Data-Free PINN Approach for Predictions of Unsteady Vortex Shedding

Malavika Nair
Department of Aerospace Engineering
Indian Institute of Technology Kanpur
Kanpur, India 208016
malavikan23@iitk.ac.in

Prashant Kumar
Department of Aerospace Engineering
Indian Institute of Technology Kanpur
Kanpur, India 208016
kumarp22@iitk.ac.in

Rajesh Ranjan
Department of Aerospace Engineering
Indian Institute of Technology Kanpur
Kanpur, India 208016
rajeshr@iitk.ac.in

1 Introduction

The simulation of incompressible fluid flows around bluff bodies remains a central problem in computational fluid dynamics (CFD), owing to the rich vortex dynamics that arise even at moderate Reynolds numbers. Among these configurations, flow over a circular cylinder serves as a canonical benchmark for validating numerical schemes, assessing physical modelling strategies, and developing new computational methodologies. Traditional CFD solvers such as finite volume and finite element methods rely on discretization of the governing equations and require meshing, iterative linear solvers, and stabilization techniques. While these methods are well established, they are computationally intensive and lack flexibility when applied to inverse, data-scarce, or parameter-sensitive problems.Raissi et al. [2022]

In this work, we investigate the performance of PINNs in simulating the two-dimensional incompressible flow over a circular cylinder, with emphasis on flows at Re = 40 and Re = 100 in unbounded domains and highlight the practical strategies that make PINNs more competitive with conventional CFD for moderately complex laminar flows

2 Methodology

2.1 Governing Equations

We consider the two-dimensional, incompressible flow of a Newtonian fluid past a circular cylinder at Reynolds numbers $Re = 40$(steady) and $Re = 100$ (unsteady). The non-dimensional governing equations are the incompressible Navier-Stokes equations, expressed in conservative form.

Let the velocity field be

$$\mathbf{u}(x, y) = (u(x, y),\, v(x, y))\,,$$

and the pressure field be $p(x, y)$. The continuity and momentum equations are:

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

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

where $\mathbf{u} \otimes \mathbf{u}$ denotes the dyadic product of the velocity field. Re is the Reynolds number.

For the steady-state flow considered in this study, the transient term is omitted. These equations form the physical constraints embedded within the PINN loss function. Boundary conditions including

2nd Conference on Applied AI and Scientific Machine Learning (CASML 2025).

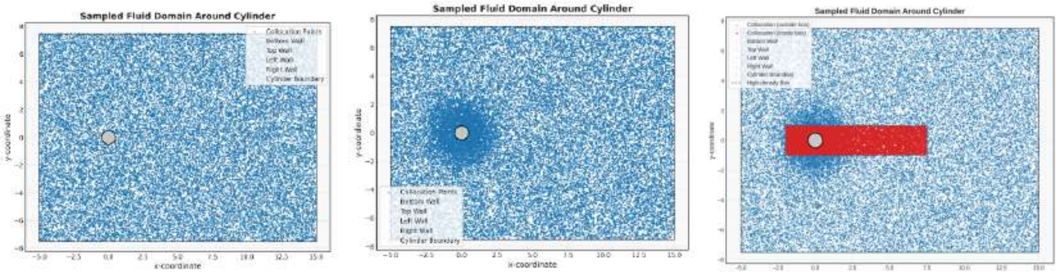
no-slip on the cylinder wall, prescribed inflow velocity, farfield on boundary walls, and outflow conditions, are enforced through additional penalty terms to ensure physically consistent solutions.

2.2 Numerical Approach

This subsection outlines a few numerical approaches used in the simulation and it's benefits.

2.2.1 Domain sampling.

Interior and boundary collocation points are generated using the `CylinderDomainSampler`, which combines uniform sampling in the bulk region with KDTree-weighted sampling near the cylinder to increase point density where gradients are expected to be larger. Boundary sets include a dense discretizations of the circular cylinder and points on the inlet, outlet and far-field conditions.



(a) Uniform distribution of collocation points

(b) Spatially adaptive distribution

(c) Extra High Density Box added

Figure 1: Different iterations of collocation point distribution used.

