Learning non canonical Hamiltonian ODEs

Emmanuel Franck[1], C. Courtès[12], M. Kraus[4], L. Navoret[12], L. Tremant[3]

February 20, 2025
DTE AICOMAS, Paris

[1]MACARON project-team, Université de Strasbourg, CNRS, Inria, IRMA, France
[2]IRMA, Université de Strasbourg, France
[3]Université de Lens, France
[4]Max Planck of Plasma physics, Garching bei Munchen, Germany

Outline

Conservative ODE and structure preserving numerical methods

Conservative ODE and structure preserving numerical methods

Objective

Structure preserving learning

Learning of canonical Hamiltonian systems

Learning of non-canonical Hamiltonian systems

Objective

- We have temporal data $(\mathbf{x}_1^1, ..., \mathbf{x}_t, ..., \mathbf{x}_T^1)....(\mathbf{x}_1^n, ..., \mathbf{x}_t^n, ..., \mathbf{x}_T^n)$
- A classical task in ML is to learn a model to predict the next state.
- In general we try to find

$$\mathbb{P}(\mathbf{x}_t \mid \mathbf{x}_1, ..., \mathbf{x}_{t-1}), \quad \forall t > 0,$$

- No assumption on the temporal phenomena.

Learn ODE

In this case the objective is to learn f_θ such that

$$\dot{\mathbf{x}}(t) = \mathbf{f}_\theta(\mathbf{x}(t))$$

predict well dynamic associated to the data.

- Reduced modeling for PDEs
- Discover new physical laws (huge amount of data in astrophysics for example)

Lagrangian systems

- Here we are interested by conservative problems.
- The conservative problem are modeled by some specific class of ODEs.

Lagrangian system

Let generalized coordinates $\mathbf{q}(t)$. A Lagrangian system is of the form:

$$\left(\frac{\partial^2 \mathcal{L}}{\partial \dot{\mathbf{q}} \partial \dot{\mathbf{q}}}\right) \ddot{\mathbf{q}} + \left(\frac{\partial^2 \mathcal{L}}{\partial \mathbf{q} \partial \dot{\mathbf{q}}}\right) \dot{\mathbf{q}} = \frac{\partial \mathcal{L}(\mathbf{q}, \dot{\mathbf{q}})}{\partial \mathbf{q}}.$$

where the function $\mathcal{L}(\mathbf{q}, \dot{\mathbf{q}})$ is called the Lagrangian.

- **Noether theorem**: A symmetry of $\mathcal{L}$ induce a conserved quantity.
- Invariance compared to spatial translation and rotation gives generalized impulsion and angular momentum conservation.
- Since $\mathcal{L}$ is time translation invariant the energy $\mathcal{H} = (\mathbf{p}, \dot{\mathbf{q}}) - \mathcal{L}$ with $\mathbf{p} = \frac{\partial \mathcal{L}(\mathbf{q}, \dot{\mathbf{q}})}{\partial \dot{\mathbf{q}}}$ the generalized impulsion.

Canonical Hamiltonian systems

Canonical Hamiltonian system

The dynamic of $\mathbf{x}(t) = (\mathbf{q}(t), \mathbf{p}(t))$ is given by the Hamiltonian system

$$\dot{\mathbf{x}}(t) = \mathcal{J}^{-1}\nabla_{\mathbf{x}}\mathcal{H}(\mathbf{x})$$

with

$$\mathcal{J} = \begin{pmatrix} 0 & -I_N \\ I_n & 0 \end{pmatrix}$$

and $\mathcal{H} = (\mathbf{p}, \dot{\mathbf{q}}) - \mathcal{L}$ the Hamiltonian function.

- We note $\varphi_{\mathcal{H}}(\mathbf{x}_0; t)$ the flow associated to the ODE.
- Properties of the flow:
  - The flow conserve the Hamiltonian function: $\frac{d\mathcal{H}(\varphi_{\mathcal{H}}(\mathbf{x}_0;t))}{dt} = 0$
  - The flow conserve the volume: $Vol(\varphi_{\mathcal{H}}(A;t)) = Vol(A)$ with $Vol(A) = \int_A d\mathbf{x}$

Conservative ODE and structure preserving numerical methods

Objective

Structure preserving learning

Learning of canonical Hamiltonian systems

Learning of non-canonical Hamiltonian systems

Structure preserving learning

ODE learning

- We come back to original question: How learn a time dynamic ?
- Data:

$$\mathcal{X} = \left[(\mathbf{x}_1^1, ..., \mathbf{x}_t, ..., \mathbf{x}_T^1)....(\mathbf{x}_1^n, ..., \mathbf{x}_t^n, ..., \mathbf{x}_T^n)\right]$$

- Using these data we can construct approximation of time derivative. We obtain

$$\dot{\mathcal{X}} = [(\dot{\mathbf{x}}_1^1, ..., \dot{\mathbf{x}}_t, ..., \dot{\mathbf{x}}_T^1)....(\dot{\mathbf{x}}_1^n, ..., \dot{\mathbf{x}}_t^n, ..., \dot{\mathbf{x}}_T^n)]$$

  using you favorite interpolation/finite difference method.

How learn the ODE

We just need to minimize:

$$\min_\theta \sum_{i=1}^{n} \sum_{t=1}^{T-1} \| \dot{\mathbf{x}}_t^i - F_\theta(\mathbf{x}_t^i) \|_2^2$$

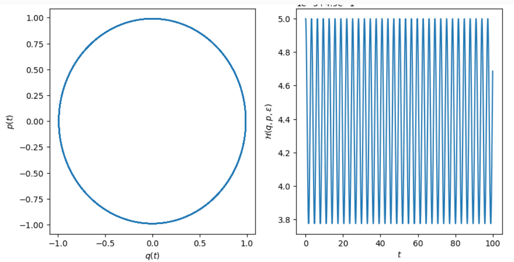
on the full set of trajectories. It is supervised learning.

