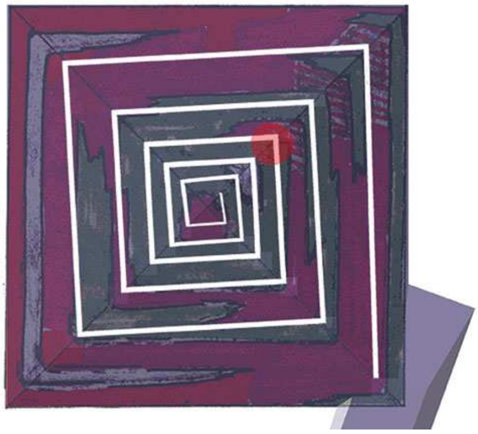
Universität Stuttgart

Geodätisches Institut

Photography-aided gravity modeling of solid bodies



Studienarbeit im Studiengang

Geodäsie und Geoinformatik

an der Universität Stuttgart

Christof Lorenz

Stuttgart, März 2010

Betreuer: Prof. Dr.-Ing. Nico Sneeuw
Universität Stuttgart

Dr.-Ing. (Univ. of Calgary) Matthias Weigelt
Universität Stuttgart

Erklärung der Urheberschaft

Ich erkläre hiermit an Eides statt, dass ich die vorliegende Arbeit ohne Hilfe Dritter und ohne Benutzung anderer als der angegebenen Hilfsmittel angefertigt habe; die aus fremden Quellen direkt oder indirekt übernommenen Gedanken sind als solche kenntlich gemacht. Die Arbeit wurde bisher in gleicher oder ähnlicher Form in keiner anderen Prüfungsbehörde vorgelegt und auch noch nicht veröffentlicht.

Ort, Datum

Unterschrift

Abstract

Not all secrets of the The Great Pyramid of Giza were revealed, even after centuries of observation and research. One of the main questions concerns the construction of the pyramid. The most popular and reasonable theory assumes the old Egyptians to use an exterior ramp in the lower third and an interior ramp in the upper two thirds of the pyramid on which the stones were carried upstairs. However, there is no evidence that this is really true. Microgravimetry-measuring techniques are able to give information about the inner mass distribution of the pyramid and hence reveal yet unknown facts about the inner structure.

Therefore, a reference gravity signal must be computed in order to detect mass deviations in the inside. In this work, an approach is discussed which uses photographs to construct a three-dimensional model of a body. It is shown that the information gained from three-dimensional reconstruction can be used to construct a solid body.

For the computation of the gravity signal of this solid body an algorithm is applied which transforms the volume integral in Newton's law of gravity into line integrals, which allows the computation of gravitational quantities for arbitrary polyhedra. With the help of a small section of the Great Pyramid it is shown that detecting inner mass deviations from a reference body requires detailed knowledge about the surface. As the errors in the measured gravity signal caused by a mis-modeled body might have a high magnitude, the signal from inner mass deviations might completely vanish. However, if the surface of an object is well known it is indeed possible to make a statement about the inner structure of a body based on close-mesh measurements on its surface.

Keywords: Gravity modeling, surface reconstruction, Gaussian divergence theorem, Great Pyramid of Khufu, Polyhedron

Zusammenfassung

Trotz jahrelanger Beobachtungen und Nachforschungen wurden nicht alle Geheimnisse der Großen Pyramide von Gizeh gelöst. Eine der Hauptfragen betrifft den Bau der Pyramide. Die gängigste Theorie besagt, dass die alten Ägypter im unteren Drittel eine äußere aber in den zwei oberen Dritteln der Pyramide eine innere Rampe benutzt haben, um die Steine nach oben zu befördern. Allerdings gibt es bisher keinen Beweis, ob diese Theorie der Wahrheit entspricht. Aber es wird angenommen, dass mit mikrogravimetrischen Beobachtungen Informationen über die innere Massenverteilung der Pyramide gesammelt werden können um dadurch bisher unbekannte Eigenschaften über die innere Zusammensetzung aufzudecken.

Daher muss ein Referenzsignal berechnet werden um Massenabweichungen im Inneren entdecken zu können. In dieser Arbeit wird ein Ansatz behandelt, bei dem Photographien genutzt werden, um ein drei-dimensionales Oberflächenmodell eines Körpers zu berechnen. Es wird gezeigt, dass die bei der Rekonstruktion enstandenden Daten genutzt werden können, um einen festen Körper zu konstruieren.

Zur Berechnung des Schweresignals dieses Körpers wird ein Ansatz genutzt, bei dem die Volumenintegrale in Newton's Gravitationsgesetz in Linienintegrale transformiert werden, was die Berechnung von gravitationellen Größen beliebiger Polyhedren erlaubt. Mit Hilfe eines kleinen Ausschnittes der Großen Pyramide von Gizeh wird gezeigt, dass die Detektion von inneren Massenvariationen ein genaues Oberflächenmodell benötigt. Da die durch eine unsaubere Modellierung des Objekts hervorgerufenen Fehler bereits eine hohe Signalstärke haben können, ist es möglich dass das wahre Signal von inneren Massevariationen komplett darin verschwindet. Allerdings ermöglichen ein genaues Oberflächenmodell sowie engmaschige Messungen auf der Oberfläche die Detektion und Beschreibung von Massevariationen im Inneren.

Schlüsselwörter: Schweremodellierung, Oberflächenrekonstruktion, Gauss'sches Divergenztheorem, Große Pyramide von Khufu, Polyhedron

Contents

1 Introduction **1**

2 Analytical expression for the gravitational field of a polyhedron **5**
- 2.1 Introduction in the evaluation procedure 5
- 2.2 From volume to surface integrals 7
- 2.3 From surface to line integrals 9
- 2.4 Evaluation of the line integrals 11
- 2.5 Singularities 13
  - 2.5.1 P inside the face S_p 13
  - 2.5.2 P outside the face S_p but P' inside the face S_p 13

3 Computational aspects and validation **17**
- 3.1 Implementation of the algorithm 17
- 3.2 Data handling 24
- 3.3 Validation 26
  - 3.3.1 Simple body, point-wise validation 26
  - 3.3.2 Simple body, line-wise validation 27
  - 3.3.3 Complex body, area-wise validation 30
- 3.4 Case study: Signal strength of a varying mass distribution 32

4 From photography to a solid body **41**
- 4.1 Creating a three-dimensional surface model 41
  - 4.1.1 Taking the photographs 41
  - 4.1.2 Using a web-service for 3D-modelling 43
  - 4.1.3 Download, processing and visualization of the model 44
  - 4.1.4 Orientation of the 3D-model 46
  - 4.1.5 Building a solid body from a surface model 49

5 Sensitivity analysis **53**
- 5.1 Generation of a reference model 53
- 5.2 Direct comparison detailed vs. simple model 56

6 Summary and outlook **59**

A Attraction and potential of a solid body **63**

B Line-wise validation, further results **67**

List of Figures

1.1 Schematic demonstration of the interiour ramp theory 1
1.2 Signal observed by a French team of scientists in 1982 2

2.1 Elements of a polyhedron 5
2.2 Geometrical elements of a plane of a polyhedron 6
2.3 Singular cases and definition of the circular area around P' 14

3.1 Example of a polyhedron 17
3.2 Computation of the line segments $\boldsymbol{G}_{pq}$ 18
3.3 Computation of the surface unit normals $\boldsymbol{N}_i$ 19
3.4 Computation of the segment unit normals $\boldsymbol{n}_{1,j}$ of the face S_1 19
3.5 Projection P_1' of P onto the plane which contains the face S_1 20
3.6 Projection pf P onto a plane (P') and a line segment (P'') 21
3.7 Elements of the linear system to compute the coordinates of P'' 22
3.8 Example of a cube and its vertices 25
3.9 Location of the computation points for point-wise validation 27
3.10 Three lines parallel to the $x-$axis used for line-wise validation 28
3.11 Components of the attraction along the three validation lines 29
3.12 Difference between the results of `polygrav.m` and `accrrp.m` 29
3.13 Difference between the results of `polygrav.m` and `gravmag3d.m` 30
3.14 Complex body for validation 30
3.15 Attraction computed on the surface with `polygrav.m` 31
3.16 Difference of the attracion of the body between `polygrav.m` and `accrrp.m` 31
3.17 Location and extent of the removed elements in the reference body 33
3.18 Signal of a descending hole 34
3.19 Signal of a contracting hole 35
3.20 Signal of a growing column 38
3.21 Signal of a growing vacuty 39

4.1 Optimal image sequence for reconstruction 42
4.2 Image sequence of a close-up view of the Great Pyramid in Gizeh 43
4.3 Parameters of the model-creation in the model viewer provided by ARC3D 45
4.4 Preprocessing of the reconstructed scene in `Meshlab` 47
4.5 Hidden areas caused by the recording angle 48
4.6 Finished reconstruction of the scene 49
4.7 Example of a solid body constructed from a triangulated surface 50
4.8 Side-view of an overhanging area and the corresponding elevation of the normal vectors 50

5.1 Different views of the detailed and simplified model 54
5.2 Height differences between the surface of the detailed and simplified model 55
5.3 Plane containing the computation points 55

5.4 Attraction by the detailed and simplified model 56
5.5 Difference between the attraction of the detailed and simple model 57

A.1 Components of the gravitaional force vector 64
A.2 Force vector of an element of mass 65

B.1 Three lines used for line-wise validation 67
B.2 Components of the attraction along the three validation lines 68
B.3 Difference between the results of `polygrav.m` and `accrrp.m` 68
B.4 Difference between the results of `polygrav.m` and `grvmag3d.m` 69
B.5 Components of the attraction along the three validation lines 69
B.6 Difference between the results of `polygrav.m` and `accrrp.m` 70
B.7 Difference between the results of `polygrav.m` and `grvmag3d.m` 70

Introduction 1

Even after many years of research and observations, not all secrets of the Great Pyramid of Khufu in Egypt are revealed. One of the remaining but at the same time most important questions concerns the construction of the pyramid. By now, most scientists agree that it was built by moving huge stones from a quarry. However, there are still many opinions on how the stones really got to their designated positions (see e.g. Brier (2007)). For a long time, most of these theories assumed the Egyptians using an exterior ramp or a crane to move the stones. But these theories always led to discrepancies which could not be solved. For example, if the Egyptians really used one exterior ramp to move the stones upstairs, this ramp had to be over one mile long to reach the top. But there is neither enough free space on the Giza plateau nor evidence of such a massive construction, to say nothing of the required manpower. The crane theory would have needed an immense amount of timber, which was simply not available in Egypt. These are only two examples of many theories which caused serious problems.

In 2000, the French architect Jean-Pierre Houdin introduced a completely new theory (see e.g. Houdin (2006)). He argued that an exterior ramp was only used in the lower third of the pyramid. After this part was finished, a second ramp was build in the inside (see fig. 1.1).

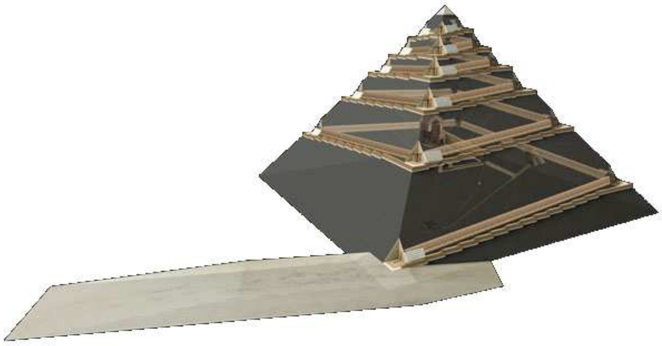


Figure 1.1: Schematic demonstration of the interiour ramp theory

Unfortunately, none of the theories could be confirmed. But a French team of scientists surveyed the pyramid in 1986 with microgravimetry. By observing the density of the different sections of the pyramid, they wanted to detect hidden chambers in the inside.

The team could not interpret the measured signal but when Jean-Pierre Houdin introduced his new theory, their observations suddenly maked sense. The image showed a spiral in the inside which must have been the interiour ramp.

However, as the french team could not interpret their observations at first, there is still no reasonable proof of Houdin's theory. Therefore, clarifying the question on how the pyramid was build would require a new mission with modern gravimeters and measuring techniques. The interior ramp theory might be accepted if similar observations to the French scientists can be made.

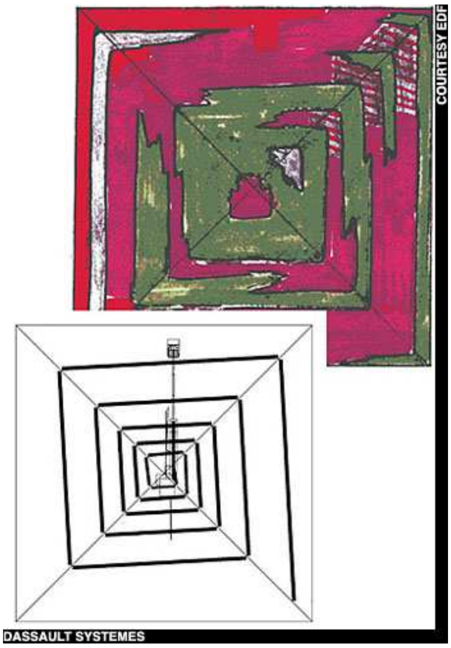


Figure 1.2: Signal observed by a French team of scientists in 1982

Moreover, using gravimetry to explore the inner structure of a solid body is an elegant and nondestructive way to gain knowledge about density, material and anomalies inside an object. Thus, it is an important tool to observe and examine heritage sites. In any case, information about a reference body or forward model is needed in order to perform a reasonable comparison of measured and predicted values which can lead to conclusions about the inner structure of an object. The next question is how this reference body can be constructed. Today, modern surveying techniques like laser scanning allow the computation of a very detailed model of the outer surface of an object. However, such equipment is expensive and the observation would take very long. But as the three-dimensional modeling from photographs is on advance, this work deals with an approach to build a reference body and its gravity purely from images. Therefore, a method is discussed which allows the computation of potential and attraction of arbitrary polyhedra. As it is assumed that there is no information about the interior structure of an object, it will be shown that such polyhedra can be used to fill a reconstructed scene from photographs with mass, i.e. to build a solid body. In the end, it is discussed how such a detailed model might differ from a simple body in terms of gravity. The successive steps will be visualized with images and results from a close-up view of the Great Pyramid. The work is thus divided in the following parts:

- description of an algorithm which allows the computation of potential and attraction of arbitrary polyhedra (chapter 2),
- some aspects about the implementation of the algorithm in a computer program and the validation of such a program (chapter 3),
- a step-for-step guide on how to generate a gravity model purely from photographs (chapter 4) and
- a sensitivity analysis on the magnitude of a very detailed gravity model vs. a simplified reference body (chapter 5).

Analytical expression for the gravitational field of a polyhedron

2

In this chapter, a closed analytical expression for the gravitational field of an arbitrary homogeneous polyhedron is derived. It is thus mainly based on the relations found by Petrovic (1996), even if the beginnings of the presented method reach back to e.g. Götze (1976). However, the successive steps are explained in detail and singularities are dealt with more precisely.

2.1 Introduction in the evaluation procedure

By *polyhedron* a three-dimensional body is meant which surface consists of planar faces connected along straight edges called segments or at points called vertices (see Fig. 2.1) (Tsoulis, 1999).

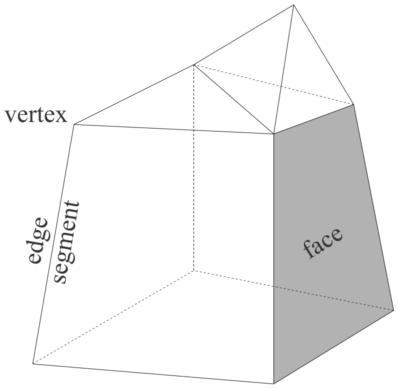


Figure 2.1: Elements of a polyhedron

The gravitational potential and its derivatives (see e.g. Hofmann-Wellenhof & Moritz (2006)) are given by

$$V = G\rho \iiint_v \frac{1}{l} dv \tag{2.1.1}$$

$$V_{x,y,z} = G\rho \iiint_v \frac{\partial}{\partial x, \partial y, \partial z} \frac{1}{l} dv \tag{2.1.2}$$

where

G	Gravitational constant with $G = 6.67259 \cdot 10^{-11} \mathrm{m}^3/\mathrm{kg} \cdot \mathrm{s}^2$
ρ	density
l	distance between the mass element dv and the computation point $P(x, y, z)$
dv	mass element

For the following derivations, it is assumed that the computation point is located at the origin of a local Cartesian coordinate system. Then, the distance l of the mass element dv from the computation point is

$$l = \sqrt{x^2 + y^2 + z^2} \tag{2.1.3}$$

However, a direct solution of the volume integrals in equation (2.1.1) is obviously very cumbersome due to the arbitrary shape and form of a polyhedron. By applying the Gaussian divergence theorem, one can transform the volume integrals into surface integrals when the computation point P is projected onto the n faces of the polyhedron. Thus, we obtain n projected computation points P'_n. If this strategy is applied again, the surface integral of one face transforms into m line integrals over each line segment of the face. Therefore, the projection of P'_n on each single line segment P''_n is needed. It will be shown that there exist closed analytical solutions for these line integrals, which can be looked up in standard integral tables. Finally, the potential and its derivatives of the polyhedron are obtained by summing over all line segments of all faces of the body. The geometrical quantities which will be used for these derivations are illustrated in Fig. 2.2.

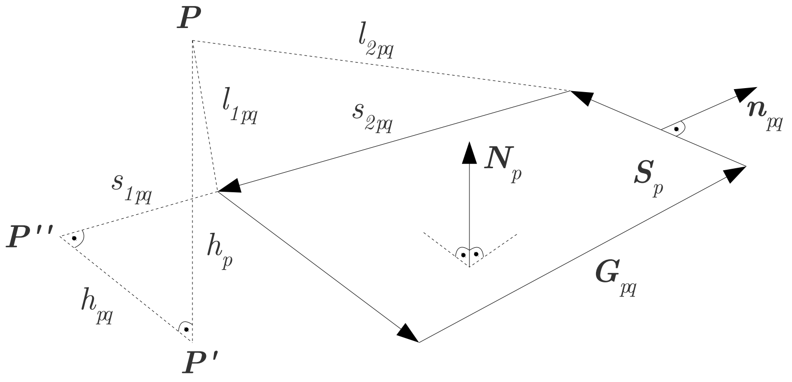


Figure 2.2: Definition of the geometrical elements of a specific face of a polyhedron which are needed for the evaluation procedure

2.2 From volume to surface integrals

Let v be a regular body with the closed surface S which consists of finite regular faces and $\boldsymbol{N}$ the outer unit normals to S. If $\boldsymbol{v}$ is a vector field with continuous derivatives, the following relation holds:

$$\iiint\limits_{v} \operatorname{div} \boldsymbol{v} \, dv = \oiint\limits_{S} \boldsymbol{v} \cdot \boldsymbol{N} dS \tag{2.2.1}$$

where $\cdot$ is the inner product of the vectors $\boldsymbol{v}$ and $\boldsymbol{N}$.

Applying equation (2.2.1) to the present problem means to find a vector field $\boldsymbol{v}$, which satisfies the following relations:

$$\operatorname{div} \boldsymbol{v} = \frac{1}{l} \tag{2.2.2}$$

$$\operatorname{div} \boldsymbol{v} = \frac{\partial}{\partial x, \partial y, \partial z} \frac{1}{l} \tag{2.2.3}$$

A solution of this problem can be found in Petrovic (1996) and Tsoulis (1999). Thus, the first relation (2.2.2) is satisfied if

$$\boldsymbol{v} = \frac{1}{2l} \begin{bmatrix} x & y & z \end{bmatrix}^{\mathsf{T}} = \nabla \frac{l}{2} = \operatorname{grad} \frac{l}{2} \tag{2.2.4}$$

where ∇ is the Nabla operator $\nabla = \left(\frac{\partial}{\partial x}, \frac{\partial}{\partial y}, \frac{\partial}{\partial z} \right)$. This can be proven if one applies the divergence operator:

$$\begin{aligned} \operatorname{div} \boldsymbol{v} &= \nabla \cdot \nabla \frac{l}{2} = \nabla \cdot \begin{bmatrix} \frac{x}{2l} & \frac{y}{2l} & \frac{z}{2l} \end{bmatrix}^{\mathsf{T}} \\ &= \frac{1}{2} \left[\frac{\partial}{\partial x} \left(\frac{x}{2l} \right) + \frac{\partial}{\partial y} \left(\frac{y}{2l} \right) + \frac{\partial}{\partial z} \left(\frac{z}{2l} \right) \right] \\ &= \frac{1}{2} \left[\frac{l^2 - x^2}{l^3} + \frac{l^2 - y^2}{l^3} + \frac{l^2 - z^2}{l^3} \right] \\ &= \frac{1}{l} \end{aligned}$$

If we look at the first derivative of the potential, a simple solution for relation (2.2.3) would be

$$\boldsymbol{v} = \frac{1}{l} \begin{bmatrix} \delta_{ix} & \delta_{iy} & \delta_{iz} \end{bmatrix}^{\mathsf{T}} \tag{2.2.5}$$

where $i = x, y, z$ and δ_{ij} is the Kronecker delta, which is defined as

$$\delta_{ij} = \begin{cases} 0 & \text{if} \quad i \neq j \\ 1 & \text{if} \quad i = j \end{cases} \tag{2.2.6}$$

Again, this can be proven if the divergence operator is applied:

$$
\begin{aligned}
\operatorname{div} \boldsymbol{v} &= \nabla \cdot \left(\frac{1}{l} \begin{bmatrix} \delta_{ix} & \delta_{iy} & \delta_{iz} \end{bmatrix}^{\mathsf{T}} \right), \quad i = x, y, z \\
&= \begin{cases} \nabla \cdot \left(\frac{1}{l} \begin{bmatrix} 1 & 0 & 0 \end{bmatrix}^{\mathsf{T}} \right) & i = x \\ \nabla \cdot \left(\frac{1}{l} \begin{bmatrix} 0 & 1 & 0 \end{bmatrix}^{\mathsf{T}} \right) & i = y \\ \nabla \cdot \left(\frac{1}{l} \begin{bmatrix} 0 & 0 & 1 \end{bmatrix}^{\mathsf{T}} \right) & i = z \end{cases} \\
&= \begin{cases} \frac{\partial}{\partial x} \left(\frac{1}{l} \right) & i = x \\ \frac{\partial}{\partial y} \left(\frac{1}{l} \right) & i = y \\ \frac{\partial}{\partial z} \left(\frac{1}{l} \right) & i = z \end{cases} \\
&= \frac{\partial}{\partial x, \partial y, \partial z} \frac{1}{l}
\end{aligned}
$$

Inserting these relations into equation (2.2.1), the following expression for the volume integrals of the potential and its first derivatives is obtained:

$$
V = \frac{G\rho}{2} \oiint_S \frac{1}{l} \begin{bmatrix} x & y & z \end{bmatrix}^{\mathsf{T}} \cdot \boldsymbol{N} dS \tag{2.2.7}
$$

$$
V_i = G\rho \oiint_S \frac{1}{l} \begin{bmatrix} \delta_{ix} & \delta_{iy} & \delta_{iz} \end{bmatrix}^{\mathsf{T}} \cdot \boldsymbol{N} dS, \quad i = x, y, z \tag{2.2.8}
$$

Since $|\boldsymbol{N}| = 1$ and thus $N_i = \cos(\boldsymbol{N}, \boldsymbol{e}_i)$ with $i = x, y, z$, equations (2.2.7) and (2.2.8) can be written as

$$
V = \frac{G\rho}{2} \oiint_S \frac{1}{l} \left[x \cos(\boldsymbol{N}, \boldsymbol{e}_x) + y \cos(\boldsymbol{N}, \boldsymbol{e}_y) + z \cos(\boldsymbol{N}, \boldsymbol{e}_z) \right] dS \tag{2.2.9}
$$

$$
V_i = G\rho \oiint_S \frac{1}{l} \cos(\boldsymbol{N}, \boldsymbol{e}_i) \, dS, \quad i = x, y, z \tag{2.2.10}
$$

The goal of these derivations is to express the volume integral over a body in terms of surface integrals only over its closed surface. Therefore, one can say that the surface S of a polyhedron consists of a closed surface polygon with n faces S_p, i.e.

$$
S = \bigcup_{p=1}^{n} S_p \tag{2.2.11}
$$

where each face is a plane with constant outer unit normal $\boldsymbol{N}_p$. In equations (2.2.9) and (2.2.10) the quantity inside the brackets is the equation of a plane in Hessian normal form. For a specific face S_p, it reads as

$$
\sigma_p h_p = \left[x \cos(\boldsymbol{N}_p, \boldsymbol{e}_x) + y \cos(\boldsymbol{N}_p, \boldsymbol{e}_y) + z \cos(\boldsymbol{N}_p, \boldsymbol{e}_z) \right] \tag{2.2.12}
$$

where h_p is the orthogonal distance of the computation point from that face and $\sigma_p = -1$ if $\boldsymbol{N}_p$ points towards the half-space containing the computation point and $\sigma_p = 1$ if not. There are certain singular cases when $\sigma_p = 0$. This will be discussed in section 2.5. As both parameters remain constant for a specific face S_p, relation (2.2.11) can be used to replace the ring integral over the whole surface S with the sum of the surface integrals of the faces S_p, which simplifies equations (2.2.9) and (2.2.10) to

$$V = \frac{G\rho}{2} \sum_{p=1}^{n} \sigma_p h_p \iint\limits_{S_p} \frac{1}{l} dS \tag{2.2.13}$$

$$V_i = G\rho \sum_{p=1}^{n} \cos\left(\boldsymbol{N}_p, \boldsymbol{e}_i\right) \iint\limits_{S_p} \frac{1}{l} dS \quad i = x, y, z \tag{2.2.14}$$

Thus, the evaluation of the volume integral is reduced to the evaluation of the surface integral

$$\mathcal{A}_p = \iint\limits_{S_p} \frac{1}{l} dS \tag{2.2.15}$$

as all other quantities can be computed directly.

