Magnetic Signature Translation for Magnetic Ranging with Drones

Bachelor Thesis

B.O. Analikwu

TU Delft

The cover image is an adaptation of a photo of the HMS Northumberland and contains public sector information licensed under the Open Government Licence v3.0

Bachelor Thesis

Magnetic Signature Translation for Magnetic Ranging with Drones

by

B.O. Analikwu

to obtain the degree of Bachelor of Science
at the Delft University of Technology.

Student number:	4589076	
Project duration:	April 6, 2020 – August 26, 2020	
Supervisors:	Ir. A.R.P.J. Vijn,	TU Delft,
	Dr. ir. N.H. van Dijk,	TU Delft,
	Dr. ir. E.S.A.M. Lepelaars,	TNO & TU Delft
Committee:	Prof. dr. ir. A.W. Heemink,	TU Delft,
	Dr. W.G. Bouwman,	TU Delft,
	Dr. ir. M.B. van Gijzen,	TU Delft

An electronic version of this thesis is available at `http://repository.tudelft.nl/`.

TU Delft

Abstract

In this thesis, an algorithm to model the magnetic perturbation field caused by ships is designed and implemented. A systematic description of methods used for modelling the magnetic signature of ships is given. The algorithm fits coefficients of a prolate spheroidal harmonic expansion of the scalar potential of the magnetic field using a least angle regression method (LARS) modified to implement Lasso regularisation. A Monte Carlo method with model selection based on Akaike's information criterion (AIC) is used to select optimal parameters specifying the prolate spheroidal coordinate system centred on the ship. Furthermore, a method to restrict the degree and order of the harmonic expansion is presented and an extension of the scikit-learn module in Python is given. The predictive power of the model was verified using simulated test data, which showed that the designed model is able to make adequate predictions, but improvements are needed. Different analyses on the inputs of the model showed that the model is succesful for low levels of noise, but is susceptible to overfitting for higher levels of noise. Several recommendations for further research are made.

Acknowledgements

This thesis was written by me, Brendan Analikwu, as a part of the double degree programme of the bachelor degrees *Technische Wiskunde* (Applied Mathematics) and *Technische Natuurkunde* (Applied Physics) at the Delft University of Technology. The conditions under which I wrote this thesis were rather strange, as this project started at the beginning of the Covid-19 pandemic. However, due to the support from my supervisors, family and friends, I was able to stay motivated during these challenging times. I am thankful that the members of the reviewing committee are taking the time and effort to review my thesis on a relatively short notice. I am grateful that a group of six respected people at the university are willing to free their time for me. Specifically, I would like to thank my supervisors Aad Vijn, Niels van Dijk and Eugene Lepelaars.

Firstly, I would like to thank Aad Vijn, who is a PhD candidate researching magnetic signatures of ships. Not only did he offer interesting insights into the problem I was faced with, but he also always went out of his way to offer more help and more of his time for extra meetings.

I would also like to sincerely express my gratitude to Niels van Dijk, who a researcher at the faculty of Applied Sciences. From the beginning, I was impressed with (what I understood of) his research and I am therefore extremely grateful that he was willing to supervise my thesis. His comments and insights always helped me further along and his constructive criticism definitely helped me raise the bar.

I am also extremely grateful for my supervisor Eugene Lepelaars. His experience at TNO showed through his insights and certainly helped make this thesis more relevant in practice. I would like to thank Eugene for going out of his way to provide the simulations of data I needed.

Lastly, I would like to thank my friends and family for the support they provided during these times. I am thankful for my parents who have always supported me and inspired me in my education. I am especially grateful for my boyfriend Chamon, who provided a very patient listening ear for my ramblings and without whom this would not have been possible.

B.O. Analikwu
Delft, August 2020

Contents

List of Symbols

This is a list of symbols that are used in this thesis. The values for physical constants were taken from [1, p. 577].

Physical constants

ϵ_0	Permittivity of vacuum	8.85×10^{-12} $\mathrm{C^2 \cdot N^{-1} \cdot m^{-2}}$
μ_0	Permeability of vacuum	$4\pi \times 10^{-7}$ $\mathrm{N \cdot A^{-2}}$
χ_m	Magnetic susceptibility of a certain material	
μ	Permeability of a certain material	$\mathrm{N \cdot A^{-2}}$

