Physics Informed Neural Networks for Fluid Dynamics

Numerical Analysis for PDEs Course Project

Giulia Mescolini, Luca Sosta

Politecnico di Milano

08/02/2022

Tutors: Stefano Pagani, Francesco Regazzoni

POLITECNICO MILANO 1863

Introduction

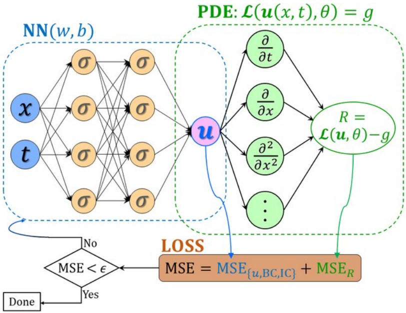
The idea

Can Machine Learning-based techniques be exploited to study phenomena regulated by Partial Differential Equations?

In this project, we studied with *Physics Informed Neural Networks* (PINNs) several fluid-dynamics test cases governed by the Navier-Stokes equations.

PINNs - General Structure

With PINNs, the information of physical laws can be taken into account in a deep learning framework.



They consist in *Artifical Neural Networks* trained to solve supervised learning tasks, while respecting some given partial differential equations.

PINNs and Data Availability

Differently from the classical Neural Networks setting, with PINNs we often are in a *small data* regime, when few measures are available.

Physical quantities such as pressures, indeed, may be costly and difficult to evaluate.

PINNs - Optimization Problems

PINNs training can be formulated as an *optimization problem*, with unknowns represented by the weights and the biases of the network, summarized in the variable $\mathbf{W}$.
The optimization problem reads as follows:

$$\begin{cases} \min\limits_{u,\mathbf{W}} \mathcal{J}(u, \mathbf{W}) \\ s.t. \;\; u(\mathbf{x}, t) = \mathcal{N}\mathcal{N}(\mathbf{x}, t; \mathbf{W}) \\ \text{where } \mathcal{J}(u, \mathbf{W}) = \dfrac{\lambda_{obs}}{N_{obs}} \displaystyle\sum_{i=1}^{N_{obs}} (u(\mathbf{x}_i, t_i) - y_i)^2 + \lambda_{res} R(u, \mathbf{W}) \end{cases}$$

PINNs - Point Sets

- **Collocation Points**: inside the domain, where we impose the PDE constraint.
- **Boundary Points**: on the domain boundary, where we impose Dirichlet or Neumann Boundary Conditions.
- **Fitting Points**: inside the domain, where we impose the measured (or the exact/numerical with noise) value of the solution.

First Test Cases

We have first applied PINNs to test cases in which an analytical solution is known.

We have analyzed two meaningful examples of fluid-dynamics problems:

1. *Poiseuille Flow*
2. *Colliding Flows*

Poiseuille Flow

We have studied the laminar flow in a channel, regulated by the equations:

$$
\begin{cases}
\frac{\partial u}{\partial x} + \frac{\partial v}{\partial y} = 0 & \text{in } \Omega \\
\rho \left(u \frac{\partial u}{\partial x} + v \frac{\partial u}{\partial y} \right) + \frac{\partial p}{\partial x} - \mu \Delta u = 0 & \text{in } \Omega \\
\rho \left(u \frac{\partial v}{\partial x} + v \frac{\partial v}{\partial y} \right) + \frac{\partial p}{\partial y} - \mu \Delta v = 0 & \text{in } \Omega \\
u = v = 0 & \text{on } \Gamma_{D1} \\
u = u_{ex}, v = v_{ex} & \text{on } \Gamma_{D2} \\
\rho \frac{\partial u}{\partial x} + p = p_{end} & \text{on } \Gamma_N \\
\frac{\partial v}{\partial x} = 0 & \text{on } \Gamma_N
\end{cases}
$$

Poiseuille Flow

Goals

- approach the study of differential systems with PINNs
- implement losses associated to PDEs and BCs of type Dirichlet and Neumann, which appeared in the following more complex test cases, too
- strategy for data normalization

Poiseuille Flow

We have formulated the PINN training as an optimization problem by including in the regularization term R losses related to:

- Partial Differential Equations:

$$\mathcal{J}_{col} = \frac{1}{N_{col}} \sum_{n=1}^{N_{col}} [\mathcal{R}_{mass}(x_n, y_n)^2 + \mathcal{R}_{mom_x}(x_n, y_n)^2 + \mathcal{R}_{mom_y}(x_n, y_n)^2]$$