2.3 From surface to line integrals

The next step will be the transformation of the surface integrals to line integrals. Therefore, the computation point P must be orthogonally projected on each plane of the polyhedron, which results in n projected computation points P'. This is performed by simply multiplying the surface normal vector with the distance of the plane from the computation point and thus

$$\boldsymbol{X}_{P'} = -\sigma_p \boldsymbol{N} \cdot h_p \tag{2.3.1}$$

In the latter equation, the factor σ_p is used to ensure a correct orientation of the projection.

For the following derivations, P' will be the origin of a new rectangular local coordinate system $\boldsymbol{e}_{x'}, \boldsymbol{e}_{y'}, \boldsymbol{e}_{z'}$. The $z-$axis of this coordinate system is $\boldsymbol{N}_p$, i.e. the surface unit normal. The $x-$ and $y-$ axis are arbitrary, but must be located in the plane and complete the system to a right handed rectangular coordinate system.

The transformation of surface to line integrals is similar to the derivations in section 2.2. Thus, the Gaussian divergence theorem is applied again:

$$\iint\limits_{S_p} \operatorname{div} \boldsymbol{v} dS = \oint\limits_{G_p} \boldsymbol{v} \cdot \boldsymbol{n} ds \tag{2.3.2}$$

where S_p is a regular surface which is limited by the closed boundary line of the polygon G_p, $\boldsymbol{n}$ the outer unit normals in the plane to G_p and $\boldsymbol{v}$ a vector field with continuous derivatives.

Again, the strategy is to find a vector field, which satisfies the following relation:

$$\operatorname{div}\boldsymbol{v} = \frac{1}{l} \tag{2.3.3}$$

which is the integral part of equations (2.2.13) and (2.2.14). According to the definition of the local coordinate system, the parameter r is the radial distance from the projected computation point P' in the plane, i.e.

$$r = \sqrt{x'^2 + y'^2} \tag{2.3.4}$$

Thus, the reciprocal distance from the computation point P is

$$\frac{1}{l} = \frac{1}{\sqrt{h_p^2 + r^2}} \tag{2.3.5}$$

as h_p is the radial distance of P' from P. It should be mentioned again that these relations hold only for a specific plane (or face), i.e. the coordinates which were reduced to a projected computation point P'. Now, a solution for the problem in equation (2.3.3) is according to Petrovic (1996) to express the vector field $\boldsymbol{v}$ as a gradient

$$\boldsymbol{v} = \operatorname{grad}\left(l + \frac{h_p}{2}\ln\frac{l - h_p}{l + h_p}\right) \tag{2.3.6}$$

As $\frac{\partial l}{\partial x'} = \frac{x'}{l}$ and $\frac{\partial l}{\partial y'} = \frac{y'}{l}$, the vector field can be written as

$$\boldsymbol{v} = \frac{1}{l}\begin{bmatrix} x' & y' \end{bmatrix}^{\mathsf{T}} + \frac{h_p^2}{r^2 l}\begin{bmatrix} x' & y' \end{bmatrix}^{\mathsf{T}} \tag{2.3.7}$$

Inserting these relations into equation (2.2.15) and using the Gaussian divergence theorem from equation (2.3.3), the following expression for the evaluation of the surface integral is found:

$$\mathcal{A}_p = \oint\limits_{G_p} \frac{1}{l}\begin{bmatrix} x' & y' \end{bmatrix}^{\mathsf{T}} \cdot \boldsymbol{n}_p ds + h_p^2 \oint\limits_{G_p} \frac{1}{r^2 l}\begin{bmatrix} x' & y' \end{bmatrix}^{\mathsf{T}} \cdot \boldsymbol{n}_p ds \tag{2.3.8}$$

where each G_p is a closed (i.e. ring-) polygon which consists of m segments G_{pq}, i.e.

$$G_p = \bigcup_{q=1}^{m} G_{pq} \tag{2.3.9}$$

with constant outer segment normals $\boldsymbol{n}_{pq}$. Note that this procedure is very similar to the transformation from volume to surface integrals. Thus, the sum operator can be applied again to evaluate the integral over the whole surface:

$$\mathcal{A}_p = \sum_{q=1}^{m} \int\limits_{G_{pq}} \frac{1}{l}\begin{bmatrix} x' & y' \end{bmatrix}^{\mathsf{T}} \cdot \boldsymbol{n}_{pq} ds + h_p^2 \sum_{q=1}^{m} \int\limits_{G_{pq}} \frac{1}{r^2 l}\begin{bmatrix} x' & y' \end{bmatrix}^{\mathsf{T}} \cdot \boldsymbol{n}_{pq} ds \tag{2.3.10}$$

Again, the latter expression can be rewritten as

$$\begin{aligned}\mathcal{A}_p =& \sum_{q=1}^{m} \int\limits_{G_{pq}} \frac{1}{l} \left[x' \cos\left(\boldsymbol{n}_{pq}, \boldsymbol{e}_{x'}\right) + y' \cos\left(\boldsymbol{n}_{pq}, \boldsymbol{e}_{y'}\right)\right] ds \\ &+ h_p^2 \sum_{q=1}^{m} \int\limits_{G_{pq}} \frac{1}{r^2 l} \left[x' \cos\left(\boldsymbol{n}_{pq}, \boldsymbol{e}_{x'}\right) + y' \cos\left(\boldsymbol{n}_{pq}, \boldsymbol{e}_{y'}\right)\right] ds\end{aligned} \tag{2.3.11}$$

Just like the equation of a plane in the three dimensional space, the term inside the brackets can be expressed with two parameters in the plane where $z' = 0$. Thus, it can be written as

$$\sigma_{pq} h_{pq} = \left[x' \cos\left(\boldsymbol{n}_{pq}, \boldsymbol{e}_{x'}\right) + y' \cos\left(\boldsymbol{n}_{pq}, \boldsymbol{e}_{y'}\right)\right] \tag{2.3.12}$$

where h_{pq} is the distance of the projected point P' from the line segment G_{pq} and $\sigma_{pq} = -1$ if $\boldsymbol{n}_{pq}$ points towards the halt-plane containing P' and $\sigma_{pq} = 1$ if not. Just like the parameters σ_p and h_p for a plane, the parameters σ_{pq} and h_{pq} remain constant for the whole line. Again, a singular case might appear if $\sigma_{pq} = 0$ which is discussed in section 2.5.

If equation (2.3.12) is inserted into (2.3.11), we get an expression for the surface integral (2.2.15) in terms of line integrals:

$$\mathcal{A}_p = \sum_{q=1}^{m} \sigma_{pq} h_{pq} \int\limits_{G_{pq}} \frac{1}{l} ds + h_p^2 \sum_{q=1}^{m} \sigma_{pq} h_{pq} \int\limits_{G_{pq}} \frac{1}{r^2 l} ds \tag{2.3.13}$$

2.4 Evaluation of the line integrals

The final gap is to find closed expressions for the two line integrals

$$\mathcal{B}_{pq} = \int\limits_{G_{pq}} \frac{1}{l} ds \tag{2.4.1}$$

$$\mathcal{C}_{pq} = \int\limits_{G_{pq}} \frac{1}{r^2 l} ds \tag{2.4.2}$$

in equation (2.3.13). Therefore, the point P', which was the orthogonal projection of the computation point P onto a plane, must be projected on each m line segments G_{pq} of face in this plane, which results in m new projected points P''. Every point P'' is taken as the origin of a one-dimensional coordinate system. The variable in this system will be the distance from its origin s. However, we can avoid the reduction to this coordinate system as the radial distance r from P' only depends on the coordinate s:

$$r = \sqrt{h_{pq}^2 + s^2} \tag{2.4.3}$$

Thus, the reciprocal distance can be written as

$$\frac{1}{l} = \frac{1}{\sqrt{h_p^2 + h_{pq}^2 + s^2}} \tag{2.4.4}$$

If these relations are inserted into equations (2.4.1) and (2.4.2), we only have to find an analytical solution for the two integrals

$$\mathcal{B}_{pq} = \int\limits_{G_{pq}} \frac{1}{l} ds = \int\limits_{G_{pq}} \frac{1}{\sqrt{h_p^2 + h_{pq}^2 + s^2}} ds \tag{2.4.5}$$

$$\mathcal{C}_{pq} = \int\limits_{G_{pq}} \frac{1}{r^2 l} = \int\limits_{G_{pq}} \frac{1}{\left(h_{pq}^2 + s^2\right)\sqrt{h_p^2 + h_{pq}^2 + s^2}} ds \tag{2.4.6}$$

According to standard integral tables like Bronstein et al. (2000) or Gröbner & Hofreiter (1961), the solutions of the integrals are

$$\mathcal{B}_{pq} = \ln\left(s + \sqrt{h_p^2 + h_{pq}^2 + s^2}\right) \tag{2.4.7}$$

$$\mathcal{C}_{pq} = \frac{1}{h_{pq} h_p} \arctan \frac{s h_p}{h_{pq}\sqrt{s^2 + h_p^2 + h_{pq}^2}} \tag{2.4.8}$$

The integration limits are obviously the end-points of the line segment G_{pq}. Thus, they will be denoted with $s_{1_{pq}}$ and $s_{2_{pq}}$ respectively. The square term in both integrals, which is the distance of the computation point P from the end-points of the line segment, simplifies to

$$l_{1_{pq}} = \sqrt{s_{1_{pq}}^2 + h_p^2 + h_{pq}^2} \quad \text{and} \quad l_{2_{pq}} = \sqrt{s_{2_{pq}}^2 + h_p^2 + h_{pq}^2} \tag{2.4.9}$$

Inserting these integration limits into equations (2.4.7) and 2.4.8, we can evaluate the integrals since

$$\mathcal{B}_{pq} = \ln\left(s_{2_{pq}} + l_{2_{pq}}\right) - \ln\left(s_{1_{pq}} + l_{1_{pq}}\right) \tag{2.4.10}$$

$$\mathcal{C}_{pq} = \frac{1}{h_{pq} h_p}\left(\arctan \frac{h_p s_{2_{pq}}}{h_{pq} l_{2_{pq}}} - \arctan \frac{h_p s_{1_{pq}}}{h_{pq} l_{1_{pq}}}\right) \tag{2.4.11}$$

In close analogy to Petrovic (1996) and Tsoulis (1999), the terms in equations (2.4.10) and (2.4.11) will be abbreviated with

$$LN_{pq} = \ln\left(s_{2_{pq}} + l_{2_{pq}}\right) - \ln\left(s_{1_{pq}} + l_{1_{pq}}\right) = \ln \frac{s_{2_{pq}} + l_{2_{pq}}}{s_{1_{pq}} + l_{1_{pq}}} \tag{2.4.12}$$

$$AN_{pq} = \arctan \frac{h_p s_{2_{pq}}}{h_{pq} l_{2_{pq}}} - \arctan \frac{h_p s_{1_{pq}}}{h_{pq} l_{1_{pq}}} \tag{2.4.13}$$

By replacing the line integrals in equations (2.4.1) and (2.4.2) with these analytical expressions and inserting them into equation (2.3.13), we obtain the analytical expression for the integral terms:

$$\mathcal{A}_p = \sum_{q=1}^{m} \sigma_{pq} h_{pq} LN_{pq} + h_p \sum_{q=1}^{m} \sigma_{pq} AN_{pq} \tag{2.4.14}$$

and finally for the potential and its first derivatives

$$V = \frac{G\rho}{2} \sum_{p=1}^{n} \sigma_p h_p \left[\sum_{q=1}^{m} \sigma_{pq} h_{pq} LN_{pq} + h_p \sum_{q=1}^{m} \sigma_{pq} AN_{pq}\right] \tag{2.4.15}$$

$$V_i = G\rho \sum_{p=1}^{n} \cos\left(\boldsymbol{N}_p, \boldsymbol{e}_i\right) \left[\sum_{q=1}^{m} \sigma_{pq} h_{pq} LN_{pq} + h_p \sum_{q=1}^{m} \sigma_{pq} AN_{pq}\right] \tag{2.4.16}$$

If it is ensured that the outer unit normal of the planes $\boldsymbol{N}_p$ is a unit vector, the first derivatives of the potential could be further simplified to

$$V_i = G\rho \sum_{p=1}^{n} N_{p_i} \left[\sum_{q=1}^{m} \sigma_{pq} h_{pq} LN_{pq} + h_p \sum_{q=1}^{m} \sigma_{pq} AN_{pq} \right] \tag{2.4.17}$$

2.5 Singularities

It was already mentioned that we have to deal with singularities in some cases. In fact, the presented method is not allowed in these cases. It was always assumed that P and P' are both located outside the mass. This means that the vector field $\boldsymbol{v}$ has continuous derivatives in the whole integration area. If this is not ensured, we would not be able to apply the Gaussian divergence theorem. This becomes clear if the vector field in equation (2.3.6) is considered again. If P and/or P' is located in S_p, the term $r = \sqrt{x'^2 + y'^2}$ would become zero in P' which means that the expression $\frac{1}{r^2 l}$ is not defined in P'. Thus, the derivation in P' does not exist. The situations when this case appears are discussed in the following.

2.5.1 P inside the face S_p

If P is located in the face S_p, its orthogonal projection P' would be P itself. This means that the distance h_p from P to P' equals zero. In such a case, the integral $\mathcal{C}_{pq}$ which is undefined in P' (see equation (2.3.13)) is multiplicated with zero. This means that for practical considerations the non-existence of a continuous derivative in P' must not be taken into account as the problematic term vanishes. Thus, no further terms for this singularity must considered.

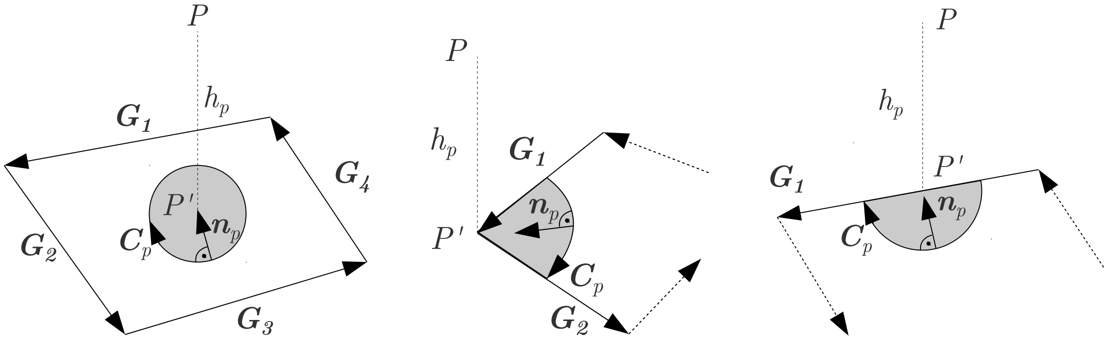
2.5.2 P outside the face S_p but P' inside the face S_p

This situation is a bit more complex. According to Tsoulis (1999), it is well known in potential theory and a satisfactory solution can be derived. The common approach is to divide the surface into two parts. One of them is a small area around P', which has the form of a circle (Fig. 2.3a), a sector (Fig. 2.3b) or a semicircle (Fig. 2.3c), depending on the position of P'. The other part contains the rest of the surface. In the latter, the Gaussian divergence theorem can be applied without difficulties. The contribution of the area surrounding P' can be taken equal to the limit of the expression for the area's potential or the attraction as the radius of the area including P' approaches zero (Kellogg, 1954).

For the beginning, we will reconsider equation (2.3.8), where the surface integral was transformed into line integrals:

$$\mathcal{A}_p = \iint_{S_p} \frac{1}{l} ds = \oint_{G_p} \frac{1}{l} \begin{bmatrix} x' & y' \end{bmatrix}^\mathsf{T} \cdot \boldsymbol{n}_p ds + h_p^2 \oint_{G_p} \frac{1}{r^2 l} \begin{bmatrix} x' & y' \end{bmatrix}^\mathsf{T} \cdot \boldsymbol{n}_p ds \tag{2.5.1}$$

It was already mentioned that in the case of a singularity the second integral can not be evaluated. Thus, we do not take the whole polygon at once, but divide it in an outer part G_p and the shaded circular area C_p in Fig. 2.3, which contains P'. This gives us a



(a) P' inside S_p, $\varphi = 2\pi$ (b) P' on a vertex, $\varphi = \vartheta$ (c) P' on a line segment, $\varphi = \pi$

Figure 2.3: Singular cases and definition of the circular area around P'

new boundary, which can be expressed as

$$\tilde{G_p} = G_p \bigcup C_p \tag{2.5.2}$$

i.e. $\tilde{G_p}$ is the union of the areas with and without P'. Due to equation (2.5.2), the integrals of (2.5.1) can be split up into integrals over G_p and C_p:

$$\begin{aligned} \tilde{\mathcal{A}}_p = & \oint\limits_{G_p} \frac{1}{l} \begin{bmatrix} x' & y' \end{bmatrix}^\mathsf{T} \cdot \boldsymbol{n}_p ds + h_p^2 \oint\limits_{G_p} \frac{1}{r^2 l} \begin{bmatrix} x' & y' \end{bmatrix}^\mathsf{T} \cdot \boldsymbol{n}_p ds \\ & + \oint\limits_{C_p} \frac{1}{l} \begin{bmatrix} x' & y' \end{bmatrix}^\mathsf{T} \cdot \boldsymbol{n}_p ds + h_p^2 \oint\limits_{C_p} \frac{1}{r^2 l} \begin{bmatrix} x' & y' \end{bmatrix}^\mathsf{T} \cdot \boldsymbol{n}_p ds \end{aligned} \tag{2.5.3}$$

where $\boldsymbol{n}_p$ is either the unit normal of G_p or C_p. The integrals over C_p can be evaluated when the radius of the circular area approaches zero. Therefore, we express the coordinates x' and y' in polar coordinates, where R is the radius of C_p and φ is the angular coordinate:

$$x' = R\cos\varphi \tag{2.5.4}$$

$$y' = R\sin\varphi \tag{2.5.5}$$

$$z' = h_p \tag{2.5.6}$$

Writing the radial distance r, the reciprocal distance $\frac{1}{l}$ and the line element ds in polar coordinates gives

$$r = \sqrt{x'^2 + y'^2} = R\sqrt{\sin^2\varphi + \cos^2\varphi} = R \tag{2.5.7}$$

$$l = \sqrt{x'^2 + y'^2 + z'^2} = \sqrt{R^2 + h_p^2} \tag{2.5.8}$$

$$ds = R d\varphi \tag{2.5.9}$$

The inward unit normal to C_p in polar coordinates is

$$\boldsymbol{n}_p = \begin{bmatrix} -\cos\varphi & -\sin\varphi \end{bmatrix}^\mathsf{T} \tag{2.5.10}$$

Finally, the integration limits ϑ must be discussed. They depend on the position of P'. Obviously, if P' is located inside S_p, we will have to integrate over the whole circle, i.e. $\vartheta = 2\pi$ (see Fig. 2.3(a)). If P' is located on a vertex (see Fig. 2.3(b)), ϑ is the angle between the line segments $\boldsymbol{G}_1$ and $\boldsymbol{G}_2$ and can be computed with the dot product:

$$\vartheta = \arccos \frac{\boldsymbol{G}_2 \cdot -\boldsymbol{G}_1}{|\boldsymbol{G}_1|\,|\boldsymbol{G}_2|} \tag{2.5.11}$$

If P' is on a line segment (see Fig. 2.3(c)) we would have to integrate over a semicircle, i.e. $\vartheta = \pi$. Furthermore, the dot product between $\begin{bmatrix} x' & y' \end{bmatrix}$ and $\boldsymbol{n}_p$ is obviously $-R$.

Now, all quantities in the integrals over C_p in equation (2.5.3) can be replaced with the latter expressions:

$$\oint\limits_{C_p} \frac{1}{l} \begin{bmatrix} x' & y' \end{bmatrix}^{\mathsf{T}} \cdot \boldsymbol{n}_p ds = \int\limits_0^{\vartheta} \frac{-R^2}{\sqrt{R^2 + h_p^2}} ds = \frac{-R\vartheta}{\sqrt{R^2 + h_p^2}} \tag{2.5.12}$$

$$h_p^2 \oint\limits_{C_p} \frac{1}{r^2 l} \begin{bmatrix} x' & y' \end{bmatrix}^{\mathsf{T}} \cdot \boldsymbol{n}_p ds = h_p^2 \int\limits_0^{\vartheta} \frac{-R^2}{R^2\sqrt{R^2 + h_p^2}} ds = \frac{-h_p^2\vartheta}{\sqrt{R^2 + h_p^2}} \tag{2.5.13}$$

As we only have to consider the location where P' is exactly located on a singularity, we let R approach zero. Thus, the contribution of C_p is given by the following limiting values of the integrals:

$$\lim_{R\to 0} \frac{-R\vartheta}{\sqrt{R^2 + h_p^2}} = 0 \tag{2.5.14}$$

$$\lim_{R\to 0} \frac{-h_p^2\vartheta}{\sqrt{R^2 + h_p^2}} = -\vartheta h_p \tag{2.5.15}$$

This means, depending on the position of P', the following correction terms must be taken into account, which will be denoted with $\mathcal{SING}_p$

1. P' is located inside S_p: $\mathcal{SING}_p = -2\pi h_p$
2. P' is located on a vertex: $\mathcal{SING}_p = -\vartheta h_p = -\arccos \frac{\boldsymbol{G}_2 \cdot -\boldsymbol{G}_1}{|\boldsymbol{G}_1|\,|\boldsymbol{G}_2|} h_p$
3. P' is located on a line segment: $\mathcal{SING}_p = -\pi h_p$

If we reconsider the final expressions for the potential and its first derivatives and insert the latter correction terms for each plane, we get the following terms:

$$V = \frac{G\rho}{2} \sum_{p=1}^{n} \sigma_p h_p \left[\sum_{q=1}^{m} \sigma_{pq} h_{pq} LN_{pq} + h_p \sum_{q=1}^{m} \sigma_{pq} AN_{pq} + \mathcal{SING}_p \right] \tag{2.5.16}$$

$$V_i = G\rho \sum_{p=1}^{n} \cos\left(\boldsymbol{N}_p, \boldsymbol{e}_i\right) \left[\sum_{q=1}^{m} \sigma_{pq} h_{pq} LN_{pq} + h_p \sum_{q=1}^{m} \sigma_{pq} AN_{pq} + \mathcal{SING}_p \right] \tag{2.5.17}$$

Note that $\mathcal{SING}_p$ equals zero if P' is located outside S_p.

Computational aspects and validation

3

In the last chapter, a method was derived to compute the potential and its first derivatives of an arbitrary homogeneous polyhedron. This chapter will deal with the question how to implement the algorithm in a computer program. Furthermore, a program, which was written during this work, is validated with similar tools.

3.1 Implementation of the algorithm

First of all, most following considerations are of geometric nature. This means that it will be very important to adhere certain rules for a successful computation of the desired quantities. This concerns especially the order of the vertices. It must be ensured that the normals always point outward of the polyhedron. This can be achieved if the vertices are arranged counterclockwise if seen from the outside of the body.

