An Approach to Solving Model Problems in Computational Fluid Dynamics Based on Physics-Informed Neural Networks

A.A. Tarasov, M.M. Stepanova

Saint Petersburg State University

DLCP 2026

07 July 2026

Introduction

- The long-term goal is to model heat transfer and fluid flow in cooling systems with complex geometries.
- Classical CFD and FEM-based approaches are reliable and widely used as reference methods. However, they require the construction of a high-quality computational mesh, which can be labor-intensive for complex geometries and may strongly affect the accuracy and cost of the simulation.
- Physics-Informed Neural Networks offer an alternative mesh-free approach: the governing equations and boundary/initial conditions are incorporated directly into the loss function using automatic differentiation.

PINNs: Advantages and Disadvantages

Advantages:

- Meshless
- Naturally incorporates experimental data
- No truncation errors when computing derivatives
- Robust and straightforward

Disadvantages:

- Computationally expensive
- Convergence not guarantied
- Limited theoretical basis
- Require careful hyperparameter tuning

Physics-Informed Neural Networks

1. **Boundary value problem**

$$\begin{cases} \mathbf{u}_t + \mathcal{N}[\mathbf{u}] = 0, & t \in [0,T], \quad \mathbf{x} \in \Omega, \\ \mathbf{u}(0,\mathbf{x}) = \mathbf{g}(\boldsymbol{x}), & t = 0, \quad \mathbf{x} \in \Omega, \\ \mathcal{B}[\mathbf{u}] = 0, & t \in [0,T], \quad \mathbf{x} \in \partial\Omega, \end{cases}$$

where $\mathcal{N}[\cdot]$ — differential operator, $\mathcal{B}[\cdot]$ — boundary operator

2. **Loss**

$$\mathcal{L}(\theta) = \mathcal{L}_{ic}(\theta) + \mathcal{L}_{bc}(\theta) + \mathcal{L}_r(\theta),$$

where

$$\mathcal{L}_{ic}(\theta) = \frac{1}{N_{ic}} \sum_{i=1}^{N_{ic}} \left| \mathbf{u}_\theta\left(0, \mathbf{x}_{ic}^i\right) - \mathbf{g}\left(\mathbf{x}_{ic}^i\right) \right|^2,$$

$$\mathcal{L}_{bc}(\theta) = \frac{1}{N_{bc}} \sum_{i=1}^{N_{bc}} \left| \mathcal{B}[\mathbf{u}_\theta]\left(t_{bc}^i, \mathbf{x}_{bc}^i\right) \right|^2,$$

$$\mathcal{L}_r(\theta) = \frac{1}{N_r} \sum_{i=1}^{N_r} \left| \frac{\partial \mathbf{u}_\theta}{\partial t}\left(t_r^i, \mathbf{x}_r^i\right) + \mathcal{N}[\mathbf{u}_\theta]\left(t_r^i, \mathbf{x}_r^i\right) \right|^2,$$

$\left(t^i, \mathbf{x}^i\right)$ — collocation points

Implemented Training Techniques

- **Fourier Feature Embedding** [1]:

$$\gamma(\boldsymbol{x}) = \begin{bmatrix} \cos B\boldsymbol{x} \\ \sin B\boldsymbol{x} \end{bmatrix},$$

where $\boldsymbol{x} \in \mathbb{R}^n$ — input parameters, $B \in \mathbb{R}^{m \times n} \sim \mathcal{N}(0, \sigma^2)$, $\gamma(x) \in \mathbb{R}^{2m}$

- **Loss balancing** [2]:

$$\|\lambda_{ic}\nabla_\theta \mathcal{L}_{ic}(\theta)\| = \|\lambda_{bc}\nabla_\theta \mathcal{L}_{bc}(\theta)\| = \|\lambda_r \nabla_\theta \mathcal{L}_r(\theta)\| =$$

$$\|\nabla_\theta \mathcal{L}_{ic}(\theta)\| + \|\nabla_\theta \mathcal{L}_{bc}(\theta)\| + \|\nabla_\theta \mathcal{L}_r(\theta)\|$$

$$\lambda_{\text{new}} = \alpha\lambda_{\text{old}} + (1-\alpha)\hat{\lambda}_{\text{new}}$$

- **Causal training** [3]:

$$\mathcal{L}_r(\theta) = \frac{1}{M}\sum_{i=1}^{M} w_i \, \mathcal{L}_r^i(\theta), \qquad w_i = \exp\left(-\varepsilon \sum_{k=1}^{i-1} \mathcal{L}_r^k(\theta)\right), \qquad i = 2, \ldots, M,$$

where ϵ — parameter, $\mathcal{L}_r^k(\theta)$ — residual loss function term for k-th time domain

- **Time marching**. Train u_{θ_k} PINN for k-th time window $[t_{k-1}, t_k]$

1D Heat Equation

$$\begin{cases} u_t = u_{xx}, \\ u\big|_{x=0} = 0, \\ u\big|_{x=2\pi} = 0, \\ u\big|_{t=0} = 2\sin 3x\,, \end{cases}$$

$$u = 2\exp(-9t)\sin 3x$$

PINN:

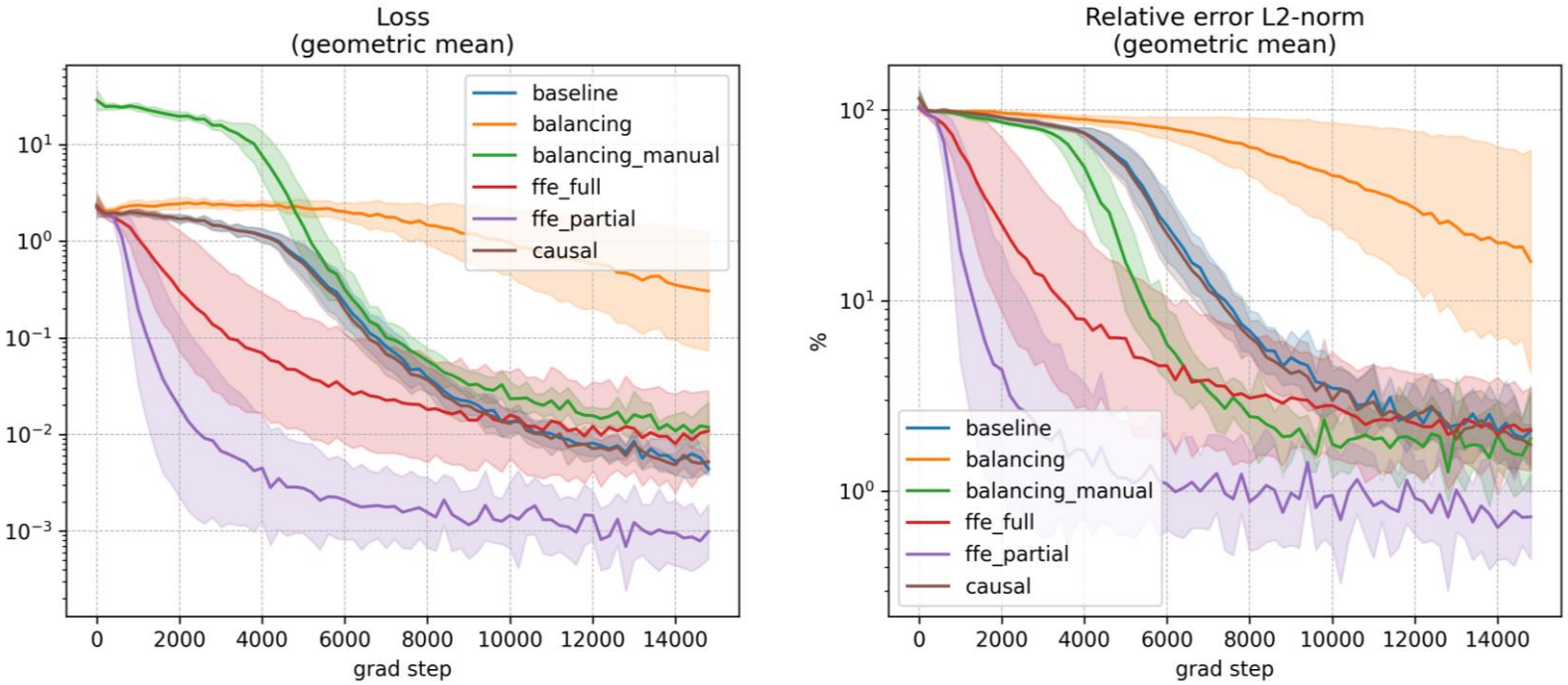
$$\mathbf{u}_\theta : (t, x) \to u,$$

Loss function:

$$\mathcal{L} = \lambda_r \mathcal{L}_r + \lambda_{ic}\mathcal{L}_{ic} + \lambda_{bc}\mathcal{L}_{bc}$$