2.2.2 Domain Decomposition and Shifting

The solution is approximated as a weighted sum of three subnetworks:

$$\widehat{\mathbf{q}}(x) = \sum_{i=1}^{n} w_i(x)\, S_i(x - \Delta_i),$$

where the subnetworks S_i are instances of `SubdomainNetwork`. Each subnetwork is shifted by a fixed vector Δ_i and blended through smooth window functions $w_i(x)$ constructed using `construct_window_function_ND`. This decomposition reduces the conditioning burden by allowing each network to specialise in a portion of the domain. Moseley et al. [2023]

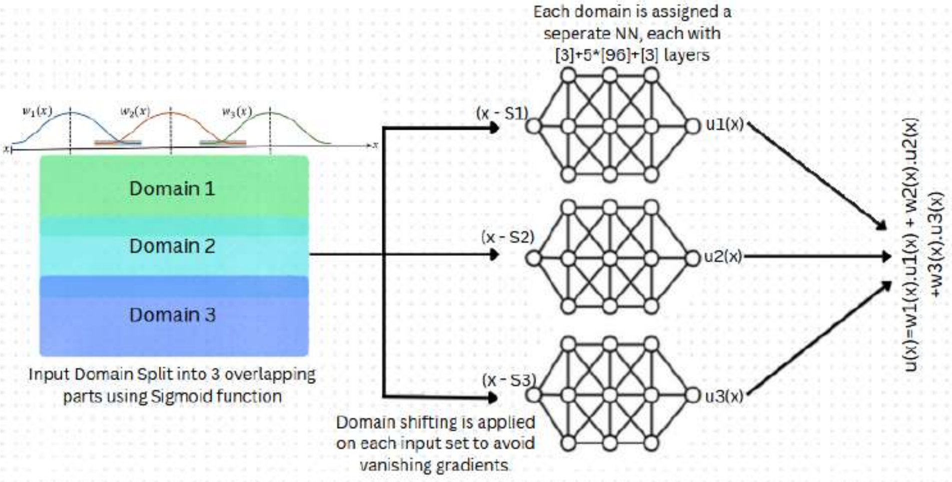

Each domain is assigned a seperate NN, each with [3]+5*[96]+[3] layers
$w_1(x)$
$w_2(x)$
$w_3(x)$
x
Domain 1
Domain 2
Domain 3
Input Domain Split into 3 overlapping parts using Sigmoid function
(x - S1)
(x - S2)
(x - S3)
u1(x)
u2(x)
u3(x)
Domain shifting is applied on each input set to avoid vanishing gradients
u(x)=w1(x).u1(x) + w2(x).u2(x) +w3(x).u3(x)

Figure 2: Overall Network Architecture

For the Steady Case, each domain has been assigned a neural network with 2 input neurons, 5 hidden layers with 192 neurons each, and finally 3 output neurons. Correspondingly for the Unsteady Case, each domain has been assigned a neural network with 3 input layers(to account for the third dimension, time), 5 hidden layers with 96 neurons each, and finally 3 output layers.

2.2.3 Residual formulation.

For a set of collocation points, the method computes the continuity and momentum residuals using PyTorch automatic differentiation. These residuals 3 form the core physics loss. The pointwise residuals of the steady incompressible Navier–Stokes equations are

$$r_c = u_x + v_y, \tag{3}$$

$$r_x = \partial_x(u^2) + \partial_y(uv) + p_x - \frac{1}{Re}(u_{xx} + u_{yy}), \tag{4}$$

$$r_y = \partial_x(uv) + \partial_y(v^2) + p_y - \frac{1}{Re}(v_{xx} + v_{yy}). \tag{5}$$

2.2.4 Boundary constraints.

Boundary conditions are imposed weakly through mean-squared penalties. No-slip conditions are enforced on the cylinder and far-field conditions are enforced on the top and bottom boundary of the domain, while inflow and outlet constraints are assigned to the left and right boundaries, respectively.

Dirichlet velocity boundary conditions are imposed through

$$\mathcal{L}_b^u = \frac{1}{N_b} \sum_{j=1}^{N_b} \Big[\big(u_\theta - u^{BC}\big)^2 + \big(v_\theta - v^{BC}\big)^2 \Big]_{x_j}. \tag{6}$$

The outlet pressure condition uses

$$\mathcal{L}_b^p = \frac{1}{N_p} \sum_{j=1}^{N_p} \big(p_\theta - p^{BC}\big)_{x_j}^2. \tag{7}$$

2.2.5 Residual-based activity weighting.

To emphasise poorly resolved regions during training, the implementation includes a residual-based activity (RBA) mechanism. Exponential accumulators (`rsum1`, `rsum2`, `rsum3`) track smoothed normalised residual magnitudes with decay factor γ and step size η. These accumulators act as multiplicative weights in the physics loss, increasing the contribution of regions where the PDE residual is high. Mao and Meng [2023] Residual-based activity (RBA) weights are updated as

$$R^{(t)} = \gamma\, R^{(t-1)} + \eta\, \frac{|r^{(t)}|}{\max |r^{(t)}|}, \tag{8}$$

and applied multiplicatively to form the physics loss:

$$\mathcal{L}_r = \frac{1}{N_r} \sum_{k=1}^{N_r} \Big[(R_x\, r_x)^2 + (R_y\, r_y)^2 + (R_c\, r_c)^2 \Big]_{x_k}. \tag{9}$$