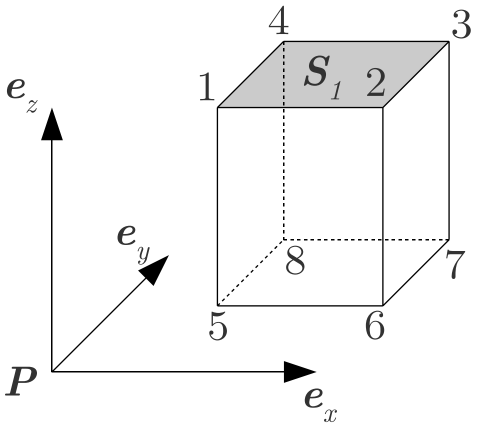


Figure 3.1: Example of a polyhedron which is used to explain the computation process

For the following steps a slightly different notation will be used:

- $\boldsymbol{X}_P \rightarrow$ coordinates of the computation point, i.e. $\boldsymbol{X}_P = \begin{bmatrix} x_P & y_P & z_P \end{bmatrix}$

- $\boldsymbol{X}$ → coordinates of the vertices of a polyhedron, i.e $\boldsymbol{X} = \begin{bmatrix} x_1 & y_1 & z_1 \\ x_2 & y_2 & z_2 \\ & \cdots & \\ x_n & y_n & z_n \end{bmatrix}$
- nop → Number of planes of the polyhedron (e.g. for the body in Fig. 3.1 $nop = 6$)
- nov → Number of vertices of a specific plane (e.g. for the plane S_1 in Fig. 3.1 $nov = 4$)

The index i will be used for the single planes (faces) of the body, i.e. $i = 1, ..., nop$. The index j will be used for the vertices of a plane, i.e. $j = 1, ..., nov$.

1. Preparations: The presented method assumes that the computation point P is the origin of a local coordinate system. Thus, the coordinates of the vertices must be reduced to this origin:

$$\tilde{\boldsymbol{X}} = \boldsymbol{X} - \boldsymbol{X}_p \tag{3.1.1}$$

2. Building the line segments $\boldsymbol{G}_{i,j}$: These segments are the vectors which connect the single vertices j of the face i (see Fig. 3.2). Thus, they can be computed by

$$\boldsymbol{G}_{i,j} = \tilde{\boldsymbol{X}}_{i,j+1} - \tilde{\boldsymbol{X}}_{i,j} \tag{3.1.2}$$

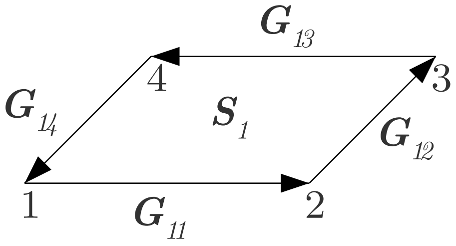


Figure 3.2: Computation of the line segments $\boldsymbol{G}_{pq}$

3. Building the plane unit normals $\boldsymbol{N}_i$: The plane unit normals can be computed by building the cross product of two line segments (see Fig. 3.3). The resulting vector is perpendicular to both segments and points outward

$$\boldsymbol{N}_i = \frac{\boldsymbol{G}_{i,1} \times \boldsymbol{G}_{i,2}}{|\boldsymbol{G}_{i,1}|\,|\boldsymbol{G}_{i,2}|} \tag{3.1.3}$$

4. Building the segment unit normals $\boldsymbol{n}_{i,j}$: The segment unit normal of a line segment (see Fig. 3.4) can be computed by building the cross product of the line segment $\boldsymbol{G}_{i,j}$ itself and the plane unit normal:

$$\boldsymbol{n}_{i,j} = \frac{\boldsymbol{G}_{i,j} \times \boldsymbol{N}_i}{|\boldsymbol{G}_{i,j}|\,|\boldsymbol{N}_i|} \tag{3.1.4}$$

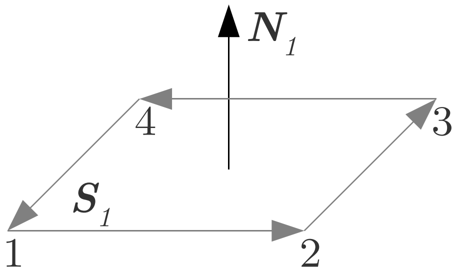


Figure 3.3: Computation of the surface unit normals $\boldsymbol{N}_i$

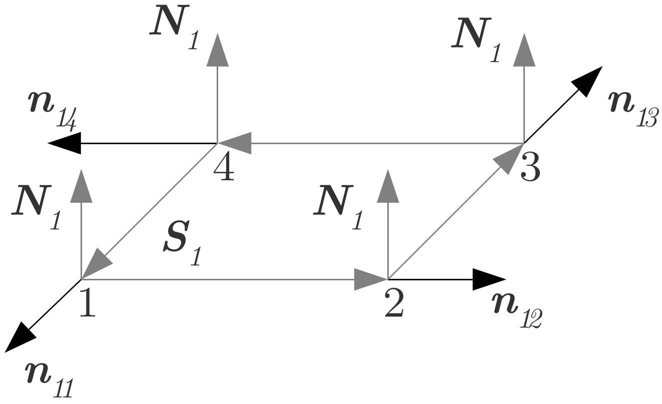


Figure 3.4: Computation of the segment unit normals $\boldsymbol{n}_{1,j}$ of the face S_1

5. Orientation of the plane unit normals: In equations (2.5.16) and (2.5.17) the parameter σ_i is -1, 1 or 0, depending on the orientation of the plane. If we evaluate the dot product of the plane unit normal vector and the *negative* position vector of a vertex, we get the projection of the position vector on the normal vector as $|\boldsymbol{N}_i| = 1$. This means that the result is negative if $\boldsymbol{N}_i$ and $\tilde{\boldsymbol{X}}_{i,1}$ point towards the same half-space. If both vectors point to different half-spaces, the result will be positive. A third case appears if $\boldsymbol{N}_i$ and $\tilde{\boldsymbol{X}}_{i,1}$ are perpendicular. Then, the dot product equals zero and the concerned plane is parallel to one of the three coordinate planes.

$$\boldsymbol{N}_i \cdot \left(-\tilde{\boldsymbol{X}}_{i,1}\right) \begin{cases} > 0, & \sigma_i = -1 \\ < 0, & \sigma_i = 1 \\ = 0, & \sigma_i = 0 \end{cases} \tag{3.1.5}$$

   Here, $\tilde{\boldsymbol{X}}_{i,1}$ is the position vector of one (arbitrary) vertex of a specific face S_i as the sign of the dot product is the same no matter which vertex is chosen.

6. Projection P_i' of P onto the plane containing S_i: This step needs a closer look as there are multiple methods how one can perform such a projection (see Fig. 3.5). In the following, an approach is discussed which is easy to compute.
   a) Determination of the parameters of a plane: A plane can be defined by three points, which are located in this plane. Here, we are dealing with faces, which are defined through their vertices. These points can be used to compute the parameters of the corresponding plane. We have already computed the plane

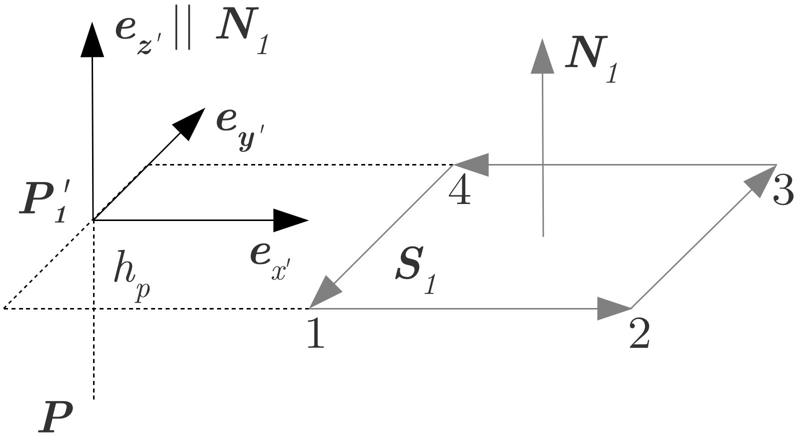


Figure 3.5: Projection P_1' of P onto the plane which contains the face S_1

unit normals $\boldsymbol{N}_i$ of the faces. The equation of a plane in the so-called Hessian normal form reads as

$$ax + by + cz + d = 0 \tag{3.1.6}$$

where a, b, c are the elements of the plane normal, i.e. $\boldsymbol{N}_i = \begin{bmatrix} a & b & c \end{bmatrix}^{\mathsf{T}}$, d is a constant factor and x, y, z the coordinates of a point on that plane. Thus, equation (3.1.6) can be written in vector notation:

$$\boldsymbol{N}_i \cdot \boldsymbol{X}_{i,1} + d = 0 \Rightarrow d = -\boldsymbol{N}_i \cdot \boldsymbol{X}_{i,1} \tag{3.1.7}$$

where $\tilde{\boldsymbol{X}}_{i,1}$ is again the position vector of an arbitrary vertex of the face S_p.

b) Computation of the distance h_i: The distance of a point from a plane in Hessian normal form is usually computed by

$$h = \frac{ax + by + cz + d}{\sqrt{a^2 + b^2 + c^2}}, \tag{3.1.8}$$

where x, y, z are the coordinates of the point of interest. In our case, this would be the computation point P and thus $x = y = z = 0$ as the coordinates of all vertices were reduced to that origin. Thus, the distance of a face S_i from the computation point P is given by

$$h_i = \frac{d}{|\boldsymbol{N}_i|}. \tag{3.1.9}$$

c) Perfoming the projection: We have already computed the parameter σ_i, which depends on the orientation of the face S_i. Thus, the coordinates of the projected point P_i' can be computed through

$$\boldsymbol{X}_{P_i'} = \sigma_i h_i \boldsymbol{N}_i. \tag{3.1.10}$$

7. Orientation of the segment unit normals $\boldsymbol{n}_{i,j}$: This step is very similar to the orientation of the plane unit normals. Again, we are only interested whether the segment unit normal $\boldsymbol{n}_{i,j}$ points towards the half-space containing the *projected*

computation point P'_i or not. Therefore, we first subtract the coordinates of P' from the coordinates of the vertices, i.e. P'_i is the new origin of a local coordinate system (see Fig. 3.5). Now, the dot product of the segment unit normal and the (transformed) position vector is the projection of $\boldsymbol{n}_{i,j}$ onto the length of the position vector. This means that the result will be negative if $\boldsymbol{n}_{i,j}$ points towards the half-space containing P'_i and positive if not. Again, The third case, i.e. the dot product equals zero, can only appear if both vectors are perpendicular which means that P'_i is located *on* the line segment i, j.

$$\boldsymbol{n}_{i,j} \cdot \left(\boldsymbol{X}_{P'_i} - \tilde{\boldsymbol{X}}_{i,j} \right) \begin{cases} > 0, & \sigma_{i,j} = -1 \\ < 0, & \sigma_{i,j} = 1 \\ = 0, & \sigma_{i,j} = 0 \end{cases} \tag{3.1.11}$$

8. Projection $P''_{i,j}$ of P'_i on the line segments $\boldsymbol{G}_{i,j}$: First of all, it must be checked if P'_i and $P''_{i,j}$ are identical since the presented method does not work in this case.

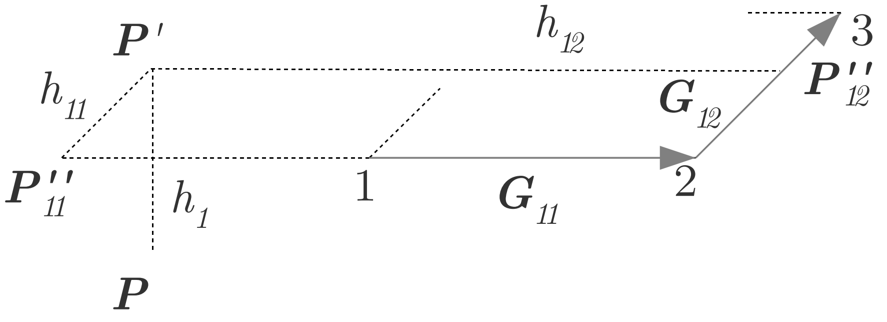


Figure 3.6: Projection pf P onto a plane (P') and a line segment (P'')

This can be ensured by looking at the orientation of the segment unit normals. If $\sigma_{i,j} = 0$ for a specific line segment i, j, the (transformed) position vector of the vertex and the segment unit normal are perpendicular, which means that P'_i is located on that line segment and $P''_{i,j} = P'_i$. For all other cases, we will solve a linear system to compute the coordinates of $P''_{i,j}$.

For the sake of simplicity, a different notation will be used for this step:

- $\boldsymbol{x}' \rightarrow$ position vector of P'_i
- $\boldsymbol{x}'' \rightarrow$ unknown position vector of $P''_{i,j}$,
- $\boldsymbol{r} \rightarrow$ vector of the line segment i, j, i.e. $\boldsymbol{r} = \boldsymbol{G}_{i,j}$,
- $\boldsymbol{e} \rightarrow$ vector which connects P'_i with one of the two vertices of $\boldsymbol{G}_{i,j}$
- $\boldsymbol{x}_1 \rightarrow$ position vector of this vertex

See Fig. 3.7 for the geometric definition of the elements. The coordinates of $P''_{i,j}$ can be computed by solving a linear system. We use three relations which must hold for the position of $P''_{i,j}$:

a) $\boldsymbol{r}$ is perpendicular to both vectors $\boldsymbol{x}''$ and $\boldsymbol{x}'$

$$\boldsymbol{r} \cdot \boldsymbol{x}'' = \boldsymbol{r} \cdot \boldsymbol{x}' \tag{3.1.12}$$

b) $\boldsymbol{x}'$, $\boldsymbol{x}''$ and $\boldsymbol{r}$ are co-planar

$$(\boldsymbol{r} \times \boldsymbol{e}) \cdot \boldsymbol{x}'' = (\boldsymbol{r} \times \boldsymbol{e}) \cdot \boldsymbol{x}' \tag{3.1.13}$$

c) $\boldsymbol{x}''$ and $\boldsymbol{r}$ are co-linear

$$((\boldsymbol{r} \times \boldsymbol{e}) \times \boldsymbol{r}) \cdot \boldsymbol{x}'' = ((\boldsymbol{r} \times \boldsymbol{e}) \times \boldsymbol{r}) \cdot \boldsymbol{x}' \tag{3.1.14}$$

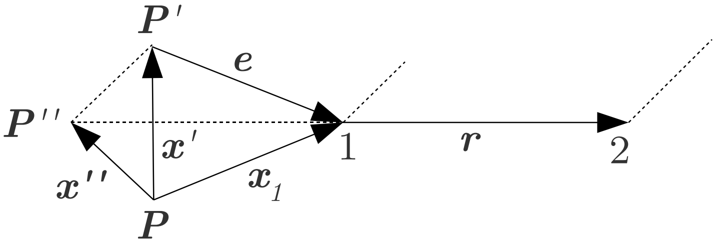


Figure 3.7: Elements of the linear system to compute the coordinates of P''

With these properties of $P''_{i,j}$, we can set up a linear system to compute the unknown coordinates. For the sake of simplicity, the following abbreviations will be used:

$$\begin{aligned} \boldsymbol{r} = \boldsymbol{a} &= \begin{bmatrix} a_1 & a_2 & a_3 \end{bmatrix}^{\mathsf{T}} \\ \boldsymbol{r} \times \boldsymbol{e} = \boldsymbol{b} &= \begin{bmatrix} b_1 & b_2 & b_3 \end{bmatrix}^{\mathsf{T}} \\ (\boldsymbol{r} \times \boldsymbol{e}) \times \boldsymbol{r} = \boldsymbol{c} &= \begin{bmatrix} c_1 & c_2 & c_3 \end{bmatrix}^{\mathsf{T}} \end{aligned}$$

The quantities which are needed to compute the coordinates of $P''_{i,j}$ are

$$\boldsymbol{A} = \begin{bmatrix} a_1 & a_2 & a_3 \\ b_1 & b_2 & b_3 \\ c_1 & c_2 & c_3 \end{bmatrix}; \quad \boldsymbol{x}'' = \begin{bmatrix} x_{P''_{i,j}} \\ y_{P''_{i,j}} \\ z_{P''_{i,j}} \end{bmatrix} = \boldsymbol{X}_{P''_{i,j}}; \quad \boldsymbol{y} = \begin{bmatrix} \boldsymbol{a} \cdot \boldsymbol{x}' \\ \boldsymbol{b} \cdot \boldsymbol{x}' \\ \boldsymbol{c} \cdot \boldsymbol{x}' \end{bmatrix}$$

As there are three unknowns and three equations, we can compute the position of $P''_{i,j}$ directly via

$$\boldsymbol{x}'' = \boldsymbol{A}^{-1}\boldsymbol{y} \tag{3.1.15}$$

$\boldsymbol{A}$ becomes rang deficient if $P'_i = P''_{i,j}$. Therefore, the condition $\sigma_{i,j} = 0$ must be checked before inversion. Otherwise, one could compute the inverse of this 3 × 3 matrix with Cramer's algorithm (see e.g. Bronstein et al. (2000)) and perform the check afterwards.

9. Computation of $LN_{i,j}$ and $AN_{i,j}$: These coefficients both depend on the distance of the end-points of a line segment and $P''_{i,j}$ which can be computed by

$$s_{1_{i,j}} = \left| \boldsymbol{P}'_i - \tilde{\boldsymbol{X}}_{1_{i,j}} \right|; \quad s_{2_{i,j}} = \left| \boldsymbol{P}'_i - \tilde{\boldsymbol{X}}_{2_{i,j}} \right| \tag{3.1.16}$$

where $\tilde{\boldsymbol{X}}_{1_{i,j}}$ and $\tilde{\boldsymbol{X}}_{2_{i,j}}$ are the two end-points of the line segment $\boldsymbol{G}_{i,j}$. Additionally, one needs the distances of the segment's vertices from the computation point P

$$l_{1_{i,j}} = \left|\tilde{\boldsymbol{X}}_{1_{i,j}}\right|; \quad l_{2_{i,j}} = \left|\tilde{\boldsymbol{X}}_{2_{i,j}}\right| \tag{3.1.17}$$

and the length of a line segment

$$g_{i,j} = |\boldsymbol{G}_{i,j}| \tag{3.1.18}$$

As the integration limits are the coordinates of the end-points of a line segment in a one dimensional coordinate system, the relative position of these points to the origin, i.e. $P''_{i,j}$ is needed. Therefore, the sign of this position must be computed correctly, which depends on the location of P, P'_i and $P''_{i,j}$ and the corresponding line segment $\boldsymbol{G}_{i,j}$. This can be achieved if the latter quantities $s_{1_{i,j}}$, $s_{2_{i,j}}$, $l_{1_{i,j}}$, $l_{2_{i,j}}$ and $g_{i,j}$ are compared. Thus, the correct sing can be computed by considering the following cases:

- $P''_{i,j} = P$ and P inside $\boldsymbol{G}_{i,j}$:

  $$s_{1_{i,j}} - l_{1_{i,j}} = s_{2_{i,j}} - l_{2_{i,j}} = 0, \quad s_{1_{i,j}} + s_{2_{i,j}} = g_{i,j}$$

  $P''_{i,j}$ and P have the same coordinates and are both located *inside* the line segment $\boldsymbol{G}_{i,j}$. Thus, $s_{1_{i,j}}$ and $l_{1_{i,j}}$ must be multiplicated with -1 as the coordinates (in the one dimensional coordinate system) of the first vertex must be negative.
- $P''_{i,j} = P$ and P outside $\boldsymbol{G}_{i,j}$:

  $$s_{1_{i,j}} - l_{1_{i,j}} = s_{2_{i,j}} - l_{2_{i,j}} = 0, \quad s_{1_{i,j}} + s_{2_{i,j}} > g_{i,j}$$

  $P''_{i,j}$ and P have the same coordinates and are both located *outside* the line segment $\boldsymbol{G}_{i,j}$.
  - $P''_{i,j}$ and P left of $\boldsymbol{G}_{i,j}$:

    $$s_{1_{i,j}} < s_{2_{i,j}}$$

    Both vertices are left of the origin, i.e $s_{1_{i,j}}$, $l_{1_{i,j}}$, $s_{2_{i,j}}$ and $l_{2_{i,j}}$ must be multiplicated with -1
  - $P''_{i,j}$ and P right of $\boldsymbol{G}_{i,j}$

    $$s_{1_{i,j}} > s_{2_{i,j}}$$

    Both vertices are right of the origin, i.e $s_{1_{i,j}}, l_{1_{i,j}}, s_{2_{i,j}}, l_{2_{i,j}} > 0$.
- $P''_{i,j} \neq P$ and $P''_{i,j}$ inside $\boldsymbol{G}_{i,j}$:

  $$s_{1_{i,j}} \neq l_{1_{i,j}}, \quad s_{2_{i,j}} \neq l_{2_{i,j}}, \quad s_{1_{i,j}} + s_{2_{i,j}} < g_{i,j}$$

  P and $P''_{i,j}$ do not have identical coordinates and $P''_{i,j}$ is located *inside* the line segment $\boldsymbol{G}_{i,j}$. Thus, the coordinate (in the one dimensional coordinate system) of the first vertex is negative, i.e $s_{1_{i,j}}$ must be multiplicated with -1.
- $P''_{i,j} \neq P$ and $P''_{i,j}$ outside $\boldsymbol{G}_{i,j}$:

  $$s_{1_{i,j}} \neq l_{1_{i,j}}, \quad s_{2_{i,j}} \neq l_{2_{i,j}}, \quad s_{1_{i,j}} + s_{2_{i,j}} > g_{i,j}$$

  P and $P''_{i,j}$ do not have identical coordinates and $P''_{i,j}$ is located *outside* the line segment $\boldsymbol{G}_{i,j}$.

  - $P''_{i,j}$ left of $\boldsymbol{G}_{i,j}$:

    $$s_{1_{i,j}} < s_{2_{i,j}}$$

    Both vertices are left" of the origin, i.e $s_{1_{i,j}}$ and $s_{2_{i,j}}$ must be multiplicated with -1
  - $P''_{i,j}$ "right" of $\boldsymbol{G}_{i,j}$

    $$s_{1_{i,j}} > s_{2_{i,j}}$$

    Both vertices are right of the origin, i.e $s_{1_{i,j}}, s_{2_{i,j}} > 0$.

The upper conditions are given in such way that they can be easily implemented in a program. Now, one can compute the coefficients $LN_{i,j}$ and $AN_{i,j}$ according to equations (2.4.12) and (2.4.13) in section 2.4.

10. Treatment of possible singularities: Finally, it must be checked if one of the singularities described in section 2.5 appears. Therefore, we will have to compute some auxiliary variables:

$$g_{1_{i,j}} = \left| \boldsymbol{P}'_i - \tilde{\boldsymbol{X}}_{i,j+1} \right| \tag{3.1.19}$$

$$g_{2_{i,j}} = \left| \boldsymbol{P}'_i - \tilde{\boldsymbol{X}}_{i,j} \right| \tag{3.1.20}$$

Depending on these quantities, the length of a line segment $g_{i,j}$ and the parameter $\sigma_{i,j}$ (orientation of the segment unit normal $\boldsymbol{n}_{i,j}$), the three singular cases are discussed in the following:

- P'_i is located inside S_i: If $\sigma_{i,j} = 1$ for $j = 1, ..., nov$, all segment unit normals of the face S_i point away from P'_i. This only happens if P'_i is located inside S_i, i.e, $\mathcal{SING}_i = -2\pi h_i$
- P'_i is located on a vertex: If $\sigma_{i,j} = 0$ and $g_{1_{i,j}} = 0$, P'_i is located on the second (end) vertex of $\boldsymbol{G}_{i,j}$ and thus

  $$\vartheta = \arctan \frac{\boldsymbol{G}_{i,j} \cdot (-\boldsymbol{G}_{i,j+1})}{|\boldsymbol{G}_{i,j}| \, |\boldsymbol{G}_{i,j+1}|} \tag{3.1.21}$$

  If $\sigma_{i,j} = 0$ and $g_{2_{i,j}} = 0$, P'_i is located on the first (start) vertex of $\boldsymbol{G}_{i,j}$ and thus

  $$\vartheta = \arctan \frac{\boldsymbol{G}_{i,j-1} \cdot (-\boldsymbol{G}_{i,j})}{|\boldsymbol{G}_{i,j-1}| \, |\boldsymbol{G}_{i,j}|} \tag{3.1.22}$$

  In both latter cases, the correction term would be $\mathcal{SING}_i = -\vartheta h_i$. However, if the dot product in the numerator equals zero, the successive line segments are perpendicular and $\mathcal{SING}_i = -\frac{\pi}{2} h_i$.
- P'_i is located on a line segment: If $\sigma_{i,j} = 0$, $g_{1_{i,j}} < g_{i,j}$ and $g_{2_{i,j}} < g_{i,j}$, P'_i must be located inside the line segment $\boldsymbol{G}_{i,j}$ since in all other cases this condition would not be fulfilled, i.e. $\mathcal{SING}_i = -\pi h_i$.