Physical quantities

$\mathbf{F}$	Lorentz force	N
q	Particle charge	C
$\mathbf{E}$	Electric field	$\mathrm{N \cdot C^{-1}}$
$\mathbf{v}$	Velocity	$\mathrm{m \cdot s^{-1}}$
$\mathbf{B}$	Magnetic field	T
ρ	Charge density	$\mathrm{C \cdot m^{-3}}$
$\mathbf{J}$	Total current density	$\mathrm{A \cdot m^{-2}}$
$\mathbf{m}$	Magnetic moment	$\mathrm{A \cdot m^{2}}$
$\mathbf{M}$	Magnetic dipole moment per unit volume	$\mathrm{A \cdot m^{-1}}$
$\mathbf{J}_b$	Bound current density	$\mathrm{A \cdot m^{-2}}$
$\mathbf{J}_f$	Free current density	$\mathrm{A \cdot m^{-2}}$
$\mathbf{H}$	Auxiliary magnetic field	$\mathrm{A \cdot m^{-1}}$
Ψ	Scalar potential of magnetic field, see (2.7) or (2.15)	A
ρ_m	Magnetic charge density	$\mathrm{A \cdot m^{-2}}$

Abbreviations

PDE	Partial differential equation, see (2.13)
AIC	Akaike's Information Criterion

Model variables

(x_0, y_0, z_0)	The centre point of the prolate spheroidal coordinate system, see page 17	m
$\mathbf{B}^J_{N,M}$ or $\mathbf{B}^J$	The vector containing magnetic field values split into each direction for J measurements, as defined in (3.9) and (4.1)	nT

$\mathbf{B}_j$	The magnetic field vector of the j-th measurement, see page 17	nT
$\mathbf{B}_{N,M}$	Terminated prediction of the magnetic field, see (3.6)	
A_Ψ	Matrix of the linear problem in (3.2) giving the relation between the multipole coefficients $\boldsymbol{\beta}$ and scalar potential $\boldsymbol{\Psi}_{N,M}$, see (3.4)	
$\boldsymbol{\beta}$ or $\boldsymbol{\beta}$	Vector of multipole coefficients with maximum degree N and maximum order M, see (3.2) and (3.3)	
$\boldsymbol{\epsilon}$	The vector of residuals in the regression, see page 19	nT
$\boldsymbol{\Psi}^J_{N,M}$	Vector of scalar field values for J observations and maximum degree N and maximum order M, see (3.2)	
$\Psi_{N,M}$	Scalar potential of the auxiliary field with terminated sums at N and M, see (3.1)	
M	Largest evaluated order in $\Psi_{N,M}$, see (3.1)	
N	Largest evaluated degree in $\Psi_{N,M}$, see (3.1)	
L	Half of the focal length of a prolate spheroidal coordinate system, see (2.10)	m
$x,\ y,\ z$	Cartesian coordinates	m
ξ	Radial coordinate of prolate spheroidal coordinate, see (2.11a)	
η	Angular (azimuthal) coordinate of prolate spheroidal coordinates, see (2.11b)	
ϕ	Angular (elevation) coordinate of prolate spheroidal coordinates, see (2.11c)	
h_u	Scale factor for any coordinate u, see page 9	
$\hat{\mathbf{u}}$	Unit vector for any coordinate u	
$\Xi(\xi)$	Radial part of Ψ, see (2.13)	
$\mathrm{H}(\eta)$	Azimuthal part of Ψ, see (2.13)	
$\Phi(\phi)$	Elevational part of Ψ, see (2.13)	
P_n^m	Associated Legendre function of the first kind of the n-th degree and m-th order, see page 11	
Q_n^m	Associated Legendre function of the second kind of the n-th degree and m-th order, see page 11	
c_n^m	Multipole expansion coefficient of the term $\mathrm{Q}_n^m(\xi)\mathrm{P}_n^m(\eta)\cos(m\phi)$, see (2.15)	
s_n^m	Multipole expansion coefficient of the term $\mathrm{Q}_n^m(\xi)\mathrm{P}_n^m(\eta)\sin(m\phi)$, see (2.15)	

LARS variables

$\hat{\mathbf{B}}_{\mathcal{A}}$	The working solution belonging to the active set $\mathcal{A}$
$\hat{c}_i$	Correlation of column i of matrix A from the linear problem with the residuals
$\mathcal{A}$	Active set in the LARS algorithm, meaning that it contains the indices of coefficients that are unequal to zero in the working solution
$\mathbf{a}_i$	The i-th column of matrix A from the linear problem

1

Introduction

Naval mines are still an active threat. On the one hand, recent news reports of naval mines being fished up in the North Sea (such as [2]) show that naval mines from passed wars still pose a threat to for example civilian ships. On the other hand, a massive anti-naval mine drill by the Chinese People's Liberation Army Navy, just as recently as 2018 [3], shows naval mines still play an active roll in modern (naval) warfare. Protecting ships from naval mines is therefore still crucial for safety at sea.

The Netherlands Organisation for Applied Scientific Research (TNO) is working together with the Delft University of Technology in this area. They are working on advanced numerical models that predict the interactions of ships with the electromagnetic field. These interactions make ships susceptible to detection by naval mines.