2.2.6 Optimisation procedure.

The total loss comprises the weighted residuals, boundary terms, and optional data penalties. Network parameters are updated using the Adam optimiser with a multiplicative exponential learning-rate schedule implemented through `LambdaLR`. Gradient clipping is applied to prevent numerical instabilities, and prediction snapshots are periodically generated and for monitoring and diagnostics. The total loss minimized during training is

$$\mathcal{L}_{\text{total}} = \mathcal{L}_r + 2\left(\mathcal{L}_b^{u,\text{cyl}} + \mathcal{L}_b^{u,\text{walls}} + \mathcal{L}_b^{u,\text{inlet}} + \mathcal{L}_b^{p} \right) \tag{10}$$

2.3 Computational Domain and Grid

The computational domain consists of a two-dimensional channel extending from $x = -5$ to $x = 20$ and $y = -7.5$ to $y = 7.5$, with a circular cylinder of radius 0.5 centered at the origin. The outer boundary is divided into inlet, outlet, and upper/lower boundary, while the inner circular boundary represents the no-slip cylinder surface.

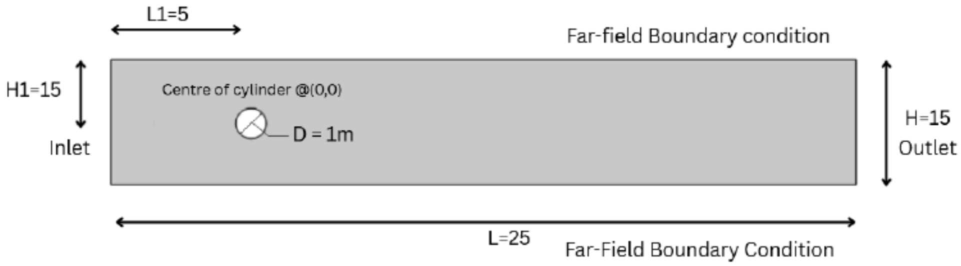
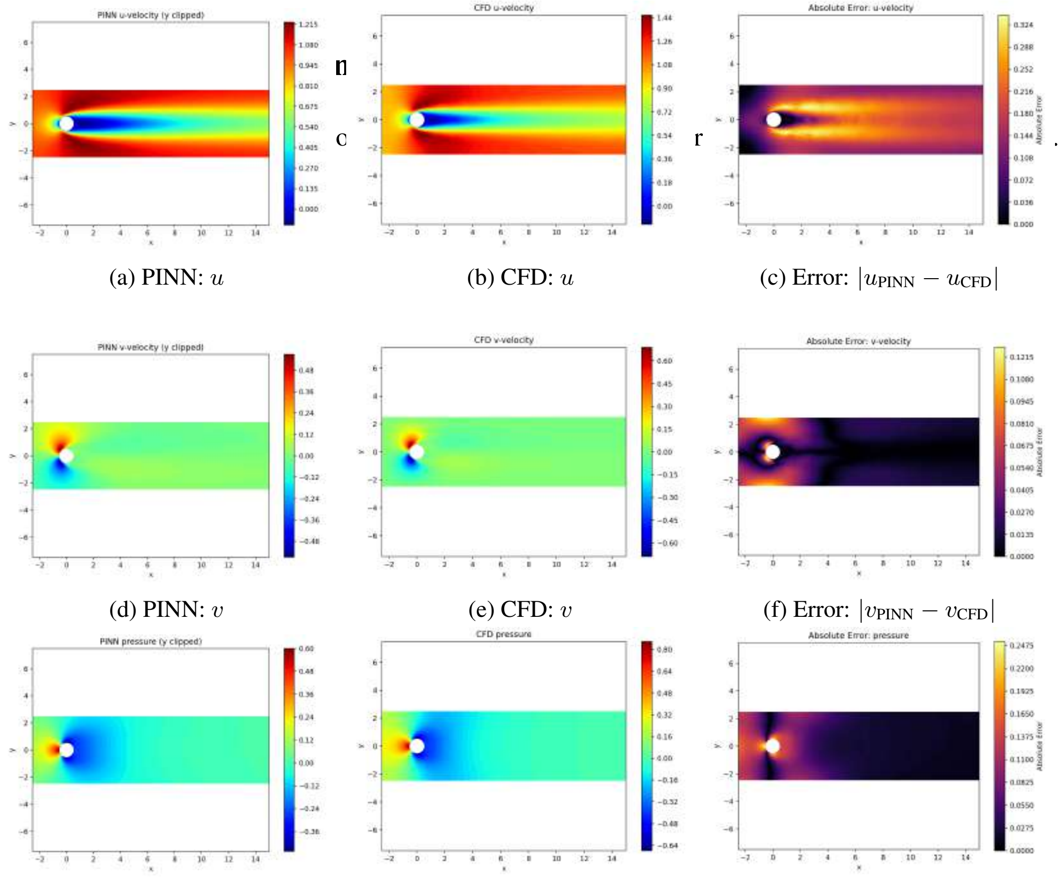

