From ODEs to Neural ODEs

Yiqiu Liu [1,2] William De Deyn [1,3]

[1]Institut für Geometrie und Praktische Mathematik, RWTH Aachen University, Germany
[2]Data Science and Analytics, HKUST(GZ), China
[3]Department of Computer Science, KU Leuven, Belgium

11.02.2026

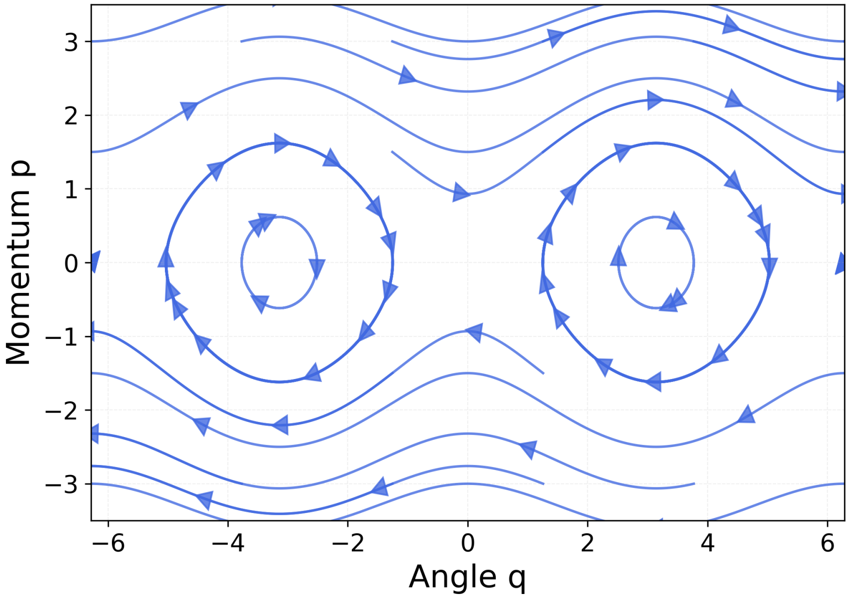
Intro to Myself

- Exchange undergraduate student from HKUST (Guangzhou), Data Science and Analytics
- Research interests: time series forecasting, spatio-temporal analysis, foundation models

Dynamical System

Definition

- **State Space**:
  $\mathcal{X} = \{x = (x_1, \ldots, x_n) \in \mathbb{R}^n\}$
- **Evolution Law (Vector Field)**:
  $\dot{x} = f(t, x)$
- **Initial Condition**:
  $x(0) = x_0 \in \mathcal{X}$
- **Solution (Trajectory)**:
  $x(t; x_0)$ for $t \in [0, T]$
- **Flow Map**:
  $\varphi_t(x_0) := x(t; x_0)$



Phase space of $H(q, p) = \frac{1}{2}p^2 + \cos q$

Dynamical System

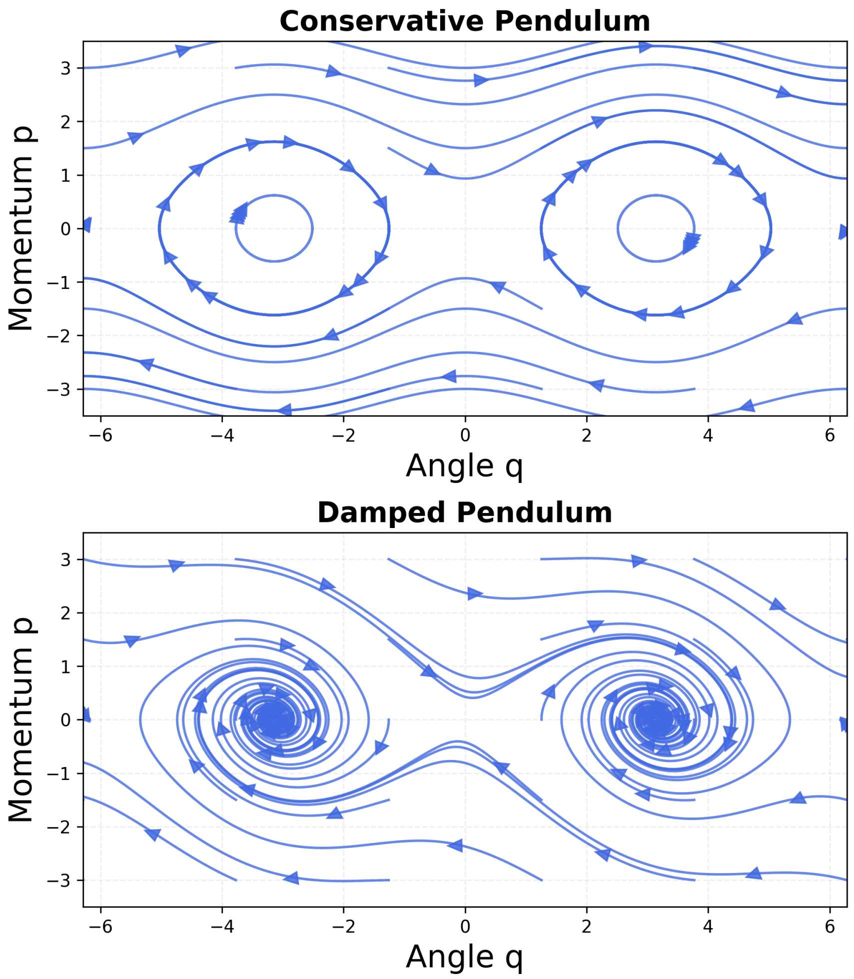
Hamiltonian System $H(q, p)$

- **Formulation:**
  - q stands for the generalized positions, and p is its corresponding momentum:

$$\dot{q} = \frac{\partial H}{\partial p}, \quad \dot{p} = -\frac{\partial H}{\partial q}.$$

