

REINFORCEMENT LEARNING AND ORBIT-DISCOVERY ENHANCED BY SMALL-BODY PHYSICS-INFORMED NEURAL NETWORK GRAVITY MODELS

John R. Martin*, and H. Schaub†

The novel Physics-Informed Neural Network (PINN) gravity model enables accurate and computationally efficient representations of complex gravity fields. Prior work has studied the use of PINNs for gravity field modeling of large celestial bodies and asteroids. This paper explores how PINN gravity model's accuracy and speed can be leveraged to address two problems of interest pertaining to small-body exploration: 1) augmenting the behavior of traditional spacecraft safe mode to account for the complex dynamics about small-bodies, and 2) rapid discovery of near-periodic orbits under the influence of inhomogeneous gravity fields. These problems are challenging to address due to the computationally intense algorithms needed to solve them which include reinforcement learning and boundary value problem methods. By taking advantage of the PINN gravity model's efficiency, these once cumbersome algorithms can be evaluated orders-of-magnitude more quickly than before — enabling simulations and solvers to be run without unnecessary simplification of the gravitational dynamics. This research demonstrates that the PINN gravity model uniquely enables the training of more robust and dynamically-informed reinforcement learning agents, as well as assists traditional boundary value problem solvers to identify near-periodic orbits from arbitrary initial conditions.

INTRODUCTION

With current asteroid missions like OSIRIS-REx and future missions like Psyche and Janus capturing the attention of the astrodynamics community, there grows an increasing need for powerful tools and techniques to assist spacecraft with operating near small-bodies.¹⁻⁴ Small-bodies are high priority science targets as they provide useful insights into the origin of the planets and the solar system at large, however they present a challenging environment for spacecraft to operate in due to their irregular shapes and corresponding gravity fields. These environments are difficult to represent analytically, as mission designers and dynamicist cannot rely on traditional gravity models like spherical harmonics. As soon as spacecraft enter the circumscribing (Brillouin) sphere of the body, the spherical harmonic gravity model begins to diverge which can jeopardize proximity, landing, or touch-and-go operations.⁵

To combat this divergence, the polyhedral gravity model is used to estimate accelerations generated by the small-body.⁵ The polyhedral representation uses a shape model of the body and an assumption about its density profile to compute the consequent accelerations. While this representation provides stable dynamics down to the surface of the asteroid, it relies on the assumption that a sufficiently high-fidelity shape model exists which can be time-consuming to acquire.⁶ In addition, the polyhedral model often requires assumptions about the density of the body which can bias the accelerations and potential.^{7,8} Finally, even if these requirements can be satisfied, the polyhedral gravity model comes with a high computational cost that can make it difficult to evaluate on synchronous processors.

*NSF Graduate Research Fellow, Ann and H.J. Smead Department of Aerospace Engineering Sciences, University of Colorado, Boulder, 431 UCB, Colorado Center for Astrodynamics Research, Boulder, CO, 80309.

†Glenn L. Murphy Chair in Engineering, Ann and H.J. Smead Department of Aerospace Engineering Sciences, University of Colorado, Boulder, 431 UCB, Colorado Center for Astrodynamics Research, Boulder, CO, 80309. AAS Fellow, AIAA Fellow.

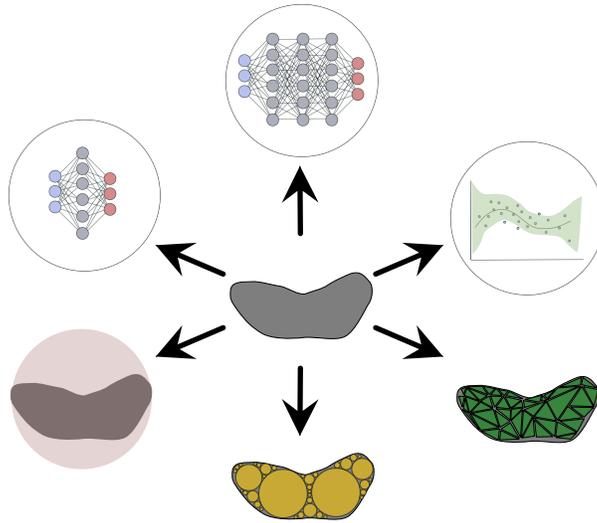


Figure 1: Popular gravity modeling options include analytic (spherical harmonics, mascons, polyhedral) and numerical representations (extreme learning machines, neural networks, and Gaussian processes).

These challenges have fueled the development of a second generation of gravity models amenable to small-body research. These models center on using numerical machine learning methods rather than the prior analytic approaches. Such models include extreme learning machines, Gaussian processes, and neural networks — each trained to learn a function which maps between a position vector input and a corresponding acceleration vector output without prescribing a set of analytic basis functions.^{9–11} These second generation models maintain stable dynamics within the Brillouin sphere like the polyhedral model but with a significantly reduced computational cost. Further studies have verified that these machine learning gravity representations yield an advantage in computational efficiency, sample efficiency, and robustness to noise over their traditional counterparts.^{12,13}

While the modeling superiority of these machine learning gravity models has been demonstrated, little literature exists highlighting their application to other research domains within astrodynamics. Many problems within astrodynamics often rely on a simplified representation of the gravity field, particularly around small-bodies^{14–16} — this paper seeks to investigate if some of these problems can be revisited and make use of the richer Physics-Informed Neural Network (PINN) gravity model. For example, consider the frozen orbit problem or the search for near-periodic, stable orbits about asteroids. Current methods rely on representing the potential of the celestial body to some truncated spherical harmonic degree (often below degree 4), differentiating that potential in accordance with Lagrange Planetary Equations, and searching for initial conditions for which certain orbital elements are time invariant, or constant on average over a fixed orbit period.¹⁷ This approach can provide reasonable first approximations of conditions for which an orbit may be slowly varying, but it relies entirely on the assumption that the truncated model is sufficiently representative of the true system. Moreover, it eliminates the possibility of finding periodic solutions that enter the Brillouin sphere where the spherical harmonic representation begins to diverge — a region generally considered to be of high-value for science operations and imaging. By using a PINN rather than a truncated spherical harmonic model, astrodynamists do not need to simply their problem for analytic tractability. In fact, there is not even a need to exclusively search in regions for which there were frozen orbits. Rather PINNs can be used to find near-periodic orbits from arbitrary initial conditions due to its fast performance.

Another encouraging application of the PINN gravity model is its utility for reinforcement learning. In reinforcement learning, it is important to train agents in an environment that is sufficiently representative of

the real environment to increase the likelihood of robustness. This is often referred to as minimizing the sim-to-real gap.¹⁸ Within an astrodynamics context, this means simulating a gravity field with to fidelity representative of the true dynamics. Rather than using a simplified model to represent the gravity field within the simulation (which is inaccurate) or a polyhedral model (which is orders-of-magnitude slower to compute), the PINN gravity model is able to simulate full-fidelity dynamics in a fraction of the time required by older analytic models. This enables more simulations, more training data, and, as consequence, more robust agents. This paper explores these claims by setting up an environment to train a fault handling agent which will be referred to as “Enhanced Safe Mode”. The agent is trained to take a spacecraft with arbitrary initial conditions and maneuver to orbits which are naturally stable, require very little fuel to maintain, and are unlikely to collide with the body.

The paper is organized into the following sections. First, a brief survey of the models used to represent the gravity fields of asteroids is presented. Following this discussion is an investigation of how PINNs can be leveraged in reinforcement learning applications — specifically designing an agent that can be enabled as an extension of traditional safe-mode operations. Next, a second application of PINNs is explored, combining PINN’s modeling efficiency with boundary value problem solvers to identify orbits that are nearly periodic. Finally, a brief conclusion is provided which highlights the advantages of using the PINN gravity model in broader astrodynamics contexts.

TECHNIQUES

There exists a diverse set of gravity field representations used to model the environments about asteroids and comets. Broadly, these representations can be divided into two groups: 1) traditional representations which rely on analytic constructions like spherical harmonics and polyhedral models, and 2) machine learning methods like the physics-informed neural network and the physics-informed transformer gravity models. This section aims to summarize these different models and identify some of their advantages and disadvantages in context of the two applications considered in this paper.

Traditional Methods

The longest standing gravity models are those with analytic roots, particularly the spherical harmonic model, the polyhedral model, and the mascon model.

Spherical Harmonics Spherical harmonics are one of the most popular representations for the gravitational potential of large and predominately spherical celestial bodies. Using spherical harmonics the potential can be expressed as:

$$U_l(r) = \frac{\mu}{r} \sum_{l=0}^l \sum_{m=0}^l \left(\frac{R}{r}\right)^l P_{l,m}[\sin(\phi)] [C'_{l,m} \cos(m\lambda) + S'_{l,m} \sin(m\lambda)] \quad (1)$$

where r is the field point, μ is the gravitational parameter of the body, R is the circumscribing radius of the celestial body, l is the degree of the model, m is the order of the model, $C'_{l,m}$ and $S'_{l,m}$ are the regressed spherical harmonic coefficients, λ is the longitude, ϕ is the latitude, and $P_{l,m}$ are the associated Legendre polynomials.¹⁹

Spherical harmonics’ popularity is a function of two features. First, the representation provides an analytic solution to Laplace’s equation:

$$\nabla^2 U = 0 \quad (2)$$

Second, spherical harmonics are extremely efficient at representing planetary oblateness — the dominate perturbation for large celestial bodies. This oblateness can be expressed by expanding Equation 2 to a mere degree $l = 2$.

