

GUIDED INVERSE GRAVITY MODELING FOR
ASTEROIDS USING NEURAL NETWORKS

JAN-FREDERIK STOCK

Universität Bremen
December 4th, 2024

Matrikelnummer: 451193
Erstgutachter: Prof. Dr. Gabriel Zachmann
Zweitgutachter: Dr. Felix Putze

CONTENTS

1	Introduction	1
2	Theoretical background and Related Work	2
2.1	Forward Modeling of Gravitational Fields	2
2.1.1	Spherical Harmonics	2
2.1.2	Polyhedral Method	3
2.1.3	Mass Concentration Method	3
2.2	Classical Methods for Inverse Gravity Modeling	5
2.3	Inverse Gravity Modeling with Machine Learning	9
2.3.1	Inverse Gravity Modeling with Convolutional Networks	9
2.3.2	Inverse Gravity Modeling with a NeRF inspired approach	12
3	Building the dataset	16
3.1	Obtaining and cleaning up the data	16
3.2	Modeling and calculating the gravitational field	17
3.2.1	Utilized Libraries	21
3.2.2	Usage	21
4	Modifications to GeodesyNet	22
4.1	The Loss Function	22
4.2	Heterogeneous Ground Truth Data	25
4.3	Guidance Point samplers	28
4.4	Performance Considerations	28
4.5	Post Training	29
4.6	Mascon based comparative method	29
4.7	Validation procedure	31
5	Results and Discussion	32
5.1	GeodesyNet vs. masconCUBE	32
5.1.1	Results on the gravitational field	32
5.1.2	Comparing the guidance losses	39
5.1.3	Conclusion	40
5.2	Finding a good guidance factor	41
5.3	Comparing guidance approaches for the masconCUBE	42
5.4	Post Training	42
6	Future Work	51
7	Conclusion	52
	Bibliography	53

LIST OF FIGURES

Figure 2.1	Body modeled with blocks according to [28].	8
Figure 2.2	Test case of the U-Net based inversion method with two shapes above one another. Figure from [34].	10
Figure 2.3	Training procedure of a GeodesyNet. Figure from [11].	13
Figure 3.1	Predicted body mass with Polynomial Regression and Random Forest using the mean diameter.	17
Figure 3.2	Filling the bounding box with mascons. Here ten subdivisions were used, leading to 1000 mascons.	19
Figure 3.3	Subdividing a cube with the asteroid placed inside. Here the cubes sides have the length of the asteroids maximum extent. Each dimension has ten subdivisions leading to 1000 mascons.	19
Figure 3.4	The sampled points creating a sampling sphere around the asteroid.	20
Figure 4.1	Comparing the last iterations of a training run with the initial implementation. The separated components of the loss are shown, one based on the ground-truth gravitational field and the other on the given guidance density for one guidance volume, with the loss of an unmodified GeodesyNet depicted for comparison.	25
Figure 4.2	Comparing the last iterations of a training run after introducing a scaling factor to weight the two loss terms. The loss of an unmodified GeodesyNet is shown for comparison.	26
Figure 4.3	67P Churyumov-Gerasimenko with a density heterogeneity introduced for its head.	27
Figure 5.1	Excerpt from the results on Churyumov-Gerasimenko with a heterogeneity of one sphere in the middle of the body, with radius 0.05 and density 1. A guidance factor of 0.05 was used for the guidance models.	33

- Figure 5.2 Excerpt from the results on Bennu with a heterogeneity of one sphere close to the surface of the body, with radius 0.05 and density 1.2. A guidance factor of 0.05 was used for the guidance models. 34
- Figure 5.3 Excerpt from the results for Churyumov-Gerasimenko with three spherical heterogeneities distributed through the body. The spheres have radii of 0.05, 0.035 and 0.05 and densities of zero. A guidance factor of 0.0125 was used for the guidance models. 36
- Figure 5.4 Excerpt from the results for Churyumov-Gerasimenko with three spherical heterogeneities distributed through the body. The spheres have radii of 0.05, 0.035 and 0.05 and densities of 1, 0.5 and 2.5. A guidance factor of 0.0125 was used for the guidance models. 37
- Figure 5.5 Excerpt from the results for Bennu with three spherical heterogeneities distributed through the body. The spheres have radii of 0.05, 0.035 and 0.05 and densities of 1, 0.5 and 1.5. A guidance factor of 0.05 was used for the guidance models. 38
- Figure 5.6 Excerpt from the results for Churyumov-Gerasimenko with the heterogeneity defined through a plane. The density is set to 1.83 on the positive side of the plane. A guidance factor of 0.0125 was used for the guidance models. 39
- Figure 5.7 Excerpt from the results for Bennu with the heterogeneity defined through a plane. The density is set to 0.99 on the positive side of the plane. A guidance factor of 0.0125 was used for the guidance models. 40
- Figure 5.8 Full guidance factor study on Churyumov-Gerasimenko with a heterogeneity of one sphere. 44
- Figure 5.9 Excerpt from the guidance factor study on Churyumov-Gerasimenko with a heterogeneity of one sphere in the middle of the asteroid with radius 0.05 and density 1. 45
- Figure 5.10 Excerpt from the guidance factor study on Churyumov-Gerasimenko with three spherical heterogeneities with radii 0.05, 0.035 and 0.05 and densities of zero. 45

Figure 5.11	Excerpt from the guidance factor study on Bennu with three spherical heterogeneities with radii 0.05, 0.035 and 0.05 and densities of 1, 0.5 and 1.5. 46
Figure 5.12	Excerpt from the guidance factor study on Bennu with a heterogeneity of one sphere close to the surface of the asteroid with radius 0.05 and density 1.2. 46
Figure 5.13	Excerpt from the guidance factor study on Churyumov-Gerasimenko with the heterogeneity defined through a plane. The density is set to 1.83 on the positive side of the plane. 47
Figure 5.14	Excerpt from the guidance factor study on Bennu with the heterogeneity defined through a plane. The density is set to 0.99 on the positive side of the plane. 47
Figure 5.15	Excerpt from the validation results on Bennu for the masconCUBE model, with the heterogeneity defined through a plane, with the density set to 0.99 on the positive side of the plane. The figure shows both versions of guidance I implemented, with the masses defined in every iteration on the left and the guidance loss based version on the right. 48
Figure 5.16	Results of the post training method on Bennu with the heterogeneity defined through a plane. A guidance loss of 0.0125 was used. 49
Figure 5.17	Results of the post training method on Churyumov-Gerasimenko with the heterogeneity defined through a sphere in the middle of the body. A guidance loss of 0.05 was used. 50

LIST OF TABLES

Table 5.1	Final guidance losses for the studied ground-truths, for GeodesyNets and masconCUBEs. 41
-----------	--

GLOSSARY

HARMONIC FUNCTION “[A] function is called harmonic in a region v of space if it satisfies Laplace’s equation at every point of v ” [10, p. 15]. 3

MASCON Short for mass concentration. In this thesis the term is used as shorthand to mean an object e.g. a sphere, cube or tetrahedron with its mass concentrated at a point at the center of the shape. 4–6, 13, 15, 16, 19

NERF Neural Radiance Field. xi, 10, 13

POLYHEDRON “[T]hree-dimensional solid body whose surface consists of planar faces meeting along straight edges or at isolated points called vertices. Exactly two faces meet at each edge. Three or more edges and a like number of faces meet at each vertex” [32]. 4, 6

TETRAHEDRON . 4, 6, 13

INTRODUCTION

In the recent decades, there have been multiple space missions with the goal to visit and gather data about small-bodies in our solar system. These include the NEAR mission to the asteroid 433 Eros [18], Hayabusa to the asteroid Itokawa [7], the Dawn mission to the proto-planets Vesta and Ceres [12] and most recently the OSIRIS-REx mission to the asteroid Bennu. Some of these missions were able to get close enough to the body to measure detailed gravitational potentials. The gravitational potential outside of a body is produced by its internal mass distribution, which is why it can be used to reconstruct the internal mass distribution of a body. This process is called gravity inversion [1]. Studying the internal structure of a small-body can give insight into its history and how it has changed over time. Knowledge about the internal structure also helps to build more accurate models of small bodies [31]. Information about the internal mass distribution of a body can also be obtained through other methods than analyzing the gravitational field, for example through radar measurements or analysis of collected dust particles [27]. Recently, a gravity inversion method based on Neural Networks has been developed, which requires fewer assumptions about the internal mass distribution and shape of a body, and can yield more accurate results in some situations, compared to classical inversion methods [11]. There also have been recent works that employ Machine Learning based approaches to perform gravity inversion on earth [34] [15] [5]. My goal in this thesis it to incorporate available information about the mass distribution of a small-body into the gravity inversion process of the Machine Learning based method most suited to this task, and thus obtain a mass distribution that is consistent with this information.

In chapter 2 I will give an introduction into the theoretical background of forward modeling of gravitational fields. I will also present classical methods of gravity inversion and introduce machine learning based gravity inversion methods, on which later parts of this thesis will be based on. In chapter 3 I will describe how I built a dataset to be used as training data for a machine learning based inverse gravity modeling approach. I will describe the modifications I made to one of the machine learning based methods to incorporate additional information into the gravity inversion process in chapter 4. Chapter 5 is about the results I obtained with these modifications. In chapter 6 I will motivate possible future work and I will conclude in chapter 7.

THEORETICAL BACKGROUND AND RELATED WORK

2.1 FORWARD MODELING OF GRAVITATIONAL FIELDS

In order to do inverse gravity modeling, gravitational data which the chosen inverse gravity modeling method should try to match as closely as possible, is needed. In the case of a space mission like NEAR visiting the asteroid Eros [30] or more recently OSIRIS-REx visiting the asteroid Bennu [14], this data might come from a spacecraft taking measurements around the body of interest. The other option is to use data generated through forward modeling, where the gravitational field around a body is calculated from the shape of the body as well as an assumption about the distribution of mass inside the body. For the development of inverse gravity modeling methods this approach is suited well, because it allows the easy generation of multiple variants of mass distributions and their corresponding gravitational fields to test with. Data generated by this method is also much more easily accessible as gravitational data measured by spacecraft. There are multiple methods to do forward modeling of gravitational data, the most common of which will be presented and discussed in the following.

2.1.1 *Spherical Harmonics*

The classical method of calculating the gravitational field of a body is using spherical harmonics. This method exploits the fact that the gravitational potential is a harmonic function outside the attracting mass. This means that the gravitational potential can “be expanded into a series of spherical harmonics” [10, p. 57]. The coefficients of this series can then be computed to obtain a representation of the gravitational potential [32]. A drawback of the spherical harmonics method is that the expansion of the gravitational potential into spherical harmonics is only guaranteed to converge towards the gravitational potential outside of the smallest sphere that encloses the body completely. Inside of this sphere the series usually diverges [10, p. 60]. This issue is less pronounced for bodies which are closer to a sphere in shape, like the earth. For the bodies studied in this thesis, which often have a shape that differs strongly from a sphere, using the spherical harmonics method would leave a lot of space around the body, where the series of spherical harmonics does not converge. It is still possible to calculate the gravitational field in this case, but results may be less accurate [25]. Methods to calculate spherical harmonics for arbitrarily

shaped bodies do exist, but are hard to apply in practice [32]. Due to these issues with the spherical harmonics method, I decided against using it in this thesis.

2.1.2 Polyhedral Method

When using the polyhedral method to calculate the gravitational field around a body, the body is represented as a polyhedron. The polyhedral method provides an exact solution for the gravity field corresponding to this polyhedron and its accuracy is only limited by the level of discretization of the polyhedron. The exactness of the solution also does not depend on the position in relation to the body, it is as exact close to the surface as it is farther away [32]. In the form of the polyhedral method initially discussed in [32], the assumption that the studied body has constant density is made. This makes the method unsuitable in this form, as the main goal of this thesis is to study and model bodies with heterogeneous density distributions. The authors propose to model density variations inside a body “by adding and subtracting small polyhedra internal to an overall polyhedral model” [32]. The gravity field for these added shapes can then be calculated and added to the gravitational field for the rest of the body. A more complete adaptation to the polyhedra method for heterogeneous density distributions is presented in [28]. The authors subdivide the body into tetrahedra formed by connecting the corners of the triangles of the surface mesh to the center of the body. These tetrahedra are subdivided into regions, and the polyhedral gravitational field is calculated for each region. The results for each region are summed to attain the complete gravitational field for the whole body. This leads to a representation of the gravitation with the benefits discussed above, the authors note though that this method is computationally intensive and recommend other methods if fast computation times are needed.

2.1.3 Mass Concentration Method

When calculating the gravitational field of a body using the mass concentration method, the body is approximated as a set of point masses. The gravitational acceleration or potential at a given point \mathbf{p} is then derived by summing up the contributions of every point mass.

$$a = -G \sum_i \frac{\rho_i V_i}{r_i^3} \mathbf{r}_i \quad (2.1)$$

Equation 2.1 shows the formula for the gravitational acceleration, with G denoting the gravitational constant, ρ_i and V_i the density and volume of the mass concentration point or mascon i and with r_i denoting the distance between the mascon \mathbf{x}_i and the sample point \mathbf{p}

[21]. The vector \mathbf{r}_i runs from the sample point \mathbf{p} to the mascon \mathbf{x}_i and can be calculated as shown in equation 2.2 [22].

$$\mathbf{r}_i = \mathbf{x}_i - \mathbf{p} \quad (2.2)$$

The gravitational potential at a sample point \mathbf{p} is analogously calculated by

$$U = G \sum_i \frac{m_i}{r_i}, \quad (2.3)$$

with m_i representing the mass of mascon i [4].

Implementations of the mascon method differ in how the body of interest is subdivided into smaller sections and how these sections are placed throughout the body. Commonly used methods include filling the body with elements such as spheres or cubes. In case of [20], these elements are distributed uniformly throughout the body and the results compared to the polyhedral method. The authors find that the mascon approach is able to approximate the polyhedral gravitational field closely, with noticeable differences only close to the bodies surface.

The authors of [16] propose a method to improve upon the greater errors of mascon models when evaluating points close to the surface a body. They use a sphere packing algorithm to fill the body with non-uniformly sized spheres while focusing on placing a greater amount of smaller spheres close to the surface of the body. This allows for a greater accuracy close to the surface. With this approach, they are able to reduce the error by almost 30% in comparison to a similar method.

In [2], the authors take a different approach to determining the positions of the mascons. They start out with a polyhedral shape model of the body. Each triangular face of the polyhedral shape “is connected to the centre of the asteroid to form a tetrahedron” [2]. The point mass is then set to be proportional to the volume of the tetrahedron and placed at its centroid. The authors propose a second method as well, where they further divide each of the tetrahedra into three parts and place the point masses at the centroid of each part. This divides the body into three layers and results in an increased resolution of the mascon model. With these methods, the authors achieve similar results to the polyhedral method close to the surface of the body, with reduced computation time in comparison to the polyhedral method. Differences in the gravitational potential between the mascon models and the polyhedral model occur mainly “in the interior of the body and at the edges, where the distance with the body’s centre of mass is greater” [2].

To sum up, the mass concentration method provides similar accuracy of the calculated gravitational field as the polyhedral method while being able to model heterogeneous mass distributions. It also provides faster computation times as the polyhedral method. Additionally, it does not suffer from inaccuracies of the gravitational field close to the surface of the body to the same degree as the spherical harmonics method. Though it is not completely immune to this like the polyhedral method, there are ways to mitigate these inaccuracies. The mascon method is thus well suited to model the gravitational fields of small bodies like asteroids, which often have shapes that differ strongly from a sphere and might have heterogeneous mass distributions.

2.2 CLASSICAL METHODS FOR INVERSE GRAVITY MODELING

Inverse gravity modeling, meaning the derivation of the mass or density of a body from its gravitational field, is an ill-posed problem. The main reason for this is that a solution for inverse gravity modeling is non-unique. To solve this issue, a condition needs to be imposed on the searched for density [17]. This can be done by introducing a-priori knowledge about the studied body. In case of the earth, this knowledge is available through measurements of seismic or magnetic properties, which are used to constrain the solution for the density. For small bodies such as asteroids, such information about the internal structure of the body might not be available [1]. Methods to address these problems for small bodies like asteroids are presented in the following.

In [24], the authors assume measured spherical harmonic coefficients and a polyhedral model of the asteroid shape are given. The shape is comprised of tetrahedra which fill the volume of the asteroid and are grouped into polyhedra either to fill the volume uniformly or according to some physical reasoning about the internal structure of the asteroid. The authors explore a uniform distribution of tetrahedra, with 507 being assigned the same density and one density outlier. They also test a grouping into eight larger quadrants with different densities, and a core-mantle discrimination, with a lower density mantle and a higher density core. To determine the densities of these formed polyhedra, the authors use a least-squares formulation to minimize the difference between the given spherical harmonic gravity field and the gravity field created by the chosen polyhedral grouping. They then use singular-value decomposition to solve the resulting equation.

In the uniform distribution of 508 tetrahedra with one outlier, the problem is under-determined, meaning that the resolution of the measured gravitational field is not high enough to find an exact solution for the amount of densities, that need to be determined. The method is still able to distinguish the one outlier density from the others,

although no precise density values can be determined. The one outlier also only affects the determined densities around it and does not distort the found densities in the rest of the asteroid, meaning the model is able to detect “local distortions in the density distribution” [24]. In the over-determined case, where only eight densities need to be determined, one for each quadrant, the model was able to determine the densities with a very small error. While testing the same eight quadrant model with a lower resolution gravitational field to make the problem under-determined, the authors found that the error increases drastically in this case. The model was also able to find a good solution for the core-mantle distribution, given a gravitational field with sufficient resolution.

The authors of [1] divide the studied body into a surface layer with constant density, and an interior layer discretized into a generic mesh. The elements forming the interior layer are then “grouped to form density anomalies” [1], which are regions of uniform density. This grouping reduces the numbers of parameters that need to be estimated. The authors estimate the initial shape of the density anomalies by using the level-set method adapted from [8]. The final shapes of the anomalies and their densities are then obtained by applying an iterative least-squares optimization.

The authors note that due to the non-unique nature of the gravitational inversion problem, the single solution provided by the least-squares optimization is not sufficient because of the other possible solutions that would fit the data equally well. To tackle this, the authors apply their inversion algorithm multiple times, to generate multiple possible solutions and cluster these solutions into similar families. They propose that this leads to “information on the uncertainties associated with a single family of solutions ... [and provides] an exploration of the range of possible interior distributions” [1]. While exploring this approach the authors find that they only detect “less than 3 significantly different families of interior, although generally the true solution was included in one of those” [1]. They conclude that additional work is needed to remove biases from the method.

The authors find that their method is able to approximate the true density reasonably well in most of their tested cases. Their method generally performed better on more irregularly shaped models than on more spherical bodies. Due to the initial assumption in the method, that the body is composed of distinct regions of density, the method is well suited for bodies which are composed of distinct anomalies. The method is still able to find solutions for bodies with smooth density distributions, but other methods are better suited for cases such as this, as well as when a more complete view of possible solutions to the gravity inversion problem is needed.