- **Properties**
  - Conservation:
    $H(q, p)$ is conserved over time:

$$\frac{d}{dt} H(q, p) = 0$$



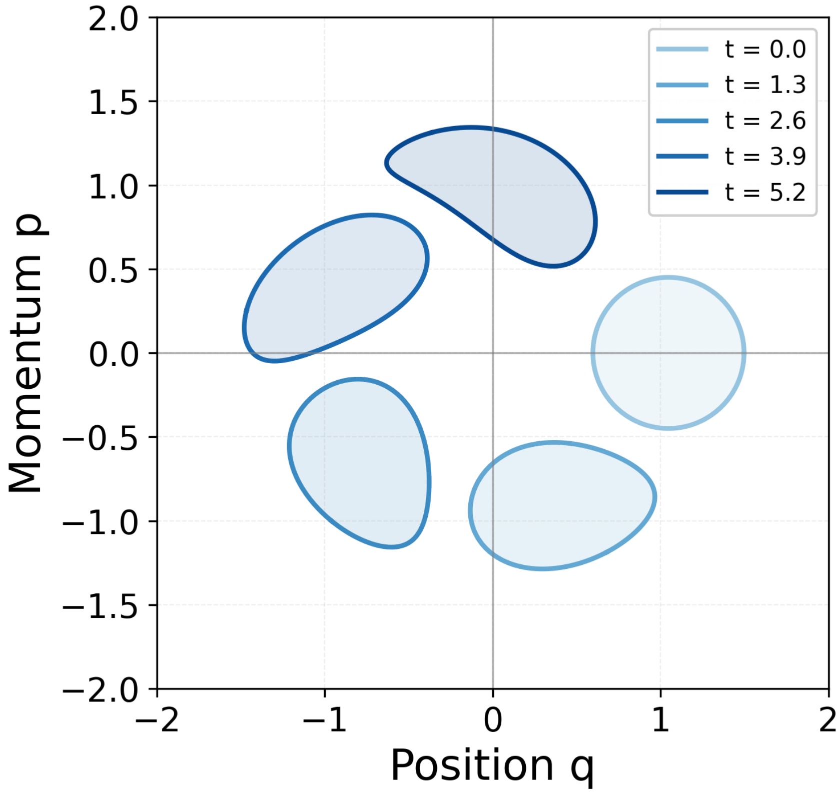
Dynamical System

Hamiltonian System $H(q, p)$

- **Formulation:**
  - q stands for the generalized positions, and p is its corresponding momentum:

$$\dot{q} = \frac{\partial H}{\partial p}, \quad \dot{p} = -\frac{\partial H}{\partial q}.$$

- **Properties**
  - Symplectic Structure Preservation



ODE Types and Numerical Methods

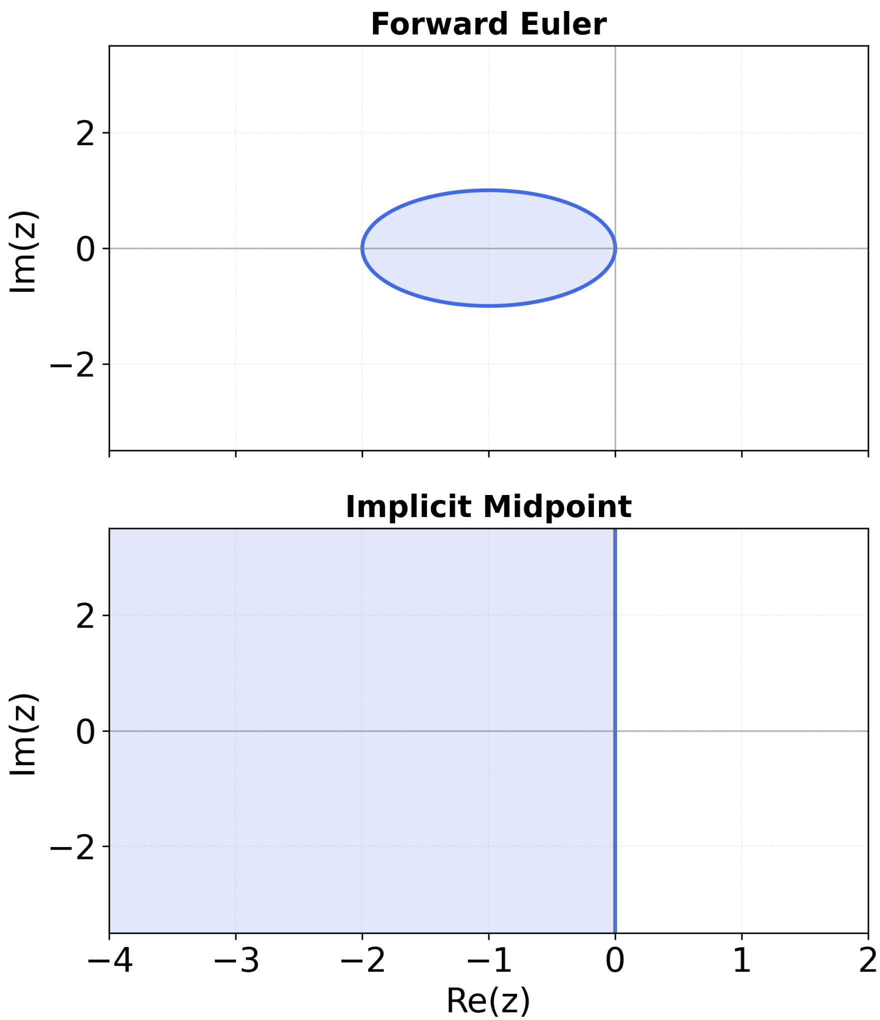
ODE Characteristics

- **Stiffness**: Widely separated time scales
  $\Rightarrow$ Requires A-stable methods
- **Differential Order**: Highest derivative in equation
  $\Rightarrow$ Reduce high-order ODEs to 1st-order systems
- **Geometric Structure**: Special mathematical properties
  $\Rightarrow$ Use geometric integrators