Parameter	Value
Network	
Architecture	MLP
Activation	tanh
Neurons per layer	128
Number of hidden layers	2
Initialization scheme	Glorot Normal
Training	
Number of epochs	15 000
Number of points	$n_r = 1000$, $n_{ic} = 126$, $n_{bc} = 16$
Optimizer	Adam
Learning rate	10^{-3}

1D Heat Equation. Comparison of Training Strategies



Comparison of training strategies for PINNs solving the 1D heat equation.
The curves and shaded regions represent the geometric mean
and one geometric standard deviation of 10 runs

Loss balancing: $\alpha = 0.95, f = 500$.
Malual loss balancing: $\lambda_r = 1.8, \lambda_{ic} = 12.8, \lambda_{bc} = 9.5$.
FFE: $m = 3, \sigma = 1.0$.
Causal training: $m = 5, \varepsilon = 1.0$.

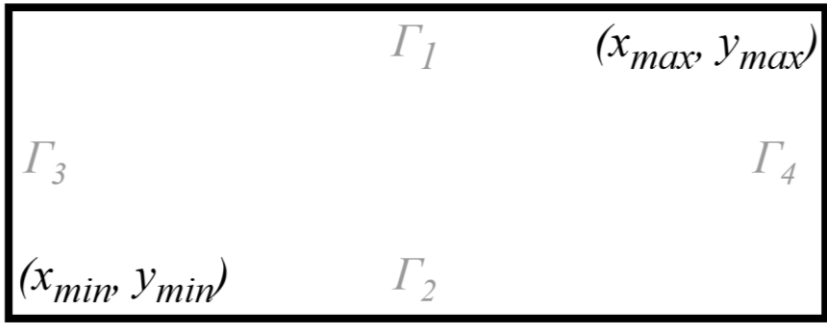
Poiseuille Flow

$$
\begin{cases}
\dfrac{\partial \boldsymbol{v}}{\partial t} + (\boldsymbol{v}\cdot\nabla)\boldsymbol{v} = \dfrac{1}{\rho}\nabla\cdot\sigma + \boldsymbol{f}, \\
\nabla\cdot\boldsymbol{v} = 0, \\
\boldsymbol{v}|_{t=t_{\min}} = -\dfrac{U_{\max}}{h^2}(y-y_{\max})(y-y_{\min}), \\
\boldsymbol{v}|_{\Gamma_1} = \boldsymbol{v}|_{\Gamma_2} = p|_{\Gamma_4} = 0, \\
\boldsymbol{v}|_{\Gamma_3} = -\dfrac{U_{\max}}{h^2}(y-y_{\max})(y-y_{\min}),
\end{cases}
$$

where

- $v = (u, \mathrm{v})^T$ — velocity vector field,
- $\rho = \mathrm{const}$ — fluid density,
- $\sigma = -pI + \frac{1}{\mathrm{Re}}[\nabla\boldsymbol{v} + (\nabla\boldsymbol{v})^T]$,
- $h = (y_{\max} - y_{\min})/2$

$$
u = -\frac{U_{\max}}{h^2}(y-y_{\max})(y-y_{\min}),
$$
$$
v = 0,
$$
$$
p = \frac{1}{\mathrm{Re}}\frac{2U_{\max}}{h^2}(x_{\max} - x)
$$



Computational domain for the Poiseuille flow problem

Dimensionless parameters:

- $\widetilde{U} = 2U_{\max}/3$ — characteristic speed,
- $\widetilde{L} = h$ — characteristic length,
- $\mathrm{Re} = \widetilde{U}\widetilde{L}/\nu$

Poiseuille Flow. Mixed-Variable PINN Formulation

PINN:

Following [4]:

$$\mathbf{u}_\theta : (t, x, y) \to \left(\psi, p, \sigma_{xx}, \sigma_{xy}, \sigma_{yy}\right),$$

where ψ: $\mathbb{R}^3 \to \mathbb{R}^1$ — stream function, such that $u = \partial\psi/\partial y$, $\mathrm{v} = -\partial\psi/\partial x$

Loss function:

$$\mathcal{L} = \lambda_r \mathcal{L}_r + \lambda_{ic} \mathcal{L}_{ic} + \lambda_{\text{walls}} \mathcal{L}_{\text{walls}} + \lambda_{\text{in}} \mathcal{L}_{\text{in}} + \lambda_{\text{out}} \mathcal{L}_{\text{out}}$$

Poiseuille Flow. Sampling

IC/BC Point Counts:

$$L_x = (x_{\max} - x_{\min}), \qquad L_y = (y_{\max} - y_{\min}), \qquad T = (t_{\max} - t_{\min}),$$

$$h_r \sim \left(L_x L_y T / n_r\right)^{1/3}$$

$$h_{ic} \sim \left(L_x L_y / n_{ic}\right)^{1/2}$$

$$h_{bc_x_{min}} \sim \left(L_y T / n_{bc_x_{min}}\right)^{1/2}$$

$$h_{bc_y_{min}} \sim \left(L_y T / n_{bc_y_{min}}\right)^{1/2}$$

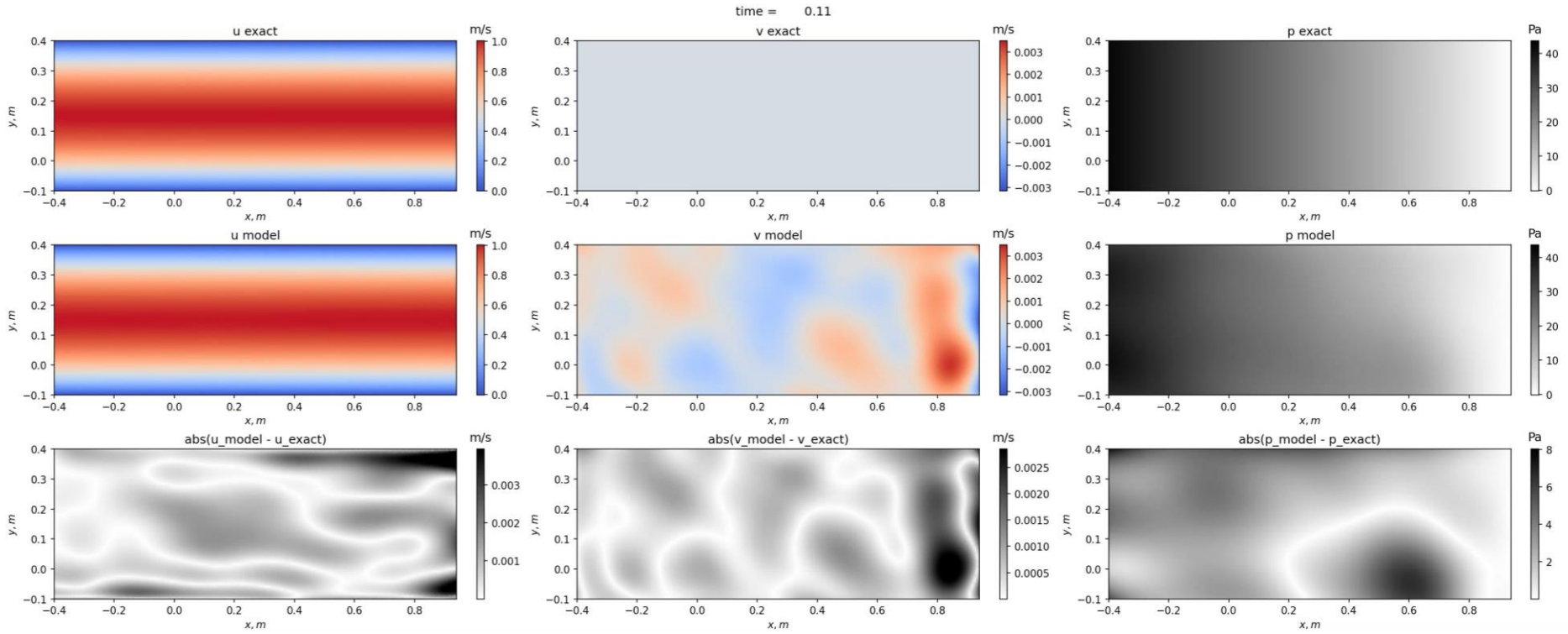
$$h_{obs} \sim \left(\pi d T / n_{obs}\right)^{1/2}$$