In [20] the authors discretize the studied body into finite elements, either cubes or spheres, beginning with 211 elements. They start out with an initial estimate for the densities of these elements and use a batch least-squares covariance analysis to optimize the densities to best fit the measurement gravity data. The gravity data to which the estimates are compared is modeled in the shape of trajectories around the body, like a spacecraft taking measurements would produce them. The authors find that they only get significant improvements on their initial density estimate for elements which are positioned near the surface of the body, where the points the gravity field is measured at are close to the elements. They also find that when they increase the number of elements to 3292, their method is unable to meaningfully improve upon the initial density estimate. They conclude that the points where the gravity is measured at are too far away to derive a good estimate for each element at this resolution. They propose to start the estimation with fewer elements and increase their number, as measurements from closer trajectories become available in practice.

The author of [29] conceptualizes gravity modeling in terms of matrices, resulting in a matrix which transforms a density vector into the corresponding gravitational potential. The inverse of this matrix would solve the inverse gravity modeling problem. The kind of matrix inversion problem studied in the paper is underconstrained, which leads to infinite solutions for the density distribution, which all exactly match the input gravitational field. To generate these solutions, the author obtains a reference solution using the pseudo-inverse matrix. This inversion is performed using QR decomposition and also yields kernel basis vectors of the matrix. Other solutions to the inversion are now described by equation 2.4, with $c_{i_e j_e k_e}$ representing the density solution, $[c_{i_e j_e k_e}]_{ref.}$ the reference solution and $[u_q]$ being the kernel basis vectors, with s_q representing arbitrary factors.

$$[c_{i_e j_e k_e}] = [c_{i_e j_e k_e}]_{ref.} + \sum_q s_q [u_q] \quad (2.4)$$

With the help of this equation the space of exact solutions to the inversion problem can be explored, for example using Monte Carlo techniques. To test the method, the author applies it to a sample body assuming a uniform mass distribution, one composed of three layers and the layered model again with added noise. Due to the fact that there are infinitely many solutions to the inversion problem, assumptions about the solution can be introduced to make exploring the solutions space easier. These assumptions are used in the Monte Carlo search to find solutions that match them well. The author gives examples of these assumption which include assuming a certain density range, looking for the solutions which have a minimal or a maximal range of densities and looking for solutions without local minima inside of the body. Several of these target functions can be combined

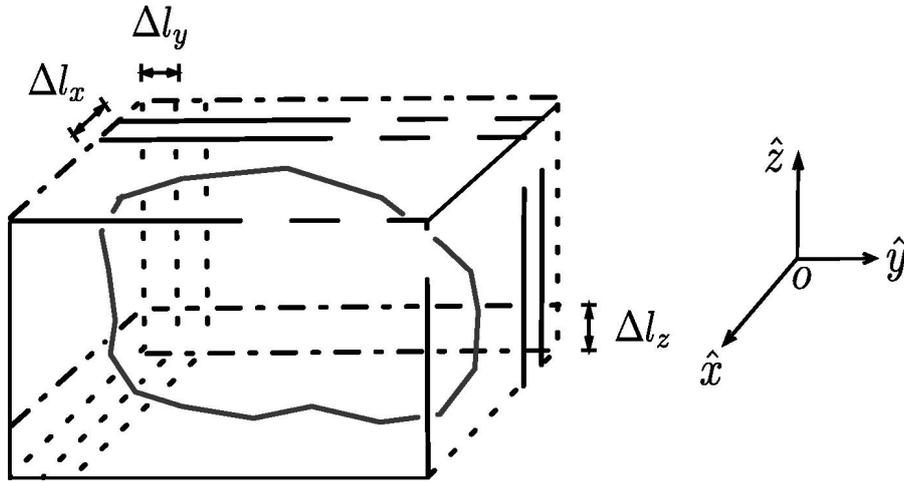


Figure 2.1: Body modeled with blocks according to [28].

and balanced by weights. In the tests, the solutions generally converge on the underlying ground truth density distribution while increasing the resolution of the input gravitational field, if the assumptions constraining the search for solutions are correct. This is true for both the uniform and the layered mass distribution. Solutions are also found in the test case with added noise, but the noise can limit the maximum input resolution of the gravitational field, where a solution can still be found.

In [28], the authors utilize so called density maps, which “refer[s] to the distribution of different density regions within the body, and the density value is the value of density associated with each density region” [28]. Given this density map and a gravitational field, they determine the density value for each of the density regions using least-squares estimation. If the number of density regions is kept low enough in comparison to the gravitational fields resolution, the least-squares estimation yields a unique solution. The density regions are created by covering the body with equally sized blocks, as shown in figure 2.1. To calculate the gravitational field from the estimated density distribution, the authors use their adaptation of the polyhedral method for heterogeneous bodies as described in section 2.1.2. Each of the described sections gets its density for the calculation from the block from the block model, which overlaps with its center.

The authors evaluate their method on the asteroid Castalia. They start out with two blocks or density regions, then increase to 10 and 90 blocks. They find that their method works well, with a percentage error between the estimated and true densities lower than 10% in most parts of the asteroid. The error between the gravitational field calculated from the estimated density distribution and the input gravitational field “decrease[s] by about an order of magnitude each time the number of blocks is increased” [28]. When they evaluate the gravitational

field inside of smallest possible sphere encompassing the body as well as on its surface, they find errors under 10% and under 5% respectively.

In general the classical approaches to inverse gravity modeling presented here involve imposing constraints on the solution for the density distribution to deal with the non-uniqueness of the inversion problem and make finding a unique solution possible. In case of [29] a general solution space is derived first, after which different constraints can be employed to explore this solution space. The solution for the inversion is usually found using optimization methods.

2.3 INVERSE GRAVITY MODELING WITH MACHINE LEARNING

There are generally two kinds of approaches used to do inverse gravity modeling with machine learning. The first approach usually involves a kind of convolutional neural network being trained to do the inversion, which is then able to generalize to unseen data. This approach will be presented first. The second approach takes inspiration from so called Neural Radiance Fields or NeRFs.

2.3.1 *Inverse Gravity Modeling with Convolutional Networks*

Approaches which employ convolutional networks to do inverse gravity modeling usually apply their method to gravitational measurements on the surface of the earth, to predict the density in the earth's crust underneath. An example for this is [34], where the authors train a U-Net and apply it to gravity data from East Antarctica. The U-Net is a fully convolutional network, which was developed by [23] with the aim to provide good performance with very few training samples, in contrast to previous deep convolutional networks. In the initial work, it was used for an image segmentation task. To adapt the U-Net for gravity inversion, the authors of [34] use a 32x32 image with one channel as an input, each pixel representing a gravitational measurement. The output of the model is a 32x32 image with 16 channels, with the channels being used to represent the third dimension of the output, which is the depth in this case. The so called Dice function is used as a loss function that measures the agreement between the predicted density distribution and the ground truth distribution. The authors add an additional constraint term, which consists of the sum of the weighted squared L2-Norm between the predicted and ground-truth density distributions and the predicted and ground-truth gravity data respectively. A synthetic dataset is used to train the model, which is generated using a random walk approach. The authors subdivide their research area into 32x32x16 cubes with one kilometer side length each, matching the output dimensions of their U-Net. They then start from an initial 2x2x2 cube starting point in this grid and move a number

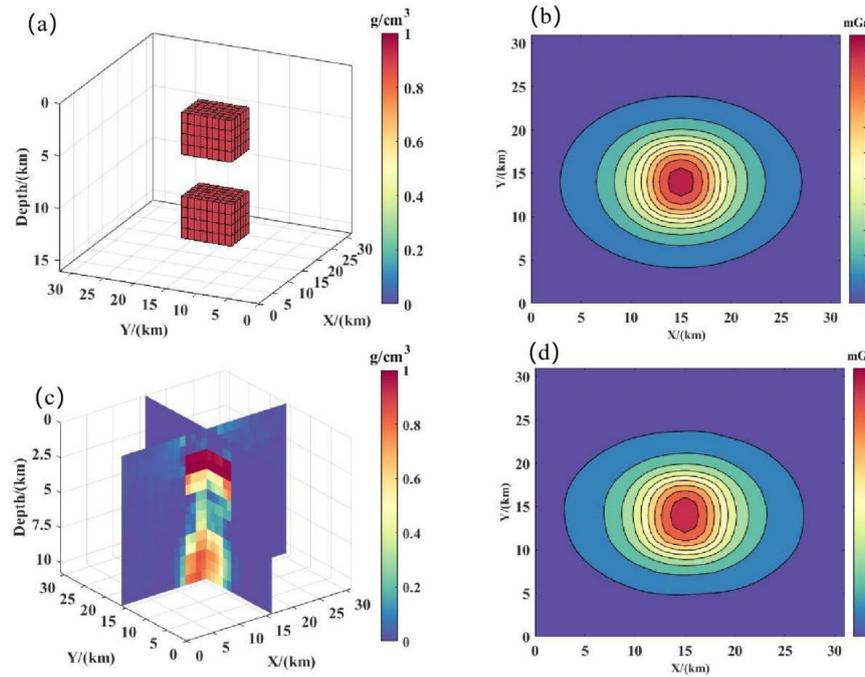


Figure 2.2: Test case of the U-Net based inversion method with two shapes above one another. Figure from [34].

of steps ranging between 60 and 80 in random directions. Each step is two kilometers long. The visited cubes become part of a higher density area with a density of $1g/cm^3$, with rest of the cubes being set to $0g/cm^3$. The authors generate a dataset of 20000 density models using this method. The corresponding gravitational fields are calculated using a method specific to calculating the gravity of a point on a surface, with mass distributed in prisms underneath.

To test the accuracy of their method, the authors construct different test density models, calculate the gravitational field and apply their inversion. The model is able to reconstruct the general shape of the high densities regions well, although the density values are not exact, especially when more complex shapes are tested. The model is more exact in regions closer to the surface, with higher divergences in the deeper sections. It is able to distinguish two shapes above one another, with space in between, as can be seen in figure 2.2, but the authors note that accuracy needs to be improved in this case. They also evaluate the difference between their method with the additional constraint term described above and without. They find that the constraint term reduces the error between actual and predicted density distribution slightly, and noticeably reduces the error between actual and predicted gravitational field. In addition to the synthetic tests, the authors also apply their model to real measurements of gravitational data gathered in East Antarctica. They find that their model yields a density distribution which is consistent with previous studies.

The authors of [15] use a similar method, the main difference being the addition of residual connections to a U-Net based approach. They attain slightly better results in comparison to a U-Net based approach. Another similar approach is presented in [5], where the authors use a similar data creation method, but use a 3D-Unet instead of the two dimensional version. Results are difficult to compare with the other presented works, as F1-Scores are given instead of attained errors.

The models developed in the presented works work well on the applications on earth for which they were designed. In the context of this thesis however, they would have to be adapted to work on small-bodies in space. The main difference between these applications is the kind of gravitational data available. Where in the case of earth the data is usually measured or generated in a two dimensional plane with the mass distributed underneath, measurements in space are usually taken by space probes in orbit around a body of interest, see for example [14]. This leads to measurements in the shape of spherical trajectories around the body, and this spatial information should be utilized when adapting approaches based on convolutional networks. If input data for a network to predict a small-bodies density distribution were limited to a two dimensional plane on one side of the body, the model would probably yield less accurate results in the parts farthest away from the gravitational measurements, as could be seen in [34]. The inclusion of data closer to the densities in question might improve the results. There have been approaches to adapt the U-Net to work on spherical data, for example in [35], where the authors adapt the convolution operation to spherical surfaces and build a Spherical U-Net with it. The concept of the 3D U-Net as proposed in [3] and adapted to gravitational inversion in [5] is also interesting in the context of this thesis, as the output of the inversion is a density distribution in three dimensional space. The inversion might thus benefit from the additional 3D processing capabilities the 3D U-Net provides. Ideally, a U-Net based network for gravity inversion on small-bodies would process the spherical gravitational data of the input with spherical convolutions in the encoder part of the network, similar to [35]. In the decoder 3D operations similar to the ones in [3] would be used to arrive at the three dimensional output of the density distribution. The problem with this approach is the fact that the architecture of the U-Net is symmetrical and this results in the shape of the inputs and outputs being usually similar. For example, the Spherical U-Net takes a spherical surface as an input and also yields a spherical surface as an output [35]. Similarly the 3D U-Net gets three dimensional input data and outputs three dimensional data. The authors of [34] are able to generate three dimensional output from two dimensional input by encoding the third dimension in the channels of their output image, but the data flowing through the U-Net is still two-

dimensional. This is important because one of the features of the U-Net is the concatenation of features from the encoding part with features from the decoding path. This would be difficult to accomplish with two representations in the encoder and decoder part of the network that are as different as the spherical representation of the Spherical U-Net and the three dimensional representation of the 3D U-Net. Due to these challenges I chose to focus on the NeRF inspired approaches instead, which will be presented in the following.

2.3.2 *Inverse Gravity Modeling with a NeRF inspired approach*

The authors of [11] propose a solution to the inverse gravity modeling problem, where they train a neural network to represent a mapping from Cartesian coordinates to the body density. The training procedure of the so called GeodesyNets can be seen in figure 2.3. The input in the form of Cartesian coordinates is fed into the network, flowing through a number of fully connected layers and the output of the network being the predicted density at the input point. An arbitrary number of points and their corresponding densities can be sampled from the network like this. The gravitational field corresponding to these densities can then be calculated and compared to either a gravitational field obtained through forward modeling or actual measurements of a gravitational field. The error between the predicted and ground-truth gravitational field is calculated and used in backpropagation to update the neural networks parameters, to minimize the error. This makes the gravitational field the predicted density produces more closely resemble the ground-truth gravitational field step by step. Notably, the GeodesyNet does not require a shape model to be given and can learn the shape of the body from the gravitational data. This stands in contrast to the classical inverse gravity modeling methods discussed in section 2.2, which need a shape model. However, GeodesyNets can incorporate a shape model for improved results.

The authors generate the ground-truth gravitational data using the mascon method (see section 2.1.3). The masses are distributed by subdividing the volume of the body into tetrahedra and placing the masses at their centroids. They create models for “the asteroids 433 Eros, 25,143 Itokawa, and 10,1955 Bennu and the comet 67P Churyumov–Gerasimenko, as well as a fictitious Planetesimal and a toroidal-shaped body” [11]. These models are created assuming homogeneous mass distributions, however for Bennu, Itokawa and Planetesimal the authors create additional models with heterogeneous mass distributions. To standardize the approach across the different bodies, non-dimensional units for this ground-truth data are introduced. The body is set into the hypercube with dimensions $[-1, 1]^3$ and scaled so that “so that the maximum absolute value of its coordinates is

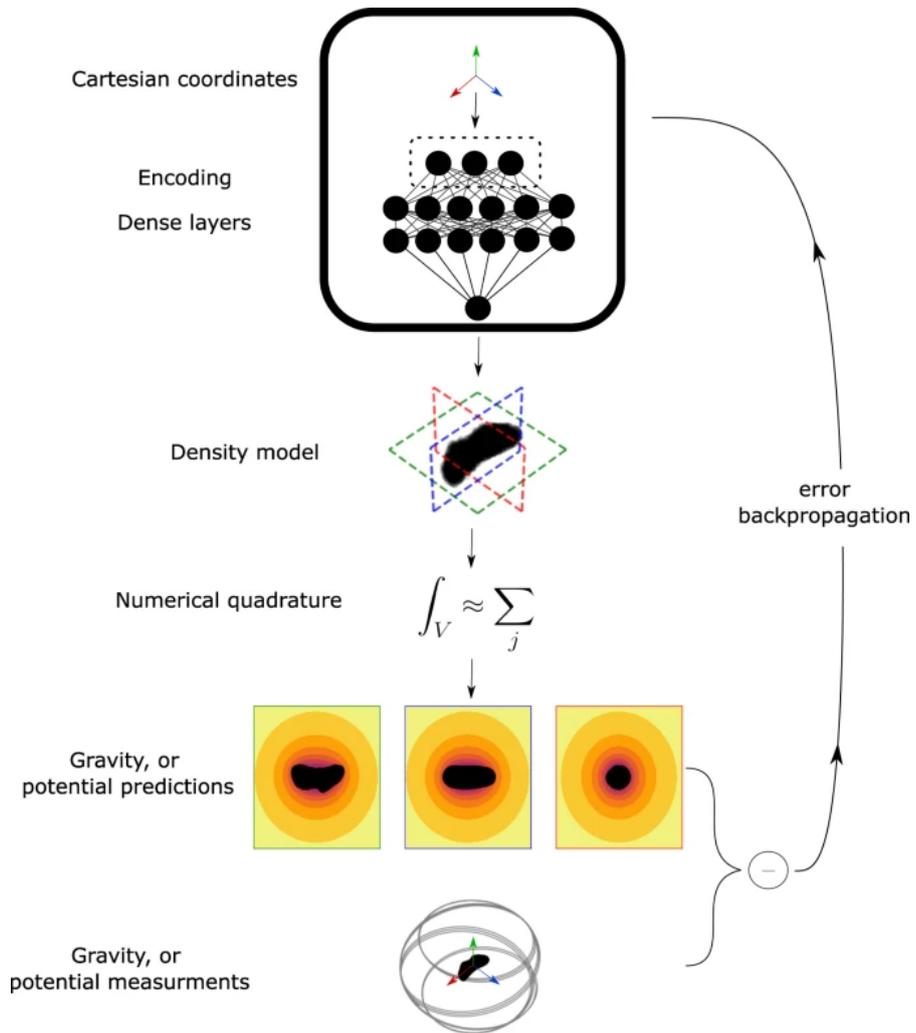


Figure 2.3: Training procedure of a GeodesyNet. Figure from [11].

$\delta_{max} = 0.8''$ [11, Supplementary Method 4]. The unit of mass is set to the body mass and the gravitational constant to one.