The appeal of spherical harmonics, however, is not universal. The analytics become unstable when evaluating field points within the circumscribing sphere of the celestial body (Brillouin sphere) due to the $\left(\frac{R}{r}\right)^l$ term in Equation 1. This divergence can be particularly risky for small-body operations as the shape of the

asteroid or comet may have a surface that exists well beneath this sphere.⁵ In addition, spherical harmonics struggle to capture discontinuous features like mountain ranges or craters using its periodic basis set due to the three-dimensional analog of Gibbs phenomenon.¹² To combat these challenges, there exist two alternative gravity models — the mascon representation and the polyhedral representation.

Mascons The mascon gravity model distributes a collection of point mass elements underneath the surface of the body and uses the collective sum of these elements to form the total gravitational potential of the system. Such an approach is useful for accommodating the complex geometries of asteroids while avoiding the model divergence within the Brillouin sphere. Unfortunately the mascons can become problematic when spacecraft attempt close proximity operations as the discrete nature of the elements grows increasingly apparent.²⁰

Polyhedral The divergence of the spherical harmonic model and the discrete nature of the mascon model can both be resolved using the polyhedral gravity model. The polyhedral gravity model takes a shape model of the body comprised faces and vertices and predicts accelerations using the following equation:

$$\nabla U = -G\sigma \sum_{e \in \text{edges}} \mathbf{E}_e \cdot \mathbf{r}_e \cdot L_e + G\sigma \sum_{f \in \text{faces}} \mathbf{F}_f \cdot \mathbf{r}_f \cdot \omega_f \quad (3)$$

where G is the gravitational constant, σ is the density of the body, \mathbf{E}_e is an edge dyad, \mathbf{r}_e is the position vector between the center of the edge and the field point, L_e is an analog to the potential contribution by the edge, \mathbf{F}_f is the face normal dyad, \mathbf{r}_f is the distance between the face normal and the field point, and ω_f is analog to the potential contribution by the face. More details can be found in the original publication.⁵

While the polyhedral gravity model offers researchers one of the strongest analytic gravity models, it suffers from two key disadvantages. First, an assumption must be made about the density of the body as denoted by the lack of subscript for σ in Equation 3. While a inhomogeneous density can be projected onto the model, such profiles are difficult to resolve uniquely.^{7,21} The second disadvantage is the computational overhead of the model. There exist asteroid shape models that use in excess of 200,000 facets — requiring Equation 3 to execute expensive for loop operations to evaluate the acceleration at a single field point.²² This quickly becomes computationally prohibitive and poses a challenge for both offline simulation and onboard control unless properly parallelized.¹²

Machine Learning Methods

The challenges posed by the traditional gravity representations have motivated a second generation of gravity models which rely on machine learning methods rather than pure analytics. These models include using Gaussian processes, extreme learning machines, and most recently neural networks to represent the gravity fields of both large, near-spherical bodies and irregularly shaped small-bodies.^{10,11} These approaches have demonstrated orders-of-magnitude improvements in computational efficiency over the expensive polyhedral representation. The most recent developments have shown that there exist classes of neural network gravity models which also offer significant improvements in sample efficiency and robustness to noise by using physics priors to constrain the learned models.^{12,13} These representations are referred to as Physics-Informed Neural Network (PINN) gravity models.

Physics-Informed Neural Network Gravity Model In 2019, Raissi et. al. introduced the Physics-Informed Neural Network (PINN) — a neural network which learns to approximate a physical process or system by training traditional deep learning models with an augmented cost function that penalizes both modeling inaccuracies and violations of underlying differential equations.²³ For example — in the case of the gravity field modeling problem — a traditional neural network would learn the mapping between the position vector and the corresponding acceleration vector by minimizing the following cost function

$$J(\Theta) = \frac{1}{N_f} \sum_{i=1}^{N_f} |\mathbf{a}_i - f(\mathbf{x}_i|\Theta)|^2 \quad (4)$$

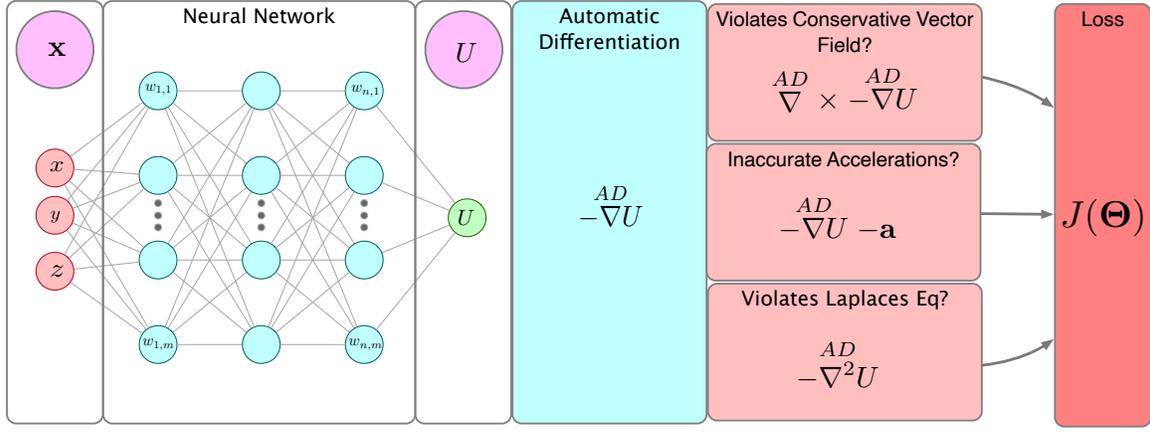


Figure 2: Physics-Informed Neural Network Gravity Model

where $f(\mathbf{x}|\Theta)$ is the learned mapping of the acceleration vector (i.e. $f(\mathbf{x}|\Theta) = \hat{\mathbf{a}}$). While this cost function will yield a mapping between these two quantities, the underlying dynamics of the system may not be satisfied.

To elaborate, given that the network is attempting to represent the force of gravity, it is also true that the acceleration is a byproduct of a more fundamental scalar potential U through:

$$\mathbf{a} = -\nabla U \quad (5)$$

and that scalar potential must also satisfy Laplace's equation outside of the body

$$\nabla^2 U = 0 \quad (6)$$

and the curl of the acceleration vector produced by said potential must be zero

$$\nabla \times \nabla U = 0 \quad (7)$$

By training with only the cost listed in Equation 4, the network will be agnostic to these constraints — potentially generating a gravity field model which does not satisfy these properties. PINNs, in contrast, specifically includes these dynamical constraints in the cost function — forcing that the network learns representations that remain consistent with the physics. To accomplish this, the network must instead learn a representation of the scalar potential, $f(\mathbf{x}|\Theta) = \hat{U}$, and then use automatic differentiation (AD) to take the corresponding gradient of the potential to enforce that Equations 5, 6, 7 are satisfied through the following cost function:

$$J_{\text{ALC}}(\Theta) = \frac{1}{N_f} \sum_{i=1}^{N_f} \left| \mathbf{a}_i + \nabla f(\mathbf{x}_i|\Theta) \right|^2 + \left| \nabla^2 f(\mathbf{x}_i|\Theta) \right|^2 + \left| \nabla \times f(\mathbf{x}_i|\Theta) \right|^2 \quad (8)$$

Including these additional constraints in the cost function provides a number of advantages. Most obvious is the fact that the network is now trained with the knowledge that gravitational acceleration is a function of a more fundamental scalar potential, and that scalar potential must satisfy a number of differential equations. Second, by imposing priors on the permissible solutions generated by the network, PINNs become very sample efficient — requiring very little training data to achieve impressive modeling accuracy as demonstrated in Martin et. al.¹³

Physics-Informed Transformers With the introduction of PINN gravity models,¹² there remained an open question if different network architectures were better suited to the gravity field modeling problem. Just

as convolutional neural networks are useful in capturing translational and rotational invariances in image datasets, perhaps there exists a different architecture better suited at expressing physical invariances. Such topic of research was explored by Wang et. al.²⁴ which demonstrated that a transformer-inspired architecture can offer an order-of-magnitude modeling improvement in accuracy over the vanilla PINN representation. This architecture was adopted and studied in Martin et. al.¹³ confirming such findings for the asteroid gravity modeling problem. As such, the transformer-inspired architecture presented in Martin et. al. will be PINN architecture of choice for remainder of this paper. Table 1 provides the set of hyperparameters used to train these models, and further implementation details can be found in the original publication.¹³

