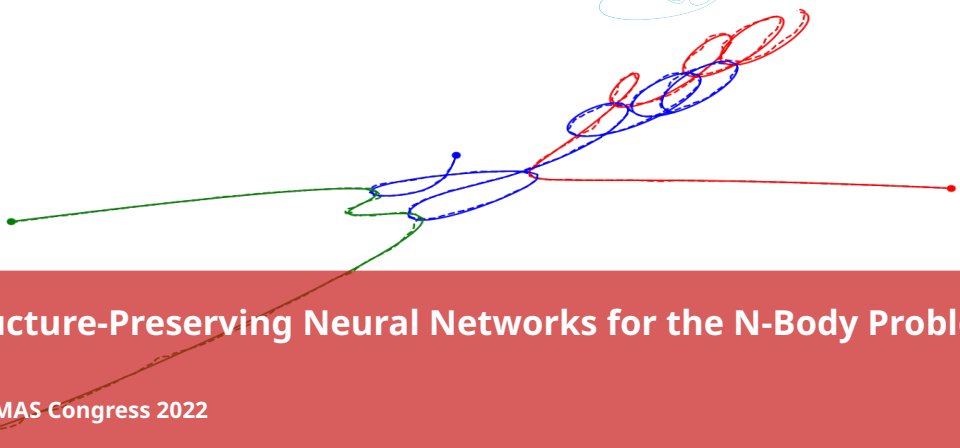


Structure-Preserving Neural Networks for the N-Body Problem

ECCOMAS Congress 2022

Philipp Horn and Barry Koren (Eindhoven University of Technology)
Veronica Saz Ulibarrena and Simon Portegies Zwart (Leiden Observatory)

Mathematics and Computer Science

Motivation to use NNs for the N-body problem

- 3-body problem is chaotic, without an analytical solution
- Requires small time steps to solve it numerically
- Main bottleneck in large *N*-body simulations
- An NN surrogate could speed up calculations

Motivation to use NNs for the N-body problem

- 3-body problem is chaotic, without an analytical solution
- Requires small time steps to solve it numerically
- Main bottleneck in large *N*-body simulations
- An NN surrogate could speed up calculations

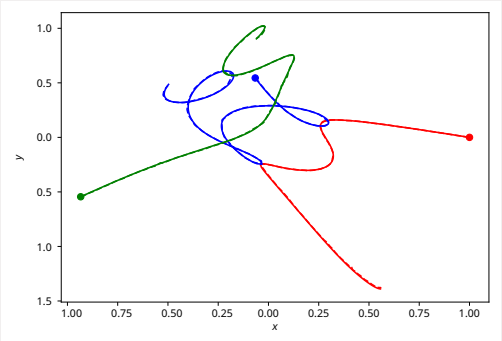


Figure: First proof of concept [1]

[1] P.G. Breen *et al.*, Newton versus the machine: solving the chaotic three-body problem using deep neural networks, *Monthly Notices of the Royal Astronomical Society*, Vol. **494**, pp. 2465-2470, 2020.

What is a Hamiltonian System?

The ODE describing a Hamiltonian system can be written as:

$$\begin{aligned} \begin{pmatrix} \dot{p} \\ \dot{q} \end{pmatrix} &= \mathbb{J} \nabla H(p;q); \quad \text{with } \mathbb{J} = \begin{pmatrix} 0 & I_d \\ I_d & 0 \end{pmatrix} \\ &= \begin{pmatrix} \nabla_q H(p;q) \\ \nabla_p H(p;q) \end{pmatrix}; \quad p;q \in \mathbb{R}^d: \end{aligned} \tag{1}$$

What is a Hamiltonian System?

The ODE describing a Hamiltonian system can be written as:

$$\begin{matrix} \dot{p} \\ \dot{q} \end{matrix} = \mathbb{J} \nabla H(p;q), \quad \text{with } \mathbb{J} = \begin{matrix} 0 & I_d \\ I_d & 0 \end{matrix} \tag{1}$$

$$= \begin{matrix} \nabla_q H(p;q) \\ \nabla_p H(p;q) \end{matrix}; \quad p;q \in \mathbb{R}^d.$$

In some Hamiltonian systems the Hamiltonian $H : \mathbb{R}^d \times \mathbb{R}^d \to \mathbb{R}$ is separable, meaning:

$$H(p;q) = T(p) + U(q). \tag{2}$$

Then, $T : \mathbb{R}^d \to \mathbb{R}$ is referred to as the kinetic energy and $U : \mathbb{R}^d \to \mathbb{R}$ as the potential energy.