ODE Types and Numerical Methods

Choosing a Numerical Method

- **A-Stability**: $\{z = \lambda h : \mathrm{Re}(z) < 0\} \subseteq$ stability region with h the step size. Handles stiff problems well
- **Order** p: Local error $\sim O(h^{p+1})$
- **Symplecticity**: Conserves geometric structure



Stability Regions

Experiments

- Implement some numerical methods for two ODE systems
- Aim: To verify the properties of the numerical methods and ODEs
- Verified Numerical Methods:

Method	Order	Structure-preserving	A-Stable
Explicit Euler	1	No	No
Implicit Midpoint Rule	2	Yes	Yes
Classical Runge–Kutta Method (RK4)	4	No	No
3-stage Gauss–Legendre RK	6	Yes	Yes

ODE Types and Numerical Methods

Case Study 1: Coffee Cooling (Non-stiff ODE)

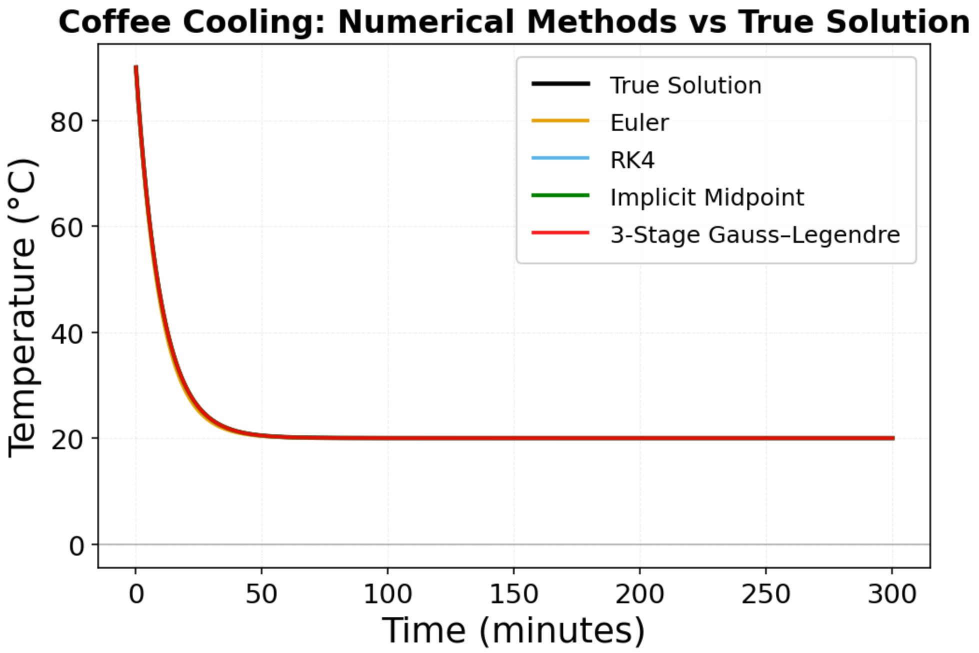
$$\frac{dT}{dt} = -k(T - T_{\text{room}})$$

Properties:

- Linear, autonomous, non-stiff
- True solution available

Result:

- All methods converge for $h = 1$



ODE Types and Numerical Methods

Case Study 1: Coffee Cooling (Non-stiff ODE)

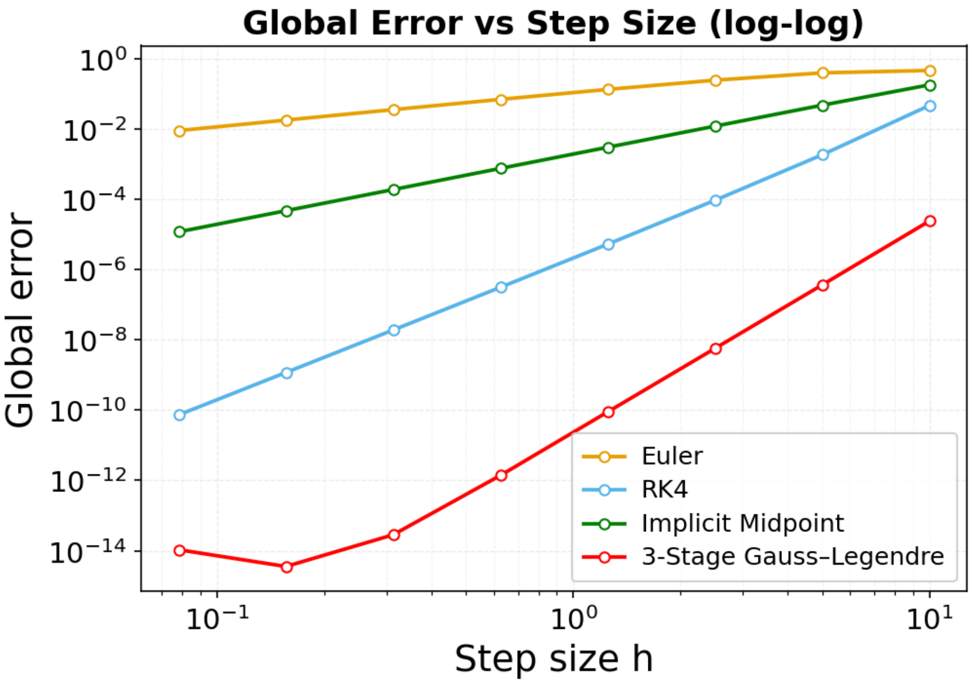
$$\frac{dT}{dt} = -k(T - T_{\text{room}})$$