The authors train a GeodesyNet for each of the bodies and evaluate the error with respect to the ground-truth gravitational field at three different altitudes: 0.05, 0.1 and 0.25 units of length. They find an error between 0.11% and 0.28% for all cases. The authors compare their method to [33], where a hybrid mascon and spherical harmonics method is used. In this work, the parameters of the model are determined using least-squares estimation. Notably, this does not lead to a direct representation of the density distribution of a body, but the model allows for arbitrary calculations of gravitational fields. The authors note that the relation of the model parameters to a density distribution still needs to be determined. The GeodesyNet is found to reach comparable accuracies to some of those presented in [33], even though no shape model is used. The authors of [11] note that the usage of a shape model with the GeodesyNet “would ... result in orders of magnitude smaller errors” [11, Supplementary Method 5]. They point out that it is generally difficult to compare results of gravity modeling across the literature, as a common validation practice does not exist. To allow for better quantitative comparison, the authors implement their own mascon approach, which does not require shape information. This approach uses a uniform grid of masses inside the hypercube with dimensions $[-1, 1]^3$, which was also used to create the ground-truth data for the GeodesyNet. These masses are then optimized to fit the ground-truth gravitational data using gradient descent. The number of mascons is set to equal the number of parameters in the largest GeodesyNet used. Both models are trained on the homogeneous ground-truth data and evaluated close to the surface of the body and farther away, about 0.15 length units and between about 0.15 and 0.3 length units respectively. When looking at the performance farther away, both models perform well, with the better performing method depending on the studied body. Closer to the surface, the GeodesyNet performs consistently better than the mascon method, confirming this case as a slight weakness of the mascon method, as discussed in section 2.1.3. The authors also test the performance of the GeodesyNet when utilizing a shape model on the heterogeneous ground-truth data. They find that this improves the error in comparison to not utilizing a shape model, especially at low altitudes above the bodies surface. Utilizing the shape model also results in a density distribution that is closer to the ground-truth density distribution.

Summing up, the GeodesyNet provides several advantages over classical method for gravity inversion. Firstly, it does not require a shape model, but can utilize one to improve the results. It also does not need any assumptions to be made about the internal density distribution and yields a continuous density distribution, in contrast to a mascon based method. The gravitational field corresponding to

the GeodesyNet yields better results closer to the surface of the body as well, when compared to a pure mascon based approach.

BUILDING THE DATASET

3.1 OBTAINING AND CLEANING UP THE DATA

In anticipation of utilizing an approach based on a convolutional neural networks to do inverse gravity modeling as described in section 2.3.1, I created a training dataset for such an approach. The dataset consists of a number of asteroid density distributions and their corresponding gravitational fields. It is based on the shape models and additional information about small-bodies provided on the 3D Asteroid Catalogue website [6]. The shape models and information on this website were gathered from many different sources, for more details refer to the sources provided on the pages for the different small bodies.

Since there is no easy way on the website to download all the shape models at once, together with the information about the corresponding bodies, I started out by downloading the websites sources to my computer, together with the shape models. I then wrote a python script to process the information from all the individual html files into one csv file, to allow for easier processing. While looking over the data, I noticed that only 26 out of the 1635 asteroids came with information about their mass, although 1563 came with a diameter. Since the total mass of a body is needed to calculate a gravitational field that is scaled correctly, I decided to reconstruct the masses of the bodies by using the diameter. I assumed that the volume and total mass of a body would be roughly correlated. Since the volume and radius of a sphere are in a cubical relation I figured trying to fit a polynomial of degree three to the available data would be worth investigating. To make sure that the resulting function would yield a mass of zero for a mean diameter of zero, I disabled the fitting of the intercept. I also forced the regression to have positive coefficients to make sure positive diameters would always yield positive masses. The result of this experiment is shown in the upper part of figure 3.1. The learned function fits the data well and seems to represent the relation between mean diameter and body mass meaningfully. I also tested other regression methods like the Random Forest, the results of which can be seen in the bottom part of figure 3.1. I was able to get a better score with the Random Forest in comparison to the Polynomial Regression when I split the available data with mass into a train and test set, but I think in a case with so little training data the Random Forest is prone to overfit. It looks like this happened when looking at the plot of the Random Forest in figure 3.1. To avoid overfitting, I

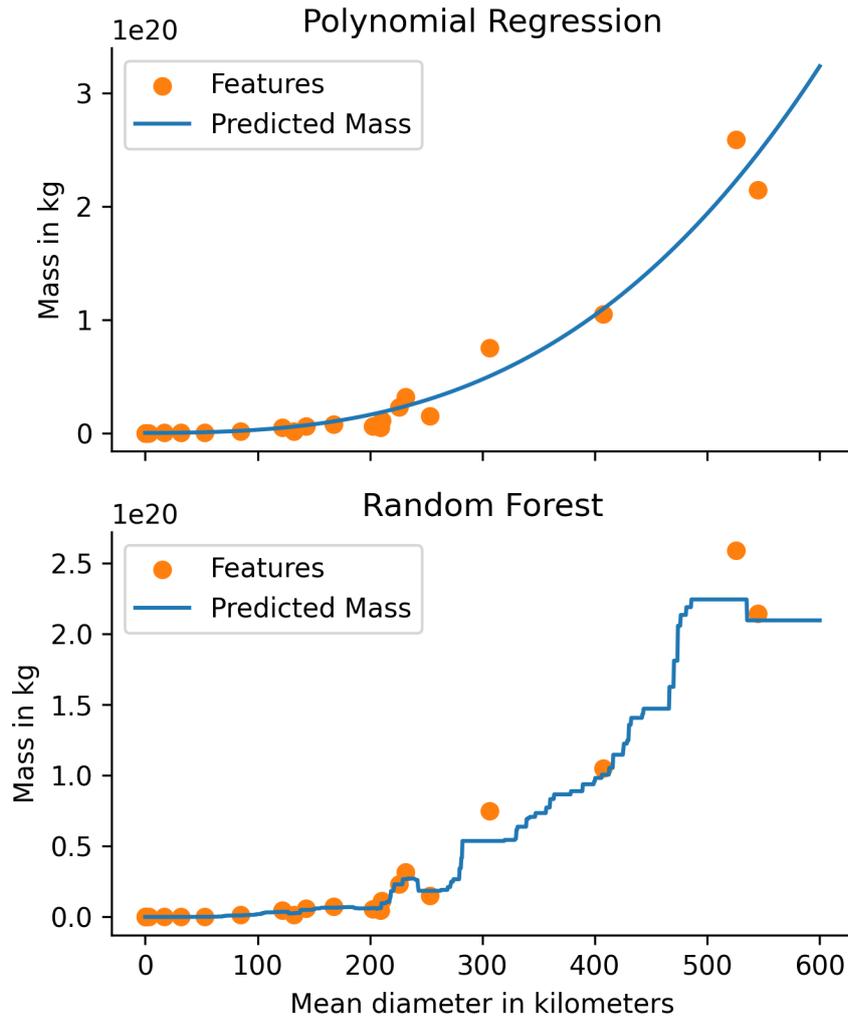


Figure 3.1: Predicted body mass with Polynomial Regression and Random Forest using the mean diameter.

think in this case the simpler model which can be more easily derived from the real physical relationship of these two properties is the better choice.

3.2 MODELING AND CALCULATING THE GRAVITATIONAL FIELD

With the masses for most of the asteroids restored, I started working on the mascon method to calculate the asteroids gravitational fields. After loading an asteroids mesh, I scale the maximum extent of the asteroid to be its diameter, since the real extents in x , y and z direction are again only available for a limited number of asteroids. This is only an approximation of the actual size of the asteroid, but the size is in the correct order of magnitude. The next step is the subdivision of the asteroid into multiple mascons. As I was building the dataset

in anticipation of training a model like the ones presented in section 2.3.1, my aim was to create mascon models with the same dimensions for all asteroids. These methods require all of their inputs to have the same dimensions. Mascon models usually produce different numbers and volumes of mascons, depending on the shape of the modeled body. All the methods presented in section 2.1.3 share this property. My initial solution to this problem was to fill the bounding box of the asteroid with mascons shaped like cuboids. The length of each cuboid in each dimension is the length of the bounding box in this dimension divided by the desired number of subdivisions per dimension. This leads to the volume of the bounding box being completely covered by mascons. Figure 3.2 shows this method with ten subdivisions leading to 1000 mascons. The advantage of this method is the relatively small number of mascons which do not overlap with the asteroid while always providing the same dimensions no matter the shape of the asteroid. A potential issue is the fact that information about the position and proportions of the mascons gets lost when the mascons masses are the only input to a machine learning model. This leads to the property that an asteroid can be scaled in different dimensions and will still yield the same mascon representation from the view of a machine learning model, as long as the volume of the mascons stays the same. For example, an asteroid with a bounding box of dimensions $(10, 1, 1)$ will yield the same representation as the same asteroid rescaled to a bounding box of $(1, 10, 1)$. To address this issue, I implemented a different subdivision method, which is similar in layout to the mascon based approach in [11], which the authors compare the GeodesyNet to. With this approach, I place the asteroid into a cube, which is then subdivided into a grid of cubes. Figure 3.3 shows the result of this for ten subdivisions in each dimension and the side length of the cube set to the maximum extent of the asteroid. This method leads to the same number of mascons for all asteroids as well, but the mascons are all cubes and therefore have the same relative dimensions across asteroids. This method also does not have the scaling invariance problem discussed before. A potential drawback of this method might be the greater number of mascons that do not overlap with the asteroid. This leads to a lower resolution of mascons covering the asteroid compared to subdividing the bounding box with the same total number of mascons. Both of these methods need to be compared in practice, but subdividing the cube has fewer theoretical issues.

After subdividing, densities can be assigned to all of the mascons. I implemented both a uniform distribution of density, as well as a random one. For the uniform distribution, the total mass is divided equally between the mascons whose centerpoints lie inside of the asteroid. I then calculate the density by dividing this mass by the volume of each mascon. For the random distribution I sample a random

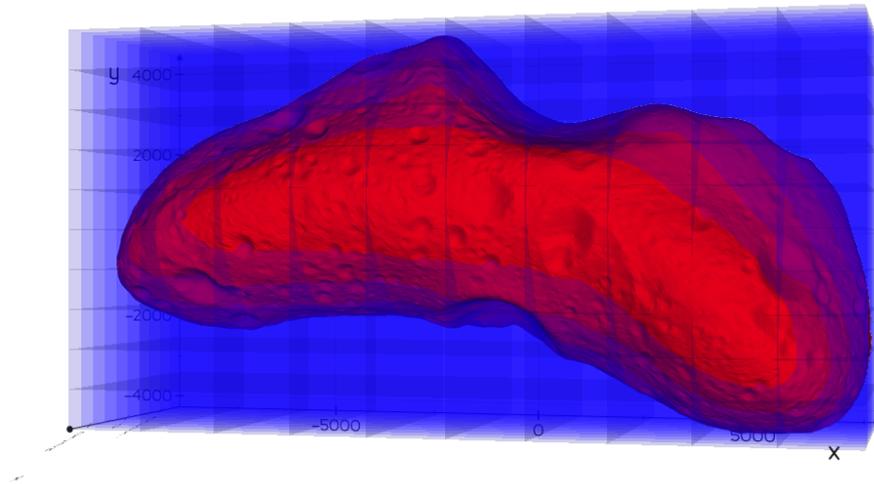


Figure 3.2: Filling the bounding box with mascons. Here ten subdivisions were used, leading to 1000 mascons.

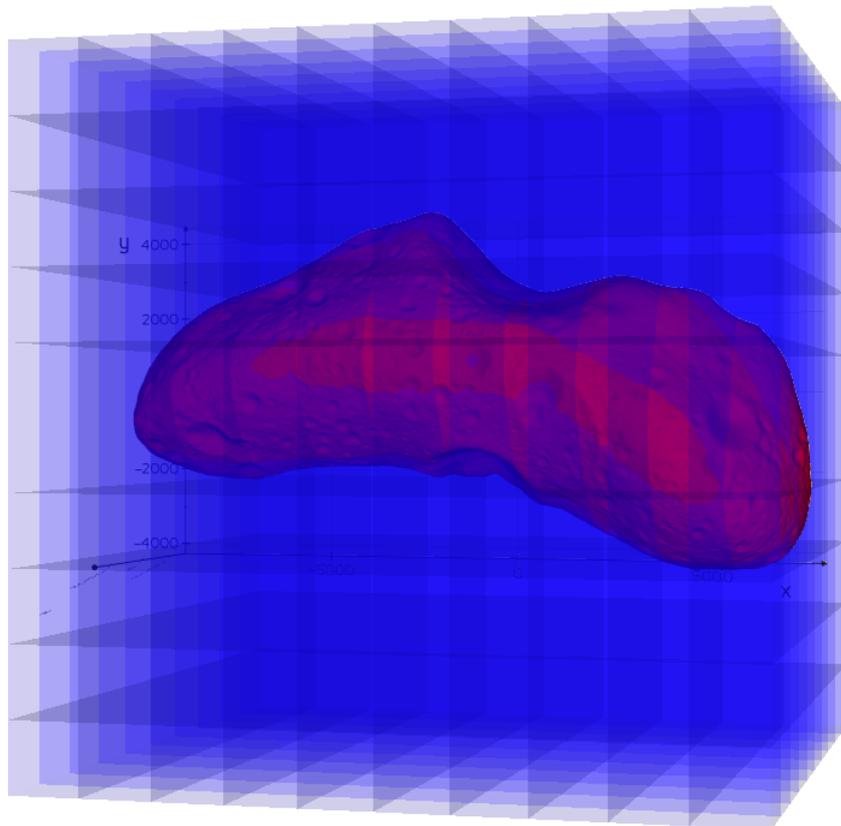


Figure 3.3: Subdividing a cube with the asteroid placed inside. Here the cube's sides have the length of the asteroid's maximum extent. Each dimension has ten subdivisions leading to 1000 mascons.

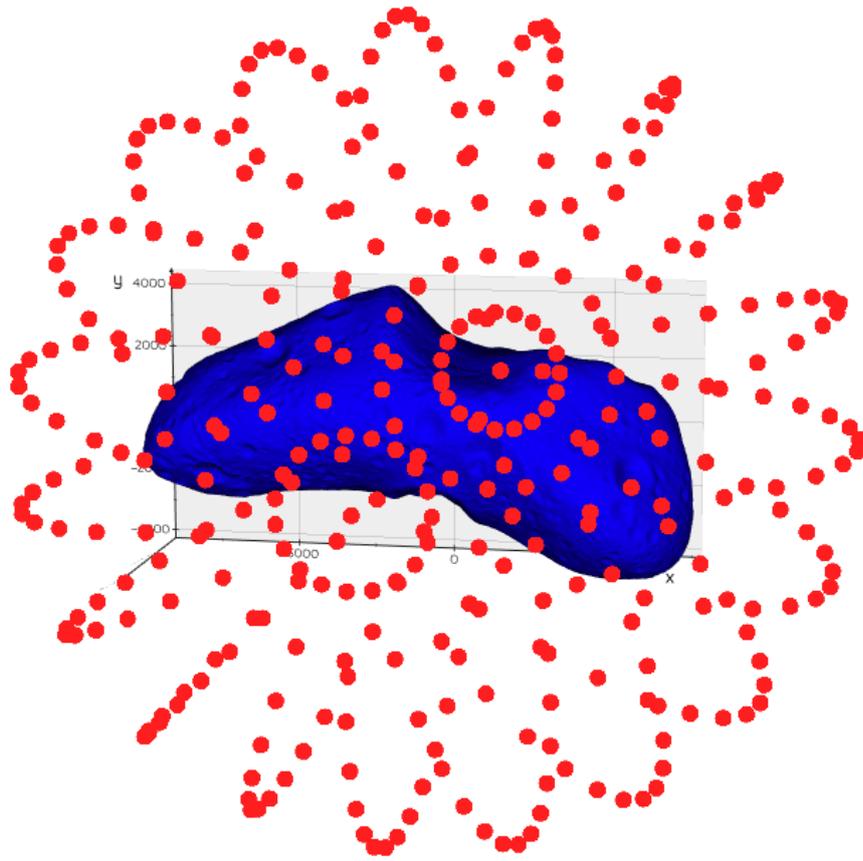


Figure 3.4: The sampled points creating a sampling sphere around the asteroid.

value from the continuous uniform distribution in the interval $[0, 1)$ for every mascon whose centerpoint lies within the asteroid. I then scale these values so their sum equals the total mass of the asteroid. The densities are then calculated again by dividing the masses by the mascons volume.

To be able to calculate the gravitational field corresponding to the density distributions, I create a sphere of points at which the gravitational acceleration or potential can be calculated. To describe the points on the sphere I use spherical coordinates similar to the ones introduced in [10, p. 18]. A point P on the sphere is described by its radius r from the center of the sphere, as well as the angles λ and θ . The angle λ describes the angle between P and the x -axis in the x - y plane and the angle θ is the angle between the line connecting the center of the sphere and the point P and the z -axis. By subsampling the angles λ and θ I create evenly spaced rings of points around the z -axis, which are themselves evenly spaced in the range of θ . Figure 3.4 shows the resulting sampling sphere.

With the mascons and the sampling points determined, I calculate the gravitational accelerations or potentials with equation 2.1 or equation 2.3 respectively.

3.2.1 Utilized Libraries

I used vedo for the handling and some operations on 3D objects and meshes [19]. Numpy is used heavily for data handling and its functions [9]. I used numba to JIT compile some of the most computationally intensive functions to improve speed [13]. To parallelize the code and work on multiple asteroids at once I used the *multiprocessing* package from Python's standard library.

3.2.2 Usage

After creating the dataset I decided to implement the incorporation of additional information into the inversion process, based on the method described in [11]. Since this method does not need the amount of data in this dataset to be trained, I was unfortunately unable to use this dataset in this thesis. It can still serve as a basis for future work, for example to train gravity inversion models like the ones described in section 2.3.1.

MODIFICATIONS TO GEODESYNET

The aim of this thesis is to achieve an inverse gravity modeling method whose performance is comparable to other inversion methods and which is able to incorporate additional expert information into the inversion process, in the form of regions inside of the asteroid with predefined density. I have presented and compared different machine learning based methods for inverse gravity modeling in section 2.3 and decided to use the most promising for asteroids, which are the GeodesyNets introduced by [11]. I also chose this approach because it seemed the most straightforward to implement the introduction of additional information into the inversion process for. In this chapter, I will describe the modifications I made to the GeodesyNet, so that it can incorporate these regions of predefined density into its training process. I will refer to methods which incorporate regions of predefined density as guided inverse gravity modeling from hereon.

4.1 THE LOSS FUNCTION

In the unmodified form implemented by [11], a training iteration of the GeodesyNet starts with sampling its output on a grid inside the hypercube with dimensions $[-1, 1]$. The gravitational acceleration or potential is then calculated at a number of target points outside of the asteroid using the densities sampled from the GeodesyNet. This gravitational field represented by the GeodesyNet is used to calculate the loss of the neural network together with the gravitational field calculated from the mascon ground-truth. After comparing a number of different loss functions, the authors of [11] settle on a modified version of the Mean Absolute Error. They introduce a mass normalization factor κ which is multiplied with the networks predictions to scale them. This allows the network to focus on learning the difference to a homogeneously filled volume and not on finding the correct absolute mass of the body as well. The value of κ is calculated analytically in each training iteration of the network and the authors find the optimal value for it to be

$$\kappa = \frac{\sum_{i=1}^n \hat{y}_i y_i}{\sum_{i=1}^n y_i^2} \quad (4.1)$$

with $y_i, i = 1..n$ being the ground-truth accelerations or potentials from the mascon model and $\hat{y}_i, i = 1..n$ the predicted accelerations or potentials of the network [11, Supplementary Method 1]. When

incorporated into the Mean Absolute Error, the resulting loss function is

$$L_{\kappa MAE} = \frac{1}{n} \sum_{i=1}^n |y_i - \kappa \hat{y}_i| \quad (4.2)$$

To implement the guided inverse gravity modeling for the GeodesyNets, I took inspiration from an optional part of the training implemented by the authors of [11] called Vision Loss. The Vision Loss is an additional term that is added to the loss calculated on the gravitational field. It is calculated by sampling a number of points outside of the body, for which a shape model is needed. The expectation is for these points to have a mass of zero, as they are outside of the body. Deriving the Vision Loss now involves calculating the Mean Absolute Error between the sampled points and zero. The result is added to the loss calculated on the gravitational field to form the total loss. This urges the densities at the points to be zero.