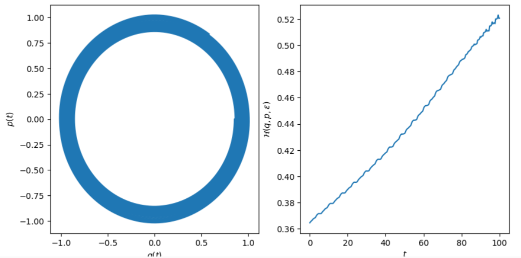
Stability

- How assure that the model are globally well posed, stable etc.
  - Learning on the full trajectories which is equivalent to solve
$$\min_\theta \left(\sum_{i=1}^{n} \int_0^T \parallel \mathbf{x}_\theta(t)^i - \mathbf{x}(t)^i \parallel_2^2 \right)$$
with $\mathbf{x}_\theta(t)^i$ solution of
$$\begin{cases} \frac{d\mathbf{x}_{i,\theta}(t)}{dt} = \mathbf{f}_\theta(\mathbf{x}_{i,\theta}(t)) \\ \mathbf{x}_{i,\theta}(t_0) = \mathbf{x}_i^0 \end{cases}$$
  - Impose a stable structure in the learning.
  - For dissipative system we learn with a gradient flow structure or assuring the existence of Lyapunov function.
  - For conservation problem we can impose the structure in the training.
- **Reference**: *Hamiltonian Neural Networks*, S. Greydanus and al.
- We minimize:
$$\min_\theta \sum_{i=1}^{n} \sum_{t=1}^{T-1} \parallel \dot{\mathbf{x}}_t^i - \mathcal{J}^{-1} \nabla_{\mathbf{x}} H_\theta(\mathbf{x}_t^i) \parallel_2^2$$

Numerical examples

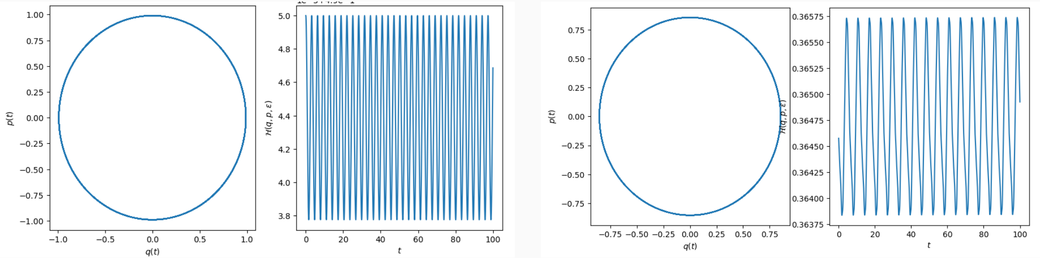
- We learn the oscillator system $H(q,p) = \frac{1}{2}(p^2 + q^2)$. Data generated with $\Delta t = 1e^{-3}$ and $T = 20$
- We solve the learned systems with $\Delta t = 1e^{-2}$ and $T = 100$





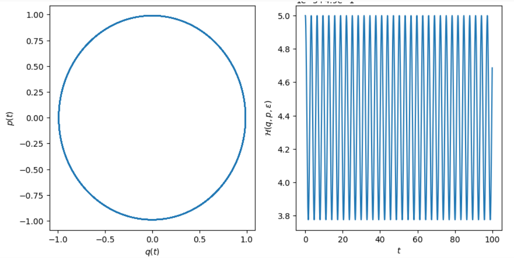
- **Left**: reference solution. **Right**: we learn $\mathbf{f}_\theta$ solved with RK4.

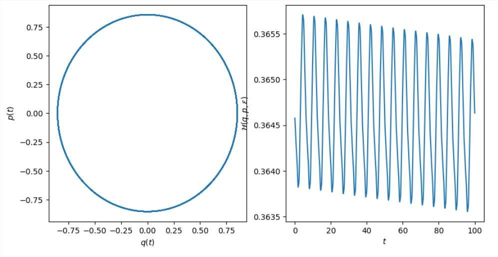
- We learn the oscillator system $H(q,p) = \frac{1}{2}(p^2+q^2)$. Data generated with $\Delta t = 1e^{-3}$ and $T=20$
- We solve the learned systems with $\Delta t = 1e^{-2}$ and $T=100$



- **Left**: reference solution. **Right**: we learn $\mathcal{H}_\theta$ solved with RK4.

- We learn the oscillator system $H(q,p) = \frac{1}{2}(p^2 + q^2)$. Data generated with $\Delta t = 1e^{-3}$ and $T = 20$
- We solve the learned systems with $\Delta t = 1e^{-2}$ and $T = 100$

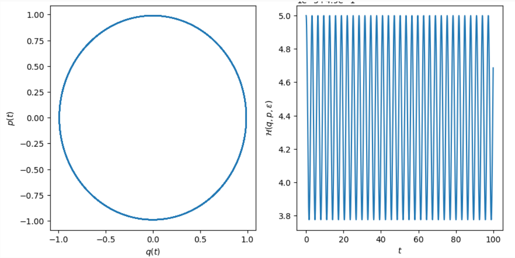


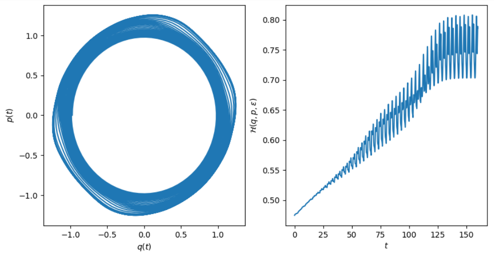


- **Left**: reference solution. **Right**: we learn $\mathcal{H}_\theta$ solved with Verlet scheme.

