Data-Driven Modeling of Landau Damping by Physics-Informed Neural Networks

Hunt Feng

November 14, 2022

Abstract

In this report, we successfully simulate the Landau damping using finite difference algorithm. The physics-informed neural network (PINN) is constructed. The PINN prediction for the first two time steps is good.

1 Introduction

The fluid description of plasma is widely used today to mitigate the computational cost of kinetic models. It is derived by taking the velocity moments of the kinetic Vlasov equation. However, one major difficulty in constructing the fluid closures in collisionless plasmas is that it typically requires very nontrivial physical and mathematical analysis applied to the specific regime. With the prosperity of Artificial Intelligence in the past decade, the question naturally arises: Can Machine Learning assists in completing the challenging task by exploring the simulation data.

2 Landau Damping

Consider a collisionless plasma in the absence of magnetic field, the dynamics of the plasma can be described by the particle distribution $f_s(\mathbf{x}, \mathbf{v}, t)$, where $s = \{electron, ion\}$ stands for species. The evolution of distribution function is governed by the Vlasov equation,

$$\frac{\partial f_s}{\partial t} + \mathbf{v}_s \cdot \boldsymbol{\nabla}_{\mathbf{r}} f_s + \frac{q_s}{m_s} \mathbf{E} \cdot \boldsymbol{\nabla}_{\mathbf{v}} f_s = 0 \tag{1}$$

where $\boldsymbol{\nabla}_{\mathbf{r}} = (\partial_x, \partial_y, \partial_z)$ and $\boldsymbol{\nabla}_{\mathbf{v}} = (\partial_{v_x}, \partial_{v_y}, \partial_{v_z})$. In this report, we consider one-dimensional model in $x - v_x$ space for convenience.

To complete the system, we introduce the Poisson equation for electric potential,

$$\nabla^2 \phi = -\rho/\epsilon_0 \tag{2}$$

For convenience, we set the vacuum permittivity $epsilon_0 = 1$, and the charge density ρ is defined by,

$$\rho = \sum_s q_s n_s \tag{3}$$

where the number density $n_s(x, t)$ is defined as

$$n_s(x, t) = \int f_s(x, v_s, t) dv_s \tag{4}$$

By Gauss's law, we can get electric field,

$$E(x, t) = -\nabla \phi \tag{5}$$

Equations (1)-(5) forms a complete PDE system. We can investigate the evolution of distribution $f_s(x, v_s, t)$.

Starting from a plasma at equilibrium. We assume the ions are too heavy to move in a short period of time. Then two perturbation modes is given to the number density of electron. Then the initial number densities of ions and electrons are

$$n_e(x, t = 0) = n_0(1 + A_1 \cos(k_1 x) + A_2 \cos(k_2 x + \varphi)) \tag{6}$$
$$n_i(x, t = 0) = n_0 \tag{7}$$

where n_0 is the number density of each species at equilibrium. k_1 and k_2 are the wavenumbers of the two perturbation modes, A_1 and A_2 are their amplitudes, and φ is a random phase.

Using the initial setup parameters listed in Table.1, we got the results as shown in Fig.1.

Table 1: Parameters of finite difference simulation.

k_1	k_2	A_1	A_2	φ
0.6	1.2	0.05	0.4	0.38716

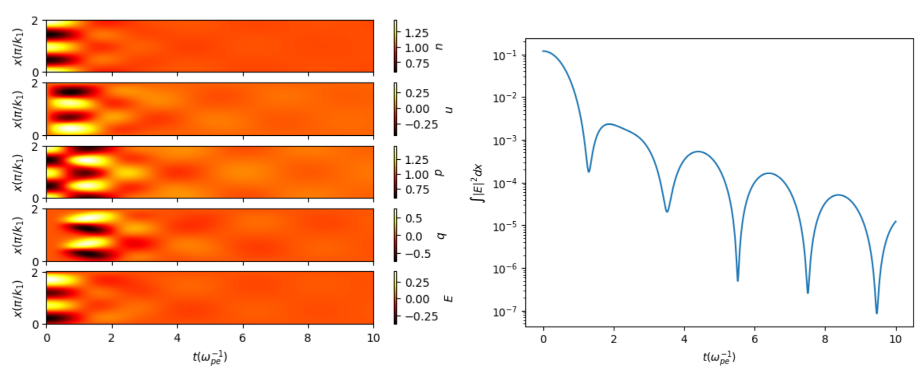


Figure 1: The simulation is run for 10 unit time in the region $\Omega = [0, 2\pi/k_1]$

3 Physics-Informed Neural network (PINN)

Physics-informed neural network is a type of neural network which embeds physical information. The loss function of PINN consists of residuals to the PDE system which governs the physics. [1] See part C of Fig.(2).