3.2 Data handling

An algorithm which can be easily implemented was described in the last section. However, one needs a proper input data format. In this thesis, the $x-$, $y-$ and $z-$coordinates of the vertices are stored in a first matrix. A second matrix contains the indices of the vertices which belong to one surface, and thus the topographical information. Each row defines one face of the body. A third matrix contains the $x-$, $y-$ and $z-$coordinates of the computation point(s). This becomes clear if a small example is considered. In the next section, a simple cube is used for the validation of the presented method (see Fig. 3.8). This cube has its center in the origin of a local coordinate system and a dimension of 2 x 2 x 2 m. Thus, the vertex- and topography-matrix could be set up in the following way:

$$\boldsymbol{X} = \begin{bmatrix} -1 & -1 & 1 \\ 1 & -1 & 1 \\ 1 & 1 & 1 \\ -1 & 1 & 1 \\ -1 & -1 & -1 \\ 1 & -1 & -1 \\ 1 & 1 & -1 \\ -1 & 1 & -1 \end{bmatrix} \qquad \boldsymbol{topo} = \begin{bmatrix} 1 & 2 & 3 & 4 \\ 5 & 8 & 7 & 6 \\ 1 & 5 & 6 & 2 \\ 2 & 6 & 7 & 3 \\ 3 & 7 & 8 & 4 \\ 4 & 8 & 5 & 1 \end{bmatrix} \tag{3.2.1}$$

Note that the order of the vertices is arbitrary, but in the topography-matrix, the vertices must be arranged counterclockwise from an outside view. This ensures that the normal vectors always point outside.

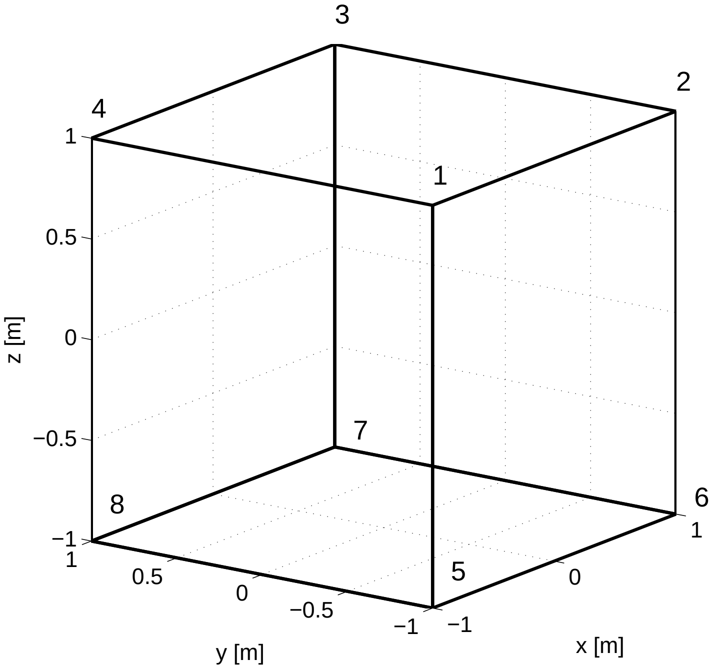


Figure 3.8: Example of a cube and its vertices

3.3 Validation

The described algorithm was implemented in the `Matlab`-program `polygrav.m` which was written for this thesis. For its validation, various tests are performed and the results are compared with two similar programs:

- `accrrp.m` (author: Dr. Matthias Weigelt, Geodetic Institute, University of Stuttgart): The program calculates the acceleration of a right rectangular prism, i.e., the first derivatives towards the $x-$, $y-$ and $z-$direction of the potential of the prism.
- `grvmag3d.m` (author: Singh & Guptasarama (2001)): Program for simultaneous computation of gravity and magnetic fields from a 3-D polyhedron.

All tests were performed on a Linux-machine with Ubuntu 9.04 and Matlab 2009b, a 2.00 Ghz Intel Core2 Duo CPU and 4 Gb RAM.

3.3.1 Simple body, point-wise validation

First, the acceleration of a simple cube is computed with all three programs. The main focus will be on the singularities, i.e. when the computation point is located on an edge or vertex of the cube. Therefore, this cube has its center in the origin of a local coordinate system and a dimension of 2 x 2 x 2 m. The density will be 2670 $^{kg}/_{m^3}$, which will be used for all bodies in this work. For the test, the acceleration of points on a vertex, edge, surface and in and outside of the body are computed. The coordinates of the vertices ($\boldsymbol{X}$) and the computation points ($\boldsymbol{X}_p$) (see Fig. 3.9) and the topography matrix (***topo***) are defined in such way, that the following values can be easily recomputed as they can be directly used as input parameters for `polygrav.m`. Thus, they are set up in the following way:

$$\boldsymbol{X} = \begin{bmatrix} -1 & -1 & 1 \\ 1 & -1 & 1 \\ 1 & 1 & 1 \\ -1 & 1 & 1 \\ -1 & -1 & -1 \\ 1 & -1 & -1 \\ 1 & 1 & -1 \\ -1 & 1 & -1 \end{bmatrix} \qquad \boldsymbol{topo} = \begin{bmatrix} 1 & 2 & 3 & 4 \\ 5 & 8 & 7 & 6 \\ 1 & 5 & 6 & 2 \\ 2 & 6 & 7 & 3 \\ 3 & 7 & 8 & 4 \\ 4 & 8 & 5 & 1 \end{bmatrix} \qquad \boldsymbol{X}_p = \begin{bmatrix} -1 & -1 & -1 \\ 0 & -1 & 1 \\ 0 & -1 & 0 \\ 0 & 0 & 0 \\ 0 & 2 & 1.5 \end{bmatrix}$$

The differences between the acceleration computed by the three programs are shown in Tab. 3.1. The differences between the single results have a magnitude of less than 10^{-16} mGal, which means that they might be caused by the limiting accuracy of double values and are thus negligible. Furthermore, these deviations are far off the the reading accuracy of a gravimeter, which is e.g. about 1 μGal for the Scintrex CG-5. Thus, this validation was successful as all three programs assume identical accelerations in the computation points.

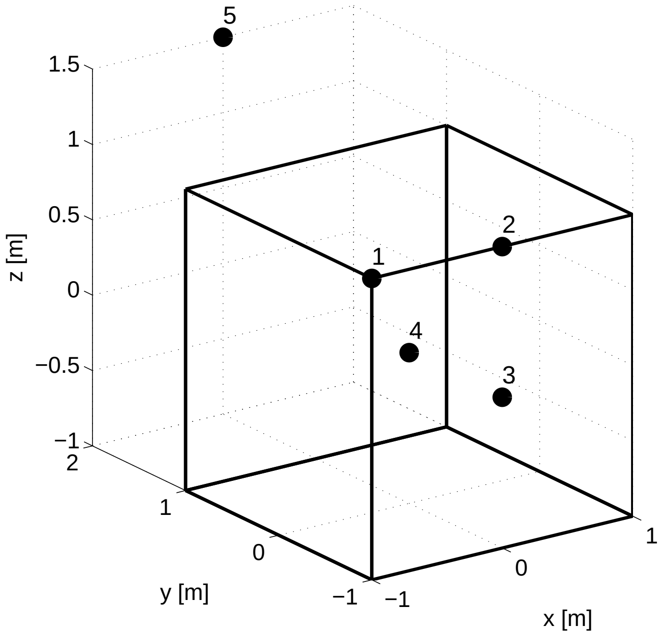


Figure 3.9: Location of the computation points for point-wise validation

Difference	**Point Nr.**	$\Delta \boldsymbol{V}_x$	$\Delta \boldsymbol{V}_y$	$\Delta \boldsymbol{V}_z$
	1	$3.2 \cdot 10^{-17}$	$3.2 \cdot 10^{-17}$	$3.2 \cdot 10^{-17}$
`polygrav.m`	2	0	$-9.2 \cdot 10^{-18}$	$-1.9 \cdot 10^{-17}$
-	3	$-7.9 \cdot 10^{-18}$	$5.6 \cdot 10^{-17}$	0
`accrrp.m`	4	$-4.0 \cdot 10^{-18}$	$-5.9 \cdot 10^{-18}$	0
	5	$2.0 \cdot 10^{-17}$	$-1.4 \cdot 10^{-17}$	$9.3 \cdot 10^{-18}$
	1	$-1.4 \cdot 10^{-17}$	$-1.4 \cdot 10^{-17}$	$-1.4 \cdot 10^{-17}$
`polygrav.m`	2	0	$9.2 \cdot 10^{-18}$	$-9.2 \cdot 10^{-18}$
-	3	$-7.9 \cdot 10^{-18}$	$5.6 \cdot 10^{-17}$	0
`gravmag3d.m`	4	0	0	0
	5	$2.4 \cdot 10^{-17}$	$-2.3 \cdot 10^{-18}$	$-4.9 \cdot 10^{-17}$

Table 3.1: Difference of the computed acceleration between `polygrav.m` and `accrrp.m` and `gravmag3d.m` respectively in [mGal]

3.3.2 Simple body, line-wise validation

The next step is to analyze a line of computation points above, top of and inside the body. Therefore, three lines were defined, where the $x-$ values go from -2 to +2 (see Fig. 3.10) and the $z-$value is 1.5, 1 and 0.5 respectively. The results for further lines are shown in the Appendix B.

The $x-$, $y-$ and $z-$component of the attraction in each single point of the three lines is computed with all three programs. The results from `polygrav.m` are shown in Fig. 3.11. These will be the reference for this validation. The $x-$component of the attraction is decreasing until the line approaches the body ($x = -1$). Inside the body, the signal increases up to a value of 0.02 mGal when the line reaches the right face of the

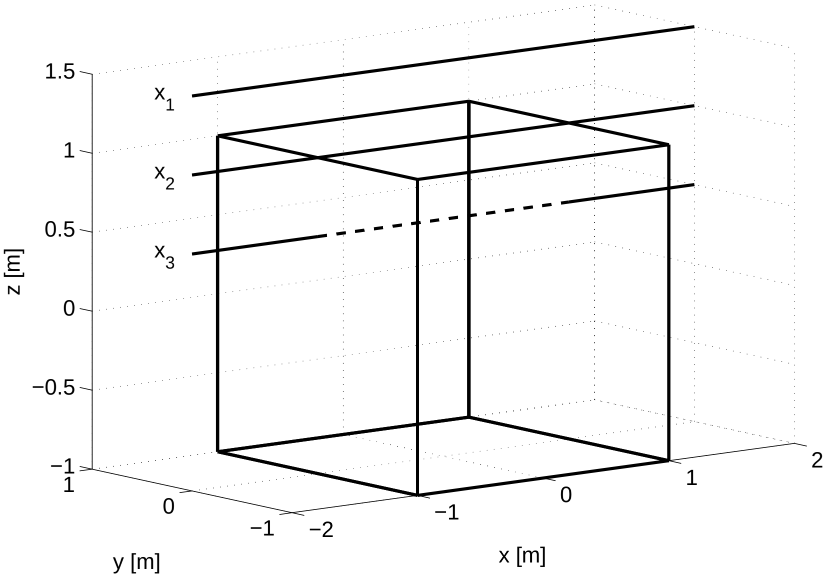


Figure 3.10: Location of three lines parallel to the $x-$axis with varying $z-$component along which the acceleration is computed

cube ($x = 1$), after which it decreases again. This characteristic is reasonable if we assume that only the potential of the cube affect the computation lines. As the attraction always opposites the direction of the maximum potential increase (which in this case always points towards the cube), the attraction is negative (positive) for computation points left (right) of the cube. This holds as well for the $z-$component. As all lines are located above the center of the cube, the direction of the maximum potential increase points downwards. Thus, the attraction has to be positive and must increase by approaching the cube. As the $y-$values of all three lines are zero, the $y-$component of the attraction in all points has to be zero by theory, which agrees with the illustrated results.

The differences between the results from `polygrav.m` and `accrrp.m` (see Fig. 3.12) and `gravmag3d.m` (see Fig. 3.13) are considered. A magnitude of the deviations of about $5 \cdot 10^{-17}$ mGal in the plots agrees with the point-wise validation. Furthermore, the deviations showed in Fig. 3.12 can be assumed to be noise due to computational inaccuracies. But the lower left plot in Fig. 3.13 shows a negative drift. Thus, it can be concluded that both programs `polygrav.m` and `accrrp.m` produce similar results but differ slightly from `gravmag3d.m`.

These comparisons were performed for six further lines with varying $y-$ and $z-$component. The results are presented in the Appendix B and confirm the latter findings. The deviation between `polygrav.m` and `accrrp.m` seem to be pure noise. Thus, both programs are assumed to be of similar accuracy.

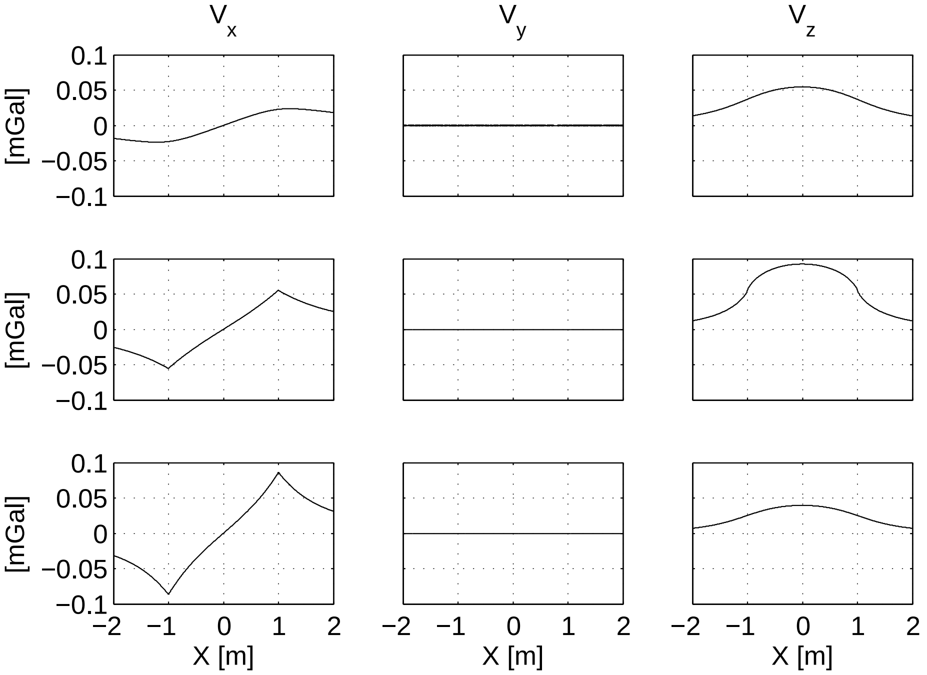


Figure 3.11: $x-$ (left column), $y-$ (center column) and $z-$ (right column) component of the acceleration along line 1 (top row), line 2 (middle row) and line 3 (bottom row)

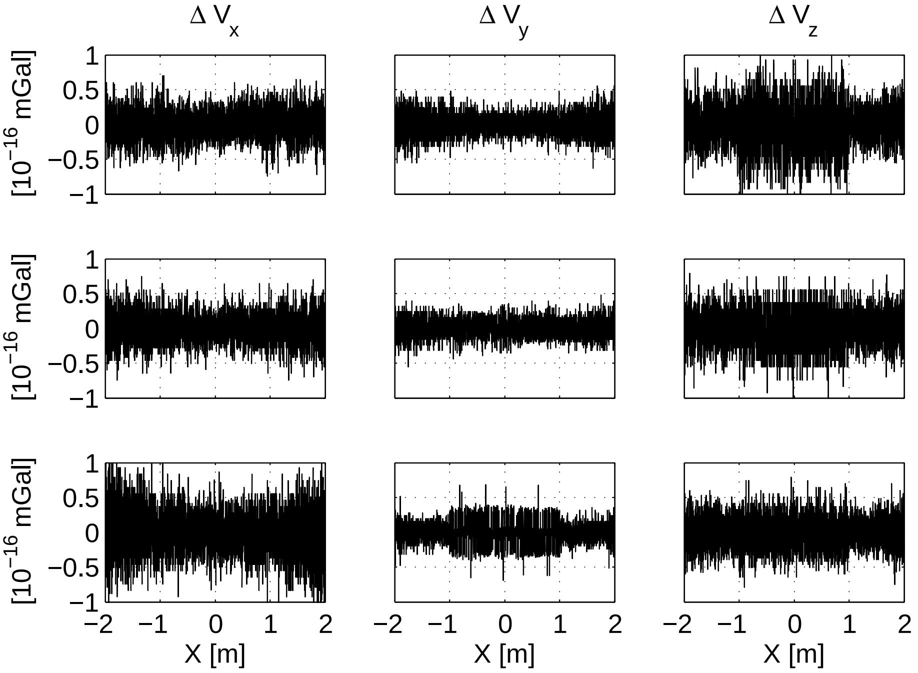


Figure 3.12: Difference between the acceleration computed with `polygrav.m` and `accrrp.m` along line 1 (top row), line 2 (middle row) and line 3 (bottom row)

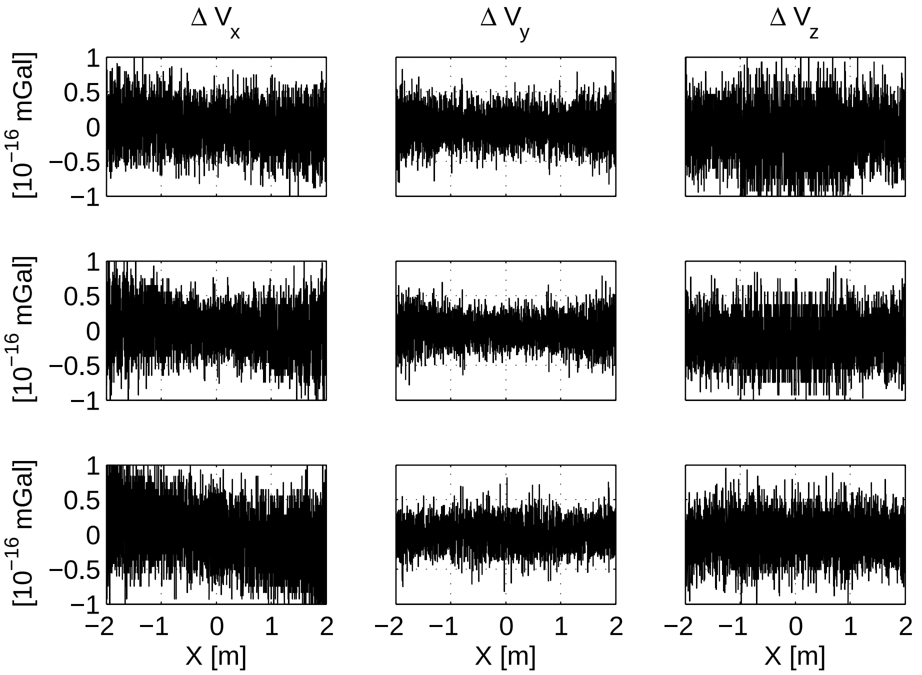


Figure 3.13: Difference between the acceleration computed with `polygrav.m` and `gravmag3d.m` along line 1 (top row), line 2 (middle row) and line 3 (bottom row)

3.3.3 Complex body, area-wise validation

Finally, a complex body is considered for the validation. This body is inspired by the stair-structure of a pyramid, which will be explained more clearly in the next chapter. Due to the intricate handling of such bodies with `gravmag3d.m` and a very long computation time, only `polygrav.m` and `accrrp.m` are used for validation.

The body is constructed in the following way: Its dimension in $x-$ and $y-$direction is 15 x 15 m and the height of the steps is successively increased by 1 m. The density of the whole body is 2670 $^{kg}/_{m^3}$. The computation points are located on the surface of the body with a distance between the points of 5 cm in both $x-$ and $y-$direction which makes a total of 90000 points. Now, the $x-$, $y-$ and $z-$component of the attraction in each computation point is computed with `polygrav.m` and `accrrp.m`. The results of `polygrav.m` are shown in Fig. 3.15, the difference between `polygrav.m` and `accrrp.m` is shown in Fig. 3.16. The statistics of the evaluation areas are presented in Tab. 3.2.

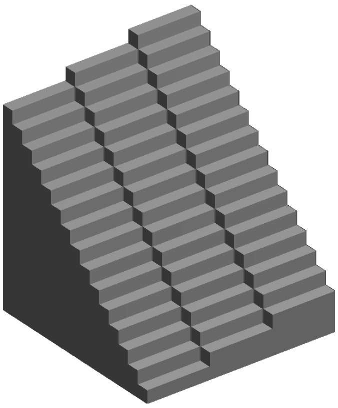


Figure 3.14: Complex body for validation

First of all, we see a similar characteristic compared to the line-wise validation. Both programs compute the attraction with a maximal deviation of about $\pm 2 \cdot$

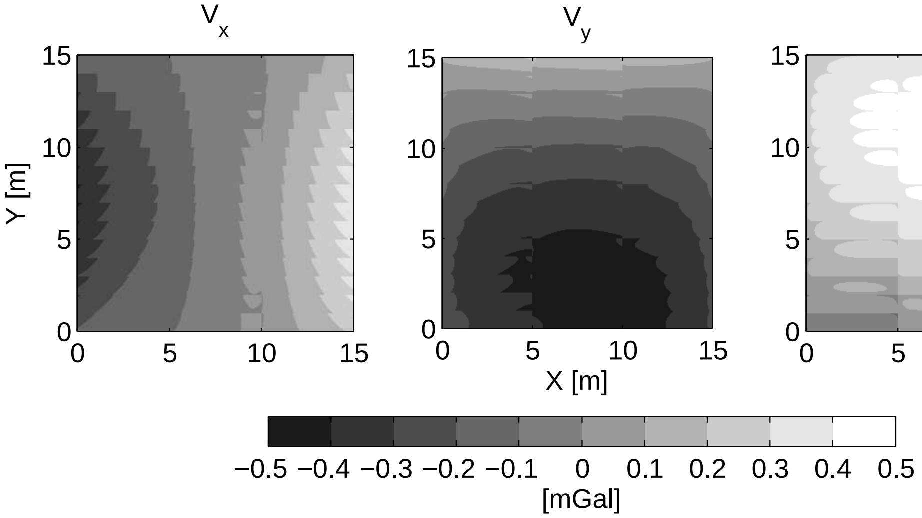


Figure 3.15: $x-$, $y-$ and $z-$component of the attraction in points on the surface of the body computed with `polygrav.m`

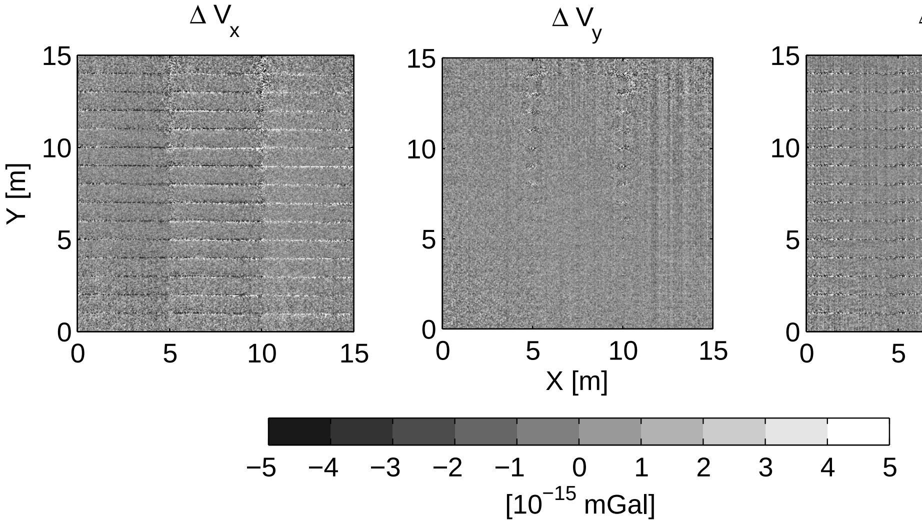


Figure 3.16: Difference of the attracion of the body between `polygrav.m` and `accrrp.m`

10^{-14} mGal, which was already considered as computational inaccuracies. However, the maximal deviations mostly occur on edges or vertices of the body while all other regions show smaller differences. This might come from the different computation principles but should not be taken too serious as the dimension of the deviations is far off the measurable magnitude. This holds as well for the other statistics. Even a comparison of the RMS-values shows that both programs produce equivalent results.