Numerical examples

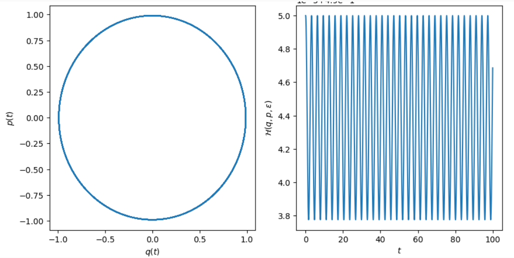
- We learn the oscillator system $H(q,p) = \frac{1}{2}(p^2 + q^2)$. Data generated with $\Delta t = 1e^{-3}$ and $T = 20$
- We solve the learned systems with $\Delta t = 8e^{-2}$ and $T = 160$

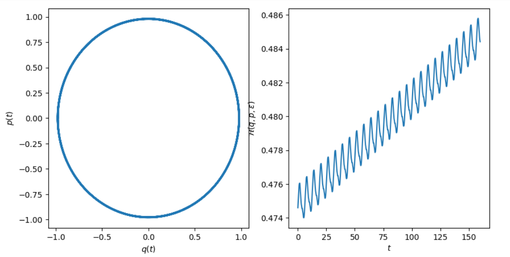




- **Left**: reference solution. **Right**: we learn $\mathbf{f}_\theta$ solved with RK4.

- We learn the oscillator system $H(q,p) = \frac{1}{2}(p^2 + q^2)$. Data generated with $\Delta t = 1e^{-3}$ and $T = 20$
- We solve the learned systems with $\Delta t = 8e^{-2}$ and $T = 160$

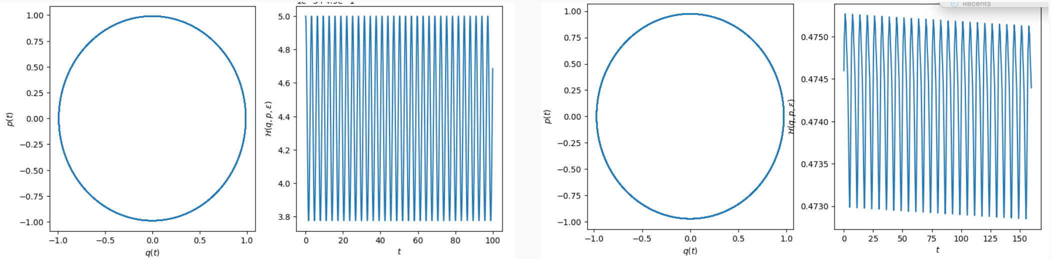




- **Left**: reference solution. **Right**: we learn $\mathcal{H}_\theta$ solved with RK4.

Numerical examples

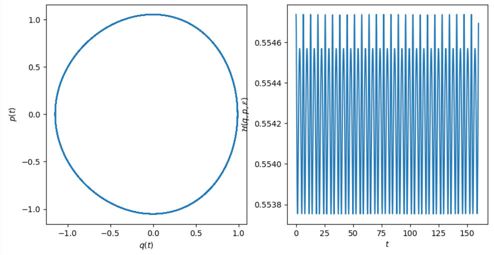
- We learn the oscillator system $H(q,p) = \frac{1}{2}(p^2 + q^2)$. Data generated with $\Delta t = 1e^{-3}$ and $T = 20$
- We solve the learned systems with $\Delta t = 8e^{-2}$ and $T = 160$

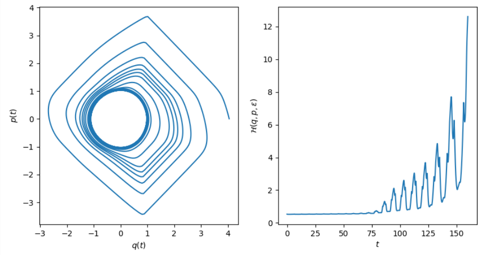


- **Left**: reference solution. **Right**: we learn $\mathcal{H}_\theta$ solved with Verlet scheme.

Numerical examples

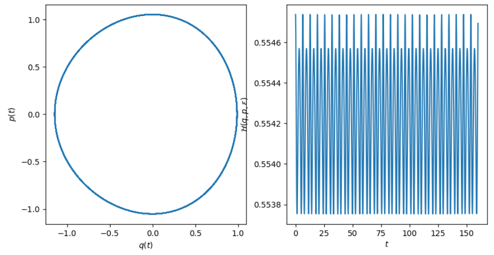
- We learn the oscillator system $H(q,p) = \frac{1}{2}(p^2 + q^2 + 0.12\frac{1}{3}q^3)$. Data generated with $\Delta t = 1e^{-3}$ and $T = 20$
- We solve the learned systems with $\Delta t = 8e^{-2}$ and $T = 160$

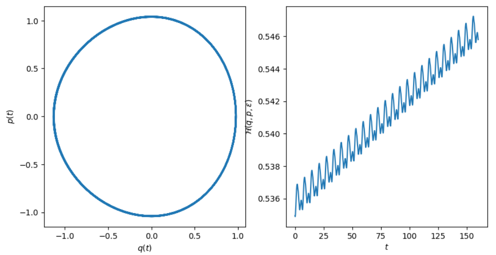




- **Left**: reference solution. **Right**: we learn $\mathbf{f}_\theta$ solved with RK4.

