Neural Non-Canonical Hamiltonian Dynamics for Long-Time Simulations

Clémentine Courtès[1,2] Emmanuel Franck[1,2] Michael Kraus[3]
Laurent Navoret[1,2] **Léopold Trémant**[4]

[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)
[4] Laboratoire de Mathématiques de Lens, Université d'Artois (Lens, France)

WCCM-ECCOMAS 2026
MS187 - Structure-Preserving Scientific Machine Learning and Neural Networks

Outline

Outline

What are Hamiltonian dynamics?

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

$$\begin{cases} \dot{q} = \nabla_p H(q,p), \\ \dot{p} = -\nabla_q H(q,p). \end{cases}$$

$$\underbrace{u = \begin{bmatrix} q \\ p \end{bmatrix}}_{\text{coordinates}}, \qquad \underbrace{J = \begin{bmatrix} 0 & -I \\ I & 0 \end{bmatrix}}_{\text{symplectic form}}$$

$$\underbrace{\dot{u} = J^{-1} \nabla H(u)}_{\substack{\text{Hamiltonian} \\ \text{dynamics}}}$$

[Greydanus, Dzamba, and Yosinski 2019]

Non-canonical problems

- the symplectic form derives from a potential

(Lotka-Volterra / magnetic field lines / guiding center)

$$\frac{\mathrm{d}}{\mathrm{d}t} A(z) = \nabla_z [A(z)\dot{z} - H(z)]$$

$$\Updownarrow \quad W(z) = D_z A(z)^\top - D_z A(z)$$

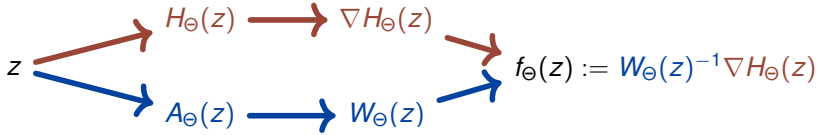
$$\dot{z} = W(z)^{-1} \nabla H(z)$$

- $A(u)\dot{u} = p \cdot \dot{q}$

Learn a symplectic potential and a Hamiltonian with a neural network

[Chen, Matsubara, and Yaguchi 2021]

Learning procedure [Chen, Matsubara, and Yaguchi 2021]



Architecture [Gnanasambandam et al. 2022]

- small fully-connected MLPs
- self-scalable Tanh activation

Dataset

$$\{(z, f(z)),\ z \text{ sampled randomly}\}$$

Loss function

$$\mathcal{C}(\Theta) = \mathbb{E}_z\left[\|f_\Theta(z) - f(z)\|^2\right]$$

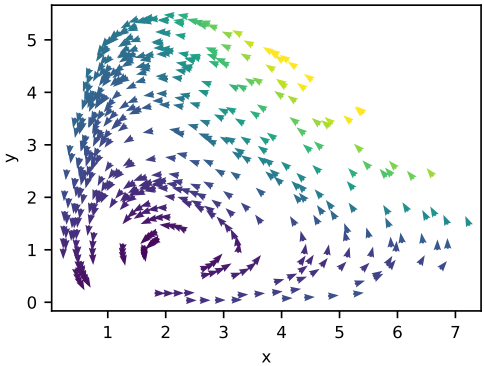


Figure: Sample of the dataset for Lotka-Volterra

The Lotka-Volterra problem

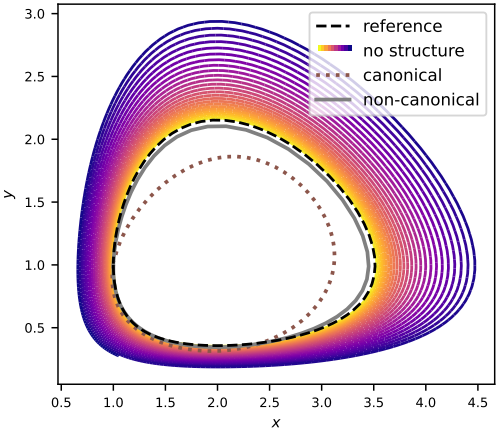


Figure: Comparison of neural network solutions for different approaches using a highly-accurate `solve_ivp`

