Solving Nonlinear Partial Differential Equations Using Physics-Informed Neural Networks

BTech Project 3rd Sem Report

Ramagiri Jaidhar
EP23BTECH11021

భారతీయ సాంకేతిక విజ్ఞాన సంస్థ హైదరాబాద్
भारतीय प्रौद्योगिकी संस्थान हैदराबाद
Indian Institute of Technology Hyderabad

Department of Physics
Supervisor: Dr. Kirit Makwana

Abstract

Physics-Informed Neural Networks (PINNs) provide a novel approach to solving partial differential equations (PDEs) by embedding physical laws into the training process of neural networks. In this project, PINNs are applied to a range of nonlinear PDEs including the Burgers equation, the Riemann problem, the 3D heat equation, and the kinematic dynamo model. These problems test the capability of PINNs to handle nonlinear dynamics, discontinuities, multidimensional inputs, and vector-valued outputs. Results show that PINNs can accurately approximate physical fields and capture underlying dynamics without explicit discretization, demonstrating potential as a flexible alternative to traditional numerical solvers.

1 Introduction

Partial Differential Equations (PDEs) are fundamental in describing various physical systems such as fluid flow, heat transfer, and electromagnetism. Traditional numerical methods like finite difference or finite element schemes rely on discretization, which becomes computationally expensive for high-dimensional or complex geometries. Physics-Informed Neural Networks (PINNs) offer a mesh-free alternative by incorporating governing equations directly into the neural network loss function.

The objective of this project is to implement and analyze PINNs to solve different nonlinear PDEs of increasing complexity. Starting with the Burgers equation as a benchmark, the study progresses to the Riemann problem, the 3D heat equation, and finally attempts a kinematic dynamo model to test vector-field learning capabilities.

2 Theoretical Background

A Physics-Informed Neural Network is a feed-forward network trained not only on data but also on the physical laws represented by PDEs. The total loss function typically includes:

- **Data Loss:** Difference between predicted and known boundary/initial data.
- **Physics Loss:** PDE residual computed via automatic differentiation.
- **Boundary Loss:** Error at domain boundaries to enforce physical constraints.

The governing equation is written in the following general form:

$$\mathcal{N}[u(x,t)] = 0,$$

where $\mathcal{N}$ is a differential operator. PINNs minimize the mean squared residual $\mathcal{N}$ at randomly sampled collocation points.

3 Problem Statements

3.1 Burgers Equation

The 1D viscous Burgers equation,

$$u_t + uu_x = \nu u_{xx},$$

was used as an initial test problem. It is a nonlinear PDE that combines advection and diffusion effects and serves as a simplified model for fluid dynamics.

3.2 Riemann Problem

The Riemann problem represents a system of hyperbolic conservation laws:

$$\begin{aligned} \rho_t + (\rho u)_x &= 0, \\ (\rho u)_t + (\rho u^2 + p)_x &= 0. \end{aligned}$$

This case involves discontinuities and shock-like behavior, making it a challenging test for PINNs in capturing sharp gradients.

3.3 3D Heat Equation

The three-dimensional heat equation by

$$u_t = \alpha(u_{xx} + u_{yy} + u_{zz}),$$

used to verify the ability of the PINN to handle multi-dimensional input (x, y, z, t) and to learn smooth spatiotemporal diffusion patterns.

3.4 Kinematic Dynamo Model

The magnetic induction equation is expressed as

$$\frac{\partial \mathbf{B}}{\partial t} = \nabla \times (\mathbf{u} \times \mathbf{B}) + \eta \nabla^2 \mathbf{B}.$$

This vector PDE models the evolution of magnetic fields in a conducting fluid and was used to explore PINNs' capability to predict vector-valued outputs.

4 Methodology

The models were implemented using the PyTorch framework on GPU. Each PINN consisted of fully connected layers with different types of activation functions. Inputs included spatial and temporal coordinates, and outputs were scalar or vector fields depending on the PDE.

The total loss was defined as the weighted sum of physics and boundary losses. Automatic differentiation was used to compute derivatives, enabling direct calculation of PDE residuals. Training was carried out using the Adam optimizer followed by L-BFGS for fine-tuning.

5 Results and Discussion

The PINN successfully reproduced analytical or reference solutions for Burgers and 3D heat equations with low residual error. For the Riemann problem, results showed smoothed shock transitions due to the difficulty of representing discontinuities with smooth neural functions. The partially completed kinematic dynamo model demonstrated stable learning of vector components but required further tuning for convergence.

Figures comparing predicted and analytical solutions, as well as loss convergence plots, are presented for each case.

6 Observations and Challenges

- PINNs handled smooth and continuous problems effectively but struggled near discontinuities.
- Higher-dimensional PDEs required larger networks and more collocation points.
- Training stability depended strongly on normalization and learning rate scheduling.
- The vector-field case (dynamo) revealed challenges in enforcing divergence-free conditions.

7 Conclusion

The project demonstrated the feasibility of using Physics-Informed Neural Networks to solve a variety of nonlinear PDEs. PINNs provided a flexible, mesh-free framework for learning both scalar and vector field dynamics. While the approach works well for continuous problems, improvements are needed for shock capturing and complex coupled systems.

8 Future Work

Future efforts will focus on:

- Completing and optimizing the kinematic dynamo model.
- Incorporating adaptive sampling and loss weighting schemes.
- Exploring advanced PINN architectures such as XPINNs or DeepONets.
- Extending to full magnetohydrodynamics (MHD) equations.

References

[1] 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*, Journal of Computational Physics, vol. 378, pp. 686–707, 2019.

[2] K. Sun et al., *Applications of Physics-Informed Neural Networks in Computational Physics*, Computer Methods in Applied Mechanics and Engineering, 2022.