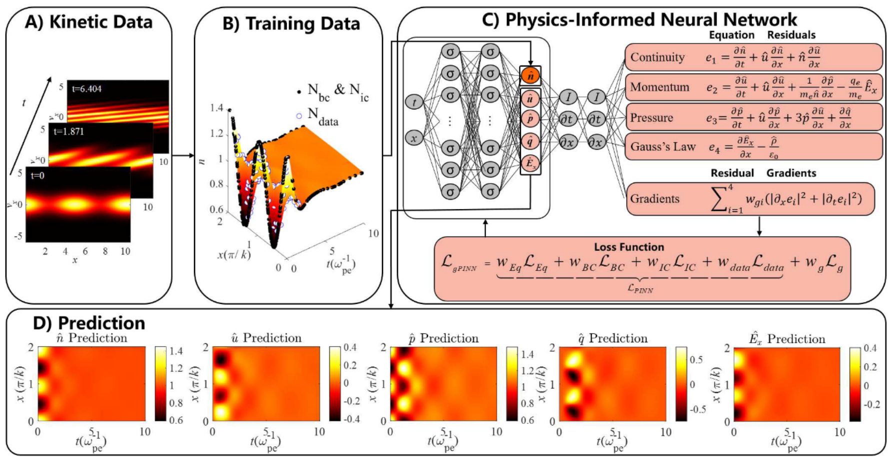


Figure 2: Physics-informed neural network (PINN) architecture for the multi-moment fluid model with an implicit fluid closure learned from the kinetic simulation data. The whole procedure includes A) kinetic simulation data generation, B) sparse sampling of training data, C) PINN construction with the constraints of different moment equation residuals and their gradients, and D) parameter prediction.

4 Methodology

4.1 Data preparation

The PINN as a parametric function approximator can be represented by a non-linear function:

$$\hat{n}, \hat{u}, \hat{p}, \hat{q}, \hat{E} = \hat{\mathbf{F}}(t, x, \theta) \tag{8}$$

where $\theta = \{\mathbf{W}, \mathbf{b}\}$ is the weight matrix and the bias vector of the neural network.

This means for each input in the array $\{t_n, x_n\}_{n\geq 0}$, the PINN produces an array of length 5, with each number corresponding to $\hat{n}(t_n, x_n)$,$\hat{u}(t_n, x_n)$, $\hat{p}(t_n, x_n)$, $\hat{q}(t_n, x_n)$, $\hat{E}(t_n, x_n)$.

We will use the finite-difference simulation data as our training and testing dataset.

4.2 Construction of loss function

The physical partial differential equation (PDE) residuals are incorporated into the loss function of the neural network as regularization, transforming the process of solving PDEs into an optimization problem by constraining the space of permissible solutions.

We want the neural network to learn the closure relation to the fluid description of plasma

$$
\begin{aligned}
\frac{\partial n}{\partial t} + u\frac{\partial n}{\partial x} + n\frac{\partial u}{\partial x} &= 0 \\
\frac{\partial u}{\partial t} + u\frac{\partial u}{\partial x} + \frac{1}{m_e n}\frac{\partial p}{\partial x} - \frac{q_e}{m_e}E &= 0 \\
\frac{\partial p}{\partial t} + u\frac{\partial p}{\partial x} + 3p\frac{\partial u}{\partial x} + \frac{\partial q}{\partial x} &= 0 \\
\frac{\partial E}{\partial x} &= \frac{\rho}{\epsilon_0}
\end{aligned}
\tag{9}
$$

The first equation is the continuity equation for number density, the second equation describes the conservation of momentum of the plasma, the third one is the conservation of pressure. The last one is the Gauss's law. We see that the fluid description involves higher and higher velocity moments of the particle distribution. To close the system, we must need a closure relation.

The loss function is defined as

$$
L_{PINN} = w_{eq}L_{eq} + w_{bc}L_{bc} + w_{ic}L_{ic} + w_{data}L_{data} \tag{10}
$$

where w_{eq}, w_{bc}, w_{ic} and w_{data} are the weights of each loss function respectively. In this study, we choose the weights $w_{eq} = w_{bc} = w_{ic} = w_{data} = 1$. In particular, we seek to minimize the residuals of the fluid moment equations,

$$
\begin{aligned}
e_1 &= \frac{\partial \hat{n}}{\partial t} + \hat{u}\frac{\partial \hat{n}}{\partial x} + \hat{n}\frac{\partial \hat{u}}{\partial x} \\
e_2 &= \frac{\partial \hat{u}}{\partial t} + \hat{u}\frac{\partial \hat{u}}{\partial x} + \frac{1}{m_e \hat{n}}\frac{\partial \hat{p}}{\partial x} - \frac{q_e}{m_e}\hat{E} \\
e_3 &= \frac{\partial \hat{p}}{\partial t} + \hat{u}\frac{\partial \hat{p}}{\partial x} + 3\hat{p}\frac{\partial \hat{u}}{\partial x} + \frac{\partial \hat{q}}{\partial x} \\
e_4 &= \frac{\partial \hat{E}}{\partial x} - \frac{\hat{\rho}}{\epsilon_0}
\end{aligned}
\tag{11}
$$

$$
L_{eq} = \frac{1}{N_{eq}} \sum_{j=1}^{N_{eq}} \sum_{i=1}^{4} |e_i(t_j, x_j)|^2, t_j \in [0, T], x_j \in \Omega \tag{12}
$$

Here e_1 denotes the continuity equation residual, e_2 the momentum equation residual, e_3 the pressure equation residual, and e_4 the Gauss's law equation residual. N_{eq} is the number of the trained data for L_{eq}. By minimizing the

Table 2: Summary of PINN parameters.