- We learn the oscillator system $H(q,p) = \frac{1}{2}(p^2 + q^2 + 0.12\frac{1}{3}q^3)$. Data generated with $\Delta t = 1e^{-3}$ and $T = 20$
- We solve the learned systems with $\Delta t = 8e^{-2}$ and $T = 160$

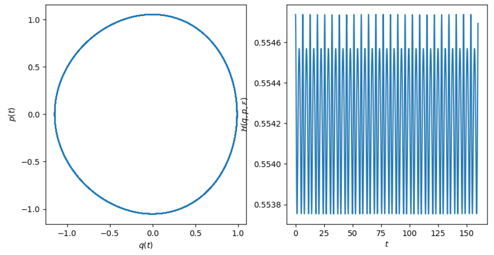


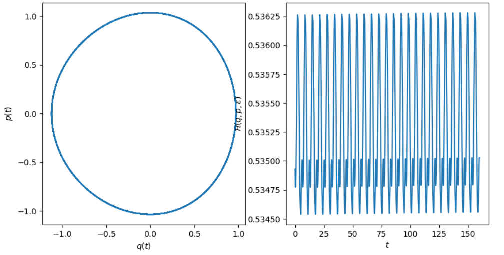


- **Left**: reference solution. **Right**: we learn $\mathcal{H}_\theta$ solved with RK4.

Numerical examples

- We learn the oscillator system $H(q,p) = \frac{1}{2}(p^2 + q^2 + 0.12\frac{1}{3}q^3)$. Data generated with $\Delta t = 1e^{-3}$ and $T = 20$
- We solve the learned systems with $\Delta t = 8e^{-2}$ and $T = 160$

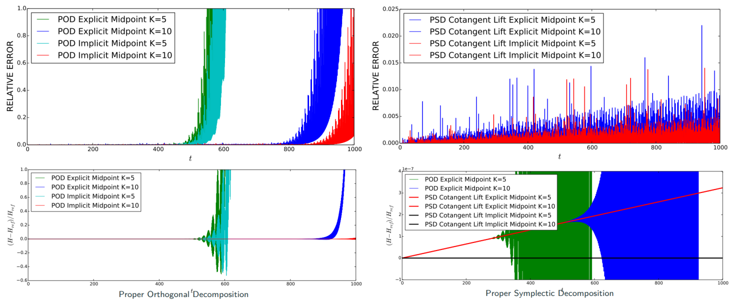




- **Left**: reference solution. **Right**: we learn $\mathcal{H}_\theta$ solved with Verlet scheme.
- Symplecticity error: $O(T\Delta t^4 + T\epsilon_{training})$ where we learn $\mathbf{f}_\theta$, $O(T\Delta t^4)$ where we learn $\mathcal{H}_\theta$ solved with RK4 and 0 where it is solved with Verlet.

Reduced modeling

- Reduced modeling
  not preserving/preserving the structure for Vlasov equation (M. Kraus simulation)



- It is important to preserve the structure of the PDE in the reduced modeling process.
- In this talk we focus on ODE structure preserving learning. The goal is to apply this to learn the dynamic on the latent space.
- Work of G. Steimer on reduction (nice talk but ... yesterday).

Conservative ODE and structure preserving numerical methods

Objective

Structure preserving learning

Learning of canonical Hamiltonian systems

Learning of non-canonical Hamiltonian systems

Non canonical Hamiltonian systems

We assume that $\mathbf{x} \in U \subset \mathbb{R}^{2N}$. A non canonical equation is on the form:

$$\dot{\mathbf{x}} = \mathcal{K}(\mathbf{x})^{-1}\nabla_{\mathbf{x}}\mathcal{H}(\mathbf{x})$$

with $\mathcal{K}(\mathbf{x})$ a invertible skew-symetric matrix satisfying the Jacobi identity:

$$\sum_{l=1}^{n}\left(\frac{\partial \mathcal{K}_{ij}(\mathbf{x})}{\partial x_l}\mathcal{K}_{lk}(\mathbf{x}) + \frac{\partial \mathcal{K}_{jk}(\mathbf{x})}{\partial x_l}\mathcal{K}_{li}(\mathbf{x}) + \frac{\partial \mathcal{K}_{ki}(\mathbf{x})}{\partial x_l}\mathcal{K}_{lj}(\mathbf{x})\right) = 0$$

The flow is *K-symplectic* if $(\nabla_{\mathbf{x}}\varphi)^t(\mathbf{x})\mathcal{K}(\varphi(\mathbf{x}))(\nabla_{\mathbf{x}}\varphi) = \mathcal{K}(\mathbf{x})$.

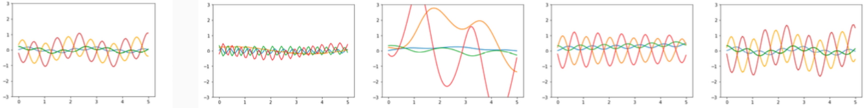
- More general modeling. How learn these type of systems ?
  - We learn $\mathcal{H}_\theta$ and a skew-symmetric matrix $\mathcal{K}_\theta$. Jacobi identity is not verified.
  - We learn $\mathcal{H}_\theta$ and a skew-symmetric matrix $\mathcal{K}_\theta$ with Jacobi identity is verified choosing

$$\mathcal{K}_{i,j}(\mathbf{x}) = \frac{\partial \mathcal{V}_{i,\theta}(\mathbf{x})}{\partial x_j} - \frac{\partial \mathcal{V}_{j,\theta}(\mathbf{x})}{\partial x_i}$$

- **Reference**: Y. Chen and al *Neural Symplectic Form: Learning Hamiltonian Equations on General Coordinate Systems*. 2021

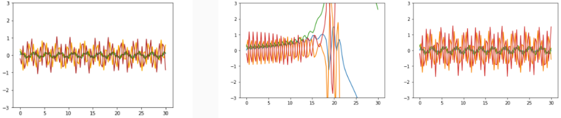
Non canonical Hamiltonian learning

- Example given in the reference. **Example**: double pendulum
- Short time, different learning.