- Boundary Conditions:

$$\mathcal{J}_{bd} = \frac{1}{N_{BCD}} \sum_{n=1}^{N_{BCD}} [\mathcal{R}_{dir_x}(x_n, y_n)^2 + \mathcal{R}_{dir_y}(x_n, y_n)^2 + \mathcal{R}_{neu_x}(x_n, y_n)^2 + \mathcal{R}_{neu_y}(x_n, y_n)^2]$$

- Measures of velocities inside the domain:

$$\mathcal{J}_{fit} = \frac{1}{N_{fit}} \sum_{n=1}^{N_{fit}} [\mathcal{R}_{fit_u}(x_n, y_n)^2 + \mathcal{R}_{fit_v}(x_n, y_n)^2 + \mathcal{R}_{fit_p}(x_n, y_n)^2]$$

Results

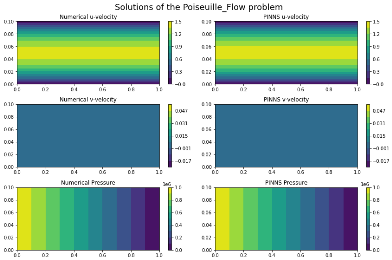


Figure 1: Exact vs PINN solution.

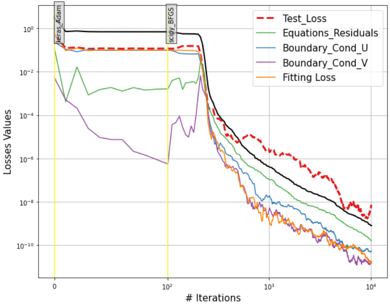


Figure 2: Loss trends.

Colliding Flows

The second test case involves two colliding flows in a square, and it is regulated by the system:

$$\begin{cases} \frac{\partial u}{\partial x} + \frac{\partial v}{\partial y} = 0 & \text{in } \Omega \\ \frac{\partial p}{\partial x} - \Delta u = 0 & \text{in } \Omega \\ \frac{\partial p}{\partial y} - \Delta v = 0 & \text{in } \Omega \\ u = 20xy^3 & \text{on } \partial\Omega \\ v = 5x^4 - 5y^4 & \text{on } \partial\Omega \end{cases}$$

Colliding Flows

Goals

- introduction of noise
- development of strategies to deal with the identification of a unique pressure

Noise

In order to simulate a real case in which measures are affected by noise, we added to boundary and fitting data a *gaussian noise*, with $\sigma = 1$ and $\mu = 0$.

It can be then scaled by a factor, that we generally set to be $1 - 5\%$ of the physical quantity.

Pressure Determination

Since the problem is *fully Dirichlet*, pressure is involved only in the PDEs, where it appears through its gradient; this means that in general it is defined up to a constant.
The exact solutions are:

$$p_{ex} = 60x^2y - 20y^3 + C \quad u_{ex} = 20xy^3 \quad v_{ex} = 5x^4 - 5y^4$$

Pressure Determination

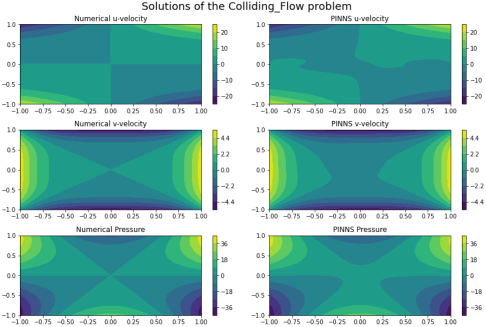
Let us reconstruct the solution with $C = 0$; we need to insert some additional information inside the PINN.

For this purpose, we developed two strategies:

- **One fitting point for pressure**
  Fixing a measure of the pressure value in a point inside the domain.
- **Pressure Mean Imposition Strategy**
  Implementing a new loss function, which penalizes deviations from zero of the mean value of pressure.

Comparison between exact and PINN solution

The reconstruction of the solution is accurate with both strategies. We report the one obtained with the *fitting strategy*, but the result is similar with *pressure mean imposition*.



Which strategy is more accurate? Let us compare the test losses.

Loss trends - Fitting Strategy

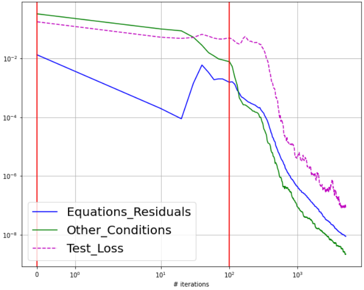


Figure 3: Case without noise.

