COBEM 2023 27TH INTERNATIONAL CONGRESS OF MECHANICAL ENGENEERING

ABCM Brazilian Society of Mechanical Sciences and Engineering
27th ABCM International Congress of Mechanical Engineering
December 04-08, 2023, Florianópolis-SC, Brazil

COB-2023-1690
Pore-scale flow prediction using physics informed neural networks

Pedro Calderano
Helon Ayala
Marcio Carvalho
Pontifical Catholic University of Rio de Janeiro
pedro.calderano@lmmp.mec.puc-rio.br
helon@puc-rio.br
msc@puc-rio.br

Abstract. *Numerical simulations are employed to study the behavior of different dynamical systems. They are usually based on the solution of a set of differential equations that models the system. Although numerical simulation may provide detailed information about the system of interest, the traditional approach consumes large amounts of computational resources. Surrogate models are a manner to circumvent the referred time-consuming characteristic and provide quick response solutions. Physical Informed Neural Networks (PINNs) is a numerical method recently developed that can attach information from a system of partial differential equations in the loss function of a neural network. As trained neural networks provide an almost instantaneous answer and PINN formulation includes information from well-established equations, PINNs can be used as accurate surrogate models. In the present work, we use PINNs to predict steady-state velocity field in the pore space of three different porous media geometry configurations. The proposed method solves the flow that results from a constant pressure differential applied to the porous media. First, we implement the basic PINN formulation to predict pressure distribution and the velocity of the flow passing through a couple of simple geometries. After that, we implement a PINN to a more complex geometry, which requires a Fourier transformation in its input space to provide an appropriate solution. The results show that PINNs can quickly predict the discussed flow cases with small errors when compared to the flow solutions obtained by the Finite Element Method. We also observe that the PINNs inference stage consumes less computational resources when compared to the direct numerical simulation.*

Keywords: *machine learning, Physics-informed neural networks, porous medium*

1. INTRODUCTION

Flow through porous media is encountered in many natural phenomena and engineering applications such as in paper industries (Masoodi *et al.* (2012)) , petroleum reservoir engineering (Abd and Abushaikha (2021)), hydrogeology (Koohbor *et al.* (2020)), chemical engineering, and biomedical engineering (Miguel (2011)). Porous media present a heterogeneous structure (solid and void distribution in the pore scale) that leads to variability in their macroscopic properties (Bear (2018)). Therefore, there is no simple manner to define the porous medium macroscopic properties, such as permeability. These properties are usually determined either through numerical simulations of the flow in the pore space or experimental measurements.

Although numerical simulations provide detailed information about a system, these simulations frequently consume high amounts of computational resources (Alizadeh *et al.* (2020); Asher *et al.* (2015)). Therefore, the development of surrogate or proxy models providing faster solutions and allocating less computational resources than direct numerical simulations is becoming popular (Razavi *et al.* (2012)). However, often the surrogate answers present lower fidelity as well. These models are based on function approximation statistical methods, mostly located in the Machine Learning sphere. Methods such as Support Vector Machines and Neural Networks can receive a large amount of data to approximate a representative surface that can be later used to infer the function value quickly and consuming low resources. Therefore surrogate models become necessary for real-time applications, e.g., system control, and can save time for other cases that require variable assessment to support decision making.

Raissi *et al.* (2019) introduced the Physics Informed Neural Networks (PINNs). They adapted automatic differentiation (AutoDiff) methods (Margossian (2019)) into the Neural Network operation. The AutoDiff uses the chain rule to compute the gradient of the math operations performed until it reaches the final value concerning the designated input. These derivatives allow the incorporation of established physical equations governing a system into the Networks. As Neural Networks fit functions according to statistical concepts, the equations introduced serve as a regularizer (Raissi *et al.* (2019); Géron (2022)), avoiding curve misfits. The direct equation information added in the PINN training expanded the type of problems Neural Networks can tackle. Traditionally, Neural Networks were restricted to regression and classification problems. PINNs can perform direct simulations. They also can associate the forward problem with the inverse. So it is possible to regress equation parameters from the data using the provided equations. PINNs demonstrate

to be slower converging than traditional direct numerical methods to solve forward problems. However, they show to be very efficient to perform parameter regression.

Adaptations to the original or vanilla PINN formulation were introduced through works to deal with their incapacity to solve the equations with certain domain dispositions. Jagtap *et al.* (2020); Kharazmi *et al.* (2019, 2021) and Jagtap and Karniadakis (2021) proposed domain decomposition formulations. Wang *et al.* (2021) proposed the introduction of Fourier feature transformation to improve the approximation of oscillatory functions. These proposed extensions to the original algorithm improve the general PINN formulation, expanding the algorithm's versatility.

This work proposes the use of PINNs to solve the flow equations in porous medium geometries considering a forward problem. We show that these methods can be effective to deal with pore-scale flow situations.