(a) Ground truth (c) HNN (d) LNN (e) Skew Matrix Learning (f) Neural Symplectic Form

- Long time



(a) Ground truth (d) Skew Matrix Learning (e) Neural Symplectic Form

- Work well but in this work they use RK4 scheme.
- It will be generate problem for long time simulations.

Lagrangian formulation and variational integrator

- For $\mathcal{V}_\theta = [0, \vartheta_\theta]$, the non canonical Hamiltonian system can be rewrite as Lagrangian system with

$$\mathcal{L}(\mathbf{q},\mathbf{p},\dot{\mathbf{q}}) = \sum_{j=1}^{n} \vartheta_{j,\theta}(\mathbf{q},\mathbf{p}) \cdot \mathbf{q}^j - \mathcal{H}(\mathbf{q},\mathbf{p})$$

with $\mathbf{x} = (\mathbf{q},\mathbf{p})^t$. We speak about properly degenerate Lagrangian. We obtain the following system:

$$\begin{cases} \dot{\mathbf{q}} = \left(D_{\mathbf{p}}\vartheta\right)^{-1} \nabla_{\mathbf{p}}\mathcal{H}, \\ \dot{\mathbf{p}} = \left(D_{\mathbf{q}}\vartheta\right)^{-\mathrm{T}} \left(\left(D_{\mathbf{q}}\vartheta - D_{\mathbf{q}}\vartheta^{\top}\right)\left(D_{\mathbf{p}}\vartheta\right)^{-1}\nabla_{\mathbf{p}}\mathcal{H} - \nabla_{\mathbf{q}}\mathcal{H}\right) \end{cases}$$

- Allows to use specific time integrator called Discrete Variational Integrator (order 1).
- Variational integrators are numerical schemes based on a discrete Lagrangian

$$\begin{cases} \mathbf{z}_{n+1} = \vartheta(\mathbf{q}_n,\mathbf{p}_n) + D_{\mathbf{q}}\vartheta(\mathbf{q}_n,\mathbf{p}_n)^{\top}(\mathbf{q}_n - \mathbf{q}_{n-1}) - \Delta t \nabla_{\mathbf{q}}\mathcal{H}(\mathbf{q}_n,\mathbf{p}_n), \\ \mathbf{q}_{n+1} = \mathbf{q}_n + \Delta t \left(D_{\mathbf{p}}\vartheta(\mathbf{q}_{n+1},\mathbf{p}_{n+1})\right)^{-\mathrm{T}} \nabla_{\mathbf{p}}\mathcal{H}(\mathbf{q}_{n+1},\mathbf{p}_{n+1}), \\ \mathbf{p}_{n+1} = \vartheta^{-1}(\mathbf{q}_{n+1},\mathbf{z}_{n+1}). \end{cases}$$

Fails of non-canonical learning

- Like for the non Canonical Hamiltonian model we learn $\mathcal{H}_\theta$ and ϑ_θ minimizing:

$$\min_\theta \sum_{i=1}^{n} \sum_{t=1}^{T-1} (J_{1,i,t}(\theta) + J_{2,i,t}(\theta))$$

with

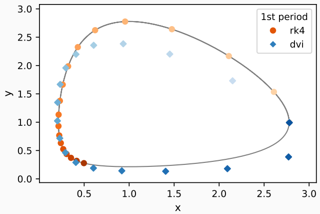
$$J_{1,i,t}(\theta) = \| \dot{\mathbf{q}}_t^i - (D_{\mathbf{p}}\vartheta_\theta)^{-1} \nabla_{\mathbf{p}} \mathcal{H}_\theta(\mathbf{q}_t^i, \mathbf{p}_t^i), \|_2^2$$

$$J_{2,i,t}(\theta) = \| \dot{\mathbf{p}}_t^i - (D_{\mathbf{q}}\vartheta_\theta)^{-\mathrm{T}} (D_{\mathbf{q}}\vartheta_\theta - D_{\mathbf{q}}\vartheta_\theta^\top)(D_{\mathbf{p}}\vartheta)^{-1} \nabla_{\mathbf{p}} \mathcal{H}(\mathbf{q}_t^i, \mathbf{p}_t^i) - \mathcal{H}(\mathbf{q}_t^i, \mathbf{p}_t^i) \|_2^2$$

- Example: Lokta Voltera

$$\vartheta(q,p) = -\ln(p)/q,$$

$$\mathcal{H}(q,p) = q + p - \ln(q) - \ln(p)$$



- Reference solution. Short time. Large time step.

- The model use Jacobian inverse of ϑ, the scheme use the local invert of ϑ.
- The scheme is sensitive to perturbations on ϑ, even if they don't impact the EDO.