Dynamics

$$\dot{x} = x(1-x), \quad \dot{y} = y(x-2).$$

Hamiltonian structure

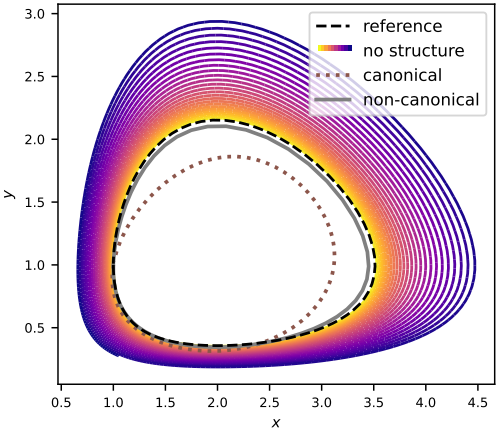
$$A(x,y) = \begin{pmatrix} -\ln(y)/x \\ 0 \end{pmatrix},$$

$$H(x,y) = x + y - \ln\left(x^2 y\right).$$

Approaches

- **Reference** known model
- **No structure** $z \mapsto f_\Theta(z)$
- **Canonical** $z \mapsto J^{-1} \nabla H_\Theta(z)$
- **NC** $z \mapsto W_\Theta(z)^{-1} \nabla H_\Theta(z)$
  $W_\Theta(z) = D_z A_\Theta(z)^\top - D_z A_\Theta(z)$

Not in phase-space



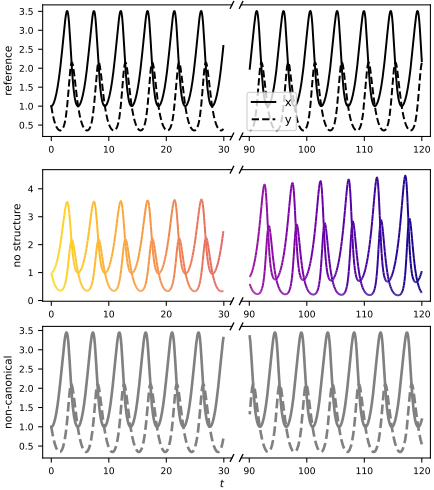


Figure: Comparison of neural network solutions for different approaches using a highly-accurate `solve_ivp`

Degeneracy condition and ansatz [Ellison et al. 2018]

Gauge ansatz

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

- For canonical problems, use the tautological potential $\vartheta(q,p) = p$ [Vermeeren 2018]

Degenerate variational integrator

One-step 1st order method based on

$$\begin{cases} \dot{p} = D_x\vartheta(x,y)\,\dot{x} - \nabla_x H(x,y) \\ 0 = D_y\vartheta(x,y)\dot{x} - \nabla_y H(x,y) \\ p = \vartheta(x,y) \end{cases}$$

- The Jacobian $D_y\vartheta$ must be invertible

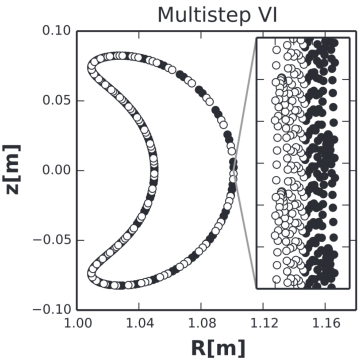


Figure: Result of a multistep VI, pulled from [Ellison et al. 2018, Fig. 5]. "Even-indexed points are marked in white and odd-indexed points in black."

Application with RK4

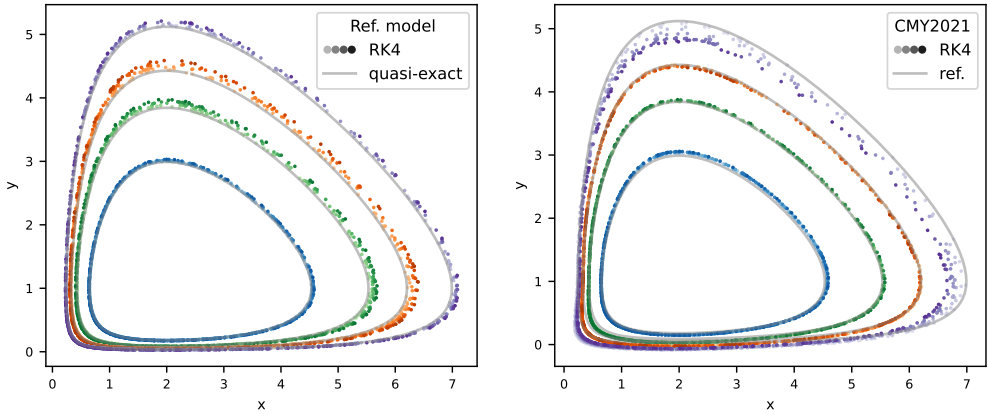


Figure: Evolution of the numerical solution using RK4 over 100k time-steps with $\Delta t = 0.1$. Grey line: exact solution of the reference model.

