Structure-Preserving Solutions of Hamiltonian Systems Based on Neural Networks

Davide Murari

Department of Applied Mathematics and Theoretical Physics, University of Cambridge.

dm2011@cam.ac.uk

SIAM Conference on Mathematics of Data Science, 21 October 2024

In collaboration with Priscilla Canizares, Carola-Bibiane Schönlieb, Ferdia Sherry, and Zakhar Shumaylov

Solving initial value problems with neural networks

- We aim to solve the autonomous initial value problem (IVP)

$$\begin{cases} \dot{\mathbf{x}}(t) = \mathcal{F}(\mathbf{x}(t)) \in \mathbb{R}^d, \\ \mathbf{x}(0) = \mathbf{x}_0 \in \mathbb{R}^d \end{cases}$$

  on the time interval $[0, T]$.

Solving initial value problems with neural networks

- We aim to solve the autonomous initial value problem (IVP)

$$\begin{cases} \dot{\mathbf{x}}(t) = \mathcal{F}(\mathbf{x}(t)) \in \mathbb{R}^d, \\ \mathbf{x}(0) = \mathbf{x}_0 \in \mathbb{R}^d \end{cases}$$

  on the time interval $[0, T]$.

- Using **neural networks** to approximate $\mathbf{x}(t)$ can be useful when
  - the dimension d is large,
  - one desires to have a (piecewise) continuous approximate solution,
  - one wants to also fit some observed data while approximately solving the IVP.

Forward invariant sets: Flow map approach

- Suppose $\mathbf{x}(t) \in \Omega \subset \mathbb{R}^d$, whenever $\mathbf{x}(0) \in \Omega$, for any $t \geq 0$.

[1] Sifan Wang and Paris Perdikaris. "Long-time integration of parametric evolution equations with physics-informed DeepONets". In: *Journal of Computational Physics* 475 (2023), p. 111855.

Forward invariant sets: Flow map approach

- Suppose $\mathbf{x}(t) \in \Omega \subset \mathbb{R}^d$, whenever $\mathbf{x}(0) \in \Omega$, for any $t \geq 0$.
- We can then work with $\mathcal{N}_\theta : [0, \Delta t] \times \Omega \to \mathbb{R}^d$, where[1]

$$\mathcal{L}(\theta) := \frac{1}{N_r} \sum_{i=1}^{N_r} \left\| \frac{d}{dt} \mathcal{N}_\theta \left(t, \mathbf{x}_0^i\right) \Big|_{t=t_i} - \mathcal{F}\left(\mathcal{N}_\theta\left(t_i, \mathbf{x}_0^i\right)\right) \right\|_2^2 \to \min .$$

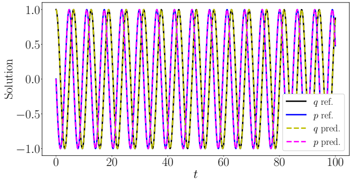

Figure 1: Network trained with $\Delta t = 1$ and applied up to $T = 100$.

[1] Wang and Perdikaris, "Long-time integration of parametric evolution equations with physics-informed DeepONets".

Canonical Hamiltonian equations

- The equations of motion of canonical Hamiltonian systems write

$$\dot{\mathbf{x}} = \mathbb{J}\nabla H(\mathbf{x}) = X_H(\mathbf{x}) \in \mathbb{R}^{2n}, \quad \mathbb{J} = \begin{bmatrix} 0 & I_n \\ -I_n & 0_n \end{bmatrix} \in \mathbb{R}^{2n\times 2n}. \tag{1}$$

Canonical Hamiltonian equations

- The equations of motion of canonical Hamiltonian systems write

$$\dot{\mathbf{x}} = \mathbb{J}\nabla H(\mathbf{x}) = X_H(\mathbf{x}) \in \mathbb{R}^{2n}, \quad \mathbb{J} = \begin{bmatrix} 0 & I_n \\ -I_n & 0_n \end{bmatrix} \in \mathbb{R}^{2n\times 2n}. \tag{1}$$