Table 1: Nominal Hyperparameters

Parameter	Value	Parameter	Value
Activation	GELU	Hidden Layers	8
Batch Size	5,000	Nodes Per Layer	20
Optimizer	Adam	Weight Initialization	Glorot Normal
Epochs	7,500	Learning Rate	0.002

APPLICATION: REINFORCEMENT LEARNING

This section showcases the first application of the PINN gravity model — specifically highlighting how the PINN gravity model can enhance the quality of agents trained via reinforcement learning. To demonstrated consider the following problem:

A spacecraft is in orbit about a small-body and has just begun a complex operation such as executing a touch-and-go (T.A.G.) maneuver. In the middle of this operation, the spacecraft experiences an unexpected loss of communication with the ground. This anomaly triggers the spacecraft’s Safe Mode — a power-positive, stationary mode that allows scientists and engineers to investigate the problem and upload a solution.²⁵ For spacecraft orbiting large-celestial bodies, enabling Safe Mode would be considered a robust and risk-free action. This is because the dynamics experienced by spacecraft around nearly-spherical bodies are sufficiently keplerian that even if the spacecraft enters Safe Mode for an extended period of time, there is relatively low probability of the spacecraft colliding with the body it is orbiting. In small-body exploration, such guarantee is less apparent.

The gravity fields produced by irregularly shaped asteroids do not provide the same nominally stable orbits that are present around large celestial bodies. Instead spacecraft experience complex accelerations coupled with the effects of solar radiation pressure which can produce unpredictable dynamics about the small-body without proper station-keeping. Consequently, by entering a traditional Safe Mode during a critical operation like T.A.G., the spacecraft is functionally guaranteed to collide with the body.

This section aims to generate a solution to this problem called Enhanced Safe Mode. When a spacecraft enters Enhanced Safe Mode, the spacecraft prioritizes three safety objectives: 1) the spacecraft must not collide with the body, 2) the spacecraft must conserve fuel, and 3) the spacecraft must remain sufficiently close to the body such that the gravity field remains the dominant perturbation. Such safety objectives ensure that, regardless of mission phase, the spacecraft will not intersect the body and will maneuver onto orbits that require relatively little station-keeping. In principle, once this mode is executed and a quasi-stable orbit is found, traditional Safe Mode operations can ensue and operators can return to diagnosing the original problem.

The challenge with designing an Enhanced Safe Mode is that there does not exist a control solution known a priori which will satisfy these high-level objectives. As such, this section turns to reinforcement learning to learn a policy capable of converting these high-level safety constraints into an rapidly executable control solution for the spacecraft.

Reinforcement Learning

Reinforcement learning is a field responsible for generating policies that maximize the expected return of some value function for a given time horizon. Explicitly, given the Markov Decision Process (MDP): (S, A, T, R, γ) where $S \in \mathcal{S}$ is the state, $A \in \mathcal{A}$ is the set of actions, $T(s'|s, a)$ is the transition function, $R(s, a)$ is the reward at state s when action a is taken, and γ is the discount factor, reinforcement learning seeks to derive a policy $\pi : \mathcal{S} \rightarrow \mathcal{A}$ which maximizes the expected return

$$R = \mathbb{E} \left[\sum_{t=0}^T \gamma^t r_t \right] \quad (9)$$

For Enhanced Safe Mode, the state space is defined $\mathcal{S} : \mathbb{R}^4 \times \mathbb{R}^3 \times \mathbb{R}$ where an instance of the state is

$$\mathbf{s} = (\bar{\mathbf{r}}, \bar{\mathbf{v}}, \bar{m}_f) \quad (10)$$

where $\bar{\mathbf{r}}$ is the normalized position vector expressed as (\bar{r}, s, t, u) where \bar{r} is the radial distance of the spacecraft with respect to the asteroid center of mass, normalized by the maximum radius of the asteroid R . $s, t, u \in [-1, 1]$ are the tangents of the angles between the cartesian basis vectors $\hat{\mathbf{x}}, \hat{\mathbf{y}}, \hat{\mathbf{z}}$ respectively. Likewise $\bar{\mathbf{v}}$ represents the normalized velocity vector of the spacecraft $v_x/v_{\text{norm}}, v_y/v_{\text{norm}}, v_z/v_{\text{norm}}$ where v_{norm} is the escape velocity as defined at the Brillouin sphere of the asteroid:

$$v_{\text{norm}} = \sqrt{2 \frac{\mu}{R}} \quad (11)$$

Finally \bar{m}_f is the normalized remaining fuel in the spacecraft, m/m_i , where m_i is the maximum fuel carried by the spacecraft.

The action space for the MDP, $\mathcal{A} : \mathbb{R}^3$, is continuous and represents impulsive ΔV s that can be applied instantaneously at the beginning of every simulation step. The magnitude of ΔV is constrained to $[-10, 10]$ centimeters per second in each cartesian direction.

The reward function is defined such that failure is heavily penalized, and the spacecraft is incentivized to remain near the asteroid

$$R(s, a) = \begin{cases} -100 & \text{if failure}(s, a) \\ 1 - \bar{r} & \text{otherwise} \end{cases} \quad (12)$$

where failure is defined as:

$$\text{failure}(s, a) = (\bar{r} \in \text{asteroid}) \vee (\text{fuel} < 0) \vee (\bar{r} < 3R) \quad (13)$$

The transition function $T(s'|s, a)$ is defined using the gravitational dynamics of the system. The acceleration is provided by one of the aforementioned gravity models, the change in fuel is governed by the rocket-equation, and the ΔV generated by the action is applied instantaneously. The discount factor γ is 0.99.

Methods

To solve this MDP, a Soft Actor-Critic (SAC) algorithm is used.²⁶ SAC algorithms generate an executable policy in a manner similar to traditional actor-critic algorithms except the value function for the SAC's critic includes a maximum entropy term.

To elaborate, in traditional actor-critic algorithms, two cooperative function approximators are used generate a policy which maximizes the value function of the MDP. Typically the value function $V(s)$ is the expected return of the MDP (Equation 9). Given that the value function is not originally known at runtime, a function approximator — often a neural network — is used instead and is referred to as the critic or V_ψ . A second function approximator forms the actor or policy π_ϕ which takes actions that maximize the return of the value estimate provided by the critic.

The actor and critic are trained to optimize the following cost function:

$$L(\psi, \phi) = \frac{1}{2} \mathbb{E} [(V_\psi(s) - V^{\pi_\phi}(s))^2] \quad (14)$$

where $V^{\pi_\phi}(s)$ is the approximation of the expected return based the actions taken by the actor. This cost function is minimized using using gradient ascent for the actor via:

$$\Delta L(\phi) = \mathbb{E} \left[\sum_{t=1}^T \Delta_\phi \log \pi_\phi(a|s) \gamma^{t-1} V_\psi(s, a) \right] \quad (15)$$

and using gradient descent for the critic using:

$$\Delta L(\psi) = \mathbb{E} \left[\sum_{t=1}^T (V_\psi(s) - V^{\pi_\phi}(s)) \Delta_\psi V_\psi(s) \right] \quad (16)$$

As mentioned, the difference between traditional actor-critic algorithms and soft actor-critic algorithms is that SAC augment the value function to include a maximum entropy term in addition to the expected return

$$R^*(\pi) = \mathbb{E}_\phi \left[\sum_{t=1}^T r(s_t, a_t) - \alpha \log(\pi_\phi(a_t|s_t)) \right] \quad (17)$$

such that the actor not only seeks reward, but it also learns a policy that promotes diverse behavior. This helps to balance exploration and exploitation during training of the agent. This modification improves sample efficiency and decreases sensitivity to initial hyperparameters. Moreover, SAC have the ability to be trained in an off-policy manner.²⁶

Soft Actor-Critic's make use of three function approximators:

1. $V_\psi(s_t)$ — the state value function
2. $Q_\theta(s_t, a_t)$ — the soft Q-function or the state-action value function
3. $\pi_\phi(a_t|s_t)$ — the tractable policy

Typically the state value function and soft Q-function are represented as neural networks such that the parameters ψ and θ are the trainable weights and biases of the network. π_ϕ is a Gaussian distribution with mean and covariance determined by neural networks.

