26th International Congress of Mechanical Engineering
COBEM 2021
November 22-26, 2021, Virtual Congress, Brazil

COB-2021-XXXX
Unsupervised Machine Learning technique for solving flows of generalized Newtonian fluids.

Bernardo Pereira Brener
Roney Leon Thompson
COPPE, Department of Mechanical Engineering, Universidade Federal do Rio de Janeiro, Centro de Tecnologia, Ilha do Fundao, 21945-970, Rio de Janeiro, RJ, Brazil
bernardobrener@coppe.ufrj.br
rthompson@mecanica.coppe.ufrj.br

Abstract. *A physics informed neural networks framework is employed in the present work in order to approximate the solution of partial differential equations through an unsupervised task. More specifically, it is solved here the set of linear and non-linear equations that rules the flows generalized Newtonian fluids. This innovative methodology uses the main advantage of automatic differentiation of neural networks to build a loss function that obeys a set of partial differential equations. When the training algorithm of the network is run, it converges to the solution of the specified equations by minimizing the loss function. The input of the network is the local coordinates of the problem and the output is the solution of the set of equations, which in the following application is the velocity and stress fields. The boundary and initial conditions are satisfied using the more traditional approach of machine learning techniques of a supervised task where a loss function is built using an error metric between the known output and the predicted one. This is added in the methodology simply by summing both loss functions, as the information in the boundary is previously known. This methodology has the advantage of not using any numerical differentiation which naturally does not bring any problems related to numerical methods. Furthermore, this methodology is mesh-independent. It was already successfully applied and validated for solving ordinary and partial differential equations in the literature. It is proposed here to apply this methodology for solving flows of non-Newtonian fluids, namely flows of generalized Newtonian fluids. The solution obtained with the neural networks is compared with the ones obtained through the classical approach of numerical methods and analytical solutions when available. It is shown that this methodology can successfully be applied to resolve this class of problems providing accurate solutions in all cases analyzed.*

Keywords:*Machine Learning, Non-Newtonian fluids, Artificial neural networks.*

1. INTRODUCTION

With the expressive increase of computational resources and amount data recording, it has allowed the application of machine learning (M.L) techniques to several fields in science, such as image recognition (Krizhevsky *et al.*, 2012), genomics (Alipanahi *et al.*, 2015) and fluid mechanics (Brunton *et al.*, 2020)(Kutz, 2017). However, the acquisition of data in several cases is very costly which still imposes an import barrier for real applications. Furthermore, when applying M.L with limited amount of data, it raises questions regarding the robustness and reliability of these models when predicting unseen data.

In fluid mechanics, the application of M.L to turbulence modeling received a number of contributions (Ling *et al.*, 2016a)(Ling *et al.*, 2016b)(Cruz *et al.*, 2019)(Kaandorp and Dwight, 2020) due to the fact that is still a critical problem for industrial applications. The first works focused on using M.L to provide a non-linear map between a number of input features of Reynolds Averaged Navier-Stokes (RANS) models and Reynolds stresses of Direct Numerical Simulations (DNS) (Wang *et al.*, 2017). However, as the cost of DNS for high Reynolds number and complex geometries is prohibitively, the previously mentioned short data regime is normally true. Another relevant issue is the fact that the DNS data available were not produced with the prior intention of being used for M.L applications. Therefore, the data available for training a model has huge gaps of characteristics between the cases. Although such databases are well suited for validating RANS modeling, for M.L implementations it is seen as sparsely scattered data in the parameter space. With this motivation, recent efforts were made in order to run DNS for M.L training by continuously varying a specific parameter in the flow (Xiao *et al.*, 2020).