$$\left.\begin{array}{c} h_r \sim h_{ic} \\ h_r \sim h_{bc_{x_{min}}} \\ h_r \sim h_{bc_{y_{min}}} \\ h_r \sim h_{obs} \end{array}\right\} \Rightarrow \boxed{\begin{array}{c} n_{bc_x_{min}} \sim L_x T \left(L_x L_y T / n_r\right)^{2/3} \\ n_{bc_x_{min}} \sim L_x T \left(L_x L_y T / n_r\right)^{2/3} \\ n_{bc_y_{min}} \sim L_y T \left(L_x L_y T / n_r\right)^{2/3} \\ n_{obs} \sim \pi d T \left(L_x L_y T / n_r\right)^{2/3} \end{array}}$$

Poiseuille Flow. Training Setup

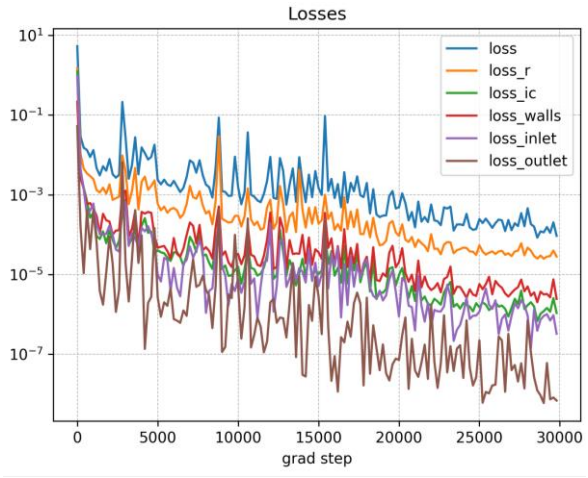
Parameter	Value
Network	
Architecture	MLP
Activation	tanh
Neurons per layer	256
Number of hidden layers	2
Initialization scheme	Glorot Normal
Training	
Number of epochs	3000
Number of batches per epoch	10
Number of points	$n_r = 10\,000$, $n_{ic} = 1770$, $n_{bc_x} = 292$, $n_{bc_y} = 780$
Optimizer	Adam
Scheduler	CosineAnnealingLR $\eta_{\min} = 10^{-4}$
Learning rate	10^{-3}

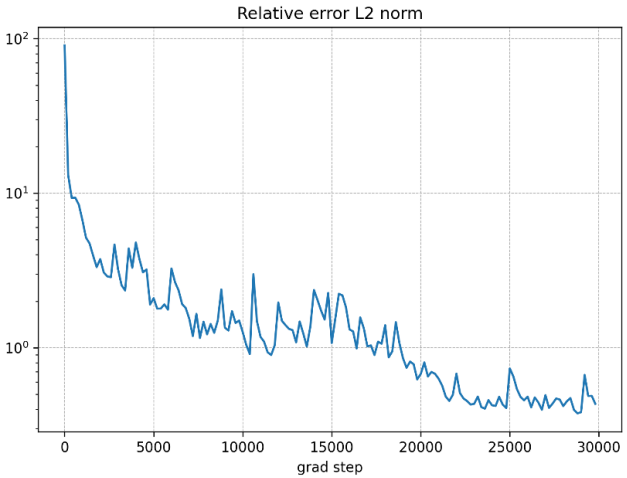
Poiseuille Flow. Reference and PINN Solutions

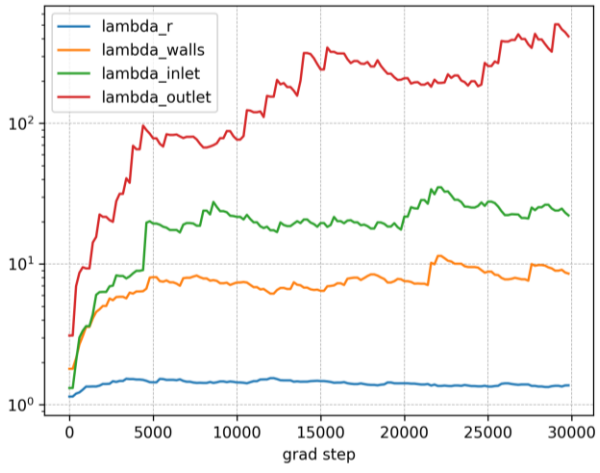


Solution of the Poiseuille flow problem. From left to right: u, v, p.
From top to bottom: reference, model solution, |reference $-$ model|.
$\mathrm{Re} = 162$

Poiseuille Flow. Training Dynamics







Training dynamics for the Navier–Stokes PINN on Poiseuille flow.
From left to right: loss terms, relative L2 error, and λ values.

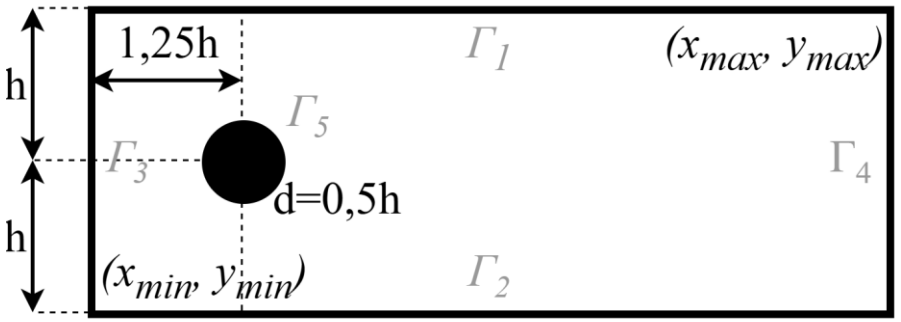
Loss balancing: $\alpha = 0.9, f = 250$.
FFE: $m = 10, \sigma = 1.0$

Flow Past a Cylinder. Problem Setup

$$
\begin{cases}
\dfrac{\partial \boldsymbol{v}}{\partial t} + (\boldsymbol{v}\cdot\nabla)\boldsymbol{v} = \dfrac{1}{\rho}\nabla\cdot\sigma + \boldsymbol{f}, \\
\nabla\cdot\boldsymbol{v} = 0, \\
\boldsymbol{v}|_{t=t_{\min}} = -\dfrac{U_{\max}}{h^2}(y-y_{\max})(y-y_{\min}), \\
\boldsymbol{v}|_{\Gamma_1} = \boldsymbol{v}|_{\Gamma_2} = \boldsymbol{v}|_{\Gamma_5} = 0, \\
\boldsymbol{v}|_{\Gamma_3} = -\dfrac{U_{\max}}{h^2}(y-y_{\max})(y-y_{\min}), \\
\sigma\cdot\vec{n}|_{\Gamma_4} = 0,
\end{cases}
$$

where

- $v = (u, \mathrm{v})^T$ — velocity vector field,
- $\rho = \mathrm{const}$ — fluid density,
- $\sigma = -pI + \frac{1}{\mathrm{Re}}[\nabla\boldsymbol{v} + (\nabla\boldsymbol{v})^T]$,
- $h = (y_{\max} - y_{\min})/2$



Computational domain for 2D incompressible flow past a circular cylinder in a channel

Dimensionless parameters:

- $\tilde{U} = 2U_{\max}/3$ — characteristic speed,
- $\tilde{L} = d$ — characteristic length,
- $\mathrm{Re} = \tilde{U}\tilde{L}/\nu$

