Learning non-canonical Hamiltonian dynamics

Clémentine Courtès[1,2] Emmanuel Franck[1,2] Michael Kraus[3]
Laurent Navoret[1,2] **Léopold Trémant**[1,2]

[1]IRMA, UMR 7501 Université de Strasbourg et CNRS
[2]Inria Nancy-Grand Est, MACARON Project (Strasbourg, France)
[3]Max-Planck-Institut für Plasmaphysik (Garching, Germany)

CANUM

28/05/2024

Outline

Outline

Learning vector fields

$$\dot{u} = f(u)$$

- model the vector field with a neural network
- recover the solution with numerical methods

Learning vector fields

$$\dot{u} = f(u)$$

- model the vector field with a neural network
- recover the solution with numerical methods

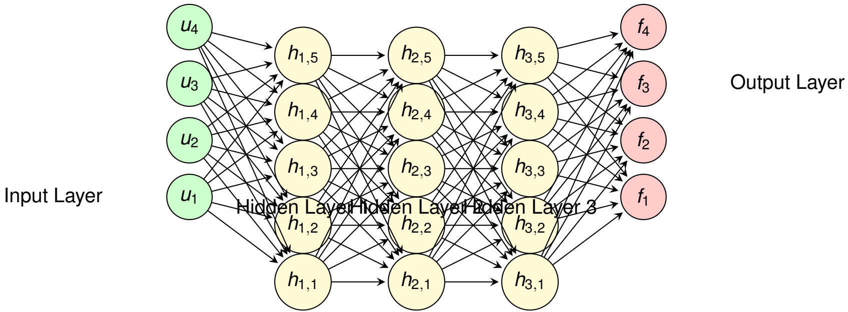


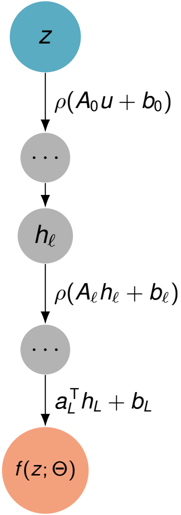
Figure: A multi-layer perceptron (MLP) according to ChatGPT.

The multi-layer perceptron (MLP)

MLPs are a class of *parametric approximation functions* based on compositions.

$$f(z;\Theta) = \underbrace{\mathcal{F}_{\Theta_L} \circ \rho_\odot \circ \mathcal{F}_{\Theta_{L-1}} \circ \ldots \circ \underbrace{\rho_\odot \circ \mathcal{F}_{\Theta_1}(z)}_{h_1}}_{h_L}$$

- $\Theta = (\Theta_1, \ldots, \Theta_L)$ are the parameters;
- $h_1, \ldots, h_L$ are called the “hidden variables”
- $\mathcal{F}_{\Theta_\ell}(h) = A_\ell h + b_\ell$ with $\Theta_\ell = (A_\ell, b_\ell)$;
- ρ is a scalar *activation* function applied component-wise, e.g.
  - $\rho : \lambda \mapsto \max(0, \lambda)$ (ReLU)
  - $\rho : \lambda \mapsto \tanh(\lambda)$
  - $\rho : \lambda \mapsto \log(1 + e^\lambda)$ (Softplus)



The learning / fitting strategy

"Learning" consists in minimizing an objective function, the "loss"

$$\min_{\Theta} \mathcal{L}(\Theta) := \frac{1}{N} \sum_{b=1}^{N} d\left(f\left(z^{(b)}, \Theta\right),\ f\left(z^{(b)}\right)\right)$$

Some important precisions

1. The gradient w.r.t. Θ is computed using automatic differentiation;
2. This is done using stochastic gradient descent, i.e. *batches*;
   - the dataset $\left\{\left(z^{(b)}, f(z^{(b)})\right)\right\}$ is too large otherwise
   - reduces overfitting, as a sort of annealing term
3. Sometimes the objective function is not the immediate output of the neural network.
   - fit the *differentials* of the output
   - penalisation on the parameters $\|\Theta\|$
   - regularisation using e.g. finite differences

What are Hamiltonian dynamics?

Derive the dynamics on the *position & momentum* from a Hamiltonian $(q, p) \mapsto H(q, p) \in \mathbb{R}$,

$$\dot{q} = \nabla_p H(q, p), \qquad \dot{p} = -\nabla_q H(q, p)$$

$$\underbrace{u = \begin{bmatrix} q \\ p \end{bmatrix}}_{\text{coordinates}}, \qquad \underbrace{\dot{u} = J^{-1} \nabla H(u)}_{\text{Hamiltonian dynamics}}, \qquad \underbrace{J = \begin{bmatrix} 0 & -I \\ I & 0 \end{bmatrix}}_{\text{symplectic form}}$$

