(Preprint) AAS 24-265

INVESTIGATING THE FUSION OF MASCON AND NEURAL NETWORKS GRAVITY MODELS

Julio C. Sanchez* **and John R. Martin†**

Accurate gravity field modeling for irregular and heterogeneous density bodies, such as asteroids, presents significant challenges. The inhomogeneous gravity field plays a crucial role in spacecraft dynamics, particularly when orbiting in low-to-medium altitudes around these bodies. To achieve a precise gravity field solution, this paper explores the fusion of mascon and physics-informed neural network (PINN) gravity models. The mascon model is a classical approach that discretizes the body into a finite number of mass elements. Conversely, the PINN model uses a deep neural network to map position coordinates into a gravitational potential, and an acceleration can be evaluated via auto-differentiation of the network. This work trains the mascon-PINN model in two stages using a position-gravity dataset. In the first stage, only the mascon distribution is regressed and then is held constant during the subsequent stage. In the second stage, the regressed mascon model is combined with a weighted PINN, where only the neural network is trained to refine the gravity solution. The mascon-PINN model effectively merges the mascon stability at high altitudes with the high accuracy of the PINNs at low altitudes. The performance of the mascon-PINN model is evaluated using test cases for the asteroid 433 Eros, with a focus on orbital applications.

INTRODUCTION

Asteroids, small rocky bodies orbiting the Sun, are known for their irregular shapes and non-uniform internal density distributions. These characteristics pose a significant challenge when attempting to accurately resolve a global gravity model within their vicinity. This issue is relevant across various mission phases, particularly when the spacecraft must operate at low altitudes where non-Keplerian gravity perturbations are most significant. It is widely acknowledged that in these gravity-dominated regimes, careful attention must be paid to the gravity field as the complex dynamics can place the spacecraft on escape trajectories or collisions paths with the body.[1] Hence, precise gravity modeling is required to ensure safe mission operations. Additionally, gravity models designed for on-board deployment to enhance spacecraft autonomy must be computationally lightweight and memory efficient.

The most fundamental method to evaluate gravity involves integrating the gravitational pull of each infinitesimal volume element of a body. However, this approach is seldom used in practice due to the typically unknown internal density distribution of the body. Traditionally, astrodynamics practitioners have modeled gravity using a spherical harmonics series expansion,[2] which efficiently encodes spatial variations in the gravity field, especially over long wavelengths. Nevertheless, spherical harmonics have a significant drawback: the divergence within the body's circumscribing sphere,

*Associate Professor, Department of Aerospace Engineering and Fluid Mechanics, University of Seville, Sevilla, Spain, 41008.
†Assistant Professor, Department of Aerospace Engineering, University of Maryland, College Park, Maryland, 20742.

a problem that becomes pronounced for elongated or irregularly shaped bodies like asteroids. A widely adopted alternative is the polyhedral model,[3] which avoids divergence in this regime and better captures the dynamics induced by these irregular geometries. However, this model assumes constant density, which may not be valid, and requires iterating over all polyhedron edges and faces, leading to computational burdens for high-fidelity models.

Discretized approaches, such as mascon (mass concentrations) models, offer a potential remedy. Initially developed to model large mass anomalies of the Moon,[4] mascon models aggregate the gravitational contributions of distributed, discretized mass volumes — often represented as point masses. These mass distributions can be prescribed, such as arranging layers of point masses at the centroids of discretized elements,[5] or regressed to best match an available dataset.[6–10] Notably, solving the non-linear optimization of point mass placement significantly impacts accuracy,[7–9] particularly with a low number of masses (< 500). Moreover, even when careful attention is paid to the regression, mascon models may place point-masses near the surface which introduce large errors near it. To minimize the impact of these errors and continue learning intricate gravity patterns, this study turns to scientific machine learning to offer a potential remedy.

Machine learning has revolutionized multiple fields within science and engineering during the last decade. As it relates to gravity modeling, multiple efforts have emerged to leverage scientific machine learning. At a high-level these efforts seek to regress a mapping between position and acceleration using tools like extreme learning machines,[11] traditional neural networks,[12] physics-informed neural networks,[13] and even neural density fields.[14] Of these models, one of the matured approaches is the use of physics-informed neural networks (PINNs)[15] which provide a powerful framework to model solutions of complex differential equations from data alone. Martin and Schaub[13,16,17] have leveraged PINNs to solve the gravity modeling in small-body environments, most recently having introduced their third generation PINN gravity model (PINN-GM-III).[13] This latest model introduces a variety of modifications to maximize modeling accuracy and reliability. One particular modification of note is the PINN-GM-III's ability to seamlessly fuse the neural gravity model with an analytic point-mass model. This fusion guarantees that the hybrid model maintains reliability even beyond the bounds of the training set by ensuring a smooth transition to a reliable gravity model when the network grows uncertain due to the absence of data.

The present study aims to further develop the concept of fusing neural gravity representations with analytic models, but instead focuses attention on the use of mascon gravity models. In comparison to the single point-mass model utilized in Ref. 13, a regressed mascon model offers a precise initial approximation of the asteroid's gravity field. This diminishes the neural network's burden of learning well-established behaviors through analytic models thus allowing the PINN to focus on deciphering the complex patterns of the gravity field. For instance, lightweight mascon models (< 500 masses) are able to maintain accuracy within the Brillouin sphere until very close to the surface but struggle to grasp complex behaviours even if they are regressed with abundant data. Hence, it proves beneficial to enhance the regressed mascon with a PINN that introduces additional basis functions able to approximate the remaining intricate patterns. The mascon-PINN setup uses the principle of superposition found in the PINN-GM-III model in Ref. 13. In particular, the present work expresses the gravitational potential $U = U_M + w_{\text{NN}} U_{\text{NN}}$ where w_{NN} is a weight function, dependent on the orbital radius, which smoothly deactivates the neural network's contribution. Differently from Ref. 13, where a second weight deactivates the analytic contribution at low altitudes, the previous approach allows to independently consider the mascon and PINN contributions.

Besides accuracy, one of the main concerns for astrodynamics practitioners is the ease of use of

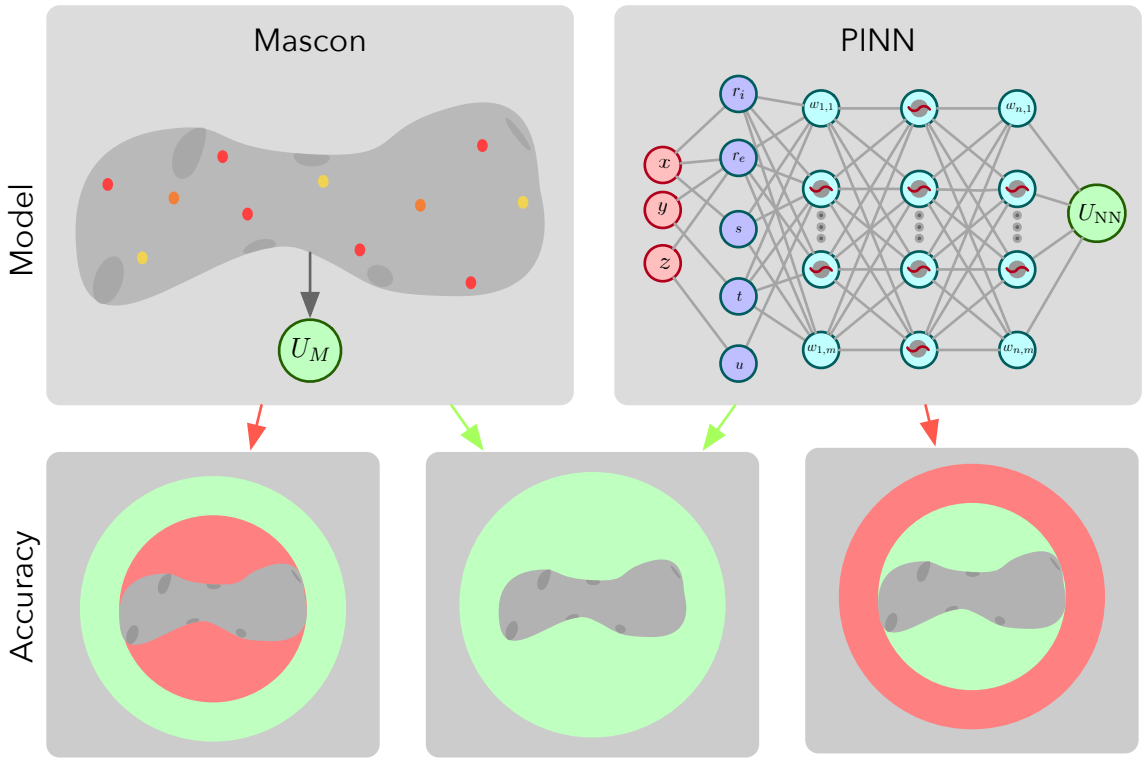

Figure 1. Fusing analytic and numerical gravity models yields superior performance.

gravity models. Traditional models like spherical harmonics, mascon, and polyhedron are widely used because their implementation is well-established, are included in most astrodynamics frameworks, and can be fused with or transformed into one another. To demonstrate that the mascon-PINN model also offers seamless integration and usability, we have incorporated it into the Basilisk[‡] (BSK) astrodynamics simulation framework.[18] Due to our choice of the gravitational potential form, the PINN model can be easily superposed to the mascon model in the BSK framework. Currently, the PINN is loaded into Basilisk via a previously generated PyTorch file, which stores a Python object with methods to evaluate the gravitational potential and acceleration. The simulations in this paper are conducted using this tool.