behavior similar to the reference model

Application to the neural model

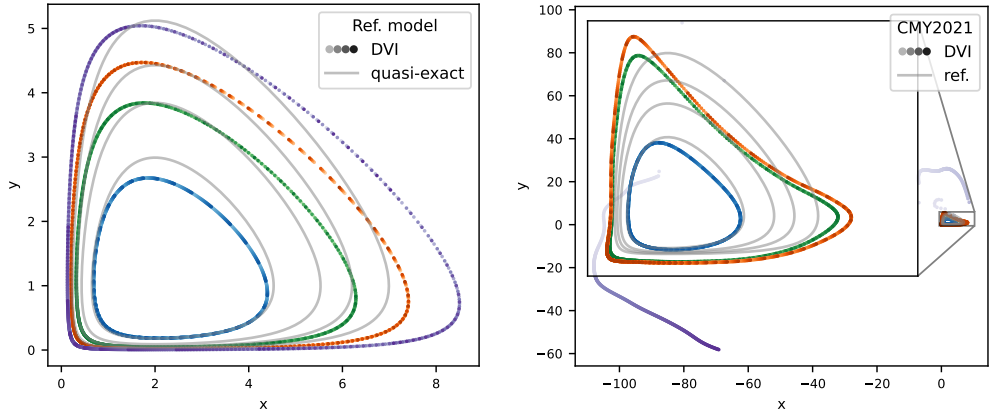


Figure: Evolution of the numerical solution using the DVI over 100k time-steps with $\Delta t = 0.1$. Grey line: exact solution of the reference model.

immediate instability, even with a small time step

Error term of the scheme

The DVI from [Ellison et al. 2018] is written

$$\begin{cases} \vartheta_{n+1} = \vartheta_n + (D_x \vartheta_n)^{\mathsf{T}}(x_n - x_{n-1}) - \Delta t \nabla_x H_n \\ x_{n+1} = x_n + \Delta t \left(D_y \vartheta_{n+1}\right)^{-\mathsf{T}} \nabla_y H_{n+1} \end{cases}$$

where indices denote the time step of the arguments.

Dominant error term

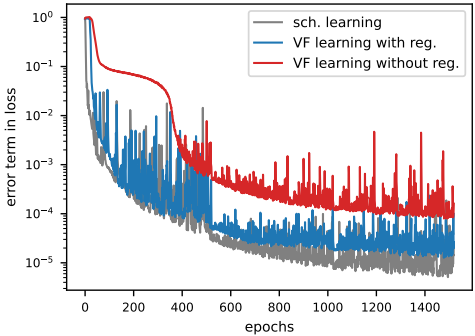
$$\begin{cases} x(h) - x_h = -\dfrac{h^2}{2}\ddot{x} + \mathcal{O}(h^3), \\ y(h) - y_h = \dfrac{h^2}{2} D_y \vartheta^{-1} \left(\ddot{\vartheta} + D_x \vartheta \ddot{x}\right) + \mathcal{O}(h^3). \end{cases}$$

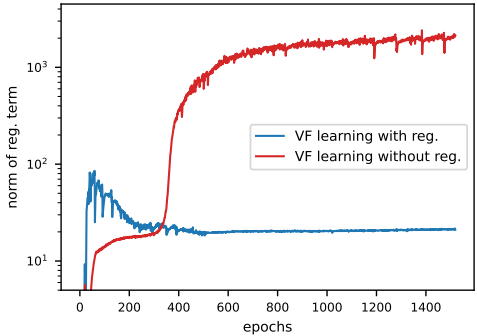
The scheme is sensitive to perturbations on ϑ, even if they don't impact the vector field!

- Compute this using ϑ_Θ, H_Θ, z and $f(z)$ in the loss,

$$\mathcal{C}(\Theta) = \mathbb{E}_z\left[\|f_\Theta(z) - f(z)\|^2 + \varepsilon \|r_\Theta(z, f(z))\|^2\right]$$

Training results





Analysis of the dominant error term

- Penalization has no impact on the vector field loss
- It does blow-up if not penalized
- Does it stabilize the scheme?

Result before regularization

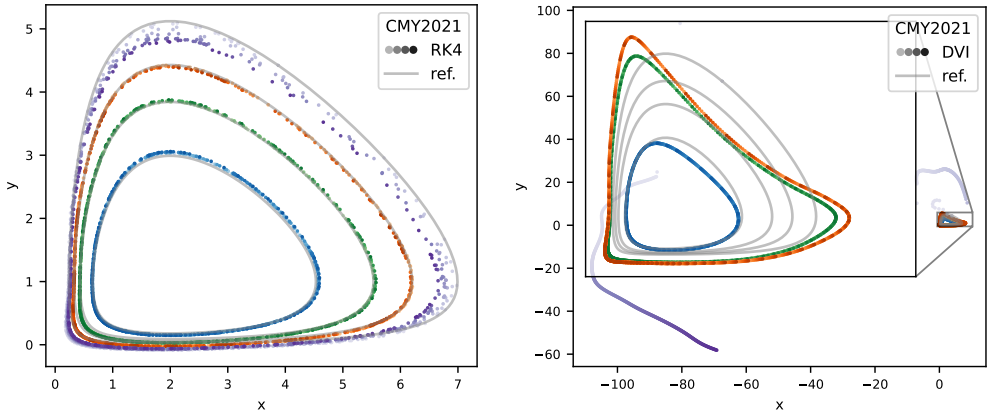


Figure: Evolution of the numerical solution for the “naive” neural model using RK4 (left) and the DVI (right) over 100k time-steps with $\Delta t = 0.1$. Grey line: exact solution of the reference model.