Fails of non-canonical learning

- Like for the non Canonical Hamiltonian model we learn $\mathcal{H}_\theta$ and ϑ_θ minimizing:

$$\min_\theta \sum_{i=1}^{n} \sum_{t=1}^{T-1} \left(J_{1,i,t}(\theta) + J_{2,i,t}(\theta)\right)$$

with

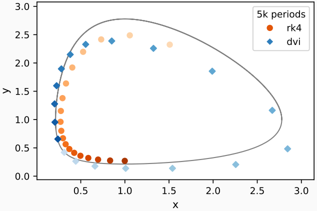
$$J_{1,i,t}(\theta) = \| \, \dot{\mathbf{q}}_t^i - \left(D_{\mathbf{p}}\vartheta_\theta\right)^{-1} \nabla_{\mathbf{p}} \mathcal{H}_\theta(\mathbf{q}_t^i, \mathbf{p}_t^i), \|_2^2$$

$$J_{2,i,t}(\theta) = \| \, \dot{\mathbf{p}}_t^i - \left(D_{\mathbf{q}}\vartheta_\theta\right)^{-\mathrm{T}} \left(D_{\mathbf{q}}\vartheta_\theta - D_{\mathbf{q}}\vartheta_\theta^{\top}\right) \left(D_{\mathbf{p}}\vartheta\right)^{-1} \nabla_{\mathbf{p}} \mathcal{H}(\mathbf{q}_t^i, \mathbf{p}_t^i) - \mathcal{H}(\mathbf{q}_t^i, \mathbf{p}_t^i) \, \|_2^2$$

- Example: Lokta Voltera

$$\vartheta(q,p) = -\ln(p)/q,$$

$$\mathcal{H}(q,p) = q + p - \ln(q) - \ln(p)$$



- Reference solution. Long time. Large time step.

- The model use Jacobian inverse of ϑ, the scheme use the local invert of ϑ.
- The scheme is sensitive to perturbations on ϑ, even if they don't impact the EDO.

Fails of non-canonical learning

- Like for the non Canonical Hamiltonian model we learn $\mathcal{H}_\theta$ and ϑ_θ minimizing:

$$\min_\theta \sum_{i=1}^{n} \sum_{t=1}^{T-1} (J_{1,i,t}(\theta) + J_{2,i,t}(\theta))$$

with

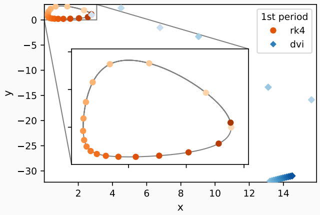
$$J_{1,i,t}(\theta) = \| \dot{\mathbf{q}}_t^i - (D_\mathbf{p}\vartheta_\theta)^{-1} \nabla_\mathbf{p} \mathcal{H}_\theta(\mathbf{q}_t^i, \mathbf{p}_t^i), \|_2^2$$

$$J_{2,i,t}(\theta) = \| \dot{\mathbf{p}}_t^i - (D_\mathbf{q}\vartheta_\theta)^{-\mathrm{T}} (D_\mathbf{q}\vartheta_\theta - D_\mathbf{q}\vartheta_\theta^\top)(D_\mathbf{p}\vartheta)^{-1} \nabla_\mathbf{p} \mathcal{H}(\mathbf{q}_t^i, \mathbf{p}_t^i) - \mathcal{H}(\mathbf{q}_t^i, \mathbf{p}_t^i) \|_2^2$$

- Example: Lokta Voltera

$$\vartheta(q,p) = -\ln(p)/q,$$

$$\mathcal{H}(q,p) = q+p-\ln(q)-\ln(p)$$



- Learning model. Short time. Large time step.

- The model use Jacobian inverse of ϑ, the scheme use the local invert of ϑ.
- The scheme is sensitive to perturbations on ϑ, even if they don't impact the EDO.

Fails of non-canonical learning

- Like for the non Canonical Hamiltonian model we learn $\mathcal{H}_\theta$ and ϑ_θ minimizing:

$$\min_\theta \sum_{i=1}^{n} \sum_{t=1}^{T-1} (J_{1,i,t}(\theta) + J_{2,i,t}(\theta))$$

with

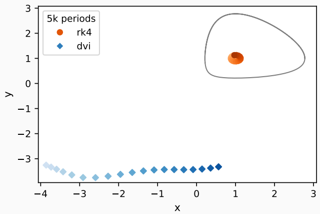
$$J_{1,i,t}(\theta) = \parallel \dot{\mathbf{q}}_t^i - \left(D_{\mathbf{p}}\vartheta_\theta\right)^{-1} \nabla_{\mathbf{p}}\mathcal{H}_\theta(\mathbf{q}_t^i, \mathbf{p}_t^i), \parallel_2^2$$

$$J_{2,i,t}(\theta) = \parallel \dot{\mathbf{p}}_t^i - \left(D_{\mathbf{q}}\vartheta_\theta\right)^{-\mathrm{T}} \left(D_{\mathbf{q}}\vartheta_\theta - D_{\mathbf{q}}\vartheta_\theta^\top\right)\left(D_{\mathbf{p}}\vartheta\right)^{-1} \nabla_{\mathbf{p}}\mathcal{H}(\mathbf{q}_t^i, \mathbf{p}_t^i) - \mathcal{H}(\mathbf{q}_t^i, \mathbf{p}_t^i) \parallel_2^2$$

- Example: Lokta Voltera

$$\vartheta(q,p) = -\ln(p)/q,$$

$$\mathcal{H}(q,p) = q + p - \ln(q) - \ln(p)$$



- Learning model. Long time. Large time step.

- The model use Jacobian inverse of ϑ, the scheme use the local invert of ϑ.
- The scheme is sensitive to perturbations on ϑ, even if they don't impact the EDO.