- Denoted with $\phi_{H,t} : \mathbb{R}^{2n} \to \mathbb{R}^{2n}$ the exact flow of (1), we have that
  - $$\frac{d}{dt} H(\phi_{H,t}(\mathbf{x}_0)) = \nabla H(\phi_{H,t}(\mathbf{x}_0))^\top \mathbb{J}\nabla H(\phi_{H,t}(\mathbf{x}_0)) = 0,$$

Canonical Hamiltonian equations

- The equations of motion of canonical Hamiltonian systems write

$$\dot{\mathbf{x}} = \mathbb{J}\nabla H(\mathbf{x}) = X_H(\mathbf{x}) \in \mathbb{R}^{2n}, \quad \mathbb{J} = \begin{bmatrix} 0 & I_n \\ -I_n & 0_n \end{bmatrix} \in \mathbb{R}^{2n\times 2n}. \tag{1}$$

- Denoted with $\phi_{H,t} : \mathbb{R}^{2n} \to \mathbb{R}^{2n}$ the exact flow of (1), we have that
  - $$\frac{d}{dt} H(\phi_{H,t}(\mathbf{x}_0)) = \nabla H(\phi_{H,t}(\mathbf{x}_0))^\top \mathbb{J}\nabla H(\phi_{H,t}(\mathbf{x}_0)) = 0,$$
  - $$\left(\frac{\partial \phi_{H,t}(\mathbf{x}_0)}{\partial \mathbf{x}_0}\right)^\top \mathbb{J} \left(\frac{\partial \phi_{H,t}(\mathbf{x}_0)}{\partial \mathbf{x}_0}\right) = \mathbb{J},$$

Canonical Hamiltonian equations

- The equations of motion of canonical Hamiltonian systems write

$$\dot{\mathbf{x}} = \mathbb{J}\nabla H(\mathbf{x}) = X_H(\mathbf{x}) \in \mathbb{R}^{2n}, \quad \mathbb{J} = \begin{bmatrix} 0 & I_n \\ -I_n & 0_n \end{bmatrix} \in \mathbb{R}^{2n\times 2n}. \tag{1}$$

- Denoted with $\phi_{H,t} : \mathbb{R}^{2n} \to \mathbb{R}^{2n}$ the exact flow of (1), we have that
  - $$\frac{d}{dt} H(\phi_{H,t}(\mathbf{x}_0)) = \nabla H(\phi_{H,t}(\mathbf{x}_0))^\top \mathbb{J}\nabla H(\phi_{H,t}(\mathbf{x}_0)) = 0,$$
  - $$\left(\frac{\partial \phi_{H,t}(\mathbf{x}_0)}{\partial \mathbf{x}_0}\right)^\top \mathbb{J} \left(\frac{\partial \phi_{H,t}(\mathbf{x}_0)}{\partial \mathbf{x}_0}\right) = \mathbb{J},$$
  - the flow preserves the canonical volume form of $\mathbb{R}^{2n}$.

The SympFlow

- We now build a neural network that approximates $\phi_{H,t} : \Omega \to \mathbb{R}^{2n}$ for a forward invariant set $\Omega \subset \mathbb{R}^{2n}$, and $t \in [0, \Delta t]$, while reproducing the qualitative properties of $\phi_{H,t}$.

The SympFlow

- We now build a neural network that approximates $\phi_{H,t} : \Omega \to \mathbb{R}^{2n}$ for a forward invariant set $\Omega \subset \mathbb{R}^{2n}$, and $t \in [0, \Delta t]$, while reproducing the qualitative properties of $\phi_{H,t}$.
- We rely on two building blocks, which applied to $(\mathbf{q}_0, \mathbf{p}_0) \in \mathbb{R}^{2n}$ write:

$$\phi_{\mathbf{p},t}\left((\mathbf{q}_0, \mathbf{p}_0)\right) = \begin{bmatrix} \mathbf{q}_0 \\ \mathbf{p}_0 - \left(\nabla_{\mathbf{q}} V\left(t, \mathbf{q}_0\right) - \nabla_{\mathbf{q}} V\left(0, \mathbf{q}_0\right)\right) \end{bmatrix},$$

$$\phi_{\mathbf{q},t}\left((\mathbf{q}_0, \mathbf{p}_0)\right) = \begin{bmatrix} \mathbf{q}_0 + \left(\nabla_{\mathbf{p}} K\left(t, \mathbf{p}_0\right) - \nabla_{\mathbf{p}} K\left(0, \mathbf{p}_0\right)\right) \\ \mathbf{p}_0 \end{bmatrix}.$$

