IMPERIAL

Discrete time Toda-lattice simulation using a Physics Informed Neural Net

Conrad Trey Indradjaja

Introduction

The **Toda lattice** is a models a 1D chain whose N nodes/particles follow a Hamiltonian $H(\vec{p},\vec{q})$ consisting with an exponential potential energy term.

$$H(\vec{p},\vec{q}) = \sum_{j=1}^{N} \frac{p_j^2}{2} + \sum_{j=1}^{N} V_{toda}(q_{j+1} - q_j)$$

where $V_{toda}(r) = e^r - r - 1$, $\vec{p}$ and $\vec{q}$ represent the momentum and position vectors, and $p_0 = p_N$ and $q_0 = q_N$ (periodic boundary conditions) [3]. This corresponds to the following governing equations:

$$\dot{q}_j = p_j; \dot{p}_j = \ddot{q}_j = e^{q_{j+1}-q_j} - e^{q_j - q_{j-1}}$$

Exact **soliton** solutions can be obtained for the Toda-lattice via Lax Pairs, making the system integrable and non-thermalising [5].

The (maximum) **Lyapunov exponent** measures how much trajectories of a dynamical system $\dot{x} = f(x)$ diverge due to a perturbation u_0 from some initial condition x_0. It is defined by the limit [4]

$$\lambda(x_0, u_0) = \lim_{t\to\infty} \frac{1}{t} \ln \frac{||u_t||}{||u_0||}; u_t = f^t(x_0 + u_0) - f^t(x_0)$$

According to [2], discrete-time simulations introduce chaos into the Toda-lattice leading to positive Lyapunov exponent values indicating chaos.

A **physics informed neural net (PINN)** is a neural network trained on a loss function that incorporates a differential equation and boundary conditions describing a physical law [1].

The goal of this project is to use the differential equations describing the Toda lattice to train a PINN to simulate the Toda-lattice while achieving a closer to 0 Lyapunov exponent than the Runge-Katta method.

Methods

1. **Controlled conditions:** The number of nodes is taken as N = 32, a power of 2. The initial conditions are $q_j(0) = \sin\frac{j\pi}{N}$ and $p_j(0) = 0$. The timespan considered is 100 timesteps with increments of 1 unit.
2. **PINN architecture and loss function:** The PINN takes in two input variables, the time t and index j, and outputs $\hat{q}_j(t), \hat{p}_j(t)$, its predictions for the position and momentum at time t at the index. It has 4 hidden layers with size 40, and uses the SiLU (sigmoid linear unit) activation function. For training, 2000 epochs with a learning rate of 10^{-3} under the Adam optimizer was used. The loss function L is described by

$$L = L_{derivative} + L_0 + L_H + L_{ref}$$

$$L_{derivative} = \frac{1}{N}\sum_{j=1}^{N} (\hat{\dot{q}}_j(t) - \dot{q}_j(t))^2 + (\hat{\dot{p}}_j(t) - \dot{p}_j(t))^2$$

$$L_0 = \frac{1}{N}\sum_{j=1}^{N} (q_j(0) - \hat{q}_j(0))^2 + (p_j(0) - \hat{p}_j(0))^2$$

$$L_H = \frac{1}{N}\sum (\hat{H}(t) - H(0))^2$$

where $\hat{\dot{q}}_j(t)$ and $\hat{\dot{p}}_j(t)$ denote the derivatives obtained from autodifferentiation in the neural network. L_H adds the time invariance of the hamiltonian, and L_{ref} is the mean squared error of its outputs w.r.t the results from solve_ivp.
3. **Maximal Lyapunov Exponent:** Let x_0 denote the fixed initial state vector. To compute the maximal Lyapunov exponent, we evolve the Toda lattice with and without a perturbation u_0 to x_0 using python's solve_ivp. Then

$$\lambda_{approx} = E\{\frac{\ln(||u_T||/||u_0||)}{T}\}$$

the mean from multiple perturbations, where u_T is the difference between the systems at time T. The neural network's exponent is given by

$$\lambda_{PINN} = E\{\frac{1}{T}\ln\frac{||x'_{PINN}(T) - x_{PINN}(T)||}{||x'_{PINN}(0) - x_{PINN}(0)||}\}$$

where $x'_{PINN}(T)$ denotes the PINN state vector outputted at time T when retrained on the perturbed initial condition. $E\{\cdot\}$ denotes expectation.

