A Hamiltonian neural network model for the three-body problem

Sunniva Meltzer

Spring 2024

1 Introduction

Disclaimer The work presented here is the continuation of the previous coursework done in FYS9429 [1]. We are now extending the Hamiltonian system from a 1-dimensional harmonic oscillator, to a 3-dimensional closed space with three moving objects. This extension increases the complexity of the problem, and due to limited time this report only shows the preliminary results of the work. We here present results for a fully observed system, while the overall goal is to apply the partial observed methodology to the three-body problem as well.

Physics-informed machine learning concerns the integration of physical constraints in machine learning to achieve more robust and flexible models whose predictions are consistent with the underlying physics of the system. This hybrid approach aims to reduce the amount of data required to train a model by utilising that the mathematical framework constrains the solution space, and also improve the accuracy of the predictions due to the added domain knowledge.

In the real world data is a limited resource, and the quantity and quality of training data can be crucial when it comes to making a machine learning model that gives accurate predictions. This motivates hybrid approaches such as those utilised in this project, because the need for data just increases with increasing size and complexity of the machine learning models. The overall goal of this project is to train Hamiltonian neural networks to high accuracy on partially observed systems, meaning that one or more variables cannot be observed at certain points in time. This leaves "holes" in the training data where observations are missing. Previous work on the simple harmonic oscillator model showed that it was possible to train a model to high accuracy using as little as 10% of the momentum data observations, indicating that you only need a few reference points in momentum space for the Hamiltonian neural network to make accurate predictions. The question is whether these results can be generalised to higher dimensional problems with more than one moving object.

The aim of this project is to train a physics-informed neural network on the three-body problem. The three-body problem is a special case of the N-body problem, with three celestial objects moving around in a closed system under influence of no other forces than the gravitational force between the planets. The total energy of the system is therefore conserved. A big challenge of the three-body problem is that it has no general solution, and for most initial conditions the movement of the planets becomes chaotic (non-periodic) and unpredictable. There exists a few special cases of initial conditions that leads to periodic solutions, but in this project we only explored random initialisations.

2 Theoretical background

In general, the N-body problem concerns the movement of N celestial objects in a closed system, where the dynamics of the system can be described by Hamilton's equations,

$$\frac{d\mathbf{q}_i}{dt} = \frac{\partial \mathcal{H}}{\partial \mathbf{p}_i}, \qquad \frac{d\mathbf{p}_i}{dt} = -\frac{\partial \mathcal{H}}{\partial \mathbf{q}_i}. \tag{1}$$

Here, $\mathbf{q}_i$ and $\mathbf{p}_i$ denotes the position and momentum of object i, respectively, and the Hamiltonian operator $\mathcal{H}$ represents the total energy of the system. In practice, this turns into a $6N$-dimensional problem, as the differential equations above must be solved for each of the (x, y, x)-components for each of the N bodies. Equation (1) can also be expressed as $\dot{\mathbf{x}} = S\nabla\mathcal{H}$ where $\mathbf{x} = (\mathbf{q}, \mathbf{p})^T$ and S is the skew-symmetric matrix $S = \left(\begin{smallmatrix} 0 & I \\ -I & 0 \end{smallmatrix}\right)$.

In general, the total energy of the system can be expressed as the sum of kinetic and potential energies,

$$\mathcal{H} = K + U = \sum_{i=1}^{N} \frac{\mathbf{p}_i^2}{2m_i} - \sum_{i=1}^{N-1} \sum_{j>i}^{N} G \frac{m_i m_j}{\|\mathbf{q}_i - \mathbf{q}_j\|_2}, \tag{2}$$

where m_i is the mass of object i. This project only investigated the three-body problem, i.e. $N = 3$.