The structure of this manuscript is as follows: First, an overview of existing gravity models is presented. Next, the fused mascon-PINN gravity model is introduced. Following this, the training process of the mascon-PINN model is detailed. To assess the effectiveness of the mascon-PINN model, a high-fidelity asteroid ground truth model is defined. The numerical results include direct evaluations of gravity field accuracy, computational efficiency and a set of orbital application tests. Finally, the manuscript concludes with a discussion of conclusions and future work.

[‡]https://hanspeterschaub.info/basilisk/

GRAVITY MODELS

By assuming the body's shape and density distribution are stationary, the gravitational acceleration, $\mathbf{a}$, on an exterior evaluation point, $\mathbf{r}$, can be expressed as

$$\mathbf{a} = -G \int_V \rho \frac{\mathbf{r} - \mathbf{r}_V}{\|\mathbf{r} - \mathbf{r}_V\|^3} dV \tag{1}$$

where G is the gravitational constant, V is the body's volume, ρ refers to the density field and $\mathbf{r}_V$ evaluates each point within V. Although Eq. (1) is the most fundamental gravity description, the body's internal density field $\rho(\mathbf{r}_V)$ is usually not available in practice. Moreover, the evaluation of a volume integral is computationally cumbersome which may preclude on-board execution. Due to the previous facts, alternate gravity models based on certain simplifying assumptions are preferred in astrodynamics. Some of these models, such as polyhedron, mascon, and neural network models, are used throughout this work and are described below. For the sake of brevity, the popular spherical harmonics model is omitted in the discussion as it is not used along this paper.

Polyhedron gravity

The polyhedron gravity model is conceived in Ref. 3 to be an accurate model for small bodies. It assumes the body has a constant density and its shape is that of a polyhedron. Then, the exterior gravity evaluation resorts to the following formula

$$\mathbf{a} = \frac{\mu}{V} \left(- \sum_{e \in \text{edges}} \mathbf{E}_e \cdot \mathbf{r}_e L_e + \sum_{f \in \text{faces}} \mathbf{F}_f \mathbf{r}_f w_f \right) \tag{2}$$

where μ is the body standard gravity parameter, V is the body volume, $\mathbf{r}_e$ is the relative position of the evaluation point with respect to the origin of an edge, $\mathbf{E}_e$ is the dyad product resulting from an edge and its face normal, L_e is the 1D wire gravity potential of an edge, $\mathbf{r}_f$ is the relative position of the evaluation point with respect to a vertex on a face, $\mathbf{F}_f$ is the outer product of the face normal vector and w_f is the solid angle of the face as viewed from the evaluation point. The explicit details of these terms can be consulted in Ref. 3. The constant density can be easily obtained as $\rho = \mu G / V$.

An evident drawback of the polyhedron model is its constant density assumption, which reduces accuracy for bodies with heterogeneous density distributions. Another weakness is the need to iterate through all edges and faces of the polyhedron for gravity evaluation, making it computationally intensive when high-resolution shape models are used. In addition to gravity, the polyhedron also represents a shape model, providing formulas to compute volume and determine whether a point is within the polyhedron, as detailed in Ref. 3. These features are useful for validation purposes.

Mascon gravity

The mascon model was introduced in the 1970s as an alternative to spherical harmonics, designed to fit sparse data and represent large mass anomalies.[4] It describes the gravity field by summing the individual contributions of a finite number of volume elements, known as mascons, effectively resembling a discretized version of Eq. (1). For simplicity, mascons are typically modeled as point masses thus the mascon gravity is expressed as

$$\mathbf{a} = -\sum_{k=0}^{n_M} \mu_k \frac{\mathbf{r} - \mathbf{r}_k}{\|\mathbf{r} - \mathbf{r}_k\|^3} \tag{3}$$

where μ_k and $\mathbf{r}_k$ are, respectively, the standard gravity parameter and position of each point mass. The number of point masses is $n_M + 1$. The mascon model avoids spherical harmonics Brillouin sphere divergence which makes it suitable for small bodies. Nonetheless, large errors may arise when point-masses are placed near the surface.

Neural network gravity

Recently neural networks have been proposed to represent gravity fields. The first efforts[11,12] on the matter proposed to directly describe the acceleration field as

$$\mathbf{a} = \mathbf{f}_{\text{NN}}(\mathbf{r}) \tag{4}$$

where $\mathbf{f}_{\text{NN}}$ is an artificial neural network which maps position coordinates to gravity. Additional research by Martin and Schaub[16,17] proposed to incorporate physics knowledge into the network design through the use of Physics-Informed Neural Networks, or PINNs. These authors use the neural network to represent the gravity potential, rather than the acceleration field, and differentiate it via automatic differentiation to obtain the acceleration field as

$$\mathbf{a} = \nabla U_{\text{NN}}(\mathbf{r}) \tag{5}$$

By learning the gravity potential, the network is demonstrated to be more accurate at generalizing samples outside the dataset and requires fewer network parameters than competing alternatives. The initial gravity PINN versions, named PINN-GM-I[16] and PINN-GM-II,[17] learn a differentiable form of the potential along with a penalty term for the Laplacian. While these versions are very accurate within the training dataset region, they are unreliable in areas without data. The new PINN-GM-III[13] addresses this issue by fusing an analytic model with the PINN, smoothly transitioning from a neural network solution to a point mass approximation as the model extends into domains lacking data, thereby forming a global gravity solution. This ensures that the gravity solution satisfies important boundary conditions, although the choice of a point mass approximation in the limit remains sub-optimal. This strategy is further detailed in the next section, as it is the foundational concept behind the proposed mascon-PINN model.

FUSION OF MASCON AND NEURAL NETWORK GRAVITY

The main contribution of this work is the fusion of mascon and neural network gravity models to obtain a global gravity solution. The idea is to use a previously regressed mascon model to be further refined by a neural network. The mascon model ensures an accurate starting point for the neural network training which can effectively focus its computational power on approximating the very complex gravity field features. The fusion strategy extensively leverages the law of superposition: the fact that multiple potentials can be added together to yield a global potential.

In Ref. 13, the law of superposition is exploited to add extrapolation robustness to the gravity PINN. They introduce a boundary potential U_{BC} that smoothly contributes fully to the gravity past a certain altitude. This strategy deactivates the PINN in the dataless areas thus avoiding unreliable results at high altitudes. Reference 13 uses a weighted potential between the boundary model and the PINN as

$$U(\mathbf{r}) = w_{\text{BC}}(r)U_{\text{BC}}(\mathbf{r}) + w_{\text{NN}}(r)U_{\text{NN}}(\mathbf{r}) \tag{6}$$

The weights w_{BC} and w_{NN} only depend on the radial coordinate $r = \|\mathbf{r}\|$. To deactivate the PINN when $r \to \infty$, Ref. 13 proposes an hyperbolic tangent function as

$$w_{\text{NN}}(r) = 1 - H(r), \quad H(r) = \frac{1 + \tanh(K(r - r_{ref}))}{2} \tag{7}$$

where r_{ref} places the zero-crossing of the hyperbolic tangent and K controls the smoothness of the transition from activation to deactivation.