Flow Past a Cylinder. Mixed-Variable PINN Formulation

PINN:

$$\mathbf{u}_\theta : (t, x, y) \to \left(\psi, p, \sigma_{xx}, \sigma_{xy}, \sigma_{yy}\right),$$

where $\psi: \mathbb{R}^3 \to \mathbb{R}^1$ — stream function, such that $u = \partial\psi/\partial y$, $\mathrm{v} = -\partial\psi/\partial x$

Loss function:

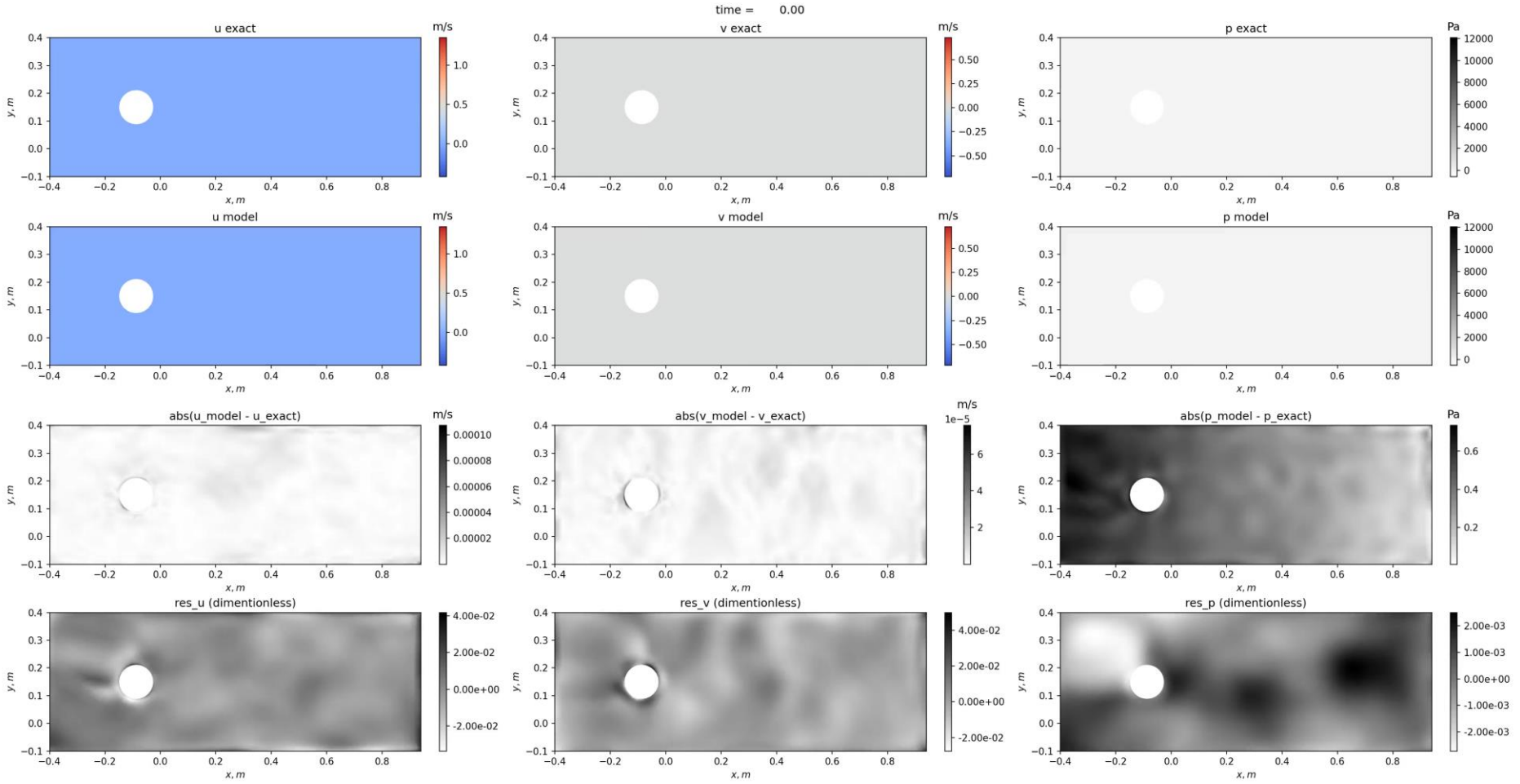
$$\begin{gathered}\mathcal{L} = \lambda_{u_r}\mathcal{L}_{u_r} + \lambda_{\mathrm{v}_r}\mathcal{L}_{\mathrm{v}_r} + \lambda_{\sigma_{xx_r}}\mathcal{L}_{\sigma_{xx_r}} + \lambda_{\sigma_{xy_r}}\mathcal{L}_{\sigma_{xy_r}} + \lambda_{\sigma_{yy_r}}\mathcal{L}_{\sigma_{yy_r}} + \lambda_{p_r}\mathcal{L}_{p_r} + \\ + \lambda_{u_{ic}}\mathcal{L}_{u_{ic}} + \lambda_{\mathrm{v}_{ic}}\mathcal{L}_{\mathrm{v}_{ic}} + \lambda_{p_{ic}}\mathcal{L}_{p_{ic}} + \\ + \lambda_{u_{\mathrm{walls}}}\mathcal{L}_{u_{\mathrm{walls}}} + \lambda_{\mathrm{v}_{\mathrm{walls}}}\mathcal{L}_{\mathrm{v}_{\mathrm{walls}}} + \\ + \lambda_{u_{\mathrm{in}}}\mathcal{L}_{u_{\mathrm{in}}} + \lambda_{\mathrm{v}_{\mathrm{in}}}\mathcal{L}_{\mathrm{v}_{\mathrm{in}}} + \\ + \lambda_{\sigma_{xx_{\mathrm{out}}}}\mathcal{L}_{\sigma_{xx_{\mathrm{out}}}} + \lambda_{\sigma_{xy_{\mathrm{out}}}}\mathcal{L}_{\sigma_{xy_{\mathrm{out}}}} + \\ + \lambda_{u_{\mathrm{data}}}\mathcal{L}_{u_{\mathrm{data}}} + \lambda_{\mathrm{v}_{\mathrm{data}}}\mathcal{L}_{\mathrm{v}_{\mathrm{data}}}\end{gathered}$$

Flow Past a Cylinder. Training Setup

Parameter	Value
Network	
Architecture	MLP
Activation	gelu
Neurons per layer	256
Number of hidden layers	4
Initialization scheme	Glorot Normal
Training	
Number of epochs	20 000
Number of batches per epoch	10
Number of points	$n_r = 100\,000$, $n_{ic} = 571$, $n_{bc_x} = 2557$, $n_{bc_y} = 6852$, $n_{\text{obs}} = 50200$
Optimizer	Adam
Scheduler	CosineAnnealingLR $\eta_{\min} = 10^{-4}$
Learning rate	10^{-3}

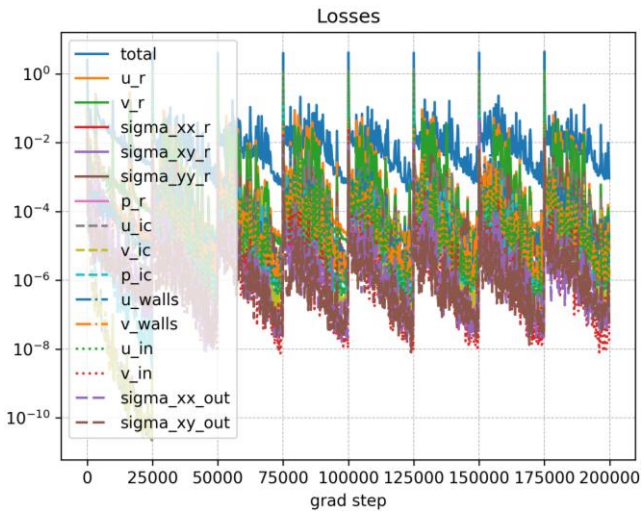
Parameter	Value
Fourier Embedding	$m = 256, \sigma = 1.0$
Loss balancing	$\alpha = 0.9, f = 1000$
Causal training	$m = 16, \varepsilon = 10.0$
Number of time windows	8