In Hamiltonian neural networks (HNNs), a neural network is used to model the Hamiltonian operator $\mathcal{H}$. Classically, one would find the evolution of a system by training a neural network to learn the mapping $\mathcal{NN}(\mathbf{q}, \mathbf{p}) \to (\dot{\mathbf{q}}, \dot{\mathbf{p}})$, and then use the time derivatives to predict the next point in time. However, for HNNs the output of the neural network is a single scalar quantity: the Hamiltonian. In other words, we have the mapping $\mathcal{NN}(\mathbf{q}, \mathbf{p}) \to \mathcal{H}_\theta$, and the time derivatives of $\mathbf{q}$ and $\mathbf{p}$ are found by using equation (1). The loss function of HNNs can be defined by using the same relation,

$$\mathcal{L} = \left\| \frac{\partial \mathbf{q}}{\partial t} - \frac{\partial \mathcal{H}_\theta}{\partial \mathbf{p}} \right\|_2 + \left\| \frac{\partial \mathbf{p}}{\partial t} + \frac{\partial \mathcal{H}_\theta}{\partial \mathbf{q}} \right\|_2, \tag{3}$$

which guarantees that the learned $\mathcal{H}_\theta$ is conserved. An important note here is that although the formulation above guarantees that the total energy of the *learned* system, $\mathcal{H}_\theta$, is conserved, this is not the same as saying that the network has learned the true Hamiltonian $\mathcal{H}$. As shown by Greydanus *et al*, a HNN will often preserve a quantity that is very close to the true energy, but shifted by a constant factor [2].

3 Methods

The methodology builds on the same approach as the previous work on the harmonic oscillator model (see [1]).

Since there exists no general solution describing the time-evolution of the three-body problem, the training data was generated numerically. In this process, three different integration methods were compared based on how well they conserved the total energy; the implicit midpoint method, the fourth order Runge-Kutta method and the dopri8 odeint solver from the `torchdiffeq` package (Runge-Kutta of order 8 of Dormand-Prince-Shampine) [3]. By analysing the total energy as a function of time for these three integrators, one could see that the total energy had big spikes when the planets came close to each other (see figure 1). These numerical errors became smaller if the integration step size was decreased, and we therefore had to evaluate a trade-off between accuracy and computation time. The implicit midpoint solver was chosen as the best integration method based on the size of the energy spikes, and a step size of $dt = 0.01$ was chosen to give fluctuations smaller than the standard deviation of the energy. The standard deviation of the energy was estimated by assuming a 1% measurement error in the $\mathbf{x}$ data, as it is reasonable to believe that real-world positions and momenta are not measured to higher accuracy than this. By generating noisy $(\mathbf{q}, \mathbf{p})$ samples where the noise was drawn from a zero-mean normal distribution scaled to 1% of $\mathbf{x}_0$, the standard deviation of the corresponding Hamiltonians was estimated to $\sigma_{\mathcal{H}} \approx 0.01$. For comparison, the numerical errors of the implicit midpoint was of order ~ 0.001 when using a time step $dt = 0.01$.

Although the errors of the numerical methods were small compared to the uncertainty in $\mathcal{H}$, a problem with chaotic systems is that even small errors grow fast and accumulate over time, and small changes in initial conditions can lead to big dynamical differences in the time evolution of the system. Since the potential energy scales as $\frac{1}{r}$, where $r = |\mathbf{q}_i - \mathbf{q}_j|$, it blows up and goes to infinity when the distance between two objects goes to zero. To count for this problem and avoid zero-division, the potential energy was softened by introducing a small constant ϵ,

$$U_\epsilon = -\sum_{i=1}^{N-1} \sum_{j>i}^{N} G \frac{m_i m_j}{\|\mathbf{q}_i - \mathbf{q}_j\|_2 + \epsilon}. \tag{4}$$