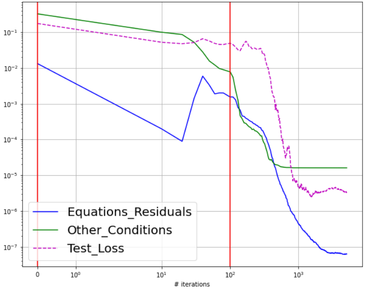


Figure 4: Case with noise.

Loss trends - Pressure Mean Imposition Strategy

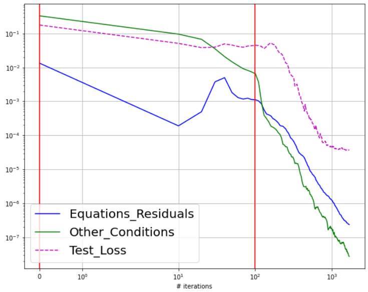


Figure 5: Case without noise.

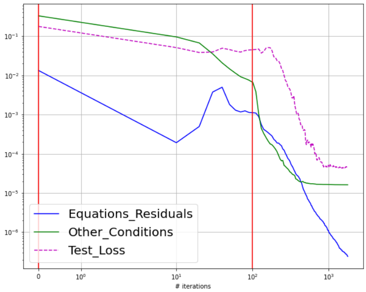


Figure 6: Case with noise.

Lid-Driven Cavity

Let us now move to a test case whose analytical solution is not known: the flow in a square cavity subject to a moving upper plate.

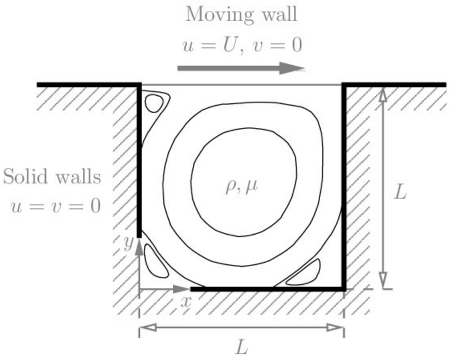


Figure 7: Geometry of the problem.

Lid-Driven Cavity

The equations describing the phenomenon read as:

$$
\begin{cases}
\frac{\partial u}{\partial x} + \frac{\partial v}{\partial y} = 0 & \text{in } \Omega \\
\frac{\partial u}{\partial t} + \left(u\frac{\partial u}{\partial x} + v\frac{\partial u}{\partial y} \right) + \frac{\partial p}{\partial x} - \Delta u = 0 & \text{in } \Omega \\
\frac{\partial v}{\partial t} + \left(u\frac{\partial v}{\partial x} + v\frac{\partial v}{\partial y} \right) + \frac{\partial p}{\partial y} - \Delta v = 0 & \text{in } \Omega \\
u = v = 0 & \text{on } \Gamma_{D1} \\
u = U, v = 0 & \text{on } \Gamma_{D2} \\
+ \text{ initial condition: } u, v, p = 0 \text{ in } \Omega
\end{cases}
$$

Lid-Driven Cavity

Goals

- create an interface between numerical solutions (used for testing and fitting) and the PINN
- adapt the method to non-stationary problems

Steady Case

If we drop the temporal derivative in the equations and the initial condition, we still have the issue that pressure is not uniquely determined.
We therefore imposed one measure on it inside the domain.

Results

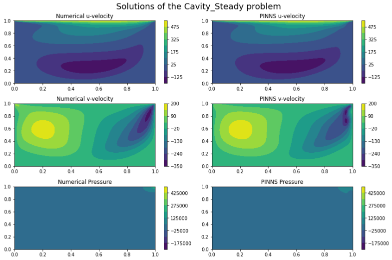


Figure 8: Numerical vs PINN solution.

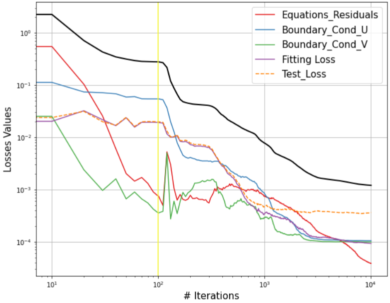


Figure 9: Loss trends.

Unsteady Case

The time variable has been introduced by building a 3D grid whose points are of the form (t, x, y).
The input layer, hence, has grown to 3 neurons, and we have rewritten the losses. Note that time has been considered as a third spatial coordinate.