The weight w_{BC} to the boundary model is a topic on its own. When the boundary model is unreliable at low altitudes (e.g. point-mass or spherical harmonics) it could be convenient to force its deactivation as $w_{\text{BC}} = H(r)$. This also admits further refinement as it is explored in Ref. 13 which adds an additional deactivation term to the point-mass model. However, this paper uses the mascon model which is fairly accurate within the Brillouin sphere though it may introduce large errors near the surface. Accordingly we choose $w_{\text{BC}} = 1$ to maintain the base mascon model as it is thus ensuring a seamlessly addition of its contribution to the total gravity. Then, the fused mascon-neural network gravity potential is

$$U(\mathbf{r}) = \sum_{k=0}^{n_M} \frac{\mu_k}{\|\mathbf{r} - \mathbf{r}_k\|} + (1 - H(r))U_{\text{NN}}(\mathbf{r}) \tag{8}$$

and the gravity evaluation is

$$\hat{\mathbf{a}}(\mathbf{r}) = \underbrace{-\sum_{k=0}^{n_M} \mu_k \frac{\mathbf{r} - \mathbf{r}_k}{\|\mathbf{r} - \mathbf{r}_k\|^3}}_{\text{mascon contribution}} + \underbrace{\nabla[(1 - H(r))U_{\text{NN}}(\mathbf{r})]}_{\text{PINN contribution}} \tag{9}$$

TRAINING OF THE MASCON-PINN MODEL

The training of the fused mascon-PINN model comprises two separate stages. First, the mascon model is regressed and its parameters are kept constant. Second, a neural network is attached to the mascon model following Eq. (8)-(9) and the network weights and biases are learned through training. In these processes, the main assumption is that a dataset of n position and gravity acceleration samples is available as

$$\mathbf{r}_{\text{data}} = \begin{bmatrix} \mathbf{r}_1 \\ \vdots \\ \mathbf{r}_n \end{bmatrix}, \quad \mathbf{a}_{\text{data}} = \begin{bmatrix} \mathbf{a}(\mathbf{r}_1) \\ \vdots \\ \mathbf{a}(\mathbf{r}_n) \end{bmatrix} \tag{10}$$

Mascon regression

The mascon regression aims to determine the point-mass mascon distribution, characterized by masses values μ_k and planetocentric positions $\mathbf{r}_k = [x_k, y_k, z_k]^T$ with $k = 0 \dots n_M$, that best fits the dataset in Eq. (10). For the sake of brevity. the term mass is extensively used to denote the standard gravity parameter in the context of the mascon model. The mascon parameters arise in the mascon gravity evaluation as per Eq. (3). Following Ref. 9 the mascon distribution is regressed by means of gradient descent with the enforcement of physical constraints. This means that masses are required to be positive and within the body's shape. Since gradient descent is to be used, an initial mascon distribution is required to start with.

One of the main assumptions to the mascon regression is the availability of a shape model with an expression to check if masses are within the body's shape. In that sense, the polyhedron model of Ref. 3 provides the following conditions for a point to be exterior or interior

$$\begin{cases} \mathbf{r} \in V & \text{if } \sum\limits_{f \in \text{faces}} w_f(\mathbf{r}) = 0 \\ \mathbf{r} \notin V & \text{if } \sum\limits_{f \in \text{faces}} w_f(\mathbf{r}) = 4\pi \end{cases} \tag{11}$$

where w_f is the solid angle of each face as seen by the evaluation point. The Eq. (11) is used throughout this paper to discriminate interior and exterior points to the asteroid.

Initial distribution: An initial mascon distribution $\mu_k^{[0]}$ and $\mathbf{r}_k^{[0]}$ is required to start the regression process. Each mass is initialized with the same value $\mu_k^{[0]} = \mu/(n_M + 1)$. Their positions are distributed randomly while ensuring a certain level of sparsity between them as follows: the 0th mass point is placed at the origin, $\mathbf{r}_0^{[0]} = \mathbf{0}$, by default; the rest of the n_M masses are divided equally within the eight octants of the body's shape (please refer to Ref. 9 for more details). Note that at most there is a difference of one point mass between the octants if n_M is not divisible by eight. For each octant, the masses positions are random and are ensured to be within the body's shape $\mathbf{r}_k^{[0]} \in V$.

Physical constraints: When using data regression to find the optimal mascon distribution, the physical consistency of the solution is not directly ensured. In particular we desire a distribution with positive masses, $\mu_k \geq 0$, and positions contained within the body shape $\mathbf{r}_k \in V$. To reduce the number of multiple solutions, the total mass is also enforced as $\sum\limits_{k=0}^{n_M} \mu_k = \mu$.

To force all masses to be positive, the decision variable μ_k can be transformed to its square-root as $\sqrt{\mu_k}$. The total mass compliance can be directly encoded by expressing the 0th mass as a function of the remaining n_M ones

$$\mu_0 = \mu - \sum_{k=1}^{n_M} \mu_k \tag{12}$$

Still the 0th mass can undergo a negative value. A mitigation could be to track the 0th mass during training and decrease the remaining n_M masses in the same proportion if the 0th mass becomes negative

$$\text{If} \quad \mu_0 < 0 \quad \text{then} \quad \mu_k \leftarrow \mu_k \frac{\mu}{\sum\limits_{k=1}^{n_M} \mu_k} \tag{13}$$

Ensuring all masses are within the body's shape follows the same previous logic of the 0th mass. This technique is well known in optimization and consists in projecting the actual unfeasible solution to the closest point within the feasible set. In order to keep masses contained to the body's shape we can detect if one goes outside of it with Eq. (11) and project that mass to the closest point in the polyhedron surface ∂V as

$$\text{If} \quad \mathbf{r}_k \notin V \quad \text{then} \quad \mathbf{r}_k \leftarrow \underset{\mathbf{r} \in \partial V}{\arg\min} \|\mathbf{r} - \mathbf{r}_k\| \tag{14}$$

Gradient descent: The mascon regression algorithm uses Adam gradient descent to find the mascon distribution (μ_k, $\mathbf{r}_k$) for $k = 1 \dots n_M$ that best fits a loss function based on the dataset of Eq. (10). To design the loss function, absolute, δa_{ABS}, and relative, δa_{REL}, errors are defined as

$$\delta a_{\text{ABS},j} = \|\hat{\mathbf{a}}(\mathbf{r}_j) - \mathbf{a}_j\|, \quad \delta a_{\text{REL},j} = \frac{\|\hat{\mathbf{a}}(\mathbf{r}_j) - \mathbf{a}_j\|}{\|\mathbf{a}_j\|} \tag{15}$$

where $\hat{\mathbf{a}}(\mathbf{r}_j)$ is the model gravity prediction and $\mathbf{a}_j$ is the truth value. Relative errors are especially useful for validation as one would like to know the prediction accuracy regardless of the absolute value. One could just use relative errors to design the loss but it is also convenient to introduce some prioritization for large gravity values. Consequently, we consider the following mean squared error (MSE) loss function to train the mascon distribution

$$\mathcal{L}_{\text{MSE}} = \frac{1}{n} \sum_{j=1}^{n} \left(\delta a_{\text{REL},j}^2 + \frac{\delta a_{\text{ABS},j}^2}{\bar{a}^2} \right) \tag{16}$$

where $\bar{a}$ is a normalization factor to have similar order of magnitudes in the relative and absolute terms. In this work, the variable $\bar{a}$ is chosen as the maximum gravity acceleration of the entire dataset.

The training decision variables are also normalized to avoid extreme numerical differences due to unit dimensions. In particular, we normalize masses μ_k and their position coordinates $\mathbf{r}_k = [x_k, y_k, z_k]^T$ as

$$\bar{\mu}_k = \frac{\mu_k}{\bar{\mu}}, \quad \bar{x}_k = \frac{x_k}{a}, \quad \bar{y}_k = \frac{y_k}{b}, \quad \bar{z}_k = \frac{z_k}{c} \tag{17}$$

where $\bar{\mu} = \mu/(n_M + 1)$ and (a, b, c) are the principal dimensions of the body's ellipsoid approximation. This normalization is very convenient for elongated asteroids beause it scales each direction according to its relative size.

Neural network training

Once the mascon distribution is regressed its parameters are fixed and a PINN model is attached according to Eq. (8)-(9). The goal now is to train the neural network weights and biases with the same dataset as per Eq. (10). For the neural network training this paper follows the fundamental results of Martin and Schaub works related to gravity PINNs which explored impact of feature engineering, network architectures and suitable loss functions on neural network gravity modeling performance.[13,16,17]

Feature engineering: A neural network maps inputs, $\mathbf{x}_{\text{NN}}$, to outputs, $\mathbf{y}_{\text{NN}}$, as $\mathbf{y}_{\text{NN}} = \mathbf{f}(\mathbf{x}_{\text{NN}})$. Feature engineering processes raw inputs and outputs into a more effective set that boost neural network training capabilities. For example, it is well-known that neural networks inputs-outputs work well in the $[-1, 1]$ interval where activation functions vary meaningfully. Therefore, it is convenient that gravity PINN inputs-outputs are engineered to exist within that interval. Moreover, it is often convenient to encode the inputs into features that strong correlate with variations in the gravity field. To accomplish this, this work follows the PINN-GM-III feature engineering scheme as detailed below.

The neural network input features are transformed from Cartesian coordinates (expressed in the

planetocentric rotating frame) $\mathbf{r} = [x, y, z]^T$ to network inputs $\mathbf{x}_{\text{NN}} = [\bar{r}_i, \bar{r}_e, s, t, u]^T$ as

$$\bar{r}_i = \begin{cases} \dfrac{r}{R} & \text{if } r \in [0, R), \\ 1 & \text{if } r \in [R, \infty), \end{cases} \qquad \bar{r}_e = \begin{cases} 1 & \text{if } r \in [0, R), \\ \left(\dfrac{r}{R}\right)^{-1} & \text{if } r \in [R, \infty), \end{cases} \qquad \begin{bmatrix} s \\ t \\ u \end{bmatrix} = \frac{1}{r} \begin{bmatrix} x \\ y \\ z \end{bmatrix} \tag{18}$$

where $r = \sqrt{x^2 + y^2 + z^2}$ and R is the radius corresponding with the Brillouin sphere. Accordingly, the features $\bar{r}_i$ and $\bar{r}_e$ capture the Brillouin sphere's interior and exterior in a distinctive manner, varying along the positive interval as $\bar{r}_i \in [\min_{\mathbf{r} \in \partial V}(\|\mathbf{r}\|/R), 1]$ and $\bar{r}_e \in (0, 1]$. The features s, t and u represent the sines of the evaluation point with respect to the Cartesian axes. These features vary in the desired interval as $s, t, u \in [-1, 1]$.