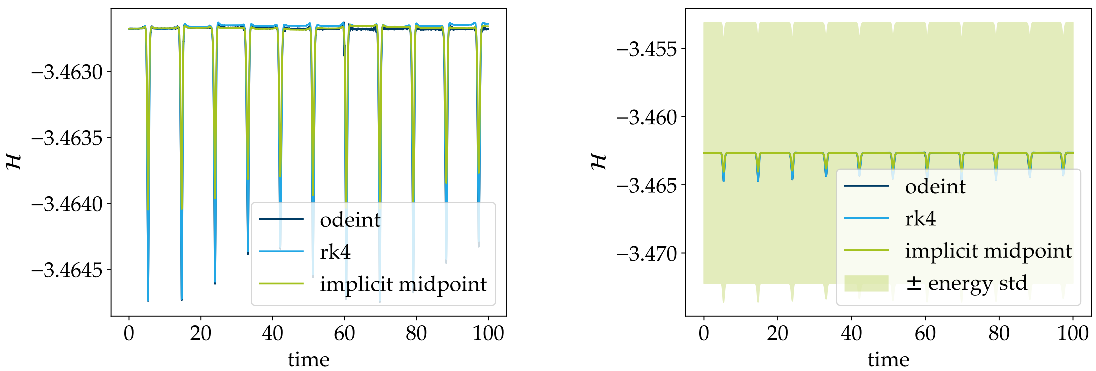
This technique is called Plummer softening and is commonly applied in N-body simulations of celestial bodies [4].

For practical reasons, the units in this project were adjusted by setting the gravitational constant to $G = 10$ and the masses of the three planets to 10, 20 and 30, respectively. This is because it is unfeasible to do calculations on real planets where the system dynamics evolve on a time scale of years or even thousands of years.

The Hamiltonian neural network used to model the Hamiltonian in this project has three hidden layers, with 64, 64 and 32 nodes, respectively. The network takes an input tensor with 18 features (3 bodies $\times$ 3 dimensions $\times$ 2 (q, p)), and gives a single scalar quantity ($\mathcal{H}$) as output. The initial positions and momenta of the planets were chosen randomly in a range between -1 and 1, and the training data was generated by using the implicit midpoint integration method with a tolerance of 10^{-6} and a step size of $dt = 0.01$ for 10 000 time steps. The HNN was trained on a single trajectory for 100 epochs using a batch size of 100, and the smoothing factor was set to $\epsilon = 0.5$.

4 Results and Discussion

Figure 1 shows a comparison between the three integration methods that were tested for generating training data; the dopri8 odeint solver from `torchdiffeq`, the fourth-order Runge-Kutta method and the implicit midpoint method. Figure 1a simply shows the Hamiltonian as a function of time for these three methods, while figure 1b includes a highlighted green area to indicate the uncertainty area of $\pm\sigma_{\mathcal{H}}$ around the implicit midpoint curve.



(a) Comparison between integration methods.

(b) Comparison with $\sigma_{\mathcal{H}} = 0.01$.

Figure 1: (a) Comparison of integration methods to create training data. The dark blue curve gives the result for the dopri8 adaptive step method, the light blue curve gives the result for Runge-Kutta 4 and the green curve gives the result for the implicit midpoint method. (b) A light green area representing $\pm\sigma_{\mathcal{H}}$ is added around the implicit midpoint curve as a comparison to the size of the numerical errors (energy spikes).

Figure 2 shows the x-, y- and z-trajectories of the training data for the three planets (here named A, B and C). By comparing figure 1 with the trajectories of the planets in figure 2, it is clear that the spikes in total energy appear at the same points in time where the trajectories of the three planets are crossing each other. These spikes can be explained by numerical errors that arise when the planets are passing close to each other and the potential energy drops very quickly in a short period of time. However, as mentioned in section 3, figure 1b clearly shows that the energy fluctuations are smaller than the standard deviation of the energy. In other words, the spikes are small compared to the energy uncertainty due to potential measurement errors, and are therefore acceptable.