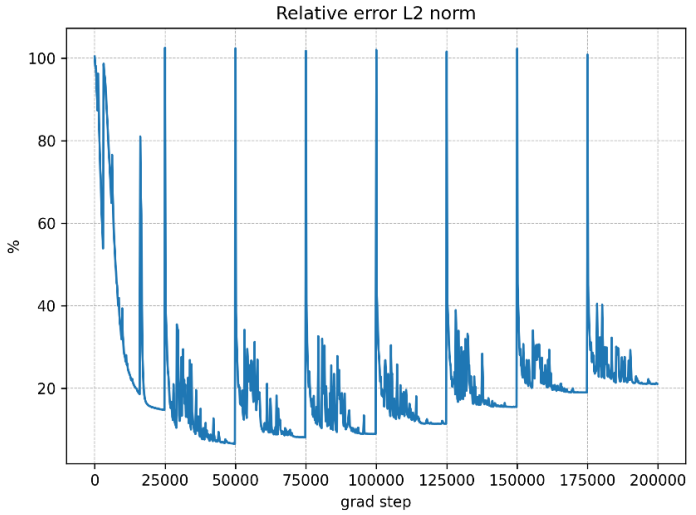
Flow Past a Cylinder. Reference and PINN Solutions

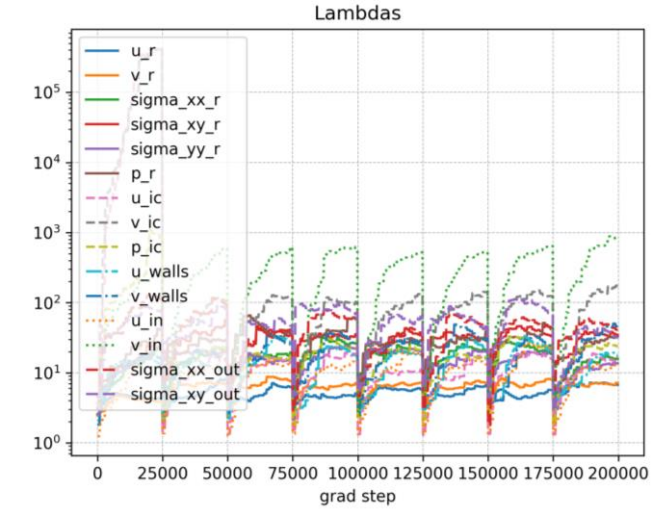


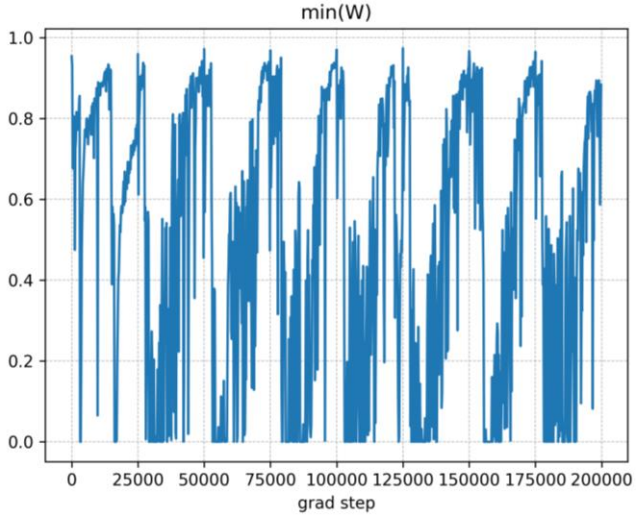
Solution of the flow past a cylinder problem. From left to right: u, v, p.
From top to bottom: reference, model solution, |reference $-$ model|, model residual.
Re $= 80.9$

Flow Past a Cylinder. Training Dynamics









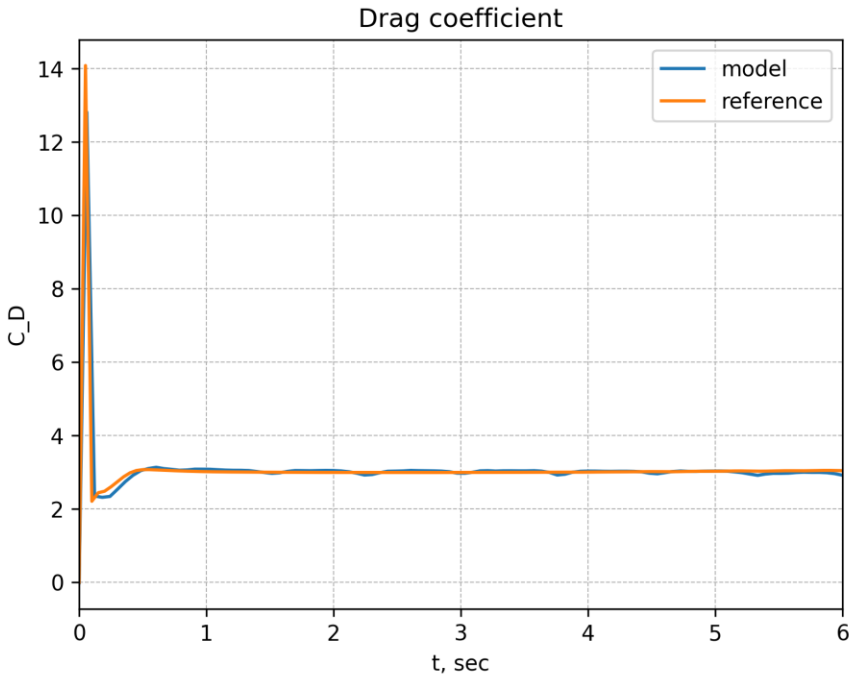
Training dynamics for the Navier–Stokes PINN on flow past a cylinder problem.

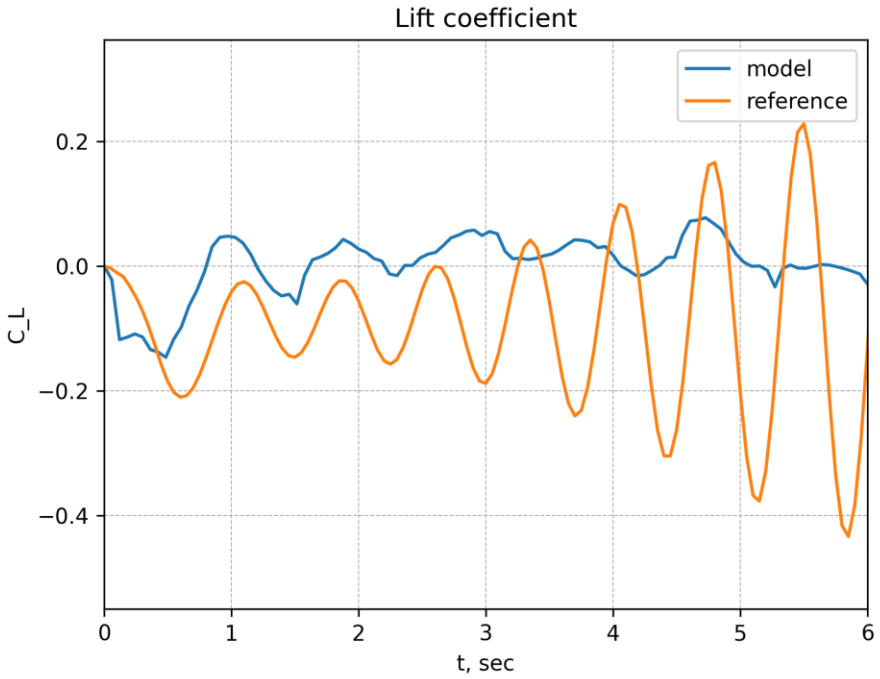
Flow Past a Cylinder. Drag and Lift Coefficients