Result after regularization

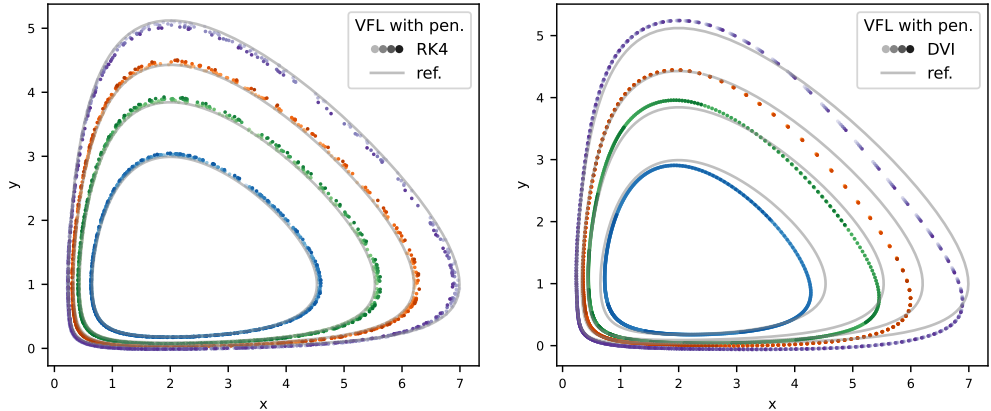


Figure: Evolution of the numerical solution for the neural model **with penalization** using RK4 (left) and the DVI (right) over 100k time-steps with $\Delta t = 0.1$. Grey line: exact solution of the reference model.

The instability is fixed, and the solution is better than on the reference model

Outline

Why learn the exact vector field?

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

$$z(0) \xrightarrow[t \mapsto \varphi_t]{\varphi_{\Delta t}} z(\Delta t) \xrightarrow[t \mapsto \varphi_t]{\varphi_{\Delta t}} z(2\Delta t)$$

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

Snapshot dataset

$$\Big\{\big(z, \varphi_{\Delta t}(z), \varphi_{2\Delta t}(z)\big), z \text{ sampled randomly}\Big\}$$

- high-precision short-time simulation
- Δt chosen at the limit of stability

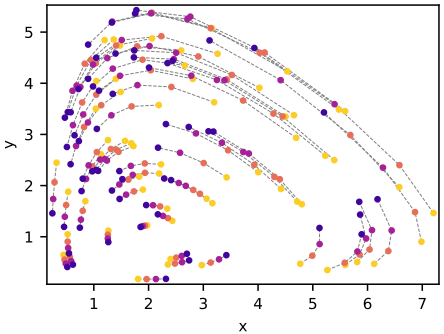


Figure: Sample of the snapshot dataset for LV

Strategies for **canonical** dynamics

The flow $(t, u) \mapsto \varphi_t(u)$ is *symplectic*, meaning that for all (t, u),

$$\left(D_u\varphi_t(u)\right)^\mathsf{T} J\, D_u\varphi_t(u) = J$$

1. **Impose it weakly** by adding a term to the loss.

   👎 Will drift over time.

2. **Use another architecture** e.g. a SympNet [Jin, Zhang, Zhu, et al. 2020] or a HénonNet [Burby, Q. Tang, and Maulik 2021]

   $$\varphi_{\Delta t}(u) = \varphi^{[L]} \circ \ldots \circ \varphi^{[1]}(u).$$

   👎 No guarantees for non-canonical systems [Jin, Zhang, Kevrekidis, et al. 2020]

3. **Use a numerical integrator** i.e. learn a Hamiltonian such that the dynamics are recovered numerically, e.g. for the midpoint [David and Méhats 2021]

   $$\varphi_{\Delta t}(u) = u + \Delta t\, J^{-1} \nabla \widetilde{H}_{\Delta t}\left(\tfrac{1}{2}\left[u + \varphi_{\Delta t}(u)\right]\right)$$

Fit a numerical integrator?