The network output feature y_{NN} is a scalar also subject to engineering. Due to the prior mascon regression, the PINN is trying to learn the gravity potential error (see Eq. (8)) between the mascon and the ground truth model. In that scenario, test samples with equal relative errors could be subject to a large difference in absolute terms due to the fact that $U \to 0$ when $r \to \infty$. To effectively scale the network output and mitigate the potential decay with altitude, y_{NN} is designed as a proxy potential[13] which can be converted to the true potential through

$$U_{\text{NN}} = \left(\frac{R}{r}\right)(U^*)\, y_{\text{NN}} \tag{19}$$

The normalization is done with $U^* = \max(|U - U_M|)$ which represents the largest potential absolute error of the mascon model U_M to be corrected. The scaling with the radial distance R/r mitigates the potential decay of $U \to 0$ when $r \to \infty$ and keeps the network learning gravity differences (other than the point-mass contribution) at medium to high altitudes.

Fused mascon-PINN potential: During the mascon-PINN training, the neural network is forwarded and auto differentiation is applied to obtain the gradient of the fused potential with respect to input position coordinates. In that sense it is required to work with the following fused mascon-PINN potential during training

$$U(\mathbf{r}) = \sum_{k=0}^{n_M} \frac{\mu_k}{\|\mathbf{r} - \mathbf{r}_k\|} + (1 - H(r)) U_{\text{NN}}(\mathbf{r}) \tag{20}$$

where U_{NN} is the physical network potential which relates to the network output as per Eq. (19). A gravity prediction is done by autodifferentiating Eq. (20) as $\mathbf{a}(\mathbf{r}) = \nabla U(\mathbf{r})$. In evaluation mode, only the PINN contribution is autodifferentiated since accounting for the mascon component with a computational graph is slow when n_M is large.

Network architecture: The recommended PINN architecture is that of a multilayer perceptron with l hidden layers with m neurons per layer. We will use the notation $l \times m$ to compactly express the capacity of the network. The number of network weights can be obtained as $n_w = n_{in}m + (l-1)m^2 + n_{out}m$ and biases as $n_b = lm + n_{out}$ where $n_{in} = 5$ and $n_{out} = 1$ in the case under consideration. The total number of network parameters is $n_{\text{NN}} = n_w + n_b$.

The activation function is another key component of a neural network. For a PINN, there is a need to do a gradient operation to the network hence functions with non-constant derivatives are required (e.g. ReLU is a bad choice). In this work, SIREN (sinusoidal activation) is used

$$y_{ij} = \sin\left(\sum_{j=1}^{m} w_{ij} y_{i-1,j} + b_{ij}\right) \tag{21}$$

where the index i refers to the i-th hidden layer and j to the j-th neuron in the layer. Another choice for a gravity PINN activation could be the GeLU (Gaussian error linear units) function.

Gradient descent: Like the mascon distribution, neural networks are also trained using gradient descent to minimize some chosen loss function. For gravity PINNs, previous works[13] have demonstrated the effectiveness of a loss aiming mean absolute errors (MAE) instead of mean squared ones. Then, the PINN loss is defined to minimize mean absolute and relative gravity errors as

$$\mathcal{L}_{\text{MAE}} = \frac{1}{n}\sum_{j=1}^{n}\left(\delta a_{\text{REL},j} + \frac{\delta a_{\text{ABS},j}}{\bar{a}}\right) \tag{22}$$

Note that by their definition in Eq. (15), the relative and absolute gravity errors are always positive as $\delta a_{\text{REL}} \geq 0$ and $\delta a_{\text{ABS}} \geq 0$.

GROUND TRUTH GRAVITY

The chosen test case asteroid is 433 Eros which is an elongated shape object with abundant data obtained by NEAR mission. The asteroid 433 Eros has a sidereal period of 5.27 h and a standard gravity parameter of $\mu = 4.46275 \cdot 10^5$ m^3/s^2. Eros polyhedron shape models are publicly available[§]. In particular, the high resolution model with 200,700 faces is used as ground truth model in this paper. Eros ellipsoidal approximation can be roughly quantified as $a = 16.342$ km, $b = 8.410$ km and $c = 5.973$ km. However, the normalization radius $R = 16$ km is widely used as the Brillouin sphere boundary.

Two scenarios are studied in this manuscript. The first one corresponds to the usual constant density polyhedron case. The second one adds two perturbing masses to the constant density polyhedron to create an asymmetry in the gravity field (namely heterogeneous polyhedron). One of the masses is positive as $+0.1\mu$ and is placed at $[10, 0, 0]^T$ km while the remaining one is negative as -0.1μ and is placed at $[-10, 0, 0]^T$ km. The gravity magnitude, $a_{con} = \|\mathbf{a}\|$, in Eros equatorial plane xy for each scenario is reported in the left panel of Fig. 2. The heterogeneous polyhedron relative difference, $(a_{het} - a_{con})/a_{con}$, with respect to the constant density magnitude is also depicted in the right panel of Fig. 2. It can be clearly observed that the constant density gravity level curves tend to resemble Eros shape. The heterogeneous polyhedron breaks that pattern by decreasing the gravity magnitude on Eros left-side and increasing it on the right-side as expected.

For each scenario under consideration, the training dataset consists of 100,000 samples randomly distributed at altitudes between $0-50$ km $(0-3.125R)$. The test set can be generated independently of the training one to evaluate regions of interest like the surface or high altitudes. Specifically, a position-gravity test set of 61,246 samples ranging from the surface to $10R$ $(160$ km$)$ is defined to evaluate the exterior gravity field solution. An additional test set of 200,000 samples is defined on the surface to assess the errors in the most challenging region to approximate.

NUMERICAL RESULTS

The numerical results assess the mascon-PINN gravity field solutions for both constant density and heterogeneous polyhedron models of Eros. This section includes a direct evaluation of the global gravity field accuracy and an analysis of the model's performance in astrodynamics applications, such as orbital propagation and ballistic transfer design. Comparisons with standalone mascon gravity models are provided throughout the section.

[§]`https://sbnarchive.psi.edu/pds3/near/NEAR_A_5_COLLECTED_MODELS_V1_0/data/msi/`

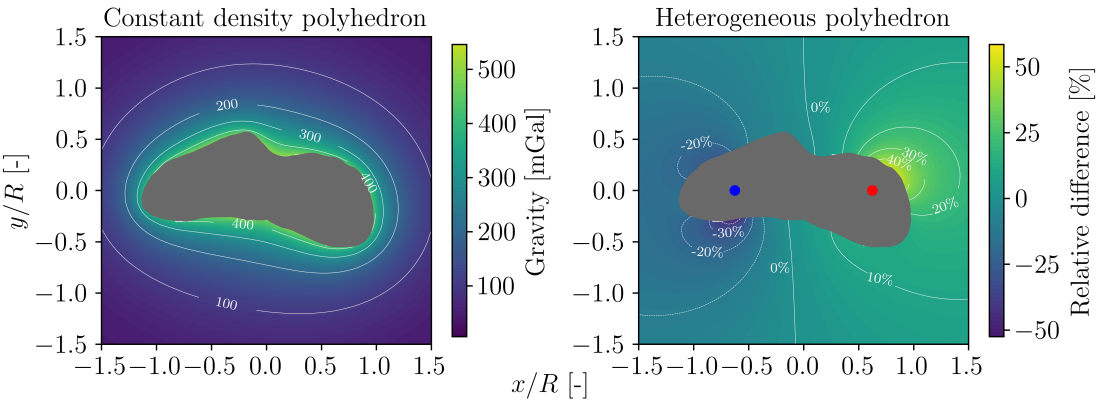

Figure 2. Gravity magnitude in Eros equatorial plane for constant density polyhedron (*left*) and the relative difference induced by the heterogeneous polyhedron (*right*).

	Mascon	PINN
Optimizer	Adam	Adam
Loss	MSE	MAE
Mini-batch size	10,000	2,000
Epochs	5,000	5,000
Learning rate	10^{-3}	10^{-3}
Activation function	-	SIREN
Initialization	$\mu_k^{[0]} \leftarrow \mu/(n_M+1)$ $\mathbf{r}_k^{[0]} \leftarrow$ random per octant	Xavier Glorot

Table 1. Mascon-PINN training hyperparameters.