Properties:

- Linear, autonomous, non-stiff
- True solution available

Result:

- Observed orders match theory



Case Study 2: Duffing Oscillator

$$\ddot{x} + \gamma\dot{x} + \alpha x + \beta x^3 = \delta\cos(\omega t)$$

Hamiltonian (when $\gamma = \delta = 0$):

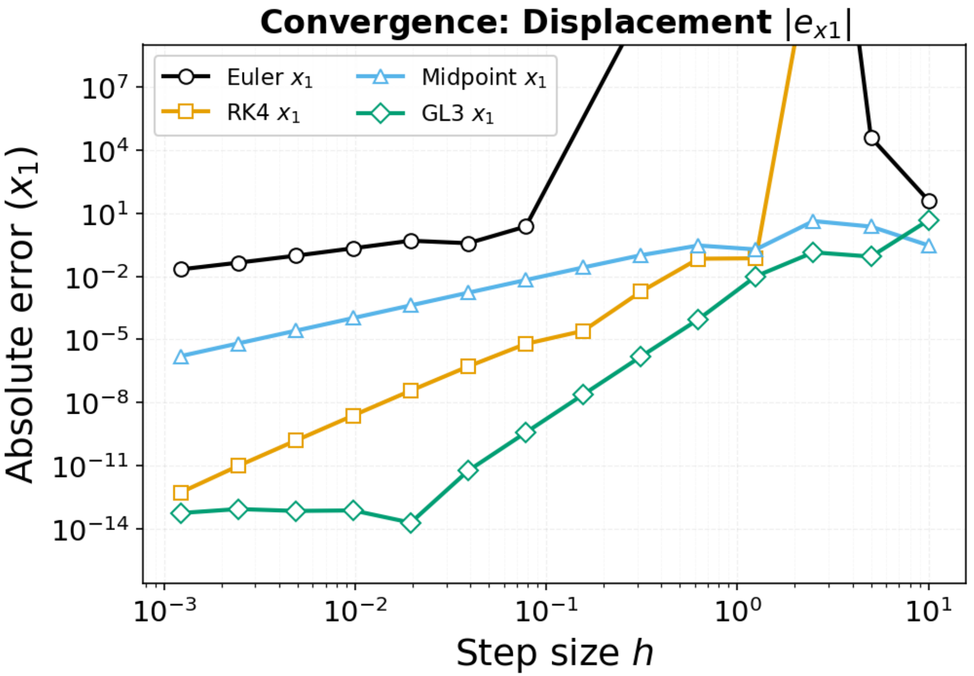
$$H(q,p) = \tfrac{1}{2}p^2 + \tfrac{1}{2}\alpha q^2 + \tfrac{1}{4}\beta q^4$$

Properties:

- Non-stiff under our setting, linear
- Symplectic, non-autonomous

Result:

- Only A-stable methods remain stable as h increases.
- Before instability, the observed orders match theory.



ODE Types and Numerical Methods

Case Study 2: Duffing Oscillator

$$\ddot{x} + \gamma\dot{x} + \alpha x + \beta x^3 = \delta\cos(\omega t)$$

Hamiltonian (when $\gamma = \delta = 0$):

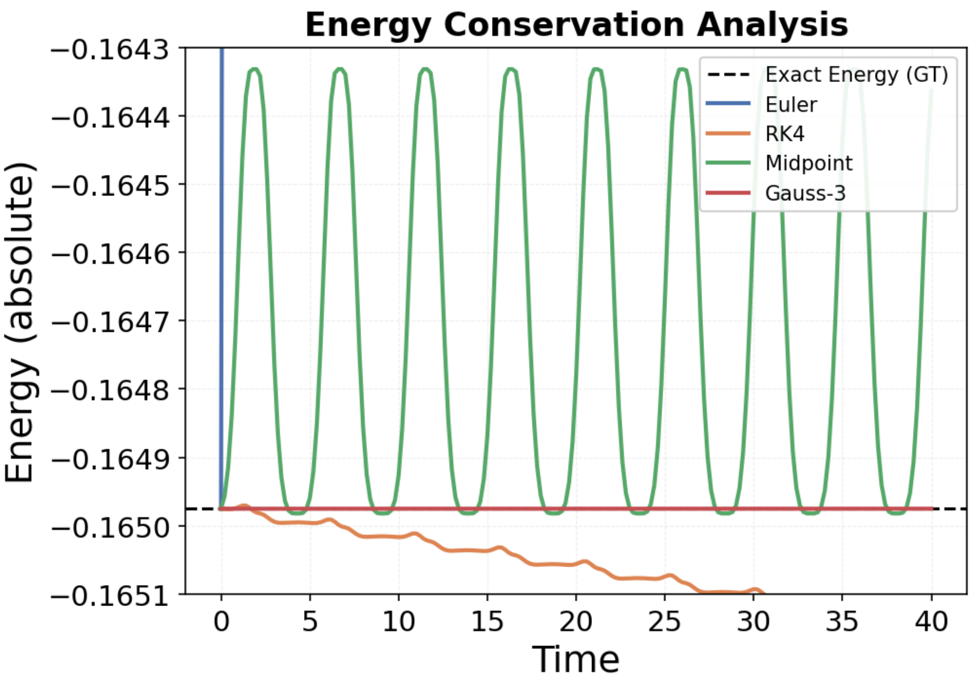
$$H(q,p) = \frac{1}{2}p^2 + \frac{1}{2}\alpha q^2 + \frac{1}{4}\beta q^4$$

Properties:

- Non-stiff under our setting, linear
- Symplectic, non-autonomous

Result:

- The energy is bounded only for Symplectic methods.