The goal of the reinforcement learning algorithm is to optimize the following cost functions:

$$J_V(\psi) = \mathbb{E}_{\mathbf{s}_t \sim \mathcal{D}} \left[\frac{1}{2} (V_\psi(\mathbf{s}_t) - \mathbb{E}_{\mathbf{a}_t \sim \pi_\phi} [Q_\theta(\mathbf{s}_t, \mathbf{a}_t) - \log \pi_\phi(\mathbf{a}_t | \mathbf{s}_t)])^2 \right] \quad (18)$$

$$J_Q(\theta) = \mathbb{E}_{(\mathbf{s}_t, \mathbf{a}_t) \sim \mathcal{D}} \left[\frac{1}{2} (Q_\theta(\mathbf{s}_t, \mathbf{a}_t) - \hat{Q}(\mathbf{s}_t, \mathbf{a}_t))^2 \right] \quad (19)$$

$$J_\pi(\phi) = \mathbb{E}_{\mathbf{s}_t \sim \mathcal{D}} \left[\text{DKL} \left(\pi_\phi(\cdot | \mathbf{s}_t) \parallel \frac{\exp(Q_\theta(\mathbf{s}_t, \cdot))}{Z_\theta(\mathbf{s}_t)} \right) \right] \quad (20)$$

$$(21)$$

which can all be solved using stochastic gradient descent. Specifically the value and soft Q networks can be updated using:

$$\hat{\nabla}_\psi J_V(\psi) = \nabla_\psi V_\psi(\mathbf{s}_t) (V_\psi(\mathbf{s}_t) - Q_\theta(\mathbf{s}_t, \mathbf{a}_t) + \log \pi_\phi(\mathbf{a}_t | \mathbf{s}_t)) \quad (22)$$

$$\hat{\nabla}_\theta J_Q(\theta) = \nabla_\theta Q_\theta(\mathbf{a}_t, \mathbf{s}_t) (Q_\theta(\mathbf{s}_t, \mathbf{a}_t) - r(\mathbf{s}_t, \mathbf{a}_t) - \gamma V_{\bar{\psi}}(\mathbf{s}_{t+1})) \quad (23)$$

$$(24)$$

The policy distribution could be updated via likelihood ratio gradient estimator (avoids backpropagating gradients), but because the target density is the Q-function and can be differentiated, the reparameterization trick is used instead to produce a lower variance estimator. This can be done by reparameterizing the policy using a neural network transformation

$$\mathbf{a}_t = f_\phi(\epsilon_t; \mathbf{s}_t) \quad (25)$$

where ϵ_t is a noise vector. This allows the cost function to be rewritten as

$$J_\pi(\phi) = \mathbb{E}_{\mathbf{s}_t \sim \mathcal{D}, \epsilon_t \sim \mathcal{N}} [\log \pi_\phi(f_\phi(\epsilon_t; \mathbf{s}_t) | \mathbf{s}_t) - Q_\theta(\mathbf{s}_t, f_\phi(\epsilon_t; \mathbf{s}_t))] \quad (26)$$

and solved via

$$\hat{\nabla}_\phi J_\pi(\phi) = \nabla_\phi \log \pi_\phi(\mathbf{a}_t | \mathbf{s}_t) + (\nabla_{\mathbf{a}_t} \log \pi_\phi(\mathbf{a}_t | \mathbf{s}_t) - \nabla_{\mathbf{a}_t} Q(\mathbf{s}_t, \mathbf{a}_t)) \nabla_\phi f_\phi(\epsilon_t; \mathbf{s}_t) \quad (27)$$

Note that in the prior equations \mathcal{D} represents the distribution from previous states and action tuples as available in the replay buffer, the update process uses a target value network $V_{\bar{\psi}}$ where the parameters are an exponentially moving average of the value network weights, and

$$\hat{Q}(\mathbf{s}_t, \mathbf{a}_t) = r(\mathbf{s}_t, \mathbf{a}_t) + \gamma \mathbb{E}_{\mathbf{s}_{t+1} \sim p} [V_{\bar{\psi}}(\mathbf{s}_{t+1})] \quad (28)$$

Additional implementation details about soft actor-critic can be found in the original paper.²⁶

Simulation

The environment used to train the Enhanced Safe Mode agent is configured as follows: The spacecraft begins at some initial radius from the asteroid center of mass that exists between 10 kilometers above the surface of asteroid and three times the maximum radius of the asteroid. The spacecraft is also initialized with a random initial velocity that is bound in magnitude to less than $\sqrt{3}$ meters per second. Such initial conditions are designed to replicate states that might be seen in a T.A.G. scenario, such that if no action is taken the spacecraft will likely collide with the asteroid. A set of these initial conditions propagated without an Enhanced Safe Mode agent enabled is shown in Figure 3.

Because SAC is an off-policy algorithm, a replay buffer is initialized with 10,000 (s, a, r, s') tuples which are generated by evolving keplerian initial condition with a policy that applies random δV impulses. This initial buffer is used to warm-start the training of the SAC networks. After this initial training data is generated, the replay buffer is populated with tuples generated by the interactions of agent in training. The buffer continues to fill until 100,000 tuples are stored, after which tuples will be loaded into and out of the replay buffer using the first-in, first-out paradigm.

The actor and critic networks are trained for an epoch using a 1024 batch size after a new tuple is loaded into the replay buffer / an additional step is taken in the environment. The length of a step corresponds to 10 minutes in simulation time and the maximum duration of an entire episode is 10 simulation hours. It is after every step that the impulsive ΔV action produced by the actor is applied to the state. After every 1000 steps in the environment, the performance of the policy is evaluated on 10 randomly generated episodes and the mean return and standard error of those episodes are saved.

Experiment

Using this simulation environment, three different scenarios are run to highlight the utility of the PINN gravity model for reinforcement learning. The first scenario populates the RL environment with the analytic polyhedral gravity model with a 200,000 facet shape model of the asteroid 433-Eros. As the agent interacts with the environment, the polyhedral gravity model is queried for accelerations that are used by the transition function to evolve the state forward in time. These accelerations are the most accurate of the scenarios, but come with the greatest computational cost. This decreases the number of interactions the agent will have with the environment and limits the experiences from which the agent can learn.

The second scenario populates the reinforcement learning environment with a simple point mass gravity model. This choice considerably simplifies the dynamics of the environment, making it faster to evaluate than

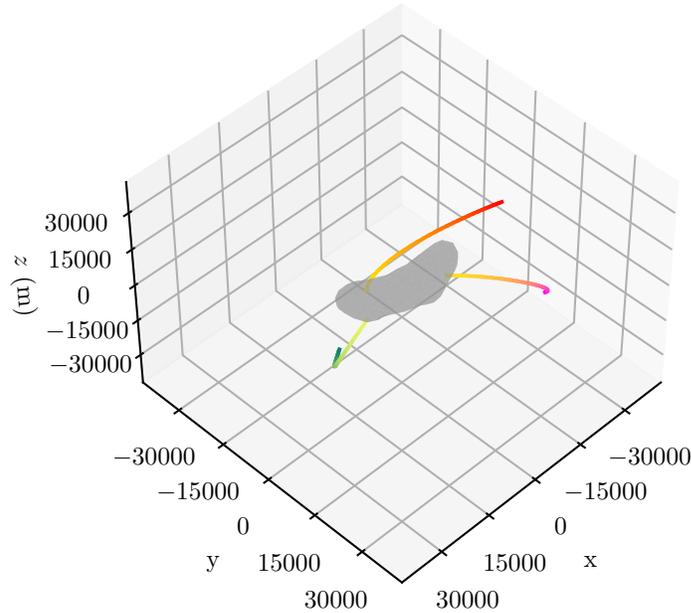


Figure 3: Trajectories taken **without** the Enhanced Safe Mode agent enabled. The darker shades correspond with earlier parts of the trajectory.

the polyhedral scenario. By decreasing the computational effort required to simulate the environment, the agent will have a significantly higher number of interactions from which it can learn. Unfortunately, by using the simple gravity model the accuracy of the simulation is degraded and the sim-to-real gap widened. The question becomes: do the additional environment interactions counteract the degradation of model fidelity?

The third and final scenario uses a PINN gravity model within the reinforcement learning environment. The PINN gravity can circumvent the challenges associated with the prior two scenarios due to its rapid executability and high accuracy. In this scenario the agent can experience more training episodes than its polyhedral counterpart, without sacrificing simulation fidelity like the simple point mass gravity model.

Results

One agent is trained in each scenario for three wall-clock hours, and the average return for each scenario’s agent is plotted as a function of wall-clock times and training steps in Figure 4.

Figure 4 provides a number of key insights into the value of the PINN gravity model. First, the agent trained in the polyhedral environment has the most accurate dynamics, but was only able to take 2,000 steps in the environment over the three hour period. This is two orders-of-magnitude fewer environment interactions than the simple gravity environment. Consequently, the agent trained in this environment was unable to learn a policy that enabled safe behavior. The computational burden of the polyhedral gravity model prevented the agent from acquiring enough experience to take safe actions.

The simple gravity model scenario was also unable to learn safe behavior, but not for lack of enough environment interactions. Having run over 300,000 steps through the environment in the three hour training window, the agent trained in the simple environment had sufficient amounts of data from which to learn. Instead, this agent’s demise came from the simplified dynamics of the environment. Because the environment did not provide the agent with sufficiently representative episodes, the agent learned behaviors that were unsafe which failed to account for the more complex dynamics of the system.