The training hyperparameters of the mascon-PINN gravity model are reported in Table 1. The PINN weight w_{NN} (see Eq. (7)) parameters are chosen ad hoc as $r_{ref} = 70$ km and $K = 1$. This is motivated by $\max(r_{\text{data}}) \approx 67.5$ km thus being convenient to deactivate the PINN beyond that boundary. Finally, a unit value of K ensures a smooth deactivation of the PINN model hence avoiding loss of accuracy by gradient explosion.

Global gravity accuracy

The mascon-PINN global gravity accuracy is analyzed for several number of model parameters. Specifically the mascon number of masses is varied as $n_M = \{8, 20, 50, 100, 1,000\}$ for a 6x40 SIREN PINN. Alternatively, the case where the mascon masses are kept fixed to $n_M = 100$ and the PINN neurons per layer varies as $\{10, 20, 40, 80, 160\}$ is also studied.

Constant density polyhedron: The results for Eros constant density polyhedron are presented in Fig. 3-5. Figures 3-4 show the evolution of average gravity error (at intervals of $0.1R$) with respect to altitude by varying the number of masses and neurons per layer respectively. In Figure 3, there is a clear correlation between increased accuracy (especially at low to medium altitudes, $0 - 3R$) and a higher number of masses (n_M). Interestingly, the difference in accuracy becomes

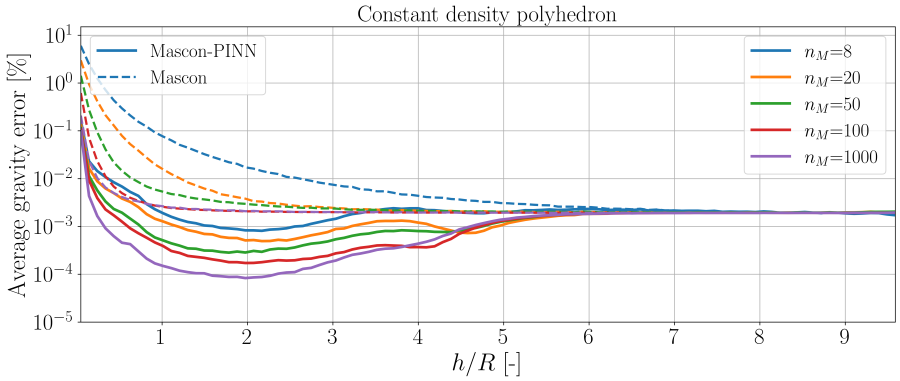

Figure 3. Average gravity error with respect to altitude of constant-density polyhedron by varying the number of masses for a SIREN 6x40 network.

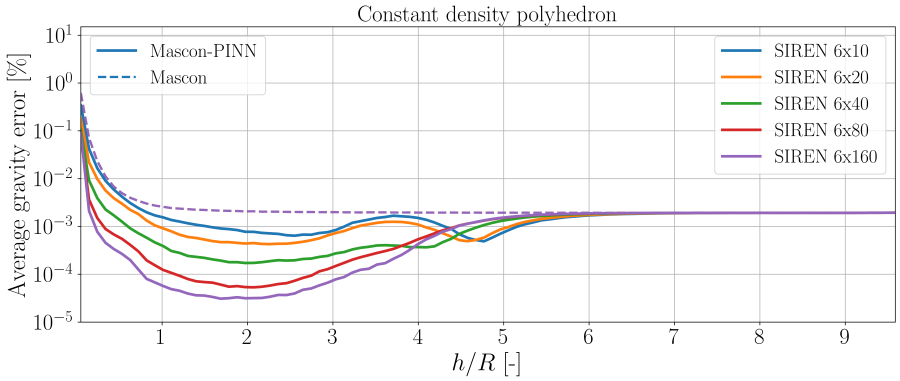

Figure 4. Average gravity error with respect to altitude of constant-density polyhedron by varying the number of neurons for $n_M = 100$.

less pronounced between 100 and 1,000 masses, with almost all models converging to the same accuracy at altitudes over $6R$. The mascon-PINN models have improved the global accuracy for all studied cases, demonstrating that even the model with only 8 masses fused with the PINN SIREN 6x40 is as competitive as the heavy mascon model with 1,000 masses. Figure 4 also shows a correlation between global accuracy and the number of neurons per layer. Even with a few neurons, the PINN can learn features unseen by the mascon model. These figures reveal a transition period around an altitude of approximately $4.5R$, due to the weight w_{NN}, characterized by $r_{ref} = 70$ km ($4.375R$). Nonetheless, the errors in that region remain below than those of the standalone mascon model.

Lastly, Fig. 5 shows the gravity error evaluated on the asteroid surface for a mascon model with $n_M = 100$ and its corresponding fused PINN with 6x40 SIREN layers and neurons. The latitude (ϕ) and longitude (λ) are ellipsoidal and related to Cartesian coordinates as $x = as\cos\lambda\cos\phi$, $y = bs\sin\lambda\cos\phi$, and $z = cs\sin\phi$. The variable s represents ellipsoids of different size but equal aspect ratio. The ellipsoidal bi-dimensional projection is not exact due to the asteroid irregular shape but is less distorted than a spherical one. Going back to Fig. 5, it can be observed that the mascon-PINN significantly reduces surface errors overall from 1.72% to 0.61% on average. However, the mascon induces a large error of approximately 10%, which is difficult for the PINN to mitigate. This may suggest that the choice of $w_{\text{BC}} = 1$ in Eq. (6) should be revisited in the future.

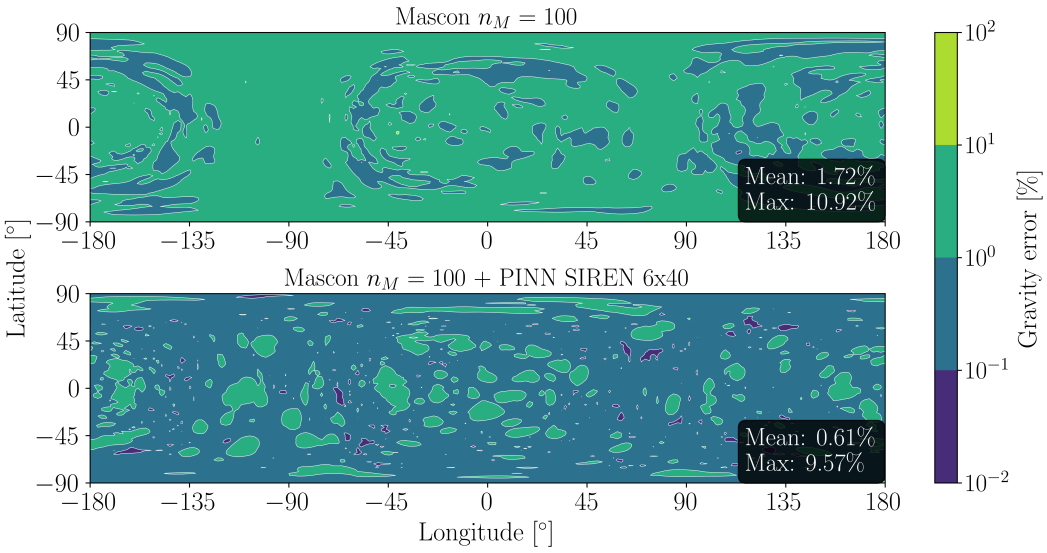

Figure 5. Gravity error on the surface (ellipsoidal coordinates) for the constant density polyhedron.

Heterogeneous polyhedron: The results for Eros heterogeneous polyhedron are shown in Fig. 6-8. The accuracy trends with respect to the number of masses and neurons (see Fig. 6-7) are equivalent to the constant density polyhedron. It is notable that this scenario is more challenging for the mascon regression since the accuracy is lower with respect to the constant density polyhedron. Specifically, as shown in Fig. 6, a distribution with at least 20 masses is required to converge to the heavier mascon models at high altitudes. The PINN also shows a similar trend with the number of neurons as the 6×10 network barely improves the mascon regression as depicted in Fig. 8. Nonetheless, an enough number of masses and neurons provides an accurate gravity representation which validates the mascon-PINN model in a scenario other than the constant density polyhedron.

Finally, Fig. 8 shows the gravity error evaluated on the asteroid surface for a mascon-PINN and its corresponding base mascon. Under the heterogeneous polyhedron scenario, large errors on the surface remain as can be observed in Fig. 8 and also in Fig. 6 where the evaluated mascon models usually present higher errrors at low altitudes than with constant density (see Fig. 3). Although the PINN significantly reduces errors on the surface in Fig. 8, it struggles again to drive down the largest error of $\approx 50\%$ produced by the mascon model.

Computational efficiency: For its use on-board, a gravity model has to be lightweight and fast in execution. Here the computational performance of polyhedron, mascon-PINN and mascon models is analyzed. Light and heavy versions of each model are defined: for the polyhedron shapes with 7,790 and 200,700 faces, for the PINN 6×40 and 6×160 versions, and for the mascon 100 and 1,000 masses respectively. The computational times of single executions are shown in Table 2 for 1,000 samples. Both mascon models are very efficient and position themselves ≈ 100 times faster than the light polyhedron version (7,790 faces).