To implement guided inverse gravity modeling, I added a similar term to the loss based on the gravitational field, which I call the guidance loss. The predefined regions of density can be given by the user either in the form of spheres or a plane that splits the asteroid into two halves. The spheres are characterized by their centerpoints and radii, as well as densities which define the desired densities in the volumes covered by the spheres. The plane is defined by a point in the plane and the planes normal vector. The density corresponding to the plane describes the desired density on the positive side of the plane. I calculate the guidance loss by first sampling a number of points inside of the spheres or inside the asteroid on the positive side of the plane. The output of the GeodesyNet is sampled at these points and the guidance loss calculated using

$$L_{\kappa G} = \frac{1}{n} \sum_{i=1}^n |y_{\delta_G} - \kappa \hat{y}_i| \quad (4.3)$$

with y_{δ_G} representing the predefined guidance density and $\hat{y}_i, i = 1..n$ the predicted densities at a number of points inside of the predefined regions or region. The mass normalization factor κ is also used to scale the networks predictions, so that the densities \hat{y}_i are correct. Incorporating the guidance loss into the total loss leads to the following equation for the total loss:

$$L = L_{\kappa MAE} + L_{\kappa G} \quad (4.4)$$

Initial testing with this approach revealed that it performed worse when compared to the ground-truth based on the mascon model than the unmodified GeodesyNet, which does not incorporate additional

density information. I looked at how the two components of the loss, gravitation based and guidance density based, developed over the duration of a training run. Figure 4.1 shows both loss components as well as the loss of an unmodified GeodesyNet for the last 2000 iterations of the run. It looked to me like the network was optimizing the guidance loss at the detriment of the gravitational field based loss, which would explain the increased error with regard to the ground-truth data. To test this hypothesis and correct this potential issue, I introduced a factor $g, 0 \leq g \leq 1$ to be able to scale both losses. This results in the modified equation

$$L = (1 - g) \cdot L_{\kappa MAE} + g \cdot L_{\kappa G} \quad (4.5)$$

to calculate the total loss. Introducing the factor and finding a suitable value for it through empirical means lead to the new losses depicted in figure 4.2. The loss based on the gravitational field is now an order of magnitude smaller and is close to the loss of the unmodified GeodesyNet. The guidance loss nearly doubles after introducing the scaling factor when comparing the iterations with the smallest overall loss across both runs. This is a worthwhile trade-off for the improvement in the gravitational field based loss.

With the addition of scaling the losses the guidance loss works well and achieves the goal of being able to incorporate expert information and previous knowledge about the density in certain regions into the inversion process of the GeodesyNet.

The optimal scaling factor varies between different bodies and guidance regions. Since it is impractical to do an empirical study for every new body and guidance region to find the optimal scaling factor, I introduced a method to scale the guidance loss automatically. Introducing a good scaling factor found through empirical means leads to the gravitational field based loss and the guidance loss ending in the same order of magnitude at the end of a training run, as seen in figure 4.2. To force this throughout the whole training run, I scale the guidance loss to always be the same absolute number as the gravitational field based loss:

$$L_{SG} = \left(\frac{L_{\kappa MAE}}{L_{\kappa G}} \right) \cdot L_{\kappa G} \quad (4.6)$$

The resulting automatically scaled guidance loss L_{SG} is then just added to the gravitational field based loss to form the total loss.

$$L = L_{\kappa MAE} + L_{SG} \quad (4.7)$$

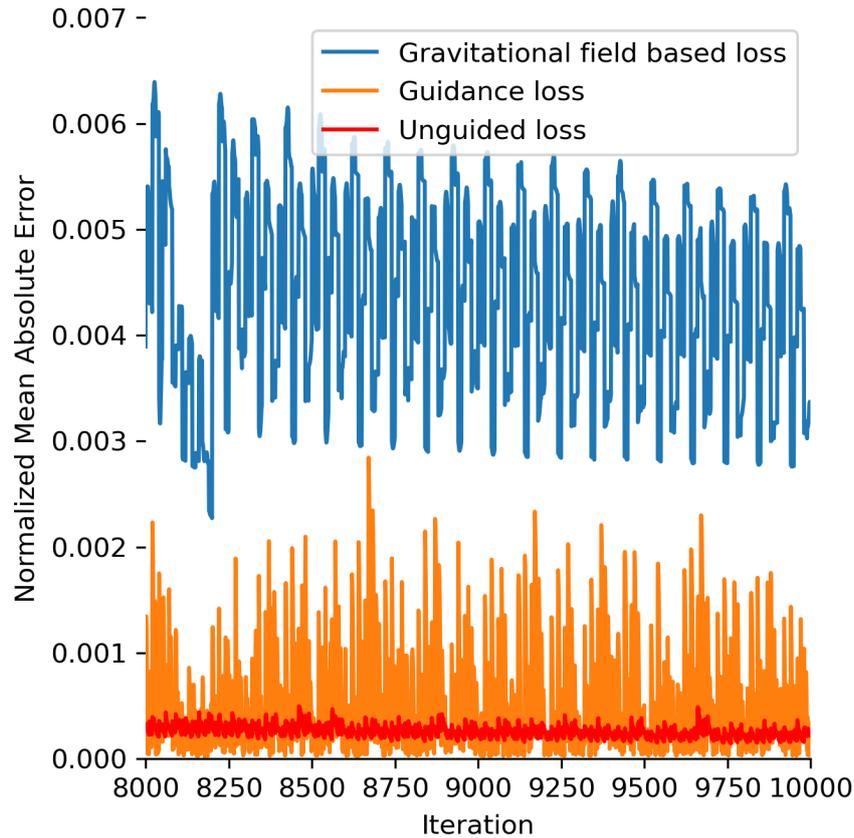


Figure 4.1: Comparing the last iterations of a training run with the initial implementation. The separated components of the loss are shown, one based on the ground-truth gravitational field and the other on the given guidance density for one guidance volume, with the loss of an unmodified GeodesyNet depicted for comparison.

4.2 HETEROGENEOUS GROUND TRUTH DATA

To accurately reflect the use-case that a scientist with expert knowledge might use the guidance density to specify areas that in fact match the actual density of an asteroid, I created versions of the ground-truth data that reflect the predefined guidance densities. The authors of [11] provide mascon models with homogeneous mass distributions for “the asteroids 433 Eros, 25,143 Itokawa, and 10,1955 Bennu and the comet 67P Churyumov–Gerasimenko, as well as a fictitious Planetesimal and a toroidal-shaped body (...) [they] call Torus” [11]. For Bennu, Itokawa and Planetesimal the authors provide models with heterogeneous mass distributions as well. For this thesis, I chose to work on the model for the comet 67P Churyumov-Gerasimenko, as well as the asteroid Bennu. Churyumov-Gerasimenko has a density of about 1.7, Bennu of 0.7, in the unmodified version. My first heterogeneity is a small sphere in the middle of Churyumov-Gerasimenko, with a

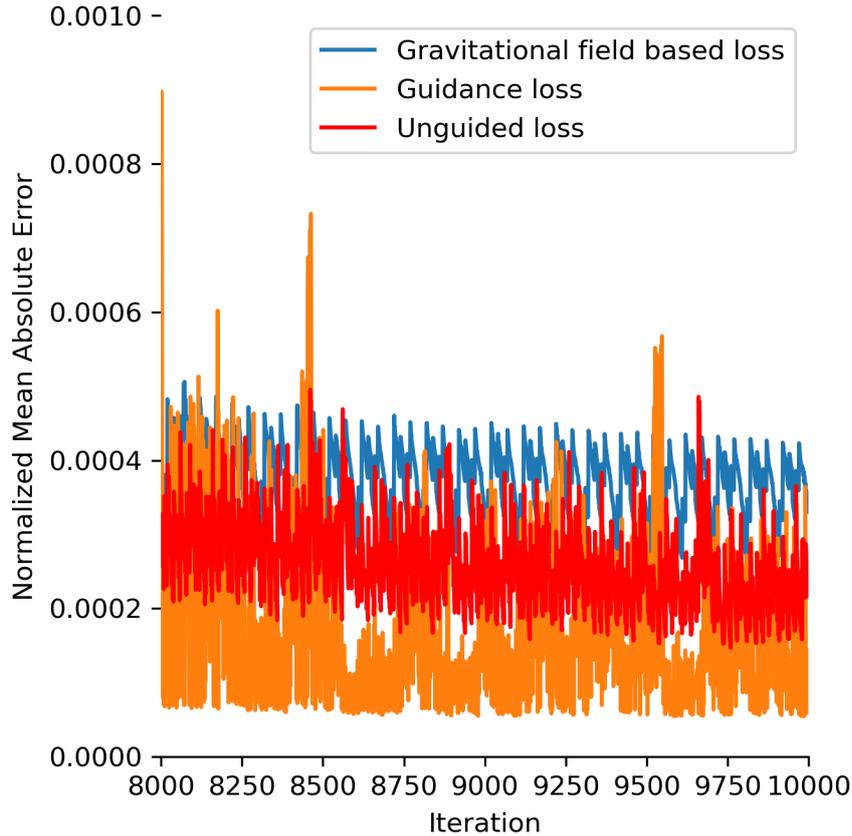


Figure 4.2: Comparing the last iterations of a training run after introducing a scaling factor to weight the two loss terms. The loss of an unmodified GeodesyNet is shown for comparison.

density of 1, for Bennu a placed a small sphere of density 1.2 close to the surface. The second type of heterogeneity was created by placing three spheres of different sizes inside of the asteroid. For Churyumov-Gerasimenko I generated two versions of this, one with densities 1, 0.5 and 2.5 for the spheres, the other with density 0 for all spheres, to simulate cavities inside the body. For Bennu I choose densities of 1, 0.5 and 1.5. To test specifying the guidance density by a plane and evaluate a heterogeneity much bigger proportionally to the rest of the asteroid, I introduced a heterogeneity for the head of the comet similar to the way the authors of [11] did for the asteroid Itokawa. The result can be seen in figure 4.3. I generated a similar version for Bennu, with the guidance density being specified for about a third of the asteroid.

The ground-truth is generated by subdividing the interior volume of the surface mesh of a given small body into tetrahedra, using TetGen [26]. A mascon is placed at the center of each tetrahedron and a mass proportional to the tetrahedrons volume is assigned to it, with all masses totalling the bodies mass, in case of a homogeneous mass distribution. In the version used by the authors of [11], this method

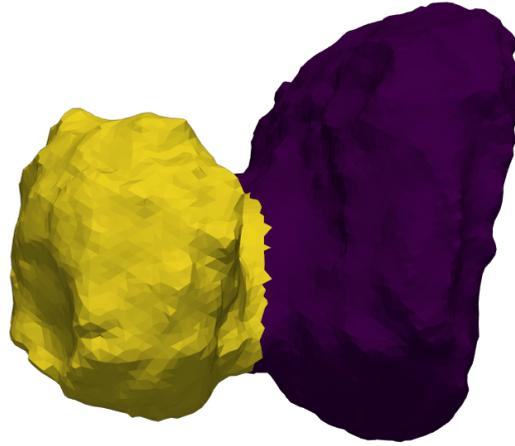


Figure 4.3: 67P Churyumov-Gerasimenko with a density heterogeneity introduced for its head.

results in a mascon model with about 57000 mascons for the comet 67P Churyumov-Gerasimenko, and about 38000 for Bennu. To create the heterogeneous versions of the bodies, I first selected all tetrahedra whose centers overlapped with any of the guidance spheres or are on the positive side of the plane defining the guidance area. I then assigned a mass to each mascon, so that the density calculated with respect to the tetrahedrons volume matched the guidance density. This process results in either a loss or a gain of total mass of the body, as mass is removed or added to the mascons in question. To bring the total bodies mass back to its state before the modifications, I removed or added a proportionally equal amount of mass to each mascon that was not affected by the modifications. While modifying the generation of the mascon model to create the heterogeneity of the small sphere in the middle of the body for Churyumov-Gerasimenko, I noticed that the volume of the tetrahedra whose centerpoints overlap with the sphere is about double the volume of the sphere, leading to a mass heterogeneity about double the size it should be. The reason for this is the relatively coarse resolution of tetrahedra on the inside of the body compared to the spheres size, leading to only two tetrahedra overlapping with the

sphere which are unable to accurately approximate the sphere. To solve this problem I increased the number of tetrahedra used to subdivide the body, leading to a model with about one million mascons for Churyumov-Gerasimenko and about 175000 for Bennu. The higher resolution leads to more tetrahedra centerpoints overlapping with the sphere which results in a better approximation and a volume of the tetrahedra that is much closer to the volume of the sphere.

4.3 GUIDANCE POINT SAMPLERS

To be able to calculate the guidance loss, the density needs to be evaluated inside of the guidance regions. I was able to utilize the spherical sampler provided by the authors to do this for the spherical guidance regions, and chose to sample 1000 points for every sphere every 10 training iterations. For the guidance region defined through a plane, I implemented my own plane guidance sampler. I start by creating random coordinates in the region covered by the bodies extent. I then calculate the distances of all the points to the guidance plane, defined by a point in the plane and its normal vector. For all points where the distance to the plane is negative, meaning they are on the wrong side of the plane, I mirror them across the plane to the other side. I then check if all of the points are inside of the asteroid and discard those that are not. This is repeated until the desired number of points is found. For the plane guidance sampling I chose to sample 100000 points every 10 training iterations to adequately cover the guidance region, as I expect the plane guidance regions to be much larger than the spherical guidance regions.

4.4 PERFORMANCE CONSIDERATIONS

The results in this thesis were mostly generated with 10000 iterations of training with a batch size of 1000 points (samples of the gravitational field around the asteroid) and 300000 integration points sampling the density inside the unit cube, to allow for comparisons with the results of [11]. Testing these parameters with the unmodified code provided by the authors of [11], a full training run took eight to nine hours. The machine I was testing with was equipped with a Nvidia RTX 2080 Super, an AMD Ryzen 3900X and 32GB of RAM. In one of the notebooks from the code accompanying [11] the authors write that a full training should take roughly one hour on a Nvidia RTX 2080ti. When I contacted the author of [11], Dario Izzo, he told me that a training run for the results of the paper took about two hours.

While investigating the codebase for bottlenecks, I found that the most computationally intensive part of a training iteration by far is the calculation of the gravitational field at the target points around the asteroid based on the densities sampled in the unit cube. In the

original code by the authors of [11] this is implemented as a loop over the target points. I rewrote this code so that the gravitational accelerations are calculated in batches, allowing for multiple target points to be processed at the same time on the GPU. With enough GPU memory to process all target points in one batch, this leads to a speedup greater than 2x, reducing the time for a full training run down to between three and a half and four hours, tested on a Nvidia V100 GPU with 32GB of GPU memory. On GPUs with less memory like the 8GB of the RTX 2080 super, the batch size needs to be carefully selected, so that the data fits into the GPUs memory, while still allowing the maximum number of points to be processed at the same time. I have found processing the points in 25 batches, leading to a batch size of 40 points to work well on the RTX 2080 super. Interestingly, processing the points in batches also leads to a speedup on GPUs where the full data would fit into the memory at the same time. Using a batch size of 40 points leads to a final training time of between two and three and a half hours, depending on the guidance regions used.

The ray-triangle intersection algorithm that is used by the authors of [11] to determine if points are inside or outside a body is implemented as a loop over the points for which to check as well. When the guidance region is specified through a plane this algorithm is used heavily, leading to a slowdown in the training. To fix this problem, I rewrote the implementation to make use of batch processing as well.

4.5 POST TRAINING

The training time of a GeodesyNet of three and a half to four hours makes it difficult for a user to test different guidance regions quickly and see which one fits the given gravitational field best. To address this issue I split the training of the GeodesyNet into two steps. First an unmodified GeodesyNet gets trained for 10000 iterations to fit the given gravitational field, without specifying any guidance regions. The resulting model serves as the base for the next step, where the training is continued with the addition of guidance regions. To evaluate the performance I compare the training in two steps to a training run that is trained with the guidance loss from the start, for 10000 iterations.

4.6 MASCON BASED COMPARATIVE METHOD

The authors of [11] use a mascon based approach they call mascon-CUBE, to evaluate the performance of the GeodesyNet in comparison to, as discussed in section 2.3.2. I adapted this approach to be a suitable comparative method for the GeodesyNet with guidance, by implementing guidance for this approach as well. My initial approach was to except the mascons lying inside the volume covered by the guidance

spheres or the guidance plane from being optimized with regard to the gravitational field during the gradient descent. Instead, I set their masses to correspond to the predefined densities in every iteration of the optimization. During the training the total mass of all mascons in the masconCUBE constantly fluctuates. The guidance density ρ_G is defined with regard to the ground truth, where all mascon masses sum up to one. This problem occurs as well when comparing the gravitational accelerations obtained through a GeodesyNet or masconCUBE to the ones calculated on the ground-truth. The authors of [11] solved this by introducing the scaling factor κ , which I described in section 4.1. To make sure that the guidance density is correct with regard to the fluctuating total mass, I used this factor as well. Since the factor κ is calculated using the predictions of the masconCUBE on the gravitational field, and the guidance masses are in turn needed to calculate the predictions of the masconCUBE, I can only ever use the scaling factor from the last training iteration to scale the guidance masses. This might lead to slightly distorted guidance masses, harming the results. The parameters of the masconCUBE are squared to form the mascon masses to ensure they are positive [11]. Taking this into account, I take the square root of my defined guidance masses, so that they are correct in the squared form. Together with the scaling factor, I set the guidance masses m_G in every iteration to

$$m_G = \sqrt{\frac{\rho_G \cdot V_{mascon}}{\kappa}} \quad (4.8)$$

with ρ_G denoting the guidance density and V_{mascon} denoting the volume of a mascon.

