PHYSICS INFORMED NEURAL NETWORKS FOR PARTIAL DIFFERENTIAL EQUATION SOLVING

UNIVERSIDAD NACIONAL DE COLOMBIA

Alejandro José Yepes Peñaranda

ayepesp@unal.edu.co

Prof. Frank Fonseca Fonseca

frfonsecaf@unal.edu.co

Programa de Pregrado en Física

All the project here

Abstract

Physics Informed Neural Networks (PINNs) are a new and interesting methodology of Machine Learning training, constructed to correctly predict the solution of differential problems which consist not only of partial differential equations (PDEs) but also that include boundary or initial conditions. The goal of the present project is to develop a python module dedicated to easily constructing this type of architecture that is wide enough in order to adapt to many possible differential equation systems as well as the multiple conditions to impose. With this module fully developed, the goal was then to test it using multiple systems such as simple harmonic oscillator, Lotka-Volterra equations, reaction diffusion system among other, then to applied the finding for computational fluid simulation using the Kovasznay formulation of the Navier-Stokes equations for modeling its interaction with an obstacle, Multiple great results were obtained, reassuring the great potential that this technology has in many fields that require a ton of computational resources.

Introduction

PDEs are a fundamental tool for modelling complex phenomena and are used extensively in a wide range of fields, including physics, engineering, material science, finance and biology. With the massive popularization of artificial intelligence tools in the recent years, neural networks (NNs) had arisen as viable and innovative way for PDE solution approximation. NNs can learn the underlying structure of the system and accurately predict the solutions without a regularization (grid) of the domain.

The module was developed in Python using pytorch. The selected architecture is a fully connected feed forward NN due to its property of universal approximator. The differential peculiarity of a PINN resides in its loss function, defined with the residuals of the treated differential problem as:

$$L = \sum_i MSE(ResPDE_i(\hat{y}), 0) + \sum_j \alpha_j MSE(ResBC_j(\hat{y}_\partial), y_\partial)$$

Results

PINNs In Ordinary Differential Equations

A Simple Example:

This example is proposed to test the impact that different optimizer algorithms have in the convergence of the PINN modules. The differential problem is:

$$\frac{d^2x}{dt^2} = -\pi^2 \sin(\pi t + \phi), \qquad t \in [-1,1]$$

With solution:

$$x(t) = \sin(\pi t + \phi)$$

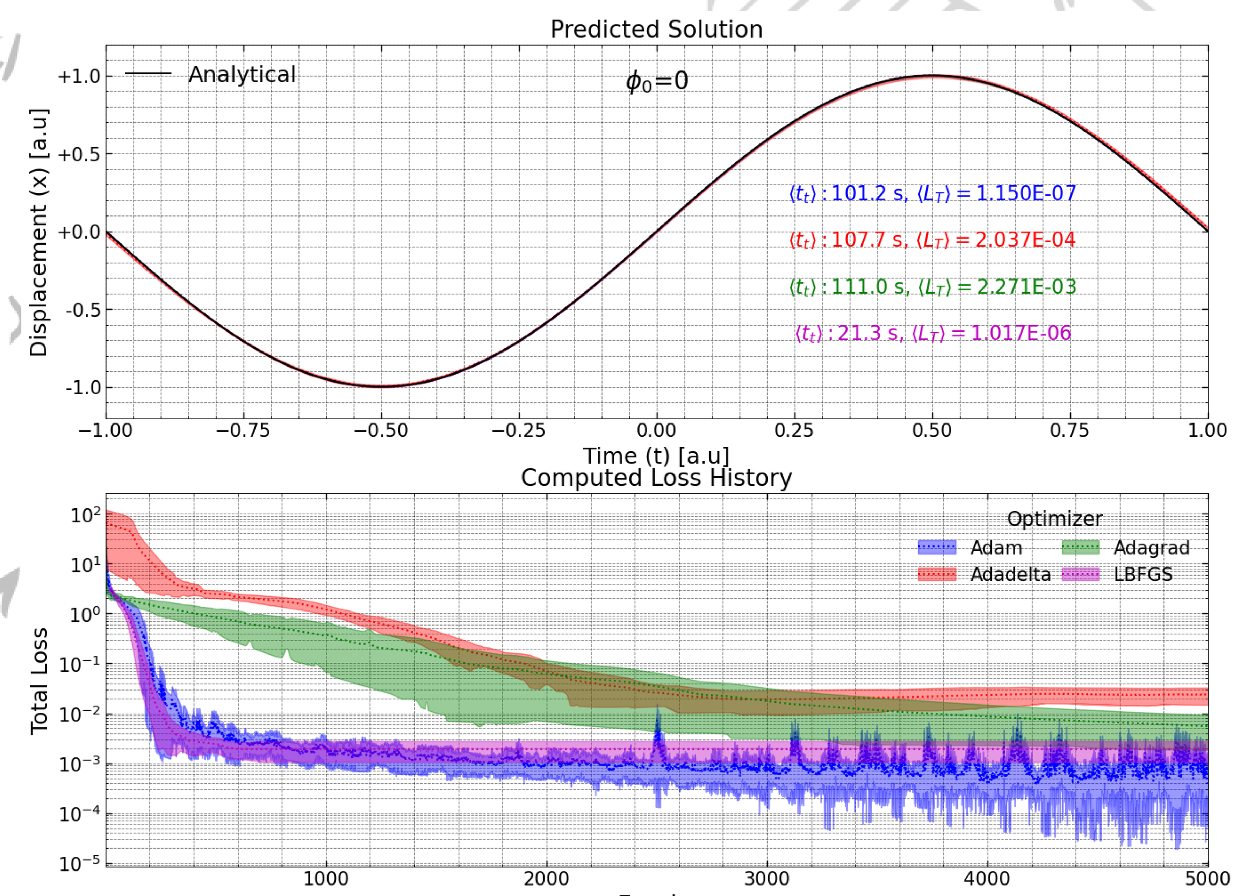

Figure 1:.Example of training results for Optimizer test using [1,50x2,20,50x2,1]

Conclusions

The constructed module can be easily adapted for solving multiple differential problems with no restrictions about boundary conditions neither number of equations nor variables, meaning that it is sufficiently general to be considered a suitable tool for solving a wide range of problems. As well as other NN implementations the convergence velocity of it is strongly bounded to parameters such as optimizer algorithms, activation functions and number of neurons/layers.

PINNs are a viable alternative to traditional methods for generating high-complexity simulations, although they come also at the cost of heavy computational requirements and training time. A better-optimize PINN option shall be studied including other architectures such as convolutional NNs.

PINNs In Partial Differential Equations

The biggest potential of PINNs is in their application to PDEs, in the following figure there are some examples