The remainder of this paper is organized as follows. Section 2presents the numerical methods employed in this work; Section 3addresses the results, and Section 4states our main conclusions regarding the proposal.

2. NUMERICAL METHODS

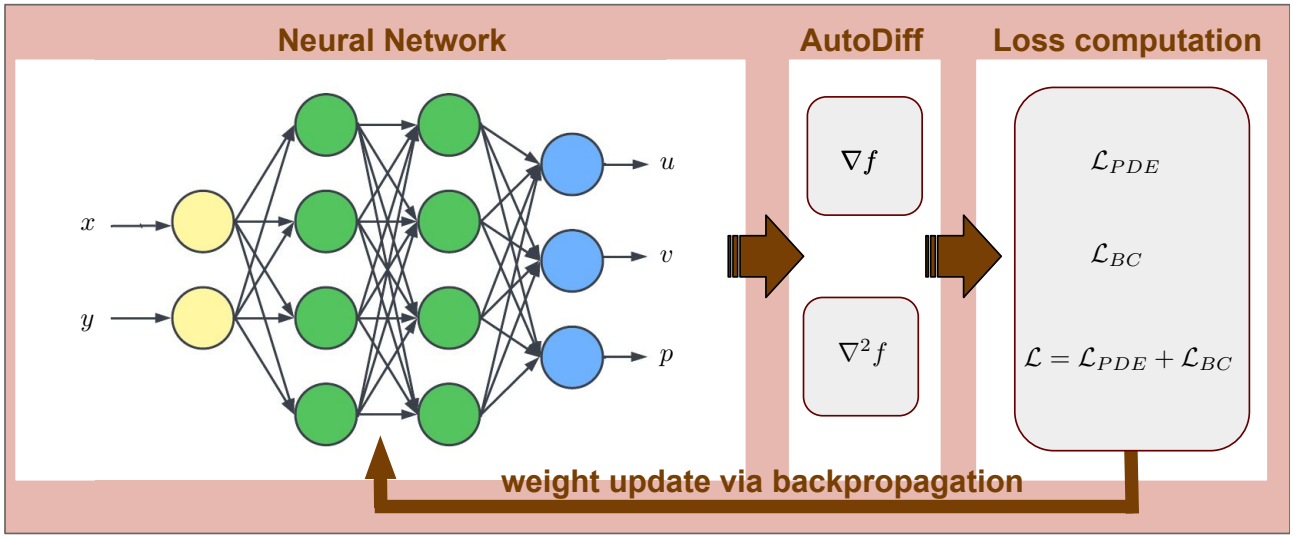


Figure 1. Scheme of Physics Informed Neural Network (PINN). The neural network has spatial coordinates as input and outputs the flow velocity and pressure fields. During the training process, an automatic differentiation algorithm computes the derivatives of the output variables regarding the input variables. Considering these derivatives, the loss function that recovers the differential equation system is computed and employed to update the network weights.

In this work, we apply PINNs to solve a fluid flow problem. The difference between PINNs and conventional neural networks is in the definition of the loss function (Raissi *et al.* (2019)). The loss function in a PINN considers the differential equations governing the problem of interest. Therefore, the training stage in a PINN network does not necessarily require labeled data, entering in the direct simulation field (Cai *et al.* (2021); Karniadakis *et al.* (2021)). Figure 1 shows the schematic of the training procedure of a PINN. First, an inference of the system variable from the domain variable using the designed network. An automatic differentiation algorithm (AutoDiff), which computes the derivatives of the operations computed by an algorithm (Baydin *et al.* (2018); Margossian (2019)), watches the Neural Network operations that produce its inference. Then, the AutoDiff can compute the derivatives of the predicted variables concerning input derivatives, i.e. the coordinate derivatives of the state variables. Hence, the residual of the differential equation system that describes the process may be computed. The loss function residual minimized during the network training process is the computed residual. The loss is composed only of the Partial Differential Equations (PDEs) component ($\mathcal{L}_{PDE}$), the boundary conditions components ($\mathcal{L}_{BC}$), and the initial conditions ($\mathcal{L}_{IC}$) in the purely forward problem. In the case in which the inverse problem is considered, the traditional loss relative to data of a known dataset (Géron (2022)) is also computed ($\mathcal{L}_{Data}$). As this work considers a steady-state purely forward problem, there is not any $\mathcal{L}_{IC}$ and $\mathcal{L}_{Data}$. The network weight update procedure follows the standard backpropagation algorithm (Koutroumbas and Theodoridis (2008)).