Two key properties of Hamiltonian systems

1. Total energy is conserved along trajectories:

$$H(p_0; q_0) = H(p(t); q(t)); \quad t \in \mathbb{R}_+ \tag{3}$$

where $(p(t); q(t))^T$ is the solution of ODE (1) with initial condition $(p_0; q_0)^T$

Two key properties of Hamiltonian systems

1. Total energy is conserved along trajectories:

$$H(p_0; q_0) = H(p(t); q(t)); \quad t \in \mathbb{R}_+ \tag{3}$$

where $(p(t); q(t))^T$ is the solution of ODE (1) with initial condition $(p_0; q_0)^T$

2. Symplecticity of the flow map $\varphi_h : x(t) \mapsto x(t + h)$ meaning:

$$\frac{\partial \varphi_h}{\partial x}^T J \frac{\partial \varphi_h}{\partial x} = J; \quad x = \begin{matrix} p \\ q \end{matrix} \in \mathbb{R}^{2d}: \tag{4}$$

How to solve a Hamiltonian system

Classically one knows H and, given the initial conditions $(p_0; q_0)^T$, the objective is to calculate trajectories $(p(t); q(t))^T$ or only $(p(t_{\text{end}}); q(t_{\text{end}}))^T$ at a final time t_{end}. This is done using numerical integrators.

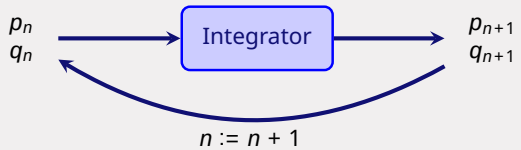


Figure: Iterative scheme to solve an ODE using a numerical integrator.

Semi-implicit (symplectic) Euler method

The symplectic Euler method for Hamiltonian systems given p_n and q_n is:

$$p_{n+1} = p_n \quad h\, r\, {}_q H(p_{n+1}; q_n)$$
$$q_{n+1} = q_n + h\, r\, {}_p H(p_{n+1}; q_n)$$

Semi-implicit (symplectic) Euler method

The symplectic Euler method for Hamiltonian systems given p_n and q_n is:

$$\begin{aligned} p_{n+1} &= p_n - h \nabla_q H(p_{n+1}, q_n) \\ q_{n+1} &= q_n + h \nabla_p H(p_{n+1}, q_n) \end{aligned}$$

For separable Hamiltonian systems it becomes an explicit method:

$$x_n := \begin{pmatrix} p_n \\ q_n \end{pmatrix} \to \begin{pmatrix} p_n \\ q_n \end{pmatrix} - \nabla U(q_n) h =: \begin{pmatrix} p_{n+1} \\ q_n \end{pmatrix} \to \begin{pmatrix} p_{n+1} \\ q_n + \nabla T(p_{n+1}) h \end{pmatrix} =: \begin{pmatrix} p_{n+1} \\ q_{n+1} \end{pmatrix} =: x_{n+1} \tag{5}$$

How to solve a Hamiltonian system using NNs

To solve a Hamiltonian system without knowledge of H but from data with the use of NNs there are two different approaches. The first simply replaces the integrator with an NN. The second one does not work in an iterative manor. Instead it also receives a time t as input and predicts $(p(t); q(t))^T$ directly from $(p_0; q_0)^T$.

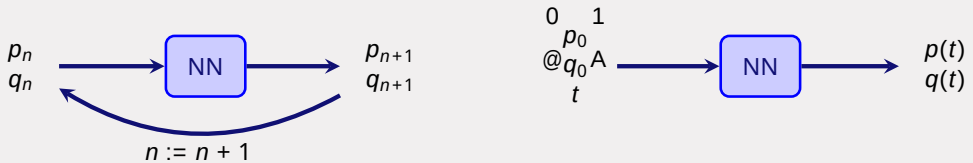
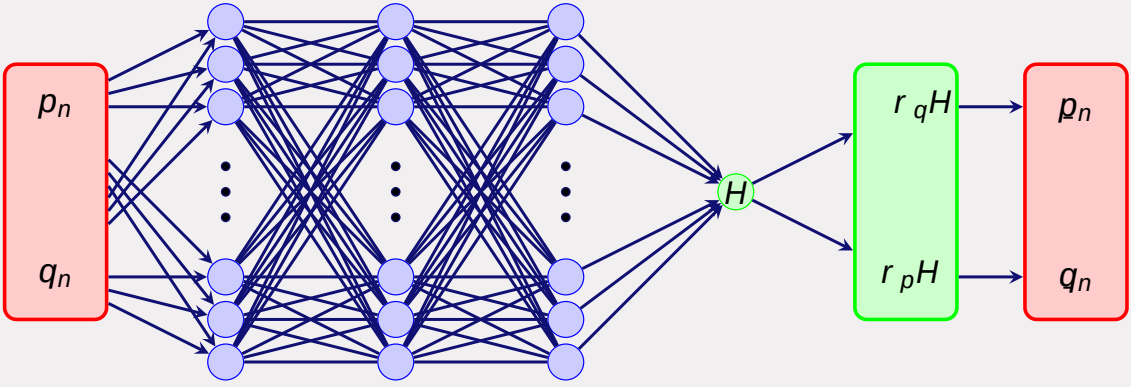