Both previously mentioned issues are critical in order to produce real applications of ML to turbulence modeling, namely the lack of data due to high costs and the fact that the ones available are sparsely scattered in the parameter space. These problems have a direct impact on the accuracy and robustness of the trained model. Other studies showed that different strategies can be adopted in other to improve the accuracy of the M.L model. In the work of Brener *et al.* (2021), it was showed that RANS equations are ill conditioned and proposed a more accurate method based on the Reynolds Force Vector instead of the traditional Reynolds Stress Tensor (RST) for data-driven turbulence modeling. This study was motivated by the work of Thompson *et al.* (2016) which revealed that a very small error in the RST could result in largely discrepant mean velocity fields after solving the mean momentum equation. This context brings into light a debate that is rarely seen in this recent literature of data-driven turbulence modeling which is whether one can predict directly the mean velocity field of a turbulent flow. As the equations are indeed ill-conditioned, the M.L predictions are often subjected to high error amplification, whereas predicting the mean velocity field would not only bypass the problem but also RANS equations. A very relevant argument against this strategy is that the M.L has no compromise in producing a field that obeys physical laws or/and has physical meaning (e.g. a mean velocity field that violates the conservation of mass or has non-null velocity at walls even with no-slip boundary conditions).

Although a briefly context of M.L applications in turbulence modeling were given, since this subject is recent but already with a great number of studies, other fields face the exact same issues, i.e., to conceive M.L models with a very limited amount of data, the data available are often not suited for M.L applications and how can one creates a M.L model that obeys physical laws? With this motivation,Raissi *et al.* (2017b) proposed a methodology based on a *Physics-Informed Neural Networks* (PINN) that not only embeds the underlying equations of the problem into the M.L model, but also can be used to resolve the set of governing equations. The latter starts an unexplored way of solving differential equations alongside with numerical methods for forward and inverse problems. This idea was already proposed in Lagaris *et al.* (1998) but only with modern tools was possible to effectively put in practice. This possibility allows the use of M.L to produce its own database without the need of numerical simulations in several applications. This is possible due to the advantage of automatic differentiation of neural networks, allowing to build a loss function that obeys a set of differential equations. Furthermore, it does not bring any problems related with numerical methods. Using this methodology Sirignano and Spiliopoulos (2018) created a neural network to efficiently solve high dimensional PDEs which has been a longstanding challenge to do with traditional numerical methods. In turbulence modeling Jin *et al.* (2021) showed that the PINNs could sustain turbulence solving the incompressible three-dimensional plane-channel flow with an accuracy very similar to DNS. The PINN was successfully applied and validated for many other non-linear partial differential equations across several fields (Raissi *et al.*, 2017b) (Raissi *et al.*, 2017a) (Raissi, 2018)(Raissi *et al.*, 2019)(Raissi *et al.*, 2020).

It is proposed in present work to apply PINNs to solve non-Newtonian fluid flows. The literature on the application of M.L to this class of problems is short and very recent when compared with turbulence modeling, for example. To name a few, Muravleva *et al.* (2018) used neural networks to build a reduced order model for viscoplastic duct flows and Mahmoudabadbozchelou and Jamali (2021) used the same methodology explained above to solve ODEs (Ordinary differential equations) of non-Newtonian constitutive models. In the present work it is used a PINN framework to solve PDEs (Partial Differential Equations) that govern the flow of power-law fluids in the square-duct geometry.

2. METHODOLOGY

The methodology of present work is divided as the following: first, it is explained the process in order to solve differential equations through Physics Informed Neural Networks, then the mathematical formulation of the Non-Newtonian fluid flow is deduced for the square duct flow and, finally, it is described the problem setup for the numerical simulations and for the PINN.