This work considers the low Reynolds approximation of the Navier-Stokes equations known as Stokes' equations (Panton (2013)). The low Reynolds approximation considers the convective terms negligible. Therefore, the $\mathcal{L}_{PDE}$ are computed using the relations presented in Equations 1 - 3. No-slip, inlet, and outlet boundary conditions are established as well, presented in Equations 4 - 7. Subsection 2. lpresents the domain shapes investigated and its respective boundary conditions. In these equations, x and y are the domain coordinates, u and v are the velocity field components, p is the pressure variable, P_{in} is the pressure established at the domain inlet, P_{out} the pressure established at the domain outlet, μ

the fluid viscosity. MSE is an error function (Mean Squared Error), in which it squares the residual for all points computed, sums all squared residuals, and then divides by the number of samples, as presented in Equation 9. Equation 8 presents the total loss to be minimized by the training procedure. The total loss is a weighted sum with weights represented by λ. The loss equations are summarized as

$$\mathcal{L}_{PDE_1} = MSE\left(-\frac{\partial p}{\partial x} + \mu\left(\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2}\right)\right), \tag{1}$$

$$\mathcal{L}_{PDE_2} = MSE\left(-\frac{\partial p}{\partial y} + \mu\left(\frac{\partial^2 v}{\partial x^2} + \frac{\partial^2 v}{\partial y^2}\right)\right), \tag{2}$$

$$\mathcal{L}_{PDE_3} = MSE\left(\frac{\partial u}{\partial x} + \frac{\partial v}{\partial y}\right), \tag{3}$$

$$\mathcal{L}_{BC_{noslip}} = MSE\left(u - 0\right), \tag{4}$$

$$\mathcal{L}_{BC_{noslip}} = MSE\left(v - 0\right), \tag{5}$$

$$\mathcal{L}_{BC_{inlet}} = MSE\left(p - P_{in}\right), \tag{6}$$

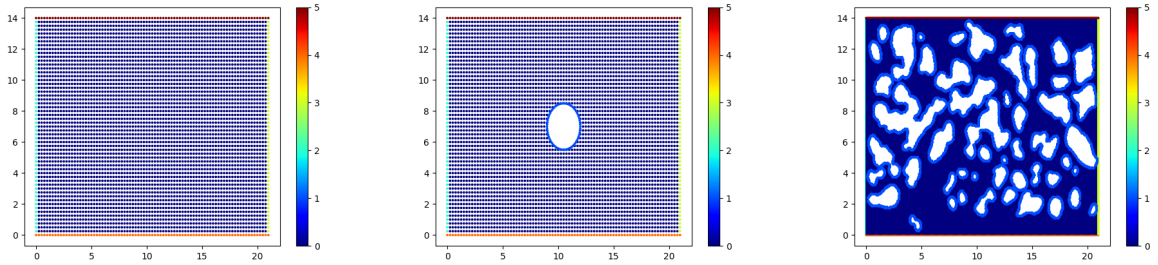
$$\mathcal{L}_{BC_{outlet}} = MSE\left(p - P_{out}\right), \tag{7}$$

$$\mathcal{L} = \lambda_1\mathcal{L}_{PDE_1} + \lambda_2\mathcal{L}_{PDE_2} + \lambda_3\mathcal{L}_{PDE_3} + \lambda_4\mathcal{L}_{BC_{noslip}} + \lambda_5\mathcal{L}_{BC_{inlet}} + \lambda_6\mathcal{L}_{BC_{outlet}}, \tag{8}$$

$$MSE\left(f\right) = \frac{1}{N}\sum_{i=0}^{N} f_i^2. \tag{9}$$

2.1 The geometries

In this work, we explore three geometries in which fluid flow goes through. The dimensions of the domain are 21 [mm] wide and 14 [mm] tall, which may provide an analysis of pore scale flow characteristics. Figure 2 exhibits these geometries. The Geometry I, in Figure 2a, has no obstacles, therefore recovering a Hagen-Poiseuille flow between parallel plates. Figure 2b, displays the Geometry II, which has a circular obstacle. Finally, Figure 2c shows a more complex porous media geometry, Geometry III, with multiple solid grains throughout the domain. The domains exhibited have different colors to the points where boundary conditions are applied. The obstacles border, the top, and the bottom wall have no-slip boundary conditions imposed, Equations 4 and 5. We impose no vertical velocity at the inlet and outlet of the domains as well, then Equation 5 is imposed at the extremities besides Equation 6 or 7. As the variance of the flow through Geometry III is higher than in the other Geometries, the Geometry III solution requires to place more collocation points (the points where the PDE losses are computed) to compute its solution. Therefore, point density at the last Geometry is higher, which forms an apparent continuous domain in Figure 2c due to image size.



(a) Geometry I (b) Geometry II (c) Geometry III

Figure 2. Schematic of the geometries explored in this work. The colors of the points indicate the equations considered in that point's total loss. The darkest blue, index 0, only considers the $\mathcal{L}_{PDE}$s in its loss. The indexes 1, 2, 3, 4, and 5 consider the $\mathcal{L}_{BC_{noslip}}$, $\mathcal{L}_{BC_{inlet}}$, $\mathcal{L}_{BC_{outlet}}$, $\mathcal{L}_{BC_{noslip}}$, and $\mathcal{L}_{BC_{noslip}}$ respectively. The points indexes are exhibited in the color bar.