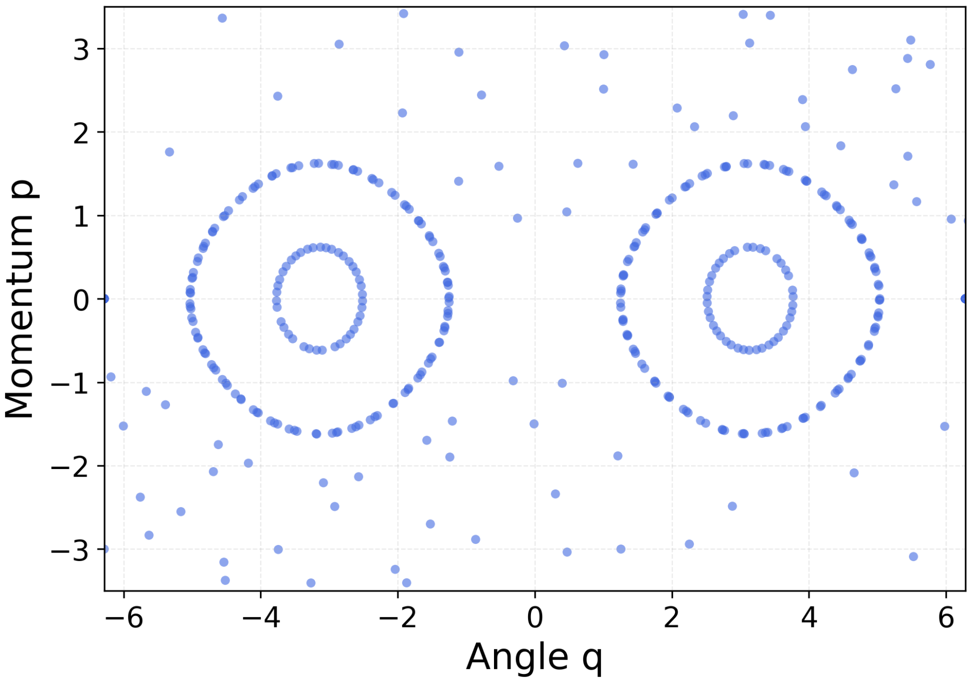
Neural ODEs

When the vector field is unknown

- $f(t, x)$ is not given.
- Only have observed samples of trajectories:

$$\{(t_i, x_i)\}$$

Numerical Methods do not apply.



Neural ODEs

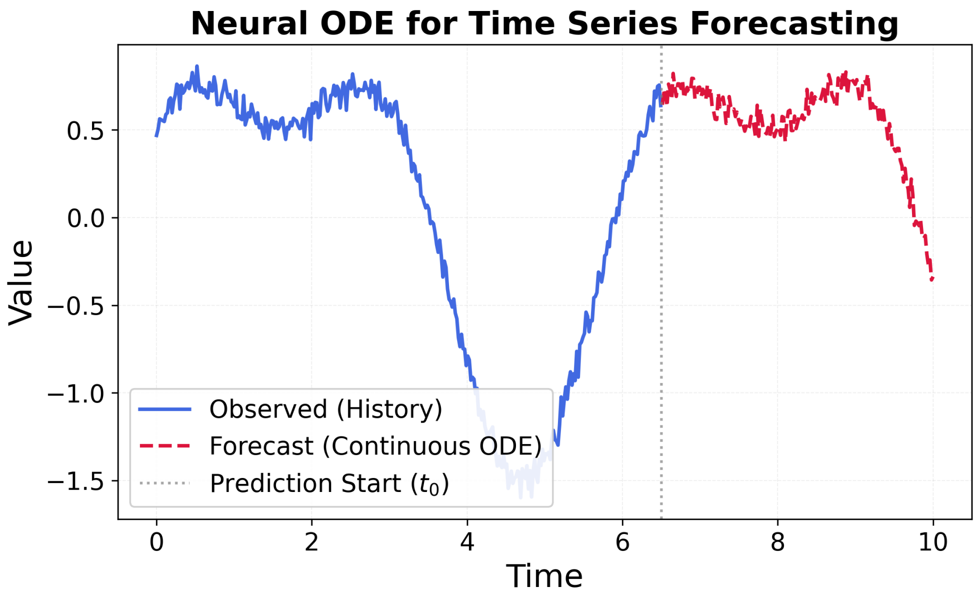
When the vector field is unknown

- $f(t, x)$ is not given.
- Only have observed samples of trajectories:

$$\{(t_i, x_i)\}$$

Numerical Methods do not apply.

Can we learn the underlying vector field from data?