The SympFlow

- We now build a neural network that approximates $\phi_{H,t} : \Omega \to \mathbb{R}^{2n}$ for a forward invariant set $\Omega \subset \mathbb{R}^{2n}$, and $t \in [0, \Delta t]$, while reproducing the qualitative properties of $\phi_{H,t}$.
- We rely on two building blocks, which applied to $(\mathbf{q}_0, \mathbf{p}_0) \in \mathbb{R}^{2n}$ write:

$$\phi_{\mathbf{p},t}\left((\mathbf{q}_0, \mathbf{p}_0)\right) = \begin{bmatrix} \mathbf{q}_0 \\ \mathbf{p}_0 - \left(\nabla_{\mathbf{q}} V\left(t, \mathbf{q}_0\right) - \nabla_{\mathbf{q}} V\left(0, \mathbf{q}_0\right)\right) \end{bmatrix},$$

$$\phi_{\mathbf{q},t}\left((\mathbf{q}_0, \mathbf{p}_0)\right) = \begin{bmatrix} \mathbf{q}_0 + \left(\nabla_{\mathbf{p}} K\left(t, \mathbf{p}_0\right) - \nabla_{\mathbf{p}} K\left(0, \mathbf{p}_0\right)\right) \\ \mathbf{p}_0 \end{bmatrix}.$$

- The SympFlow architecture is defined as

$$\mathcal{N}_\theta\left(t, (\mathbf{q}_0, \mathbf{p}_0)\right) = \phi^L_{\mathbf{p},t} \circ \phi^L_{\mathbf{q},t} \circ \cdots \circ \phi^1_{\mathbf{p},t} \circ \phi^1_{\mathbf{q},t}\left((\mathbf{q}_0, \mathbf{p}_0)\right).$$

Properties of the SympFlow

- The SympFlow is symplectic for every time $t \in \mathbb{R}$. The building blocks we compose are exact flows of time-dependent Hamiltonian systems:

$$\phi^i_{\mathbf{p},t}\left((\mathbf{q},\mathbf{p})\right) = \begin{bmatrix} \mathbf{q} \\ \mathbf{p} - \left(\nabla_{\mathbf{q}} V^i\left(t,\mathbf{q}\right) - \nabla_{\mathbf{q}} V^i\left(0,\mathbf{q}\right)\right) \end{bmatrix}$$

$$= \begin{bmatrix} \mathbf{q} \\ \mathbf{p} - \int_0^t \nabla_{\mathbf{q}} \partial_s V^i\left(s,\mathbf{q}\right) ds \end{bmatrix} = \phi_{\widetilde{V}^i,t}\left((\mathbf{q},\mathbf{p})\right),$$

with $\widetilde{V}^i(t,(\mathbf{q},\mathbf{p})) = \partial_t V^i(t,\mathbf{q})$.

Properties of the SympFlow

- The SympFlow is symplectic for every time $t \in \mathbb{R}$. The building blocks we compose are exact flows of time-dependent Hamiltonian systems:

$$\phi^i_{\mathbf{p},t}\left((\mathbf{q},\mathbf{p})\right) = \begin{bmatrix} \mathbf{q} \\ \mathbf{p} - \left(\nabla_{\mathbf{q}} V^i\left(t,\mathbf{q}\right) - \nabla_{\mathbf{q}} V^i\left(0,\mathbf{q}\right)\right) \end{bmatrix}$$
$$= \begin{bmatrix} \mathbf{q} \\ \mathbf{p} - \int_0^t \nabla_{\mathbf{q}} \partial_s V^i\left(s,\mathbf{q}\right) ds \end{bmatrix} = \phi_{\widetilde{V}^i,t}\left((\mathbf{q},\mathbf{p})\right),$$

  with $\widetilde{V}^i(t,(\mathbf{q},\mathbf{p})) = \partial_t V^i(t,\mathbf{q})$.

- The SympFlow is volume preserving.