1.1. Magnetic signature

The three typical components of a defence against naval mines are minesweeping, minehunting and the reduction of a vessel's signature. Minesweeping is the oldest method of the three. Essentially, a ship (or more recently a helicopter) drags a wire through the water to either trigger the mine or cut the line holding it in place. This is a crude, yet effective method.

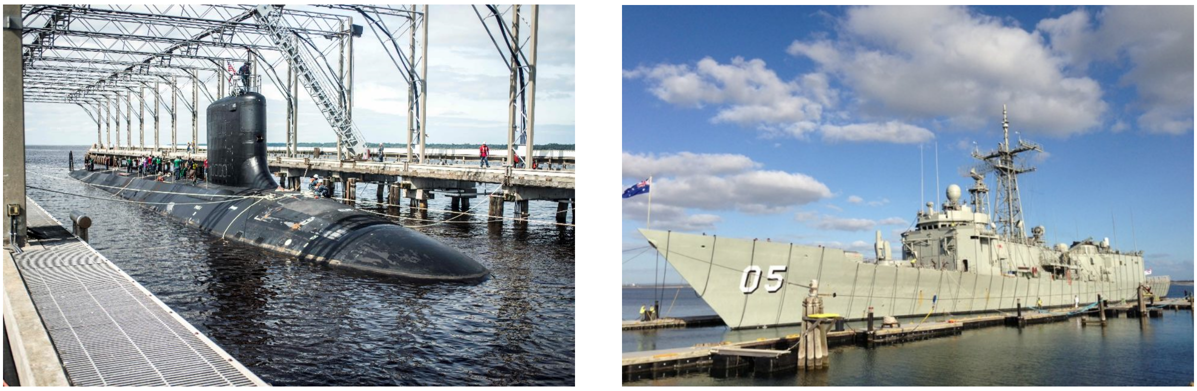
Minehunting, on the other hand, is a more precise practice: mines are actively sought out. After detection, a diver or an unmanned underwater vehicle (UUV) either dismantles the mine or safely detonates it.

The third component is the reduction of a ship's signature. The term 'signature' was first used to refer to a vessel's underwater acoustic pressure field, by which it could be detected or even distinguished [4, p. 1]. Similarly, the spatial and temporal distribution of a ship's magnetic field is referred to as the magnetic signature. It is this magnetic signature that naval mines can detect and more advanced systems can distinguish between them and select a programmed target. Reducing the magnetic signature could therefore ideally make the ship 'magnetically invisible' for naval mines.

Since electromagnetic fields cannot be seen by the naked eye, finding and eliminating its sources might prove to be difficult. Four sources of the magnetic signature of a naval vessel can be discerned:

1. ferromagnetism induced by the Earth's magnetic field;
2. eddy currents caused by rotation in the Earth's magnetic field;
3. electric currents caused by natural electrochemical corrosion or cathodic prevention thereof;
4. currents in the electric appliances.

Only the first of these sources is considered in this thesis, since the other sources provide either an unreliable or small contribution to the total signature and would therefore not be of interest to mine system designer [4, p. 1-2].



(a) The USS West Virginia undergoing a deperming treatment in the drive-in facility of the Naval Submarine Base Kings Bay. Source: United States Navy

(b) The HMAS Melbourne undergoing a close-wrap deperming treatment. Source: Royal Australian Navy

Figure 1.1: Vessel going through different deperming methods. (a) shows the drive-in method and (b) shows the close-wrap method.

Low-alloy steels are the most common materials used in ship construction and finds use in the ship's hull, internal structure and machinery. As opposed to stainless steel (more than 10% chromium), low-alloy steel is ferromagnetic. Even when ferromagnetic steel is not used in the hull, most machinery still depends on this material to work properly and thus ships always contain ferromagnetic steel in practice. This steel has a high magnetic permeability, and therefore interacts strongly with the Earth's magnetic field [5, p. 10-12]. It is this interaction that results in the measurable disturbance in the magnetic field that constitutes the majority of the vessel's magnetic signature.

1.2. Deperming and degaussing

The total magnetic signature of a ship can be separated into a permanent and an induced magnetisation. The processes in which these two types of magnetisation are reduced are called *deperming* and *degaussing*. The deperming process aims to reduce the permanent magnetisation as much as possible and the degaussing process is used to remove the induced magnetisation and the remainder of the permanent magnetisation.

At a microscopic level, ferromagnetic material are composed of so-called magnetic domains. Within these magnetic domains, the magnetisation is uniform. Under the influence of mechanical stress, high temperatures or large magnetic fields, these domains can reorient themselves, which results in a permanent magnetisation or *perm*. Vessels are subjected to these stresses during construction, but also during their deployment [5, p. 16]. This means that the reduction of this permanent magnetisation to acceptable levels must be carried out multiple times in the ship's lifetime.

