Hybrid integration of the gravitational N-body problem with Artificial Neural Networks

Veronica Saz Ulibarrena
Leiden University
ulibarrena@strw.leidenuniv.nl

Philipp Horn
Eindhoven University of Technology
p.horn@tue.nl

Simon Portegies Zwart
Leiden University
spz@strw.leidenuniv.nl

Elena Sellentin
Leiden University
sellentin@strw.leidenuniv.nl

Barry Koren
Eindhoven University of Technology
b.koren@tue.nl

Maxwell X. Cai
Intel Corporation
maxwell.cai@intel.com

Abstract

Studying the evolution of the gravitational N-body problem becomes extremely computationally expensive as the number of bodies increases. In order to alleviate this problem, we study the use of Artificial Neural Networks (ANNs) to substitute expensive parts of the integration of planetary systems. We compare the performance of a Hamiltonian Neural Network (HNN) which includes physics constraints into its architecture with a conventional Deep Neural Network (DNN). We find that HNNs are able to conserve energy better than DNNs in a simplified scenario with two planets, but become challenging to train for a more realistic case, namely when adding asteroids. We develop a hybrid integrator that chooses between the network's prediction and the numerical computation, and show that for a number of asteroids ≥ 60, using ANNs improves the computational cost of the simulation while allowing for an accurate reproduction of the trajectory of the bodies.

1 Introduction

Planetary systems are a special case of the gravitational N-body problem where a massive central body, a star, is orbited by multiple minor bodies. To model the evolution of planetary systems, it is necessary to know the gravitational interaction between the different bodies, which can be calculated using the equations derived by Newton (1). Unlike the calculation of the gravitational force, the equations of motion can only be solved analytically for two bodies using the relations derived by Kepler in 1609 (2). This means that when the system consists of three or more bodies, the equations need to be solved numerically. We focus on the Wisdom-Holman (WH) integrator(3) which has been developed for the specific case of planetary systems.

Currently, the study of the evolution of N-body systems is limited by the large computational resources needed to obtain an accurate (low energy error) solution (4)(5). Newton's equation of gravitation implies that the computational complexity of the problem scales as N^2, where N is the number of bodies in the system. As a consequence, for multiple applications in astrophysics, such as the evolution of globular clusters or asteroids around a star (6), the large number of bodies in the system is one of the main reasons for the high computational expense. We study the efficiency of neural networks for the substitution of computationally expensive parts of the integration of N-body

Machine Learning and the Physical Sciences workshop, NeurIPS 2022.

systems for astrophysics applications. In order to encourage conservation of energy, we compare the advantages and challenges of using a network that includes physics constraints in its architecture (7)(8) with a conventional Deep Neural Network (DNN).

Previous works apply neural networks to the gravitational two- (7)(9) and three-body (7)(10) problems. However, due to their simplifying assumptions, it cannot be assumed that the introduction of physics into the neural network represents an advantage for other problem configurations. For this reason, we carry out our study for the simplified case of a system with two planets orbiting the central star, and a more realistic case with also a large number of asteroids.

2 Methodology

2.1 Numerical integration

The Hamiltonian of a system of N particles is defined as a function of the universal gravitational constant (G), the mass of each of the bodies (m_i), the position vector ($\vec{q}_i$), and the linear momentum vector ($\vec{p}_i$):

$$\mathcal{H} = \sum_{i=1}^{N} \frac{||\vec{p}_i||^2}{2m_i} - G \sum_{i=1}^{N-1} m_i \sum_{j=i+1}^{N} \frac{m_j}{||\vec{q}_j - \vec{q}_i||}. \tag{1}$$

Solving this Hamiltonian by means of Equation 2 in (7) leads to the acceleration vector $\vec{a}$.

With the acceleration vector, the state of the system can be integrated numerically. We select the integrator designed by (3) for systems in which one body is much more massive than the others. The Wisdom-Holman (WH) method is based on separating the Hamiltonian into a Keplerian part that represents the movement of the minor bodies around the massive central body and an interactive part for the mutual interactions between bodies. The WH integrator first performs the Keplerian propagation of the orbiting bodies around the central body. Then, the perturbing acceleration is calculated using Newton's equation of gravitation (1) and and used to correct the velocity of each body.

The interactive part of the Hamiltonian of the system scales quadratically with the number of bodies. Substituting the calculation of the perturbing accelerations with Artificial Neural Networks (ANNs) could accelerate this operation. We therefore apply this method to two study cases: a planetary system with Jupiter and Saturn and the second one including a large number of asteroids.