As an alternative approach I implemented a guidance loss similar to the one for the GeodesyNet, described in section 4.1. To calculate the loss I first select all points of the masconCUBE which lie inside the guidance regions. I then calculate the corresponding densities for the mascon masses at these points by dividing by the volume of a mascon V_{mascon} . With the densities, I can calculate the guidance loss using the Mean Absolute Error:

$$L_{\kappa G} = \frac{1}{n} \sum_{i=1}^n \left| \kappa \cdot \frac{m_i^2}{V_{mascon}} - \rho_G \right| \quad (4.9)$$

The mascon masses are represented by $m_i, i = 1..n$, and the guidance density by ρ_G . The scaling factor κ is used again to scale the squared masses correctly in relation to the guidance density. The guidance losses for all guidance regions are added up and then added to the gravitational field based loss, using a guidance factor determined from the results of 5.2.

Testing both approaches and comparing the results to the ground-truth revealed that the second approach achieves better results with

regard to the gravitational field, which is why I chose to use it as a comparative method for the GeodesyNet. Training a masconCUBE for 10000 iterations takes about an hour on a Nvidia V100 GPU. The results of the comparison of both guidance approaches for the masconCUBE will be presented and discussed in section 5.3.

4.7 VALIDATION PROCEDURE

To assess the performance of the GeodesyNet and the masconCUBE and to compare the results, I use the validation procedure used by the authors of [11]. It consists of two different approaches to selecting the validation points at which a model is compared to the ground-truth. The first approach was introduced by the authors to enable a comparison to [33]. The validation points are sampled randomly at a low altitude between the surface and about 0.15 length units above the surface, as well as at a higher altitude between about 0.15 and about 0.3 length units above the surface. While testing this method on the comet 67P Churyumov–Gerasimenko, I noticed that some of the validation points ended up inside of the body, which was distorting the results. I modified the method to discard those points to remedy this problem. The other method involves sampling validation points at three different altitudes, for which I chose to use the standard values in the code provided by the authors of [11]. This results in 10000 points being sampled at 0.05, 0.1 and 0.25 length units above the body respectively.

I used the same number of integration points sampled in the unit cube for the forward modeling as in the training of the GeodesyNet, because the authors state in one of the provided notebooks that using a different number leads to worse results.

The authors use the Normalized L₁ Loss, as well as the Normalized Relative Component Loss in [11] to compare their results, which can be seen in one of the provided notebooks as well. I used the Normalized Relative Component Loss as my performance metric to compare validation results.

RESULTS AND DISCUSSION

In this chapter I will present how well the GeodesyNet and the comparable masconCUBE perform while learning a density and mass distribution for different kinds of heterogeneities, both with and without guidance. The approaches will be studied on the comet 67-P Churyumov-Gerasimenko and the asteroid Bennu. These results will be presented in section 5.1. In section 5.2, I will present and discuss the results of the empirical study I did to find and evaluate good guidance factors. In section 5.3 I will compare the two different guidance approaches for the masconCUBE. In section 5.4 I will evaluate how well the post training described in section 4.5 works. All of the results were generated while training for 10000 iterations for both the GeodesyNet and the masconCUBE, unless otherwise specified. For the GeodesyNet I used 30000 sampling points inside the unit cube for the forward modeling, and the gravitational field was evaluated at 1000 target points in every training iteration.

5.1 GEODESYNET VS. MASCONCUBE

In this section I will evaluate how well the GeodesyNet with guidance performs, both compared to a GeodesyNet without guidance and the masconCUBE method, with and without guidance. Firstly I will discuss how well the methods are able to fit the ground-truth gravitational field. After that I will compare the methods on how close they are able to get to the given guidance density in the guidance region, and thus the ground-truth in that region. For the GeodesyNets with guidance, I chose the guidance factor which produced the best performance regarding the gravitational field from 5.2. I chose the same guidance factor for the corresponding masconCUBE model with guidance, as I expect the tradeoff between gravitational field loss and guidance loss to be similar in the masconCUBE. For the version of Churyumov-Gerasimenko with three spheres of different densities, I did not perform a full guidance factor study. Here, I used the same guidance factor that performed the best in the same three sphere version, but with the densities set to zero. In the figures, the GeodesyNets are labeled “GN” and the masconCUBEs “mC”.

5.1.1 *Results on the gravitational field*

Starting with results on Churyumov-Gerasimenko with a single sphere in the middle, the errors are similar for all methods, when looking

at the low altitude interval on the left side in figure 5.1. At higher altitudes the difference between the GeodesyNets and the masconCUBEs widens, with the GeodesyNet exhibiting a two to three times greater error in comparison to the masconCUBE, depending on the altitude. While the difference between the masconCUBEs with and without guidance is small, the masconCUBE without guidance performs slightly better. For the GeodesyNets the errors are similar for the low altitude interval, whereas the GeodesyNet with guidance performs noticeably better at all other altitudes. To compare the results to [11], I ran the same validation procedure on a trained model for Churyumov-Gerasimenko provided by the authors. The version of Churyumov-Gerasimenko I created with the sphere in the middle differs the least from the version with a homogeneous density distribution the model by authors is trained on, and is therefore the most useful version to compare to. At the low altitude interval, the error of the unmodified GeodesyNet is close to my versions of the GeodesyNet. At the high altitude interval and altitude 0.05 the error increases about 30 to 40 percent compared to the unguided GeodesyNet. At the altitudes 0.1 and 0.25 the error of the unmodified GeodesyNet keeps increasing, to about three and six times the error of the unguided GeodesyNet respectively.

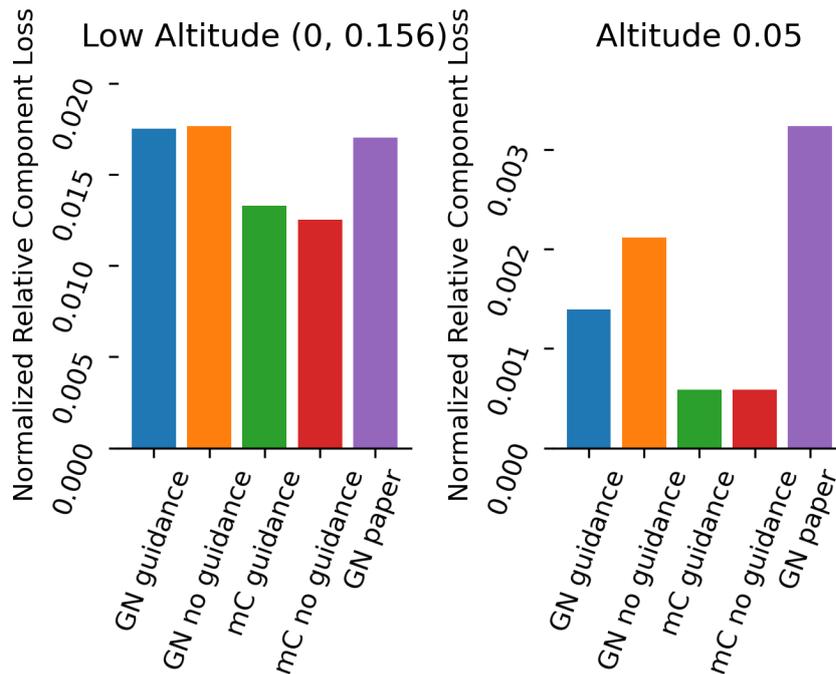


Figure 5.1: Excerpt from the results on Churyumov-Gerasimenko with a heterogeneity of one sphere in the middle of the body, with radius 0.05 and density 1. A guidance factor of 0.05 was used for the guidance models.

For Bennu with a single sphere close to the surface, the errors are similar again for the low altitude interval, depicted in figure 5.2. For all other altitudes, the masconCUBEs produce a much smaller error, with the unguided GeodesyNets error being between five to ten times greater, depending on the altitude. The error of the GeodesyNet with guidance is about double the error of the unguided GeodesyNet for all altitudes except the low altitude interval. For the masconCUBE, the version with guidance also produces higher errors, ranging between 30 and 130 percent depending on the altitude, with the exception of the low altitude interval as well. When comparing the unguided GeodesyNet to the model trained by the authors of [11], the results are similar at all altitudes, with my version of the GeodesyNet producing slightly smaller errors.

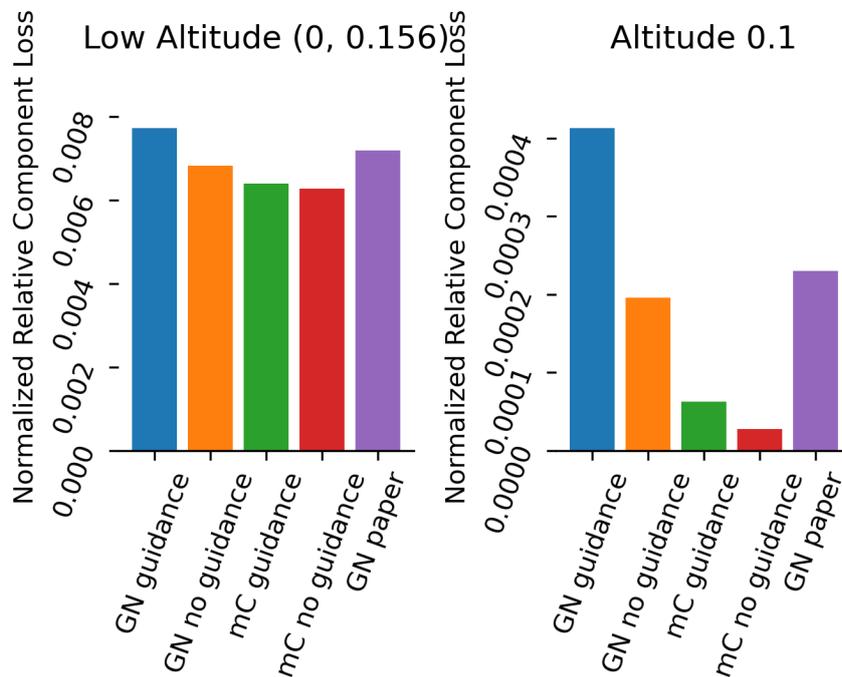


Figure 5.2: Excerpt from the results on Bennu with a heterogeneity of one sphere close to the surface of the body, with radius 0.05 and density 1.2. A guidance factor of 0.05 was used for the guidance models.

Comparing the results obtained for Bennu with the results for Churyumov-Gerasimenko, the errors are higher by about a factor of two for Churyumov-Gerasimenko when looking at the low altitude interval, and by an order of magnitude when looking at the high altitude interval, even more for the masconCUBE. At altitudes 0.05 and 0.1 the errors are more comparable, although they are still much smaller for Bennu. This might be explained by the fact that the underlying target point sampler for the intervals reproduces the shape of the body more closely than the one for the fixed altitudes. The more complex

shape of Churyumov-Gerasimenko might make these cases more difficult than the relatively round shape of Bennu. For the GeodesyNets, using guidance leads to a lower loss for Churyumov-Gerasimenko, while leading to a higher loss for Bennu. For the masconCUBE using guidance leads to higher errors for both bodies, although this effect is much more pronounced for Bennu. Comparing my results to the GeodesyNet by the authors of [11], they are close for Bennu, while the error is overall much higher for the unmodified GeodesyNet on Churyumov-Gerasimenko. I would have expected the results to be closely comparable, since my modification to the ground-truth is small in this case. The bigger difference might be that my ground-truth for Churyumov-Gerasimenko has a much higher resolution of about one million mascon points compared to the homogeneous version with 57000 points, whereas Bennu has about 175000 mascon points compared to about 37000 for the homogeneous version.

The results for Churyumov-Gerasimenko with three spherical heterogeneities with densities of zero, and the version with three spherical heterogeneities at the same positions with densities of 1, 0.5 and 2.5 are very similar. The difference in error between the GeodesyNets and the masconCUBEs are less pronounced than in the version with one sphere, with the error only slightly higher at the low altitude interval, and between two and 3.5 times higher at the high altitude interval and at the fixed altitude 0.1. The largest differences can be found at the altitudes 0.1 and 0.25, which are depicted in figure 5.3 for the version with the densities set to zero and figure 5.4 for the version with varying densities. At altitude 0.1 the errors for the GeodesyNets are very similar, while the errors for the masconCUBEs are almost twice as high for the version with the densities set to zero. At altitude 0.25 the GeodesyNets have a 20 to 30 percent higher error compared to the masconCUBEs, while they have a two to three times higher error in the version with varying densities. The overall magnitude of errors is similar to the one displayed for the version with one sphere. For both versions with three spherical heterogeneities errors between the model with and without guidance are similar.

For the version of Bennu with three spherical heterogeneities with densities of 1, 0.5 and 1.5, the masconCUBE produces an order of magnitude smaller errors at all altitudes, with the exception of the low altitude interval, shown on the left side in figure 5.5. Here the results are much closer, the GeodesyNets produce errors that are about 20 percent higher than the masconCUBEs. For the GeodesyNets, the version with guidance produces slightly smaller errors at all altitudes than the version without. The opposite is true for the masconCUBE, here the version with guidance produces 20 to 30 percent higher errors than the version without at all altitudes, with the exception of the low altitude interval.

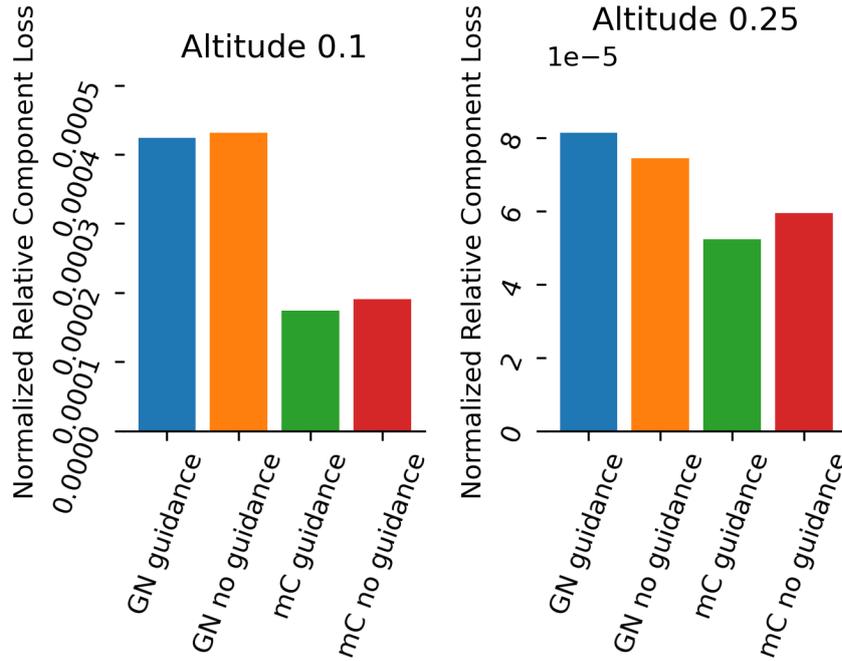


Figure 5.3: Excerpt from the results for Churyumov-Gerasimenko with three spherical heterogeneities distributed through the body. The spheres have radii of 0.05, 0.035 and 0.05 and densities of zero. A guidance factor of 0.0125 was used for the guidance models.

When comparing the results of the three sphere heterogeneity version of Bennu with the ones of Churyumov-Gerasimenko, the errors are again higher by a factor of two for Churyumov-Gerasimenko at the low altitude interval and by more than an order of magnitude for the high altitude interval. Similarly to the version with one sphere, the errors get more comparable for the other altitudes as well. For Churyumov-Gerasimenko the difference in errors between the masconCUBE and the GeodesyNets, while still being big, is much less pronounced than in Bennus case.

The GeodesyNet without guidance produces a loss about twice as high as the one for masconCUBE without guidance, for Churyumov-Gerasimenko with the heterogeneity defined through a plane, at the low altitude interval, depicted on the left in figure 5.6. The difference keeps increasing at the higher altitudes, with the error about an order of magnitude higher for the altitudes 0.1 and 0.25. The loss for the version of the GeodesyNet with guidance is about a third higher than for the unguided GeodesyNet, with the exception of altitude 0.25, where it is slightly lower. The guided version of the masconCUBE has an error about four times the error of the unguided version for the low altitude interval, making its error also about double the error of the guided GeodesyNet. For the other altitudes, the error of the guided

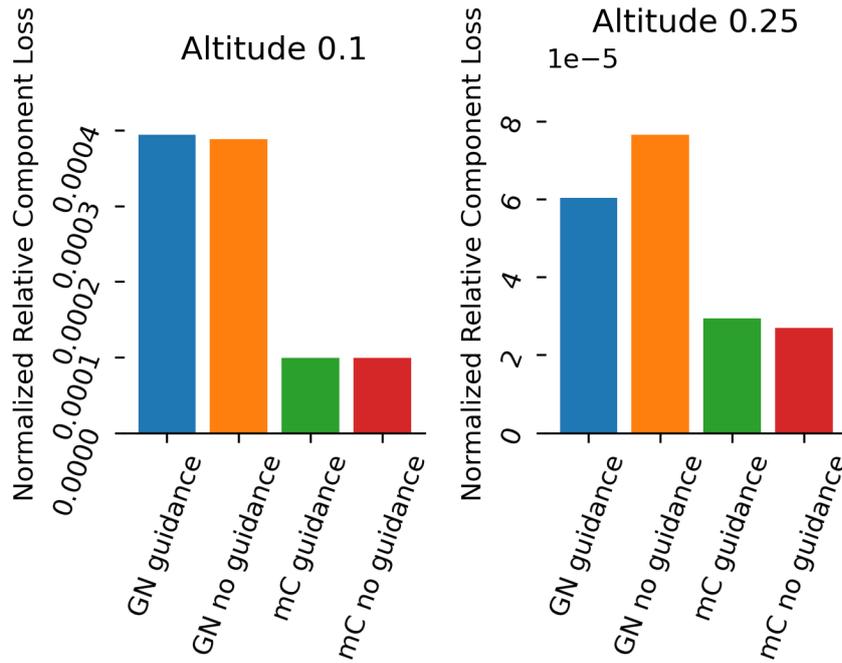


Figure 5.4: Excerpt from the results for Churyumov-Gerasimenko with three spherical heterogeneities distributed through the body. The spheres have radii of 0.05, 0.035 and 0.05 and densities of 1, 0.5 and 2.5. A guidance factor of 0.0125 was used for the guidance models.

version is between two and nine times the error of the unguided version. Compared to the other versions of Churyumov-Gerasimenko, the errors are in the same order of magnitude, although the errors for the versions with guidance are comparatively higher.

The results for Bennu with the heterogeneity defined through a plane are similar for the high altitude interval and the fixed altitudes, and generally match the graph on the right side of figure 5.7. The unguided masconCUBE exhibits the smallest error, with the masconCUBE with guidance having about four to five times greater errors. The unguided GeodesyNet has an about two to three times greater error again, with the guided versions errors being about 20 to 30 percent higher than the unguided versions. The exception to this is the altitude 0.25, where the error for the guided version is slightly lower than for the unguided version. For the low altitude interval, the masconCUBE with guidance exhibits by far the largest error, similarly to the plane version of Churyumov-Gerasimenko, although the effect is far more pronounced here. The error of the unguided version of the masconCUBE is close to the GeodesyNet, with the error of the guided GeodesyNet being about fifty percent higher.