Drag and lift coefficients [5]:

$$(F_D, F_L)^T = \int_S \sigma \cdot \vec{n}\, \mathrm{ds},$$

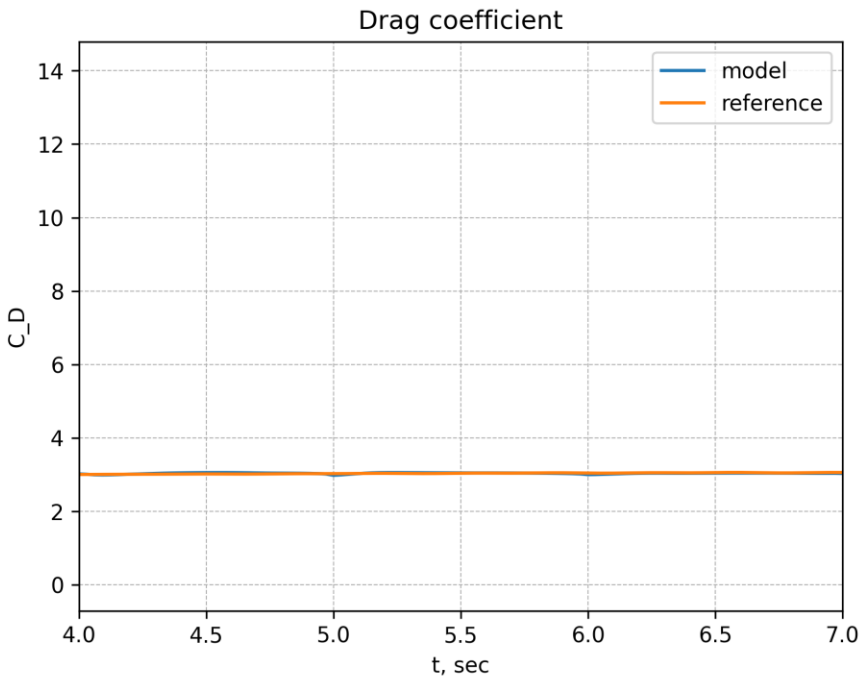
$$C_D = \frac{2F_D}{\rho \widetilde{U}^2 d}, \qquad C_L = \frac{2F_L}{\rho \widetilde{U}^2 d}$$

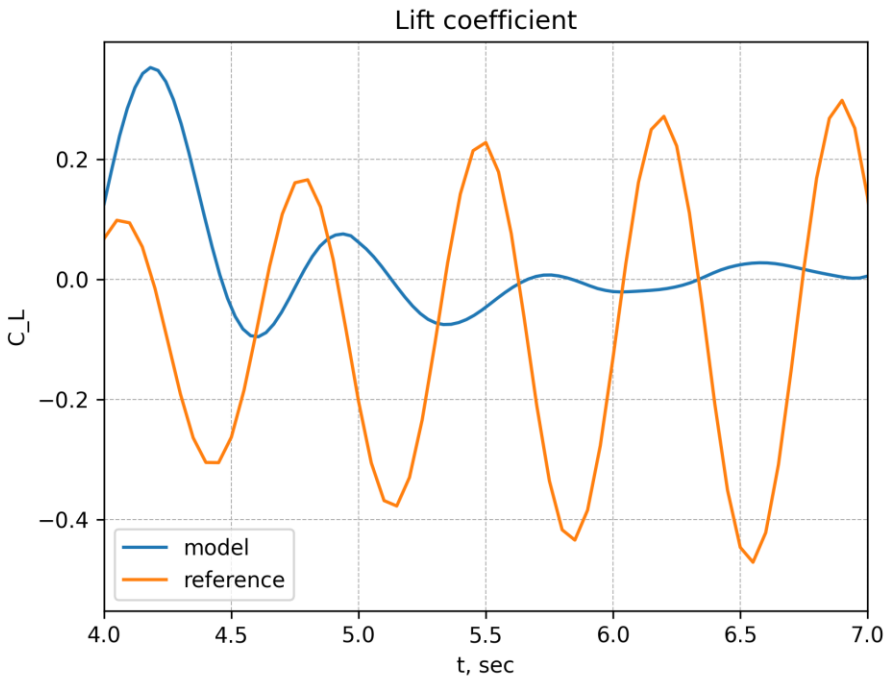




Drag and lift coefficients for the Navier–Stokes PINN on flow past a cylinder problem.

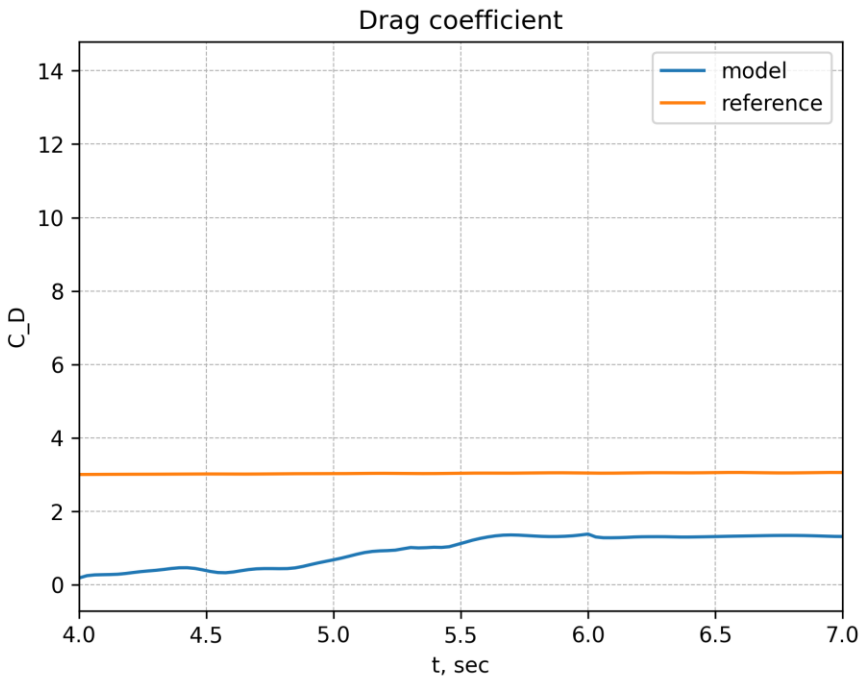
Flow Past a Cylinder. Solution with COMSOL-Based Initial Condition. Drag and Lift Coefficients

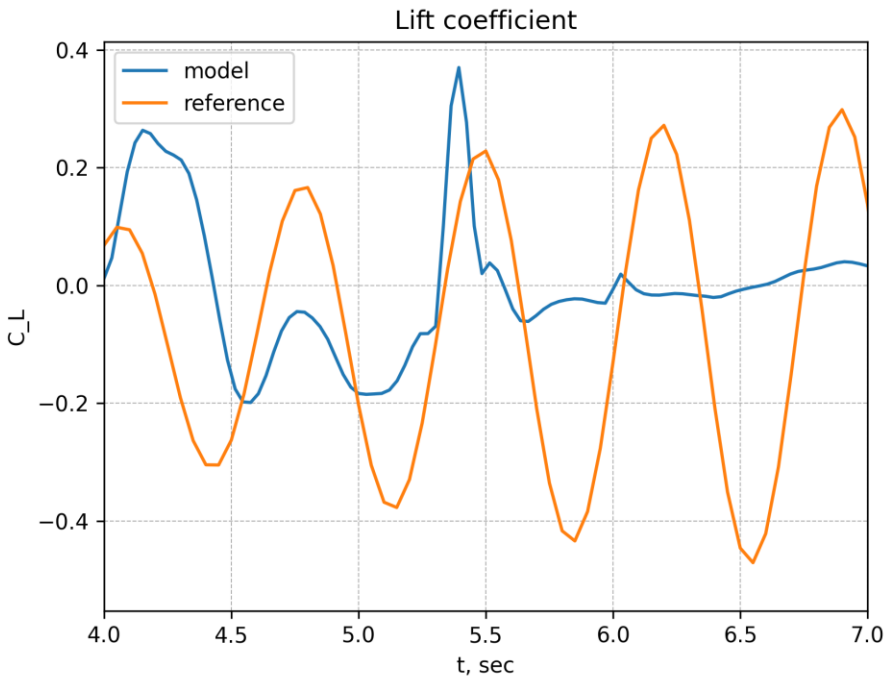




Drag and lift coefficients for the Navier–Stokes PINN on flow past a cylinder problem, reference solution at $t = 4[s]$ as initial condition.

