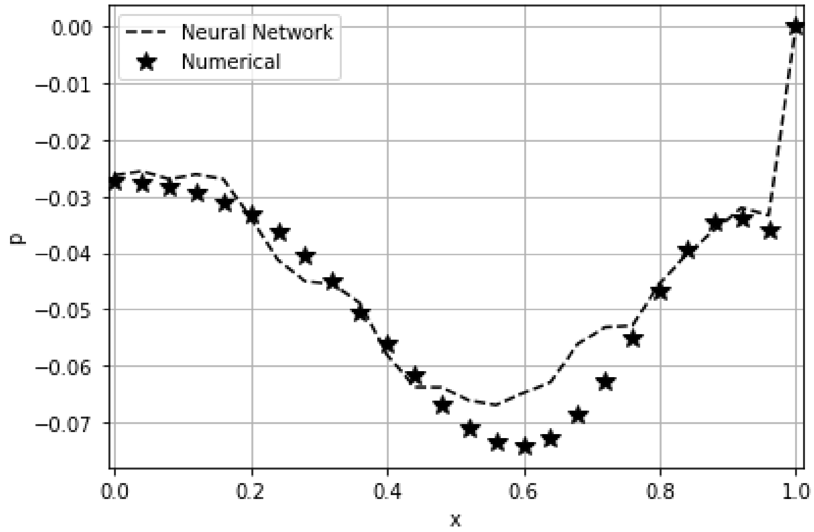


Figure 9. Comparison of (p) given $y = 0.5$ for $Re = 220$ and $t = 5.78$

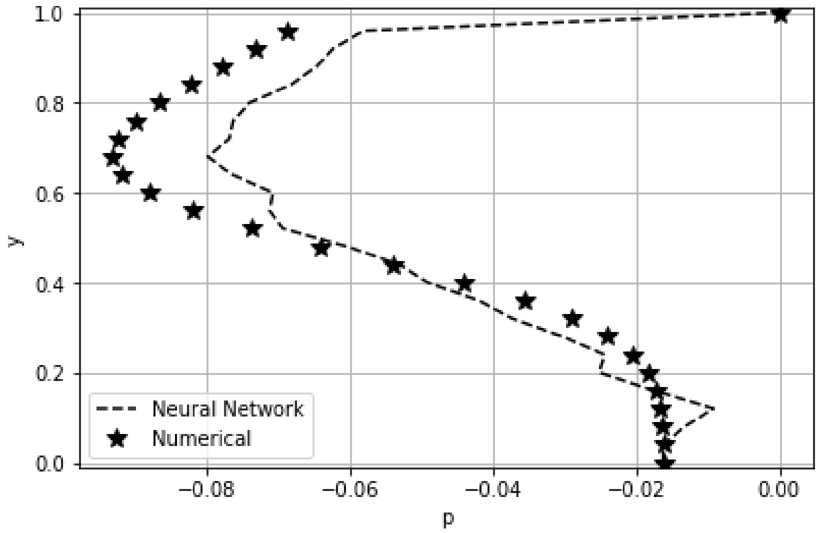


Figure 10. Comparison of (p) given $x = 0.5$ for $Re = 220$ and $t = 5.78$

For both cases, the behavior of the curve obtained through the neural network algorithm is similar to the behavior of the curve obtained numerically, however, the values have a greater discrepancy when compared to the results of velocities. This had already been evidenced in the Fig. 6, when observing the drop shape existing in the central region of the numerical result.

The simulation time using the numerical method cited on section 2, final time 5.78 and Reynold's number 220 is 18.4 seconds. While using neural networks, after training, the time elapsed is only 0.16 seconds, over 100 times faster. However, to train the algorithm it is necessary to create a large data-set, in this case with 75000 simulations, which in this case took around 8 days. Also, the training process takes around 2 hours for each model, giving a total time of 6 hours. The neural network training was done using Google Colab and numerical simulation on Linux 5.11.0-27 core i7 8th generation. So, the process of creating a model takes a longer time when compared to a single numerical simulation. Nevertheless a neural network model is a long time use alternative to get faster results, for example during the process of design decision in an engineering project.

4. CONCLUSION

The results show that Neural Networks might be a useful tool for a faster analysis of CFD models. A simple model with only one hidden layer might not be the best approach in a way that it seems necessary to do a deeper analysis using Deep Learning approaches, i.e., using more hidden layers to get better results. Also, an analysis of the results may indicate the necessity of a larger number of epochs to improve pressure results.

This is a preliminary study and there are future analysis that must be done to improve the case study. Such as, analyze different neural networks architecture.

5. REFERENCES

Aggarwal, C.C., 2018. *Neural Networks and Deep Learning*. Springer International Publishing. doi:10.1007/978-3-319-94463-0. URL https://doi.org/10.1007/978-3-319-94463-0.

Bai, T. and Tahmasebi, P., 2020. "Efficient and data-driven prediction of water breakthrough in subsurface systems using deep long short-term memory machine learning". *Computational Geosciences*, Vol. 25, No. 1, pp. 285–297. doi:10.1007/s10596-020-10005-2. URL https://doi.org/10.1007/s10596-020-10005-2.