2.2 Fourier feature PINN (FFPINN)

Although Neural Networks have long been considered universal function approximators, they struggle to approximate high-frequency functions (Basri *et al.* (2020); Rahaman *et al.* (2019)), due to the spectral bias. Moreover, the PINNs training procedure is difficult (Wang *et al.* (2021, 2022)). Tancik *et al.* (2020) applied NTK theory (Jacot *et al.* (2018)) to improve their Neural Networks at high-frequency related tasks. Wang *et al.* (2022) adapts NTK theory into PINNs, and Wang *et al.* (2021) proposes the NTK Kernel as a manner to transform the domain through the extraction Fourier features. Multiple frequency features may be extracted, which provides the network with different oscillatory attributes. Then, the network may learn behaviors happening in the target function at different rates easier in case suited features to each behavior are obtained in the input. Wang *et al.* (2021) showed that Fourier feature Physics Informed Neural Networks (FFPINN) reaches a proper solution to oscillatory fields.

The Fourier features are extracted using the following relation presented in Equation 10, in which γ is the extracted Fourier features. $\mathbf{B}$ is a square matrix of the same size as the number of dimensions. The $\mathbf{B}$ elements are randomly sampled from a Gaussian distribution with $\mu = 0$ and σ^2 chosen by the user, a hyperparameter. Higher σ^2 values make it easier for the network to determine higher frequency behaviors. Then, if various oscillatory patterns appear in the approximated function, the network ideally would be designed with an appropriate number of Fourier feature maps to speed up convergence. However, the number of characteristics is designated by the user, just as σ^2. Then, it should be tried different configurations to select a reasonable configuration. If more than one Fourier map transformation happens to perform the training, each of these maps is fed into separate parallel networks that are merged. One last layer is added to predict the network output variables. The Fourier features are given by

$$\gamma\left(\mathbf{X}\right) = \begin{bmatrix} cos(\mathbf{BX}) \\ sin(\mathbf{BX}) \end{bmatrix}, \tag{10}$$

in which $\mathbf{X}$ are the point coordinates. The $\gamma\left(\mathbf{X}\right)$ is the input of the FFPINN formulation, just as $\mathbf{X}$ is the input of the PINN formulation.

3. RESULTS

We investigate the use of PINNs to predict flow in porous media like geometries as shown in Subsection 2.1 The predicted solution is obtained in a purely forward fashion, with no aid of exogenous measurements. Then, the loss function applied to tune the network weights are only relative to the PDE and the boundary conditions, as shown in Equation 8. The velocity and pressure fields obtained by the PINNs were compared with established solutions. As the Geometry I, presented in Figure 2a, does not present obstacles to disturb the flow, its solution should approximate the Hagen-Poiseuille exact solution. This exact solution considers null vertical velocity, $v = 0$, as a result of the boundary conditions in the continuity equation for an incompressible fluid. Then, the solution for the velocity field in the horizontal direction, u, according to

$$u(y) = \frac{1}{2\mu}\frac{\partial P}{\partial x}\left(y^2 - dy\right). \tag{11}$$

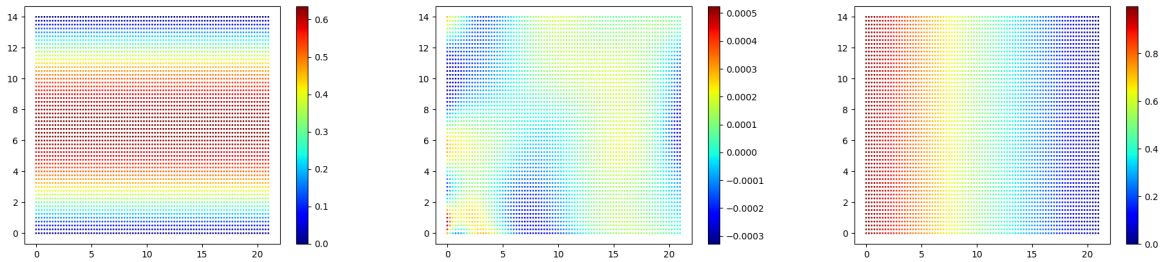
The remaining geometries, II and III, presented in Figures 2b and 2c, had their PINN solution compared to Finite Element Method (FEM) solutions. The FEM solutions used Taylor-Hood triangular elements with size 2×10^{-4}. The solutions are

mesh-independent. The solution relative tolerance and absolute tolerance are 10^{-11} and 10^{-14}, respectively. The Finite Element Method was employed using the FEniCS Project library (Logg and Wells (2010)). The vanilla PINN formulation is used to solve the geometries I and II, presented in Figures 2a and 2b. Therefore, the solution of the standard Network design solves the flow field for these simple geometries. However, the more complex Geometry III, presented in Figure 2c, could not be solved using this formulation. The high density of no-slip boundary conditions in the middle of the domain forces the vanilla PINN to approximate a trivial solution for the velocity field, not mimicking the actual velocity behavior. As the magnitude of this field oscillates according to the flow paths formed by the grains, we applied the FFPINN formulation to handle this wavy attribute. The hyperparameters of the trained networks are as described in Table 1. The hyperbolic tangent was the chosen activation function. The optimization algorithm employed during the training phase is the Adam considering 10000 training epochs and learning rate $lr = 0.0035$.