The latter findings confirm two assumptions. The presented approach of calculating the gravitational quantities of an arbitrary polyhedron were calculated by transforming the volume integral into line-integrals with a closed analytical solution, produces identical results compared to the closed expressions of a right rectangular prism. Furthermore, the values which were computed with `polygrav.m` (i.e. the program which was written for this work) are of identical accuracy compared to two programs which compute the same quantities but with different strategies. By considering this successful validation, `polygrav.m` can be used to compute the gravitational quantities of arbitrary bodies which can be defined in terms of faces and vertices.

Component	**mean**	**min**	**max**	**RMS**
		[mGal]		
ΔV_x	$2.0 \cdot 10^{-17}$	$2.3 \cdot 10^{-14}$	$2.3 \cdot 10^{-14}$	$1.8 \cdot 10^{-15}$
ΔV_y	$5.0 \cdot 10^{-17}$	$2.2 \cdot 10^{-14}$	$2.1 \cdot 10^{-14}$	$1.2 \cdot 10^{-15}$
ΔV_z	$-8.6 \cdot 10^{-17}$	$2.0 \cdot 10^{-14}$	$1.8 \cdot 10^{-14}$	$1.4 \cdot 10^{-15}$

Table 3.2: Statistics of the differences between the three components of the attraction derived from `polygrav.m` and `accrrp.m`

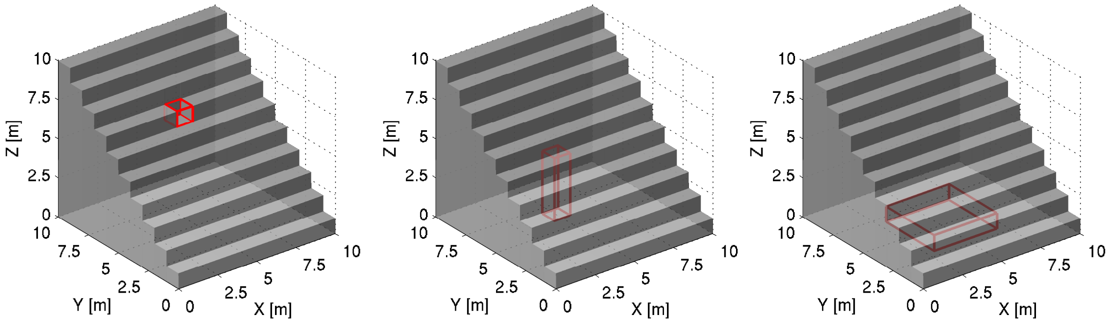
3.4 Case study: Signal strength of a varying mass distribution

As the functionality of the algorithm is validated, the next step is to analyze how changes in the inner mass distribution of a reference body affect the gravity signal on the surface. It should be mentioned that these experiments assume an accurate knowledge about the bodies surface. Furthermore, all exterior effects are ruled out and the computation points have a distance of 5 cm. The following results shall give an impression about the characteristics of the gravity signal of comparatively small bodies. Therefore, a reference body is created in the form of a stair model with a dimension of 10x10x10 m consisting of 10 steps with the height of 1 m and a density of 2670 $^{kg}/_{m^3}$. For this analysis, four different experiments are performed to evaluate the effect of a

- descending hole (Fig. 3.17a),
- contracting hole (Fig. 3.17a),
- growing empty column (Fig. 3.17b) and
- growing vacuty (Fig. 3.17c).

Thus, the case study shall give an impression about the signal characteristics of a mass deviation which is resized in one or more dimensions or translated in vertical direction. In all cases, the deviation is considered to be an empty space, i.e. to have a density equal zero. But it should be mentioned that the following findings hold only for the presented conditions. If other bodies, mass deviations with a varying density or dimension or multiple deviations are considered, the results might look different. However, a detailed analysis of all possible cases would go beyond the scope of this thesis, which will be emphasized by the following findings.

The presented graphics (Fig. 3.18 - 3.21) are titled with the volume of the deviation and the $z-$component (i.e. the height) h_m of the center of the hole. Every graphic shows the difference between the $z-$component of the attraction of a model without and with such mass deviations. The right graphics show the extent and location in $y-$ and $z-$dimension whereas the $x-$component of the holes center is always 5 m.



(a) Descending/contracting hole (b) Growing column (c) Growing vacuty

Figure 3.17: Location and extent of the removed elements in the reference body

Descending hole

For the beginning, the removal of a stone near the surface of the model is simulated (Fig. 3.17a). Afterwards, a hole in the body with the same volume as this removed stone is created in the inside. This hole is successively moved to the bottom of the model and each step, the acceleration on the surface is computed (see Fig. 3.18).

We see that the signal strength of the missing stone is decreasing the deeper it is located in the model. On the lowest step, it has such a small signal that it might not be detectable with a high-accuracy gravimeter. This means that an inner mass distribution must have a certain dimension, depending on its location, so it can be detected on the surface. Furthermore, it was assumed that there is a hole (i.e. a cuboid with a density equal zero) in the body. If this cuboid would be of a different material with a higher density, a detection on the surface might be even more difficult as the difference between the attraction of the bodies with and without a mass deviation gets smaller when the density of the deviation approaches the density of the surrounding body.

Contracting hole

The second experiment is the simulation of a contracting hole on the surface. This shall give an impression about the impact of a small deviation on the surface (e.g. a missing stone, etc.). Furthermore, this already indicates the required accuracy of a surface model which is transformed to a gravity model (see chapter 4 and 5). The hole has the form of a cube and its size (and thus its volume) is successively reduced. Each step, the acceleration on the surface is computed (see Fig. 3.19).

The first hole has a dimension of 50x50x50 cm (top graphics of Fig. 3.19). We see a point-shaped signal with a strength up to 24 μGal. This this means that the deviation is detectable, if we observe the location exactly above the hole. On the other hand, if the deviation was caused by a mis-modeled surface (i.e. the hole is not considered in a model), it already produces an erroneous signal which is significant. The next hole (middle graphic of Fig. 3.19) has a dimension of 35x35x35 cm. Even if the signal strength is lower, the deviation is still detectable by a gravimeter, if we assume a measuring accuracy of 4-5 μGal. However, this holds only for the location directly above the hole.

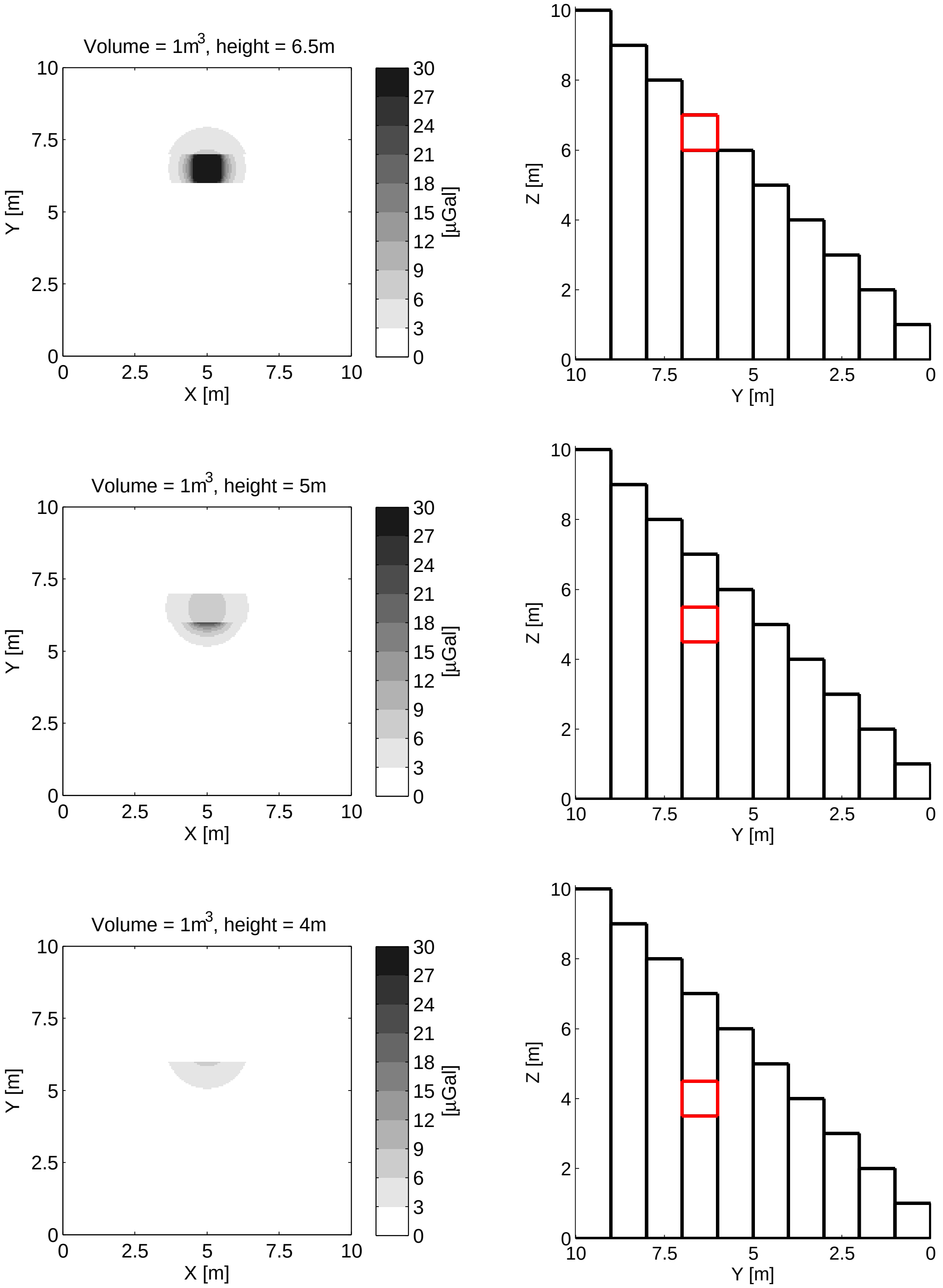
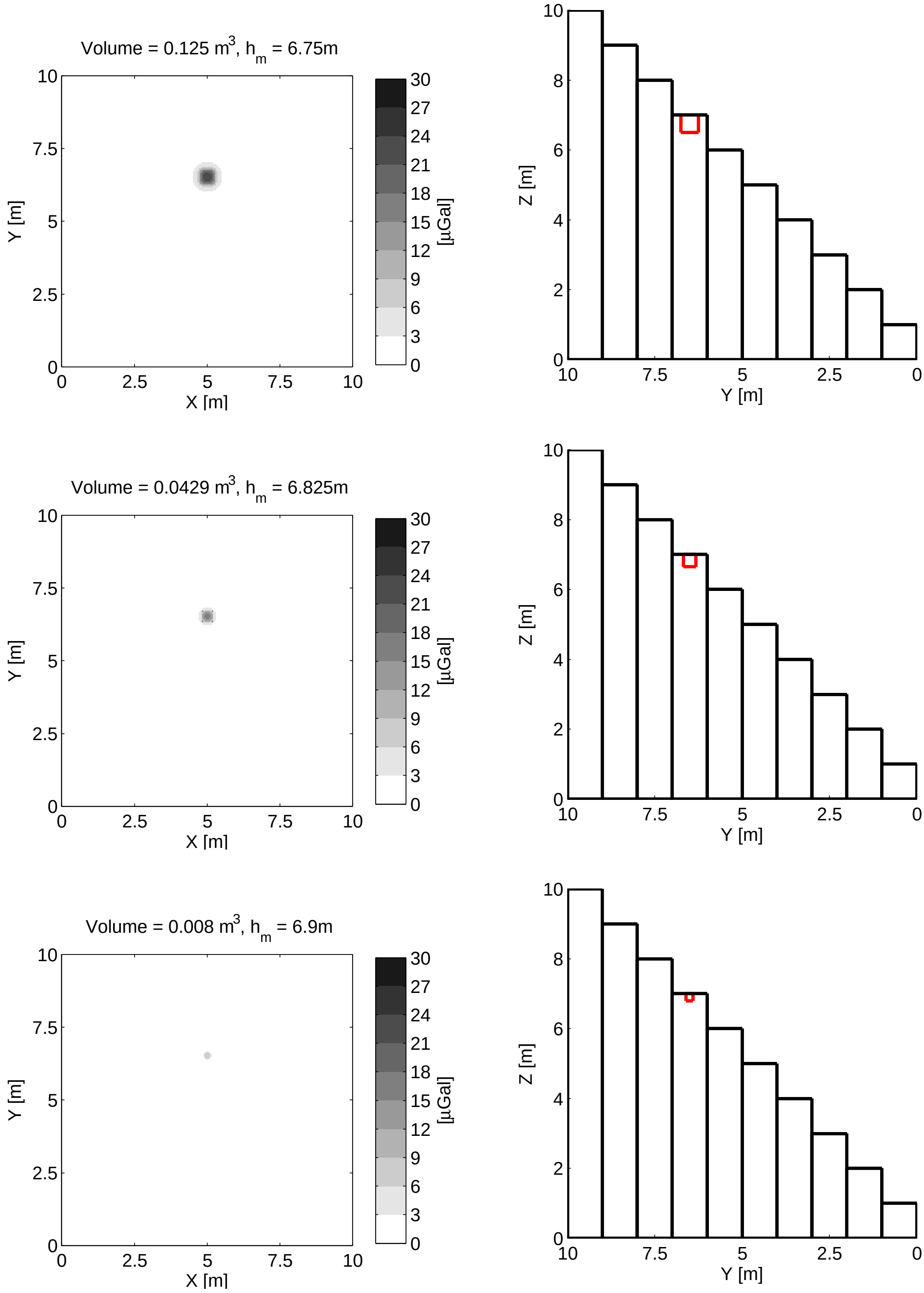


Figure 3.18: Signal of a successively descending hole on the attraction on computation points located on the surface of the stair model



Volume = 0.125 m^3, h$_m$ = 6.75m

Volume = 0.0429 m^3, h$_m$ = 6.825m

Volume = 0.008 m^3, h$_m$ = 6.9m

Figure 3.19: Signal of a successively contracting hole on the attraction on computation points located on the surface of the stair model

The third hole (bottom graphic of Fig. 3.19) considers a hole with a dimension of 20x20x20 cm. Only now we get a signal which might not be significant. However, if the dimension of this deviation is considered, this experiment shows that already small deviations on the surface of a body might produce a significant signal. In other words, we need detailed knowledge about the surface of a body as a mis-modeled surface might cause an erroneous signal, which leads to improper assumptions about the bodies mass distribution.

Growing column

Now, an empty column on the bottom of the model is assumed (see Fig. 3.17b). The volume is successively increased by increasing the dimension in the $z-$direction and keeping its dimension in $x-$ and $y-$direction constant (i.e. 1x1 m). Every step, the attraction of the body is computed on the surface of the stair model (see Fig. 3.20). The first graphic shows only a very small signal on the surface of the body, even if the column has a volume of $4\,\mathrm{m}^3$. The latter findings showed that on the surface a hole with a volume of $1\,\mathrm{m}^3$ has a much higher signal which is in the measuring tolerance of a gravimeter. This means that with increasing distance from the computation points, it gets even more difficult to detect and identify mass deviations. On the other hand, if the volume of the empty column is increased, the signal on the surface increases as well. By keeping in mind that the dimension in $x-$ and $y-$direction of the column is 1 m, the shape of the signal differs completely from the hole on the surface. The first signal was more or less concentrated on a very small area whereas the signal of the column shows a larger spread. Thus, even if the maximal signal strength is similar, such a column can be distinguished from a hole on the surface if this surface is perfectly known. Otherwise, the error in the computed gravity signal caused by a miss-modeled surface would superimpose the signal caused by inner mass deviations and thus denies reasonable conclusions about the inner mass distributions. This becomes even clearer if another experiment is performed.

Growing vacuity

Again, a hole on the bottom of the body is constructed (see Fig. 3.17c). Its volume is increased by successively increasing its dimension in the $x-$ and $y-$direction and keeping the $z-$coordinates constant (see Fig. 3.21). The first graphic shows that such deviations need a minimal volume so it can be detected on the surface. But in contrast to the empty column, the signal of such a cubic with increasing volume on the bottom has a much wider signal spread even if the maximal signal strength is similar to the removed stone or the empty column.

In conclusion, the latter findings clearly show that the detection of mass deviations inside a body needs detailed knowledge about the surface. Otherwise, the signal of mass deviations would disappear in the error caused by a mis-modeled surface. On the other hand, if we have very detailed knowledge about the surface of an object and a close-mesh grid of computation points, we can detect inner mass deviations if they have a certain minimal signal which depends on the volume and density of the deviation and the characteristics of the surrounding object. Furthermore, different deviations have different signal signatures (i.e. spread, maximal strength, etc.) which means that we can conclude (to a certain extent) from the signal on the inner mass deviation. This problem is generally known as inverse modeling as we observe gravity but want to model the metric dimension and location of such deviations. The latter findings showed that this is possible if we have knowledge about the surface, dimension and density of the surrounding object and keep the impact of exterior effects as small as possible.

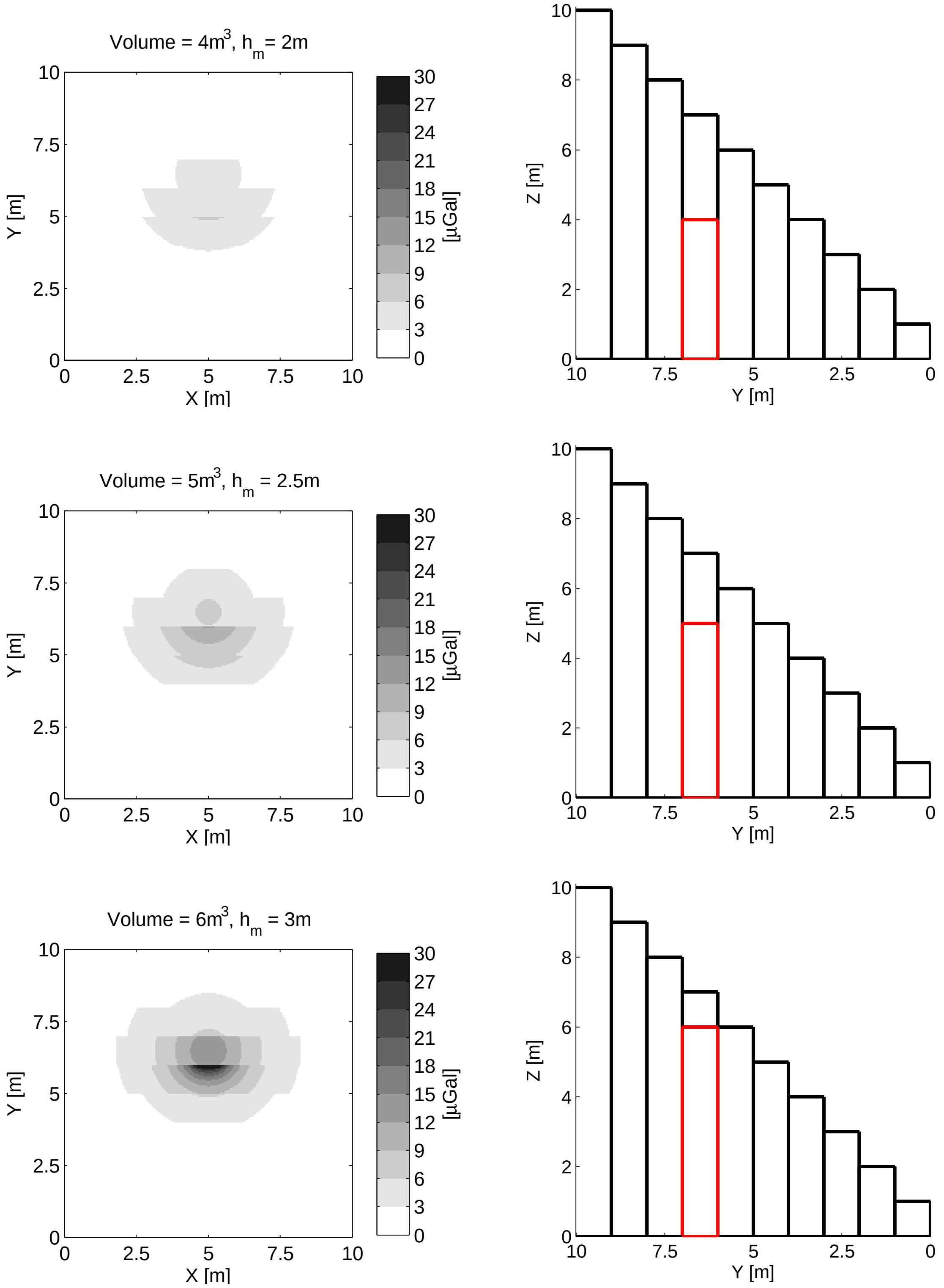


Figure 3.20: Signal of a successively growing empty column on the attraction on computation points located on the surface of the stair model

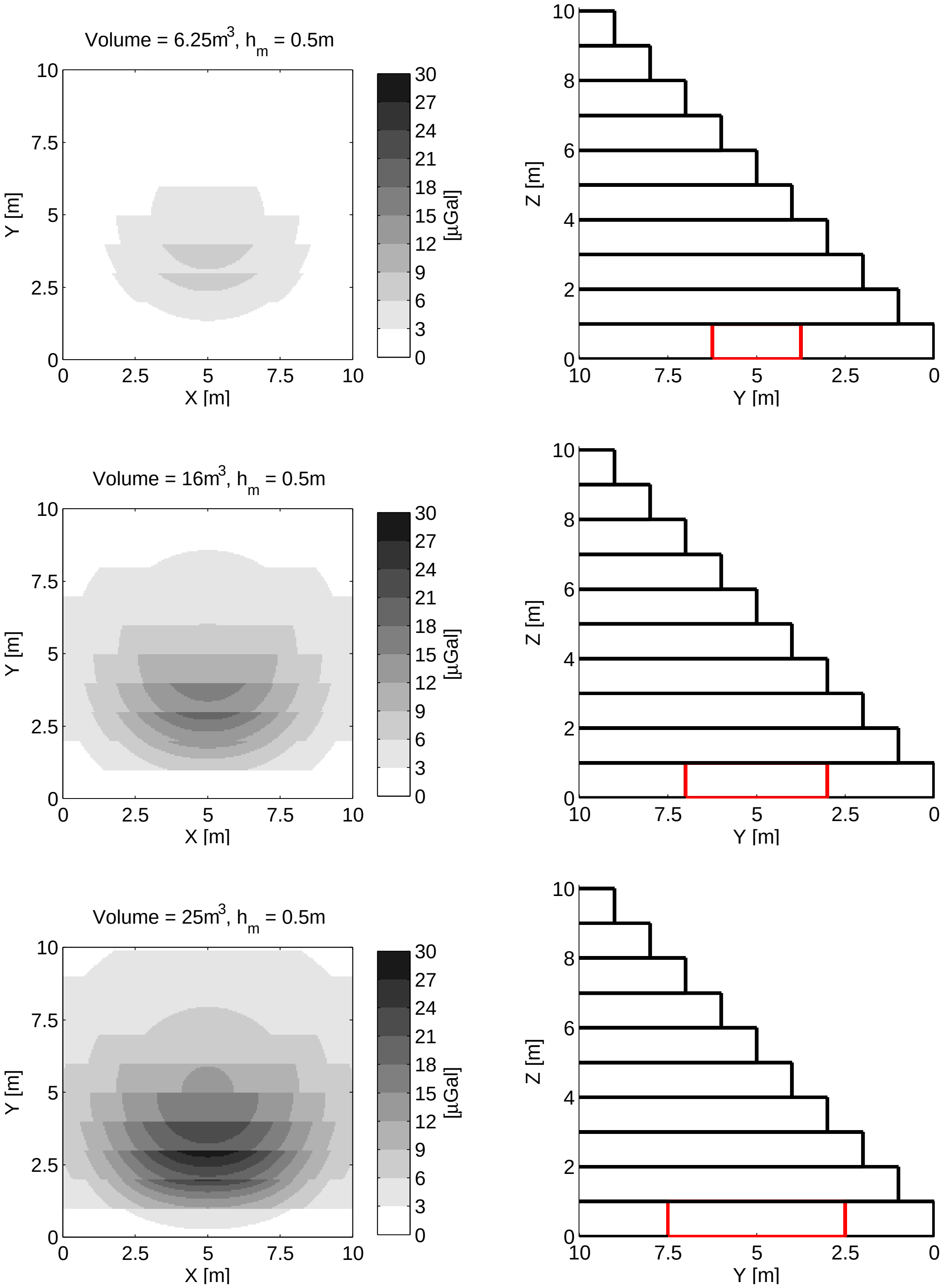


Figure 3.21: Signal of a successively growing vacuity on the attraction on computation points located on the surface of the stair model

From photography to a solid body

4

In the last chapters, an approach was presented to compute the potential and its first derivatives from an arbitrary polyhedron. Now, we use photographs to build a surface model, which will be filled with such polyhedra to construct a solid body. This allows the computation of gravitational quantities for arbitrary objects. In this chapter, the successive steps from photography to the final solid body are discussed.