In the deperming process, the ship is placed within a set of coils that, when powered, create a strong magnetic field that reorients the permanent magnetic signature of a ship. This process takes multiple iterations, since measurements need to be taken of the new magnetic signature.

Deperming usually takes one of three forms: *drive-in*, *over-run* and *close-wrap* [6, p. 186]. For the drive-in method, a permanent construction of treatment coils is needed (see Fig. 1.1a). The ship then enters the facility from one end, where it is treated and can then leave. The over-run set-up has treatment coils on the seabed over which the ship transits multiple times for a successful treatment. The close-wrap system requires the coils to be wrapped around the vessel (see Fig. 1.1b).

If the deperming treatment is successful, the permanent magnetisation is reduced to an acceptable level and the magnetic signature is dominated by the induced magnetisation. Since the induced magnetisation is dependent on the size and polarity of the Earth's magnetic field relative to the ship, which changes with the position of the ship on the Earth's surface, the reduction of the induced field must be carried out actively. The degaussing process seeks to create an equal, but opposite magnetic field with respect to the induced magnetic field and the remainder of the perm. The superposition of the two fields, ideally results in a net zero field making the ship ‘magnetically invisible’.

The system exists of current loops in the ship, that can cancel the induced magnetic field and the remaining perm when powered correctly. Three different types of coils can be discerned: the M-, A- and L-coils [5, p. 43-48]. Each coil is actually made of smaller loops, enabling more precise cancellation of the magnetisation due to local irregularities. The M-loops can compensate the vertical component of the induced magnetic field. The A- and L-loops cancel out the athwart and longitudinal components.

1.3. Signature measurements

To check whether the degaussing system is working properly, measurements of the magnetic field around a vessel need to be taken. Usually, this is done by moving the ship over a sensor array or having the ship in-place over a grid of sensors on the seabed [5, p. 49]. If a linear array is chosen, the time series data are translated to a spatial pattern. This method is expensive, because a permanent setup is required. The alternative of these permanent installations is the use of temporarily placed sensors. Some implementations of this method require divers to place the sensors and calibration of the range itself before any measurements can be taken. Other systems still require specifically trained crew members and the calibration. This method is therefore quite time consuming: it may take two days to set up and one day to dissemble [7, p. 4].

A more cost-effective method has been proposed [8]: taking measurements of the magnetic field with a drone above the ship. Translation of the magnetic field measurements to underneath the ship is needed as an additional step to make conclusions about the current perm and the effectiveness of the degaussing system.

The main advantages of the use of drones include that drones are easily deployed, which reduces time spent on these measurements, and that this system is portable. Drones would make a brief check of a ship’s perm a possibility, only necessitating the need to return to a deperming facility when needed. If the ship is at rest, a drone could be launched to take measurements of the magnetic field around the ship. With the use of positioning systems such as GPS, accelerometers and gyroscopes, an image of the field around the ship can be created. Moreover, the possibility of calibration of the degaussing system and an increased ability to assess the mine risk during a mission, make this a very attractive alternative.

1.4. Research aim

In this thesis, the possibility of the use of drones in the magnetic ranging process will be explored. The main research aim is to conclude to the possibilities and limitations of magnetic ranging with drones. To this end, three questions are formulated:

- What are the relevant aspects that come into play when magnetic ranging is performed by a drone?

  Answering this question entails investigating the usual magnetic ranging process and exploring the differences when ranging is not performed below, but above the source. These include errors due to positioning and the magnetic field of the drone itself.
- How can these aspects be implemented into a model?

  To answer this question, a prediction model of the magnetic field underneath ships will be developed. The input will constitute field measurements performed in the region accessible to drones. The quality of the prediction model will be investigated by applying the model to different simulations of measurement data, since currently no publicly accessible data sets exist.
- What are the dependencies of the model’s performance?

  An analysis on the model is performed to investigate its sensitivity to the noise level on the inputs and to offsets in certain parameters.

In Chapter 2, the relevant magnetostatic theory and the prolate spheroidal coordinate system and its harmonic functions are discussed. The definition of the forward problem is formulated in Chapter 3. In 4, the inverse problem is defined and the conditions for the well-posedness of the problem are explored. The problem at hand will turn out to be linear to a certain extent, opening up the possibility for the use of regression methods and the regularisation thereof. These methods are discussed in Chapter 5. In Chapter 6, the optimisation of the hyperparameters used in the regression are discussed. Chapter 7 treats the selection of the optimal model from models developed in earlier Chapters. The acquired knowledge from the previous Chapters is combined in Chapter 8, where the complete model design and implementation is discussed, with some analysis methods. The simulation of the test data is discussed in Chapter 9. The results are presented in Chapter 10. The discussion of the results and the conclusions are given in Chapters 11 and 12.