Comparing the results for the plane versions of Churyumov-Gerasimenko and Bennu, the errors are again much smaller for Bennu the altitude

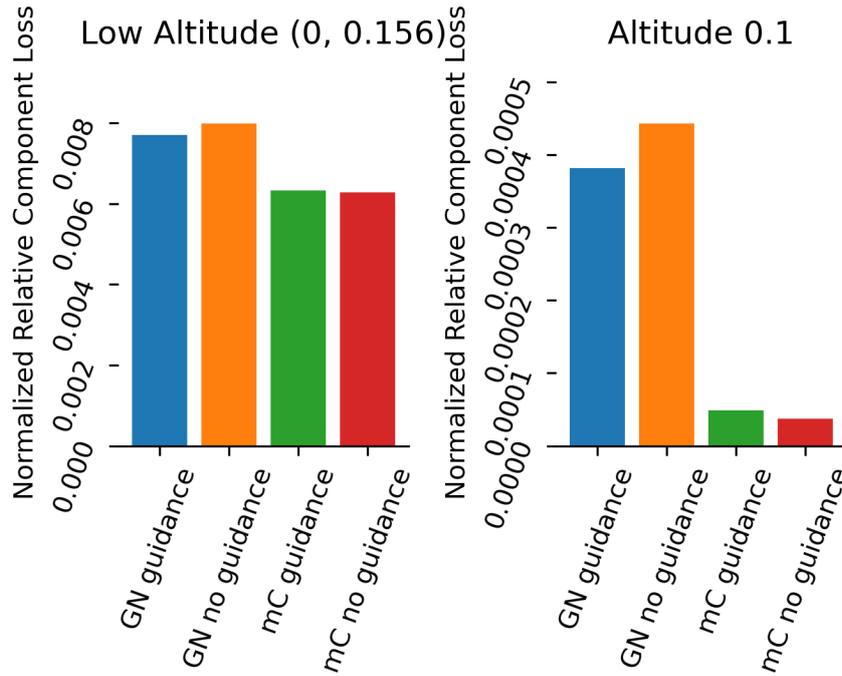


Figure 5.5: Excerpt from the results for Bennu with three spherical heterogeneities distributed through the body. The spheres have radii of 0.05, 0.035 and 0.05 and densities of 1, 0.5 and 1.5. A guidance factor of 0.05 was used for the guidance models.

intervals and more comparable at the fixed intervals. The most striking difference between the results of the two bodies is the error of the guided masconCUBE, which is lower than the GeodesyNets for all altitudes except the low altitude interval for Bennu. This is not the case for Churyumov-Gerasimenko, where the guided masconCUBE is close to the unguided GeodesyNet at altitude 0.1 and higher the altitude 0.05 and the high altitude interval.

When looking at these results, it becomes clear that the masconCUBE is able to match the ground-truth gravitational field far better than the GeodesyNet for the heterogeneities defined through single and multiple spheres, on both Churyumov-Gerasimenko and Bennu. This is true for both the guided and unguided version. On the versions of the bodies with the heterogeneities defined through a plane, the unguided masconCUBE is still the best method by far, but the errors of the masconCUBE with guidance are closer to the GeodesyNets. Here, the GeodesyNets with guidance outperform the guided masconCUBE at the low altitude intervals for both bodies.

Comparing my versions to the model trained by the authors of [11] revealed similar results for Bennu. For Churyumov-Gerasimenko the results were comparable for the interval altitudes and altitude 0.05, but the unmodified GeodesyNet exhibited a much greater error for

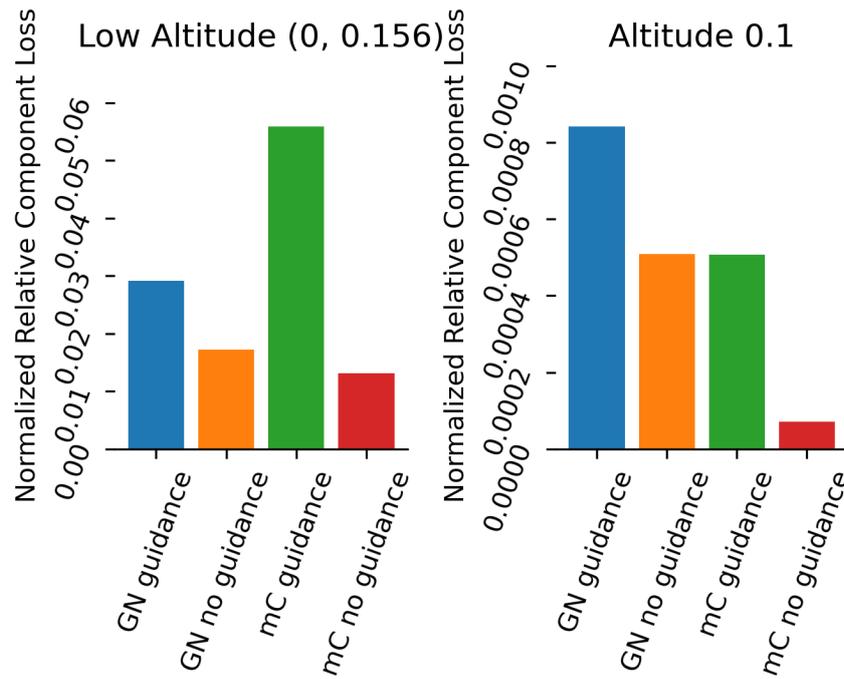


Figure 5.6: Excerpt from the results for Churyumov-Gerasimenko with the heterogeneity defined through a plane. The density is set to 1.83 on the positive side of the plane. A guidance factor of 0.0125 was used for the guidance models.

the altitudes 0.1 and 0.25. I was not able to replicate the finding of the authors that the GeodesyNet and masconCUBE reach comparable errors, in my results the unguided masconCUBE outperformed the unguided GeodesyNet in every case.

5.1.2 Comparing the guidance losses

In this section I will be evaluating how well both the guided masconCUBE and the guided GeodesyNet are able to match a given guidance density. For the GeodesyNets, I calculated the guidance loss in the guidance regions for the best model of a given training run, for the models discussed in section 5.1.1. I did the same for the masconCUBEs. For the models with multiple guidance regions, I calculated the mean of the guidance losses. Table 5.1 shows the results. For the single sphere and the multiple sphere versions of Churyumov-Gerasimenko, as well as the for the multiple sphere version of Bennu the masconCUBE has the smaller loss by a margin. Note that for the result of the masconCUBE for the multiple sphere version of Churyumov-Gerasimenko with the densities set zero, the masconCUBE sets the corresponding masses in the guidance regions to zero, leading to a density of zero, which makes the guidance loss also zero, which is the desired result. For the version of Bennu with one sphere, both methods perform

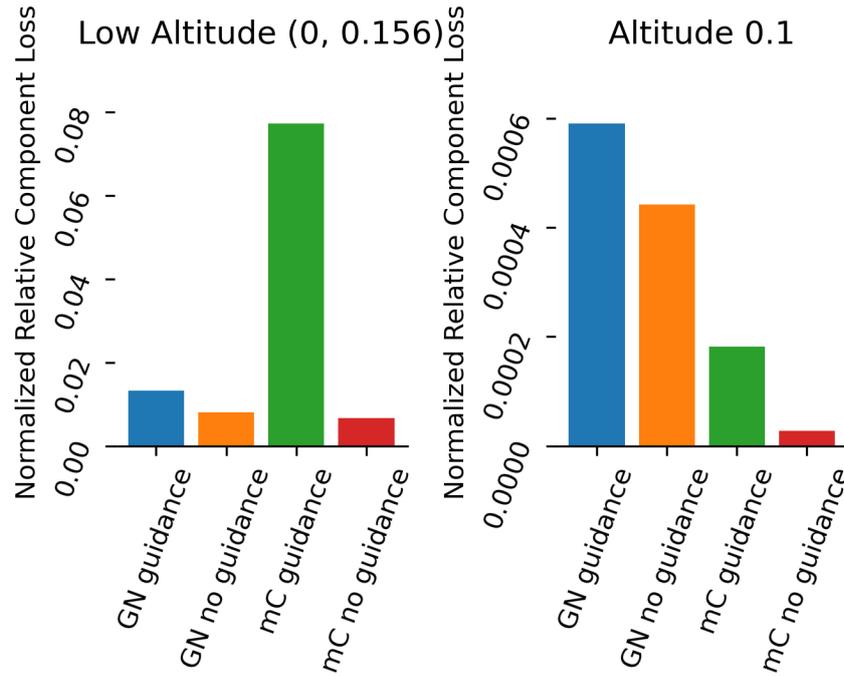


Figure 5.7: Excerpt from the results for Bennu with the heterogeneity defined through a plane. The density is set to 0.99 on the positive side of the plane. A guidance factor of 0.0125 was used for the guidance models.

similarly well. The GeodesyNet produces the smaller guidance loss for the plane versions of both bodies by a margin. Summing up, both methods are able to get close to the specified guidance density. In light of the results, a masconCUBE seems to be the better choice with regard to the guidance loss when small guidance regions are needed, while the GeodesyNet performs better on larger guidance regions. The unguided versions of both methods produce guidance losses between 0.16 and 2.84, meaning they do not fit the ground truth well in the guidance regions. This means guidance is needed if a given model is meant to match a certain density in a certain region.

5.1.3 Conclusion

In conclusion, the results in this section have shown that the masconCUBE is the better choice of model, if no guidance is needed, as it performs better than the unguided GeodesyNet in all studied configurations, relative to the ground-truth gravitational field. Another benefit of the masconCUBE is the faster training time. When guidance is needed, the masconCUBE is still the better choice in most situations, unless a situation requires the specific strengths of the GeodesyNet. It performs comparatively well on target points very close to the bodies

Ground-truth name	GeodesyNet	masconCUBE
CG sphere	4.448e-04	4.341e-05
CG multi-zero	2.320e-05	0.000e+00
CG multi	1.751e-04	7.840e-05
CG plane	7.732e-03	1.203e-02
Bennu sphere	1.251e-04	1.146e-04
Bennu multi	1.229e-04	2.864e-05
Bennu plane	7.470e-03	2.490e-02

Table 5.1: Final guidance losses for the studied ground-truths, for GeodesyNets and masconCUBEs.

surface and beat the masconCUBE at the low interval altitude, when the guidance region was specified through a plane. This is also the situation in which the GeodesyNet outperforms the masconCUBE with regard to the guidance density. It should also be noted that the GeodesyNet provides a continuous function of the density inside the body. This means that if a highly detailed interior model of a body is needed and the resolution of the studied masconCUBE is not sufficient, the GeodesyNet should be chosen for the task.

5.2 FINDING A GOOD GUIDANCE FACTOR

I did an empirical study to find out which guidance factor works best, and if it differs from body to body and between different kinds of guidance regions. I also wanted to evaluate how well the automatic scaling of the guidance factor works in comparison to a guidance factor found through empirical means. I studied different guidance values and heterogeneities defined through one and multiple spheres, as well as through a plane for both Churyumov-Gerasimenko and Bennu. I will be displaying excerpts from the results in this section. Figure 5.8 shows that the validation error massively increases above a guidance factor of 0.1. Because of this I limited my search to factors between zero and 0.1.

Comparing the results of the automatic guidance loss scaling with the results of the defined guidance factors, the error is much higher for the automatic scaling in almost every case. Examples for this can be seen in figures 5.9 and 5.11. In these instances the error for the automatic scaling is still in the same order of magnitude as the one for the guidance factors. This is not the case for most of the other cases I studied, where it is much higher. For this reason, I left the automatic scaling out of the remaining figures, as it would distort the other results too much.

When looking at the results for Churyumov-Gerasimenko with the heterogeneity defined through a sphere, there is no one guidance factor, that performs best across all altitudes. A factor of 0.05 works best in most of the cases, but not all, like on the left side in figure 5.9. The differences between the different factors are not very pronounced across all altitudes, especially between the overall two best factors, which are 0.05 and 0.0125. The results for Churyumov-Gerasimenko with three spherical heterogeneities do not differ strongly as well. The exception is the result for the low altitude interval, where the factors 0.075, 0.05 and 0.0125 perform noticeably better, as pictured on the left side in figure 5.10. For Bennu with three spherical heterogeneities, a factor of 0.05 performs better than the others on for all altitudes, with the exception the altitude 0.25, where a factor of 0.025 performs equally well. An excerpt from these results is depicted in figure 5.11. For Bennu with a heterogeneity defined through a sphere close to the surface, all guidance factors produce similar errors for the altitude 0.05, 0.1 and 0.25, with the factor 0.05 performing slightly better than the others, depicted in figure 5.12. For the low and high altitude intervals, a guidance factor of 0.025 performs better than the others by a margin. For the heterogeneity defined through a plane on Churyumov-Gerasimenko, the smallest guidance factors 0.025 and 0.0125 perform best across all altitudes, both leading to similar error figures, illustrated in figure 5.13. The same is true for Bennu with the heterogeneity defined through a plane, depicted in figure 5.14.

To summarize, there is no guidance factor that performs best across all studied bodies and altitudes. The guidance factors 0.05 and 0.025 are generally amongst the best performers in most situations, which is why they should be used in a new situation. The automatic guidance loss scaling is not competitive in performance, and should not be used over the defined guidance factors.

5.3 COMPARING GUIDANCE APPROACHES FOR THE MASCONCUBE

The validation results depicted in figure 5.15 show that the loss based version of implementing guidance for the masconCUBE works much better than setting the masses in question to the desired mass directly. Especially for the result close to the surface in the interval $(0, 0.156)$, setting the masses directly produces unusable results. At the other altitudes the guidance loss based version is between one and two orders of magnitude better. Because of this, I chose the loss based version as a comparative method for this thesis.

5.4 POST TRAINING

In this section I will discuss how well the post training method described in 4.5 works. I tested this method on Churyumov-Gerasimenko

with a heterogeneity of one sphere and on Bennu, with the heterogeneity defined through a plane. Figure 5.16 shows the results for Bennu, with each model depicted with the number of iterations it was trained for, and the normally trained guidance model on the same ground-truth on the right. The results show that for the fixed altitudes, as well as for the high altitude interval, the post training model reaches the loss of the normally trained GeodesyNet and improves from thereon. For the low altitude interval, the post training has lower loss after the first 1000 iterations.

For Churyumov-Gerasimenko the loss of the post-trained models gradually declines towards the loss of the normally trained guidance model, but never reaches it, for the interval altitudes and altitude 0.05. For altitude 0.1 the loss of the normally trained model gets reached after 10000 iterations of post-training, and for the altitude 0.25 this is the case after 7000 iterations.

These results suggest that this method of post-training is a worthwhile avenue for further research, to be able to get good results with guided GeodesyNets after fewer iterations of training, especially for larger guidance regions. It might be worthwhile to explore loss weighting that is specifically designed for this kind of post-training, to be able to preserve the optimization to the given gravitational field from the pre-trained model, while optimizing the guidance loss quickly.

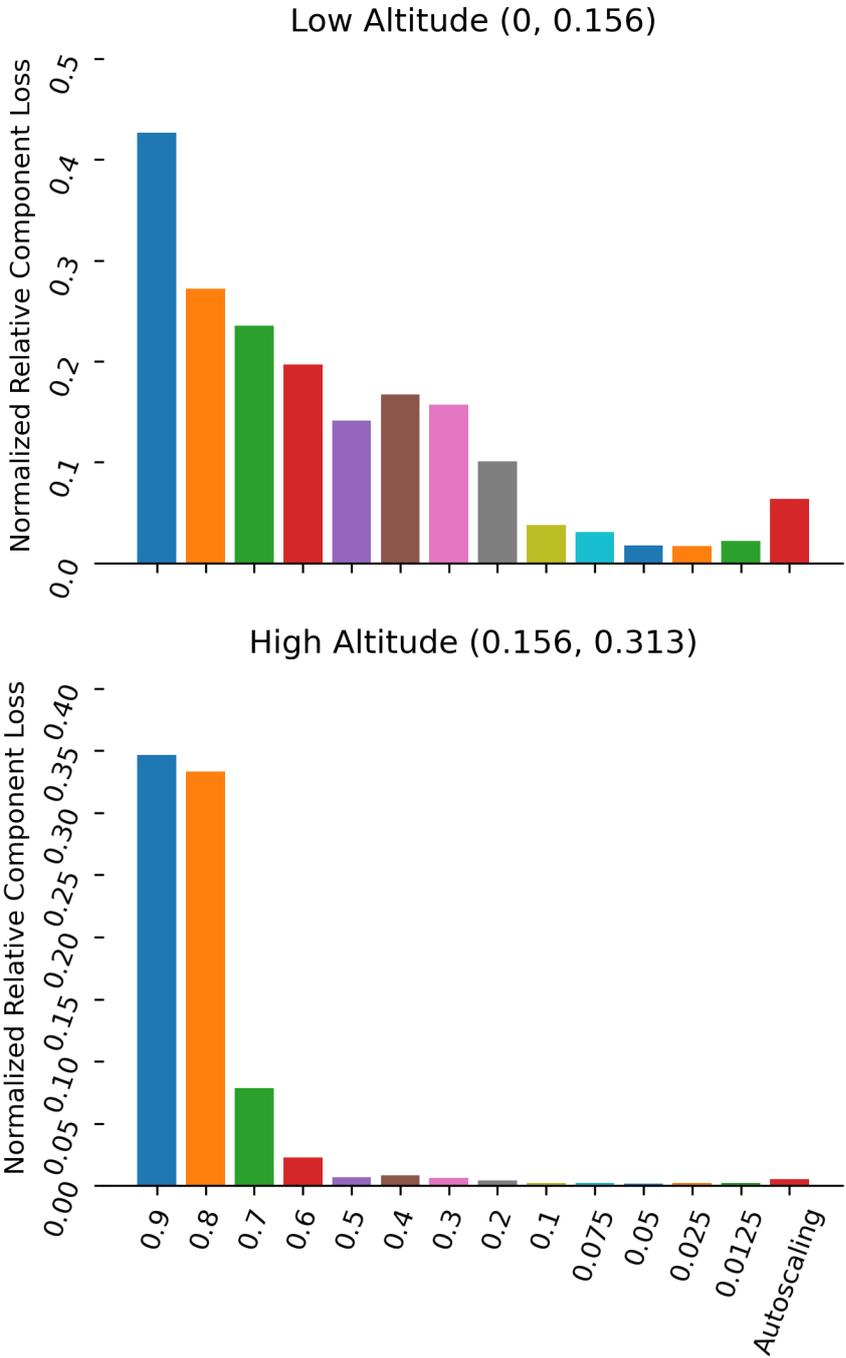


Figure 5.8: Full guidance factor study on Churyumov-Gerasimenko with a heterogeneity of one sphere.