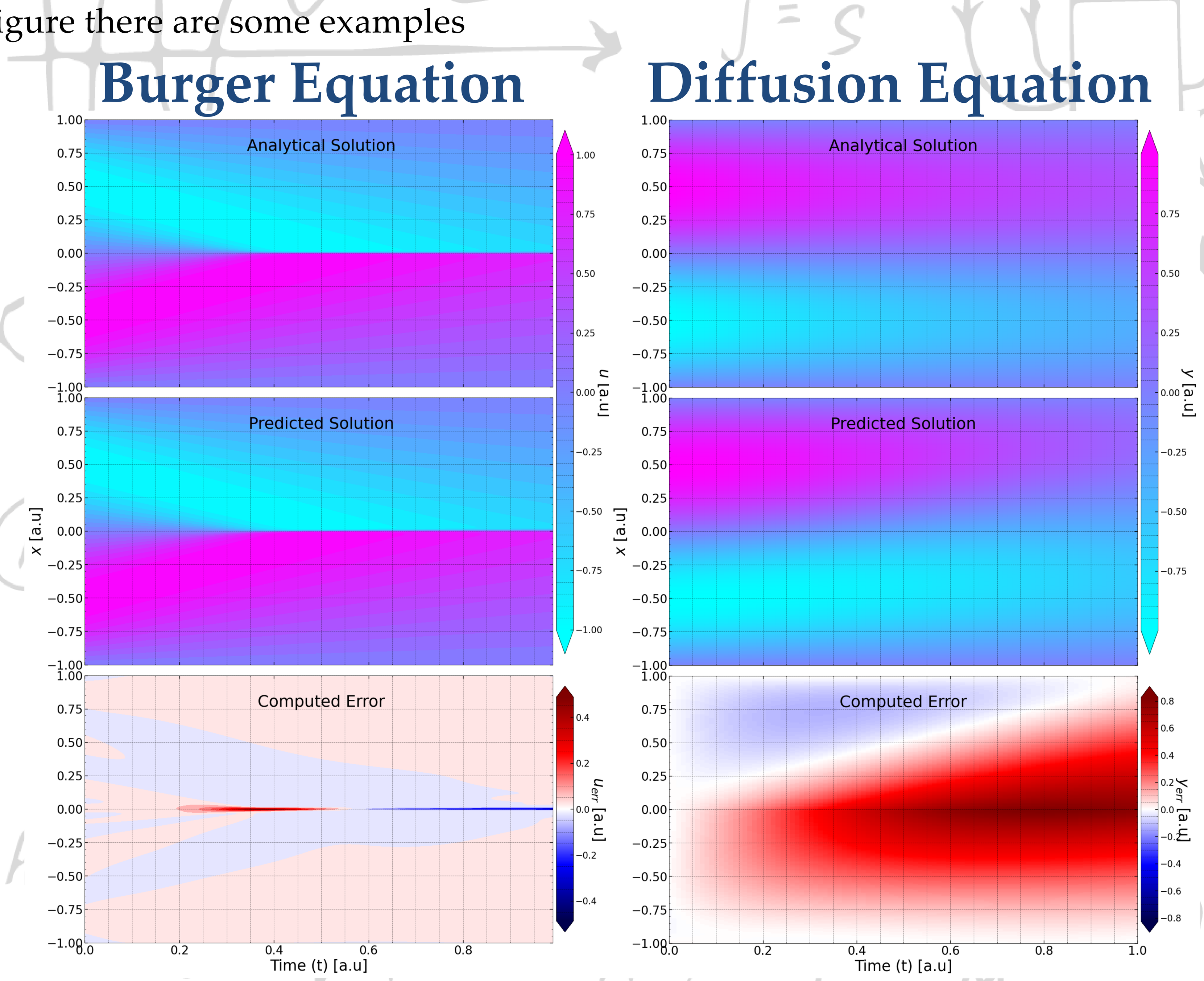

Figure 2: PINN applied to PDE examples

PINNs For Fluid Dynamics Simulation

PINNs are great for complex systems, in particular fluid dynamics is a field mostly computationally-driven due to its importance in multiple industries. The following example shows the simulation results for the interaction of a fluid with and obstacle.

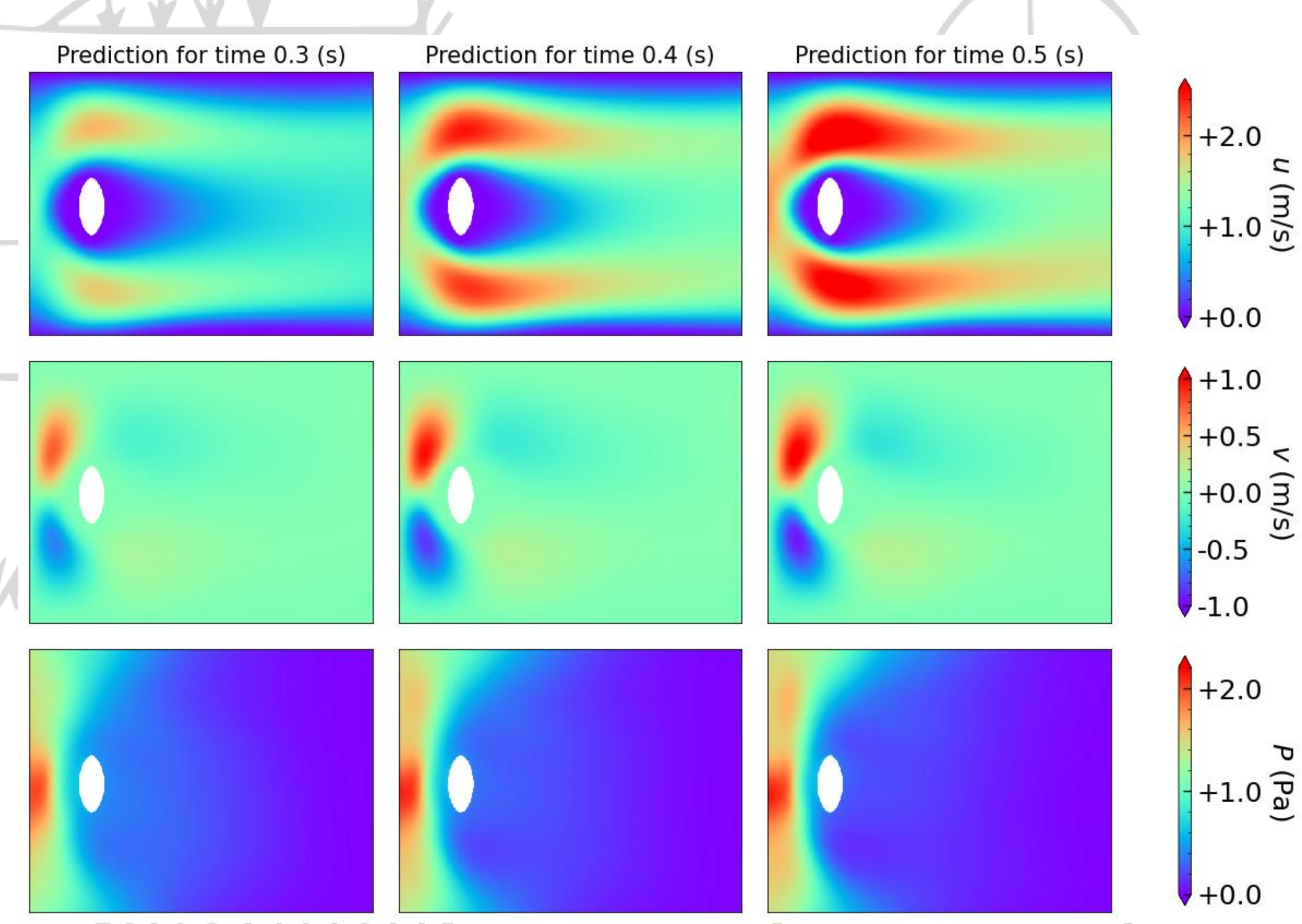

Figure 3:.Kovasznay Flow Simulation Using PINNs

References

Yadav, N., Yadav, A., & Kumar, M. (2015). *An introduction to neural network methods for differential equations* (Vol. 1, p. 114). Berlin: Springer.

Toscano, Juan Diego (2022). *Learning-Python-Physics-Informed-Machine-Learning-PINNs-DeepONets.*