2.2 Neural Networks

For systems in which energy is conserved, Hamiltonian Neural Networks (HNNs) constitute an attractive choice since the Hamiltonian is embedded in the network's architecture. We use HNNs to predict the interactive part of Equation 1, similarly to the Neural Interacting Hamiltonian (NIH) designed by (8). We therefore use HNNs and study their advantages and disadvantages by comparing them to the numerical integration, which we consider the baseline, and to a DNN. Unlike in the HNNs from (7), we are interested in the accelerations, which only depend on the position vectors and masses of the bodies.

Because the masses of the asteroids are orders of magnitude smaller than that of the planets, we neglect the gravitational interaction between asteroids and the effect of each asteroid on Jupiter and Saturn. Then, each input vector consists of the masses and position vectors of the two planets and of one asteroid. With these assumptions, the size of the input vector of the network becomes independent of the number of asteroids in the system and the same neural network can be used for all asteroids.

2.3 Hybrid integrator

The use of neural networks to substitute parts of the integration implies a loss in accuracy with respect to the numerical integration. In addition to this, since integration is an iterative procedure where the output of one iteration becomes the input for the next one, errors propagate with time. When including neural networks in non-linear systems, prediction errors can quickly lead to unphysical solutions. To increase the robustness of an integrator that uses neural networks, we develop a hybrid

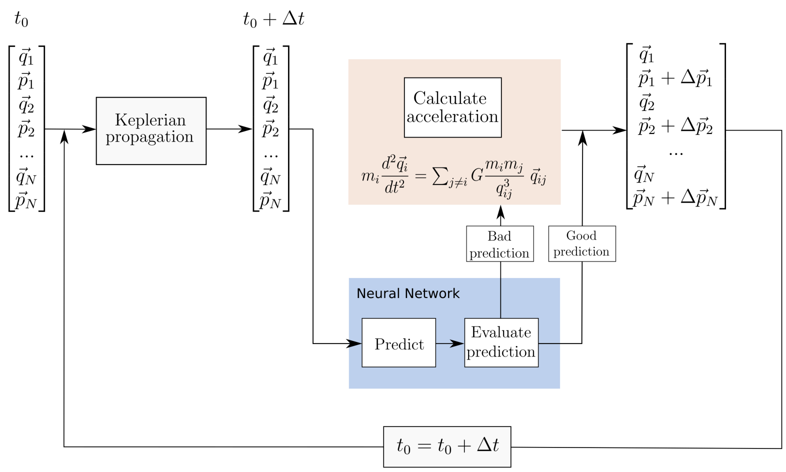

Figure 1: Schematic of the hybrid Wisdom-Holman integration.

method that selects between the prediction of the network and the numerical calculation if the former is insufficiently accurate. The criterion used to evaluate the prediction is based on the assumption that perturbations are continuous and differentiable. A large difference between the accelerations at times t_{i-1} and t_i is an indication of either a poor prediction or a region with rapid changes in the acceleration. In both cases, it is beneficial to compute those steps numerically. With this hybrid approach, we achieve a method that decreases the computational cost by using neural networks but remains sufficiently robust and reliable for scientific research.

In Figure 1 we illustrate the hybrid WH integrator. At time t_0, the state of each body is propagated for a time step Δt assuming that the particles are on a Keplerian trajectory. Afterward, the neural network calculates the accelerations induced by perturbing bodies from the predicted interactive Hamiltonian. This prediction is evaluated and, if insufficiently accurate, the accelerations are recalculated analytically. The perturbing acceleration is then converted into a correction of the velocity and the new state of the system is subsequently used as the starting point for the next iteration.

3 Results

3.1 Training of the networks

Training the DNN and the HNN for integrating two planets, we find that both networks can easily be trained to relative errors below 1%. However, when including asteroids, training the HNN becomes extremely challenging. Due to the ~ 9 orders of magnitude difference in mass between the planets and the asteroids, the training process is not able to reach a satisfactory loss value. Since the HNN includes physics constraints in the network architecture, normalization is not possible without breaking the physics. We find a workaround by focusing the training on the prediction of the accelerations of the asteroids. By doing this, we can combine two networks: one that predicts the accelerations of Jupiter and Saturn, and one that predicts the accelerations of the asteroids.

3.2 Number of bodies and computation time

The goal of this work is to speed up the integration of systems with a large number of bodies. In Figure 2 we show the computation time and the improvement in computation time and energy error with respect to the baseline integration. We compare the numerical results (WH) with the results produced by the neural network in two cases: with the HNN, and with the results by our hybrid integrator that verifies the predictions of the HNN (WH-HNN). Since the goal is to create a method that can be used for multiple experiments, the training time is considered unimportant for the evaluation of the performance. For $\geq$60 asteroids, the computation time using ANNs becomes lower than using direct numerical integration. We also find that our hybrid integrator does not increase the computation time significantly compared to the HNN case, but it contributes to improve the energy error.