2

Magnetostatics

To understand the observed magnetic field caused by the magnetisation of a ship, the theory of magnetism must first be discussed. The theory described in the following, was largely based on [1]. Even though virtually everyone is familiar with the basic concept of magnetism, the underlying principles and theories are not as easily understood. Magnetic fields cannot be seen by the naked eye, smelled or tasted. It is the interaction with materials that provide an opening into discovering its secrets.

2.1. Magnetic theory

The magnetic field and its electric counterpart interact with electrically charged particles. This interaction is captured by the electromagnetic force, or Lorentz force [1, p. 212]:

$$\mathbf{F} = q\mathbf{E} + q\mathbf{v} \times \mathbf{B},$$

where $\mathbf{F}$ [N] is the force vector, q [C] is the charge of the particle, $\mathbf{E}$ [N/C] is the electric field vector, $\mathbf{v}$ [m/s] is the velocity vector and $\mathbf{B}$ [T] is the magnetic field vector.

Contrary to stationary charges, moving charges produce not only an electric field $\mathbf{E}$, but also a magnetic field $\mathbf{B}$ [1]. These magnetic fields are, together with electric fields, governed by Maxwell's equations:

$$\nabla \cdot \mathbf{E} = \frac{\rho}{\epsilon_0}, \qquad \text{(Gauss' law)}$$

$$\nabla \cdot \mathbf{B} = 0, \qquad \text{(Gauss' law for magnetism)}$$

$$\nabla \times \mathbf{E} = -\frac{\partial \mathbf{B}}{\partial t}, \qquad \text{(Faraday's law)}$$

$$\nabla \times \mathbf{B} = \mu_0 \mathbf{J} + \mu_0 \epsilon_0 \frac{\partial \mathbf{E}}{\partial t}. \qquad \text{(Ampère's law)}$$

Here ρ [C/m^3] is the charge density, ϵ_0 [C/Nm^2] is the permittivity of vacuum, $\mathbf{J}$ [A/m^2] is the total current density and μ_0 [N/A^2] is the permeability of vacuum .

Ampère's law unveils the sources of the magnetic field: current densities and changing electric fields. This explains a magnetic field arising from a current loop, but the field generated by a permanent magnet is not as easily related to this law since there are no changing electric fields and no apparent currents. However, when a magnetic material is investigated at an atomic level, minuscule currents are discovered: electrons orbiting around atoms and showing a spin. These bound electrons generate a magnetic field. When oriented randomly, the resulting fields cancel out, but when all electrons are aligned, such as in a permanent magnet, a permanent magnetic field arises.

The *magnetic moment* $\mathbf{m}$ [Am^2] is introduced to describe the orientation and magnitude of the magnetic field originating from objects (e.g. current loops, molecules, planet cores). This quantity is often referred to as the magnetic *dipole* moment, because for large distances from the source or a sufficiently small source, the dipole moment is the dominant term. In other circumstances, more terms are needed to describe the magnetic field accurately [1, p. 254].

A magnetic dipole is an infinitesimally small current loop or a north and south pole infinitesimally close together. On a macro scale, electrons orbiting atoms are approximate magnetic dipoles. The alignment of magnetic moments is described with the magnetisation $\mathbf{M}$ [A/m], which is the magnetic dipole moment per unit volume. The relation to the bound current is given by

$$\mathbf{J}_b = \nabla \times \mathbf{M}.$$

The remainder of the currents, the free current $\mathbf{J}_f$, is actually what is experimentally controllable. The magnetic field due to the free current is described with the auxiliary field $\mathbf{H}$ [A/m^2]. It is defined as

$$\mathbf{H} \equiv \frac{1}{\mu_0}\mathbf{B} - \mathbf{M}. \tag{2.1}$$

As it turns out, for most materials (para- and diamagnetic but not ferromagnetic materials) the magnetisation is proportional to the magnetic field:

$$\mathbf{M} = \chi_m \mathbf{H},$$

where χ_m is the dimensionless quantity denotes the magnetic susceptibility. It then follows that in a material with magnetic susceptibility χ_m the relation between the magnetic and auxiliary field is given by

$$\mathbf{B} = \mu_0(1 + \chi_m)\mathbf{H} = \mu \mathbf{H}, \tag{2.2}$$

where μ is called the permeability of the material. For magnetic fields outside of magnetic materials, this expression can be simplied to

$$\mathbf{B} = \mu_0 \mathbf{H}. \tag{2.3}$$

This expression is valid for the remainder of this thesis.

For the sake of consistency and clarity, it is noted that in the literature both the $\mathbf{B}$-field and the $\mathbf{H}$-field are often referred to as the *magnetic field*. In this thesis, the nomenclature of [1] is followed: the $\mathbf{B}$-field is referred to as the *magnetic field* and the $\mathbf{H}$-field is referred to as the *auxiliary field*.