	$\#layers$	$\#neurons$	σ_1	σ_2
Geometry I	5	50	–	–
Geometry II	8	50	–	–
Geometry III	8	50	1/17.5	10/17.5

Table 1. Hyperparameters adopted in the networks trained. σ_1 and σ_2 are parameters in the FFPINN formulation. As Geometries I and II are developed using the vanilla PINN formulation, these parameters are not defined for these geometries.

Figure 3 exhibits the fields computed for Geometry I. The vertical component of the PINN solution velocity field is in the $v = 0$ vicinity, as it is supposed to be. Figure 4 presents the values of the flow fields considering vertical and horizontal cuts, so it is possible to plot together profiles of the exact solution, Equation 11, and the proposed solution. The error in the x-direction component of the predicted velocity field is at least of order $\mathcal{O}(3)$ smaller than the order of the actual solution. The pressure field, Figure 3c, varies linearly in the x-direction. Figures 5 and 6 present the flow and pressure fields predicted by the PINN algorithm, the fields solved by FEM, and their difference considering geometries II and III. The difference was computed in the FEM nodal points. The nodal points' coordinates were taken as input to the trained PINN to compute the flow field at these places. Figures 5g, 5h, 5i, 6g, 6h, 6i show the difference between the PINN inference and the FEM solution. The error presented for Geometry III is relatively high. However, we did not perform an extensive exploration of the hyperparameter space. As the FEM solution has a great resemblance with the FFPINN solution, it indicates that a more meticulous hyperparameter exploration would provide an answer with a lower discrepancy.

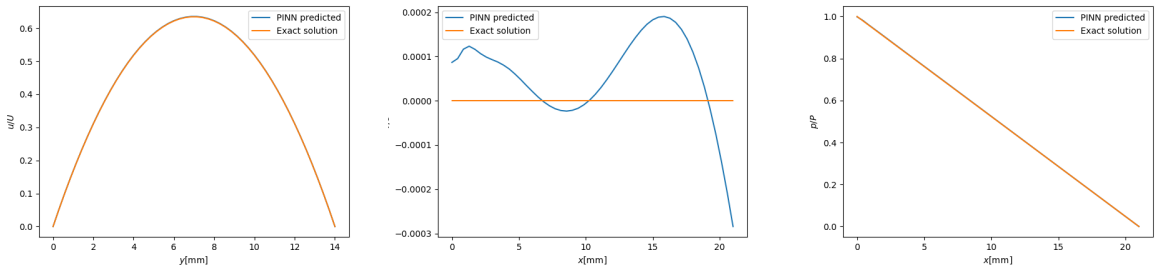


(a) velocity field (x-direction)

(b) velocity field (y-direction)