- **Deep Euler** [Shen, Cheng, and Liang 2020]
- **HyperSolvers** [Poli et al. 2020; Mathiesen, Yang, and Hu 2022]
- **General analysis** [David and Méhats 2021; Bouchereau et al. 2023]

Linear case without structure and explicit Euler, learn $\widetilde{M}_{\Delta t}$,

$$\dot{z} = Mz \quad \Longrightarrow \quad \left| \begin{array}{l} \varphi_{\Delta t} = e^{\Delta t\, M} \\ \widetilde{\Phi}_{\Delta t} = \mathrm{id} + \Delta t\, \widetilde{M}_{\Delta t} \end{array} \right\} \quad \Longrightarrow \quad \widetilde{M}_{\Delta t} = \frac{1}{\Delta t} \left(e^{\Delta t\, M} - \mathrm{id} \right).$$

Backward error analysis [Hairer, Lubich, and Wanner 2006, Chap. IX]

- Does this modified vector field exist?
- With non-linearities, some error is expected,

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

Scheme learning the DVI

Notation setting $z_{n+1} = (x_{n+1}, y_{n+1})$, we write the scheme as

$$S_{\Delta t}(z_n, z_{n+1}) = 0, \qquad \Leftrightarrow \qquad \begin{cases} \vartheta_{n+1} - \vartheta_n = (D_x \vartheta_n)^\mathsf{T}(x_n - x_{n-1}) - \Delta t \nabla_x H_n \\ 0 = \left(D_y \vartheta_{n+1}\right)^\mathsf{T}(x_{n+1} - x_n) - \Delta t \nabla_y H_{n+1} \end{cases}$$

⚠ Avoid the trivial solution $\vartheta_\Theta = 0$, $H_\Theta = 0$

Training phase minimize Newton residual: setting $\mathcal{J}_{\Delta t} = D_2 S_{\Delta t}\big(z, \varphi_{\Delta t}(z)\big)$,

$$\mathcal{C}(\Theta) = \mathbb{E}_z\left[\left\|\mathcal{J}_{\Delta t}^{-1} S_{\Delta t}\big(z, \varphi_{\Delta t}(z)\big)\right\|^2\right] + \varepsilon \log(\kappa_{\mathrm{sch}}(\mathcal{J}_{\Delta t}))$$

👍 Homogeneous in phase-space $\Rightarrow$ compatible with multi-scale data
(compared to e.g. [Offen and Ober-Blöbaum 2024])

⚠ Expensive to compute, a pre-training may be preferable

Scheme learning – Result using the DVI

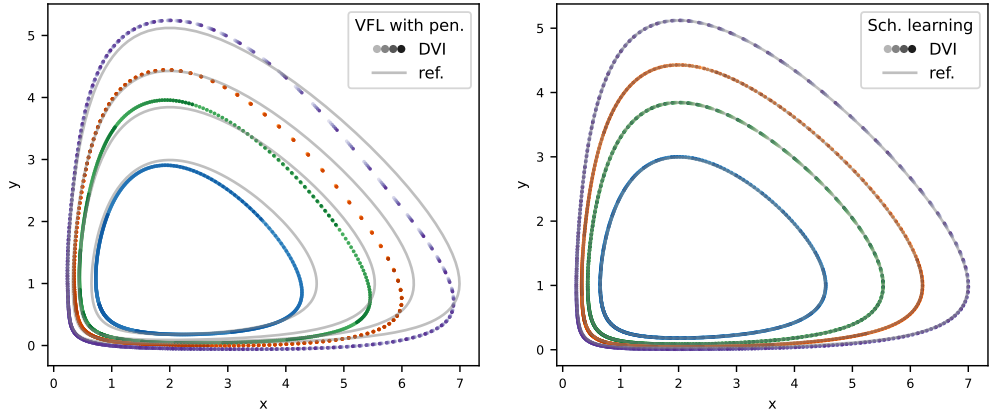


Figure: **VF learning (left) and scheme learning (right):**evolution of the numerical solution over 100k time-steps of the DVI with $\Delta t = 0.1$. Grey line: exact solution of the reference model.

More accurate when using this scheme

Scheme learning – Result using RK4

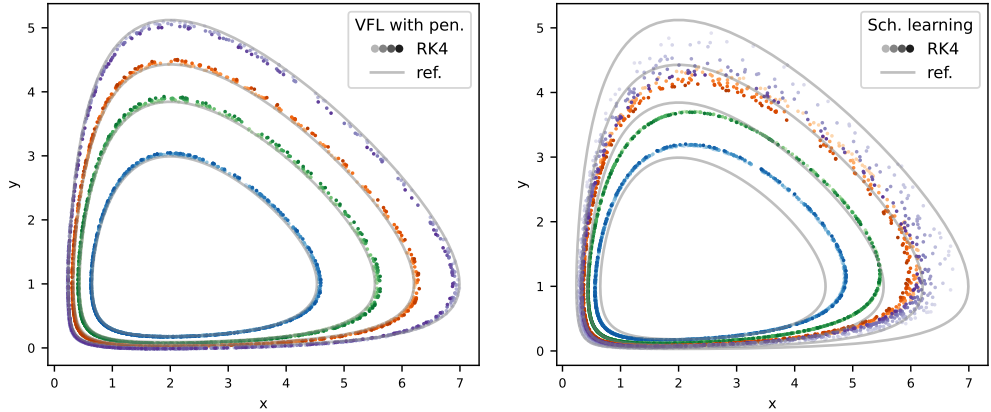
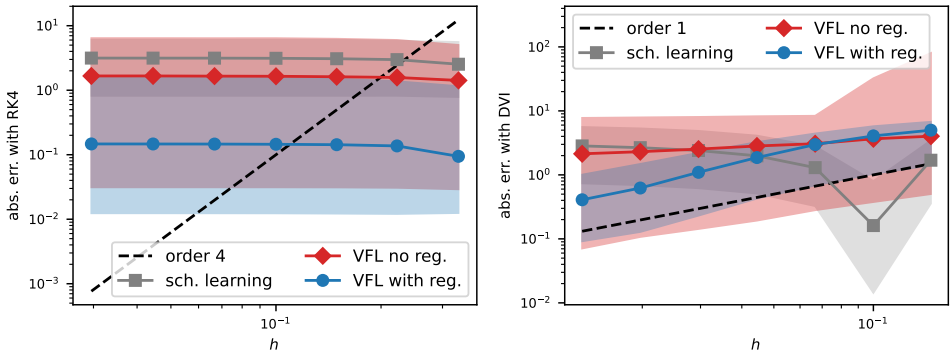