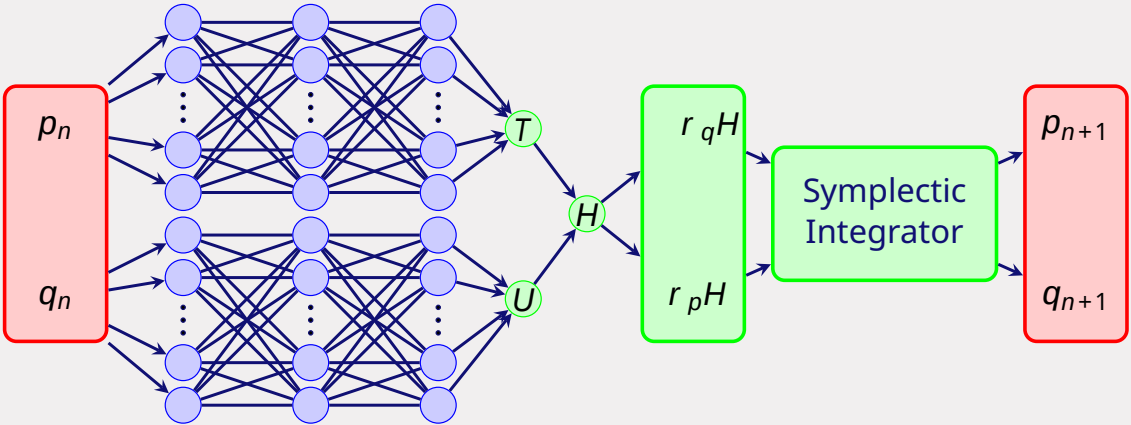
Figure: Iterative scheme to solve an ODE using an NN compared to a scheme requiring t as an input.

Hamiltonian neural network by Greydanus *et al.* [2]



[2] S. Greydanus *et al.*, Hamiltonian neural networks, *Advances in Neural Information Processing Systems*, Vol. **32**, NeurIPS, 2019.

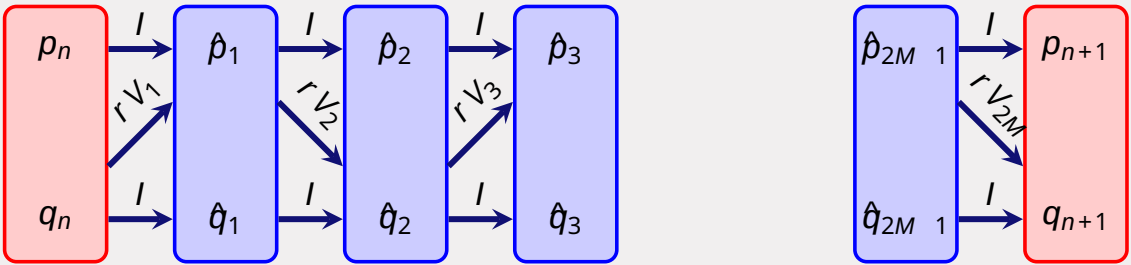
Hamiltonian neural network by Chen *et al.* [3]



[3] Z. Chen *et al.*, Symplectic recurrent neural networks, *8th International Conference on Learning Representations*, ICLR, 2020.

A standard SympNet by Jin *et al.* [4]

- Each layer: $\mathbb{R}^{2d} \to \mathbb{R}^{2d}$ and is a symplectic map
- A concatenation of symplectic maps is again symplectic $\Rightarrow$ A SympNet is a symplectic map.



[4] P. Jin *et al.*, SympNets: Intrinsic structure-preserving symplectic networks for identifying Hamiltonian systems, *Neural Networks*, Vol. **132**, pp. 166-179, 2020.

