Bridging Continuous and Discrete Physics: A Hybrid PINN Framework with Differentiable Solvers

Guillermo MorenoDepartment of Electrical Engineering
Pontifical Catholic University of Rio de Janeiro
Gávea, Rio de Janeiro - State of Rio de Janeiro, Brasil
guillermo@ica.ele.puc-rio.br

Abstract

Physics-Informed Neural Networks (PINNs) provide a flexible approach for solving partial differential equations (PDEs) without labeled data, yet their performance often degrades under varying boundary conditions or coarse discretizations. We propose a hybrid framework that integrates PINNs with differentiable solvers to learn mappings between *imperfect* and *refined* physical representations. Instead of enforcing hard boundary or initial constraints, our model leverages PDE residual optimization to assist a differentiable-physics component that refines coarse or low-accuracy numerical solutions. The network acts as a correction operator, improving solver outputs through gradient feedback derived from the discrete numerical process. We evaluate the method on single-phase Darcy flow, diffusion, and Poisson equations, training on fine simulations and testing under coarse spatial–temporal discretizations or higher solver tolerances. Results show improved generalization across discretization regimes, suggesting that coupling continuous residual learning with differentiable solvers offers a promising direction for robust, data-free PDE modeling.

1 Introduction

Physics-Informed Neural Networks (PINNs) [8] have become a popular paradigm for solving partial differential equations (PDEs) using machine learning. PINNs embed the governing physical laws directly into the training objective by penalizing the differential operator residuals of the PDE alongside data losses. This coupling enables models to infer physically consistent solutions even when data are scarce. However, the original formulation presents a key limitation: the network must be retrained for each specific boundary condition or parameter regime, and optimization can become ill-conditioned in stiff or heterogeneous systems. Various extensions have sought to alleviate these issues by explicitly incorporating boundary or initial condition information into the loss function and by refining residual sampling strategies [6, 7, 1]. In contrast, the present work does not aim to enhance PINNs by providing additional information about a specific system. Instead, we focus on optimizing the PDE residuals themselves to assist a differentiable-physics framework, while leaving boundary and initial conditions open to be modeled in a more general inference setting in future work.

In parallel, the emerging field of differentiable physics [5] leverages differentiable programming to integrate numerical solvers directly into neural network pipelines. By unrolling implicit or explicit updates of classical solvers within the computational graph, these methods enable gradients to flow through physically grounded operations

These two paradigms are inherently compatible. PINNs incorporate continuous physical constraints, while differentiable solvers provide discrete operator consistency. Integrating both yields a unified

1st Workshop on Differentiable Systems and Scientific Machine Learning @ EurIPS 2025

framework in which neural predictions are guided not only by residual minimization but also by the differentiable behavior of the numerical solver itself.

This work introduces a hybrid method that combines PINNs with differentiable solvers to improve the stability and accuracy of physics-based learning. The core idea is to augment the gradients from PINN loss with gradients from differentiable solvers, by unrolling differentiable numerical updates in the same fashion as Solver in the loop [9]. In this setting, the network acts as a correction operator refining the solver output, while the solver provides physically meaningful gradients to the network. The result is a training process that learns from both the continuous PDE residuals and the discrete dynamics of the solver.

An intuition behind this integration arises from a connection with the family of generative models called denoising models, such as denoising autoencoders [2] and more recently diffusion probabilistic models [4]. Empirically, denoising models have proven effective compared to purely latent generative methods, achieving data consistency without increasing the number of learnable parameters. A theoretical analysis [2, 3] showed that denoising autoencoders can be interpreted as learning a stationary Markov process in which the learned network approximates the transition operator that maps a corrupted sample toward its clean counterpart.

A similar analogy can be drawn for solver-in-the-loop learning [9]. In this setting, the trainable model learns a mapping between an imperfect physical representation,produced by a coarse or partially converged solver step, and a refined, physically consistent one. Ideally, this process approaches a continuously finer representation of the solution space, as in the limit where the PDE operator is defined for every point—precisely the regime addressed by PINNs. In this sense, our hybrid approach leverages the exact continuous description of the PDE (from PINNs) together with the iterative refinement of differentiable solvers, combining continuous and discrete physics to improve training efficiency without additional data or parameters.

2 Method

2.1 Problem Setup

We consider a class of partial differential equations (PDEs) of the form

$$\mathcal{N}[u](x,t) = 0, \qquad (x,t) \in \Omega \times [0,T], \tag{1}$$

subject to boundary and initial conditions

$$\mathcal{B}[u](x,t) = g(x,t), \qquad (x,t) \in \partial\Omega \times [0,T]. \tag{2}$$

Here, $u(x,t)$ denotes the field variable of interest (e.g., pressure, concentration, or potential), $\mathcal{N}[\cdot]$ represents the differential operator governing the physical law, and $\mathcal{B}[\cdot]$ encodes the boundary or initial constraints. Our goal is to approximate $u(x,t)$ with a neural model $u_\theta(x,t)$ that can generalize across different spatial and temporal discretizations and varying solver accuracies.