Unsteady Case

In this case, we have been able to reconstruct the solution without fitting points, as the initial value for pressure enables to fix it uniquely.

Moreover, we analyzed the impact of a right choice of weights given to the physical losses. The optimal combination which leaded to the most accurate reconstruction is:

Loss	Weight
PDE_MASS	10^1
PDE_MOM_X	10^0
PDE_MOM_Y	10^0

Table 1: Increased weights for physical losses.

Results

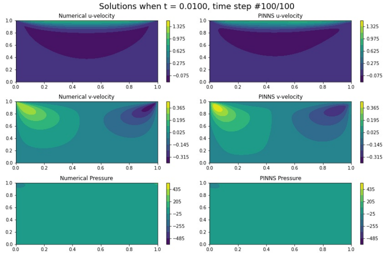


Figure 10: Solution (last time instant).

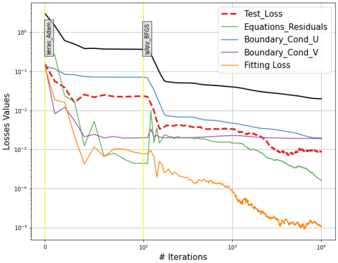


Figure 11: Loss trends.

Coronary Flow

The last test case analyzed is the blood flow in an artery affected by a *stenosis*.

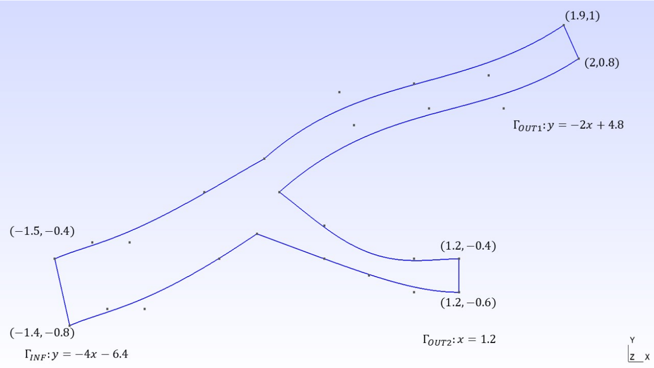


Figure 12: Geometry of the problem with specification of boundaries.

Coronary Flow

Introducing the quantities:

- $H = \sqrt{(-0.4+0.8)^2 + (-1.5+1.4)^2} cm = 0.41 cm$ (diameter of the principal vessel)
- $\theta = atan\left(\frac{1}{4}\right)$ (angle between the horizontal axis and the inflow)

The equations describing the problem read as:

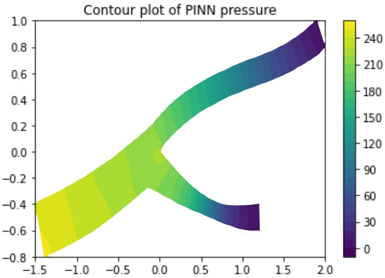
$$
\begin{cases}
\frac{\partial u}{\partial x} + \frac{\partial v}{\partial y} = 0 & \text{in } \Omega \\
\rho\left(u\frac{\partial u}{\partial x} + v\frac{\partial u}{\partial y}\right) + \frac{\partial p}{\partial x} - \mu\Delta u = 0 & \text{in } \Omega \\
\rho\left(u\frac{\partial v}{\partial x} + v\frac{\partial v}{\partial y}\right) + \frac{\partial p}{\partial y} - \mu\Delta v = 0 & \text{in } \Omega \\
u = v = 0 & \text{on } \Gamma_{WALL} \\
u = Ucos(\theta)\frac{\sqrt{((x-x_0)^2+(y-y_0)^2)}}{H}\left(1 - \frac{\sqrt{((x-x_0)^2+(y-y_0)^2)}}{H}\right) & \text{on } \Gamma_{INF} \\
v = Usin(\theta)\frac{\sqrt{((x-x_0)^2+(y-y_0)^2)}}{H}\left(1 - \frac{\sqrt{((x-x_0)^2+(y-y_0)^2)}}{H}\right) & \text{on } \Gamma_{INF} \\
\nu\frac{\partial \mathbf{u}}{\partial \mathbf{n}} + p\mathbf{n} = 0 & \text{on } \Gamma_{OUT1} \\
\nu\frac{\partial \mathbf{u}}{\partial \mathbf{n}} + p\mathbf{n} = 0 & \text{on } \Gamma_{OUT2}
\end{cases}
$$

Coronary Flow

The Neumann BC has enabled us to avoid the usage of fitting points for pressure, since it is correctly reconstructed using only few noisy measures for the velocities.