First solution: Discretization informed regularization

- We consider the solution $(\mathbf{q}, \dot{\mathbf{q}})$. By an analysis of the local error we obtain:

$$Dvi((\mathbf{q}, \dot{\mathbf{q}})) = \Delta t^2 R((\mathbf{q}, \dot{\mathbf{q}})) + O(\Delta t^3)$$

- We learn $\mathcal{H}_\theta$ and ϑ_θ minimizing:

$$\min_\theta \sum_{i=1}^{n} \sum_{t=1}^{T-1} (J_{1,t,i}(\theta) + J_{2,t,i}(\theta) + Reg_{i,t}(\theta))$$

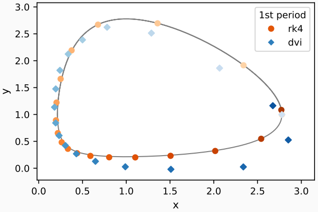
with

$$Reg_{i,t}(\theta) = \parallel R(\mathbf{q}_i^t, \dot{\mathbf{q}}_i^t) \parallel_2^2$$

- Example: Lokta Voltera

$$\vartheta(q,p) = -\ln(p)/q,$$

$$\mathcal{H}(q,p) = q + p - \ln(q) - \ln(p)$$



- Learning model with regularization. Short time. Large time step.

First solution: Discretization informed regularization

- We consider the solution $(\mathbf{q}, \dot{\mathbf{q}})$. By an analysis of the local error we obtain:

$$Dvi((\mathbf{q}, \dot{\mathbf{q}})) = \Delta t^2 R((\mathbf{q}, \dot{\mathbf{q}})) + O(\Delta t^3)$$

- We learn $\mathcal{H}_\theta$ and ϑ_θ minimizing:

$$\min_\theta \sum_{i=1}^{n} \sum_{t=1}^{T-1} (J_{1,t,i}(\theta) + J_{2,t,i}(\theta) + Reg_{i,t}(\theta))$$

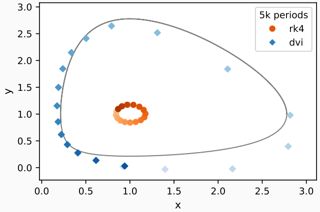
with

$$Reg_{i,t}(\theta) = \parallel R(\mathbf{q}_i^t, \dot{\mathbf{q}}_i^t) \parallel_2^2$$

- Example: Lokta Voltera

$$\vartheta(q,p) = -\ln(p)/q,$$

$$\mathcal{H}(q,p) = q + p - \ln(q) - \ln(p)$$



- Learning model with regularization. Long time. Large time step.

Second solution: Learning with the numerical integrator

- Since we will use the DVI, why not learn direct with the DVI.

$$\min_\theta \sum_{i=1}^{n}\sum_{t=1}^{T-1}\left(J_{1,i,t}(\theta) + Reg(\theta)\right)$$

with

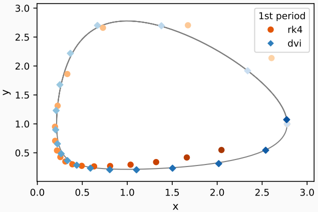
$$J_{1,i,t}(\theta) = \| DVi((\mathbf{q}_i^t, \dot{\mathbf{q}}_i^t) \|_2^2$$

- The regularization term penalize the conditioning of the matrix inverted in the scheme.

    - Example: Lokta Voltera

$$\vartheta(q,p) = -\ln(p)/q,$$

$$\mathcal{H}(q,p) = q + p - \ln(q) - \ln(p)$$



- Learning model. Short time. Large time step.

- We learn the modified ϑ associated to the DVI. Large error reducing if you fixe Δt

Second solution: Learning with the numerical integrator

- Since we will use the DVI, why not learn direct with the DVI.

$$\min_\theta \sum_{i=1}^{n} \sum_{t=1}^{T-1} (J_{1,i,t}(\theta) + Reg(\theta))$$

with

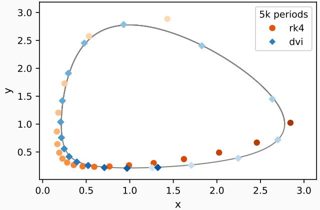
$$J_{1,i,t}(\theta) = \| DVi((\mathbf{q}_i^t, \dot{\mathbf{q}}_i^t) \|_2^2$$

- The regularization term penalize the conditioning of the matrix inverted in the scheme.

  - Example: Lokta Voltera

$$\vartheta(q,p) = -\ln(p)/q,$$

$$\mathcal{H}(q,p) = q + p - \ln(q) - \ln(p)$$



- Learning model. Long time. Large time step.

- We learn the modified ϑ associated to the DVI. Large error reducing if you fixe Δt

Guiding center

Asymptotic model of plasma physics in tokamaks with a strong magnetic field.

- The position is expressed in poloidal-toroidal coordinates $X = (r, \theta, \phi)$.
- r is the minor-radial position, θ the geometric poloidal angle, and ϕ the geometric toroidal angle.
- The momentum is reduced to a single coordinate u in the toroidal direction, parallel to the magnetic field.