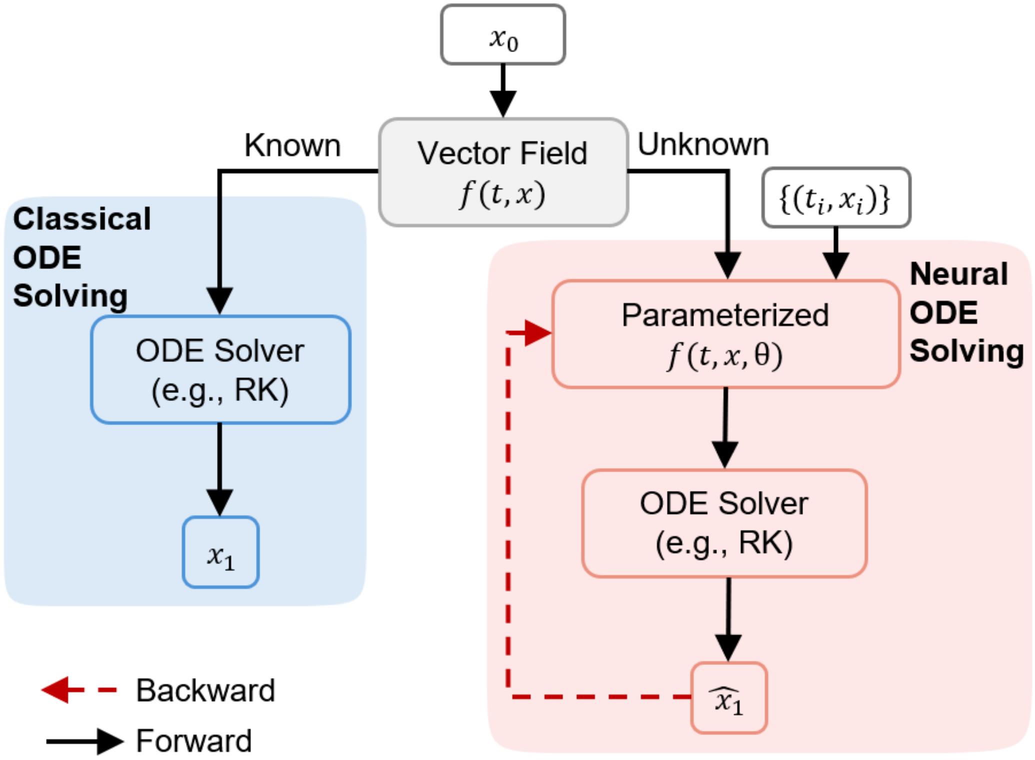
Neural ODEs

Classical ODE solving:

- Solve with known vector field

Neural ODE solving:

- Approximate f with a neural network (e.g. MLP)
- Make the f **learnable**
- Then solve it



Comparison of Classical ODE and Neural ODE

Hamiltonian Neural Network

Learnable Hamiltonian: $(q,p) \xrightarrow{\text{NN}} \hat{H}_\theta(q,p) \xrightarrow{\nabla} \nabla \hat{H}_\theta(q,p)$, where

$$\dot{q} = \frac{\partial \hat{H}_\theta}{\partial p}, \qquad \dot{p} = -\frac{\partial \hat{H}_\theta}{\partial q} \quad \Rightarrow \quad \dot{x} = f_\theta(x) = J \nabla \hat{H}_\theta(x),$$

$$\text{with } x = \begin{pmatrix} q \\ p \end{pmatrix} \text{ and } J = \begin{pmatrix} 0 & I_d \\ -I_d & 0 \end{pmatrix}.$$

Symplectic-preserving: Use the **implicit midpoint** as a constraint:

$$x_{n+1} = x_n + h\, f_\theta\Big(\tfrac{x_n + x_{n+1}}{2}\Big).$$

Train by minimizing its residual:

$$\mathcal{L}_{\text{mid}}(\theta) = \left\| \frac{x_{n+1} - x_n}{h} - J\, \nabla \hat{H}_\theta\Big(\tfrac{x_n + x_{n+1}}{2}\Big) \right\|^2.$$

Changing the order of the loss function changes the model order.

Hamiltonian Neural Network

Experiment

- Aim: examine the impact of model order and solver order.
- **Hamiltonian under study**:

$$H(q,p) = \frac{1}{2}p^2 - \cos q$$

  - Pendulum-type system; no closed-form analytic solution in elementary functions.
  - Use a 6-order symplectic integrator to produce a near-analytic ground truth to eliminate influence from the ground truth solver order.
- **Error of the network:**

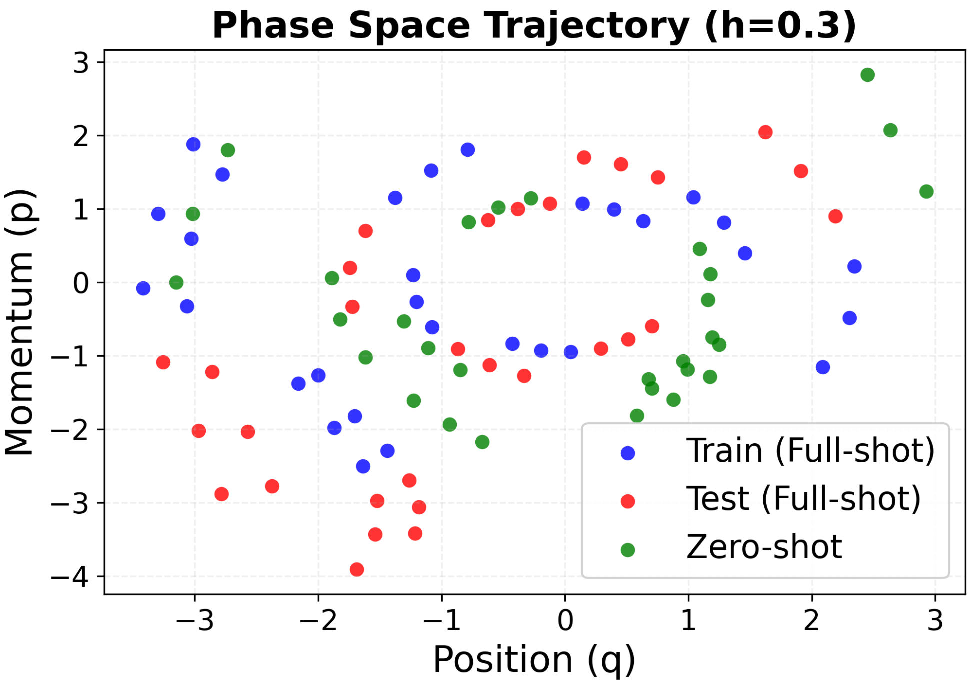
$$\text{Total Error} = \underbrace{\textbf{Model Approximation Error}}_{\text{learned } f_\theta} + \underbrace{\text{ODE Solver Error}}_{\text{integrator}}.$$