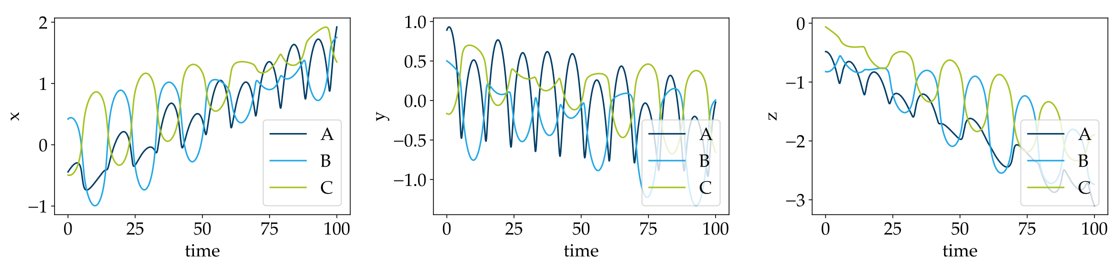


Figure 2: Trajectories of the x-, y- and z-components of the three planets A, B and C.

Figure 3 shows the training loss as a function of number of epochs when the model was trained on the trajectories shown in figure 2. The loss is clearly decreasing during the first 20 - 30 epochs, but the variation is big and the loss seems to remain relatively constant on average after that. The inconsistency and variation in loss from epoch to epoch is probably due to the chaotic nature of the system, making it difficult for the model to learn patterns in the data.

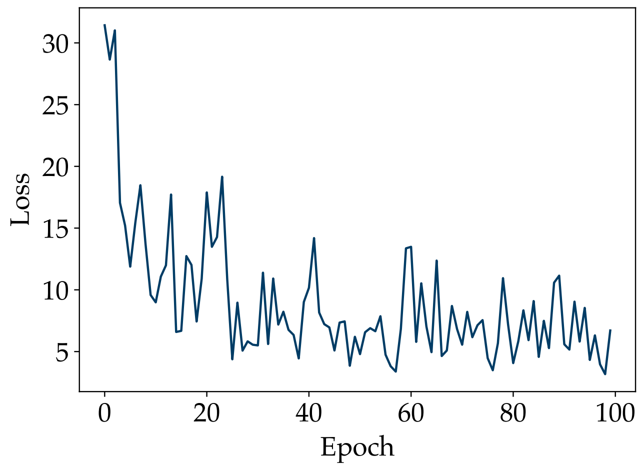


Figure 3: Training loss as a function of epochs for a HNN trained on a single trajectory with 10000 data points generated by the implicit midpoint method.

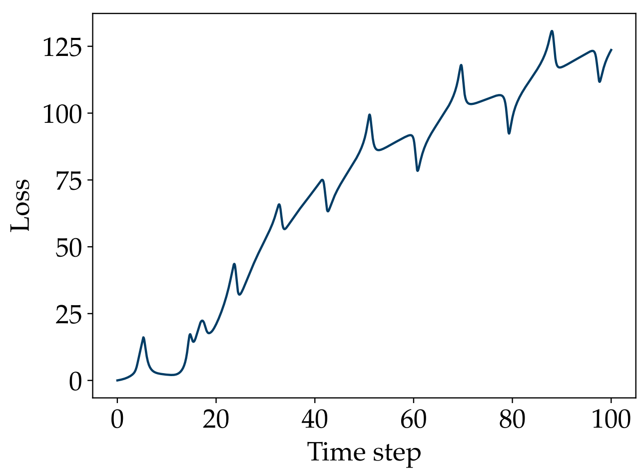
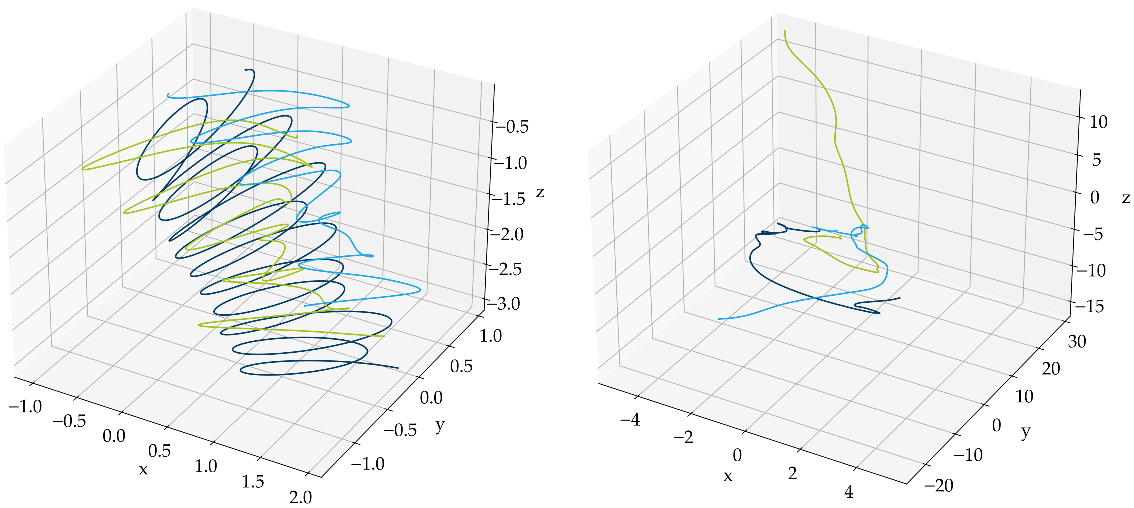