- Lagrangian:

$$L(\theta, \phi, r, u, \dot{\theta}, \dot{\phi}) = A_\theta(r, \theta)\dot{\theta} + (A_\phi(r) + u(R_0 + r\cos(\theta)))\dot{\phi} - H(r, \theta, u).$$

where $A = (0, A_\theta, A_\phi)$ is a magnetic potential associated to $\mathbf{B} = \nabla_X \times A$. It is given by

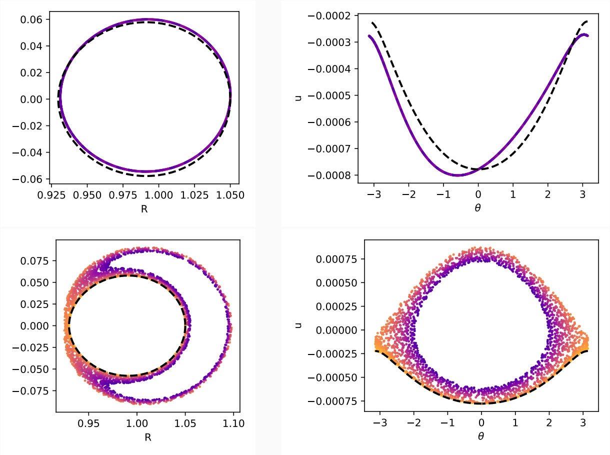
$$A_\theta(r, \theta) = \frac{B_0 R_0^2}{\cos^2(\theta)}\left(\frac{r\cos(\theta)}{R_0} - \log\left(1 + \frac{r\cos(\theta)}{R_0}\right)\right), \quad A_\phi(r) = -\frac{B_0 r^2}{2q_0}$$

and

$$H(r, \theta, u) = \frac{1}{2}u^2 + \mu B(r, \theta), \quad B(r, \theta) = \frac{B_0}{1 + \frac{r\cos(\theta)}{R_0}}\sqrt{1 + \left(\frac{r}{q_0 R_0}\right)^2}$$

Numerical results for Guiding center

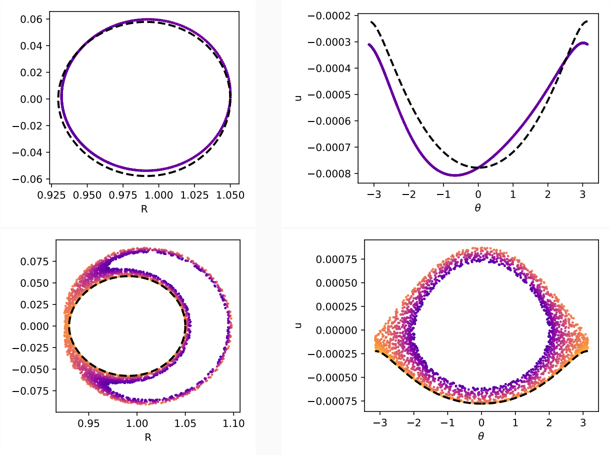
- Long time simulation



- Exact model with DVI (top), with RK4 (bottom)

Numerical results for Guiding center

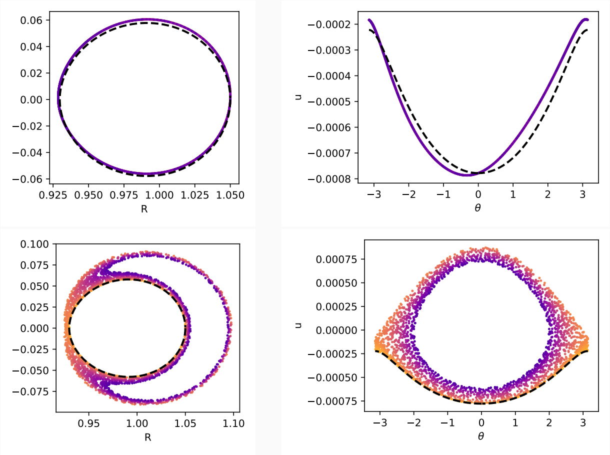
- Long time simulation



- Learned model without regularization with DVI (top) and with RK4 (bottom)

Numerical results for Guiding center

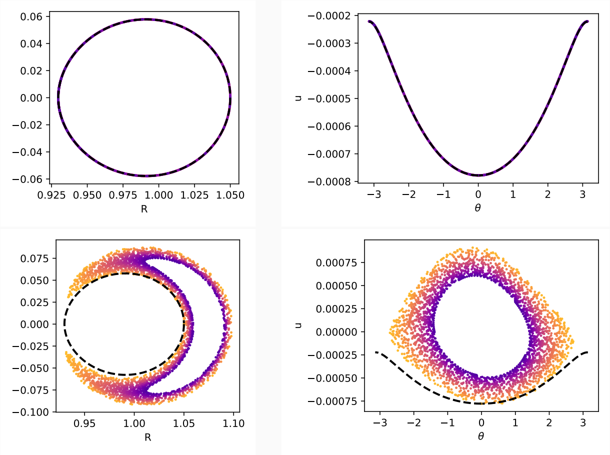
- Long time simulation



- Learned model with regularization with DVI (top) and with RK4 (bottom)

Numerical results for Guiding center

- Long time simulation



- Learned model with the scheme with DVI (top) and with RK4 (bottom)

Conclusion and perspectives

Conclusion

- Learn non-canonical systems preserving the structure gives better results.
- For long time simulations we need structure preserving scheme.
- The DVI scheme not work directly on the learned model.
- To solve the problem we introduce effect on the DVI scheme in the loss.
  - We penalize the first term in the local error of the scheme. Gives a good approximation of the model but need good reconstruction on the data derivative.
  - We learn with the scheme. We learn the modified model associated to the scheme. It is more accurate since we correct the error of the scheme. Cannot use with other scheme or too different time steps.

Perspectives

Learn Poisson system

$$\dot{\mathbf{x}} = \mathcal{B}(\mathbf{x})\nabla_{\mathbf{x}}\mathcal{H}(\mathbf{x})$$

where $\mathcal{B}(\mathbf{x})$ can be degenerated.