Hamiltonian Neural Network

- **Setting**

Part	**Order**	**Step Size**
GT	6th	h
Model	k-th	h
Solver	m-th	h

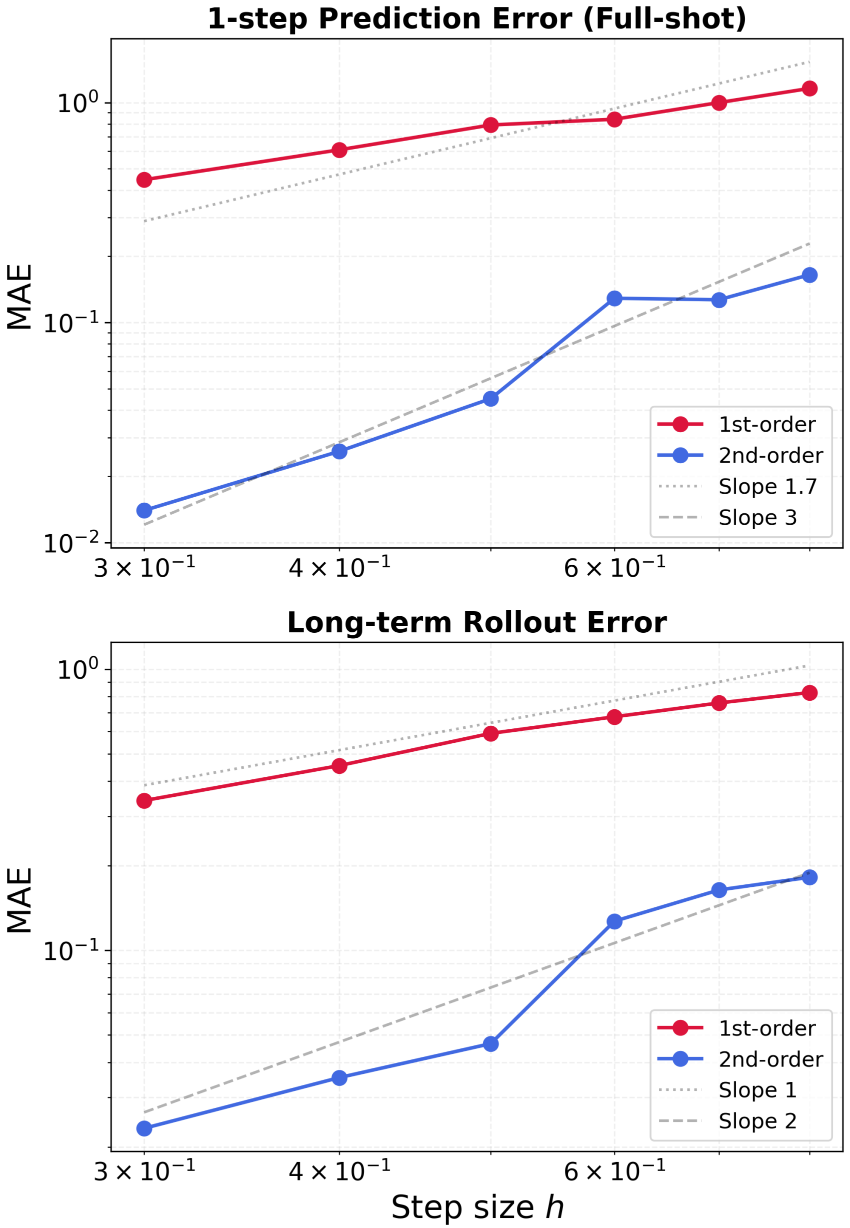
  - $m = k = \{1, 2\}$
  - $h \in \{0.8, 0.7, 0.6, 0.5, 0.4, 0.3\}$.
  - Each experiment is repeated 5 times



Hamiltonian Neural Network

Result

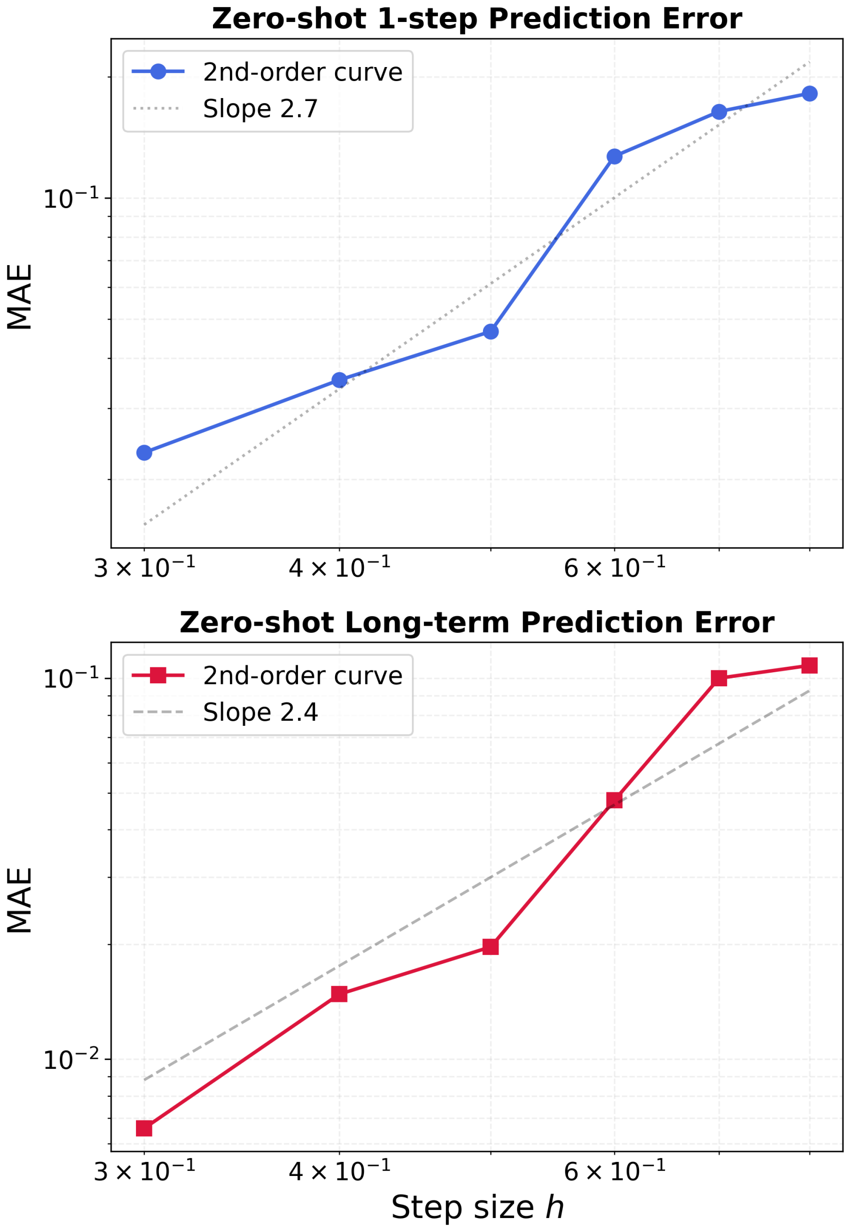
- Full-shot result
  - The truncation error matchs the theory $O(h^{k+1})$.
  - The global error matchs the theory $O(h^k)$.