4.1 Creating a three-dimensional surface model

First, the three-dimensional reconstruction of a scene from photographs is discussed. A detailed consideration of each single step or the mathematical derivations would go beyond the scope of this work but can be found in e.g. Maarten & Van Gool (2006) or Zheng (2008).

4.1.1 Taking the photographs

The first step is to take photographs of the object of which a gravity model should be computed. In fact, this step is of crucial importance for the following computations as bad or improper photographs inhibit a successful creation of a three-dimensional model. Thus, the procedure should be planed in a well-considered way and the following guidelines must be taken care of:

1. The object should be photographed from all sides with multiple photographs in such way, that there exist enough identical points in the pictures. This can be achieved by only making small steps around the object and keeping the scene always in frame (see Fig. 4.1).

2. Panning or zooming from the same location should be avoided as such sequences do not contain enough three-dimensional information for reconstruction.

3. Dead spaces (i.e. spaces where the object overlaps itself or is covered from other objects, e.g. trees) should be avoided as there is only little chance that these parts can be reconstructed correctly.

4. It is not recommended to use a object with a uniform surface as feature points are only hardly detectable. Furthermore, a moving object can not be reconstructed.

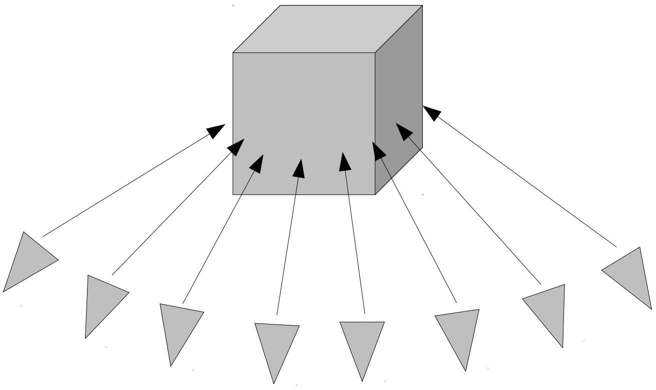


Figure 4.1: Optimal image sequence for reconstruction

5. Ensuring that the scene allows a proper georeferencation can be very important, depending on the purpose of modeling. This can be achieved by observing at least three points e.g. with GPS. Another method is to use a higher-level model, with which the scene can be linked on the basis of identical points. In both cases, the user must be able to identify these points in the reconstructed scene which makes a wise choice of these locations very important. This means on the one hand, that the points should be located close to the edges or corners of the scene with a proportionately great distance between them. On the other hand, they must be clearly distinguishable from their surrounding area in both the reconstructed and real scene.

   Additionally, it is of advantage if the scene contains an area or object, of which a scale factor can be derived. This allows a proper scaling by comparing the size of the reconstructed and real area or object.

If these rules are followed properly, a sequence of about 6 images is enough for a very detailed reconstruction of the scene. Fig. 4.2 shows such an image sequence which will be used to reconstruct a close-view of the steps of the Great Pyramid. However, a closer look shows that the upper guidelines are not completely fulfilled. For example the parts on the top of the blocks are not visible due to the recording angle. Furthermore, the texture of the scenery is very uniform, which makes an extraction of feature points difficult. But it will be shown that (at least in this case here) the presented methods turned out to be very robust and allow a proper reconstruction of the scene.

It should be mentioned that even if numerous photographs of the pyramid are available, a reconstruction of the whole pyramid was not possible due to missing reference points. Furthermore, the tools and programs which were used in this work do not allow the processing of photographs from multiple cameras or the combination of multiple image sequences. The free web application `Photosynth`[1] from Microsoft would allow such combinations but post-processing of the reconstructed scene is difficult as it is not intended to download the model (or pointcloud) for other applications. Furthermore, the presented approach turned out to be the best compromise between a very detailed reconstruction, a short computation time and user-friendly data handling.

---

[1] http://photosynth.net/

Figure 4.2: Image sequence of a close-up view of the Great Pyramid in Gizeh

4.1.2 Using a web-service for 3D-modelling

The creation of a three-dimensional model from photographs is a popular task of modern photogrammetry as it allows a detailed three-dimensional visualization of an object without the need of expensive surveying instruments like laser scanners. Thus, there are many programs available to reconstruct a scene from different images. In this work, it was decided to use the web-service ARC3D[1] and the open-source software `Meshlab`[2], which offers many features for processing three-dimensional meshes. A similar approach is described in Cignoni et al. (2008). It turned out that this web-service does a great job in reconstructing a scene from photographs. Furthermore, it does not need any user-input besides the images. After uploading the images to ARC3D via the provided upload tool, all other computations are performed automatically and the user is able to download the finished model. These computationally intensive calculations are performed on a network of many computers which drastically improves and speeds up the whole

---

[1] `http://homes.esat.kuleuven.be/~visit3d/webservice/v2/`

[2] `http://meshlab.sourceforge.net/`

process. A detailed description of the web-service can be found in Maarten & Van Gool (2006). Here, only a short overview of the service is given.

The server-side computation (i.e. after the upload of the images) of a three-dimensional model from photographs with ARC3D can be roughly split up in four successive steps:

1. Image matching and subsampling: This step aims for a correct ordering of image pairs as the photographs can be uploaded in any order. Therefore, an algorithm is implemented which is called *Global Image Comparison algorithm*, which uses a normalized cross-correlation to compare pairs of images. Afterwards, only the image pairs with a sufficient cross correlation are used for image matching. Furthermore, the images are subsampled for stability and efficiency reasons.

2. Pairwise and projective triplet matching and self-calibration: First, feature points, i.e. points which can be clearly distinguished from their neighbors, are extracted from the images. These points are matched between the image pairs from the first step and an epipolar geometry is computed. Now, image triplets are created, based on the matching results. For these triplets a projective reconstruction is computed in the form of three projective cameras and a set of 3D points (Hartley & Zisserman, 2003; Maarten & Van Gool, 2006). With the latter results, a first self-calibration is performed to obtain the intrinsic parameters of the camera.

3. Euclidean reconstruction and upscaling: Euclidean projection matrices, 3D points for the full-scale images and radial distortion parameters are computed using an upscaling algorithm. This algorithm uses a Robust Euclidean Bundle Adjustment Maarten & Van Gool (2006) to obtain these parameters. With the distortion parameters, more matching points can be found in the images which increases the total amount of 3D-points and thus the quality of the reconstruction. In the end, this Euclidean reconstruction provides a set of 3D-points and camera parameters.

4. Dense matching: This step provides pixel-wise depth maps for every image. Therefore, two different methods are suggested: the linked pair stereo, which uses a stereo algorithm (see Van Meerbergen et al. (2002) or Pollefeys et al. (2004) for further information), and the multi-view stereo algorithm, which seeks to reconstruct the 3D shape of a scene directly from a combination of more than two image (Maarten & Van Gool, 2006). The first approach performs the dense matching with high precision whether the second method puts its emphasis on speed.

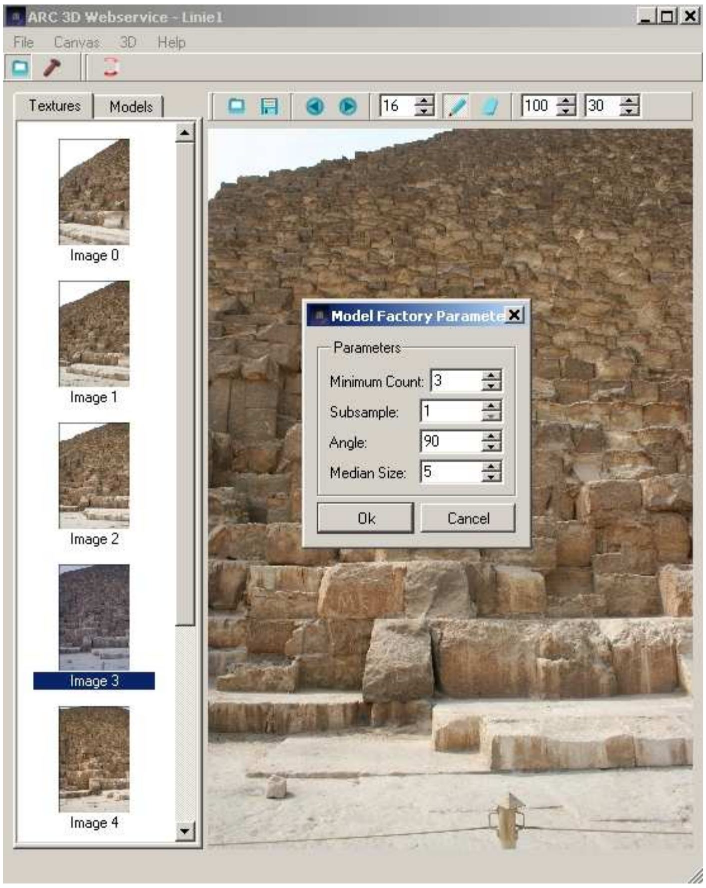
4.1.3 Download, processing and visualization of the model

After the reconstruction from ARC3D is finished, the model, which is provided in the `.V3D`-format[1], can be downloaded. It should be mentioned that the `.V3D`-file does not contain the finished model but informations about the parts which are needed to create the scene. Thus it is needed to build the model in an extra program. In this thesis, the model viewer which is provided by the ARC3D web service is used for this step.

After opening the `.V3D`-file with the viewer, the user can set some reconstruction parameters. It is suggested to set the subsample-parameter to 2. Thus, only little information

---

[1] Visual3D.NET Data File, contains data used by Visual3D.NET - a shader-based 3D engine and design toolset built on Microsoft .NET and XNA.



Minimum Count: 3
Subsample: 1
Angle: 90
Median Size: 5

Figure 4.3: Parameters of the model-creation in the model viewer provided by ARC3D

is lost during the reconstruction while keeping the memory usage and computation time on a acceptable level. Furthermore, a minimum-count value of 3 is used, which means that only pixels are considered which appear in at least three images. The parameter angle is set to 90°, i.e. no triangles are discarded during the reconstruction. After the model is created, it is saved in the `.ply`-format[1], which contains informations about the vertices and topology of the triangulated surface.

This file can be imported in the open-source software `Meshlab`, where the model is post-processed. The first step is to trim the scene until a more or less closed surface remains. Therefore, it is useful to use the implemented straighten-up tool, which allows to orient the coordinate system in such way, that the axes correspond to the geometry of the scene.

After trimming and cleaning up (i.e. removing small artifacts, etc.) a Poisson reconstruction (see e.g. Kazhdan et al. (2006)) can be performed. This step has two purposes:

- Simplification: The presented model has (even after trimming) 2371116 vertices and 4322102 faces. The Poisson reconstruction simplifies the scene to 23130 vertices and 45667 faces but tries to keep as much relevant information as possible

---

[1] Polygon File Format, a format for storing graphical objects that are described as a collection of polygons

- Closing the surface: Due to the recording angle of the photographs there might be dead spaces in the reconstructed scene, i.e. parts where no points exist. The Poisson reconstruction fills these holes with estimated data points. However, this is a delicate task as the Poisson reconstruction creates points in areas which could not be reconstructed by the web-service, i.e. areas of which no information exist. This means that the estimated points might lead to an erroneous model as it is improbable, that such a reconstructed area considers all details of the real object. The sensitivity analysis from chapter showed, that already small deviations in the surface might produce a significant signal in terms of gravitational quantities. Thus, the Poisson reconstruction is an error source for the modeling and should thus be evaluated in a future work.

Thus, the reconstruction creates a closed simplified triangulated surface which has much less data than the original model by keeping most of the necessary information. A detailed description of the parameters of this step can be found in Kazhdan et al. (2006). Here, an Octree Depth[1] of 8 and a Solver Divide value[2] of 8 is used which turned out to be a good compromise between computational time and accuracy. Finally, some minor corrections can be performed which mainly improve the optical correctness of the model. The finished model is now imported into `Matlab` via the program `plyread.m`[3] for further computations. This program creates two parameters of which one contains the surface topology information and the other the coordinates of the vertices. Thus, each row of the first parameter contains the indices of the three vertices of a single triangle in such way, that built-in functions like `trisurf.m` can be used to visualize the model.

4.1.4 Orientation of the 3D-model

The reconstructed scene might have a wrong orientation which means that a coordinate transformation must be performed. Depending on the scene there are multiple possibilities to bring the model to a correct orientation (including a scale factor). If three or more feature points in the scenery are known in a higher-level coordinate system (e.g. observed with GPS), one could simply perform a Helmert transformation to orient the model. Unfortunately, this is not the case for the presented model. But as a pyramid has a more or less simple geometry, an alternative approach can be used. The model can roughly be approximated with a tilted plane of which the orientation of the normal vector is used to rotate and tilt the surface. The scale factor can be estimated if the height of a step in the model is compared with values from literature. We know that the sides of the Great Pyramid are sloped by 51.843° and the orientation of the four sides correspond roughly to the four cardinal directions. The set of photographs which is used during this thesis shows a section of the eastern side of the pyramid. If elevation and azimuth of the normal vector of the approximated plane is computed, one can simply rotate the model in such way, that the normal vector has an elevation of $90^\circ - 51.843^\circ = 38.157^\circ$ and an azimuth of 270°. Now, the baseline of the scene is roughly parallel to the $x-$axis with the face of the pyramid pointing east and the model is sloped by 51.843°.

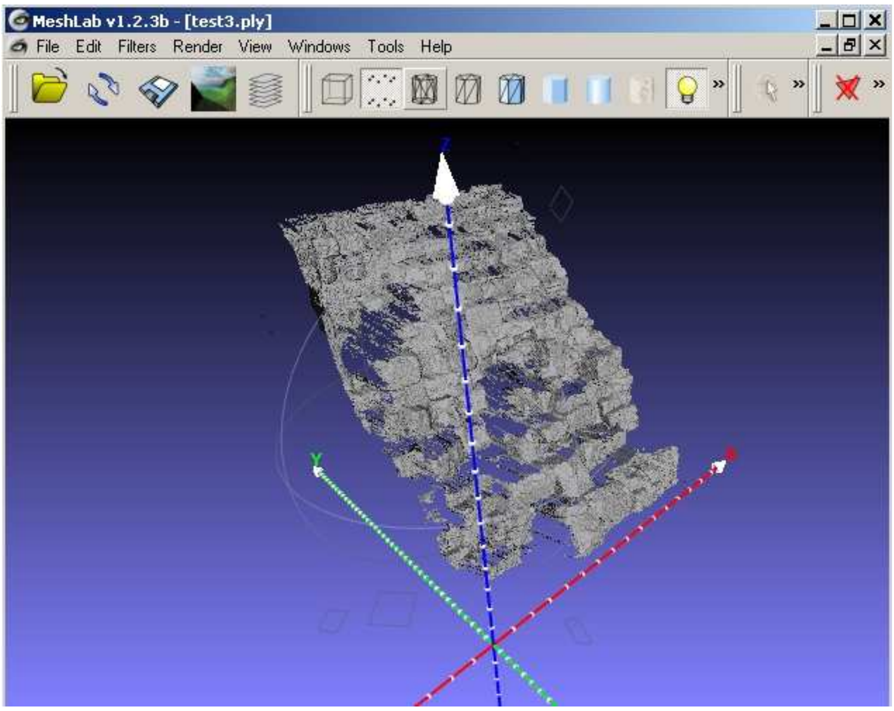
Finally, a scale factor must be considered. `Meshlab` provides a tool to measure distances in 3D-models. The measured heights and their corresponding target value (beginning

---

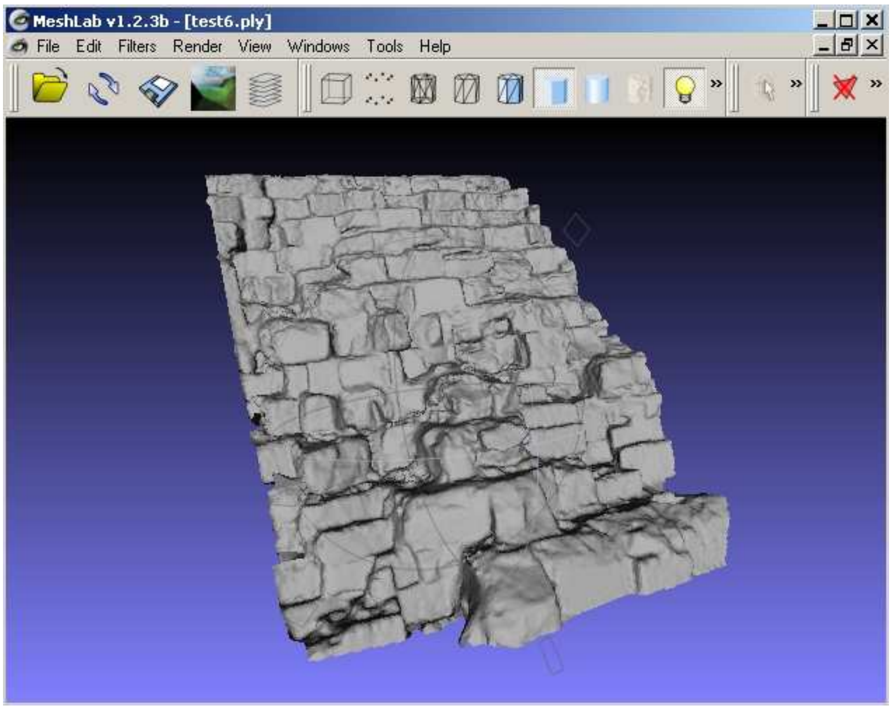
[1] The octree depth defines in how many cubes the whole scene is subdivided for reconstruction

[2] The solver divide value defines the depth at which a block Gauss-Seidel solver is used to solve the Laplacian equation

[3] `http://www.mathworks.com/matlabcentral/fileexchange/5459`



(a) Oriented and trimmed scene



(b) Scene after applying the Poisson Reconstruction

Figure 4.4: Preprocessing of the reconstructed scene in `Meshlab`

with the bottom step) are displayed in tab. 4.1. It should be mentioned that due to the very bumpy surface of the model an exact measurement of the step heights is difficult. Therefore, the displayed values should be considered as mean heights over the whole width of a step in the model. As a statement of the exact height of each step can not be made, a mean scale factor of 1.6 is assumed for the whole model.

Step	measured height [m]	target height [m]	scale factor
1	1	1.49	1.5
2	0.85	1.25	1.5
3	0.80	1.22	1.5
4	0.75	1.12	1.5
5	0.70	1.02	1.5
6	0.65	0.97	1.5
7	0.60	1.00	1.6
8	0.55	0.96	1.8
9	0.50	0.93	1.9
10	0.50	0.86	1.7
11	0.45	0.75	1.7
12	0.40	0.76	1.9

Table 4.1: Measured and target heights of the steps with the estimated scale factor

Here, a further effect appears which is caused by the recording angle of the photographs. As the photographs were recorded from the bottom of the pyramid, the upper steps are hidden behind the lower steps, which becomes clear if Fig. 4.5 is regarded. Even if all steps of the real object have the same height, the visible part of the steps from the viewpoint of the camera (the red areas in Fig. 4.5) gets smaller with increasing height. Thus, the reconstruction assumes the height of the steps to decrease which is obviously not true for the real object. Again, this is an error source which must be taken care of as such hidden areas not only make a reconstruction difficult, but might produce a distorted object.

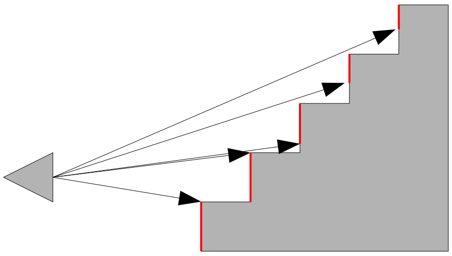


Figure 4.5: Hidden areas caused by the recording angle

The final reconstruction of the scene is presented in Fig. 4.6. We can clearly recognize the single stones of the real scene. Furthermore, the reconstruction shows a very bumpy surface, which agrees with the real object as well. Thus, it can be concluded that the parameters were chosen properly, and the presented method is able to produce a detailed model of real objects.

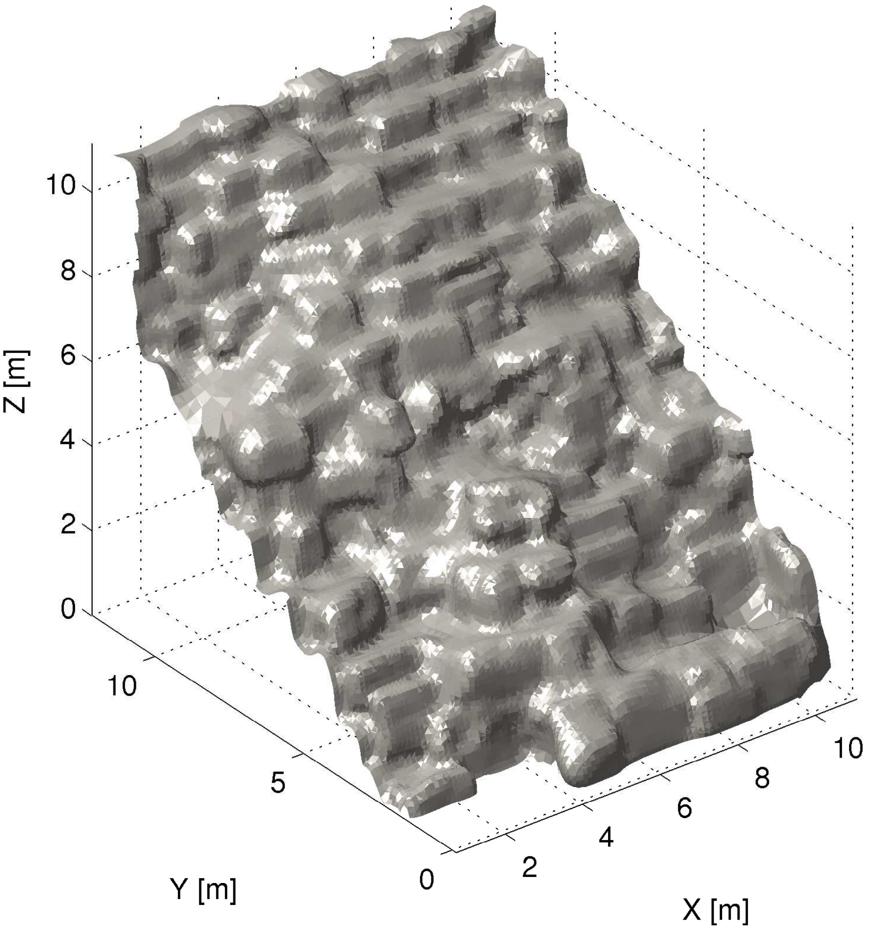


Figure 4.6: Finished reconstruction of the scene

4.1.5 Building a solid body from a surface model

In the last section a three-dimensional model was created which surface consists of triangles. The next step is to fill the inside of the object of which the surface was reconstructed with mass, i.e. to build a solid body. In this work, the body is filled with columns of constant density. Their top covers are the triangles given from the surface reconstruction whereas the base plates are the projections of the surface triangles in the plane $z = 0$. The implementation is quite simple as the topology matrix was already created during the import of the `.ply` file. As each column has the same geometry (i.e. it is constructed from two triangles on the top and at the bottom and three rectangles at the sides), a loop can be used to build the body. Fig. 4.7 shows an example surface and the corresponding definition of the columns.

As these columns are all polyhedra, the methods which were derived and discussed in chapter 2 and 3 can be used to compute the gravity and attraction of such a body. Beforehand, two checks have to be done:

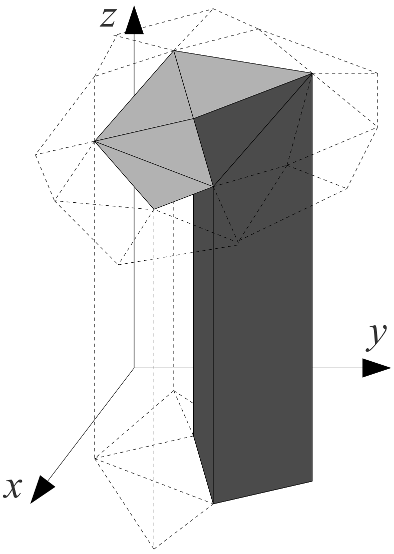


Figure 4.7: Example of a solid body constructed from a triangulated surface

- There might be regions where the model overlaps itself (see Fig. 4.8). These parts must be considered in some way as the application of the presented method would assume the empty spaces below the overhang to be filled with mass as well, which is obviously not correct. This means we need only the attraction of the solid areas above and/or below the empty space. As the total potential or attraction in a certain point is computed as the sum over the contribution of each single column, we can simply assume a negative density for these columns which represent the lower boundary of the overhang (the red areas in Fig. 4.8). This means that the contribution of such an overlapping region is removed when the summation over all columns is performed. To check if there are such areas in the model we can consider the orientation of the normal vectors of the faces (i.e. triangles). Normal vectors with a negative elevation point ”downward”, which means that the corresponding faces would produce an empty column.