Figure 4: Test loss as a function of integration steps, where the loss was calculated as $\|x_{true} - x_{pred}\|_2$. Here, x_{true} is the training data and x_{pred} is the trajectories predicted by the HNN model.

Figure 4 shows the test loss, calculated as the L_2-norm between the 'true' and the predicted trajectory as a function of integration steps. The HNN model was given the same initial positions and momenta as used during training, and the predicted trajectory was generated by using the implicit midpoint method. The difference between the true and the predicted trajectory is that the true trajectory, or training data, was generated by using the exact expression for the Hamiltonian (equation (2)), while the predicted trajectory was generated by using the learned $\mathcal{H}_\theta$ from the neural network. In both cases equation (1) was used to find the time derivatives of q and p.

The true and predicted trajectories of the three planets are plotted in figure 5. As mentioned in section 1, the three-body problem is a chaotic system where small numerical errors will grow fast. Intuitively, it therefore makes sense that the model only can give accurate predictions a few steps ahead in time, which explains the big deviation between the paths predicted by the network and the paths in the training data.



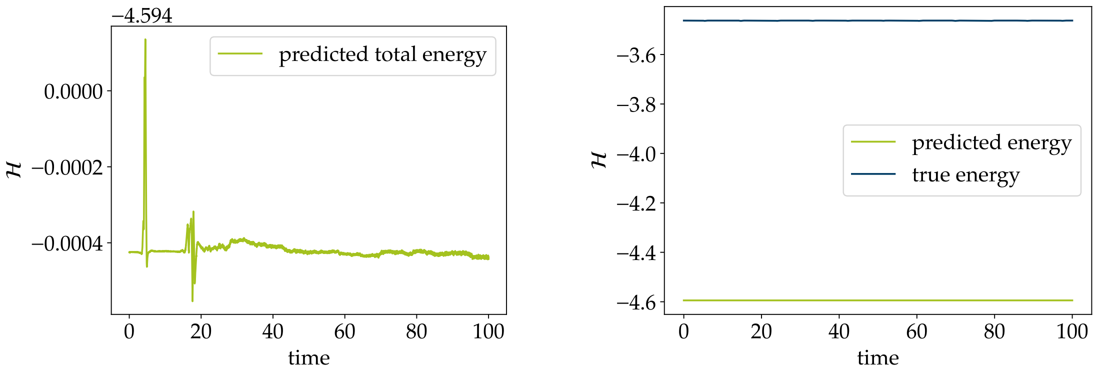
(a) Training data trajectories

(b) Predicted trajectories

Figure 5: (a) Trajectories of the training data for the three planets and (b) the trajectories predicted by the model when given the same initial conditions $\mathbf{x}_0$.