Bell, J.B., Colella, P. and Glaz, H.M., 1989. "A second-order projection method for the incompressible Navier-Stokes equations". *Journal of Computational Physics*, Vol. 85, No. 2, pp. 257–283. ISSN 0021-9991. doi:https://doi.org/10.1016/0021-9991(89)90151-4. URL https://www.sciencedirect.com/science/article/pii/0021999189901514.

Bishop, C. and James, G., 1993. "Analysis of multiphase flows using dual-energy gamma densitometry and neural networks". *Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment*, Vol. 327, No. 2-3, pp. 580–593. doi:10.1016/0168-9002(93)90728-z. URL https://doi.org/10.1016/0168-9002(93)90728-z.

Brunton, S.L., Noack, B.R. and Koumoutsakos, P., 2020. "Machine learning for fluid mechanics". *Annual Review of Fluid Mechanics*, Vol. 52, No. 1, pp. 477–508. doi:10.1146/annurev-fluid-010719-060214. URL https://doi.org/10.1146/annurev-fluid-010719-060214.

Burggraf, O.R., 1966. "Analytical and numerical studies of the structure of steady separated flows". *Journal of Fluid Mechanics*, Vol. 24, No. 1, pp. 113–151. doi:10.1017/s0022112066000545. URL https://doi.org/10.1017/s0022112066000545.

Chorin, A.J., 1968. "Numerical solution of the Navier-Stokes equations". *Mathematics of Computation*, Vol. 22, No. 104, pp. 745–762. doi:10.1090/s0025-5718-1968-0242392-2.

Erturk, E., Corke, T.C. and Gökçöl, C., 2005. "Numerical solutions of 2-d steady incompressible driven cavity flow at high Reynolds numbers". *International Journal for Numerical Methods in Fluids*, Vol. 48, No. 7, pp. 747–774. doi:10.1002/fld.953. URL https://doi.org/10.1002/fld.953.

Esfe, M.H., Saedodin, S., Sina, N., Afrand, M. and Rostami, S., 2015. "Designing an artificial neural network to predict thermal conductivity and dynamic viscosity of ferromagnetic nanofluid". *International Communications in Heat and Mass Transfer*, Vol. 68, pp. 50–57. doi:10.1016/j.icheatmasstransfer.2015.06.013. URL https://doi.org/10.1016/j.icheatmasstransfer.2015.06.013.

Geron, A., 2019. *Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow: Concepts, Tools, and Techniques to Build Intelligent Systems*. O'Reilly Media. ISBN 1492032646. URL https://www.xarg.org/ref/a/1492032646/.

Ghia, U., Ghia, K. and Shin, C., 1982. "High-Re solutions for incompressible flow using the Navier-Stokes equations and a multigrid method". *Journal of Computational Physics*, Vol. 48, No. 3, pp. 387–411. doi:10.1016/0021-9991(82)90058-4. URL https://doi.org/10.1016/0021-9991(82)90058-4.

Goodfellow, I., 2016. *Deep Learning (Adaptive Computation and Machine Learning series)*. The MIT Press. ISBN 0262035618. URL https://www.xarg.org/ref/a/0262035618/.

Jiang, Z., Do, H.N., Choi, J., Lee, W. and Baek, S., 2020. "A deep learning approach to predict abdominal aortic aneurysm expansion using longitudinal data". *Frontiers in Physics*, Vol. 7. doi:10.3389/fphy.2019.00235. URL https://doi.org/10.3389/fphy.2019.00235.

Kawaguti, M., 1961. "Numerical solution of the Navier-Stokes equations for the flow in a two-dimensional cavity". *Journal of the Physical Society of Japan*, Vol. 16, No. 11, pp. 2307–2315. doi:10.1143/jpsj.16.2307. URL https://doi.org/10.1143/jpsj.16.2307.

Marchi, C.H., Suero, R. and Araki, L.K., 2009. "The lid-driven square cavity flow: numerical solution with a 1024 x 1024 grid". *Journal of the Brazilian Society of Mechanical Sciences and Engineering*, Vol. 31, pp. 186 – 198. ISSN 1678-5878. URL http://www.scielo.br/scielo.php?script=sci$_{a}rttextpid = S1678 - 58782009000300004nrm = iso$.

Rosenblatt, F., 1958. "The perceptron: A probabilistic model for information storage and organization in the brain." *Psychological Review*, Vol. 65, No. 6, pp. 386–408. doi:10.1037/h0042519. URL https://doi.org/10.1037/h0042519.

Song, Z., Murray, B.T. and Sammakia, B., 2014. "A dynamic compact thermal model for data center analysis and control using the zonal method and artificial neural networks". *Applied Thermal Engineering*, Vol. 62, No. 1, pp. 48–57. doi: 10.1016/j.applthermaleng.2013.09.006. URL https://doi.org/10.1016/j.applthermaleng.2013.09.006.

6. RESPONSIBILITY NOTICE

The authors are solely responsible for the printed material included in this paper.