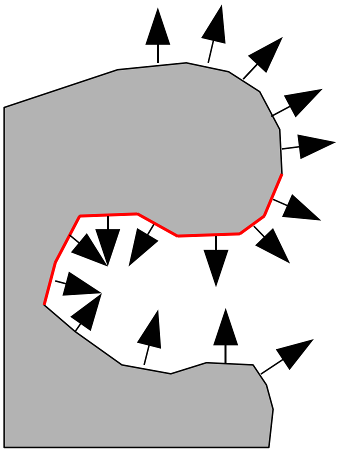


Figure 4.8: Side-view of an overhanging area and the corresponding elevation of the normal vectors

- The triangulation might produce some purely vertical triangles. Their projection in the $(z = 0)$-plane would be a line. Thus, the corresponding ”column” has no volume. Such parts can be identified if the $x-$ and $y-$coordinates of the vertices are checked for equality. If two vertices have identical $x-$ and $y-$coordinates, the corresponding triangle should be removed from the computation process.

The program which was written for this work constructs the solid body automatically during the computation of the gravity model. It only needs the topology-information, which is generated when the model is imported with the program `plyread.m`.

Sensitivity analysis

5

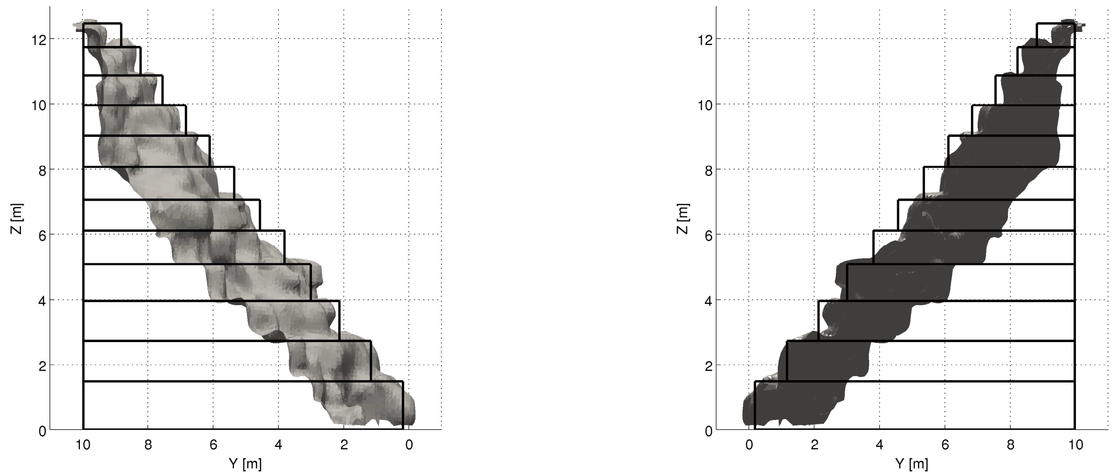
In the last chapter, an approach to generate a solid body from a surface model was described. Now, the findings from chapter 2 and 3 are used to compute the three components of the attraction of such a solid body. For the evaluation of this model, a reference model is generated. A comparison of the gravitational quantities derived from both models will give information about two things:

- the correctness of a signal computed with the line integration approach (chapters 2 and 3) and
- the overall magnitude of the gravity signal of such a model, i.e. if the difference between the reconstructed model and the simple body is measurable.

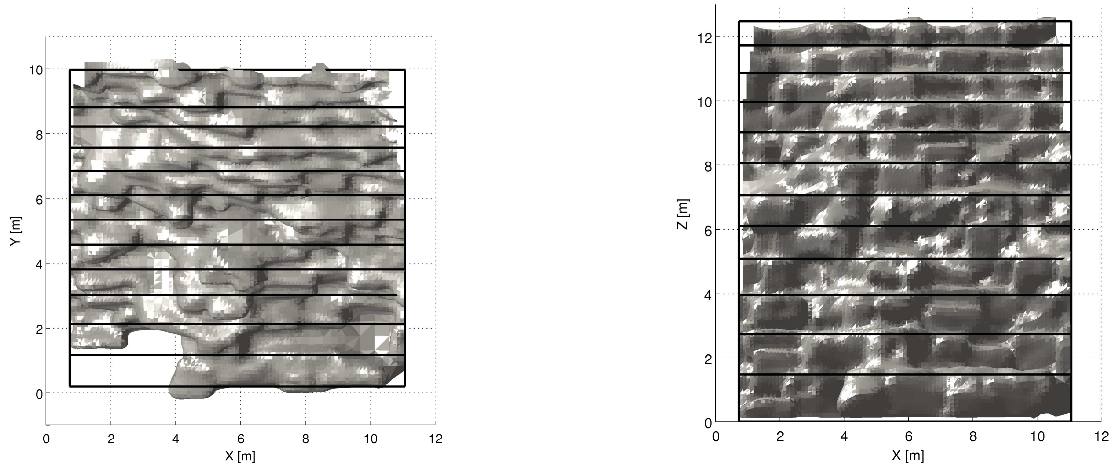
Therefore, this chapter will give information about the need of a detailed surface model if the inner mass distribution of an object shall be observed. It can be considered that a gravimeter can observe the $z-$component of the attraction with a maximal accuracy of about $1\,\mu$Gal if all exterior effects (attraction from surrounding bodies, measurement errors, etc.) can be ruled out. Thus, a more realistic estimate would be an accuracy of 4-5 μGal. This means it must be discussed if the difference between the gravity signal of the detailed and reference model has a significant magnitude, i.e. is larger than 4-5 μGal. If this would be the case, one can conclude that the surface of an object must be well known in order to make accurate conclusions about the inner mass distribution. Otherwise, the error, which is caused by improper modeling, inhibits the detection of a (usually very small) signal of inner mass deviations.

5.1 Generation of a reference model

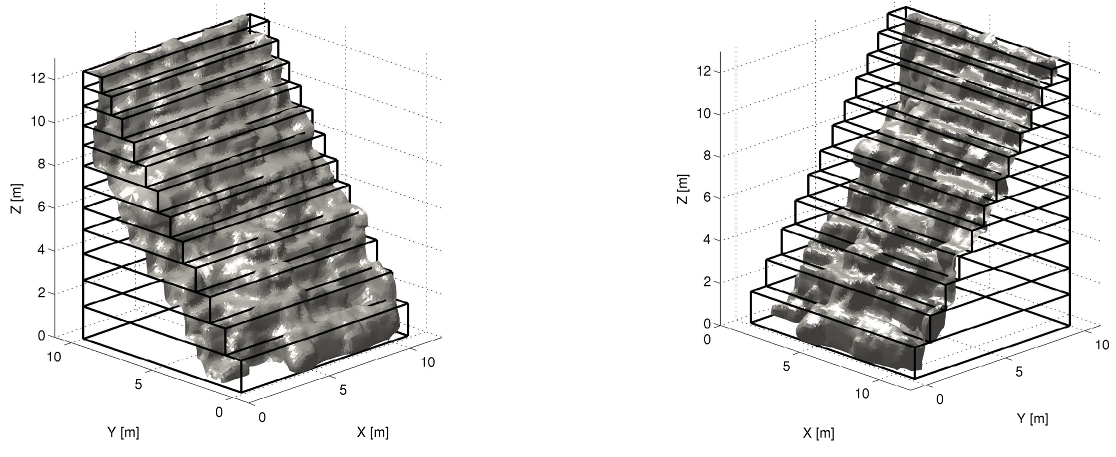
The values for the construction of the reference body are taken from literature (see Tab. 4.1 for the height-values). The outer dimensions are chosen in order to fit with the reconstructed model from chapter 4. In Fig. 5.1 both bodies are compared. It should be mentioned that for the detailed model, only the surface is displayed. A proper view of the whole solid body is not possible due to the parts where steps are overlapping. However, one has to keep the approach from last section in mind, where the triangles on the surface were projected in the $(z = 0)$-plane which allowed to fill the model with columns to create a solid body.



(a) West view (b) East view



(c) Top view (d) Front view



(e) South-West view (f) South-East view

Figure 5.1: Different views of the detailed and simplified model

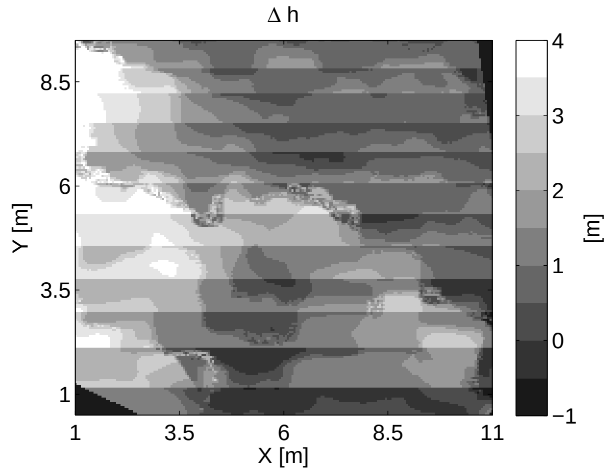


Figure 5.2: Height differences between the surface of the detailed and simplified model

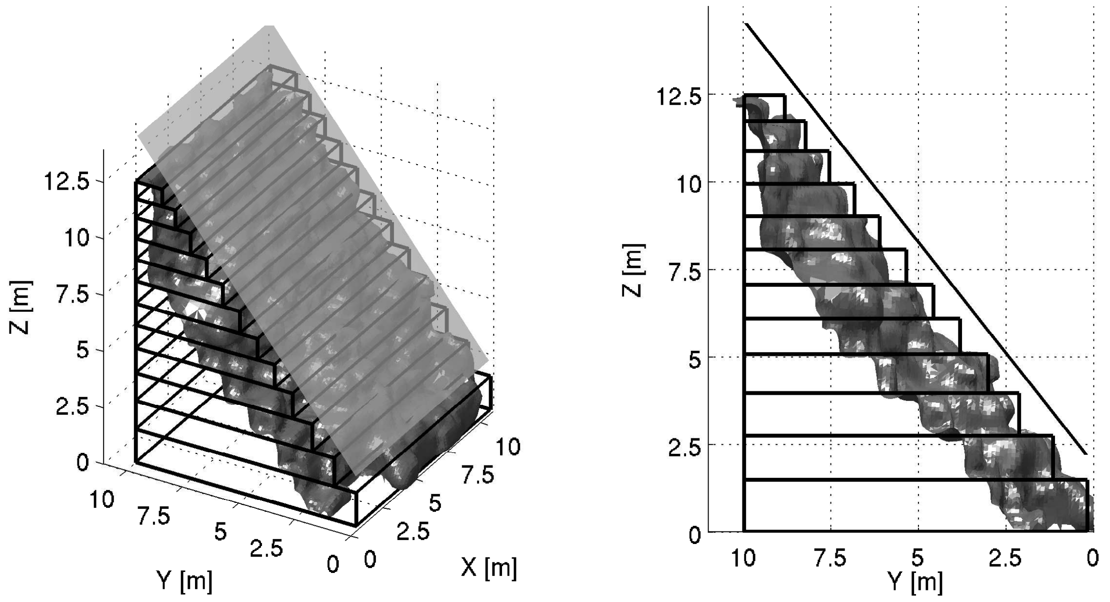


Figure 5.3: Plane in which a grid of computation points with a distance of 5 cm is created

We see that both models match very well. This actually indicates that the reconstruction of the scene from photographs was successful and that the values from literature agree with the real pyramid. Those parts where the simple body overlaps seem to come from erosion or other external influences. The lower left part shows the biggest differences between both bodies. Obviously, there are some stones missing. This is visualized in Fig. 5.2, where the height differences between the simple and the detailed model are presented[1]. It should be mentioned that the black regions can be ignored as in these

---

[1] A grid of points on the surface of the detailed model can be computed via the built-in matlab functions *griddata* or *TriScatteredInterp*

areas the detailed model has no data points. If one considers that the surface is the upper bound of the solid bodies, these differences might already give an impression how and where the gravity signals of both models will differ as they indicate where the mass distributions differ.

5.2 Direct comparison detailed vs. simple model

First, the gravity signals of both models are compared. For this, the attraction at computation points, which are located in a plane, is computed. This plane should be defined in such way, that it is on the one side very close to the models (i.e. only little signal is lost due to the vertical distance between the model and the computation points) but on the other side outside of all masses (as we can only measure the signal on the surface). Therefore, the plane is estimated into the simple stair model and translated in positive $z-$direction until it intercepts neither the detailed nor the simple body (see Fig. 5.3). For the following analysis, only the $z-$component of the attraction is considered as the $x-$ and $y-$components are mostly compensated by surrounding masses and not measurable by a gravimeter.

In Fig. 5.4, the $z-$component of the attraction of both bodies is presented. First of all, we see that both models produce a gravity signal with a magnitude up to 300 μGal at the upper part of the steps whereas the signal is nearly 0 μGal at the bottom stairs.

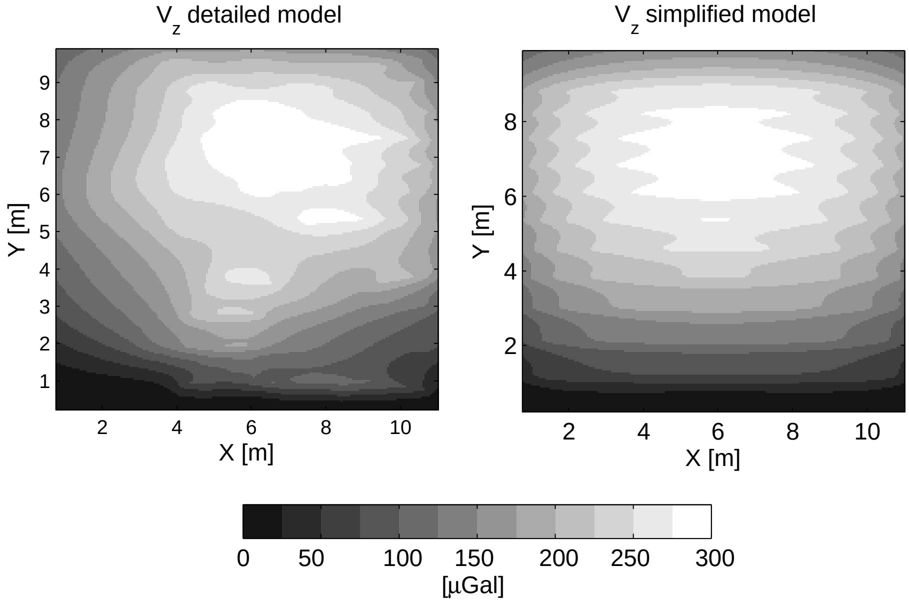


Figure 5.4: $z-$component of the attraction induced by the detailed (left) and simplified (right) stair model, evaluated at computation points in a plane above the bodies

Obviously, the signal increases with the height of the steps as the mass, which is located directly under the computation points, increases as well. This shows that the mass produces an attraction of which the $z-$component mainly impacts on the regions directly above it but has only little influence on surrounding areas.

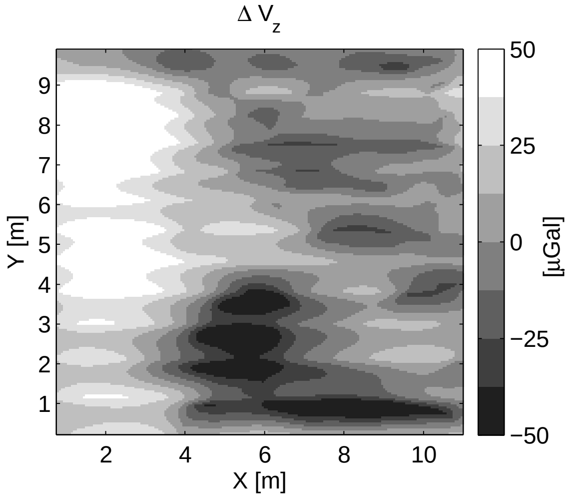


Figure 5.5: Difference between the $z-$component of the attraction in the computation points induced by the detailed and simplified model

Significant differences occur in the left area. If the height differences (and thus the different mass distributions) between both bodies are reconsidered (see Fig. 5.2), it becomes clear that this deviation is caused by those areas where the models differ. Therefore, Fig. 5.5 shows the differences between the attractions of both bodies. It is obvious that the difference between both models is significant as the attraction differs up to a magnitude of 50 μGal. If the black areas in Fig. 5.5 (the bottom steps) are compared with the height differences in Fig. 5.2, we see that the detailed model exceeds the simple model in these areas. Even if the height difference is less than 1 m, the attraction differs about 50 μGal. The bright areas at the upper left edge correspond to a large hole in the detailed model (see Fig. 5.1). Here, we see a difference in the attraction of about 50 μGal as well. Even if there are indeed some areas where only small differences in the attraction are identifiable, most parts show a significant deviation.

If the case study from section 3.4 is reconsidered in this context, it becomes obvious that the occurring differences due to missing stones, erosion, etc. between the detailed and simple model definitely cause a significant difference in the gravity signal. As a small hole on the surface already produced a signal with a strength up to 24 μGal it is obvious that using the simple body instead of the detailed model for analyzing inner mass deviations would produce large errors in the gravity signal due to mis-modeling. By comparing the signal strength of the differences in Fig. 5.5 and the signal strength of inner mass deviations from section 3.4, it can be said that and these errors due to mis-modeling would deny any conclusion about the inner mass distribution of a model. In other words, we need detailed knowledge about the surface of a body if we want to gain reasonable information about inner mass deviations.

However, a minimum level of detail of the model is hard to specify as it depends on the dimension, shape, density and inner mass distribution of an object. But if the findings from the case study in section 3.4 are considered, one should be able to appraise the required accuracy of a model if inner mass deviations shall be observed with microgravimetry.

To come back to the initial question it can be concluded that a detailed surface model of the Great Pyramid would allow the observation of inner mass deviations. But if such a model is not available, the impact (i.e. the error) of a simplified surface model would be so large that the signal of mass deviations in the inside can not be separated.

Summary and outlook 6

The objective of this thesis is to discover the influence of the outer boundary, i.e. the surface, of a solid body on its gravity signal. It was discussed if a detailed model is needed if we want to gain information about the inner structure of an object. Therefore, a model of a small section of the Great Pyramid in Giza was reconstructed with the web-application ARC3D. This model was preprocessed with the open-source software `Meshlab` to get a detailed, properly scaled and oriented stair model. By using the information gained from the three-dimensional surface triangulation, this yet empty hull was filled with columns of constant density to build a solid body. Finally, the gravitational quantities of this body were computed and compared with a reference model.

Therefore, an approach was discussed which basically transformed the volume integrals in Newton's law of gravity into line integrals by applying the Gaussian divergence theorem twice. A program which is based on this algorithm was written and validated with similar tools. The results showed that the approach can be used for the computation of gravitational quantities of solid bodies as the differences between the presented approach and other tools were caused by computational inaccuracies and thus far off the measuring accuracy of a gravimeter.

In the following case study the impact of a varying mass distribution in a body on its gravity signal was examined. It is shown that already small deviations near the surface of a body might produce a measurable signal. Thus, we need detailed knowledge about the surface if we want to make conclusions about the inner mass distribution of a body. In this section, only a few cases were considered. As general knowledge about the gravity signal of arbitrary bodies would be interesting, further investigations should deal with bodies and mass deviations with varying size, shape and density. This would allow general statements about the impact and signal characteristics of the inner mass distribution of arbitrary objects.

In chapter 4, the successive steps from photographs to the solid body were discussed. Therefore, some guidelines about the optimal characteristics of photographs for the presented approach were given. It is shown that the web-service ARC3D provides high-detailed reconstructions even if the used input photographs do not fulfill all of the described guidelines. As there were no reference points in both the model and a higher-level system or model available, matching or georeferencing of the reconstructed scene was not possible. This is an important point as the observation with microgravimetry needs knowledge about the exact location of the observed object. Thus, for future projects it must be ensured that feature points are observed in order to perform a proper georeferencation.

The reconstructed model was filled with columns of constant density to create a solid body. However, using a constant density for the hole body is improper as most objects have a varying mass distribution. If the findings from the case study are reconsidered in this context, it should be cleared in further investigations to what extend a varying density impacts on the gravity signal of a body.

In chapter 5, the z-component of the gravitational attraction of the detailed model and a simplified reference model on computation points located in a plane closely to the surface was computed. The resulting signals showed great differences which were caused by the differing dimension of both bodies. It is shown that the mass deviations caused by missing stones, erosion, etc. produce a significant gravity signal which denies the replacement of the detailed model with a simple reference model. Thus, we need a detailed surface model if we want to gain reasonable information about the interior mass distribution. The signal caused by deviations of the surface had a magnitude which would superimpose the signal from inner mass deviations and the needed signal would vanish in the errors caused by mis-modeling of the body.

To conclude these findings it can be said that the observation of a small object with microgravimetry requires detailed knowledge about its outer surface. This might be e.g. a surface model like in this thesis or, if we are dealing with a simple body, it can be constructed with elements like cubics or polyhedra while the minimum level of detail depends on the size, shape and density of the object of interest. Thus, future investigations should deal with general characteristics of the gravity signal of small bodies. As this work only considered a small section of the Great Pyramid and some bodies which were derived from the stair structure of the pyramid, the presented results hold only for these objects. However, it would be of advantage if the gravitational characteristics of a body can be appraised at an early stage. This would allow reasonable statements about the required level of detail of a model if it shall be observed with microgravimetry.

Bibliography

6

Brier B (2007) How to build a pyramid. *Archaeology* **60**.

Bronstein I, Semendjajew K, Musiol G, Mühlig H (2000) *Taschenbuch der Mathematik*. 5 edn., Verlag Harri Deutsch.

Cignoni P, Corsini M, Dellepiane M, et al. (2008) Meshlab and arc3d: Photo-reconstruction and processing 3d meshes. *EPOCH Conference on Open Digital Cultural Heritage Systems* pp. 1 – 6.

Götze HJ (1976) *Ein numerisches Verfahren zur Berechnung der gravimetrischen und magnetischen Feldgrößen für dreidimensionale Modellkörper*. Ph.D. thesis, TU Clausthal-Zellerfeld.

Gröbner W, Hofreiter N (1961) *Integraltafel*. Springer Verlag Wien.

Hartley R, Zisserman A (2003) *Multiple view geometry in computer vision*. Cambridge Univ Pr.

Hofmann-Wellenhof B, Moritz H (2006) *Physical Geodesy*. 2 edn., Springer Wien New York.

Houdin JP (2006) *Khufu: The Secrets Behind the Building of the Great Pyramid*. Farid Atiya Press.

Kazhdan M, Bolitho M, Hoppe H (2006) Poisson surface reconstruction. In: *Proceedings of the fourth Eurographics symposium on Geometry processing*, p. 70, Eurographics Association.

Kellogg OD (1954) *Foundations of potential theory*. Dover Publications.

Maarten V, Van Gool L (2006) Web-based 3d reconstruction service. *Mach Vision Appl* **17**(6):411–426.

Petrovic S (1996) Determination of the potential of homogeneous polyhedral bodies using line integrals. *Journal of Geodesy* **71**:44–52.

Pollefeys M, Van Gool L, Vergauwen M, et al. (2004) Visual modeling with a hand-held camera. *International Journal of Computer Vision* **59**(3):207–232.

Singh B, Guptasarama D (2001) New method for fast computation of gravity and magnetic anomalies from arbitrary polyhedra. *Geophysics* **66**:521–526.

Tsoulis D (1999) *Analytical and numerical methods in gravity field modelling of ideal and real masses*. Ph.D. thesis, TU München.

Van Meerbergen G, Vergauwen M, Pollefeys M, Van Gool L (2002) A hierarchical symmetric stereo algorithm using dynamic programming. *International Journal of Computer Vision* **47**(1):275–285.

Zheng B (2008) *Qualitätsvergleich von 3D Oberflächenbeschreibungen aus ARC 3D-Webservice und terrestrischen Laserscanning*. Master's thesis, University of Stuttgart, Institute of Photogrammetry.