Results

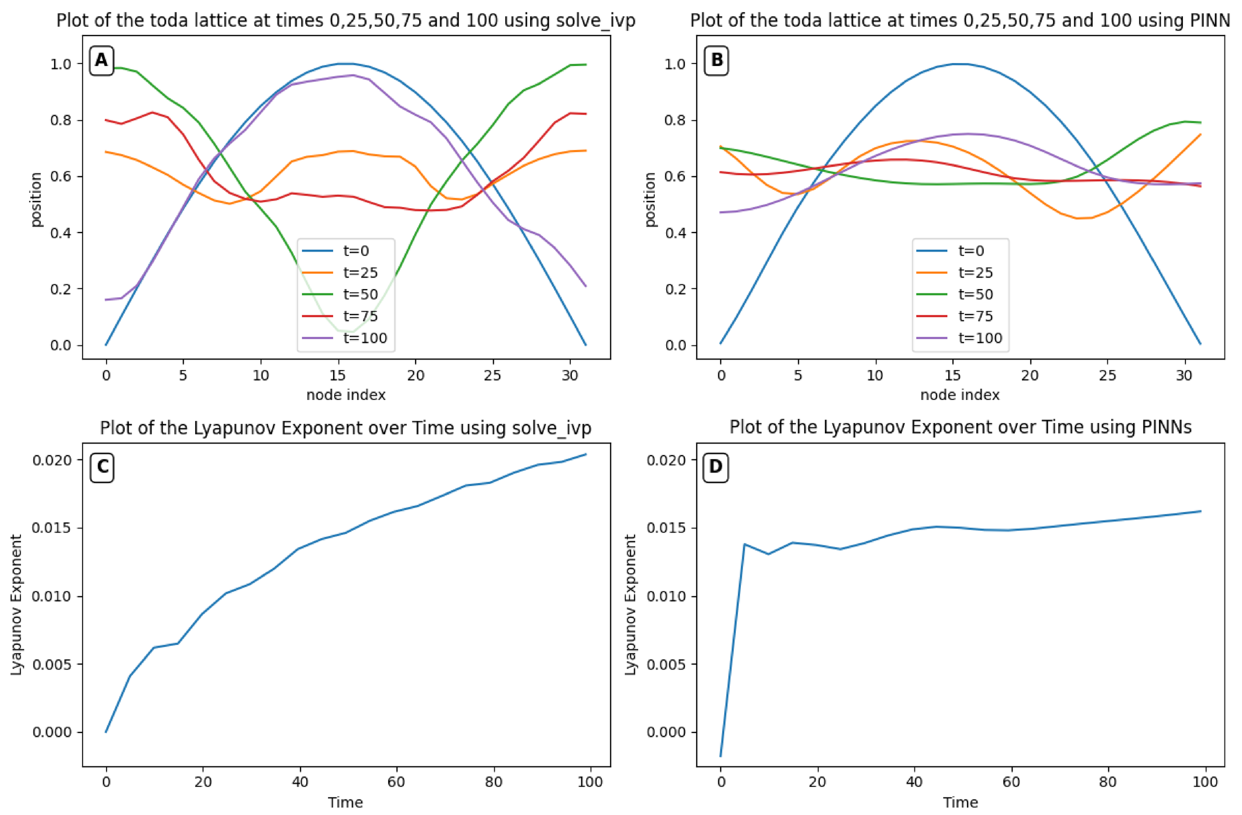

Figure 1: A: Plot of toda lattice trajectory using solve_ivp; B: Plot of toda lattice trajectory using PINNS; C: Plot of the Lyapunov exponent derived from solve_ivp over time; D: Plot of the Lyapunov exponent derived from the PINN over time.

Review

In Fig 1., it is visible from C that over time, the system generated by Python's solve_ivp method, which uses the Runge-Katta method, slowly increases in chaos over time. Comparing the final values of the Lyapunov exponent in C and D, the PINN achieves a value closer to 0. Comparing A and B, the trajectory of the PINN appears to be slightly incorrect however, despite achieving losses of 10^{-2} magnitude during training. During further testing, increasing timespan led the PINN to converge to a flat line.

Conclusion

In this poster we explored the error introduced by discrete time simulations of the Toda Lattice. Through the application of PINNs, we obtained an improvement on the Lyapunov exponent's closeness to 0.

Future work could explore the implementation of a physics informed neural operator, neural networks that act as mappings between function spaces, as general solvers for the Toda lattice given any set of initial conditions. It would also be worth exploring other integrable hamiltonian systems of 1D chains like the Calogero–Moser system, and comparing the performance of our methods against symplectic solvers such as $SABA_2$.

References

[1] Elsa Cardoso-Bihlo and Alex Bihlo. Exactly conservative physics-informed neural networks and deep operator networks for dynamical systems. Neural Networks, 181:106826, 2025.

[2] Carlo Danieli, Emil A. Yuzbashyan, Boris L. Altshuler, Aniket Patra, and Sergej Flach. Dynamical chaos in the integrable toda chain induced by time discretization. Chaos: An Interdisciplinary Journal of Nonlinear Science, 34(3), March 2024.

[3] Pierre Germain. Imperial college lecture notes: The fput equation: a 60 year old mystery, 2025.

[4] Marco Sandri. Numerical calculation of lyapunov exponents, 1996.

[5] Gerald Teschl. Almost everything you always wanted to know about the toda equation. Jahresbericht der Deutschen Mathematiker-Vereinigung, 103(4):149–162, 2001.