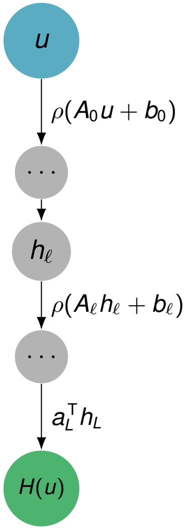
Non-canonical problems

- the symplectic form derives from a potential

Lotka-Volterra | magnetic field lines | guiding center

Learn a symplectic potential
and a Hamiltonian with a neural network

[Greydanus, Dzamba, and Yosinski 2019]
[Chen, Matsubara, and Yaguchi 2021]



The importance of structure – Lotka-Volterra

$$\begin{pmatrix} \dot{x} \\ \dot{y} \end{pmatrix} = \begin{pmatrix} 0 & -xy \\ xy & 0 \end{pmatrix} \begin{pmatrix} 1 - 1/x \\ 1 - 1/y \end{pmatrix}$$

➡ fit the symplectic potential A and the Hamiltonian H on $\dot{x}, \dot{y}$

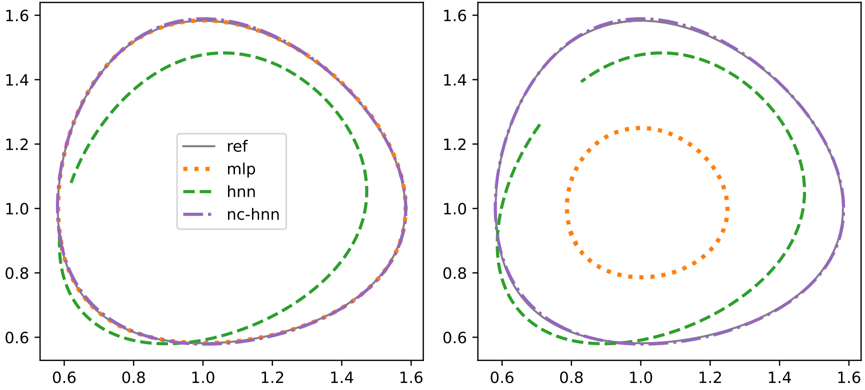
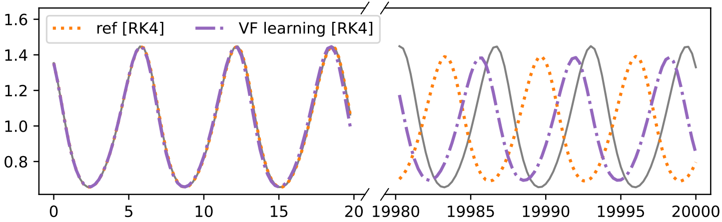
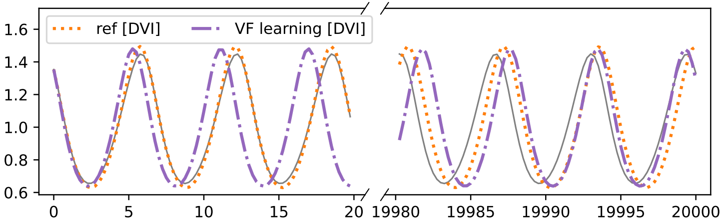


Figure: Sample trajectory in the validation set for (left) $t \in [0, 6.5]$ and (right) $t \in [793.5, 800]$, computed using DOP853 with tolerances $a_{\text{tol}} = r_{\text{tol}} = 10^{-10}$.

The importance of the scheme



Time-evolution $t \mapsto x(t)$ of a solution of the Lotka-Volterra problem, $\Delta t = 1/4$.



Standard scheme

short time

long time

Geometric scheme

short time

discard vector field learning (in this context)

Outline

Why learn the exact vector field?

Since the end goal is simulation, we could instead learn the mapping

$$u(0) \underset{t \mapsto \varphi_t}{\overset{\varphi_{\Delta t}}{\dashrightarrow}} u(\Delta t) \underset{t \mapsto \varphi_t}{\overset{\varphi_{\Delta t}}{\dashrightarrow}} u(2\Delta t)$$

! assume the dataset is now $\left\{\left(u^{(b)}(0), u^{(b)}(\Delta t)\right)\right\}$.

> Why learn $f(\,\cdot\,;\Theta) \approx f$
> when we could learn $\Phi_{\Delta t}(\,\cdot\,;\Theta) \approx \varphi_{\Delta t}$?

Use a numerical integrator e.g.