Attraction and potential of a solid body

A

Two bodies with the masses m_1 and m_2 attract each other, according to Newton's law of gravitation, with the force

$$F = G\frac{m_1 m_2}{l^2} \tag{A.1}$$

where

F	gravitational force
G	Newtonian graitational constant with $G = 6.67259 \cdot 10^{-11} \mathrm{m^3/kg \cdot s^2}$
$m_1,\ m_2$	masses of the bodies
l	distance between m_1 and m_2

Usually, one assumes the mass m_1 (m_2) to have the coordinates ξ, η, ζ (x, y, z) in a fixed Cartesian coordinate system. Thus, the distance l between both masses is given through

$$l = \sqrt{(x-\xi)^2 + (y-\eta)^2 + (z-\zeta)^2} \tag{A.2}$$

Although the masses m_1 and m_2 attract each other in a completely symmetrical way, it is convenient to call one of them the attracting mass and the other the attracted mass (Hofmann-Wellenhof & Moritz, 2006). In the following, the attracted mass located at $P(x, y, z)$ is set equal to unity and the attracting mass is denoted by m. As the force F is represented by a vector $\boldsymbol{F}$ with magnitude F in a 3d-coordinate system xyz (see fig. A), its three components can be written as

$$X = \quad -F\cos\alpha \quad = -\frac{Gm}{l^2}\frac{x-\xi}{l} = -Gm\frac{x-\xi}{l^3} \tag{A.3}$$

$$Y = \quad -F\cos\beta \quad = -\frac{Gm}{l^2}\frac{y-\eta}{l} = -Gm\frac{y-\eta}{l^3} \tag{A.4}$$

$$Z = \quad -F\cos\gamma \quad = -\frac{Gm}{l^2}\frac{z-\zeta}{l} = -Gm\frac{z-\zeta}{l^3} \tag{A.5}$$

If it is assumed that there is not only one point mass but a system of several point masses $m_1, m_2, ..., m_n$, the force acting on the unity mass located at P is given through the principle of superposition and thus

$$F = G\sum_{1}^{n}\frac{m_i}{l_i^2} \tag{A.6}$$

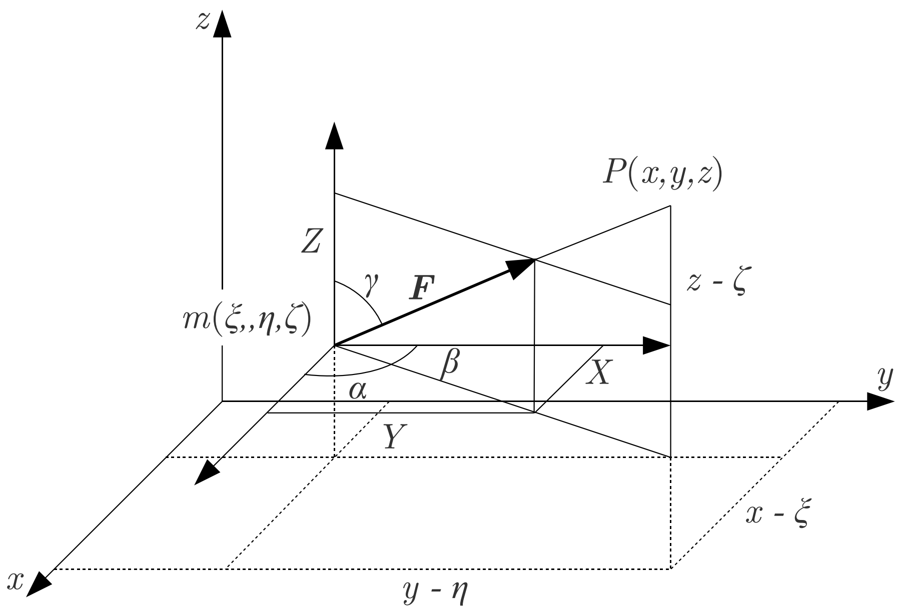


Figure A.1: Components of the gravitaional force vector

Again, the three components of F can be expressed in terms of the coordinates of P and m_i:

$$X = -G \sum_{i=1}^{n} m_i \frac{x - \xi_i}{l_i^3} \tag{A.7}$$

$$Y = -G \sum_{i=1}^{n} m_i \frac{y - \eta_i}{l_i^3} \tag{A.8}$$

$$Z = -G \sum_{i=1}^{n} m_i \frac{z - \zeta_i}{l_i^3} \tag{A.9}$$

Assuming the point masses m_i to be distributed continuously over a body with volume v (see fig. A) and the density

$$\rho = \frac{dm}{dv} \tag{A.10}$$

where

dv	element of volume (i.e. $dv = d\xi d\eta d\zeta$)
dm	element of mass

the continuous expression of equation (A.6) reads as

$$F = G \iiint_v \frac{dm}{l^2} = G \iiint_v \frac{\rho}{l^2} dv \tag{A.11}$$

However, for this work it is adequate to assume a constant density over the whole body.

By introducing a scalar function

$$V = \frac{Gm}{l} \tag{A.12}$$

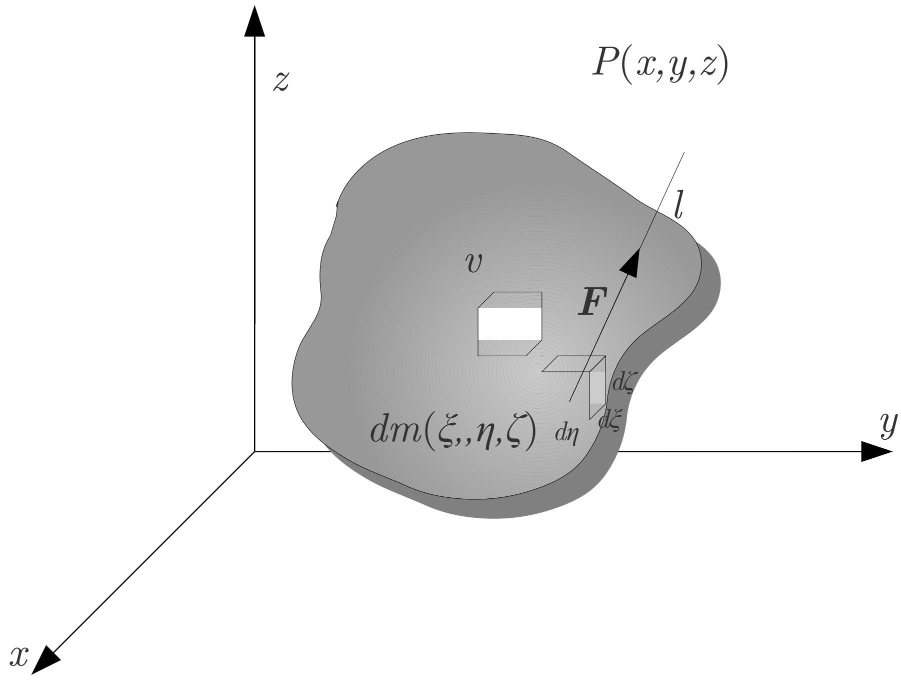


Figure A.2: Force vector of an element of mass

which is called the potential of gravitation, the components of the force vector $\boldsymbol{F}$ are given through the first derivatives of this function:

$$X = \frac{\partial V}{\partial x}, \quad Y = \frac{\partial V}{\partial y}, \quad Z = \frac{\partial V}{\partial z} \tag{A.13}$$

This is obvious since the derivative of the reciprocal distance in is given through

$$\frac{\partial}{\partial x}\frac{1}{l} = -\frac{1}{l^2}\frac{\partial l}{\partial x} = -\frac{1}{l^2}\frac{x-\xi}{l} = -\frac{x-\xi}{l^3} \tag{A.14}$$

Just like the attraction, the potential can be expanded for a continuous body as well:

$$V = \quad G\iiint\limits_v \frac{dm}{l} = \quad G\rho\iiint\limits_v \frac{1}{l}dv \tag{A.15}$$

It was already shown in equation (A.13) that the elements of the force vector $\boldsymbol{F}$ are simply the first derivatives of the potential. This is also true for the continuous expression of the potential (A.15):

$$X = \frac{\partial V}{\partial x} = G\rho\frac{\partial}{\partial x}\iiint\limits_v \frac{1}{l}dv = G\rho\iiint\limits_v \frac{\partial}{\partial x}\frac{1}{l}dv = -G\rho\iiint\limits_v \frac{x-\xi}{l^3}dv \tag{A.16}$$

This relation holds of course for the $y-$ and $z-$component of the force vector as well.

Line-wise validation, further results

B

Like in chapter, 6 further lines are created, along which all three components of the acceleration are computed. Each line consists of 4000 data points. Thus, a comparison allows a reasonable statement about the accuracy of all three programs. We see similar results compared to the lines parallel to the $x-$axis. The deviations of `accrrp.m` and `grvmag3d.m` from `polygrav.m` seem to be pure noise which is caused by computational inaccuracies due to double values. However, the deviations from `grvmag3d.m` seem to be a bit larger. As the computation procedure in `accrrp.m` is purely based on analytical solutions and the deviations from `polygrav.m` are very small, it can be concluded that using the transformation of the volume- to line-integrals for the evaluation of gravitational quantities from a solid body is valid and produces results of similar accuracy.

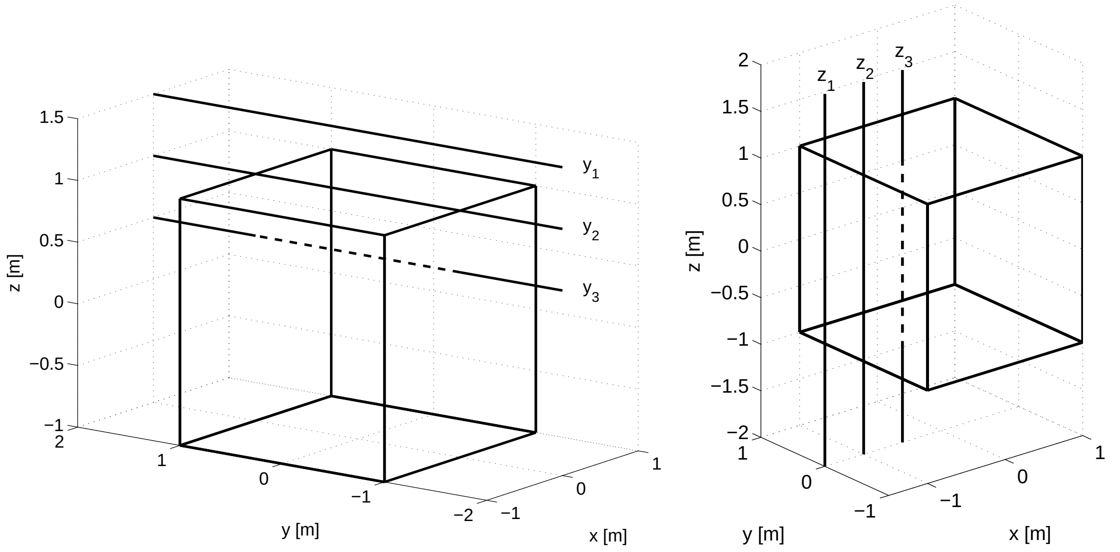


Figure B.1: Location of further lines parallel to the $y-$ and $z-$ axis with varying $z-$ and $x-$component along which the acceleration is computed

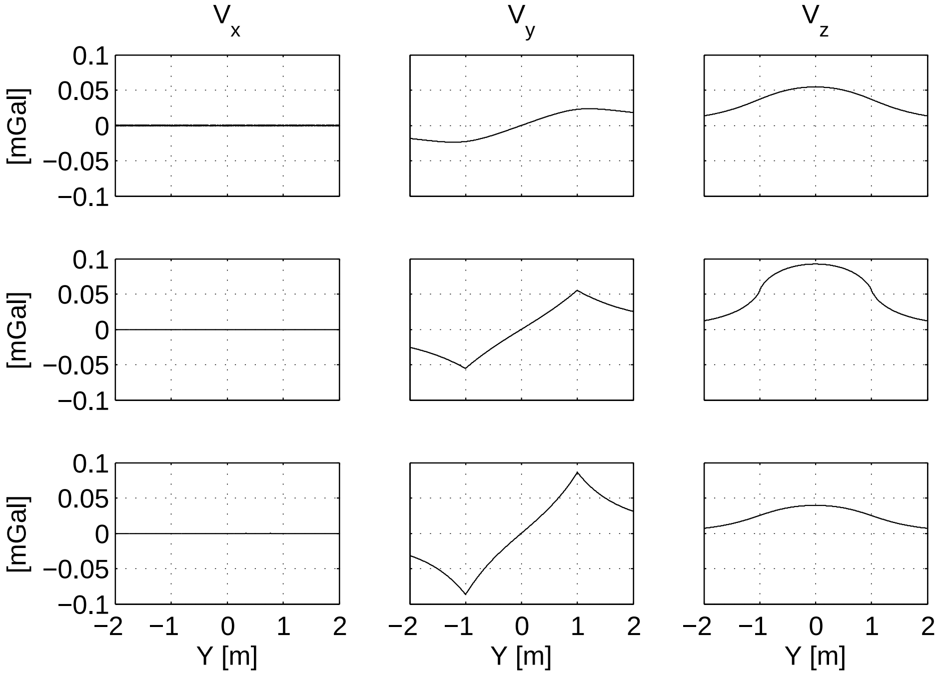


Figure B.2: $x-$ (left column), $y-$ (center column) and $z-$ (right column) component of the acceleration along line y_1 (top row), line y_2 (middle row) and line y_3 (bottom row) in [mGal]

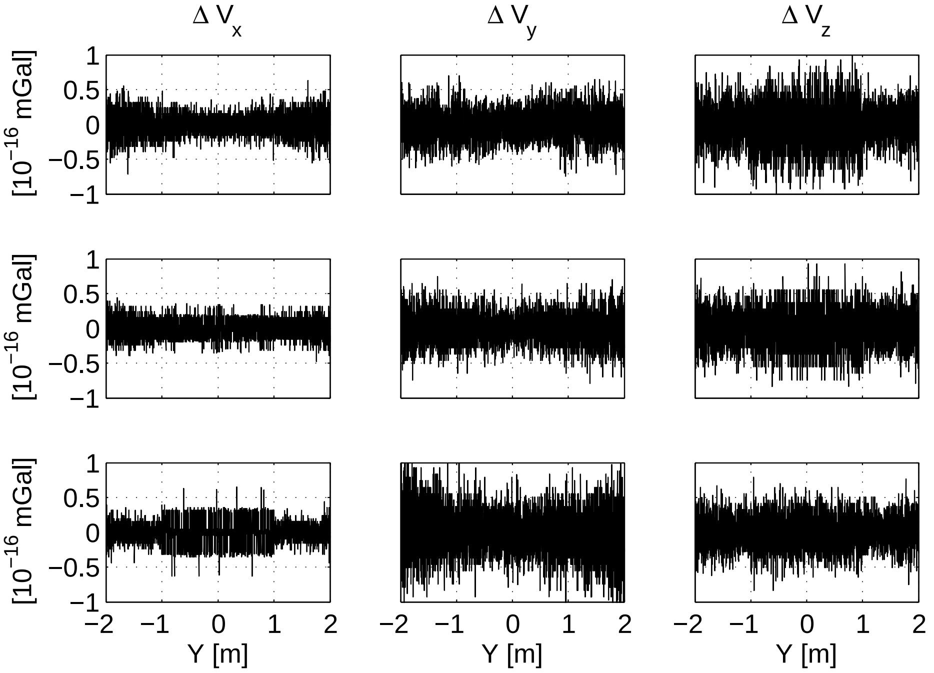


Figure B.3: Difference between the acceleration computed with `polygrav.m` and `accrrp.m` along line y_1 (top row), line y_2 (middle row) and line y_3 (bottom row) in $[10^{-16}\text{mGal}]$

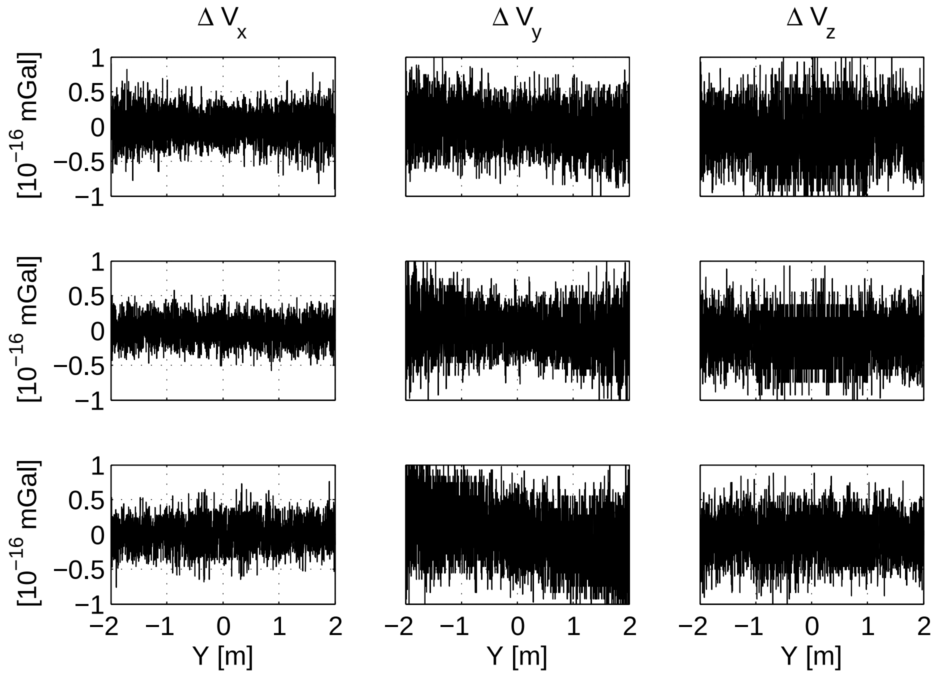


Figure B.4: Difference between the acceleration computed with `polygrav.m` and `grvmag3d.m` along line y_1 (top row), line y_2 (middle row) and line y_3 (bottom row) in [10^{-16}mGal]

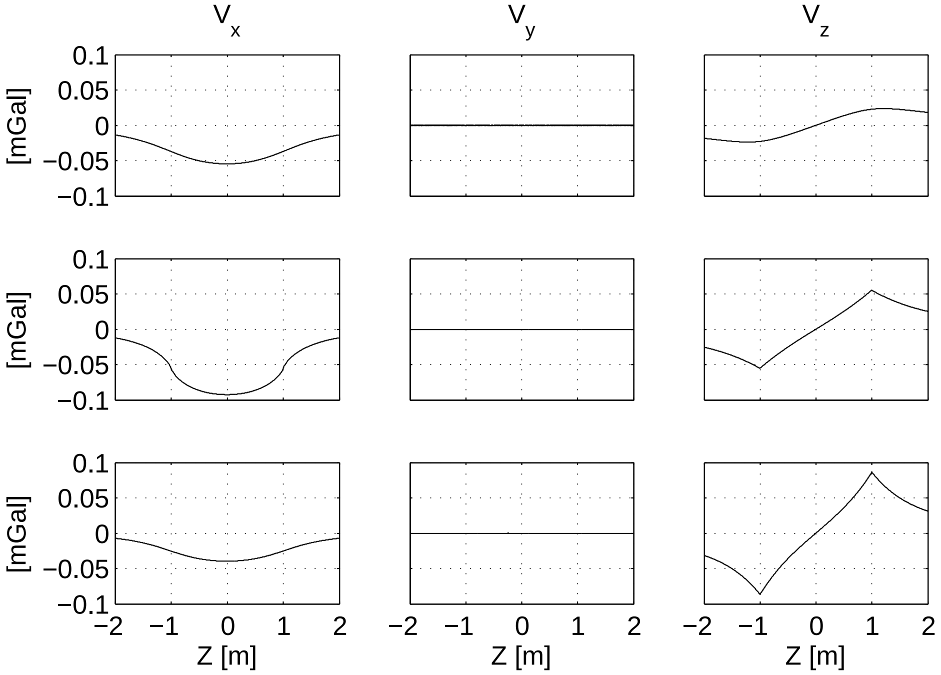


Figure B.5: $x-$ (left column), $y-$ (center column) and $z-$ (right column) component of the acceleration along line z_1 (top row), line z_2 (middle row) and line z_3 (bottom row) in [mGal]

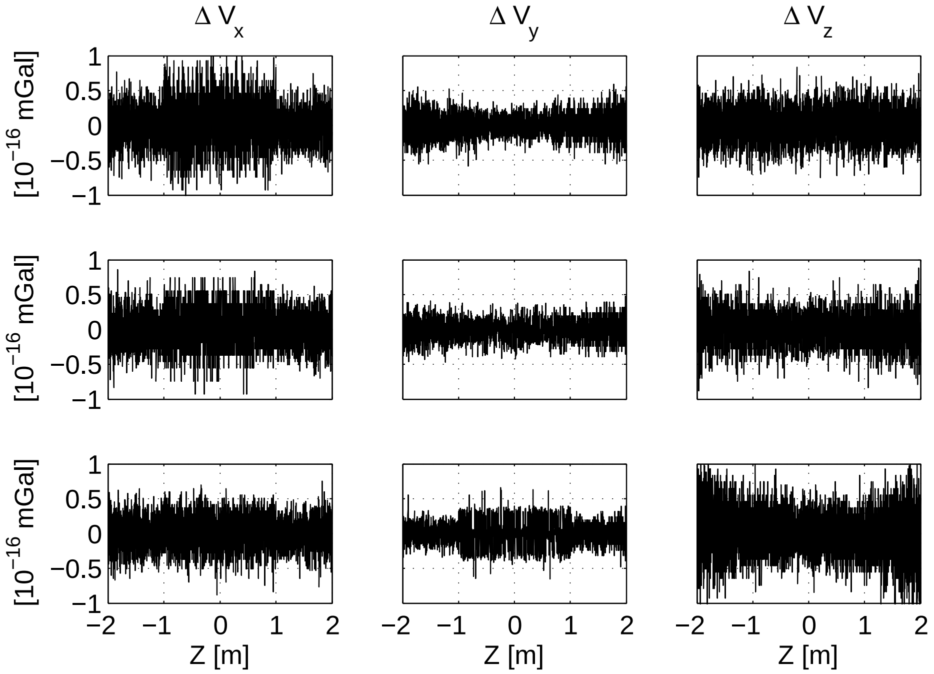


Figure B.6: Difference between the acceleration computed with `polygrav.m` and `accrrp.m` along line z_1 (top row), line z_2 (middle row) and line z_3 (bottom row) in [10^{-16}mGal]

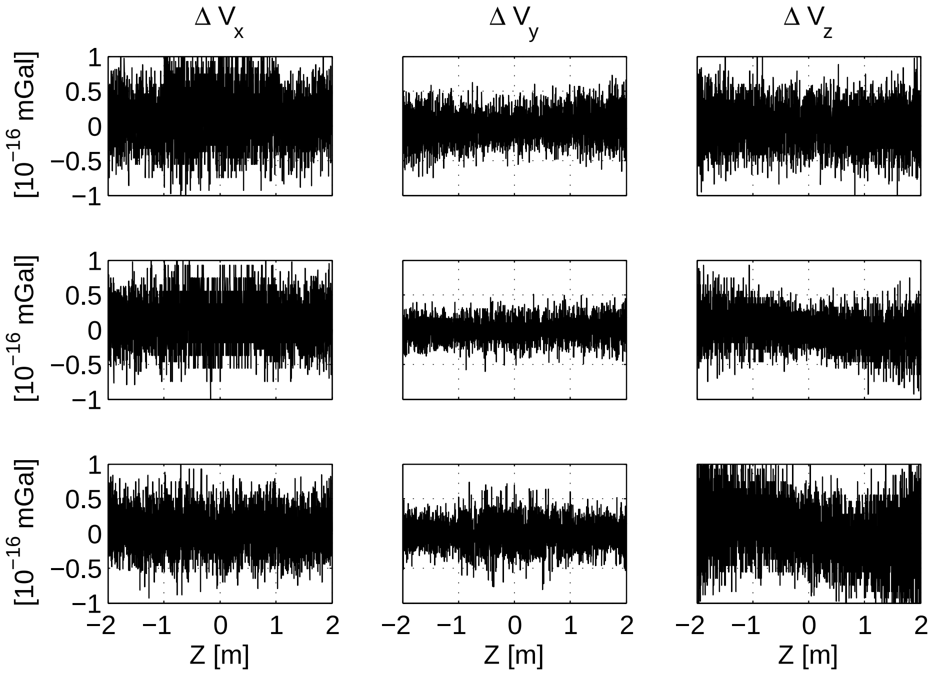


Figure B.7: Difference between the acceleration computed with `polygrav.m` and `grvmag3d.m` along line z_1 (top row), line z_2 (middle row) and line z_3 (bottom row) in [10^{-16}mGal]