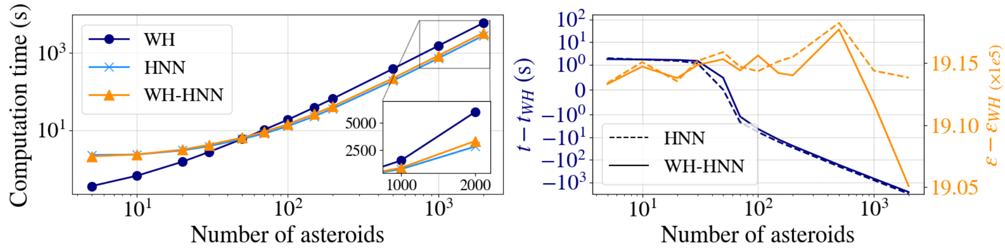

Figure 2: Comparison of the computation time (*left*) and difference in computational time and energy error with respect to the baseline numerical integration (WH) (*right*) for the integration to 20 years with a different number of asteroids.

3.3 Orbit integration

Using the trained neural networks and our hybrid method, we integrate the system with the two planets and the asteroids for 100 years. Asteroids 1 and 2 are initialized within the limits of the training dataset, whereas asteroid 3 is initialized outside to study the extrapolation capabilities of the networks. In Figure 3 we show the results with the WH integrator (first column), with the hybrid integrator using an HNN (second column), and with the hybrid integrator using a DNN (third column). The orbit evolution of the asteroids is best seen from the first row, which shows the eccentricity evolution. Both the DNN and the HNN can accurately reproduce the numerical integration results. However, this is only due to our hybrid integrator replacing the incorrect predictions. As seen in Figure 2, without these corrections the neural network is not able to produce physically-correct results. In the second row, we see the energy error, which is a measure of the quality of our integration. The energy error is dominated by the planets, and it is therefore an indication of the quality of the networks that predict the accelerations of Jupiter and Saturn. In a symplectic integrator, we do not expect energy to be perfectly conserved, but to oscillate around the correct value. We see this behavior in the numerical result, but also in the run with the HNN. In contrast, the energy error of the DNN case shows a systematic drift, which indicates that the solution diverges from the physically-correct one. We conclude that the HNN performs better as it leads to better energy conservation.

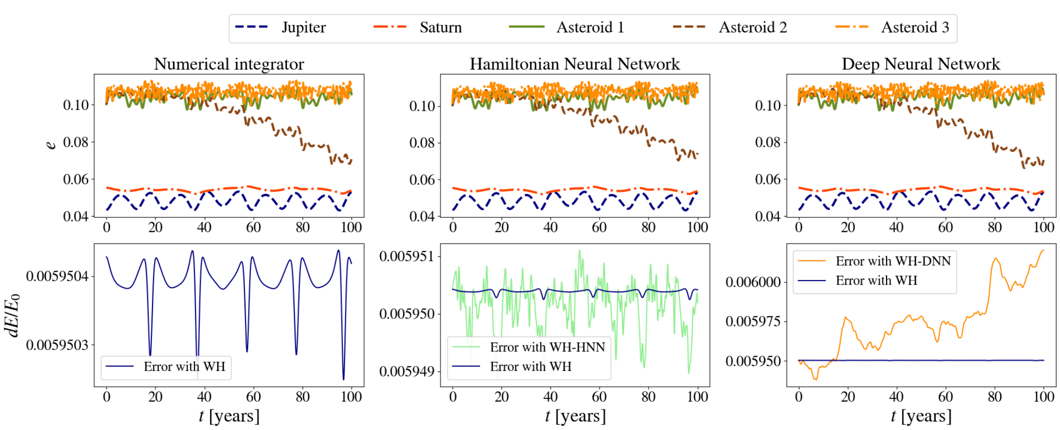

Figure 3: Simulation results for the Sun, Jupiter, Saturn, and asteroids. The results are generated with the Wisdom-Holman integrator (*left*), Wisdom-Holman with the Hamiltonian Neural Network (*center*), and Wisdom-Holman with the Deep Neural Network (*right*). The eccentricity evolution is shown in the first row and the energy error in the second row.