The PINN contribution is evaluated detached from the mascon and is subject to further improvement. The current evaluation of the PINN is sligthly slower, but in the same order of magnitude, than the light polyhedron. While attempting to improve these results we have found hints that suggest further performance can be squeezed out. For example, the execution speed seems to be insensitive with respect to the network parameters: 8,481 $(5 \cdot 40 + (6-1) \cdot 40^2 + 1 \cdot 40 + 6 \cdot 40 + 1)$ for

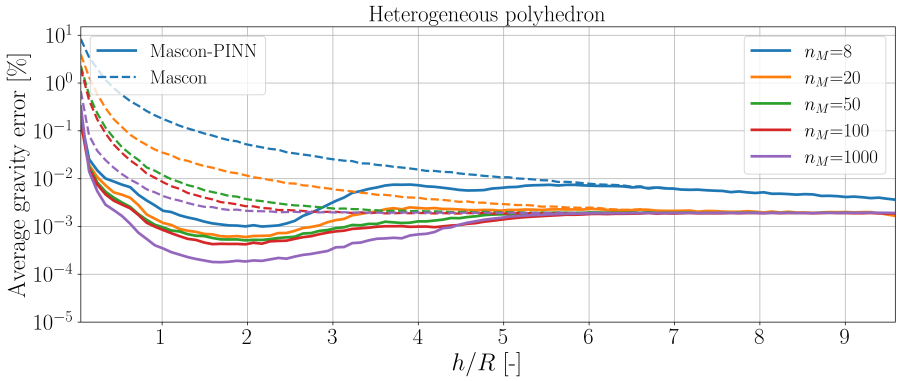

Figure 6. Average gravity error with respect to altitude of heterogeneous polyhedron by varying the number of masses for a SIREN 6x40 network.

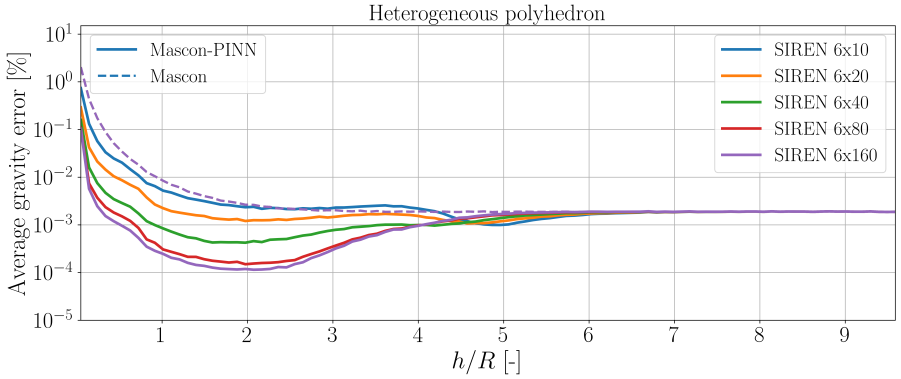

Figure 7. Average gravity error with respect to altitude of heterogeneous polyhedron by varying the number of neurons for $n_M = 100$.

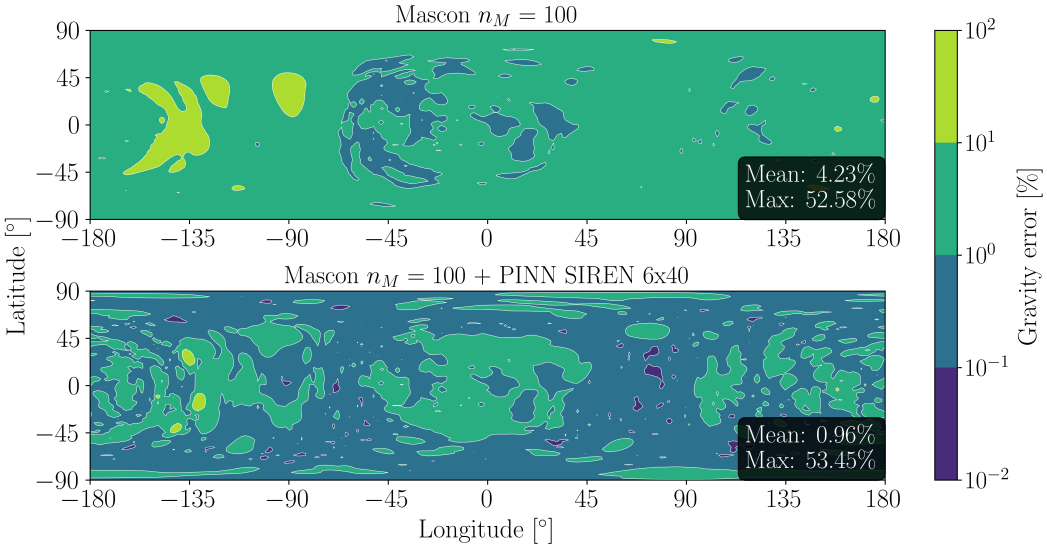

Figure 8. Gravity error on the surface for the heterogeneous polyhedron.

Model	Computation time [ms]
Poly. 200,700 faces	5.066
Poly. 7,790 faces	0.196
PINN 6×40 (BSK)	0.398
PINN 6×160 (BSK)	0.371
PINN 6×40 (JIT)	0.246
PINN 6×160 (JIT)	0.270
Mascon $n_M = 100$	$3.618 \cdot 10^{-3}$
Mascon $n_M = 1,000$	$1.425 \cdot 10^{-2}$

Table 2. Average computational times of gravity models.

the 6×40 network and 129,921 for 6×160. It is suspected that some overhead, Pytorch's C++ frontend communication with Python layer, are possibly shadowing the computational impact of augmenting the network capacity. We have tested both the current Basilisk implementation, which actually resorts to Python interpreter in C++, and PyTorch just-in-time (JIT) compilation and same trends are found though JIT proves to enhance execution speed in a $\approx 40\%$ as shown in Table 2. We are optimistic that thorough research in the subject may place the PINN 6×40 computational efficiency between the heavy mascon and the light polyhedron versions.

Astrodynamics applications

The utility and performance of the mascon-PINN model is tested on two astrodynamics applications. The first consists of an orbital propagation scenario, and the second is the design of a ballistic transfer from orbit to a low altitude target point. In these two applications, the mascon-PINN model prediction is assessed against the ground truth polyhedron models.

Orbital propagation: The orbits are propagated in an asteroid-centred inertial frame N with only the inhomogeneous gravity field perturbation as

$$\ddot{\mathbf{r}}_N = (\mathbf{C}_N^P)^T \mathbf{a}(\mathbf{C}_N^P \mathbf{r}_N) \tag{23}$$

where P is the asteroid-centred rotating frame and $\mathbf{C}_N^P$ is the direction cosine matrix from N to P frame which depends on the asteroid rotation. The asteroid is assumed to rotate uniformly, according to its sidereal period, around the z axis. Note that the inhomogeneous gravity is evaluated in the P frame. In this work, the trajectories are plotted in the P frame which provides relative positioning with respect to the asteroid shape.

To validate numerical propagation, a total test set of 15 orbits is defined. The orbits initial conditions vary the semi-major axis as $a_0 = \{28, 34, 40\}$ km and the orbital inclination as $i_0 = \{0°, 45°, 90°, 135°, 180°\}$. The remaining initial orbital elements are arbitrarily set to $e_0 = 0, \omega_0 = 48.2°, \Omega_0 = 347.8°$ and $\nu_0 = 85.3°$. The 1-day propagation of the test orbits under Eros constant density ground truth polyhedron is shown in Fig. 9. All of them exhibit stable behaviour except from a equatorial one (blue) which evolves to a escape path. Although not shown for space constraints, the heterogeneous polyhedron unstabilizes three orbits of the test set.