SympNets - Gradient layers

A gradient layer has the trainable parameters: $K \in \mathbb{R}^{n \times d}$, $b \in \mathbb{R}^n$ and $a \in \mathbb{R}^n$. Also, an activation function has to be chosen beforehand. Usually the sigmoid function $(x) = \frac{1}{1+e^{-x}}$ is used. With this (and a slight abuse of matrix vector multiplication notation) the upper and lower gradient layers are defined as:

$$G_{up} \begin{pmatrix} p \\ q \end{pmatrix} := \begin{pmatrix} I & \hat{}_{K,a,b} \\ 0 & I \end{pmatrix} \begin{pmatrix} p \\ q \end{pmatrix} = \begin{pmatrix} p + K^T \mathrm{diag}(a) \ (Kq + b) \\ q \end{pmatrix} \tag{6}$$

$$G_{low} \begin{pmatrix} p \\ q \end{pmatrix} := \begin{pmatrix} I & 0 \\ \hat{}_{K,a,b} & I \end{pmatrix} \begin{pmatrix} p \\ q \end{pmatrix} = \begin{pmatrix} p \\ q + K^T \mathrm{diag}(a) \ (Kp + b) \end{pmatrix} \tag{7}$$

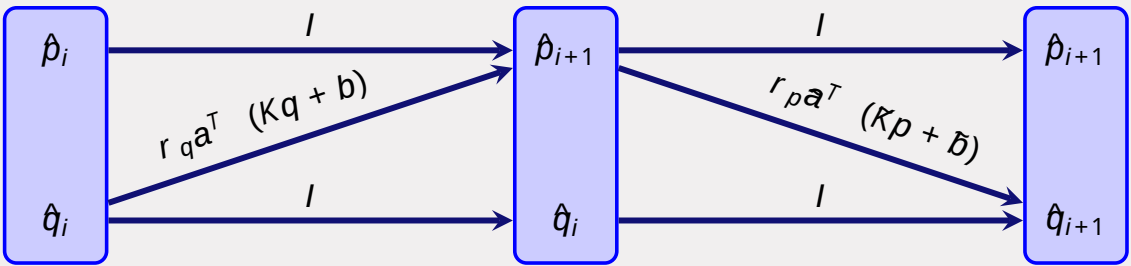
The name of this layer comes from the fact that $\hat{}_{K,a,b}$ can approximate any gradient of a function $V : \mathbb{R}^d \to \mathbb{R}$ (so any ∇V).

Two symplectic gradient layers in detail

Focusing on two alternating gradient layers and using

$$r_q a^T \ (Kq + b) = K^T \mathrm{diag}(a) \ (Kq + b) \tag{8}$$

$$r_p a^T \ (\bar{K}p + \bar{b}) = \bar{K}^T \mathrm{diag}(a) \ (\bar{K}p + \bar{b}) \tag{9}$$



Two gradient layers learn a Hamiltonian

Two layers, starting with an upper and followed by a lower gradient layer can be written as:

$$\begin{pmatrix} p_i \\ q_i \end{pmatrix} \mapsto \begin{pmatrix} p_i + r_q a^T (K q_i + b) \\ q_i \end{pmatrix} =: \begin{pmatrix} p_{i+1} \\ q_i \end{pmatrix} \mapsto \begin{pmatrix} p_{i+1} \\ q_i + r_p \tilde{a}^T (\tilde{K} p_{i+1} + \tilde{b}) \end{pmatrix} =: \begin{pmatrix} p_{i+1} \\ q_{i+1} \end{pmatrix} \tag{10}$$