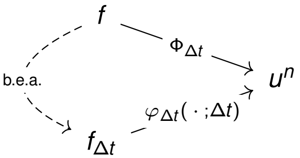
Explicit Euler: $\varphi_{\Delta t}(u) = u + \Delta t\, \widetilde{f}_{\Delta t}(u;\Theta)$

Implicit Euler: $\varphi_{\Delta t}(u) = u + \Delta t\, \widetilde{f}_{\Delta t}\big(\varphi_{\Delta t}(u);\Theta\big)$

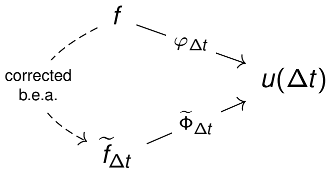
? Does the *modified* VF $\widetilde{f}_{\Delta t}$ exist? Is it Hamiltonian?

Backward error analysis in the linear case

the numerical solution is the exact solution of a perturbed vf



the exact solution is the numerical solution of a modified vf



For a linear problem $\dot{z} = Mz$,

$$\left.\begin{aligned} \varphi_{\Delta t} &= e^{\Delta t\, M} \\ \widetilde{\Phi}_{\Delta t} &= \mathrm{id} + \Delta t\, \widetilde{M}_{\Delta t} \end{aligned}\right\} \implies \widetilde{M}_{\Delta t} = \frac{1}{\Delta t}\left(e^{\Delta t\, M} - \mathrm{id}\right).$$

In the non-linear case, the best we can get is

$$\|\varphi_{\Delta t} - \widetilde{\Phi}_{\Delta t}\| \lesssim e^{-1/\Delta t}$$

[Chartier, Hairer, and Vilmart 2010]

Error convergence without structure

$$\mathrm{err} = \left\| e^{TM} - \left(\mathrm{id} + \delta t\, \widetilde{M}_{\Delta t}\right)^{T/\delta t} \right\|, \qquad M = J^{-1} I_2 = \begin{pmatrix} 0 & 1 \\ -1 & 0 \end{pmatrix}.$$

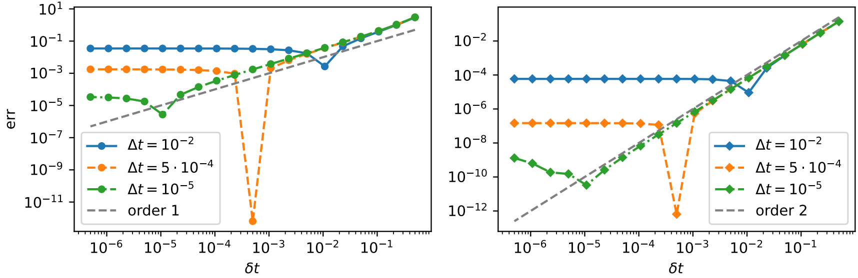


Figure: Error at a time $T = 5$ for different learning time-steps Δt and simulation time-steps δt using the explicit Euler method (left) and the midpoint method (right).

➔ simulate with the same time-step as for learning

Error convergence with structure

$$\mathrm{err} = \left\| e^{TM} - \left(\mathrm{id} + \delta t \, J^{-1} \widetilde{H}_{\Delta t}\right)^{T/\delta t} \right\|, \qquad \widetilde{H}_{\Delta t} = \frac{1}{2}\left(J\widetilde{M}_{\Delta t} + (J\widetilde{M}_{\Delta t})^{\mathsf{T}}\right)$$

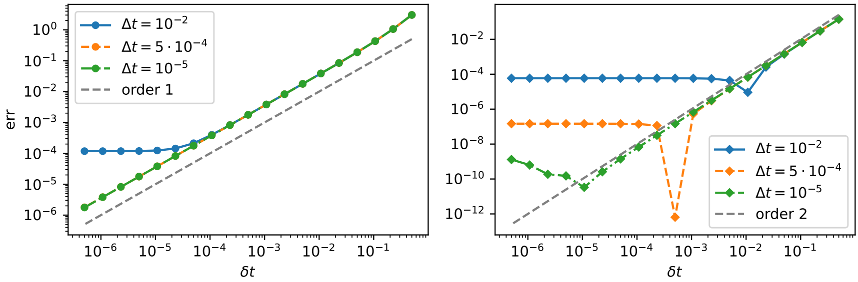


Figure: Error at a time $T = 5$ for different learning time-steps Δt and simulation time-steps δt using the explicit Euler method (left) and the midpoint method (right).

➜ use a scheme appropriate for the structure

Outline

An appropriate numerical scheme

Need the *gauge ansatz*

$$z = \begin{bmatrix} x \\ y \end{bmatrix}, \qquad A(z) = \begin{bmatrix} \vartheta(x, y) \\ 0 \end{bmatrix}.$$

which generates the dynamics

$$(D_y\vartheta)^{\mathsf{T}}\dot{x} = \nabla_y H, \qquad \left(D_x\vartheta^{\mathsf{T}} - D_x\vartheta\right)\dot{x} - D_y\vartheta\,\dot{y} = \nabla_x H.$$

The “appropriate” scheme we consider is from [Ellison et al. 2018],

$$\begin{cases} \vartheta^{n+1} = \vartheta^n + (D_x\vartheta^n)^{\mathsf{T}}(x^n - x^{n-1}) - \Delta t\nabla_x H^n \\ x^{n+1} = x^n + \Delta t\left(D_y\vartheta^{n+1}\right)^{-\mathsf{T}}\nabla_y H^{n+1} \end{cases}$$

where exponents denote the time step of the arguments.