Properties of the SympFlow

- The SympFlow is symplectic for every time $t \in \mathbb{R}$. The building blocks we compose are exact flows of time-dependent Hamiltonian systems:

$$\phi^i_{\mathbf{p},t}((\mathbf{q},\mathbf{p})) = \begin{bmatrix} \mathbf{q} \\ \mathbf{p} - \left(\nabla_{\mathbf{q}} V^i(t,\mathbf{q}) - \nabla_{\mathbf{q}} V^i(0,\mathbf{q})\right) \end{bmatrix}$$

$$= \begin{bmatrix} \mathbf{q} \\ \mathbf{p} - \int_0^t \nabla_{\mathbf{q}} \partial_s V^i(s,\mathbf{q})\, ds \end{bmatrix} = \phi_{\widetilde{V}^i,t}((\mathbf{q},\mathbf{p})),$$

  with $\widetilde{V}^i(t,(\mathbf{q},\mathbf{p})) = \partial_t V^i(t,\mathbf{q})$.

- The SympFlow is volume preserving.

- The SympFlow is the exact flow of a time-dependent Hamiltonian system.

Composition of Hamiltonian flows[2]

Theorem (The Hamiltonian flows are closed under composition)

Let $H^1, H^2 : \mathbb{R} \times \mathbb{R}^{2n} \to \mathbb{R}$ be continuously differentiable functions. Then, the map $\phi_{H^2,t} \circ \phi_{H^1,t} : \mathbb{R}^{2n} \to \mathbb{R}^{2n}$ is the exact flow of the time-dependent Hamiltonian system defined by the Hamiltonian function

$$H^3(t,x) = H^2(t,x) + H^1\left(t, \phi^{-1}_{H^2,t}(x)\right).$$

[2] Leonid Polterovich. *The Geometry of the Group of Symplectic Diffeomorphisms*. Lectures in Mathematics ETH Zürich. Basel: Springer Basel AG, 2001. ISBN: 978-3-7643-6432-8.

Composition of Hamiltonian flows[2]

Theorem (The Hamiltonian flows are closed under composition)

Let $H^1, H^2 : \mathbb{R} \times \mathbb{R}^{2n} \to \mathbb{R}$ be continuously differentiable functions. Then, the map $\phi_{H^2,t} \circ \phi_{H^1,t} : \mathbb{R}^{2n} \to \mathbb{R}^{2n}$ is the exact flow of the time-dependent Hamiltonian system defined by the Hamiltonian function

$$H^3(t,x) = H^2(t,x) + H^1\left(t, \phi^{-1}_{H^2,t}(x)\right).$$

- This Theorem implies that there is a Hamiltonian function $\mathcal{H}(\mathcal{N}_\theta) : \mathbb{R} \times \mathbb{R}^{2n} \to \mathbb{R}^{2n}$ such that

$$\mathcal{N}_\theta(t,\cdot) = \phi_{\mathcal{H}(\mathcal{N}_\theta),t}(\cdot).$$

[2] Polterovich, *The Geometry of the Group of Symplectic Diffeomorphisms*.

Training of the SympFlow

- The SympFlow is based on modelling the scalar-valued potentials $\widetilde{V}^i, \widetilde{K}^i : \mathbb{R} \times \mathbb{R}^n \to \mathbb{R}$ with feed-forward neural networks.
- To train the overall model $\mathcal{N}_\theta$ we minimise the loss function

$$\mathcal{L}(\theta) = \underbrace{\frac{1}{N_r}\sum_{i=1}^{N_r}\left\| \frac{d}{dt}\mathcal{N}_\theta\left(t, \mathbf{x}_0^i\right)\Big|_{t=t_i} - \mathbb{J}\nabla H\left(\mathcal{N}_\theta\left(t_i, \mathbf{x}_0^i\right)\right)\right\|_2^2}_{\text{Residual term}}$$
$$+ \underbrace{\frac{1}{N_m}\sum_{j=1}^{N_m}\left(\mathcal{H}\left(\mathcal{N}_\theta\right)\left(t_j, \mathbf{x}^j\right) - H\left(\mathbf{x}^j\right)\right)^2}_{\text{Hamiltonian matching}},$$

where we sample $t_i, t_j \in [0, \Delta t]$, and $\mathbf{x}_0^i, \mathbf{x}^i \in \Omega \subset \mathbb{R}^{2n}$.