Hamiltonian Neural Network

Result

- Full-shot result
  - The truncation error matchs the theory $O(h^{k+1})$.
  - The global error matchs the theory $O(h^k)$.
- Zero-shot result
  - The model generalizes well on unseen trajectory
  - The learnt vector field captures the true dynamics.



Hamiltonian Neural Network

Result

- Conservation analysis
  - Phase-space volume stays unchanged.

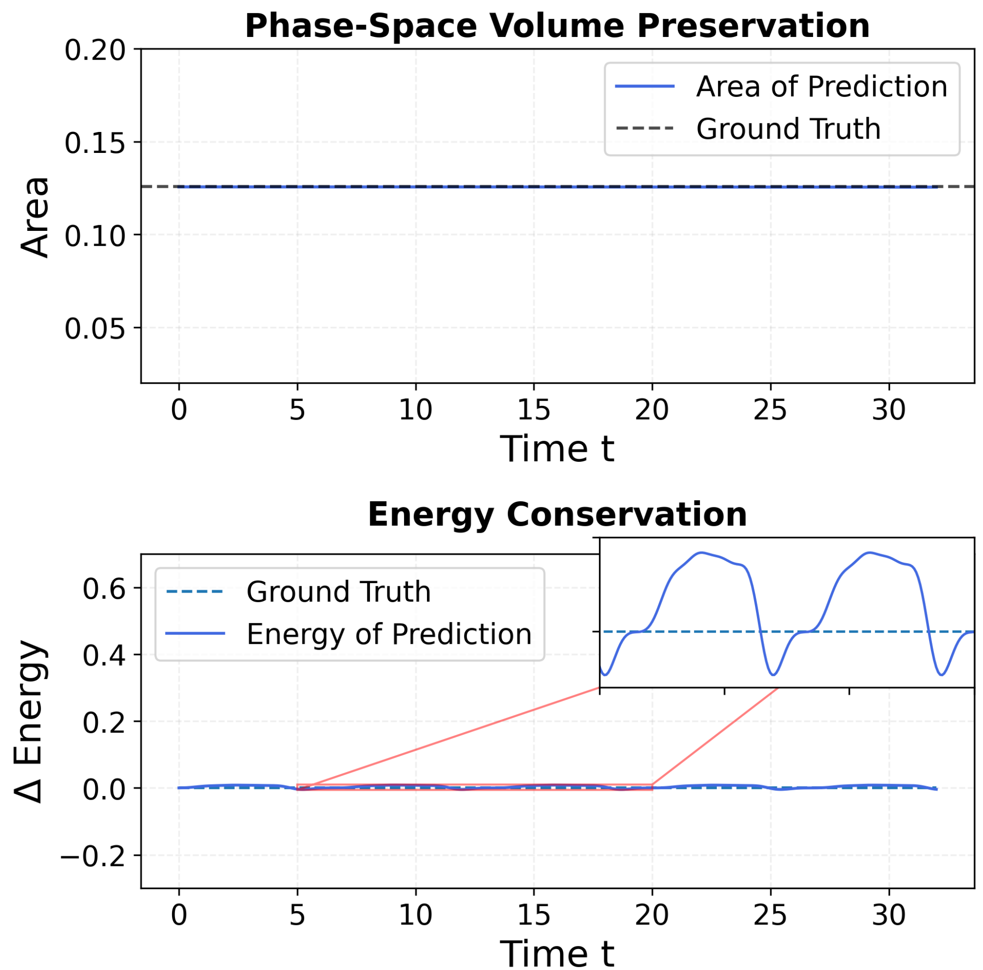
    Compute the area formed by 100 points at each time:

$$A = \frac{1}{2}\left|\sum_{i=1}^{N}\left(x_i y_{i+1} - y_i x_{i+1}\right)\right|.$$

  - Energy is bounded.

    Compute via the Hamiltonian:

$$H(q,p) = \frac{1}{2}p^2 - \cos q.$$



Conclusions

- Reviewed key properties of ODEs and numerical methods.
- Demonstrated these properties through experiments on two representative ODE systems.
- Introduced Neural ODEs as a data-driven extension of classical dynamical systems.
- Focused on Hamiltonian Neural Networks: how to model the Hamiltonian and enforce symplectic structure.
- Analyzed Hamiltonian Neural Networks performance through convergence tests and structure-preservation experiments.