2.1 Physics Informed Neural Networks

Figure 1 illustrates the schematic diagram of the PINN. The idea is to approximate the solution of a differential equation by a feed forward neural network. First, it is defined a feed forward deep neural network which the inputs are given by the coordinates of the problem, such as (x,y,z,t), and the output is given by the quantity that is being solved. Consider the general form of a differential equations as:

$$\mathbf{u}_t + \mathcal{L}[\mathbf{u}] = 0 \tag{1}$$

Where $\mathbf{u}$ is a vector quantity that is being solved and $\mathcal{L}$ is a differential operator. In order to solve this equation with neural networks it is defined the following a residual function

$$f := \mathbf{u}_t + \mathcal{L}[\mathbf{u}] \tag{2}$$

The loss function of the training process of the neural network is given by a measure of the defined residual function, which in the present work is given by the mean squared error:

$$MSE_f = \frac{1}{N_f}\sum_{i=1}^{N_f}|f_i|^2 \tag{3}$$

Where i specify each of the collocation points provided as inputs. However, in order to obtain a unique solution the equation has also to obey boundary and initial conditions. For this reason a second loss function is defined by

$$MSE_{b,0} = \frac{1}{N_{b,0}} \sum_{i=1}^{N_u} \left|u_0^i - u^i(0,.)\right|^2 + \left|u_b^i - u^i(t, boundary)\right|^2 \tag{4}$$

Where u_0^i and u_b^i are the previously known values of the field at the boundaries and initial conditions, respectively, and $u^i(0,.)$ and $u^i(t, boundary)$ is the output of the neural networks at the same places. Therefore, the actual loss function used to train the neural networks is given by the weighted sum of the previously defined ones.

$$loss = \alpha\,(MSE_{b,0}) + \beta\,(MSE_f) \tag{5}$$

During the training process the loss function is minimized. For each iteration the neural networks provides a field which feeds the loss function, then, the weights and biases are updated accordingly with the optimization algorithm until convergence. In the end of the process, the neural network has the weights and biases that approximate the solution of the differential equation and boundary conditions.

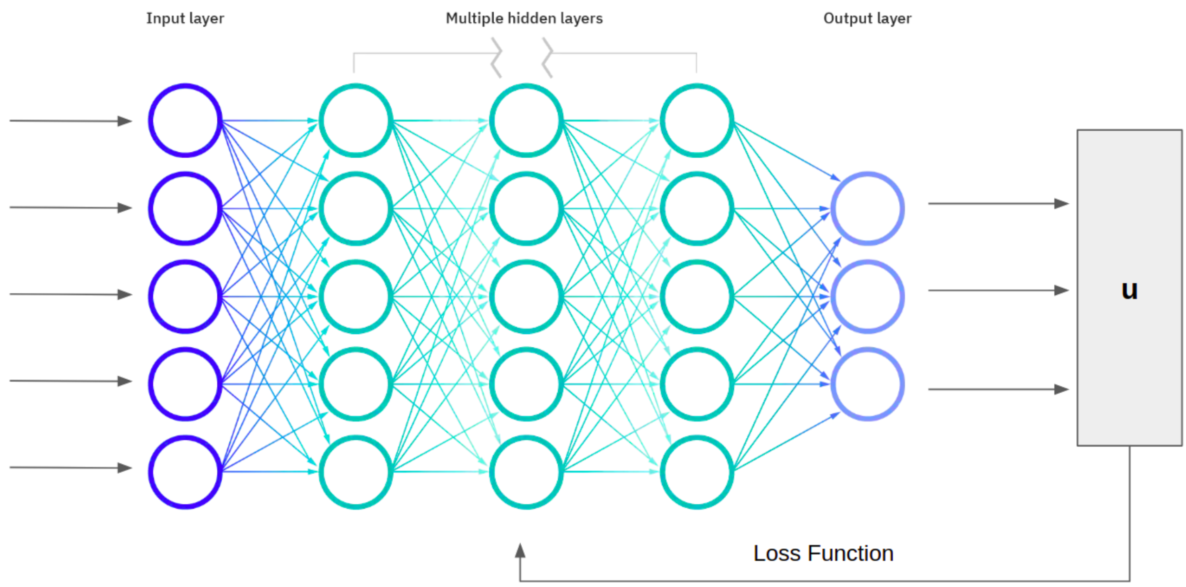

Figure 1. Schematic diagram of the PINN.