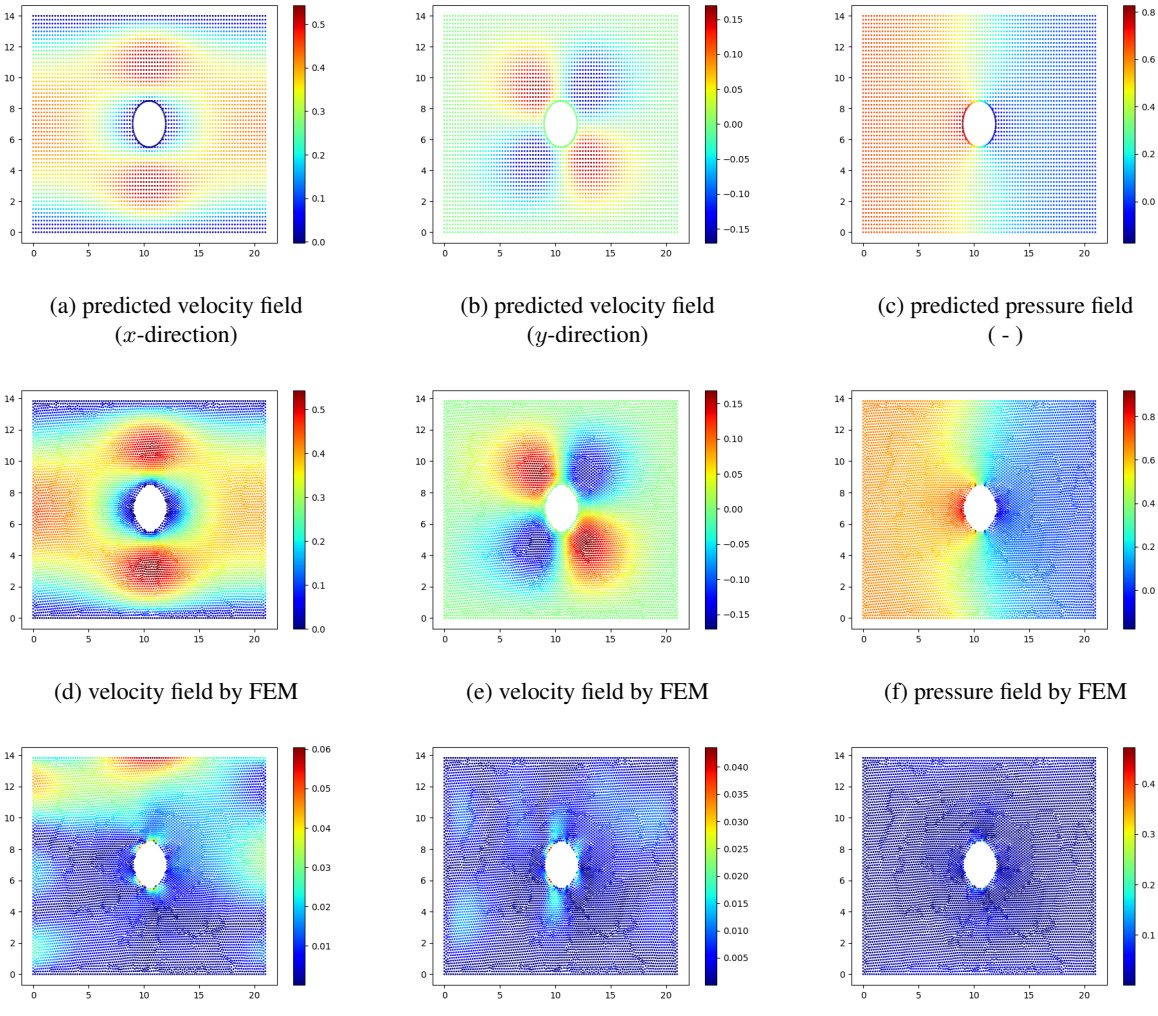
(c) pressure field (-)

Figure 3. PINN simulation results for the Hagen-Poiseuille flow.



(a) x-direction velocity profile (b) y-direction velocity profile (c) pressure profile

Figure 4. The images present the exact solution and the PINN prediction for the velocity and pressure profile in a Hagen-Poiseuille flow, in (a), and it is exhibited u in a vertical cut, in (b) v in a horizontal cut, and in (c) p in a horizontal cut.

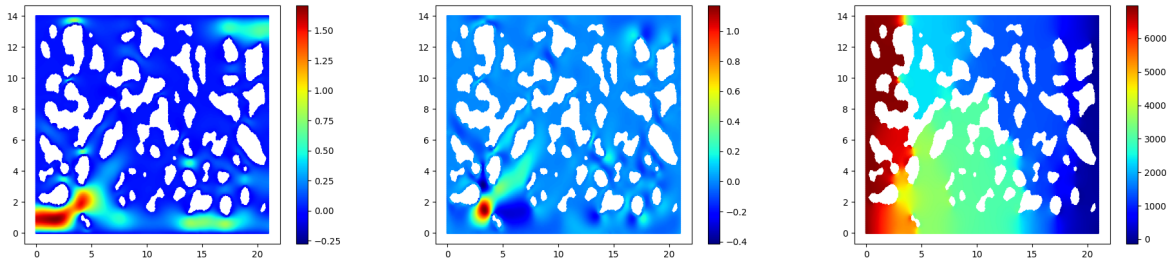


(a) predicted velocity field (x-direction) (b) predicted velocity field (y-direction) (c) predicted pressure field (-)

(d) velocity field by FEM (e) velocity field by FEM (f) pressure field by FEM

(g) absolute difference (h) absolute difference (i) absolute difference

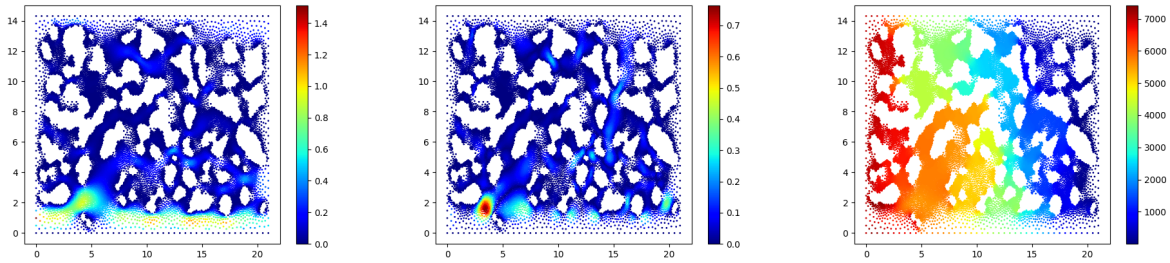
Figure 5. The images present the the PINN prediction and its absolute error in relation to a FEM prediction for the flow in a porous media with a single no flow region on its center.