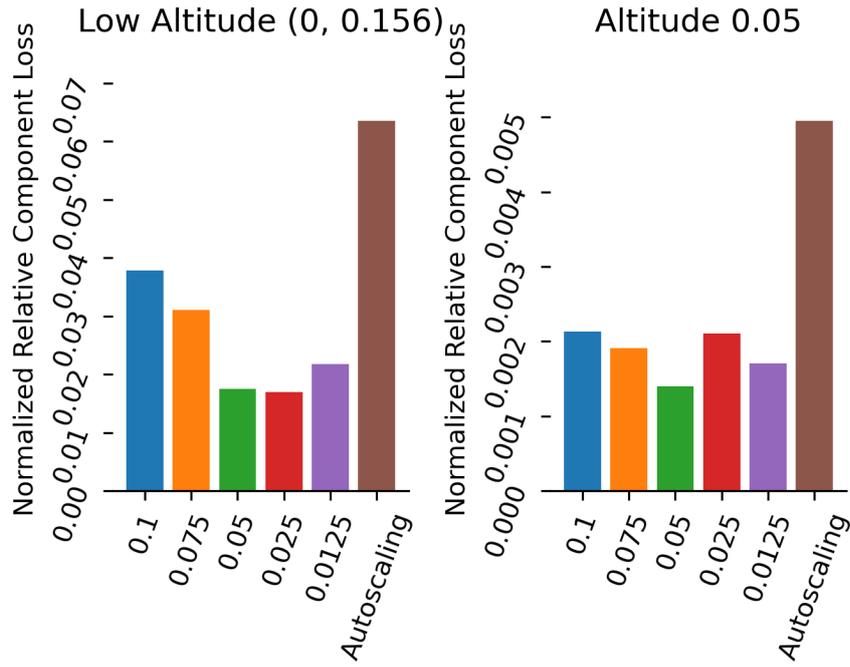


Figure 5.9: Excerpt from the guidance factor study on Churyumov-Gerasimenko with a heterogeneity of one sphere in the middle of the asteroid with radius 0.05 and density 1.

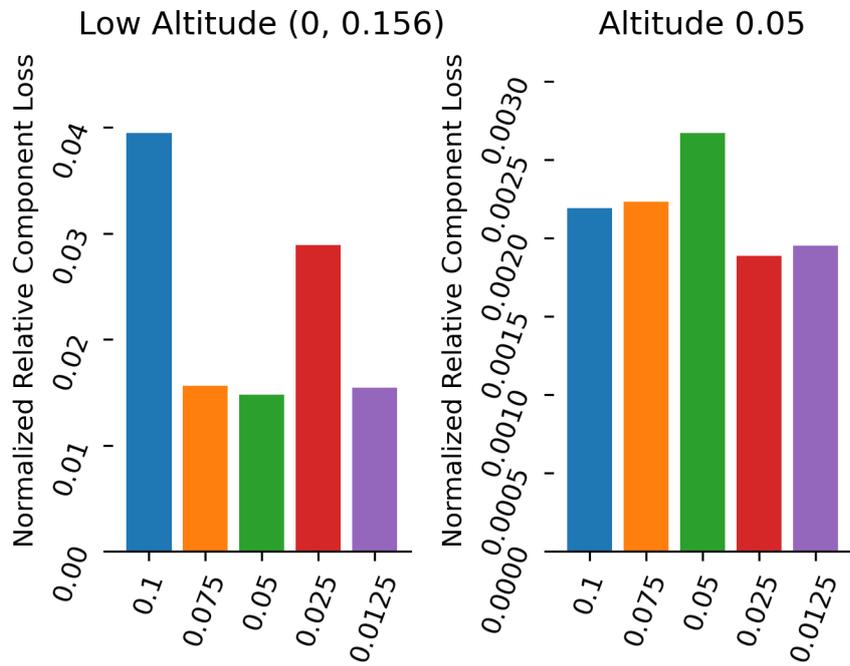


Figure 5.10: Excerpt from the guidance factor study on Churyumov-Gerasimenko with three spherical heterogeneities with radii 0.05, 0.035 and 0.05 and densities of zero.

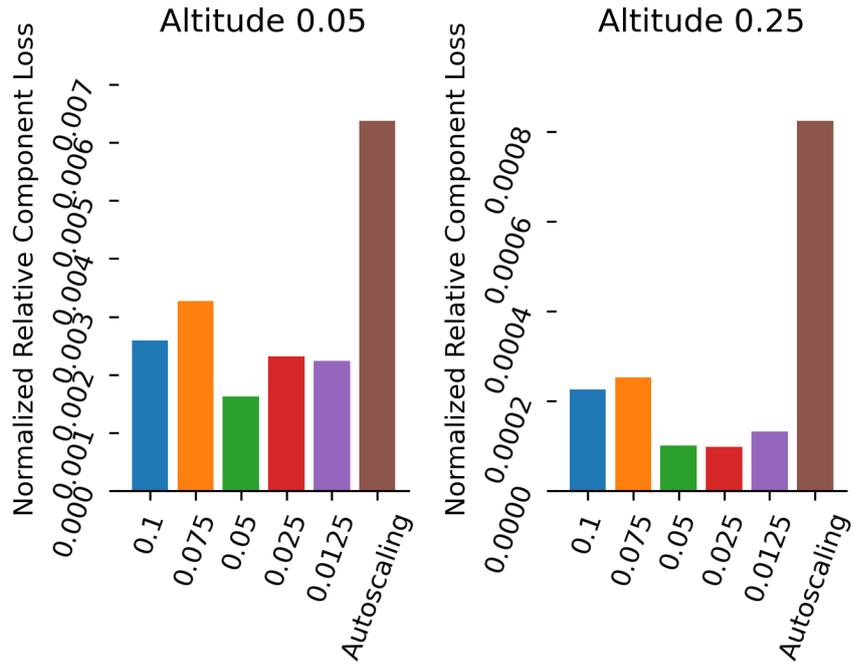


Figure 5.11: Excerpt from the guidance factor study on Bennu with three spherical heterogeneities with radii 0.05, 0.035 and 0.05 and densities of 1, 0.5 and 1.5.

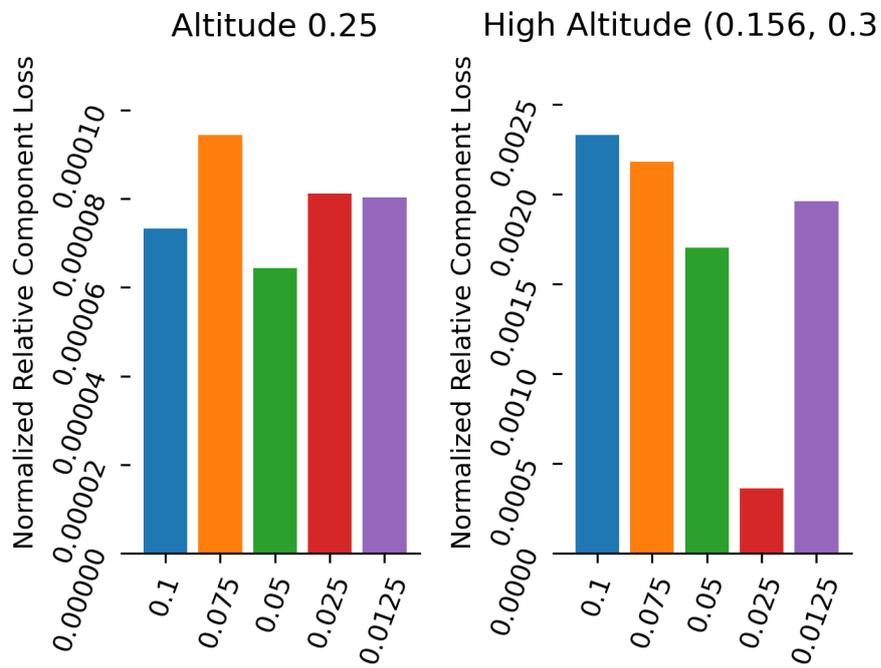


Figure 5.12: Excerpt from the guidance factor study on Bennu with a heterogeneity of one sphere close to the surface of the asteroid with radius 0.05 and density 1.2.

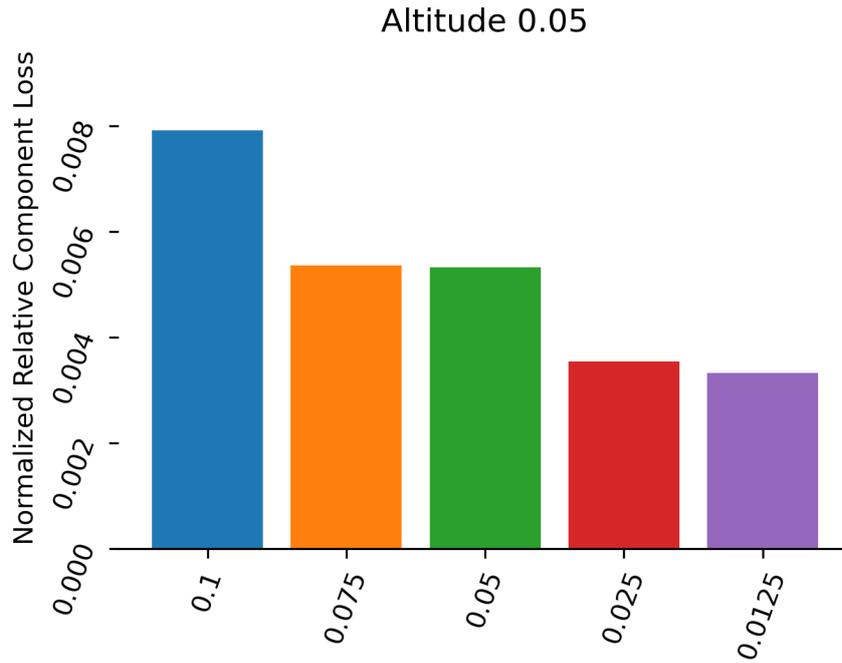


Figure 5.13: Excerpt from the guidance factor study on Churyumov-Gerasimenko with the heterogeneity defined through a plane. The density is set to 1.83 on the positive side of the plane.

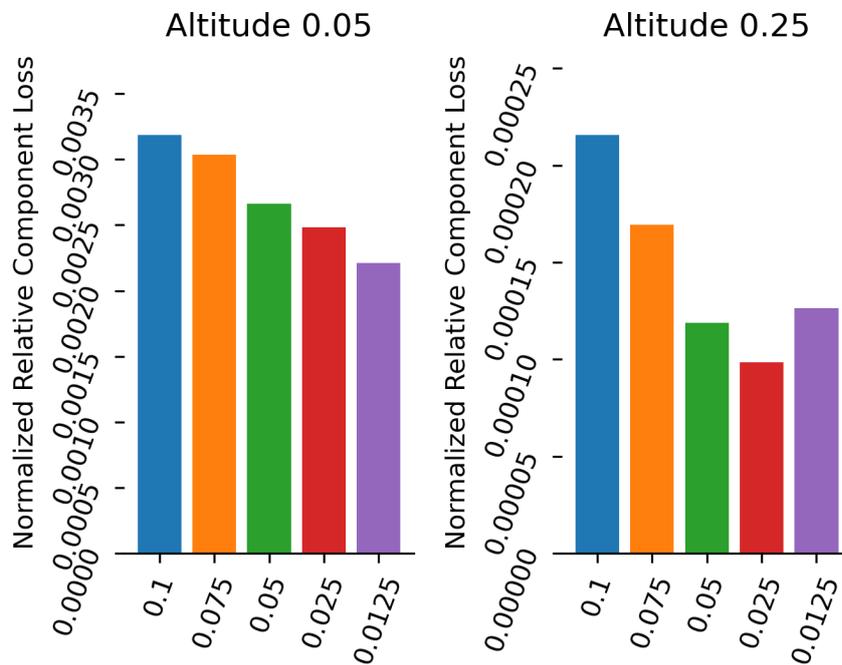


Figure 5.14: Excerpt from the guidance factor study on Bennu with the heterogeneity defined through a plane. The density is set to 0.99 on the positive side of the plane.

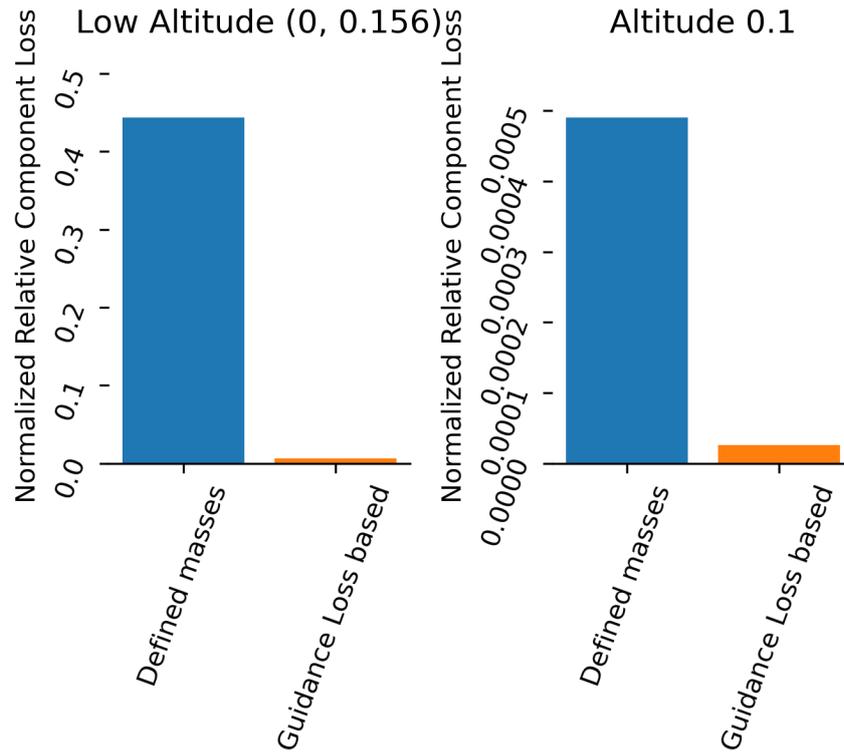


Figure 5.15: Excerpt from the validation results on Bennu for the mascon-CUBE model, with the heterogeneity defined through a plane, with the density set to 0.99 on the positive side of the plane. The figure shows both versions of guidance I implemented, with the masses defined in every iteration on the left and the guidance loss based version on the right.

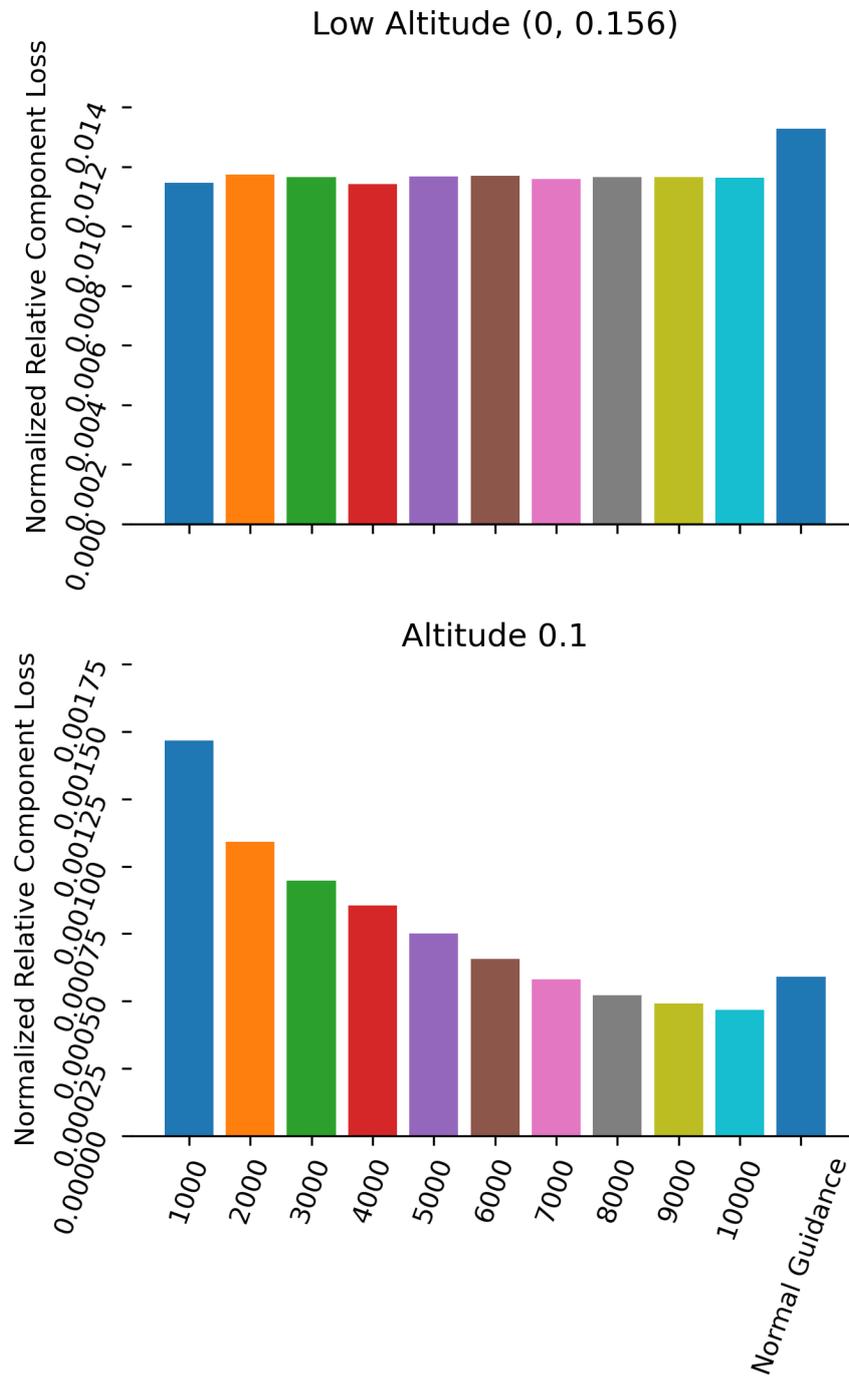


Figure 5.16: Results of the post training method on Bennu with the heterogeneity defined through a plane. A guidance loss of 0.0125 was used.