2.2 Mathematical Formulation

The class of generalized Newtonian fluids flows solved in present work is given by laminar incompressible power-law fluid flows. For theses flows, the conservation of momentum and mass equations, respectively, are given by:

$$\frac{\partial \mathbf{u}}{\partial t} + (\nabla \mathbf{u})\,\mathbf{u} = -\frac{1}{\rho}\nabla p + \nabla \cdot (2\eta(\dot{\gamma})\mathbf{D}) \tag{6}$$

$$\nabla \cdot \mathbf{u} = 0 \tag{7}$$

and

$$\eta(\dot{\gamma}) = K|\dot{\gamma}|^{n-1} = K\left[|2tr(\mathbf{D}^2)|\right]^{\frac{n-1}{2}} \tag{8}$$

Where $\mathbf{u}$ is the velocity field, p is the pressure field, $\mathbf{D}$ is the rate-of-strain tensor, K is the consistency parameter divided by the specific mass ρ and n is the power-law index. For the two-dimensional developed square duct flow, the velocity field due to physical constraints reduces to:

$$\mathbf{u} = u(x, y)\mathbf{k} \tag{9}$$

where $\mathbf{k}$ denotes the unit vector along the z-coordinate direction where the flow is developed. For these flows, the conservation of mass is automatically satisfied and the momentum equation simplifies to:

$$\frac{\partial}{\partial x}\left(\eta(\dot{\gamma})\frac{\partial u}{\partial x}\right)+\frac{\partial}{\partial y}\left(\eta(\dot{\gamma})\frac{\partial u}{\partial y}\right)=-\frac{1}{\rho}\frac{dp}{dz} \tag{10}$$

and the viscosity function simplifies to:

$$\eta=K\left[\left(\frac{\partial u}{\partial x}\right)^2+\left(\frac{\partial u}{\partial y}\right)^2\right]^{(n-1)/2} \tag{11}$$

The boundary conditions are no-slip at the walls,

$$u(x,y=0)=0 \qquad u(x=0,y)=0 \tag{12}$$

For $n=1$ the Newtonian behavior is recovered and Eq. 10 becomes linear. In this case, the solution of Eq. 10 can also be obtained analytically via Fourier series (White and Corfield, 2006) and is given by:

$$u(x,y)=\frac{16a^2}{K\pi^3}\left(-\frac{1}{\rho}\frac{d\hat{p}}{dz}\right)\sum_{i=1.3.5,...}^{\infty}(-1)^{(i-1)/2}\left[1-\frac{\cosh(i\pi x/2a)}{\cosh(i\pi/2)}\right]\frac{\cos(i\pi y/2a)}{i^3} \tag{13}$$

2.3 Problem Setup

The architecture adopted in the present work to solve the square duct flow consists of 7 hidden layer with 20 neurons each. The optimization algorithm used for training is the L-BFGS (Liu and Nocedal, 1989). The mesh used as input was formed by choosing randomly 10000 collocation points and 400 to enforce the boundary conditions. As the Neural Network approximate the solution of the partial differential equation, after the training process, the solution can be obtained at any point at the domain. The weights adopted in the loss function were $\alpha=10$ and $\beta=1$.

For the CFD (Computational Fluid Dynamics) simulations it was used the finite-volume platform OpenFoam. The mesh consisted of 125x125 points and cyclic boundary conditions were set in order to obtain a developed flow. It was also set non-slip boundary conditions at walls.

For all cases is considered the following generalized Reynolds number,

$$Re_g=\frac{\bar{U}^{n-2}H^n}{K} \tag{14}$$

Where it is adopted $H=2m$, $K=5\cdot10^{-2}m/s^n$ and $\bar{U}$ is adjusted keeping $Re_g=100$.