(a) predicted velocity field (x-direction)

(b) predicted velocity field (y-direction)

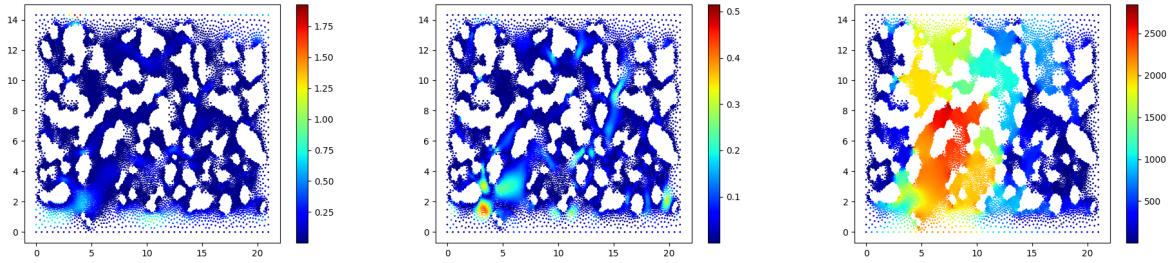
(c) predicted pressure field (-)



(d) velocity field by FEM

(e) velocity field by FEM

(f) pressure field by FEM



(g) absolute error

(h) absolute error

(i) absolute error

Figure 6. The images present the the FFPINN prediction and its absolute error in relation to a FEM prediction for the flow in a porous media.

Table 2 presents the Reynolds number at the domain outlet for the studied flows. Therefore, the Reynolds number considered is computed according to

$$Re = \frac{\rho \overline{v} D}{\mu}, \tag{12}$$

where $\overline{v}$ is the mean velocity computed at the domain outlet and D is the domain height. Table 2 also shows the time taken to perform the PINN methods inference and the FEMs solution. As expected, the PINN inference times are shorter than those of the FEM solutions. While the flow in question is steady-state, the difference between the FEM and PINN solutions is likely to be more substantial in transient cases.

	Geometry I	Geometry II	Geometry III
Re	59.3	41.4	13.0
t_{PINN} (s)	0.020	0.025	0.023
t_{FEM} (s)	–	3.73	4.51

Table 2. Reynolds number at the geometry outlet and solution execution time. t_{PINN} is the trained Network inference time while t_{FEM} is the time it took the Finite Element Method to achieve the problem solution.

4. CONCLUSION

This work used Physics Informed Neural Networks to solve flow problems in geometries on the pore scale. We showed that the vanilla PINN (Raissi *et al.* (2019)) can solve the forward problem for simple geometries. However, the vanilla PINN formulation cannot solve the problem when more complex geometry that makes the solution fields oscillatory is considered. The more complex geometry fields were solved using the Fourier feature PINN (FFPINN) (Wang *et al.* (2021)). Although the absolute error presented for the FFPINN solution is relatively high, the solution hyperparameters were not extensively explored. As the solution presented resembles the proper solution, it indicates that a thorough hyperparameter exploration would arrive at a low-error solution. This shows that the PINN formulation has the potential further investigate pore-scale flows through porous media. In future works, we intend to investigate two-phase flow using PINNs.

5. ACKNOWLEDGEMENTS

The authors would like to thank the financial support from the National Agency of Petroleum, Natural Gas and Biofuels – ANP, from Petrobras, and from the Brazilian National Council for Scientific and Technological Development – CNPq.

6. REFERENCES

Abd, A.S. and Abushaikha, A.S., 2021. "Reactive transport in porous media: a review of recent mathematical efforts in modeling geochemical reactions in petroleum subsurface reservoirs". *SN Applied Sciences*, Vol. 3, pp. 1–28.

Alizadeh, R., Allen, J.K. and Mistree, F., 2020. "Managing computational complexity using surrogate models: a critical review". *Research in Engineering Design*, Vol. 31, pp. 275–298.

Asher, M.J., Croke, B.F., Jakeman, A.J. and Peeters, L.J., 2015. "A review of surrogate models and their application to groundwater modeling". *Water Resources Research*, Vol. 51, No. 8, pp. 5957–5973.

Basri, R., Galun, M., Geifman, A., Jacobs, D., Kasten, Y. and Kritchman, S., 2020. "Frequency bias in neural networks for input of non-uniform density". In *International Conference on Machine Learning*. PMLR, pp. 685–694.