Hidden layers, and No. of neurons	Optimizer	Activation function
5 and 50	Adam	Swish
Learning rate	Batch size	Epochs
0.01	10000	10000

residuals to the fluid equations Eq.(11), the neural networks "learns" the closure relation to the fluid equations, given the boundary conditions and initial conditions are known and sparsely sampled,

$$L_{bc} = \frac{1}{N_{bc}} \sum_{j=1}^{N_{bc}} \sum_{i=1}^{5} \left| \hat{\mathbf{F}}(t_j, x_j) - \mathbf{F}(t_j, x_j) \right|^2, t_j \in [0, T], x_j \in \partial\Omega \tag{13}$$

$$L_{ic} = \frac{1}{N_{ic}} \sum_{j=1}^{N_{ic}} \sum_{i=1}^{5} \left| \hat{\mathbf{F}}(t_j, x_j) - \mathbf{F}(t_j, x_j) \right|^2, t_j = 0, x_j \in \Omega \tag{14}$$

Lastly, we sampled the number density in the first few time steps.

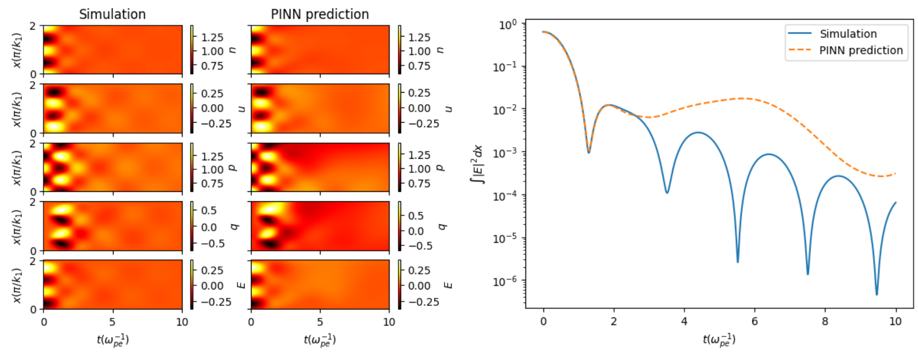
$$L_{data} = \frac{1}{N_{data}} \sum_{j=1}^{N_{data}} \left| \hat{n}(t_j, x_j) - n(t_j, x_j) \right|^2, t_j \in [0, T/5], x_j \in \Omega \tag{15}$$

When computing the loss function, we can take advantage of the automatic differentiation in neural network.

4.3 Construction of PINN

The PINN consists of 5 hidden layers with 50 neurons in each layer. The activation function is Swish. The optimizer is chosen to be Adam with constant learning rate 0.01. The batch size in each training epoch is 10000. Finally, 10000 epochs were ran.

5 Result



The first two time step of the PINN prediction is good.

6 Conclusions and Discussions

By embedding the fluid equations Eq.(9) into the loss function, the physical information is incorporated into the neural network. Minimizing the loss function, the neural network learns the closure relation to Eq.(9). The PINN prediction is good in the first two time steps.

According to the paper, we can improve the accuracy of PINN by introducing one more component to the loss function

$$L_{gPINN} = w_{eq} L_{eq} + w_{bc} L_{bc} + w_{ic} L_{ic} + w_{data} L_{data} + \mathbf{w}_g L_g \tag{16}$$

where

$$\mathbf{w}_g L_g = \frac{1}{N_g} \sum_{j=1}^{N_g} \sum_{i=1}^{4} w_{g_i} \left(\left| \partial/\partial t \; e_i(t_j, x_j) \right|^2 + \left| \partial/\partial x \; e_i(t_j, x_j) \right|^2 \right), t_j \in [0, T], x_j \in \Omega \tag{17}$$

The $\mathbf{w}_g = \{w_{g_1}, w_{g_2}, w_{g_3}, w_{g_4}\}$ is an extra hyper-parameter in the gPINN architecture for optimization.

References

[1] Yilan Qin, Jiayu Ma, Mingle Jiang, Chuanfei Dong, Haiyang Fu, Liang Wang, Wenjie Cheng, and Yaqiu Jin. Data-Driven Modeling of Landau Damping by Physics-Informed Neural Networks, November 2022. arXiv:2211.01021 [astro-ph, physics:physics].