The proposed hybrid framework is evaluated on three representative PDE systems:

(1) Single-phase Darcy flow.

$$\phi p_t - \frac{1}{\mu}\nabla \cdot (K(x,y)\nabla p) = q(x,y) \tag{3}$$

Training data are generated from fine-resolution numerical simulations, while testing is performed on coarser spatial and temporal grids. This setup evaluates the model's ability to generalize across discretization levels.

(2) Diffusion equation.

$$u_t - \nabla\cdot (D(x,y)\nabla u) = s(x,y,t), \qquad u(x,0) = u_0(x,y), \tag{4}$$

As in the Darcy case, the model is trained on fine spatio-temporal resolutions and tested on coarser ones to assess the stability of the learned coarse-to-fine correction under discretization shifts.

(3) Poisson equation.

$$-\nabla^2 u = f(x, y), \qquad u|_{\partial\Omega} = g(x, y), \tag{5}$$

where f is a forcing term. Here, the focus is on solver quality rather than grid resolution: the network is trained using low-tolerance solver outputs and evaluated on high-tolerance solutions.

These three PDE families jointly test the capability of the proposed method to learn mappings between imperfect and refined physical representations, spanning variations in discretization scale, solver tolerance, and temporal–spatial resolution.

2.2 PINN Baseline

Physics-Informed Neural Networks (PINNs) [8] approximate the solution $u(x, t)$ of a PDE by a neural network $u_\theta(x, t)$ whose parameters θ are optimized to satisfy the governing equations and boundary conditions. Instead of relying on labeled data, PINNs enforce the physical constraints by minimizing the residuals of the PDE and the associated boundary or initial conditions at randomly sampled points in the spatio-temporal domain.

The PINN objective adopted in this work focuses solely on enforcing the PDE residual, without explicitly constraining boundary or initial conditions. This choice is motivated by the goal of developing a model capable of handling variations and perturbations in boundary regimes without retraining for each specific configuration.

$$\mathcal{L}_{\text{PINN}} = \frac{1}{N_r} \sum_{i=1}^{N_r} \left| \mathcal{N}[u_\theta](x_i, t_i) \right|^2 \tag{6}$$

2.3 Differentiable Solver Consistency

Following the *solver-in-the-loop* formulation [9], we view the discretized PDE as a transition operator that evolves a discrete state over time or pseudo-time. Let $\mathcal{P}$ denote the continuous PDE operator and $\mathcal{P}_R$ and $\mathcal{P}_s$ two of its discrete counterparts, representing respectively a high-fidelity (reference) and a coarse (approximate) solver. Given a spatial domain $\Omega \subset \mathbb{R}^d$ and a time step Δt, their discrete evolutions satisfy

$$\mathbf{r}_{t+\Delta t} = \mathcal{P}_R(\mathbf{r}_t), \qquad \mathbf{s}_{t+\Delta t} = \mathcal{P}_s(\mathbf{s}_t), \tag{7}$$

where $\mathbf{r}_t \in \mathcal{R}$ and $\mathbf{s}_t \in \mathcal{S}$ are the reference and coarse solutions at time t.

Because $\mathcal{P}_R$ and $\mathcal{P}_s$ rely on different numerical approximations, their trajectories diverge: $\mathcal{P}_s(\mathcal{T}\mathbf{r}_t) \neq \mathcal{T}\mathbf{r}_{t+\Delta t}$, with $\mathcal{T}$ denoting an interpolation operator that maps the reference state to the coarse manifold. This divergence can be quantified by

$$\mathcal{L}(\mathbf{s}_t, \mathcal{T}\mathbf{r}_t) = \|\mathbf{s}_t - \mathcal{T}\mathbf{r}_t\|_2. \tag{8}$$

The goal is to learn a corrective operator $\mathcal{C}_\theta : \mathcal{S} \to \mathcal{R}$ that maps a coarse, imperfect state to a refined one such that the evolved sequence produced by $\mathcal{P}_s \circ \mathcal{C}_\theta$ remains closer to the reference manifold than $\mathcal{P}_s$ alone.

In our implementation, $\mathcal{P}_s$ is a differentiable numerical solver, and $\mathcal{C}_\theta$ is a neural correction module. The network learns, through gradient descent, to reduce the error introduced by the coarse solver by refining its outputs.