The magnetic field of a magnetised object can be analysed using a multipole expansion. The total field is then considered to be a sum of different components: the multipole terms. The first four multipole terms are the monopole, dipole, quadrupole and the octopole. The magnetic monopole has so far not been observed, but would be an isolated north or south pole. The dipole is already described above. From the description of the monopole, it can be seen that a dipole is a combination of two monopoles of different sign. This method of construction is followed for higher order multipoles: a quadrupole is a combination of two dipoles oriented oppositely and an octopole is constructed of two quadrupoles in the same manner.

This analysis method is especially useful when the magnetic field is described using potentials, which is the focus of Section 2.2. Each multipole term can then be described using harmonic functions. This is a central concept in this thesis and is explored in Section 2.3.

2.2. Derivation of scalar potential

In the mathematical analysis of the problem, it is preferable to use a scalar potential as opposed to a vector potential. This preference is due to the apparent ease of dealing with scalars contrary to dealing with vectors. The scalar potential is used as a step in the calculation of the auxiliary field, which, as a more elegant quantity to work with, simplifies some calculations. However, it is important to note that the scalar potential can only be used under certain strict conditions, which will be discussed in this section.

As discussed in Chapter 1, this thesis will be limited to the contribution of the ferromagnetism induced by the Earth's magnetic field to the magnetic signature. Since the Earth's magnetic field can be considered locally uniform, the induced magnetic field will be regarded as time-invariant for the purpose of this thesis. Therefore, the problem is studied within the realm of magnetostatics. Here, the terms in the Maxwell equations with time derivatives are eliminated.

Moreover, the principle of superposition can be applied to electromagnetic fields due to the linearity of the Maxwell equations. This principle dictates that any two fields, say $\mathbf{B}_1$ and $\mathbf{B}_2$, that obey Maxwell's equations, can be summed and the resulting field $\mathbf{B} = \mathbf{B}_1 + \mathbf{B}_2$ also obeys Maxwell's equations. Under this principle, the magnetic field around ships can be decomposed into the Earth's magnetic field and the induced magnetic field. So from now on, the induced magnetic and auxiliary fields are simply referred to as the magnetic and auxiliary fields. All in all, the set of partial differential equations we are concerned with is reduced to:

$$\nabla \cdot \mathbf{B} = 0, \tag{2.4}$$

$$\nabla \times \mathbf{B} = \mu_0 \mathbf{J}. \tag{2.5}$$

In the region where these equations will be evaluated, there are no free currents. If the total current density $\mathbf{J}$ is then written as the sum of the bound currents $\mathbf{J}_b$ and free currents $\mathbf{J}_f$, (2.5) can be written as

$$\nabla \times \mathbf{B} = \mu_0 \mathbf{J}_b + \mu_0 \mathbf{J}_f = \mu_0 \nabla \times \mathbf{M} + \mu_0 \mathbf{J}_f. \tag{2.6}$$

Finally, since the auxiliary field and magnetic field will be evaluated in regions without free currents, it follows that

$$\nabla \times \mathbf{H} = \nabla \times \left(\frac{\mathbf{B}}{\mu_0} - \mathbf{M} \right) = \mathbf{J}_f = \mathbf{0}.$$

By applying Stokes' theorem, it is noted that for any given surface S without free currents,

$$0 = \int_S (\nabla \times \mathbf{H}) \cdot \hat{\mathbf{n}} ds = \int_{\partial S} \mathbf{H} \cdot d\mathbf{l},$$

which implies that $\mathbf{H}$ is a conservative vector field in the absence of free charges. Therefore, $\mathbf{H}$ can be expressed as the gradient of a scalar potential Ψ:

$$\mathbf{H} = -\nabla \Psi. \tag{2.7}$$

The divergence of $\mathbf{H}$ then yields:

$$\nabla^2 \Psi = -\nabla \cdot \mathbf{H} = \nabla \cdot \mathbf{M} - \nabla \cdot \frac{\mathbf{B}}{\mu_0} = \nabla \cdot \mathbf{M}, \tag{2.8}$$

where both the definition of $\mathbf{H}$ from (2.1) and Gauss' law for magnetism from (2.4) were used. The result is Poisson's equation with source term $\nabla \cdot \mathbf{M}$. This formulation of Ψ is reminiscent of the definition of the electric potential V [1, p. 84]:

$$\nabla^2 V = -\rho / \epsilon_0.$$

Here, ρ is the charge density as in (2.4). Analogously, we could introduce magnetic charge density ρ_m [T/m]:

$$\rho_m = -\nabla \cdot \mathbf{M}.$$

Then, (2.8) would read:

$$\nabla^2\Psi = -\rho_m.$$

Even though magnetic charges are not thought to be physical, magnetic charges can help to interpret multipoles as was done in the previous section.