Fortunately, the agent trained in the environment with a PINN gravity model avoids both of these challenges. The agent experienced over 150,000 interactions with an environment that had representative dy-

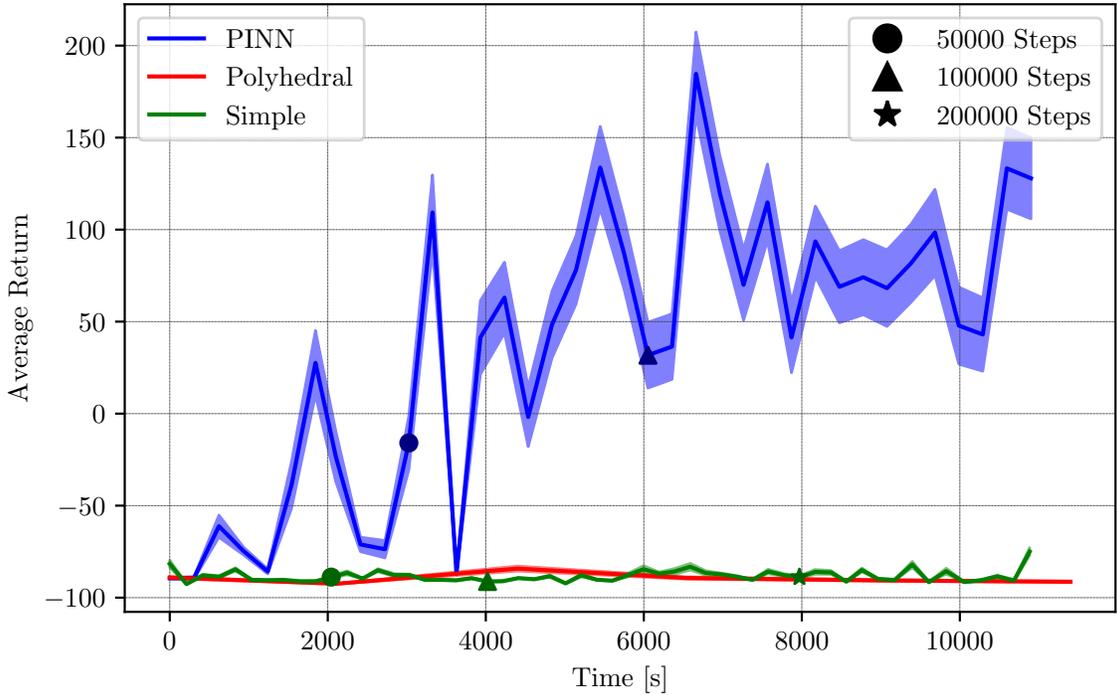


Figure 4: Average return as a function of clock time

namics of the true system. As such, there was sufficient training data for the agent’s policy to learn safe behaviors, without having to sacrifice on realistic dynamics. This combination allowed the agent to identify safe behavior that avoided collision with the asteroid. An example of the different trajectories generated by the three Enhanced Safe Mode agents are shown in Figure 5.

Figure 5 demonstrates that the agents trained in the simple and polyhedral environments are virtually no better than an untrained agent — colliding very quickly with the asteroid. The agent trained in the PINN environment, by contrast, is able to successfully leverage the complex dynamics of the gravity field to find a close, but safe, orbit around the equator of the asteroid. Early findings have shown that the trajectory found by the PINN agent is safe for significantly longer than the original 10 hour episode length implying that operators could have multiple days to diagnose the underlying anomaly. A formal stability analysis of this learned trajectory is left for future work.

There does remain the question of robustness. How many initial conditions is the Enhanced Safe Mode agent able to guarantee safety? To study this, 100 random initial conditions are given to each trained agent, and simulated for the entirety of the 10 hour episode or until the spacecraft crashes. The corresponding successes and failures of the agents are shown in Figure 6.

Figure 6 further emphasizes that neither policy generated in the simple or polyhedral environments are robust to initial conditions. There are rare occasions where those agent “succeed”, but this is actually because the initial conditions are at sufficiently high altitude that the 10 hour episode terminates before the spacecraft ever reaches an altitude low enough that it could collide with the asteroid.

The policy trained in the PINN environment is not universally robust. Only in 45% of the initial conditions provided is the agent able to reach a safe orbit and avoid collision. That said, this is a sizable improvement over the agents trained in the polyhedral and simple scenarios. Also, the performance of each agent is likely to improve with additional training time. It is not uncommon for reinforcement learning agents to be trained over the span of days rather than hours. The choice to limit this experiment to three hours of training time is

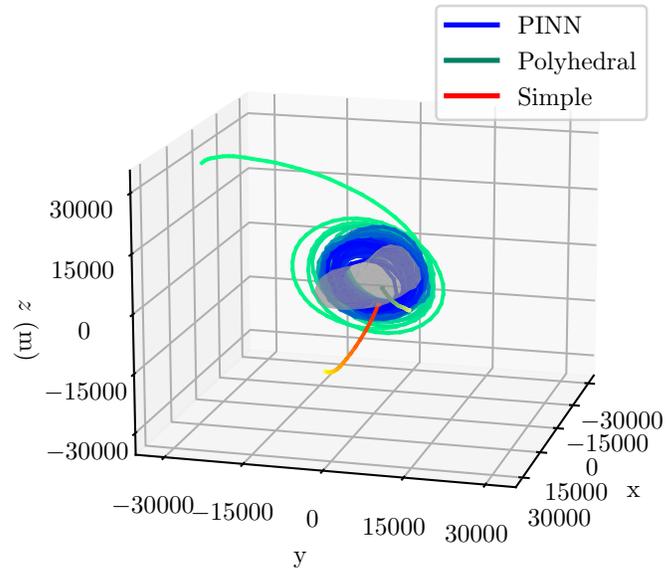


Figure 5: Trajectories taken **with** Enhanced Safe Mode agent enabled.

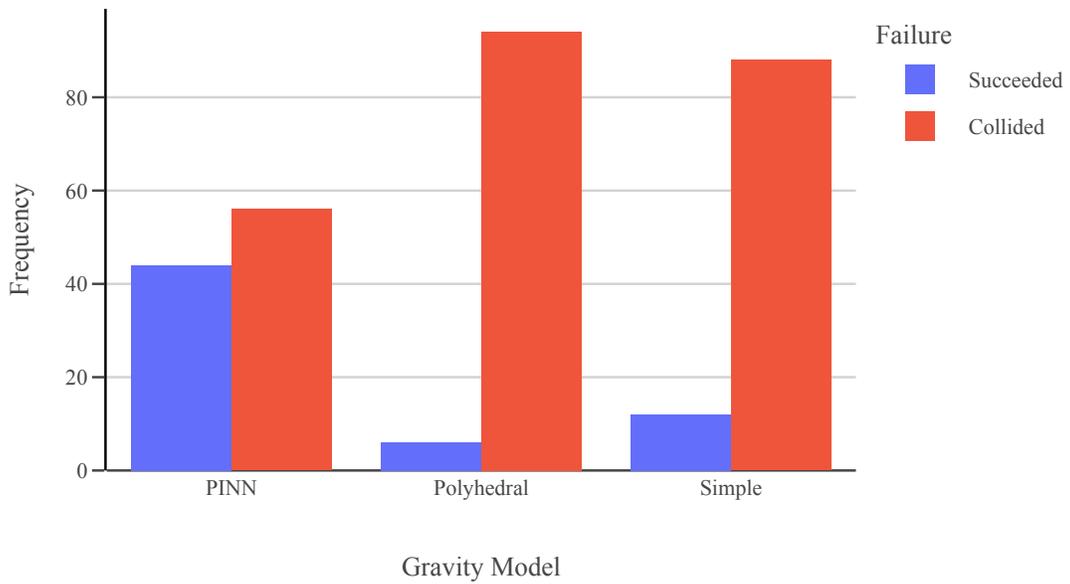


Figure 6: Success rates of the agents trained in different environments

simply to highlight the efficiency of the PINN model, and how it yields better performing agents in less time.

APPLICATION: NEAR-PERIODIC ORBIT DISCOVERY

The second application of the PINN gravity model is its use for efficient discovery of near-periodic orbits about small-bodies. When modeling orbits around large planetary bodies like the Earth or Moon, it is often reasonable to assume that the perturbations generated by inhomogeneities in the shape or density of the body are minor in comparison to that of the point mass contribution. As such, orbits about these simplified bodies are keplerian and periodic by construction. However, when inhomogeneities are introduced to the potential, this periodic behavior dissolves. This is most famously demonstrated by accounting for the effect of J_2 on satellite orbits about Earth. By including a single perturbing term in the gravitational potential, the orbital elements are no longer fixed.