L1=5
Far-field Boundary condition
H1=15
Centre of cylinder @(0,0)
D = 1m
Inlet
H=15
Outlet
L=25
Far-Field Boundary Condition

Figure 3: Computational domain

Interior collocation points are sampled using the hybrid strategy provided by `CylinderDomainSampler`. For post-processing, predictions are evaluated on a structured Cartesian grid covering the entire rectangle. Points lying inside the cylinder are removed based on their radial distance. This grid is used to produce contour and vector fields presented in the following sections.



(a) PINN: u

(b) CFD: u

(c) Error: $|u_{\text{PINN}} - u_{\text{CFD}}|$

(d) PINN: v

(e) CFD: v

(f) Error: $|v_{\text{PINN}} - v_{\text{CFD}}|$

(g) PINN: p

(h) CFD: p

(i) Error: $|p_{\text{PINN}} - p_{\text{CFD}}|$

Figure 4: Comparison between PINN and CFD predictions, and their absolute errors for u, v, and p.

The steady-state PINN solution shows strong agreement with the CFD reference at $Re = 40$. The predicted velocity and pressure fields closely match the CFD contours, with maximum absolute errors of 0.324 for u, 0.121 for v, and 0.2475 for pressure. Most of the domain exhibits much smaller differences, and the PINN correctly reproduces the smooth, non-recirculating wake expected at this Reynolds number, indicating good steady-state convergence.

Unsteady State:
We now extend the analysis to the unsteady case by introducing time as an additional dimension in the PINN formulation. This requires augmenting the loss function with an initial–condition term, which is enforced using CFD data at a chosen reference state. In this study, the network is initialized with the flow field at $t_0 = 83\,\mathrm{s}$, a time at which vortex shedding is already fully developed. The PINN is then advanced to predict the flow one second later, allowing us to observe the subsequent temporal development. Although this approach uses partial data to define the initial state, it serves as a first step toward a fully physics–driven unsteady simulation beginning at $t_0 = 0$ with quiescent initial

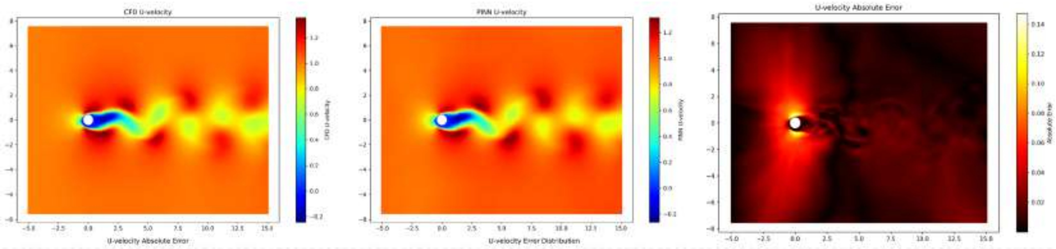


Figure 5: Unsteady Flow with to=83secs

The Unsteady PINN solution with given initial conditions shows good agreement with the CFD reference at $Re = 40$. The predicted velocity and pressure fields closely match the CFD contours, with a maximum absolute error of 0.14 for u-velocity. Most of the domain exhibits much smaller differences, and the PINN correctly reproduces the wake and vortex shedding indicating good steady-state convergence.

Acknowledgments

The authors gratefully acknowledge the GPU computing facilities provided by the Indian Institute of Technology Kanpur, which enabled the successful completion of this work.

References

Zhiping Mao and Xuhui Meng. Physics-informed neural networks with residual/gradient-based adaptive sampling methods for solving partial differential equations with sharp solutions. *Applied Mathematics and Mechanics (English Edition)*, 44(7):1069–1084, 2023. doi: 10.1007/s10483-023-2994-7.

Ben Moseley, Andrew Markham, and Tarje Nissen-Meyer. Finite basis physics-informed neural networks (fbpinns): A scalable domain decomposition approach for solving differential equations. *Journal of Computational Physics*, 474:111819, 2023. doi: 10.1016/j.jcp.2022.111819. Introduces domain decomposition to improve PINN scalability and convergence on large problems.

M. Raissi et al. Surrogate normal-forms for the numerical bifurcation and stability analysis of navier-stokes flows via machine learning. *Journal of Fluid Mechanics*, 942:A3, 2022. doi: 10.1017/jfm.2022.3. Used for understanding flow behavior and bifurcation characteristics in Navier-Stokes systems.