To evaluate the propagation accuracy, the root-mean square error along the trajectory is used as a

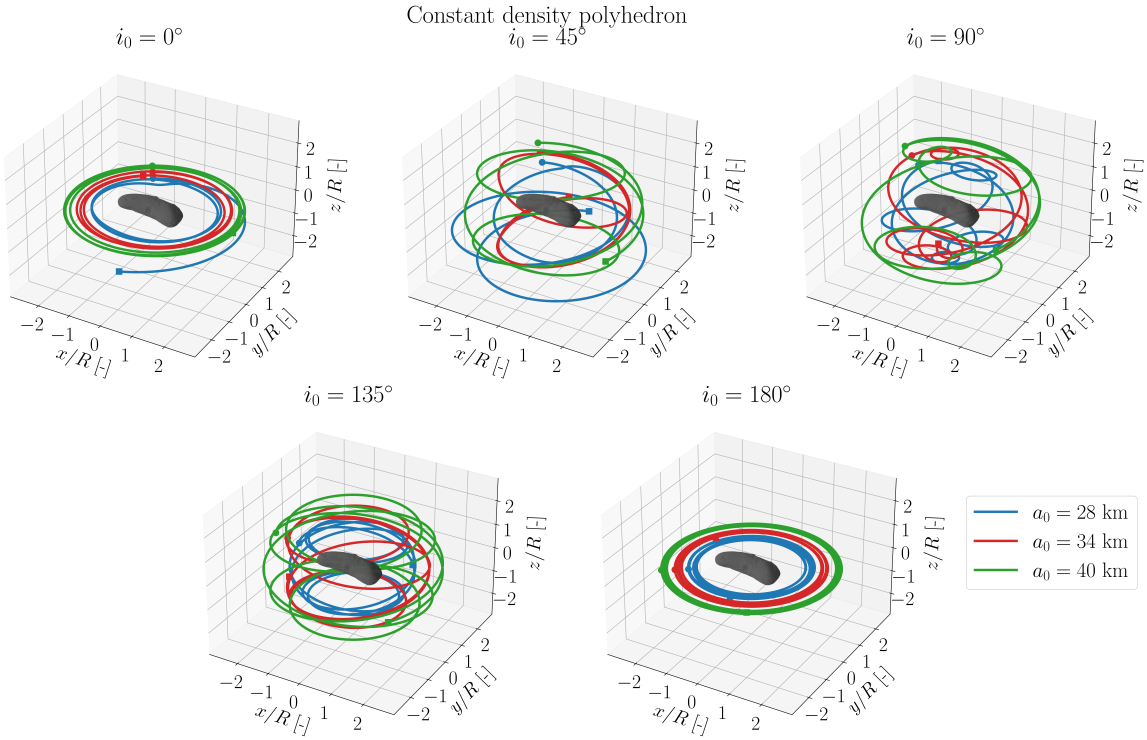

Figure 9. Test orbits during 1-day propagation with constant density polyhedron model.

metric

$$\text{RMSE} = \sqrt{\frac{1}{T}\int_0^T \|\mathbf{r} - \mathbf{r}_{\text{truth}}\|^2 dt} \tag{24}$$

where $\mathbf{r}$ is the orbital prediction of the estimated gravity model and $\mathbf{r}_{\text{truth}}$ is the one corresponding to the ground truth polyhedron model. The RMSE results for several mascon and mascon-PINN models are shown in Fig. 10-11 for Eros constant density and heterogeneous polyhedrons respectively. It can be observed that the mascon-PINN model demonstrates enhanced orbital prediction accuracy under all the test cases. Mascon-PINN models usually provide RMSE ≤ 1 m with few exceptions corresponding to unstable equatorial orbits. Differently, mascon models usually provide RMSE ≈ 10 m but raises to > 100 m for the most challenging cases. A notable mention should be made to the mascon-PINN with $n_M = 1,000$ and 6×40 SIREN which is able to kept sub-meter accuracy in all the test orbits.

Low altitude transfers: Here the fused mascon-PINN model is used to design ballistic transfers to low altitudes. The problem is to find the initial velocity $\dot{\mathbf{r}}_0$ that transfers a spacecraft from $\mathbf{r}_0$ to $\mathbf{r}_f$ in a fixed time T. The following two point boundary value problem (TPBVP) is to be solved

$$\begin{aligned} \text{Find} \quad \dot{\mathbf{r}}_0 \qquad \text{subject to} \quad & \mathbf{r}(0) = \mathbf{r}_0, \qquad \mathbf{r}(T) = \mathbf{r}_f, \\ & \ddot{\mathbf{r}} = -2\boldsymbol{\omega} \times \dot{\mathbf{r}} - \boldsymbol{\omega} \times (\boldsymbol{\omega} \times \mathbf{r}) + \mathbf{a}(\mathbf{r}) \end{aligned} \tag{25}$$

where the variables are referred to the asteroid-centred rotating frame since geographical coordinates are of interest. The term $\boldsymbol{\omega} = [0, 0, \omega]^T$ is the asteroid sidereal rotation rate. The transfer TPBVP is solved using SciPy *solve_bvp* function by sequentially increasing the gravity model complexity. First, the TPBVP Keplerian solution is found and is subsequently used as an initial guess

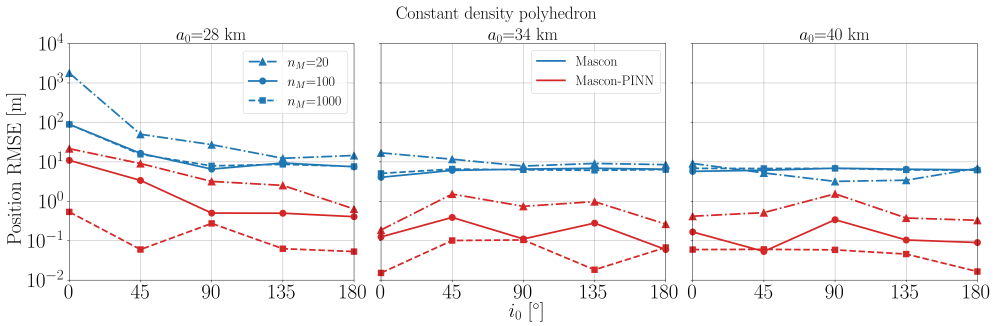

Figure 10. Root mean square error of 1-day orbital propagation for constant density polyhedron with mascon and mascon-PINN of 6x40 SIREN.

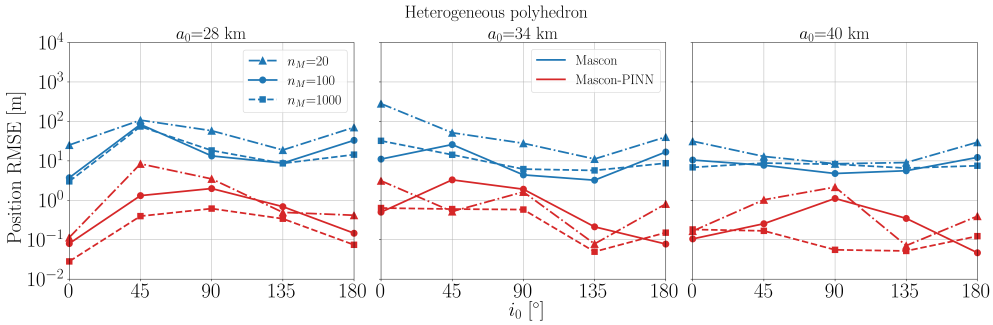

Figure 11. Root mean square error of 1-day orbital propagation for heterogeneous polyhedron with mascon and mascon-PINN of 6x40 SIREN.

for the mascon model transfer. Then, the mascon transfer is used as the initial guess to for the mascon-PINN gravity TPBVP transfer. The approach is naive and we acknowledge that finding the optimal transfer with collision avoidance has been left out of the scope of this work.

The chosen test transfers comprise two different scenarios: From an equatorial orbit to a point close to Psyche crater, and from a polar orbit to a point near Eros north pole. The terminal points are located at $\approx$ 200 m of altitude since targeting the surface requires a more sophisticated algorithm with collision avoidance capabilities. The boundary conditions of the test transfers are detailed in Table 3. For each initial orbit 40 points are sampled in terms of the true anomaly to create an ample test set. Asteroid collisions are checked by evaluating the TPBVP solution. If a collision is detected with any of the models (Kepler, mascon or mascon-PINN), the transfer is removed from the analysis. The majority of the North pole transfers occur without collisions but only 11 of the 40 initial conditions for Psyche crater transfer do not collide with the asteroid. This makes sense because the projection of Eros shape is larger in the equatorial plane xy than in the polar plane yz.

The outcome of the designed transfers is shown in Fig. 12 for the constant density polyhedron. For the sake of brevity, only the mascon with $n_M = 100$ and its fused PINN with 6×40 SIREN are compared in terms of terminal accuracy. The error in reaching the desired target is depicted in Fig. 13 with a bidimensional projection in xz and xy planes respectively. One can think of the coordinates contained in these planes as terminal lateral errors. Due to this, they are denoted as $\delta x'$ and $\delta y'$ with $\delta z'$ the longitudinal error which corresponds to the planetocentric $-y$ (transfer to Psyche crater) or z (transfer to North pole) directions depending on the scenario. For both the Psyche crater and north pole transfers, the mascon-PINN guarantees < 0.25 m in lateral accuracy

From	$\mathbf{r}_0$ [km]	To	$\mathbf{r}_f$ [km]	T [h]
Equatorial orbit	$34 \cdot [\cos\nu, \sin\nu, 0]^T$	Psyche crater	$[0, -3.75, 0]^T$	4
Polar orbit	$34 \cdot [0, \cos\nu, \sin\nu]^T$	North pole	$[0, 0, 5.75]^T$	4

Table 3. Boundary conditions for low altitude ballistic transfers. For each departure orbit 40 equispaced true anomalies ν are simulated.