Outside magnetised materials the magnetisation is equal to zero so the Laplace equation is obtained:

$$\nabla^2\Psi = 0. \tag{2.9}$$

The scalar potential Ψ fully defines the magnetic field, because the auxiliary field can be related to the magnetic field with (2.3) The goal is now to find those functions $\Psi(x, y, z)$, such that an expression of the magnetic field can be given.

2.3. Prolate spheroidal harmonic expansion

2.3.1. Prolate spheroidal coordinates

The scalar potential Ψ must satisfy the Laplace equation in (2.9). Solutions of the Laplace equation are called *harmonic functions*, so Ψ can be represented by harmonic functions outside magnetised materials. An analytic expression of the scalar potential can therefore be found for a volume outside a ship.

Since the geometry of ships can be approximated with an ellipsoid, because of their 'cigar' shape, the natural choice of coordinate system is the prolate spheroidal system. Prolate spheroidal coordinates produce ellipsoidal surfaces and this system has an orthogonal set of solutions for the Laplace equation (2.9) as will be discussed in the next subsection. Therefore, if a ship is placed in an ellipsoid, the magnetic field can be modelled outside of this ellipsoid. To express the scalar potential in the largest possible volume, and therefore as close to the ship as possible, this ellipsoid must be the smallest surface enclosing the ship.

The transformation to cartesian coordinates $(\xi, \eta, \phi) \longrightarrow (x, y, z)$ from [9] is used. It is given by

$$x = L\xi\eta, \tag{2.10a}$$

$$y = L\sqrt{(\xi^2 - 1)(1 - \eta^2)}\cos\phi, \tag{2.10b}$$

$$z = L\sqrt{(\xi^2 - 1)(1 - \eta^2)}\sin\phi, \tag{2.10c}$$

with $L > 0$, $\xi \geq 1$, $-1 \leq \eta < 1$ and $0 \leq \phi < 2\pi$.

The focal length of the system is $2L$. For sufficiently large ξ, the ellipsoidal surface described by the coordinates approaches a sphere. This behaviour is useful since for sufficiently large distances a magnetic source can be approximated as a point source and the shape matters less.

The transformation from cartesian coordinates to the prolate spheroidal system for a given value of L is given by

$$\xi = \frac{r_- + r_+}{2L} = \frac{\sqrt{(x + L)^2 + y^2 + z^2} + \sqrt{(x - L)^2 + y^2 + z^2}}{2L}, \tag{2.11a}$$

$$\eta = \frac{r_- - r_+}{2L} = \frac{\sqrt{(x + L)^2 + y^2 + z^2} - \sqrt{(x - L)^2 + y^2 + z^2}}{2L}, \tag{2.11b}$$

$$\phi = \mathrm{Arg}(y + iz). \tag{2.11c}$$

Here, r_- and r_+ are the distances from the foci located on the negative and positive x-axis respectively and $\mathrm{Arg}(y + iz)$ gives the principle angle between the y- and z-coordinate.

A visualisation of prolate spheroidal coordinates is given in Fig. 2.1. For $L = 5$ m, two surfaces are plotted: one for $\xi = 1.5$ and one for $\eta = 0.5$. It is noted that it is common practice to orient the coordinate system in the context of magnetic ranging such that the horizontal plane at $z = 0$ m coincides with the water surface and such that the positive z-axis points into the water. This makes no difference mathematically.