Figure: Evolution of the numerical solution for VF learning (left) and scheme learning (right) over 100k time-steps of RK4 with $\Delta t = 0.1$. Grey line: exact solution of the reference model.

Scheme learning needs to always use the same scheme

Effect of the time step



- Scheme learning needs to always use the same time step and scheme

Outline

What is the guiding center?

Asymptotic model in tokamaks with a strong magnetic field

- poloidal-toroidal coordinates $X = (r, \theta, \phi)$
- momentum is just u in the toroidal direction
- magnetic potential $A = (0, A_\theta, A_\phi)$, from which $\mathbf{B} = \nabla \times A$
- magnetic field unit vector $b = R_0 + r\cos(\theta)$, here $R_0 = 1$
- assume axisymmetry (invariance of dynamics w.r.t. toroidal angle ϕ)

Choose a "nice" Lagrangian e.g. [Qin, Guan, and W. M. Tang 2009]

$$\mathcal{L}(r, \theta, \phi, u, \theta_t, \phi_t) = A_\theta(r, \theta)\theta_t + \big(A_\phi(r) + ub(r, \theta)\big)\phi_t - H(r, \theta, u).$$

Interesting trajectories

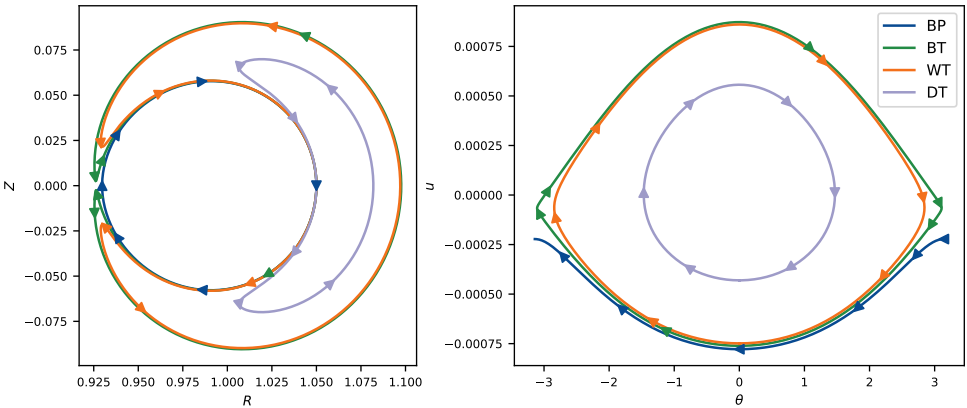


Figure: **BP** barely passing trajectory ; **BT** barely trapped trajectory ; **WT** well trapped trajectory ; **DT** deeply trapped trajectory (also called “banana orbit”).

The need for geometric integration

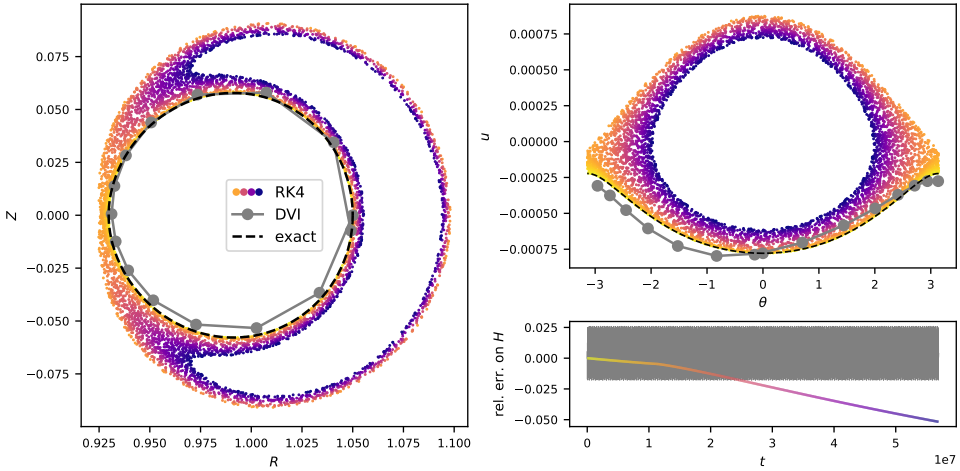


Figure: Comparison of the numerical solution over thousands of periods, DT (deeply trapped) traj.