Baydin, A.G., Pearlmutter, B.A., Radul, A.A. and Siskind, J.M., 2018. "Automatic differentiation in machine learning: a survey". *Journal of Marchine Learning Research*, Vol. 18, pp. 1–43.

Bear, J., 2018. *Modeling phenomena of flow and transport in porous media*, Vol. 1. Springer.

Cai, S., Mao, Z., Wang, Z., Yin, M. and Karniadakis, G.E., 2021. "Physics-informed neural networks (pinns) for fluid mechanics: A review". *Acta Mechanica Sinica*, Vol. 37, No. 12, pp. 1727–1738.

Géron, A., 2022. *Hands-on machine learning with Scikit-Learn, Keras, and TensorFlow*. " O'Reilly Media, Inc.".

Jacot, A., Gabriel, F. and Hongler, C., 2018. "Neural tangent kernel: Convergence and generalization in neural networks". *Advances in neural information processing systems*, Vol. 31.

Jagtap, A.D. and Karniadakis, G.E., 2021. "Extended physics-informed neural networks (xpinns): A generalized space-time domain decomposition based deep learning framework for nonlinear partial differential equations." In *AAAI Spring Symposium: MLPS*. pp. 2002–2041.

Jagtap, A.D., Kharazmi, E. and Karniadakis, G.E., 2020. "Conservative physics-informed neural networks on discrete domains for conservation laws: Applications to forward and inverse problems". *Computer Methods in Applied Mechanics and Engineering*, Vol. 365, p. 113028.

Karniadakis, G.E., Kevrekidis, I.G., Lu, L., Perdikaris, P., Wang, S. and Yang, L., 2021. "Physics-informed machine learning". *Nature Reviews Physics*, Vol. 3, No. 6, pp. 422–440.

Kharazmi, E., Zhang, Z. and Karniadakis, G.E., 2019. "Variational physics-informed neural networks for solving partial differential equations". *arXiv preprint arXiv:1912.00873*.

Kharazmi, E., Zhang, Z. and Karniadakis, G.E., 2021. "hp-vpinns: Variational physics-informed neural networks with domain decomposition". *Computer Methods in Applied Mechanics and Engineering*, Vol. 374, p. 113547.

Koohbor, B., Fahs, M., Hoteit, H., Doummar, J., Younes, A. and Belfort, B., 2020. "An advanced discrete fracture model for variably saturated flow in fractured porous media". *Advances in Water Resources*, Vol. 140, p. 103602.

Koutroumbas, K. and Theodoridis, S., 2008. *Pattern recognition*. Academic Press.

Logg, A. and Wells, G.N., 2010. "Dolfin: Automated finite element computing". *ACM Transactions on Mathematical Software (TOMS)*, Vol. 37, No. 2, pp. 1–28.

Margossian, C.C., 2019. "A review of automatic differentiation and its efficient implementation". *Wiley interdisciplinary reviews: data mining and knowledge discovery*, Vol. 9, No. 4, p. e1305.

Masoodi, R., Tan, H. and Pillai, K.M., 2012. "Numerical simulation of liquid absorption in paper-like swelling porous media". *AIChE journal*, Vol. 58, No. 8, pp. 2536–2544.

Miguel, A.F., 2011. "Lungs as a natural porous media: architecture, airflow characteristics and transport of suspended particles". In *Heat and mass transfer in porous media*, Springer, pp. 115–137.

Panton, R.L., 2013. *Incompressible flow*. John Wiley & Sons.

Rahaman, N., Baratin, A., Arpit, D., Draxler, F., Lin, M., Hamprecht, F., Bengio, Y. and Courville, A., 2019. "On the spectral bias of neural networks". In *International Conference on Machine Learning*. PMLR, pp. 5301–5310.

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.

Razavi, S., Tolson, B.A. and Burn, D.H., 2012. "Review of surrogate modeling in water resources". *Water Resources Research*, Vol. 48, No. 7.

Tancik, M., Srinivasan, P., Mildenhall, B., Fridovich-Keil, S., Raghavan, N., Singhal, U., Ramamoorthi, R., Barron, J. and Ng, R., 2020. "Fourier features let networks learn high frequency functions in low dimensional domains". *Advances in Neural Information Processing Systems*, Vol. 33, pp. 7537–7547.

Wang, S., Wang, H. and Perdikaris, P., 2021. "On the eigenvector bias of fourier feature networks: From regression to solving multi-scale pdes with physics-informed neural networks". *Computer Methods in Applied Mechanics and Engineering*, Vol. 384, p. 113938.

Wang, S., Yu, X. and Perdikaris, P., 2022. "When and why pinns fail to train: A neural tangent kernel perspective". *Journal of Computational Physics*, Vol. 449, p. 110768.

7. RESPONSIBILITY NOTICE

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