Two gradient layers learn a Hamiltonian

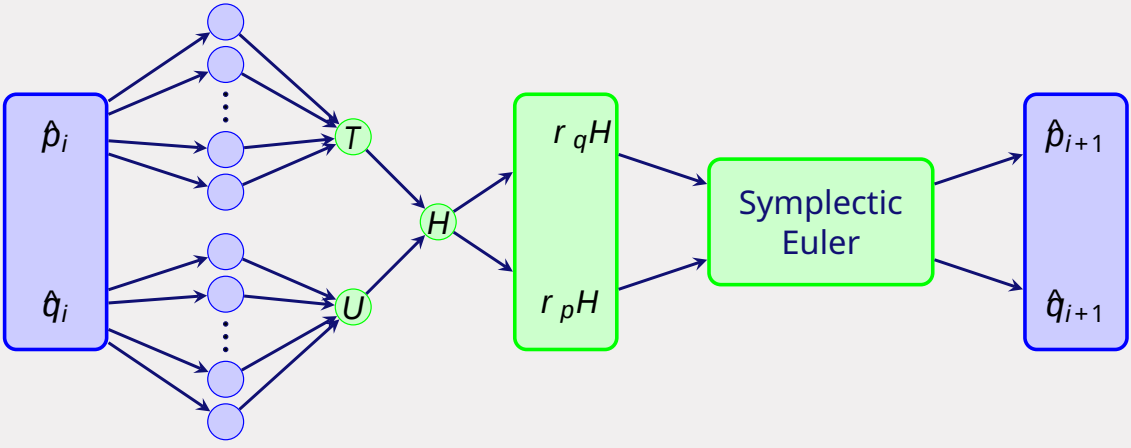
Two layers, starting with an upper and followed by a lower gradient layer can be written as:

$$\begin{pmatrix} p_i \\ q_i \end{pmatrix} \mapsto \begin{pmatrix} p_i + r_q a^T (K q_i + b) \\ q_i \end{pmatrix} =: \begin{pmatrix} p_{i+1} \\ q_i \end{pmatrix} \mapsto \begin{pmatrix} p_{i+1} \\ q_i + r_p \tilde{a}^T (\tilde{K} p_{i+1} + \tilde{b}) \end{pmatrix} =: \begin{pmatrix} p_{i+1} \\ q_{i+1} \end{pmatrix} \tag{10}$$

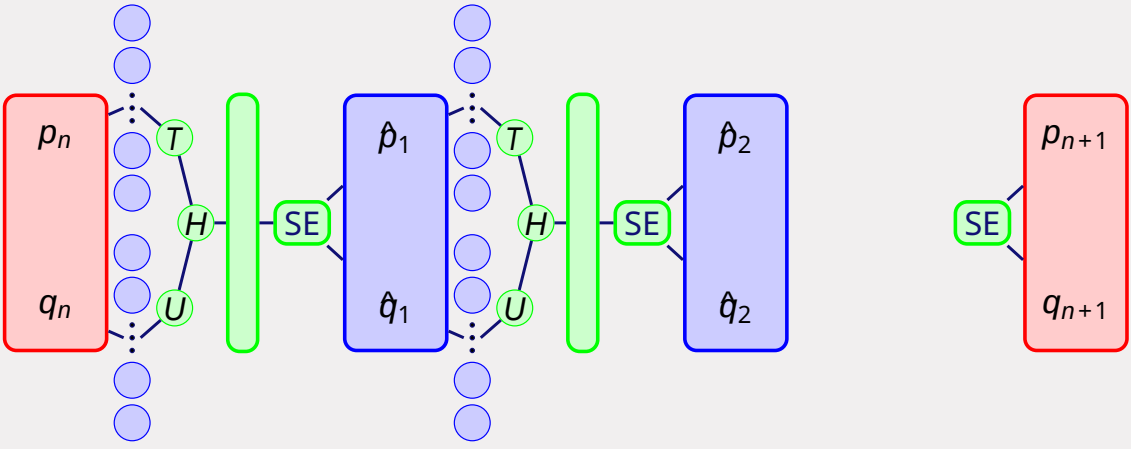
These are the updates of the symplectic Euler method for the Hamiltonian

$$H(p;q) = \underbrace{\frac{1}{h} \tilde{a}^T (\tilde{K} p + \tilde{b})}_{T(p)} \quad \underbrace{\frac{1}{h} a^T (K q + b)}_{U(q)} \tag{11}$$

Two gradient layers as an HNN



SympNet as multiple HNNs in series

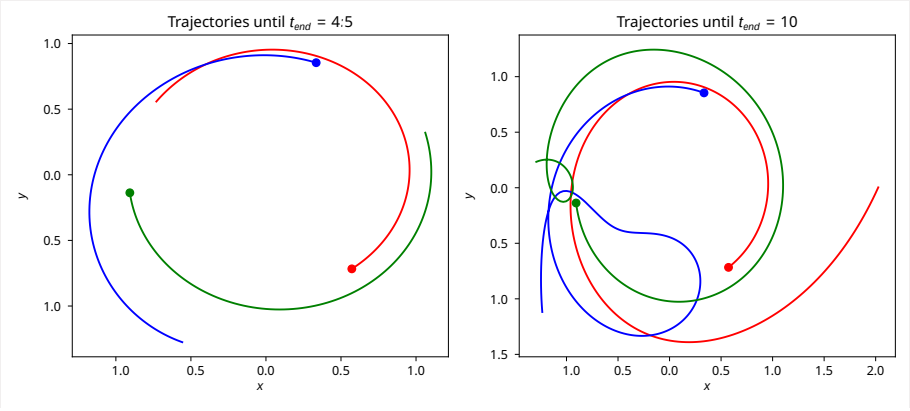


Data for first numerical experiment

- Three bodies with equal masses
- Disturbed circular orbit
- In the x-y-plane
- 5000 trajectories until $t_{\text{end}} = 10$

