# Course project: Charged particle in a electromagnetic field

Contributed by Zhen Zhang
Division of Applied Mathematics, Brown University

## Project statement

Given the trajectory data of a charged particle in a electromagnetic field, we would like to predict its future states using SympNets [1], PNNs [2] and PINNs [3]. The motion of the particle is governed by the Lorentz force

$$m\ddot{x} = q(E + \dot{x} \times B),$$

where $m$ is the mass, $x \in \mathbb{R}^3$ denotes the particle's position, $q$ is the electric charge, $B = \nabla \times A$ denotes the magnetic field, and $E = -\nabla\varphi$ is the electric field with $A, \varphi$ being the potentials. Let $\dot{x} = v$ be the velocity of the charged particle, then the governing equations of the particle's motion can be expressed as

$$\begin{aligned} \begin{pmatrix} \dot{v} \\ \dot{x} \end{pmatrix} &= \begin{pmatrix} -\frac{q}{m^2}\hat{B}(x) & -\frac{1}{m}I \\ \frac{1}{m}I & 0 \end{pmatrix} \nabla H(v, x), \\ H(v, x) &= \frac{1}{2}mv^T v + q\varphi(x), \end{aligned} \tag{1}$$

where

$$\hat{B}(x) = \begin{pmatrix} 0 & -B_3(x) & B_2(x) \\ B_3(x) & 0 & -B_1(x) \\ -B_2(x) & B_1(x) & 0 \end{pmatrix}$$

for $B(x) = (B_1(x), B_2(x), B_3(x))$. Here we test the dynamics with $m = 1$, $q = 1$, and

$$A(x) = \frac{1}{3}\sqrt{x_1^2 + x_2^2} \cdot (-x_2, x_1, 0), \quad \varphi(x) = \frac{1}{100\sqrt{x_1^2 + x_2^2}}$$

for $x = (x_1, x_2, x_3)^T$. Then

$$\begin{aligned} B(x) &= (\nabla \times A)(x) = (0, 0, \sqrt{x_1^2 + x_2^2}), \\ E(x) &= -(\nabla\varphi)(x) = \frac{(x_1, x_2, 0)}{100(x_1^2 + x_2^2)^{\frac{3}{2}}}. \end{aligned}$$

## Dataset

The initial state is chosen to be $v_0 = (1, 0.5, 0)$, $x_0 = (0.5, 1, 0)$, in which case the system degenerates into four-dimensional dynamics, i.e., the motion of the particle is always on a plane. Consequently, we ignore the third dimension and only use the trajectory data in the first two dimensions. The resulting 2d system also satisfies the Poisson structure. The data for $v$ and $x$ on $n = 1500$ time points are generated by a Stormer–Verlet integrator and provided in the plain text file (train.txt and test.txt). The data are sampled on a uniform grid of stepsize $h = 0.1$. The first two columns in the txt file represents the velocity $v$ and the next two columns represents the position $x$.

## Tasks

1. Train a SympNet model on the provided dataset to learn the flow map from $(v_{i-1}, x_{i-1})$ to $(v_i, x_i)$. Use 1200 points for training and the rest 300 points for testing. For the training set, prepare the input-target pair $\mathcal{T}=\{((v_{i-1}, x_{i-1}), (v_i, x_i))\}_1^{1200}$, i.e., the input to the SympNet will be $(x_{i-1}, v_{i-1})$ and the output should approximate $(x_i, v_i)$. After training the neural net-work, predict the system dynamics by applying the SympNet recurrently in time to generate $(\tilde{x}_{1201}, \tilde{v}_{1201}), \cdots,$ $(\tilde{x}_{1500}, \tilde{v}_{1500})$. Plot the predicted trajectory $\tilde{x}$ and ground truth $x$ in the 2d plane and compare them. Calculate the mean squared error (MSE) as a function of time. Does SympNet work well in this case? Which modeling assumption of SympNet is violated?

2. Replace SympNet by PNN and follow the same procedure in task 1. Plot the predicted trajectory $\tilde{x}$ and ground truth $x$ in the phase plane and compare them. Calculate the mean squared error (MSE) as a function of time.

3. Use PINN for estimating the value of $m$ and $q$ with the same training data. Then apply a numerical integrator on the dynamical system with learned parameter to extrapolate in time. Plot the predicted trajectory $\tilde{x}$ and ground truth $x$ in the phase plane and compare them. Calculate the mean squared error (MSE) as a function of time.

4. Compare the extrapolation results of SympNet, PNN and PINN. Discuss the different modeling assumptions. Does stronger modeling assumption always lead to a better predicting accuracy? Why?

## Programming Options

You may use TensorFlow, PyTorch, JAX or DeepXDE for completing the tasks.

## References

[1] P. Jin, Z. Zhang, I. G. Kevrekidis, and G. E. Karniadakis. Learning poisson systems and trajectories of autonomous systems via poisson neural networks. *arXiv preprint arXiv:2012.03133*, 2020.

[2] P. Jin, Z. Zhang, A. Zhu, Y. Tang, and G. E. Karniadakis. Sympnets: Intrinsic structure-preserving symplectic networks for identifying hamiltonian systems. *Neural Networks*, 132:166–179, 2020.

[3] 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*, 378:686–707, 2019.