4 Conclusions

We compared the performance of DNNs and HNNs with direct numerical integration when used in the integration of a planetary system with asteroids. In order to overcome some of the limitations of ANNs, we developed a hybrid integrator that increases the robustness of the neural network predictions by replacing poor predictions of the neural network by the numerical calculation. For $\geq$60 asteroids, the use of neural networks results in faster integration at almost no loss in accuracy. Regarding the network architecture, we find that the DNN is fast and easy to train, but fails to conserve energy. The HNN on the other hand is difficult to train and is limited by factors such as the mass ratio between the planets and the asteroids. However, it conserves energy over a much longer integration time. We conclude that the use of neural networks within the integrator allows the simulation time to be reduced, but without our hybrid integrator the results are not suitable for scientific research.

5 Impact of this work

In our search for an accurate and efficient method to solve the gravitational N-body problem, we have explored the possibility of using two popular neural network architectures. We study the performance of those networks for both an academic problem and a more realistic scientific application, and find that many limitations arise from tackling a realistic problem with neural networks. We aim to find real solutions for an astrophysics problem, such as our hybrid integrator, to take advantage of the strengths of neural networks without ignoring their limitations.

Our work is limited to a specific problem configuration, from which conclusions have been derived. However, we cannot assume that those conclusions apply to different cases of the gravitational N-body problem.

References

[1] I. Newton, *The Principia: mathematical principles of natural philosophy*. Univ of California Press, 1999.

[2] J. Kepler, "Astronomia nova.," *Pragae 1609*, 2015.

[3] J. Wisdom and M. Holman, "Symplectic maps for the n-body problem," *The Astronomical Journal*, vol. 102, pp. 1528–1538, 1991.

[4] L. Greengard, "The numerical solution of the n-body problem," *Computers in physics*, vol. 4, no. 2, pp. 142–152, 1990.

[5] A. I. Almojel, "The implementation and performance evaluation of n-body gravitational simulation algorithm on high-performance computers," *Computers & Electrical Engineering*, vol. 26, no. 3-4, pp. 297–316, 2000.

[6] D. Richardson, P. Michel, K. Walsh, and K. Flynn, "Numerical simulations of asteroids modelled as gravitational aggregates with cohesion," *Planetary and Space Science*, vol. 57, no. 2, pp. 183–192, 2009.

[7] S. Greydanus, M. Dzamba, and J. Yosinski, "Hamiltonian neural networks," *Advances in neural information processing systems*, vol. 32, 2019.

[8] M. X. Cai, S. Portegies Zwart, and D. Podareanu, "Neural symplectic integrator with hamiltonian inductive bias for the gravitational n-body problem," *arXiv preprint arXiv:2111.15631*, 2021.

[9] R. Chen and M. Tao, "Data-driven prediction of general hamiltonian dynamics via learning exactly-symplectic maps," *arXiv preprint arXiv:2103.05632*, 2021.

[10] P. G. Breen, C. N. Foley, T. Boekholt, and S. Portegies Zwart, "Newton versus the machine: solving the chaotic three-body problem using deep neural networks," *Monthly Notices of the Royal Astronomical Society*, vol. 494, no. 2, pp. 2465–2470, 2020.

Checklist

1. For all authors...
   (a) Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope? [Yes]
   (b) Did you describe the limitations of your work? [Yes]
   (c) Did you discuss any potential negative societal impacts of your work? [No]
   (d) Have you read the ethics review guidelines and ensured that your paper conforms to them? [Yes]
2. If you are including theoretical results...
   (a) Did you state the full set of assumptions of all theoretical results? [N/A]
   (b) Did you include complete proofs of all theoretical results? [N/A]
3. If you ran experiments...
   (a) Did you include the code, data, and instructions needed to reproduce the main experimental results (either in the supplemental material or as a URL)? [No]
   (b) Did you specify all the training details (e.g., data splits, hyperparameters, how they were chosen)? [No]
   (c) Did you report error bars (e.g., with respect to the random seed after running experiments multiple times)? [N/A]
   (d) Did you include the total amount of compute and the type of resources used (e.g., type of GPUs, internal cluster, or cloud provider)? [No]
4. If you are using existing assets (e.g., code, data, models) or curating/releasing new assets...
   (a) If your work uses existing assets, did you cite the creators? [Yes]
   (b) Did you mention the license of the assets? [No]
   (c) Did you include any new assets either in the supplemental material or as a URL? [No]
   (d) Did you discuss whether and how consent was obtained from people whose data you're using/curating? [No]
   (e) Did you discuss whether the data you are using/curating contains personally identifiable information or offensive content? [N/A]
5. If you used crowdsourcing or conducted research with human subjects...
   (a) Did you include the full text of instructions given to participants and screenshots, if applicable? [N/A]
   (b) Did you describe any potential participant risks, with links to Institutional Review Board (IRB) approvals, if applicable? [N/A]
   (c) Did you include the estimated hourly wage paid to participants and the total amount spent on participant compensation? [N/A]