3. RESULTS

As mentioned before, the problem considered here are the flow in a square-duct. First, it is solved the Newtonian case through a CFD simulation, analytically using Eq. 13 and using PINN. Then, it is considered the cases of a shear-thinning fluid with $n=0.5$ and shear-thickening with $n=1.5$. It is worth to mention that for the Newtonian case the momentum equation is linear, whereas for power-law fluid the dependency on the shear rate gives rise to a non-linearity. Therefore, the methodology can be tested with linear and non-linear partial differential equations.

3.1 Newtonian fluid

For the Newtonian fluid it is possible compare the solution obtained through PINN, CFD and the exact analytical solution. Figure 2 illustrates this comparison in a two-dimensional plot for the velocity field in only a quarter of duct due to symmetries in the flow. It is observed that both the CFD and ML solutions are in good agreement with the analytical solution.

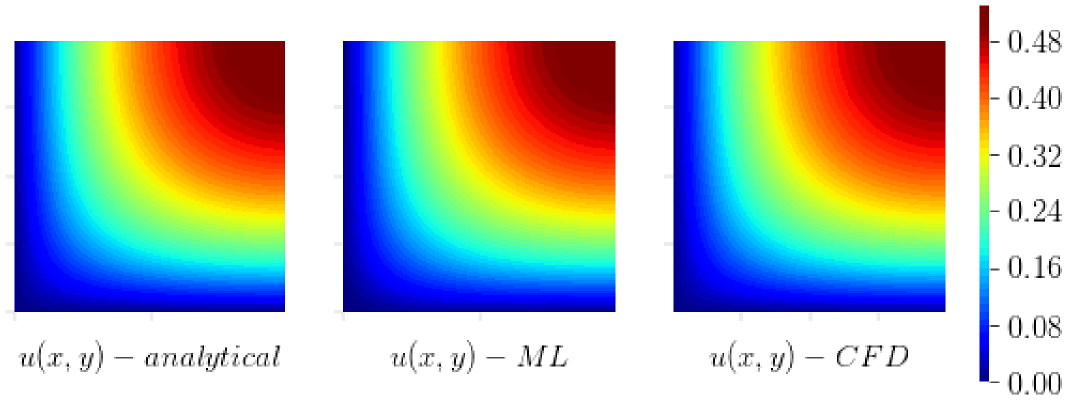

Figure 2. Comparison of the velocity field $u(x, y)$ obtained through ML, CFD and analytically for a Newtonian fluid.

In order to obtain a quantitative measure, it was defined the following metric:

$$E_j = \frac{\|u_j(x,y) - u_a(x,y)\|}{\bar{U}} \tag{15}$$

Where $u_a(x, y)$ is the analytical solution and $u_j(x, y)$ can be the solution obtained through CFD or ML. Figure 3 shows the comparison of the three methodologies using error metric. It is observed that although both discrepancies are small ($< 6\%$), the machine learning performs worse than the CFD close to the wall, whereas the CFD has a poorer performance close to the center of the duct. This happens because the ML has also predict the velocity field at the boundary, while in the CFD it is not resolved. On the other hand, at the center of the duct, where CFD the mesh is less refined, numerical errors are more expressive. Figure 4 shows the same comparisons for a profile at the center of the duct. It is observed a maximum error of 1.77 % and 1.74 % for the ML and CFD solution, respectively.

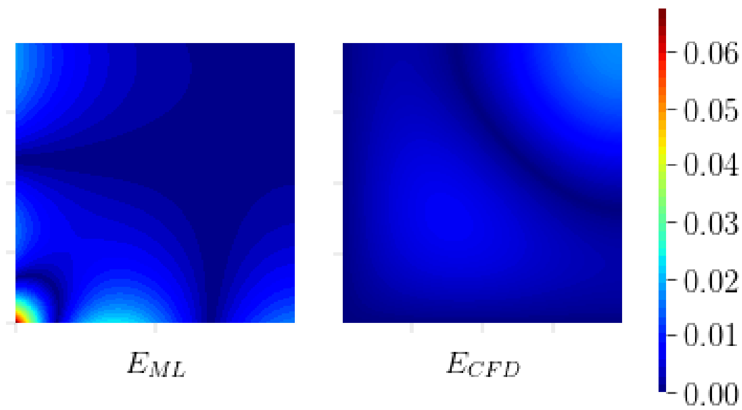

Figure 3. Relative absolute error with respect to the exact solution.