The evolution of the orbital elements are succinctly described using the Lagrange Planetary Equations (LPE) which relate the rate of change of each element to the perturbing potential R — or the contributions to the scalar potential U beyond the point mass term.²⁷

$$\frac{\partial a}{\partial t} = \frac{2}{na} \frac{\partial R}{\partial M} \quad (29)$$

$$\frac{\partial e}{\partial t} = \frac{1-e^2}{na^2e} \frac{\partial R}{\partial M} - \frac{\sqrt{1-e^2}}{na^2e} \frac{\partial R}{\partial \omega} \quad (30)$$

$$\frac{\partial i}{\partial t} = \frac{\cos i}{na^2\sqrt{1-e^2}\sin i} \frac{\partial R}{\partial \omega} - \frac{1}{na^2\sqrt{1-e^2}\sin i} \frac{\partial R}{\partial \Omega} \quad (31)$$

$$\frac{\partial \omega}{\partial t} = -\frac{\cos i}{na^2\sqrt{1-e^2}\sin i} \frac{\partial R}{\partial i} + \frac{\sqrt{1-e^2}}{na^2e} \frac{\partial R}{\partial e} \quad (32)$$

$$\frac{\partial \Omega}{\partial t} = \frac{1}{na^2\sqrt{1-e^2}\sin i} \frac{\partial R}{\partial i} \quad (33)$$

$$\frac{\partial M}{\partial t} = n - \frac{1-e^2}{na^2e} \frac{\partial R}{\partial e} - \frac{2}{na} \frac{\partial R}{\partial a} \quad (34)$$

$$(35)$$

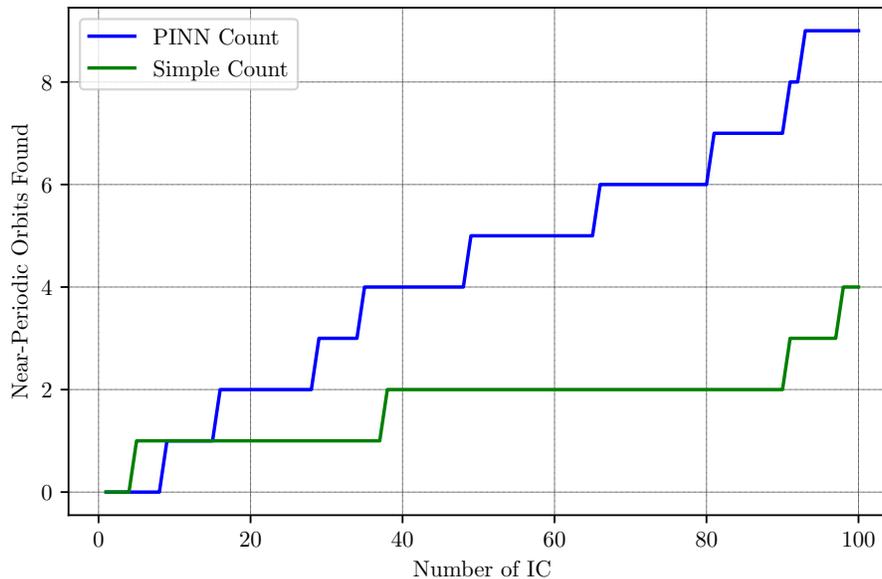
For perturbing potentials which are simple to express analytically (e.g. a J_2 model), the LPE can be expanded and solved to find initial conditions which guarantee that all orbital elements remain fixed (periodic orbits) or some elements remain fixed (frozen orbits) over a defined period. This analytical approach has been used in a variety of studies for both planetary and small-body orbiters.^{17,28}

Unfortunately the utility of expanding the LPE analytically is only apparent for these simple and low-degree disturbing potentials. As previously discussed, the potentials of small-bodies are often difficult to express in a succinct analytic form. Even when the potential is analytically expressible, identifying unique solutions to the LPE grows increasingly difficult as the potential takes on a more complex expression. As a result, this traditional method is not necessarily tractable for highly non-spherical bodies like asteroids. Instead, researchers must turn to numerical methods like multiple-shooting or collocation algorithms which pose their own unique challenges. These boundary value problem solvers often require very good initial guesses to converge — particularly when the dynamics of the problem are expensive to evaluate.¹⁷

To achieve these precise initial guesses, researchers try to blend the analytic LPE approach with the numerical solvers. Beginning with a simplified low-degree spherical harmonic model of the potential, researchers will compute analytic solutions to the LPEs using this simplified model and then use these solutions as initial guesses for the numerical solvers which are equipped with higher-degree spherical harmonic gravity models. Once a near-periodic condition is found using the solver, the researchers will increase the fidelity of the gravity model and rerun the solver with the latest near-periodic orbit as the initial condition. This process is repeated until a desired level of fidelity is reached.

Table 2: List of Initial Condition Distributions

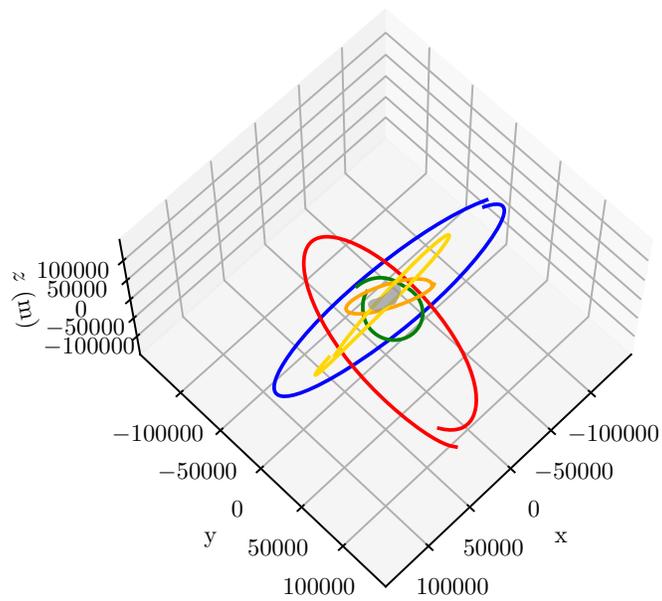
Parameter	Distribution	Parameter	Distribution
Semi-Major Axis	$\mathcal{U}[R, 10R]$	Argument of Periapsis	$\mathcal{U}[0, 2\pi]$
Eccentricity	$\mathcal{U}[0, 1]$	RAAN	$\mathcal{U}[0, 2\pi]$
Inclination	$\mathcal{U}[0, \pi]$	Mean Anomaly	$\mathcal{U}[0, 2\pi]$

**Figure 7:** Near-Periodic Orbit Solutions

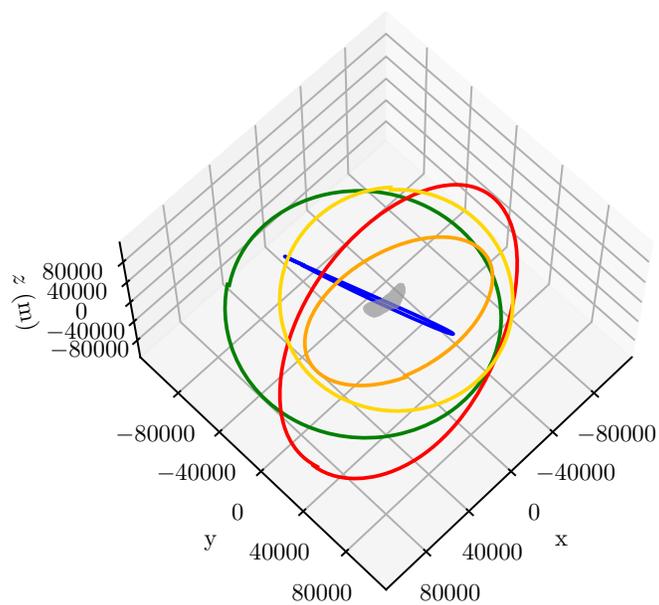
This approach has successfully identified frozen orbits about the asteroid Toutatis, but it suffers from two disadvantages. First is that it only works using a relatively sparse set of initial conditions. By requiring that the numerical solvers use solutions to the LPEs as initial guesses, the search space is immediately confined to a small set which is difficult to acquire. The second disadvantage is this approach cannot account for orbits with altitudes below the Brillouin sphere where the spherical harmonic model begins to diverge. This could be avoided if a polyhedral model could be used with the LPE, but the polyhedral form is not amenable to the LPE. Moreover, even if the polyhedral model could be used with the LPE, the model would be prohibitively expensive to evaluate as part of the numerical boundary value problem solver.

Fortunately the PINN gravity model can address both of these concerns. Because PINNs are efficient to evaluate and are stable down to the surface of the body, they can be used directly within a boundary value problem solver. A good initial guess is not even required, as the numerical solver can compensate for a poor initial guess with pure brute force iteration due to the speed of the PINN. To demonstrate, a boundary value problem solver equipped with the PINN gravity model is given 100 random initial conditions sampled from the distributions listed in Table 2 and tasked with finding near-periodic orbits. For each initial condition, the solver is given the keplerian trajectory based on those elements as an initial guess and is allowed up to 10,000 collocation points to resolve a near-periodic orbit. For the purposes of this experiment, a near-periodic orbit is defined as an orbit that returns to a 3 kilometer sphere about the initial position (approximately 1% of the permissible altitudes) with a velocity whose magnitude is deviates from the initial velocity by less than 30 centimeters per second.