Dataset and architecture

Pre-processing axisymmetry & 2π-periodicity [Dong and Ni 2021] w.r.t. θ

$$z = \begin{pmatrix} r \\ \theta \\ \phi \\ u \end{pmatrix} \mapsto \begin{bmatrix} \frac{r - r_{\min}}{r_{\max} - r_{\min}} \\ \cos_{\odot}(\theta \mathbb{1} + \boldsymbol{\varphi}^{[\Theta]}) \\ \frac{u - u_{\min}}{u_{\max} - u_{\min}} \end{bmatrix} \quad \begin{matrix} \mapsto H_\Theta \quad \sim \text{4k parameters} \\ \mapsto A_\Theta \quad \sim \text{4k parameters} \end{matrix}$$

Generating the dataset some "width" around $(r, \theta, \phi) = (0.05, 0, 0)$

$$r \in (0.03, 0.055), \qquad \theta \in (-\tfrac{1}{10}\pi, \tfrac{1}{10}\pi),$$
$$\phi \in (0, 2\pi), \qquad u \in (-9 \times 10^{-4}, -3 \times 10^{-4})$$

- 600 points sampled with a latin hypercube method
- simulate for 3 (banana —deeply trapped) periods
- 20 time-steps per period

Specific loss

Multi-scale dynamics $|\dot{u}| \sim 10^{-4} \ll |\dot{r}| \sim 10^{-2} \ll |\dot{\theta}| \sim 1$

- how to rescale data such that every component is learnt?
- change the norm in the loss using a Gram matrix

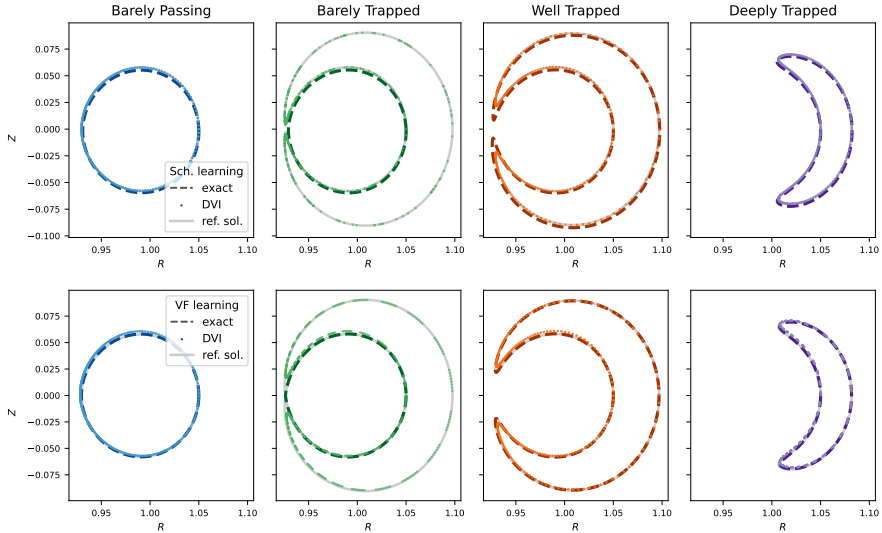
Vector-field learning $\mathcal{C}(\Theta) = \mathbb{E}_z\left[\|f_\Theta(z) - f(z)\|_{\mathrm{vf}}^2 + \varepsilon\|r_\Theta(z, f(z))\|_{\mathrm{vf}}^2\right]$

$$\|\delta\|_{\mathrm{vf}}^2 = \delta^{\mathsf{T}}\left(\mathbb{E}_z\left[f(z)f(z)^{\mathsf{T}}\right]\right)^{-1}\delta$$

Scheme learning $\mathcal{C}(\Theta) = \mathbb{E}_z\left[\left\|\mathcal{J}_{\Delta t}^{-1}\mathcal{S}_{\Delta t}(z, \varphi_{\Delta t}(z))\right\|_{\mathrm{sch}}^2\right] + \varepsilon\log(\kappa_{\mathrm{sch}}(\mathcal{J}_{\Delta t}))$

$$\|\delta\|_{\mathrm{sch}}^2 = \delta^{\mathsf{T}}\left(\mathbb{E}_z\left[(\varphi_{\Delta t}(z) - z)(\varphi_{\Delta t}(z) - z)^{\mathsf{T}}\right]\right)^{-1}\delta$$