Figure 6 shows a comparison between the total energy of the training data and the Hamiltonian conserved by the neural network. In figure 6a we see that although there are some minor fluctuations, the HNN manages to conserve the learned $\mathcal{H}_\theta$ relatively well over time. This confirms that the hard-constrained approach utilised in this project actually learns a conserved quantity. In our case this is a good sign, as the total energy should be conserved in a closed system. However, a potential disadvantage of using the hard constrained loss formulation in equation 3 is that the HNN assumes a conserved quantity exists even in systems where this might not be the case.

In figure 6b, we see that the learned Hamiltonian $\mathcal{H}_\theta$ deviates from the true energy by a constant factor. This behaviour is expected for HNNs and agrees well with literature [2]. The energy offset corresponds to the ground state energy of the Hamiltonian system, which unfortunately is not known for the three body problem. For simpler systems like the harmonic oscillator model, the offset between the true and the predicted energy could be found by taking the Hamiltonian of the ground state $(q, p) = (0, 0)$.



(a) Predicted energy

(b) True vs predicted energy

Figure 6: Comparison of the quantity conserved by the HNN and the true Hamiltonian.

4.1 Further work

When working numerically with celestial problems, the units needs to be adjusted since it is impractical to work with masses at the size of the sun at time scales in years. In this project, the mass of the bodies, the time scale and the gravitational constant was therefore adjusted to arbitrary units. Due to limited time, the choice of the gravitational constant G and smoothing factor ϵ was in this report mainly based on trial and error. The choice of G and ϵ affects the contribution of the potential energy relative to the kinetic energy, and when ϵ was increased and/or G was decreased the total energy became smoother with fewer spikes, while a decrease in ϵ and/or increase in G lead to a more unstable total energy with larger fluctuations. In the further work, we will do more hyperparameter optimisation and aim to choose constants that also are physical meaningful. To avoid unphysical bias in the potential energy expression, we will also aim to reduce the value of ϵ.

Besides that, the further work will focus on implementing the partial observed methodology to the three-body problem, to see if it is possible to train a HNN on partial observations of the planets. Since the chaotic behaviour of the three-body problem might make this a difficult task, we will start working on the two-body problem in two dimensions, and work from there to extend the system to three dimensions. We will also explore different initial conditions, and try training the HNN on several trajectories simultaneously as previously done on the simple harmonic oscillator [1].

5 Conclusion

The aim of this project was to train a Hamiltonian neural network on the three-body problem. The main findings from this preliminary work is that the HNN successfully conserved the learned Hamiltonian $\mathcal{H}_\theta$, however this quantity was shifted by a constant factor relative to the true Hamiltonian of the system. The chaotic dynamics of the three-body system caused some numerical difficulties, which we will try to overcome in the further work. The further work will also aim to train a HNN on missing data.

References

[1] Sunniva Meltzer. *Simple harmonic oscillator model with missing data*. Coursework in FYS9429 – Advanced machine learning and data analysis for the physical sciences. Spring 2024.

[2] Sam Greydanus, Misko Dzamba, and Jason Yosinski. *Hamiltonian Neural Networks*. 2019. arXiv: `1906.01563 [cs.NE]`.

[3] Ricky T. Q. Chen. *torchdiffeq*. 2018. URL: `https://github.com/rtqichen/torchdiffeq`.

[4] Walter Dehnen. "Towards optimal softening in three-dimensional N-body codes – I. Minimizing the force error". In: *Monthly Notices of the Royal Astronomical Society* 324.2 (June 2001), pp. 273–291. ISSN: 0035-8711. DOI: `10.1046/j.1365-8711.2001.04237.x`. URL: `https://doi.org/10.1046/j.1365-8711.2001.04237.x`.