3.2 Power-Law fluid

For the power-law fluid a non-linear partial differential equation is solved. For this cases the ML solutions are validated and compared using the CFD ones. Figures 5 and 6 shows the velocity fields obtained through ML and CFD for the shear-thinning ($n = 0.5$) and thickening cases ($n = 1.5$). It is observed a good agreement between solutions for both cases. However, for the case $n = 0.5$ at center of the duct is observed a slightly higher velocity. On the other hand, for $n = 1.5$ any difference can not be noticed.

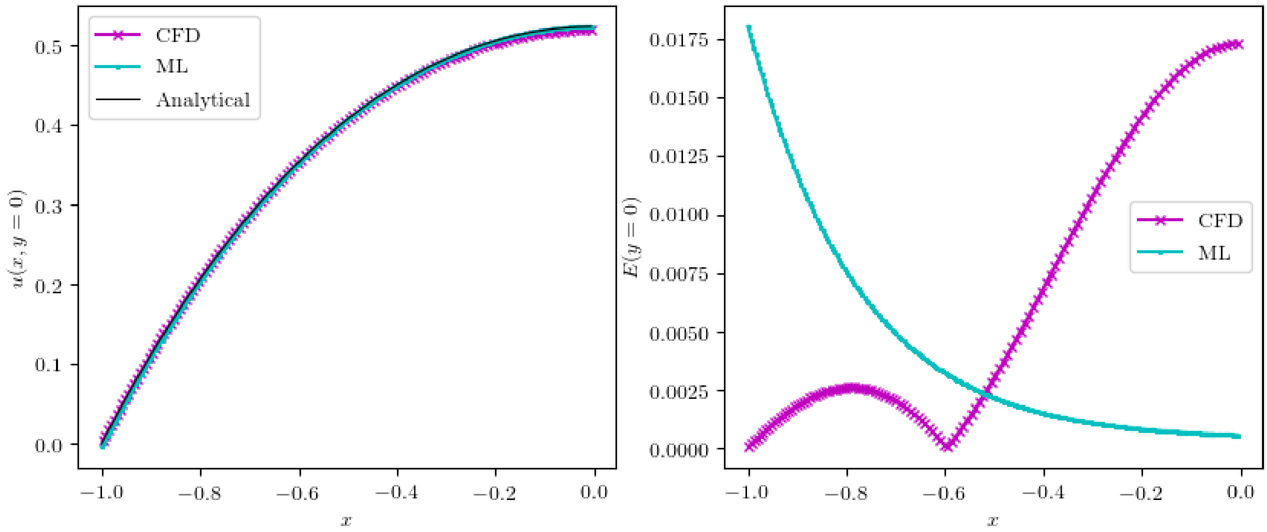

Figure 4. On the left, the profiles of the velocity field at center of the duct ($u(x, y = 0)$) obtained through ML, CFD and analytically. On the right shows the error plot of left figure.