Results on the guiding center



Summary

Quick overview

- description of non-canonical Hamiltonian dynamics
- the importance of encoding structure in learning
- two distinct learning strategies

Vector field learning

- simple approach
- accurate to physics
- adapt loss for numerics

Scheme learning

- goal-oriented loss
- long-time simulation
- fixed (but large) time-step

What's left

- time-dependent problems, e.g. magnetic field lines
- thorough backward error analysis of the DVI

Summary

Quick overview

- description of non-canonical Hamiltonian dynamics
- the importance of encoding structure in learning
- two distinct learning strategies

Vector field learning

- simple approach
- accurate to physics
- adapt loss for numerics

Scheme learning

- goal-oriented loss
- long-time simulation
- fixed (but large) time-step

What's left

- time-dependent problems, e.g. magnetic field lines
- thorough backward error analysis of the DVI

Thanks for your attention!

References I

Bouchereau, Maxime et al. (Apr. 18, 2023). *Machine Learning Methods for Autonomous Ordinary Differential Equations*. arXiv: 2304.09036 [cs, math]. URL: http://arxiv.org/abs/2304.09036 (visited on 07/21/2023). Pre-published.

Burby, J W, Q Tang, and R Maulik (Feb. 1, 2021). "Fast Neural Poincaré Maps for Toroidal Magnetic Fields". In: *Plasma Physics and Controlled Fusion* 63.2, p. 024001. ISSN: 0741-3335, 1361-6587.

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.

David, Marco and Florian Méhats (June 22, 2021). *Symplectic Learning for Hamiltonian Neural Networks*. arXiv: 2106.11753 [cs, math]. Pre-published.

Dong, Suchuan and Naxian Ni (June 15, 2021). "A Method for Representing Periodic Functions and Enforcing Exactly Periodic Boundary Conditions with Deep Neural Networks". In: *Journal of Computational Physics* 435, p. 110242. ISSN: 0021-9991.

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. ISSN: 1070-664X.

References II

Gnanasambandam, Raghav et al. (Apr. 29, 2022). *Self-Scalable Tanh (Stan): Faster Convergence and Better Generalization in Physics-informed Neural Networks*. arXiv: 2204.12589. Pre-published.

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

Hairer, Ernst, Christian Lubich, and Gerhard Wanner (2006). *Geometric Numerical Integration: Structure-Preserving Algorithms for Ordinary Differential Equations*. 2nd ed. Springer Series in Computational Mathematics. Berlin Heidelberg: Springer-Verlag. ISBN: 978-3-540-30663-4.

Jin, Pengzhan, Zhen Zhang, Ioannis G. Kevrekidis, et al. (Dec. 5, 2020). *Learning Poisson Systems and Trajectories of Autonomous Systems via Poisson Neural Networks*. arXiv: 2012.03133 [cs]. URL: http://arxiv.org/abs/2012.03133 (visited on 08/04/2023). Pre-published.

Jin, Pengzhan, Zhen Zhang, Aiqing Zhu, et al. (Dec. 2020). "SympNets: Intrinsic Structure-Preserving Symplectic Networks for Identifying Hamiltonian Systems". In: *Neural Networks* 132, pp. 166–179. ISSN: 08936080.

References III

Mathiesen, Frederik Baymler, Bin Yang, and Jilin Hu (June 28, 2022). "Hyperverlet: A Symplectic Hypersolver for Hamiltonian Systems". In: *Proceedings of the AAAI Conference on Artificial Intelligence* 36.4 (4), pp. 4575–4582. ISSN: 2374-3468.

Offen, Christian and Sina Ober-Blöbaum (Jan. 8, 2024). "Learning of Discrete Models of Variational PDEs from Data". In: *Chaos: An Interdisciplinary Journal of Nonlinear Science* 34.1, p. 013104. ISSN: 1054-1500.

Poli, Michael et al. (2020). "Hypersolvers: Toward Fast Continuous-Depth Models". In: *Advances in Neural Information Processing Systems*. Vol. 33. Curran Associates, Inc., pp. 21105–21117.

Qin, Hong, Xiaoyin Guan, and William M. Tang (Apr. 21, 2009). "Variational Symplectic Algorithm for Guiding Center Dynamics and Its Application in Tokamak Geometry". In: *Physics of Plasmas* 16.4, p. 042510. ISSN: 1070-664X.

Shen, Xing, Xiaoliang Cheng, and Kewei Liang (Mar. 21, 2020). *Deep Euler Method: Solving ODEs by Approximating the Local Truncation Error of the Euler Method.* arXiv: 2003.09573 [math]. URL: http://arxiv.org/abs/2003.09573 (visited on 11/06/2024). Pre-published.

Vermeeren, Mats (July 27, 2018). *Modified Equations for Variational Integrators Applied to Lagrangians Linear in Velocities.* arXiv: 1709.09567. Pre-published.