> ! The scheme is sensitive to perturbations on ϑ, even if they don’t impact the z-dynamics! !

Learning using the scheme

- Learn by minimizing the squared norm of the scheme

$$S^n_{\Delta t} = \frac{1}{\Delta t}\begin{bmatrix} \vartheta^{n+1} - \vartheta^n - (D_x\vartheta^n)^\mathsf{T}(x^n - x^{n-1}) + \Delta t \nabla_x H^n \\ (D_y\vartheta^{n+1})^\mathsf{T}(x^{n+1} - x^n) - \Delta t \nabla_y H^{n+1} \end{bmatrix}$$

with (x^{n-1}, z^n, z^{n+1}) known from the dataset.

- Avoid the trivial action $(\vartheta, H) \to (0, 0)$ with a regularisation term

$$\exp\left(-C \left\| \left(D_{z^{n+1}} S^n_{\Delta t}\right)^{-1} \right\|^{-2}\right)$$

- basically forces the smallest singular value of $D_y\vartheta$ to be “large”.

- In theory the loss can go to $e^{-1/\Delta t} \ll 1$

Results on scheme learning

C. Courtès, E. Franck, M. Kraus, L. Navoret, **LT**. In preparation (2024).

$$u(0) \xrightarrow[t \mapsto \varphi_t]{\Phi_{\Delta t,\mathrm{NN}}} u(\Delta t) \xrightarrow[t \mapsto \varphi_t]{\Phi_{\Delta t,\mathrm{NN}}} u(2\Delta t)$$

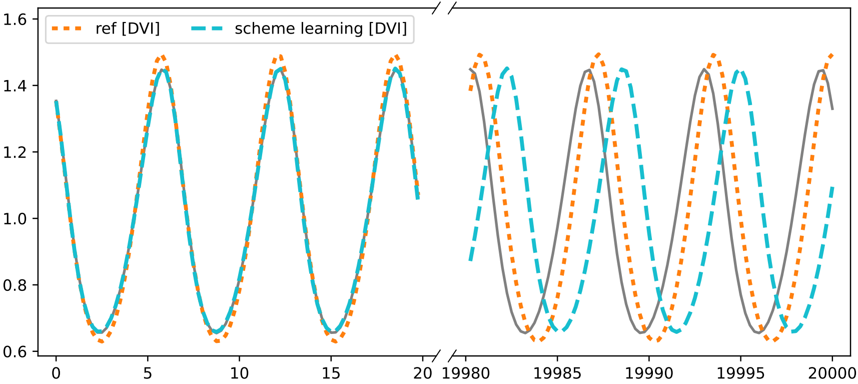


Figure: Time evolution $t \mapsto x(t)$ for a Lotka-Volterra problem, $\Delta t = 1/4$.

Summary

Quick overview

- Hamiltonian dynamics (formulas and qualitative behaviour)
- learning and the importance of structure

Vector field learning

- simple approach
- great for standard schemes
- unexpected behaviour

Scheme learning

- goal-oriented loss
- long-time simulation
- fixed time-step

What’s left

- do more test cases
- caracterize the “best” symplectic potential

Summary

Quick overview

- Hamiltonian dynamics (formulas and qualitative behaviour)
- learning and the importance of structure

Vector field learning

- simple approach
- great for standard schemes
- unexpected behaviour

Scheme learning

- goal-oriented loss
- long-time simulation
- fixed time-step

What's left

- do more test cases
- caracterize the “best” symplectic potential

Thank you for you attention!

References

Chartier, Philippe, Ernst Hairer, and Gilles Vilmart (2010). “Algebraic structures of B-series”. In: *Foundations of Computational Mathematics* 10, pp. 407–427.

Chen, Yuhan, Takashi Matsubara, and Takaharu Yaguchi (2021). “Neural Symplectic Form: Learning Hamiltonian Equations on General Coordinate Systems”. In: *Advances in Neural Information Processing Systems*. Vol. 34. Curran Associates, Inc., pp. 16659–16670.

Ellison, C. L. et al. (May 4, 2018). “Degenerate Variational Integrators for Magnetic Field Line Flow and Guiding Center Trajectories”. In: *Physics of Plasmas* 25.5, p. 052502.

Greydanus, Samuel, Misko Dzamba, and Jason Yosinski (2019). “Hamiltonian Neural Networks”. In: *Advances in Neural Information Processing Systems*. Vol. 32. Curran Associates, Inc.