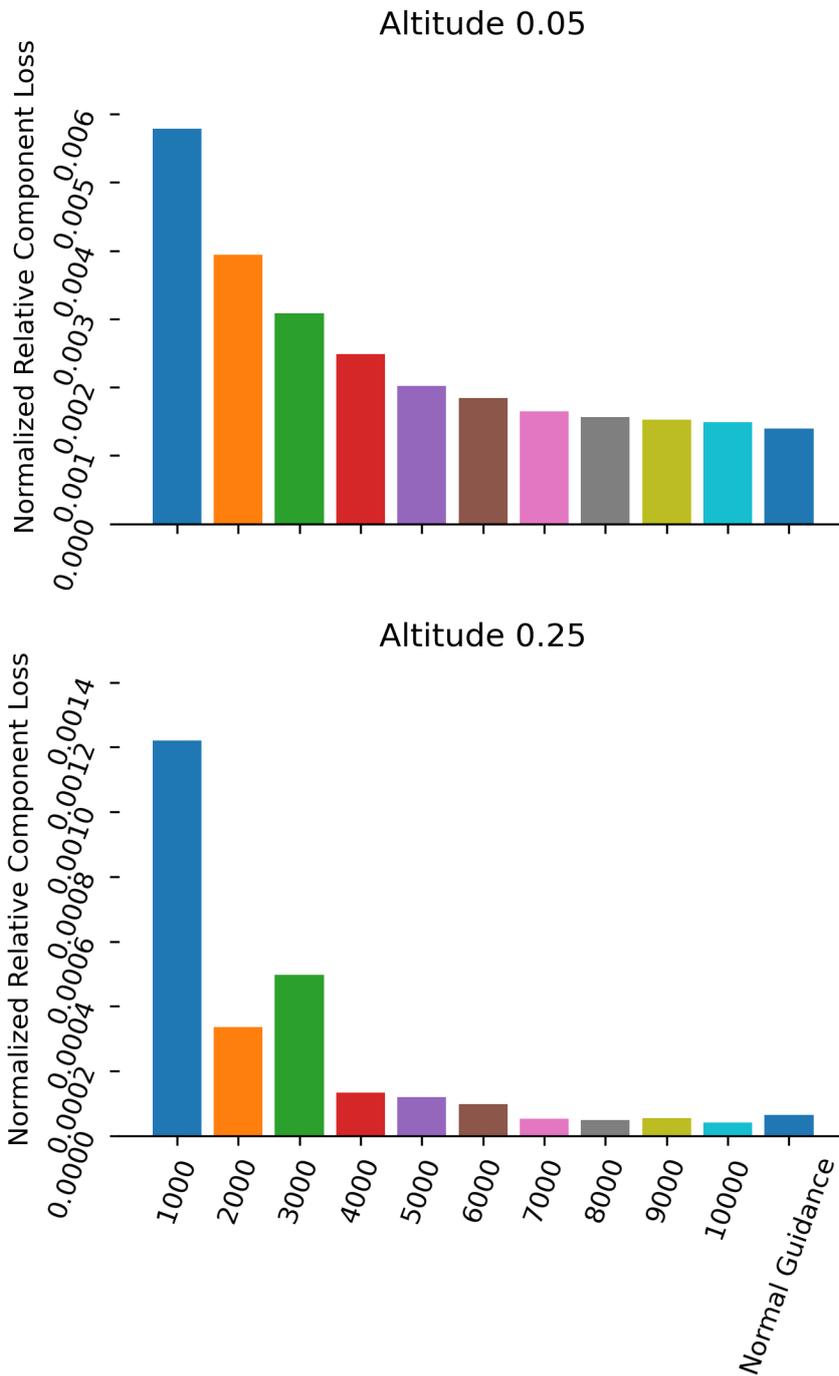


Figure 5.17: Results of the post training method on Churyumov-Gerasimenko with the heterogeneity defined through a sphere in the middle of the body. A guidance loss of 0.05 was used.

FUTURE WORK

In this chapter I will present ideas for possible future research, based on the results of this thesis.

Firstly, a possible avenue for future research would be to utilize the dataset I built in chapter 3 to train a machine learning model similar to the ones I presented in section 2.3.1. It would be interesting to see how such a model would compare performance-wise to the GeodesyNets presented here, and if it would be able to generalize to unseen small bodies. If it could, this would improve upon the long training times for the GeodesyNet and masconCUBE. As an addition to this, implementing guidance for this kind of model would be interesting as well.

Building on the results of my work on the GeodesyNets, a next step would be to utilize more sophisticated techniques from multi-task learning to weight the guidance loss and the gravitational field based loss. These might be able to incorporate the guidance loss into the training, while preserving the gravitational field based loss better than the guidance factors in this thesis. They could also remove the necessity to choose a good guidance factor. Exploring the approach from section 4.5 further would also be worthwhile, and it might benefit from the utilization of techniques from multi-task learning as well.

CONCLUSION

In conclusion, I built a dataset in the process of this thesis, which can be used to experiment with and train future machine learning models for inverse gravity modeling. In order to incorporate additional information into the gravity inversion process, I modified the GeodesyNets introduced in [11], and implemented a guidance loss, so that they are able to optimize for a specific density in a specified region inside of the body, in addition to the given gravitational field. I implemented the same loss for a more classical mascon based method and compared the results. I found that the mascon based method performed better than the GeodesyNet in most of the cases I studied, both regarding the gravitational field and how close the methods were able to get to the specified guidance density. Additionally, I identified some scenarios in which the GeodesyNet would be the better choice. To determine good guidance factors to scale the guidance loss for the different ground-truths, I performed an empirical search, and also experimented with automatically scaling the guidance loss. To mitigate the long training times of the GeodesyNet I tested post-training on an already trained model without guidance, which showed promising results in some situations.

BIBLIOGRAPHY

- [1] Alfonso Caldiero and Sébastien Le Maistre. “Small bodies global gravity inversion via the level-set method.” In: *Icarus* 411 (Mar. 2024), p. 115940. ISSN: 0019-1035. DOI: [10.1016/j.icarus.2023.115940](https://doi.org/10.1016/j.icarus.2023.115940). URL: <http://dx.doi.org/10.1016/j.icarus.2023.115940>.
- [2] T. G. G. Chanut, S. Aljbaae, and V. Carruba. “Mascon gravitation model using a shaped polyhedral source.” In: *Monthly Notices of the Royal Astronomical Society* 450.4 (May 2015), pp. 3742–3749. ISSN: 0035-8711. DOI: [10.1093/mnras/stv845](https://doi.org/10.1093/mnras/stv845). URL: <http://dx.doi.org/10.1093/mnras/stv845>.
- [3] Özgün Çiçek, Ahmed Abdulkadir, Soeren S. Lienkamp, Thomas Brox, and Olaf Ronneberger. *3D U-Net: Learning Dense Volumetric Segmentation from Sparse Annotation*. 2016. arXiv: [1606.06650](https://arxiv.org/abs/1606.06650) [cs.CV].
- [4] A. Colagrossi, F. Ferrari, M. Lavagna, and K. Howell. “Dynamical evolution about asteroids with high fidelity gravity field and perturbations modeling.” In: *Advances in the Astronautical Sciences* 156 (2016), pp. 885–903.
- [5] Wang Yu-Feng, Zhang Yu-Jie, Fu Li-Hua, and Li Hong-Wei. “Three-dimensional gravity inversion based on 3D U-Net++.” In: *Applied Geophysics* 18.4 (Dec. 2021), pp. 451–460. ISSN: 1993-0658. DOI: [10.1007/s11770-021-0909-z](https://doi.org/10.1007/s11770-021-0909-z). URL: <http://dx.doi.org/10.1007/s11770-021-0909-z>.
- [6] Greg Frieger. *3D Asteroid Catalogue*. July 3, 2021. URL: <https://3d-asteroids.space/> (visited on 08/17/2024).
- [7] A. Fujiwara et al. “The Rubble-Pile Asteroid Itokawa as Observed by Hayabusa.” In: *Science* 312.5778 (June 2006), pp. 1330–1334. ISSN: 1095-9203. DOI: [10.1126/science.1125841](https://doi.org/10.1126/science.1125841). URL: <http://dx.doi.org/10.1126/science.1125841>.
- [8] Jérémie Giraud, Mark Lindsay, and Mark Jessell. “Generalization of level-set inversion to an arbitrary number of geologic units in a regularized least-squares framework.” In: *GEOPHYSICS* 86.4 (July 2021), R623–R637. ISSN: 1942-2156. DOI: [10.1190/geo2020-0263.1](https://doi.org/10.1190/geo2020-0263.1). URL: <http://dx.doi.org/10.1190/geo2020-0263.1>.
- [9] Charles R. Harris et al. “Array programming with NumPy.” In: *Nature* 585.7825 (Sept. 2020), pp. 357–362. DOI: [10.1038/s41586-020-2649-2](https://doi.org/10.1038/s41586-020-2649-2). URL: <https://doi.org/10.1038/s41586-020-2649-2>.
- [10] Weikko Heiskanen and Helmut Moritz. *Physical Geodesy*. 1967.

- [11] Dario Izzo and Pablo Gómez. “Geodesy of irregular small bodies via neural density fields.” In: *Communications Engineering* 1.1 (Dec. 2022). ISSN: 2731-3395. DOI: [10.1038/s44172-022-00050-3](https://doi.org/10.1038/s44172-022-00050-3). URL: <http://dx.doi.org/10.1038/s44172-022-00050-3>.
- [12] A.S. Konopliv et al. “The Ceres gravity field, spin pole, rotation period and orbit from the Dawn radiometric tracking and optical data.” In: *Icarus* 299 (Jan. 2018), pp. 411–429. ISSN: 0019-1035. DOI: [10.1016/j.icarus.2017.08.005](https://doi.org/10.1016/j.icarus.2017.08.005). URL: <http://dx.doi.org/10.1016/j.icarus.2017.08.005>.
- [13] Siu Kwan Lam, Antoine Pitrou, and Stanley Seibert. “Numba: a LLVM-based Python JIT compiler.” In: *Proceedings of the Second Workshop on the LLVM Compiler Infrastructure in HPC*. SC15. ACM, Nov. 2015. DOI: [10.1145/2833157.2833162](https://doi.org/10.1145/2833157.2833162). URL: <http://dx.doi.org/10.1145/2833157.2833162>.
- [14] D. S. Lauretta et al. “OSIRIS-REx: Sample Return from Asteroid (101955) Bennu.” In: *Space Science Reviews* 212.1–2 (Aug. 2017), pp. 925–984. ISSN: 1572-9672. DOI: [10.1007/s11214-017-0405-1](https://doi.org/10.1007/s11214-017-0405-1). URL: <http://dx.doi.org/10.1007/s11214-017-0405-1>.
- [15] Hou-Pu Li, Rui Qi, Jia-Xin Hu, and Yu-Xin Sun. “3D gravity anomaly inversion based on LinkNet.” In: *Applied Geophysics* 20.1 (Mar. 2023), pp. 36–50. ISSN: 1993-0658. DOI: [10.1007/s11770-023-1020-4](https://doi.org/10.1007/s11770-023-1020-4). URL: <http://dx.doi.org/10.1007/s11770-023-1020-4>.
- [16] Hermann Meißenhelter, René Weller, Matthias Noeker, Tom Andert, and Gabriel Zachmann. “Adaptive Polydisperse Sphere Packings for High Accuracy Computations of the Gravitational Field.” In: *2023 IEEE Aerospace Conference*. IEEE, Mar. 2023. DOI: [10.1109/aero55745.2023.10115938](https://doi.org/10.1109/aero55745.2023.10115938). URL: <http://dx.doi.org/10.1109/AER055745.2023.10115938>.
- [17] V Michel and A S Fokas. “A unified approach to various techniques for the non-uniqueness of the inverse gravimetric problem and wavelet-based methods.” In: *Inverse Problems* 24.4 (July 2008), p. 045019. ISSN: 1361-6420. DOI: [10.1088/0266-5611/24/4/045019](https://doi.org/10.1088/0266-5611/24/4/045019). URL: <http://dx.doi.org/10.1088/0266-5611/24/4/045019>.
- [18] J.K. Miller et al. “Determination of Shape, Gravity, and Rotational State of Asteroid 433 Eros.” In: *Icarus* 155.1 (Jan. 2002), pp. 3–17. ISSN: 0019-1035. DOI: [10.1006/icar.2001.6753](https://doi.org/10.1006/icar.2001.6753). URL: <http://dx.doi.org/10.1006/icar.2001.6753>.
- [19] Marco Musy et al. *marcomusy/vedo: 2023.4.6*. Version v2023.4.6. June 2023. DOI: [10.5281/zenodo.8067437](https://doi.org/10.5281/zenodo.8067437). URL: <https://doi.org/10.5281/zenodo.8067437>.

- [20] Ryan S. Park, Robert A. Werner, and Shyam Bhaskaran. “Estimating Small-Body Gravity Field from Shape Model and Navigation Data.” In: *Journal of Guidance, Control, and Dynamics* 33.1 (2010), pp. 212–221. DOI: [10.2514/1.41585](https://doi.org/10.2514/1.41585). eprint: <https://doi.org/10.2514/1.41585>. URL: <https://doi.org/10.2514/1.41585>.
- [21] Jason Pearl and Darren Hitt. “Comparing the Computational Efficiency of Polyhedral and Mascon Gravity Models.” In: Feb. 2017.
- [22] Jason Pearl and Darren Hitt. “Mascon distribution techniques for asteroids and comets.” In: *Celestial Mechanics and Dynamical Astronomy* 134.6 (Dec. 2022). ISSN: 1572-9478. DOI: [10.1007/s10569-022-10115-2](https://doi.org/10.1007/s10569-022-10115-2). URL: <http://dx.doi.org/10.1007/s10569-022-10115-2>.
- [23] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. *U-Net: Convolutional Networks for Biomedical Image Segmentation*. 2015. arXiv: [1505.04597](https://arxiv.org/abs/1505.04597) [cs.CV].
- [24] D.J. Scheeres, B Khushalani, and R.A Werner. “Estimating asteroid density distributions from shape and gravity information.” In: *Planetary and Space Science* 48.10 (2000). Asteroids, Comets, Meteors (ACM) Conference, pp. 965–971. ISSN: 0032-0633. DOI: [https://doi.org/10.1016/S0032-0633\(00\)00064-7](https://doi.org/10.1016/S0032-0633(00)00064-7). URL: <https://www.sciencedirect.com/science/article/pii/S0032063300000647>.
- [25] Josef Sebera, Aleš Bezděk, Ivan Pešek, and Tomáš Henych. “Spheroidal models of the exterior gravitational field of Asteroids Bennu and Castalia.” In: *Icarus* 272 (July 2016), pp. 70–79. ISSN: 0019-1035. DOI: [10.1016/j.icarus.2016.02.038](https://doi.org/10.1016/j.icarus.2016.02.038). URL: <http://dx.doi.org/10.1016/j.icarus.2016.02.038>.
- [26] Hang Si. “TetGen, a Delaunay-Based Quality Tetrahedral Mesh Generator.” In: *ACM Trans. Math. Softw.* 41.2 (Feb. 2015). ISSN: 0098-3500. DOI: [10.1145/2629697](https://doi.org/10.1145/2629697). URL: <https://doi.org/10.1145/2629697>.
- [27] Liisa-Ida Sorsa, Mika Takala, Patrick Bambach, Jakob Deller, Esa Vilenius, Jessica Agarwal, Kieran A. Carroll, Özgür Karatekin, and Sampsa Pursiainen. “Tomographic inversion of gravity gradient field for a synthetic Itokawa model.” In: *Icarus* 336 (Jan. 2020), p. 113425. ISSN: 0019-1035. DOI: [10.1016/j.icarus.2019.113425](https://doi.org/10.1016/j.icarus.2019.113425). URL: <http://dx.doi.org/10.1016/j.icarus.2019.113425>.
- [28] Yu Takahashi and D.J. Scheeres. “Morphology driven density distribution estimation for small bodies.” In: *Icarus* 233 (May 2014), pp. 179–193. ISSN: 0019-1035. DOI: [10.1016/j.icarus.2014.02.004](https://doi.org/10.1016/j.icarus.2014.02.004). URL: <http://dx.doi.org/10.1016/j.icarus.2014.02.004>.

- [29] Pasquale Tricarico. "Global gravity inversion of bodies with arbitrary shape." In: *Geophysical Journal International* 195.1 (Aug. 2013), pp. 260–275. ISSN: 0956-540X. DOI: [10.1093/gji/ggt268](https://doi.org/10.1093/gji/ggt268). URL: <http://dx.doi.org/10.1093/gji/ggt268>.
- [30] J. Veverka et al. "NEAR at Eros: Imaging and Spectral Results." In: *Science* 289.5487 (Sept. 2000), pp. 2088–2097. ISSN: 1095-9203. DOI: [10.1126/science.289.5487.2088](https://doi.org/10.1126/science.289.5487.2088). URL: <http://dx.doi.org/10.1126/science.289.5487.2088>.
- [31] Kevin J. Walsh. "Rubble Pile Asteroids." In: *Annual Review of Astronomy and Astrophysics* 56.1 (Sept. 2018), pp. 593–624. ISSN: 1545-4282. DOI: [10.1146/annurev-astro-081817-052013](https://doi.org/10.1146/annurev-astro-081817-052013). URL: <http://dx.doi.org/10.1146/annurev-astro-081817-052013>.
- [32] Robert A. Werner and Daniel J. Scheeres. "Exterior gravitation of a polyhedron derived and compared with harmonic and mascon gravitation representations of asteroid 4769 Castalia." In: *Celestial Mechanics and Dynamical Astronomy* 65.3 (1997). ISSN: 1572-9478. DOI: [10.1007/bf00053511](https://doi.org/10.1007/bf00053511). URL: <http://dx.doi.org/10.1007/BF00053511>.
- [33] Patrick T. Wittick and Ryan P. Russell. "Mixed-model gravity representations for small celestial bodies using mascons and spherical harmonics." In: *Celestial Mechanics and Dynamical Astronomy* 131.7 (June 2019). ISSN: 1572-9478. DOI: [10.1007/s10569-019-9904-6](https://doi.org/10.1007/s10569-019-9904-6). URL: <http://dx.doi.org/10.1007/s10569-019-9904-6>.
- [34] Guochao Wu, Yue Wei, Siyuan Dong, Tao Zhang, Chunguo Yang, Linjiang Qin, and Qingsheng Guan. "Improved Gravity Inversion Method Based on Deep Learning with Physical Constraint and Its Application to the Airborne Gravity Data in East Antarctica." In: *Remote Sensing* 15.20 (2023). ISSN: 2072-4292. DOI: [10.3390/rs15204933](https://doi.org/10.3390/rs15204933). URL: <https://www.mdpi.com/2072-4292/15/20/4933>.
- [35] Fenqiang Zhao, Shunren Xia, Zhengwang Wu, Dingna Duan, Li Wang, Weili Lin, John H. Gilmore, Dinggang Shen, and Gang Li. "Spherical U-Net on Cortical Surfaces: Methods and Applications." In: *Information Processing in Medical Imaging*. Springer International Publishing, 2019, pp. 855–866. ISBN: 9783030203511. DOI: [10.1007/978-3-030-20351-1_67](https://doi.org/10.1007/978-3-030-20351-1_67). URL: http://dx.doi.org/10.1007/978-3-030-20351-1_67.