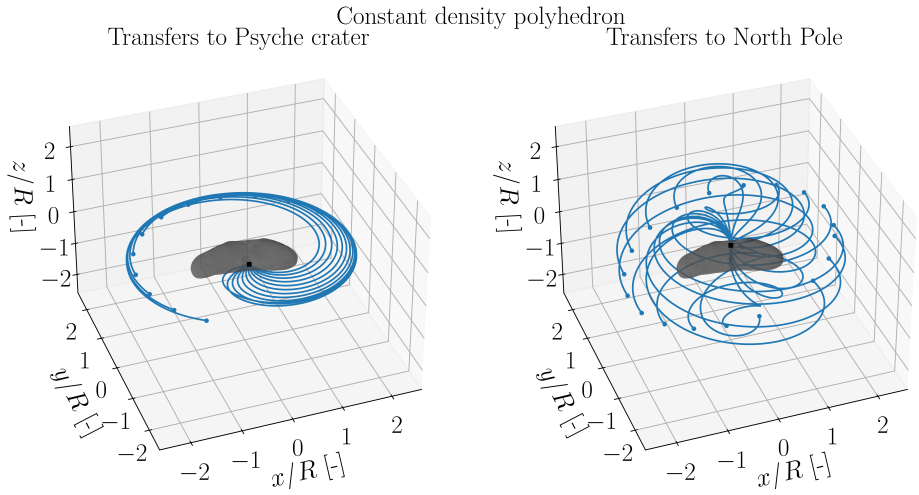

Figure 12. Transfers propagated with ground truth constant density polyhedron and designed with mascon-PINN model (n_M=100 - 6×40 SIREN).

while the mascon dispersion can be as large as 2 m (Fig. 13). When evaluating the total terminal error, $\|\mathbf{r}(t_f) - \mathbf{r}_f\|$, the mascon model provides $1.2 - 3$ m in terminal accuracy while the mascon-PINN is superior with errors contained in the $0.04 - 0.3$ m range. Similar trends are found for the heterogeneous polyhedron though they are not included in the manuscript due to length constraints.

CONCLUSIONS

This manuscript introduces the mascon-PINN gravity model, which effectively combines the traditional mascon approach with a physics-informed neural network (PINN). The mascon-PINN model can be thought of as a base mascon distribution with an attached PINN that is smoothly down-weighted at altitudes beyond the data bounds. This fusion strategy mimics the PINN-GM-III model (see Ref. 13), where a point-mass model is used to robustify the model solution at high altitudes. The mascon-PINN training process is based on a position-gravity dataset and encompasses two stages: first, the mascon distribution is regressed and kept invariant; afterward, a weighted PINN is added to form the mascon-PINN gravity model, with only the network weights and biases being trained.

The usefulness of the proposed mascon-PINN model is supported by ample evidence. The ground truth models are diverse, encompassing both a constant density and a heterogeneous polyhedron. When approximating the previous models the mascon-PINN has demonstrated their superiority over the standalone mascon in all test scenarios. The testing has included the evaluation of global gravity accuracy, orbital propagation, and ballistic transfer design. These results pertain low-to-medium altitudes as both models are equivalent at high altitudes due to the PINN deactivation. The

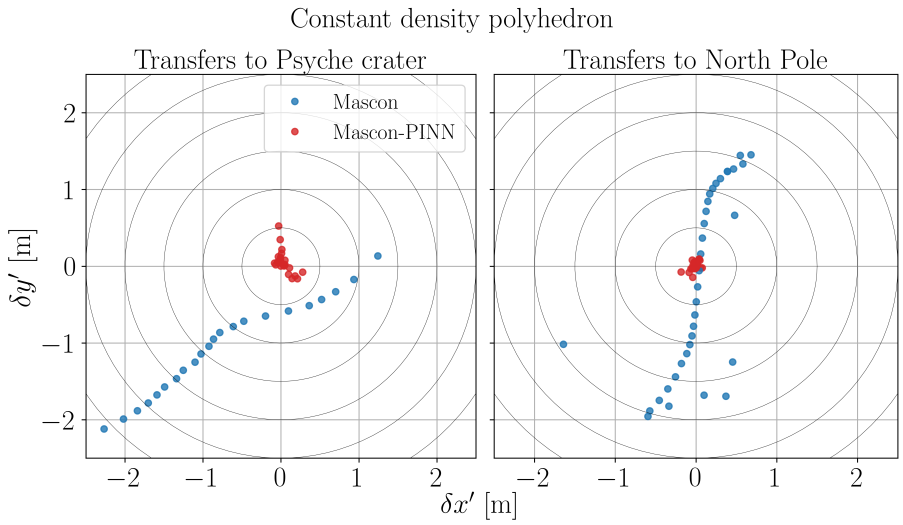

Figure 13. Transfers terminal lateral errors for mascon (n_M=100) and mascon-PINN (n_M=100 - 6×40 SIREN) models.

simulations are conducted using the Basilisk open-source astrodynamics library, demonstrating the mascon-PINN model's ease of use. Future work may focus on finding and mitigating the root cause of the computational time overhead when evaluating the PINN contribution. In addition, down-weighting the mascon model at low altitudes is to be explored as that strategy could possibly allow to remove large errors on the surface.

ACKNOWLEDGMENT

This work has received funding from the European Union's Horizon 2020 research and innovation programme under the Marie Skłodowska-Curie grant agreement No. 101025257.

REFERENCES

[1] D. J. Scheeres, "Orbit Mechanics About Asteroids and Comets," *Journal of Guidance, Control, and Dynamics*, Vol. 35, No. 3, 2012, pp. 987–997, 10.2514/1.57247.

[2] W. Kaula, *Theory of Satellite Geodesy: Applications of Satellites to Geodesy*. Waltham, Mass.: Blaisdell Publishing Co, 1966.

[3] 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, 1996, pp. 313–344, 10.1007/BF00053511.

[4] P. M. Muller and W. L. Sjogren, "Mascons: Lunar Mass Concentrations," *Science*, Vol. 161, No. 3842, 1968, pp. 680–684, 10.1126/science.161.3842.6.

[5] T. G. G. Chanut, S. Aljbaae, and V. Carruba, "Mascon gravitation model using a shaped polyhedral source," *Monthly Notices of the Royal Astronomical Society*, Vol. 450, No. 4, 2015, pp. 3742–3749, 10.1093/mnras/stv845.

[6] R. P. Russell and N. Arora, "Global Point Mascon Models for Simple, Accurate, and Parallel Geopotential Computation," *Journal of Guidance, Control, and Dynamics*, Vol. 35, No. 5, 2012, pp. 1568–1581, 10.2514/1.54533.

[7] A. Colagrossi, F. Ferrari, M. Lavagna, and K. Howell, "Dynamical evolution about asteroids with high fidelity gravity field and perturbations modeling," *AAS/AIAA Astrodynamics Specialist Conference*, Vail, United States, 2015.

[8] J. Garcia-Bonilla, P. Machuca, and M. Sanjurjo-Rivo, "Small-body Gravitational Modeling for On-board Operations and Mass Distribution Estimation: Trade-off Analysis and Novel Approach," *International Astronautical Congress*, Dubai, United Arab Emirates, 2021.

[9] J. C. Sanchez and H. Schaub, "Simultaneous navigation and mascon gravity estimation around small bodies," *Acta Astronautica*, Vol. 213, 2023, pp. 725–740, 10.1016/j.actaastro.2023.09.023.

[10] S. McArdle and R. P. Russell, "High-Resolution Global Point Mascon Models for the Moon," *Journal of Guidance, Control, and Dynamics*, Vol. 46, No. 7, 2023, pp. 1390–1396, 10.2514/1.G006921.

[11] R. Furfaro, R. Barocco, R. Linares, F. Topputo, V. Reddy, J. Simo, and L. L. 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] 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, 2020, pp. 66–79, 10.1016/j.actaastro.2019.11.039.

[13] J. Martin and H. Schaub, "The Physics-Informed Neural Network Gravity Model: Generation III," *submitted to the Journal of the Astronautical Sciences*, 2024.

[14] D. Izzo and P. Gomez, "Geodesy of irregular small bodies via neural density fields," *Communications Engineering*, Vol. 1, No. 48, 2022, 10.1038/s44172-022-00050-3.

[15] 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.

[16] J. Martin and H. Schaub, "Physics-informed neural networks for gravity field modeling of the Earth and Moon," *Celestial Mechanics and Dynamical Astronomy*, Vol. 134, No. 13, 2022, 10.1007/s10569-022-10069-5.

[17] J. Martin and H. Schaub, "Physics-informed neural networks for gravity field modeling of small bodies," *Celestial Mechanics and Dynamical Astronomy*, Vol. 134, No. 46, 2022, 10.1007/s10569-022-10101-8.

[18] P. W. Kenneally, S. Piggott, and H. Schaub, "Basilisk: A Flexible, Scalar and Modular Astrodynamics Simulation Framework," *Journal of Aerospace Information Systems*, Vol. 17, No. 9, 2020, pp. 496–507, 10.2514/1.I010762.