Extension of the SympFlow outside of $[0, \Delta t]$

- Once we have trained $\mathcal{N}_\theta$ to be reliable for $t \in [0, \Delta t]$, we extend it for longer times as

$$\psi\left(t, \mathbf{x}_0\right) := \bar{\psi}_{t-\Delta t\lfloor t/\Delta t\rfloor} \circ \left(\bar{\psi}_{\Delta t}\right)^{\lfloor t/\Delta t\rfloor}\left(\mathbf{x}_0\right),$$

for $t \in [0, +\infty)$ and $\mathbf{x}_0 \in \Omega \subset \mathbb{R}^{2n}$, where

$$\bar{\psi}_s\left(\mathbf{x}_0\right) := \mathcal{N}_\theta\left(s, \mathbf{x}_0\right), \;\; s \in [0, \Delta t),$$
$$\left(\bar{\psi}_{\Delta t}\right)^k := \underbrace{\bar{\psi}_{\Delta t} \circ \cdots \circ \bar{\psi}_{\Delta t}}_{k \text{ times}}, \;\; k \in \mathbb{N}.$$

Extension of the SympFlow outside of $[0, \Delta t]$

- Once we have trained $\mathcal{N}_\theta$ to be reliable for $t \in [0, \Delta t]$, we extend it for longer times as

$$\psi\left(t, \mathbf{x}_0\right) := \bar{\psi}_{t-\Delta t\lfloor t/\Delta t\rfloor} \circ \left(\bar{\psi}_{\Delta t}\right)^{\lfloor t/\Delta t\rfloor}\left(\mathbf{x}_0\right),$$

  for $t \in [0, +\infty)$ and $\mathbf{x}_0 \in \Omega \subset \mathbb{R}^{2n}$, where

$$\bar{\psi}_s\left(\mathbf{x}_0\right) := \mathcal{N}_\theta\left(s, \mathbf{x}_0\right), \; s \in [0, \Delta t),$$
$$\left(\bar{\psi}_{\Delta t}\right)^k := \underbrace{\bar{\psi}_{\Delta t} \circ \cdots \circ \bar{\psi}_{\Delta t}}_{k \text{ times}}, \; k \in \mathbb{N}.$$

- $\psi(t, \cdot) = \phi_{\widetilde{H},t}$ for the piecewise continuous Hamiltonian

$$\widetilde{H}\left(t, \mathbf{x}\right) := \mathcal{H}\left(\mathcal{N}_\theta\right)\left(t - \Delta t\lfloor t/\Delta t\rfloor, \mathbf{x}\right).$$

Equations of motion

$$\dot{x} = p, \;\; \dot{p} = -x.$$

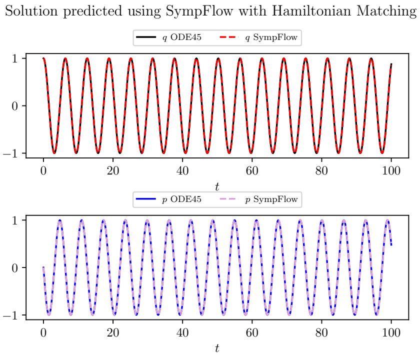

Figure 2: $\mathbf{x}_0 = [1, 0]$.

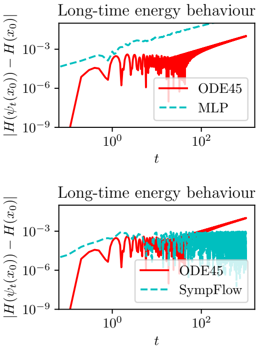

Figure 3: Long time energy behaviour.