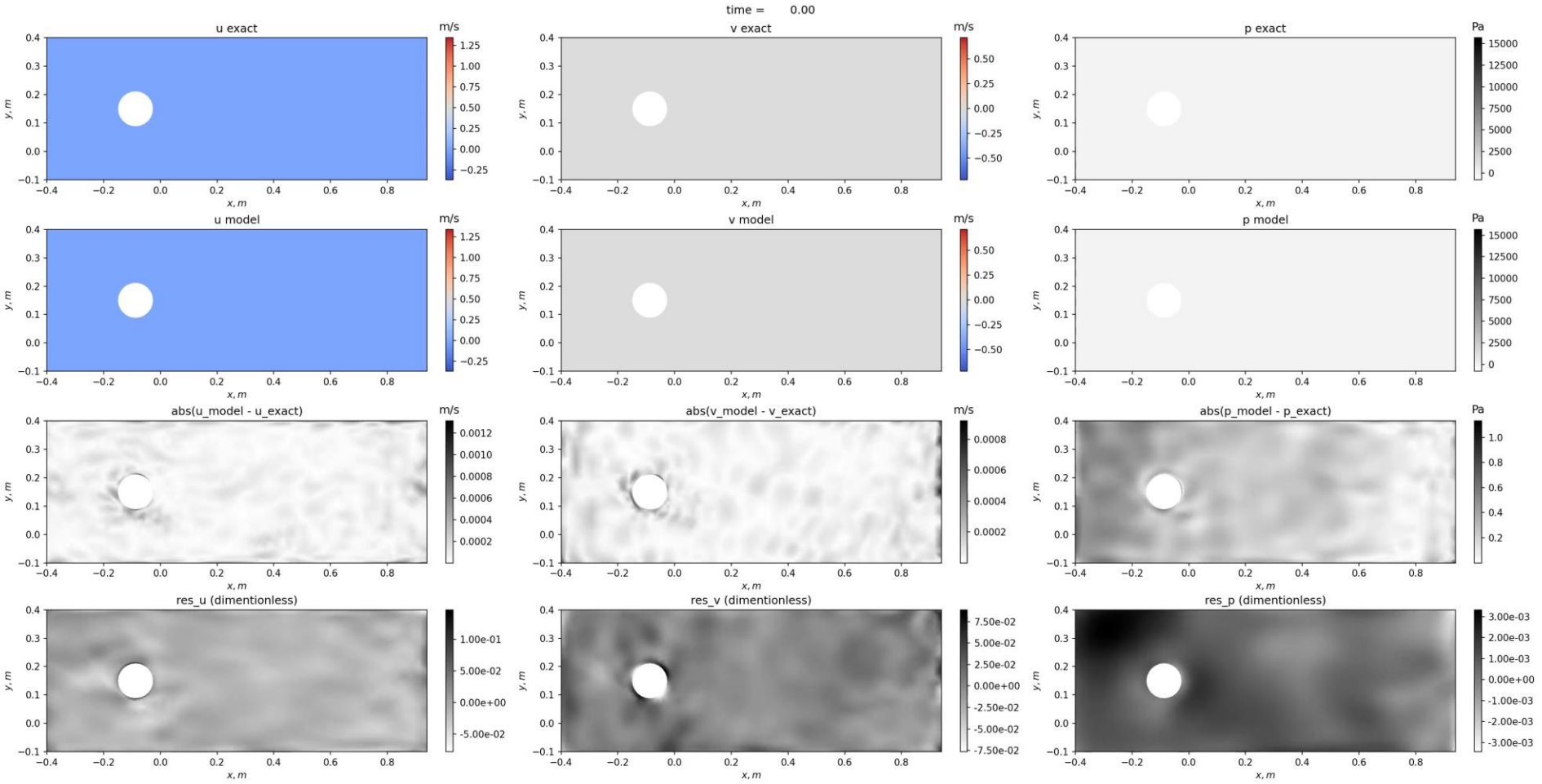
Flow Past a Cylinder. Solution with Pretrained Model as Initial Condition. Drag and Lift Coefficients





Drag and lift coefficients for the Navier–Stokes PINN on flow past a cylinder problem, pretrained model at $t = 4[s]$ as initial condition.

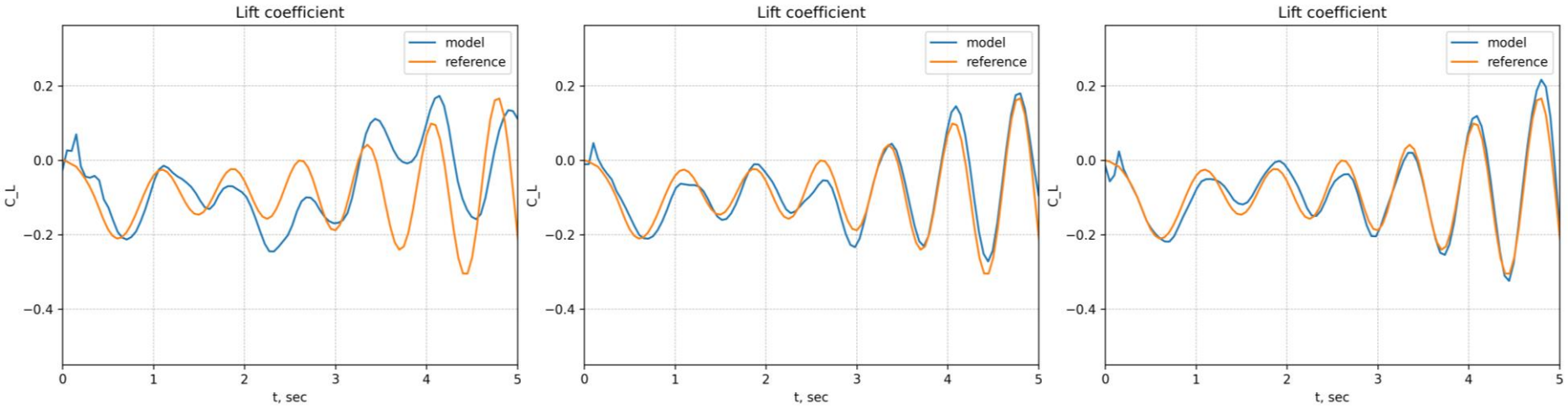
Flow Past a Cylinder. Solution. Data-Assisted PINN. Reference and PINN Solutions



Solution of the flow past a cylinder problem, COMSOL reference data as an additional data-driven loss term. Reference data fraction of 1% relative to the n_r. From left to right: u, v, p. From top to bottom: reference, model solution, |reference $-$ model|, model residual.

Re $= 80.9$

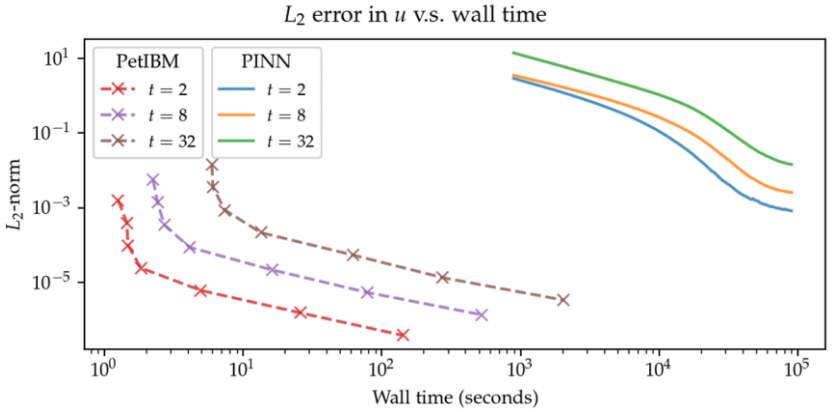
Flow Past a Cylinder. Solution. Data-Assisted PINN. Drag and Lift Coefficients