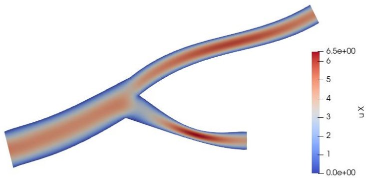
Comparison between numerical and PINN solution - Pressure

Figure 13: Numerical solution



Comparison between numerical and PINN solution - Velocity

Figure 14: Numerical solution



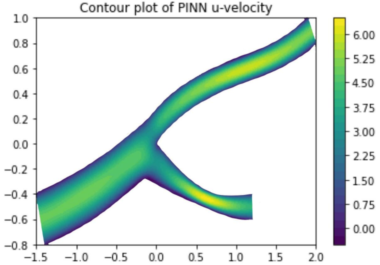
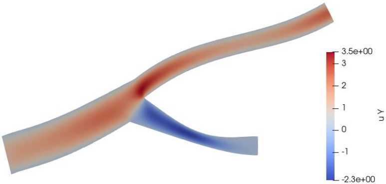
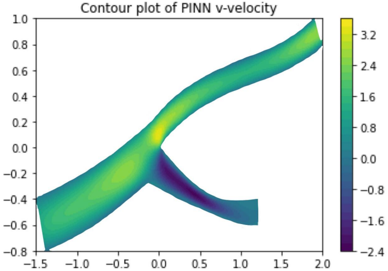
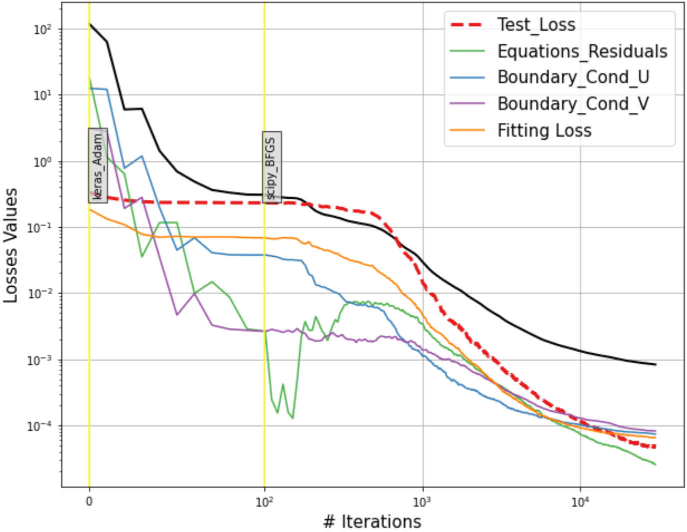


Figure 15: Numerical solution





Loss trends


Test_Loss
Equations_Residuals
Boundary_Cond_U
Boundary_Cond_V
Fitting Loss
keras_Adam
scipy_BFGS
Losses Values
Iterations
10^2
10^1
10^0
10^{-1}
10^{-2}
10^{-3}
10^{-4}
0
10^2
10^3
10^4

References I

[1] M. Raissi, A. Yazdani, and G. E. Karniadakis, *Hidden fluid mechanics: A Navier-Stokes Informed Deep Learning Framework for Assimilating Flow Visualization Data*, 2018. arXiv: 1808.04327 [cs.CE].

[2] D. E. Knuth, "Machine Learning and Deep Neural Networks Applications in Coronary Flow Assessment," *Journal of Thoracic Imaging*, vol. 35, S66–S71, 2020.

[3] S. Cai, Z. Mao, Z. Wang, M. Yin, and G. E. Karniadakis, *Physics-informed neural networks (PINNs) for fluid mechanics: A review*, 2021. arXiv: 2105.09506 [physics.flu-dyn].

[4] A. Quarteroni, *Modellistica Numerica per Problemi Differenziali*. Springer, 2012.

References II

[5] M. Raissi, P. Perdikaris, and G. E. Karniadakis, "Physics Informed Deep Learning (Part I & Part II): Data-driven solutions of nonlinear partial differential equations," *arXiv preprint arXiv:1711.10561*, 2017.

[6] Y. Ye, Y. Li, H. Fan, X. Liu, and H. Zhang, *Deep neural network methods for solving forward and inverse problems of time fractional diffusion equations with conformable derivative*, 2021. arXiv: 2108.07490 [math.NA].