Equations of motion

$$\dot{x} = p_x, \ \dot{y} = p_y, \ \dot{p}_x = -x - 2xy, \ \dot{p}_y = -y - (x^2 - y^2).$$

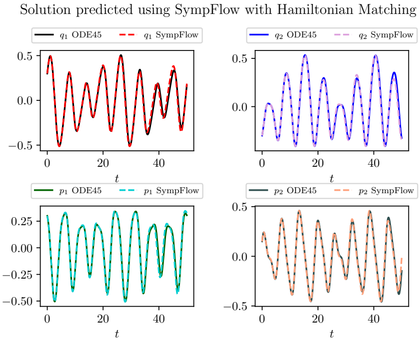

Figure 4: $\mathbf{x}_0 = [0.3, -0.3, 0.3, 0.15]$.

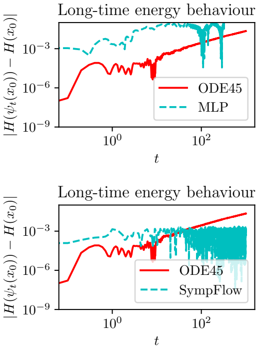

Figure 5: Long time energy behaviour.

THANK YOU FOR THE ATTENTION

dm2011@cam.ac.uk

Physics-informed neural networks

- We introduce a parametric map $\mathcal{N}_\theta(\cdot, \mathbf{x}_0) : [0, T] \to \mathbb{R}^d$ such that $\mathcal{N}_\theta(0, \mathbf{x}_0) = \mathbf{x}_0$, and choose its weights so that

$$\mathcal{L}(\theta) := \frac{1}{C}\sum_{c=1}^{C}\left\| \frac{d}{dt}\mathcal{N}_\theta(t, \mathbf{x}_0)\Big|_{t=t_c} - \mathcal{F}(\mathcal{N}_\theta(t_c, \mathbf{x}_0))\right\|_2^2 \to \min$$

  for some collocation points $t_1, \ldots, t_C \in [0, T]$.

Physics-informed neural networks

- We introduce a parametric map $\mathcal{N}_\theta(\cdot, \mathbf{x}_0) : [0, T] \to \mathbb{R}^d$ such that $\mathcal{N}_\theta(0, \mathbf{x}_0) = \mathbf{x}_0$, and choose its weights so that

$$\mathcal{L}(\theta) := \frac{1}{C} \sum_{c=1}^{C} \left\| \frac{d}{dt} \mathcal{N}_\theta(t, \mathbf{x}_0) \Big|_{t=t_c} - \mathcal{F}(\mathcal{N}_\theta(t_c, \mathbf{x}_0)) \right\|_2^2 \to \min$$

  for some collocation points $t_1, \ldots, t_C \in [0, T]$.

- Then, $t \mapsto \mathcal{N}_\theta(t, \mathbf{x}_0)$ will solve a different IVP

$$\begin{cases} \dot{\mathbf{y}}(t) = \mathcal{F}(\mathbf{y}(t)) + \left(\frac{d}{dt} \mathcal{N}_\theta(t, \mathbf{x}_0) \big|_{t=t} - \mathcal{F}(\mathbf{y}(t)) \right) \in \mathbb{R}^d, \\ \mathbf{y}(0) = \mathbf{x}_0 \in \mathbb{R}^d, \end{cases}$$

  where hopefully the residual $\frac{d}{dt} \mathcal{N}_\theta(t, \mathbf{x}_0) \big|_{t=t} - \mathcal{F}(\mathbf{y}(t))$ is small in some sense.

Training issues with neural network

- Solving a single IVP on $[0, T]$ with a neural network can take long training time.
- The obtained solution can not be used to solve the same ordinary differential equation with a different initial condition.

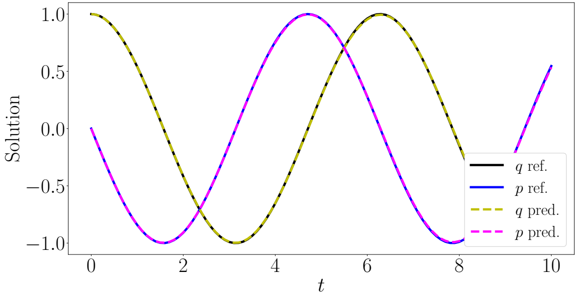

Figure 6: Solution comparison after reaching a loss value of 10^{-5}. The training time is of 87 seconds (7500 epochs with 1000 new collocation points randomly sampled at each of them).

Training issues with neural network

- It is hard to solve initial value problems over long time intervals.