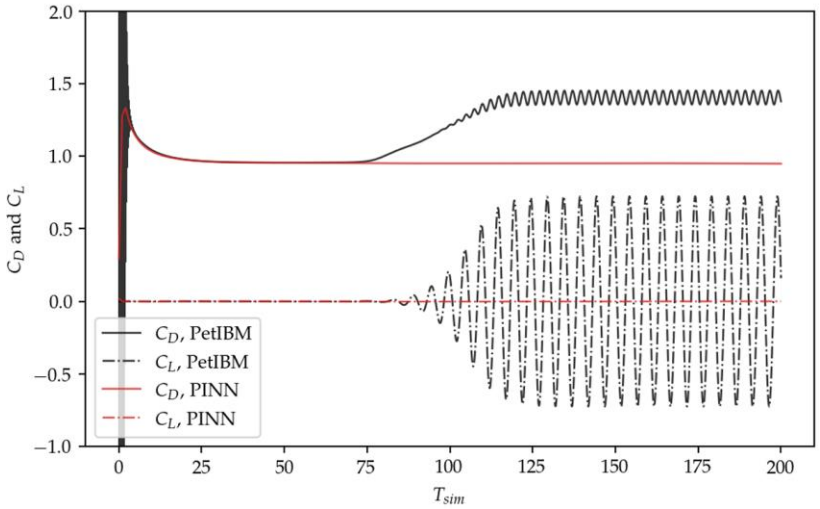
Lift coefficient for the Navier–Stokes PINN solution of the flow past a cylinder problem using COMSOL reference data as an additional data-driven loss term. From left to right: reference data fractions of 1%, 5%, and 10% relative to the n_r

Literature Evidence: PINNs Can Be Slow and Inaccurate

PetIBM — traditional solver using staggered-grid finite difference methods with MPI parallelization and GPU computing [6]



L2-norm error versus wall-clock time for PINN and PetIBM on the Taylor–Green vortex problem [6]



Drag and lift coefficients for 2D flow past a circular cylinder obtained with PINN and PetIBM [6]

Conclusions and Future Work

- Despite its attractive formulation, PINNs do not currently appear to be a promising standalone tool for complex real-world CFD problems.
- Acceptable accuracy may require extensive tuning and additional data, which limits the practical efficiency of the method.
- Further research is required to improve PINN's robustness and to better understand how training strategies and hyperparameters affect convergence
- As future work, we plan to consider neural-operator-based approaches.

Thank you for your attention

References

1. **Tancik M.** Fourier features let networks learn high frequency functions in low dimensional domains / M. Tancik, P. Srinivasan, B. Mildenhall [et al.] // Advances in Neural Information Processing Systems. — 2020. — Vol. 33. — P. 7537–7547.
2. **Wang, S., Sankaran, S., Wang, H., Perdikaris, P.** An Expert's Guide to Training Physics-informed Neural Networks // ArXiv, 2023, abs/2308.08468
3. **Wang S., Sankaran S., Perdikaris P.** Respecting causality is all you need for training physics-informed neural networks // Computer Methods in Applied Mechanics and Engineering. –2024. – Vol. 421.
4. **Rao C.** Physics-informed deep learning for incompressible laminar flows / C. Rao, H. Sun, Y. Liu // Theoretical & Applied Mechanics Letters. — 2020. — Vol. 10, № 3. — P. 207–212.
5. **Schäfer M., Turek S., Durst F., Krause E., Rannacher R.** Benchmark Computations of Laminar Flow Around a Cylinder // Flow Simulation with High-Performance Computers II / ed. by E. H. Hirschel. Wiesbaden : Vieweg+Teubner Verlag, 1996. P. 547–566. (Notes on Numerical Fluid Mechanics ; Vol. 48).
6. **Chuang P.-Y.** Experience report of physics-informed neural networks in fluid simulations: pitfalls and frustration / P.-Y. Chuang, L. A. Barba // Proceedings of the 21st Python in Science Conference. — 2022. — P. 28–36.
7. **Raissi, M., Perdikaris, P., & Karniadakis, G. E.** Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations. // Journal of Computational Physics, 2019

Implementation Details

Hardware

Local: GeForce GTX 1650, 4 GB VRAM, 16 GB RAM
Google Colab: T4, L4, A100, G4 GPUs

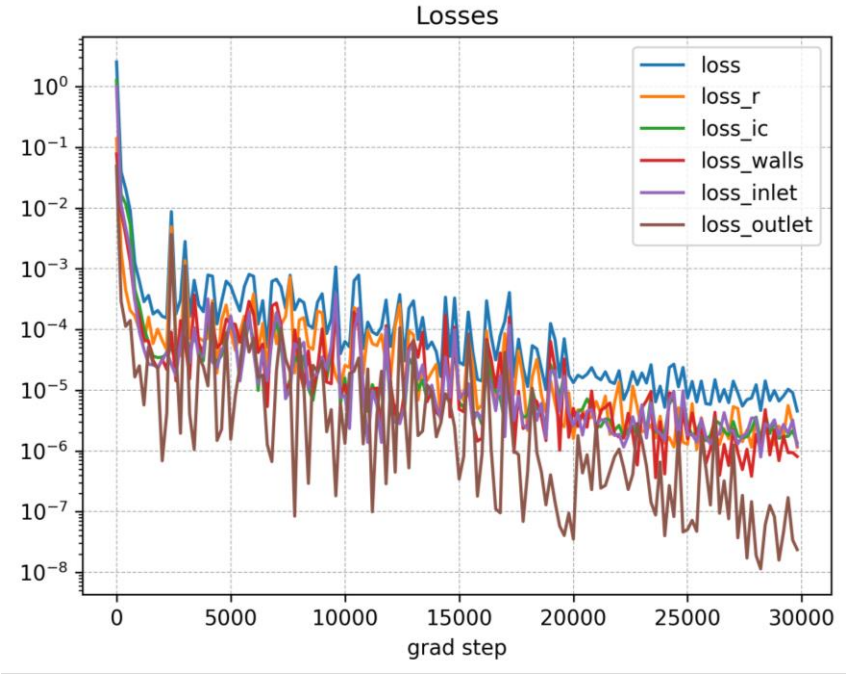
Environment

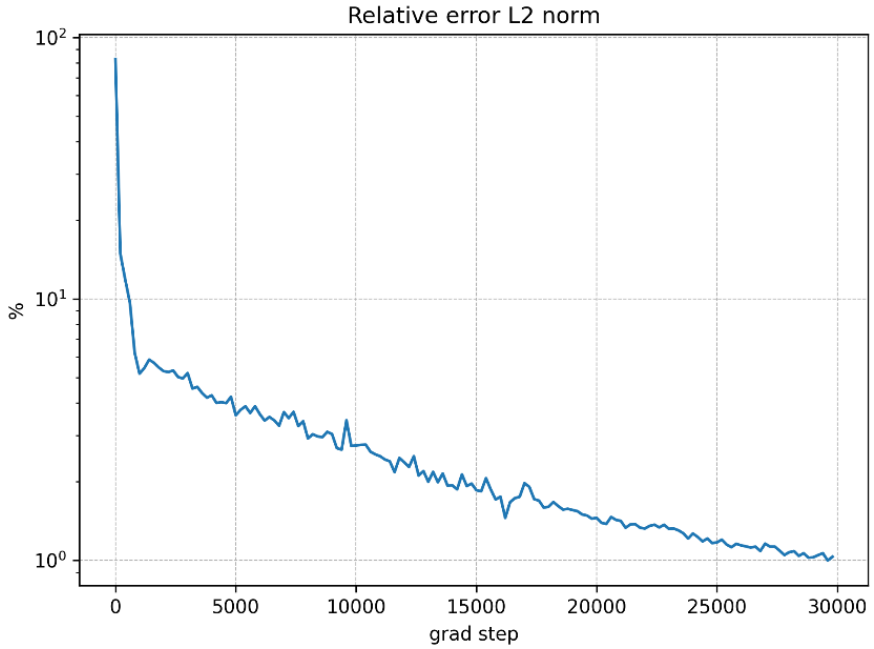
Local runs: fixed virtual environment (PyTorch 2.9.1+cu130, NumPy 2.3.5)
Colab runs: package versions may vary due to platform updates

Reference solutions

When an analytical solution was not available, reference solutions were obtained using the Finite Element Method in COMSOL

Poiseuille Flow. Training Dynamics Without Training Techniques





Training dynamics for the Navier–Stokes PINN on Poiseuille flow without any training techniques. From left to right: loss terms, relative L2 error