Data for first numerical experiment

- Three bodies with equal masses
- Disturbed circular orbit
- In the x-y-plane
- 5000 trajectories until $t_{\text{end}} = 10$



SympNets vs physics-unaware neural networks

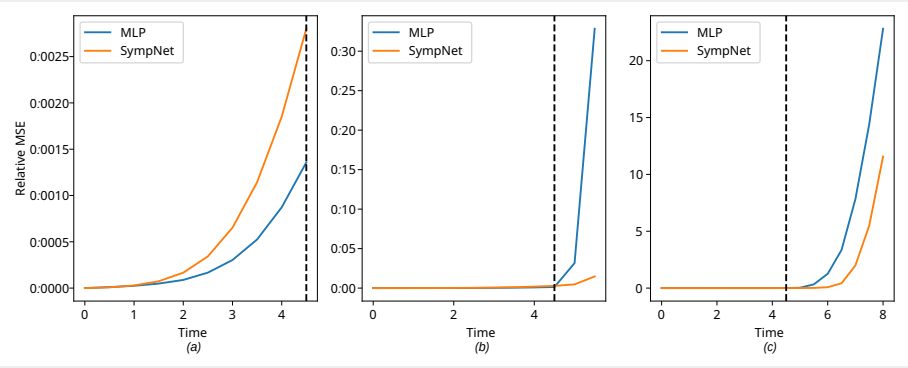


Figure: Relative mean square error (MSE) for a physics-unaware multilayer perceptron (MLP) and a SympNets trained on data until $t_{\text{end}} = 4{:}5$ (vertical line).

SympNets vs the symplectic Euler method

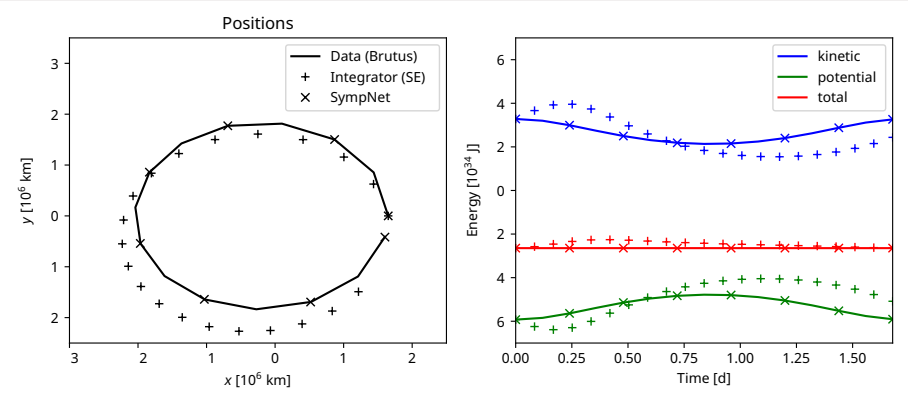


Figure: Trajectories of a planet orbiting a heavy central body. Predicted by the symplectic Euler method and a SympNet.

Conclusion

- We have shown a new connection between SympNets and HNNs
- SympNets generalize better than physics-unaware MLPs
- NNs for Hamiltonian systems can outperform numerical integrators

Conclusion

- We have shown a new connection between SympNets and HNNs
- SympNets generalize better than physics-unaware MLPs
- NNs for Hamiltonian systems can outperform numerical integrators

Outlook

- Improved SympNet topology to be more exact, also inside the range of training data
- Tackle the more chaotic 3-body problem using a non-iterative approach

Conclusion

- We have shown a new connection between SympNets and HNNs
- SympNets generalize better than physics-unaware MLPs
- NNs for Hamiltonian systems can outperform numerical integrators

Outlook

- Improved SympNet topology to be more exact, also inside the range of training data
- Tackle the more chaotic 3-body problem using a non-iterative approach

Thank you for your attention!