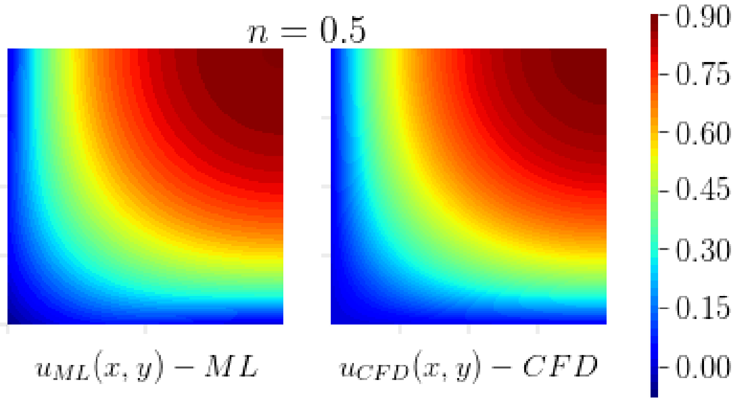

Figure 5. Comparison of the velocity field $u(x, y)$ obtained through ML and CFD for a shear-thinning fluid ($n = 0.5$).

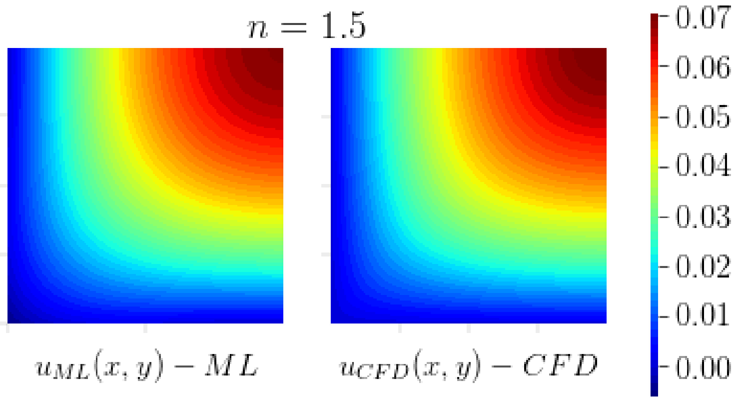

Figure 6. Comparison of the velocity field $u(x, y)$ obtained through ML and CFD for a shear-thickening fluid ($n = 1.5$).

4. CONCLUSION

The present work applied a Physics-Informed Neural Networks framework to approximate the solution of non-linear partial differential equations that rules the flow of generalized Newtonian fluids. The solutions were also obtained through traditional numerical methods and analytically for the Newtonian case. It was shown that for all cases, the solutions obtained through ML provided accurate results when compared with respective reference solution. Furthermore, in the Newtonian case, when compared with the analytical solution, the CFD solution performed better closer to the boundaries while the ML solution were more accurate closer to the center of the duct. In both, shear-thickening and shear-thinning cases, where a non-linear partial differential equation is solved, the ML could provide solutions very close to the CFD counterpart. Also, it is worth to mention that the ML solution was successfully obtained without the need for committing to any prior assumptions such as linearization.

5. ACKNOWLEDGEMENTS

We would like to thank Conselho Nacional de Pesquisa e Desenvolvimento (CNPq) Grant 304095/2018-4 and Coordenacao de Aperfeicoamento de Pessoal de Nivel Superior (CAPES) Grant PROEX 803/2018 for financial support.

6. REFERENCES

Alipanahi, B., Delong, A., Weirauch, M.T. and Frey, B.J., 2015. "Predicting the sequence specificities of dna-and rna-binding proteins by deep learning". *Nature biotechnology*, Vol. 33, No. 8, pp. 831–838.

Brener, B.P., Cruz, M.A., Thompson, R.L. and Anjos, R.P., 2021. "Conditioning and accurate solutions of reynolds average navier–stokes equations with data-driven turbulence closures". *Journal of Fluid Mechanics*, Vol. 915.

Brunton, S.L., Noack, B.R. and Koumoutsakos, P., 2020. "Machine learning for fluid mechanics". *Annual Review of Fluid Mechanics*, Vol. 52, pp. 477–508.

Cruz, M.A., Thompson, R.L., Sampaio, L.E. and Bacchi, R.D., 2019. "The use of the reynolds force vector in a physics informed machine learning approach for predictive turbulence modeling". *Computers & Fluids*, Vol. 192, p. 104258.