Figure 7 demonstrates that the boundary value solver is able to identify 9 near-periodic orbits from the original 100 initial conditions. While this accounts for a success rate of less than 10%, it is over twice as

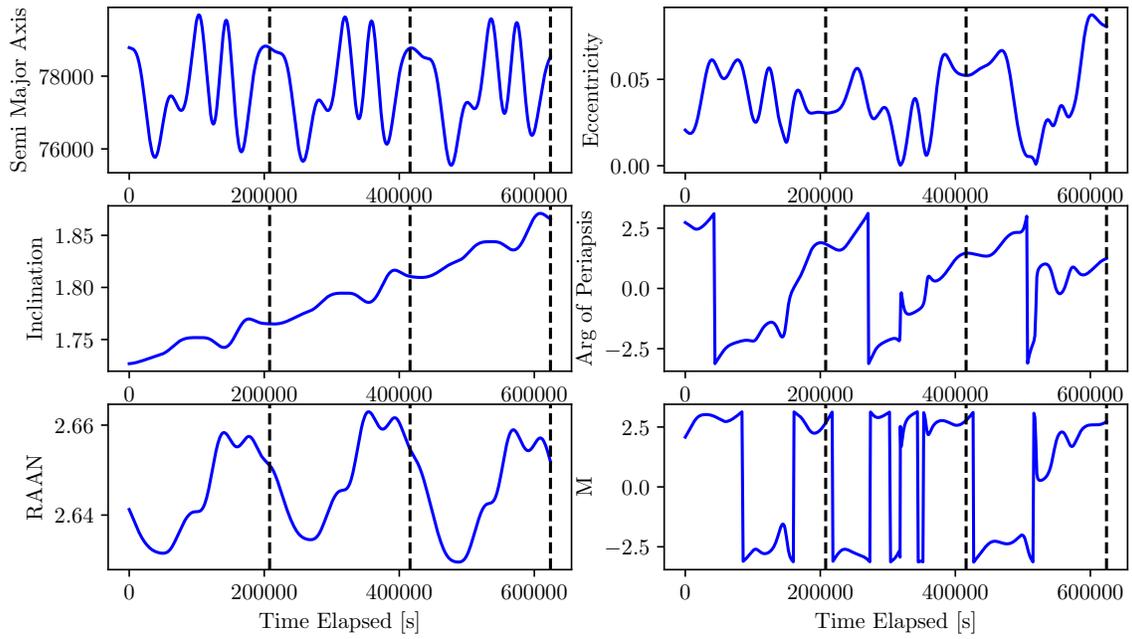


(a) Random Initial Conditions

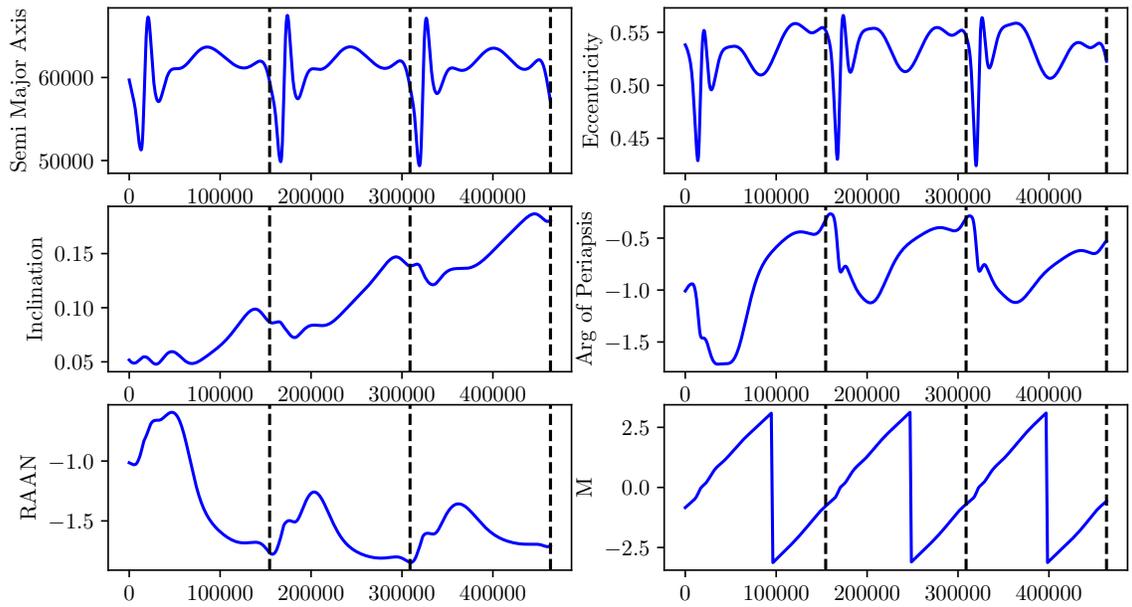


(b) Near-Periodic Initial Conditions

Figure 8: Propagated Trajectories about Asteroid 433-Eros



(a) Random Initial Condition



(b) Near-Periodic Initial Condition

Figure 9: Orbital Elements Over Time

successful than randomly guessing near-periodic initial conditions (the “Simple Count” curve). Figure 8a propagates five randomly sampled initial conditions to the point of closest approach with the initial state. In each case, the orbits diverge considerably highlighting the non-keplerian motion experienced about asteroids and the difficulty of finding these near-periodic orbits randomly. In contrast, when these random initial conditions are run through the PINN-BVP solver, the resulting orbits achieve a much tighter residuals which would require significantly less station-keeping to keep periodic. Shown differently, Figure 9 compare how the orbit elements of a PINN-BVP solution and a random initial orbit vary over the course of three periods. For the randomly sampled initial conditions, the orbital elements have both secular drift and inter-period variations (see eccentricity and mean anomaly for most egregious changes). In contrast, the orbital elements of the PINN-BVP solution achieve much lower inter-period variations, although there remains secular drift in inclination.

These results imply that the PINN-BVP solver is a powerful tool capable of identifying near-periodic orbits from arbitrary initial conditions. A separate experiment was run where the BVP solver was equipped with the polyhedral model as a backend, and it was unable to complete a single BVP iteration in the same amount of time it took all 100 initial conditions to be explored by the PINN-BVP solver.

There remain a number of promising avenues for PINN-BVP future research. Foremost, the boundary value problem solver used in this study sought solutions for which all state vector components were periodic after some period T . A potentially simpler condition to satisfy would to search for a subset of orbit elements that remain fixed on average over the period T — i.e. a frozen orbit. Given that the PINN gravity model produces an estimate of the potential that can be auto-differentiated with respect to any input, the PINN could be used directly with the LPE to determine these conditions. Moreover, the PINN gravity models trained for this study learned from a relatively small domain — i.e. data that extended only to ten times the maximum radius of the asteroid. As such, the BVP was restricted to solve only for orbits that remained within this radial bound. If this experiment was to be repeated with a PINN trained with data that extended to higher altitudes, it is possible that a greater number of near-periodic orbits could be found. Finally, there also remains the question of the stability of these near-periodic orbits. For how long do each of these orbits maintain these conditions? Is it possible that purely periodic orbits exist, but require longer time scales to repeat? Such questions are left to future work.

CONCLUSIONS

Physics-Informed Neural Network (PINN) gravity models offer an accurate and efficient means to represent the complex gravity fields of small-bodies. While there exists sufficient literature investigating the accuracy and robustness of these models, this paper provides the first discussion of how these models can be enablers for additional topics of research within the broader astrodynamics community. Specifically, this paper shows that PINN gravity models can enable substantially more productive reinforcement learning environments by providing full-fidelity dynamics at a fraction of the cost of older analytic models. This efficiency allows agents to be trained on substantially more simulations, leading to greater return and robustness. In addition, PINN gravity models can also be used in a broader trajectory design context — specifically in the search for near-periodic orbits of irregularly shaped bodies. The efficiency of the PINN enables traditional collocation boundary value problem solvers to be run substantially faster, allowing researchers to test wider trade spaces with relatively low computational costs.

REFERENCES

- [1] D. S. Lauretta, S. S. Balram-Knutson, E. Beshore, W. V. Boynton, C. Drouet d’Aubigny, D. N. DellaGiustina, H. L. Enos, D. R. Golish, C. W. Hergenrother, E. S. Howell, C. A. Bennett, E. T. Morton, M. C. Nolan, B. Rizk, H. L. Roper, A. E. Bartels, B. J. Bos, J. P. Dworkin, D. E. Highsmith, D. A. Lorenz, L. F. Lim, R. Mink, M. C. Moreau, J. A. Nuth, D. C. Reuter, A. A. Simon, E. B. Bierhaus, B. H. Bryan, R. Ballouz, O. S. Barnouin, R. P. Binzel, W. F. Bottke, V. E. Hamilton, K. J. Walsh, S. R. Chesley, P. R. Christensen, B. E. Clark, H. C. Connolly, M. K. Crombie, M. G. Daly, J. P. Emery, T. J. McCoy, J. W. McMahon, D. J. Scheeres, S. Messenger, K. Nakamura-Messenger, K. Righter, and S. A. Sandford, “OSIRIS-REx: Sample Return from Asteroid (101955) Bennu,” *Space Science Reviews*, Vol. 212, No. 1-2, 2017, pp. 925–984, 10.1007/s11214-017-0405-1.