3 Results

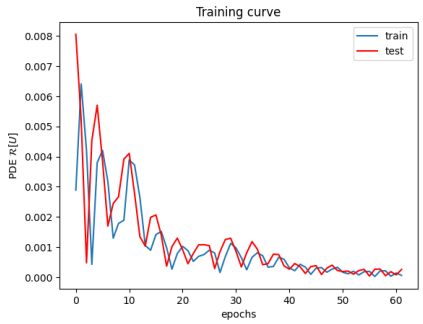
Table 1 and figure 1 shows the results of the experiments for each PDE.

4 Future Directions

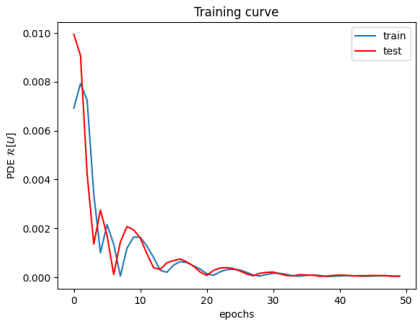
The present work establishes a foundation for combining PINNs and differentiable solvers to learn mappings between imperfect and refined physical representations. Several extensions are planned to further expand this framework:

Table 1: Summary of results with this method

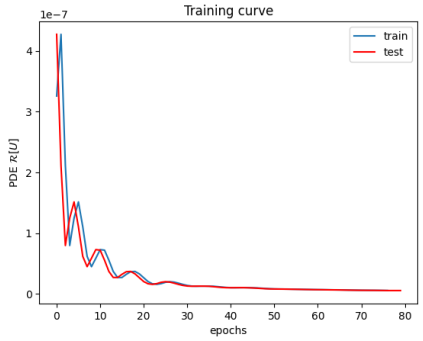
PDE	Fine version	Coarse version	Train loss $\mathcal{R}$	Test loss $\mathcal{R}$
Single-phase Darcy flow	$\Delta t = 0.01$ $\Delta s = 0.05$ (same explicit solver)	$\Delta t = 0.005$ $\Delta s = 0.025$ (same explicit solver)	3.247×10^{-4}	9.596×10^{-5}
Diffusion equation	$\Delta t = 0.01$ $\Delta s = 0.2$ (same CG-solver)	$\Delta t = 0.005$ $\Delta s = 0.1$ (same CG-solver)	3.252×10^{-6}	4.921×10^{-6}
Poisson equation	$\Delta s = 0.2$ (CG-solver: tol=10^{-5} max_iter=10^3)	$\Delta s = 0.2$ (CG-solver: tol=5 max_iter=10^2)	5.669×10^{-9}	5.543×10^{-9}



(a) Training curve for single-phase Darcy flow



(b) Training curve for the Diffusion equation



(c) Training curve for the Poisson equation

Figure 1: Results

Neural architecture design. Systematic exploration of alternative network architectures (e.g., Fourier neural operators, graph-based solvers, or hybrid convolutional MLPs) may improve generalization across heterogeneous discretizations and accelerate convergence.

Inverse problem formulation. The differentiable-physics component makes the framework naturally suited for inverse problems, where unknown physical coefficients (e.g., permeability fields or diffusion constants) can be inferred by backpropagating through the solver network.

Perturbation modeling via boundary inference. Future work will consider boundary and initial conditions as latent variables to be inferred rather than fixed constraints, allowing the model to capture and reason about perturbations in external conditions without retraining.

These directions aim to develop a more flexible, uncertainty-aware, and inference-capable hybrid physics-learning framework.

References

[1] G. Akrivis, C. G. Makridakis, and C. Smaragdakis. Runge-kutta physics informed neural networks: Formulation and analysis, 2025.

[2] Y. Bengio, L. Yao, G. Alain, and P. Vincent. Generalized denoising auto-encoders as generative models, 2013.

[3] Y. Bengio, Éric Thibodeau-Laufer, G. Alain, and J. Yosinski. Deep generative stochastic networks trainable by backprop, 2014.

[4] J. Ho, A. Jain, and P. Abbeel. Denoising diffusion probabilistic models, 2020.

[5] P. Holl, V. Koltun, and N. Thuerey. Learning to control pdes with differentiable physics, 2020.

[6] A. D. Jagtap and G. E. Karniadakis. Extended physics-informed neural networks (xpinns): A generalized space-time domain decomposition based deep learning framework for nonlinear partial differential equations. *Communications in Computational Physics*, 28(5):2002–2041, 2020.

[7] L. D. McClenny and U. M. Braga-Neto. Self-adaptive physics-informed neural networks. *Journal of Computational Physics*, 474:111722, Feb. 2023.

[8] M. Raissi, P. Perdikaris, and G. E. Karniadakis. Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations. *J. Comput. Phys.*, 378:686–707, 2019.

[9] K. Um, R. Brand, Yun, Fei, P. Holl, and N. Thuerey. Solver-in-the-loop: Learning from differentiable physics to interact with iterative pde-solvers, 2021.