Jin, X., Cai, S., Li, H. and Karniadakis, G.E., 2021. "Nsfnets (navier-stokes flow nets): Physics-informed neural networks for the incompressible navier-stokes equations". *Journal of Computational Physics*, Vol. 426, p. 109951.

Kaandorp, M.L. and Dwight, R.P., 2020. "Data-driven modelling of the reynolds stress tensor using random forests with invariance". *Computers & Fluids*, p. 104497.

Krizhevsky, A., Sutskever, I. and Hinton, G.E., 2012. "Imagenet classification with deep convolutional neural networks". *Advances in neural information processing systems*, Vol. 25, pp. 1097–1105.

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

Lagaris, I.E., Likas, A. and Fotiadis, D.I., 1998. "Artificial neural networks for solving ordinary and partial differential equations". *IEEE transactions on neural networks*, Vol. 9, No. 5, pp. 987–1000.

Ling, J., Jones, R. and Templeton, J., 2016a. "Machine learning strategies for systems with invariance properties". *Journal of Computational Physics*, Vol. 318, pp. 22–35.

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

Liu, D.C. and Nocedal, J., 1989. "On the limited memory bfgs method for large scale optimization". *Mathematical programming*, Vol. 45, No. 1, pp. 503–528.

Mahmoudabadbozchelou, M. and Jamali, S., 2021. "Rheology-informed neural networks (rhinns) for forward and inverse metamodelling of complex fluids". *Scientific reports*, Vol. 11, No. 1, pp. 1–13.

Muravleva, E., Oseledets, I. and Koroteev, D., 2018. "Application of machine learning to viscoplastic flow modeling". *Physics of Fluids*, Vol. 30, No. 10, p. 103102.

Raissi, M., Perdikaris, P. and Karniadakis, G., 2017a. "Physics informed deep learning (part ii): Data-driven, discovery of nonlinear partial differential equations,âĂİ, arxiv e-prints, p". *arXiv preprint arXiv:1711.10566*.

Raissi, M., 2018. "Deep hidden physics models: Deep learning of nonlinear partial differential equations". *The Journal of Machine Learning Research*, Vol. 19, No. 1, pp. 932–955.

Raissi, M., Perdikaris, P. and Karniadakis, G.E., 2019. "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.

Raissi, M., Perdikaris, P. and Karniadakis, G.E., 2017b. "Physics informed deep learning (part i): Data-driven solutions of nonlinear partial differential equations". *arXiv preprint arXiv:1711.10561*.

Raissi, M., Yazdani, A. and Karniadakis, G.E., 2020. "Hidden fluid mechanics: Learning velocity and pressure fields from flow visualizations". *Science*, Vol. 367, No. 6481, pp. 1026–1030.

Sirignano, J. and Spiliopoulos, K., 2018. "Dgm: A deep learning algorithm for solving partial differential equations". *Journal of computational physics*, Vol. 375, pp. 1339–1364.

Thompson, R.L., Sampaio, L.E.B., de Bragança Alves, F.A., Thais, L. and Mompean, G., 2016. "A methodology to evaluate statistical errors in dns data of plane channel flows". *Computers & Fluids*, Vol. 130, pp. 1–7.

Wang, J.X., Wu, J.L. and Xiao, H., 2017. "Physics-informed machine learning approach for reconstructing reynolds stress modeling discrepancies based on dns data". *Physical Review Fluids*, Vol. 2, No. 3, p. 034603.

White, F.M. and Corfield, I., 2006. *Viscous fluid flow*, Vol. 3. McGraw-Hill New York.

Xiao, H., Wu, J.L., Laizet, S. and Duan, L., 2020. "Flows over periodic hills of parameterized geometries: A dataset for data-driven turbulence modeling from direct simulations". *Computers & Fluids*, Vol. 200, p. 104431.

7. RESPONSIBILITY NOTICE

The author(s) is (are) solely responsible for the printed material included in this paper.