- [2] D. Y. Oh, S. Collins, T. Drain, W. Hart, T. Imken, K. Larson, D. Marsh, D. Muthulingam, J. S. Snyder, D. Trofimov, L. T. Elkins-Tanton, I. Johnson, P. Lord, and Z. Pirkl, "Development of the Psyche Mission for NASA's Discovery Program," *36th International Electric Propulsion Conference*, Vienna, Austria, 2019.
- [3] D. Y. Oh, D. Goebel, C. Polanskey, S. Snyder, G. Carr, S. M. Collins, G. Lantoine, D. Landau, L. Elkins-Tanton, P. Lord, and S. Tilley, "Psyche: Journey to a metal world," *52nd AIAA/SAE/ASEE Joint Propulsion Conference*, 2016, Vol. 2016, 2016, pp. 1–11, 10.2514/6.2016-4541.
- [4] D. J. Scheeres, J. W. McMahon, E. B. Bierhaus, J. Wood, L. A. Benner, C. Hartzell, P. Hayne, J. Hopkins, R. Jedicke, L. LeCorre, S. Naidu, P. Pravec, and M. Ravine, "Janus: A NASA SIMPLEX mission to explore two NEO binary asteroids," *Proceedings of the International Astronautical Congress, IAC*, Vol. 2020-October, 2020.
- [5] R. A. Werner and D. J. Scheeres, "Exterior gravitation of a polyhedron derived and compared with harmonic and mascon gravitation representations of asteroid 4769 Castalia," *Celestial Mechanics and Dynamical Astronomy*, Vol. 65, No. 3, 1996, pp. 313–344, 10.1007/bf00053511.
- [6] M. C. Nolan, C. Magri, E. S. Howell, L. A. Benner, J. D. Giorgini, C. W. Hergenrother, R. S. Hudson, D. S. Lauretta, J. L. Margot, S. J. Ostro, and D. J. Scheeres, "Shape model and surface properties of the OSIRIS-REx target Asteroid (101955) Bennu from radar and lightcurve observations," *Icarus*, Vol. 226, No. 1, 2013, pp. 629–640, 10.1016/j.icarus.2013.05.028.
- [7] Y. Takahashi and D. J. Scheeres, "Morphology driven density distribution estimation for small bodies," *Icarus*, Vol. 233, 2014, pp. 179–193, 10.1016/j.icarus.2014.02.004.
- [8] D. J. Scheeres, A. S. French, P. Tricarico, S. R. Chesley, Y. Takahashi, D. Farnocchia, J. W. McMahon, D. N. Brack, A. B. Davis, R. L. Ballouz, E. R. Jawin, B. Rozitis, J. P. Emery, A. J. Ryan, R. S. Park, B. P. Rush, N. Mastrodemos, B. M. Kennedy, J. Bellerose, D. P. Luby, D. Velez, A. T. Vaughan, J. M. Leonard, J. Geeraert, B. Page, P. Antreasian, E. Mazarico, K. Getzandanner, D. Rowlands, M. C. Moreau, J. Small, D. E. Highsmith, S. Goossens, E. E. Palmer, J. R. Weirich, R. W. Gaskell, O. S. Barnouin, M. G. Daly, J. A. Seabrook, M. M. Al Asad, L. C. Philpott, C. L. Johnson, C. M. Hartzell, V. E. Hamilton, P. Michel, K. J. Walsh, M. C. Nolan, and D. S. Lauretta, "Heterogeneous mass distribution of the rubble-pile asteroid (101955) Bennu," *Science Advances*, Vol. 6, No. 41, 2020, 10.1126/sciadv.abc3350.
- [9] L. Cheng, Z. Wang, Y. Song, and F. Jiang, "Real-time optimal control for irregular asteroid landings using deep neural networks," *Acta Astronautica*, Vol. 170, No. January 2019, 2020, pp. 66–79, 10.1016/j.actaastro.2019.11.039.
- [10] A. Gao and W. Liao, "Efficient gravity field modeling method for small bodies based on Gaussian process regression," *Acta Astronautica*, Vol. 157, No. December 2018, 2019, pp. 73–91, 10.1016/j.actaastro.2018.12.020.
- [11] R. Furfaro, R. Barocco, R. Linares, F. Topputo, V. Reddy, J. Simo, and L. Le Corre, "Modeling irregular small bodies gravity field via extreme learning machines and Bayesian optimization," *Advances in Space Research*, Vol. 67, No. 1, 2021, pp. 617–638, 10.1016/j.asr.2020.06.021.
- [12] J. R. Martin and H. Schaub, "Applications of Physics-Informed Neural Networks for Gravity Field Modeling of the Earth and Moon," *Celestial Mechanics and Dynamical Astronomy*, 2022.
- [13] J. Martin and H. Schaub, "Applications of Physics-Informed Neural Networks for Gravity Field Modeling of Small Bodies," *Celestial Mechanics and Dynamical Astronomy*, 2022.
- [14] E. R. Burnett and E. A. Butcher, "Linearized relative orbital motion dynamics in a rotating second degree and order gravity field," *Advances in the Astronautical Sciences*, Vol. 167, No. September, 2018, pp. 3463–3482.
- [15] D. J. Scheeres, B. M. Sutter, and A. J. Rosengren, "Design, dynamics and stability of the OSIRIS-REx sun-terminator orbits," *Advances in the Astronautical Sciences*, Vol. 148, 2013, pp. 3263–3282.
- [16] E. R. Burnett and H. Schaub, "Study of highly perturbed spacecraft formation dynamics via approximation," *Advances in Space Research*, Vol. 67, No. 11, 2021, pp. 3381–3395, 10.1016/j.asr.2020.02.030.
- [17] D. J. Scheeres, S. J. Ostro, R. S. Hudson, E. M. Dejong, and S. Suzuki, "Dynamics of Orbits Close to Asteroid 4179 Toutatis," *Icarus*, Vol. 132, No. 1, 1998, pp. 53–79, 10.1006/icar.1997.5870.
- [18] W. Zhao, J. P. Queralt, and T. Westerlund, "Sim-to-Real Transfer in Deep Reinforcement Learning for Robotics: a Survey," sep 2020, 10.1109/SSCI47803.2020.9308468.
- [19] W. M. Kaula, *Theory of satellite geodesy: applications of satellites to geodesy*. Waltham, Mass.: Blaisdell Publishing Co, 1966.
- [20] S. Tardivel, "The limits of the mascons approximation of the homogeneous polyhedron," *AIAA/AS Astrodynamics Specialist Conference*, 2016, No. September, 2016, pp. 1–13, 10.2514/6.2016-5261.
- [21] R. S. Park, R. A. Werner, and S. Bhaskaran, "Estimating small-body gravity field from shape model and navigation data," *Journal of Guidance, Control, and Dynamics*, Vol. 33, No. 1, 2010, pp. 212–221, 10.2514/1.41585.

- [22] R. Gaskell, “Gaskell Eros Shape Model V1.1.,” 2021, 10.26033/d0gq-9427.
- [23] M. Raissi, P. Perdikaris, and G. E. Karniadakis, “Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations,” *Journal of Computational Physics*, Vol. 378, 2019, pp. 686–707, 10.1016/j.jcp.2018.10.045.
- [24] S. Wang, Y. Teng, and P. Perdikaris, “Understanding and Mitigating Gradient Pathologies in Physics-Informed Neural Networks,” *arXiv*, 2020, pp. 1–28.
- [25] T. Imken, T. Randolph, M. DiNicola, and A. Nicholas, “Modeling spacecraft safe mode events,” *IEEE Aerospace Conference Proceedings*, Vol. 2018-March, 2018, pp. 1–13, 10.1109/AERO.2018.8396383.
- [26] T. Haarnoja, A. Zhou, P. Abbeel, and S. Levine, “Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor,” *35th International Conference on Machine Learning, ICML 2018*, Vol. 5, 2018, pp. 2976–2989.
- [27] G. R. Hintz, “Survey of orbit element sets,” *Journal of Guidance, Control, and Dynamics*, Vol. 31, No. 3, 2008, pp. 785–790, 10.2514/1.32237.
- [28] M. Lara, A. Deprit, and A. Elife, “Numerical continuation of families of frozen orbits in the zonal problem of artificial satellite theory,” *Celestial Mechanics and Dynamical Astronomy*, Vol. 4, No. 3, 1995, pp. 57–71.