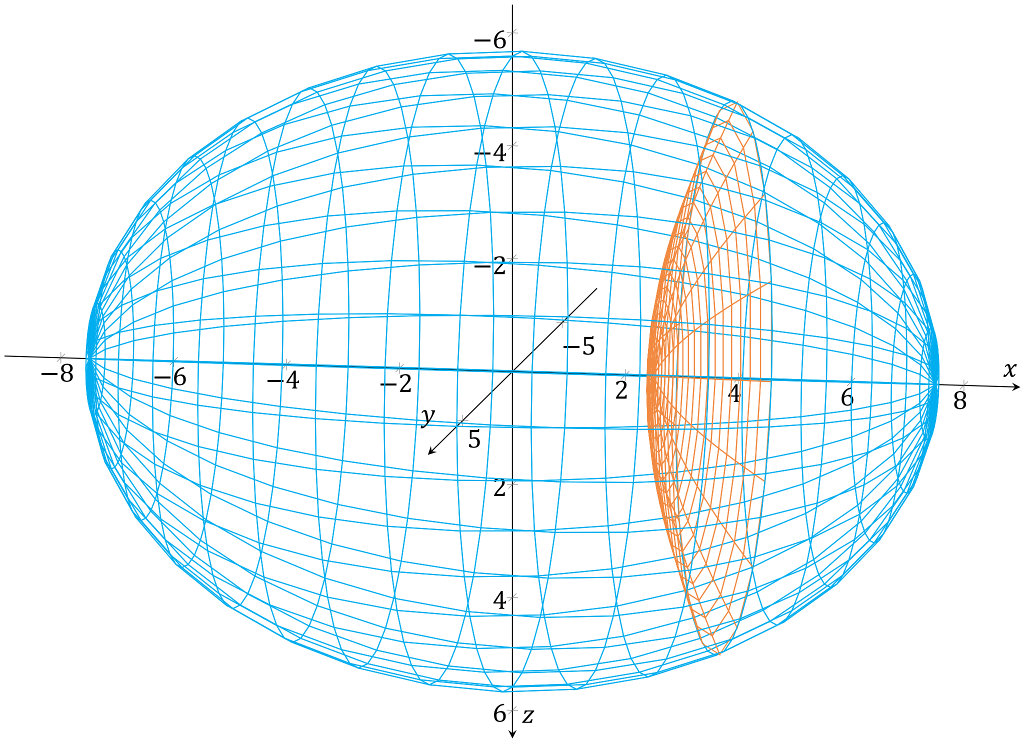


Figure 2.1: Visualisation of prolate spheroidal coordinates with $L = 5$ m. The blue surface is described by $(1.5, \eta, \phi)$ and the orange surface by $(\xi, 0.5, \phi)$ and the path drawn in dark blue ends at the point $(1.5, 0.5, 0.5)$.

To find the divergence and Laplacian of the scalar potential, these two operators need to be expressed in the prolate spheroidal coordinates defined above. Prolate spheroidal coordinates are a type of curvilinear coordinates, which means that the general procedure of finding the necessary expression of the needed operators from [1] can be followed. The gradient of a function f is given by

$$\nabla f = \frac{1}{h_\xi}\frac{\partial f}{\partial \xi}\hat{\boldsymbol{\xi}} + \frac{1}{h_\eta}\frac{\partial f}{\partial \eta}\hat{\boldsymbol{\eta}} + \frac{1}{h_\phi}\frac{\partial f}{\partial \phi}\hat{\boldsymbol{\phi}},$$

where h_u and $\hat{\mathbf{u}}$ denote the scale factor and the unit vector for a coordinate u, respectively. The position vector $\mathbf{r}$ is given by

$$\mathbf{r} = L\begin{pmatrix} \xi\eta \\ \sqrt{(\xi^2-1)(1-\eta^2)}\cos\phi \\ \sqrt{(\xi^2-1)(1-\eta^2)}\sin\phi \end{pmatrix}$$

and its derivative with respect to a coordinate u has length magnitude h_u and direction $\hat{\mathbf{u}}$. So from the partial derivatives of $\mathbf{r}$ with respect to each coordinate

$$\frac{\partial \mathbf{r}}{\partial \xi} = L\begin{pmatrix} \eta \\ \xi\sqrt{\frac{1-\eta^2}{\xi^2-1}}\cos\phi \\ \xi\sqrt{\frac{1-\eta^2}{\xi^2-1}}\sin\phi \end{pmatrix}, \quad \frac{\partial \mathbf{r}}{\partial \eta} = L\begin{pmatrix} \xi \\ -\eta\sqrt{\frac{\xi^2-1}{1-\eta^2}}\cos\phi \\ -\eta\sqrt{\frac{\xi^2-1}{1-\eta^2}}\sin\phi \end{pmatrix}, \quad \frac{\partial \mathbf{r}}{\partial \phi} = L\sqrt{(\xi^2-1)(1-\eta^2)}\begin{pmatrix} 0 \\ -\sin\phi \\ \cos\phi \end{pmatrix},$$

the scale factors and unit vectors are obtained:

$$h_\xi = L\sqrt{\eta^2 + \frac{\xi^2(1-\eta^2)}{\xi^2-1}} = L\sqrt{\frac{\xi^2-\eta^2}{\xi^2-1}}, \quad \hat{\boldsymbol{\xi}} = \sqrt{\frac{\xi^2-1}{\xi^2-\eta^2}}\begin{pmatrix} \eta \\ \xi\sqrt{\frac{1-\eta^2}{\xi^2-1}}\cos\phi \\ \xi\sqrt{\frac{1-\eta^2}{\xi^2-1}}\sin\phi \end{pmatrix} = \begin{pmatrix} \eta\sqrt{\frac{\xi^2-1}{\xi^2-\eta^2}} \\ \xi\sqrt{\frac{1-\eta^2}{\xi^2-\eta^2}}\cos\phi \\ \xi\sqrt{\frac{1-\eta^2}{\xi^2-\eta^2}}\sin\phi \end{pmatrix};$$

$$h_\eta = L\sqrt{\xi^2 + \frac{\eta(\xi^2-1)}{1-\eta^2}} = L\sqrt{\frac{\xi^2-\eta^2}{1-\eta^2}}, \qquad \hat{\boldsymbol{\eta}} = \sqrt{\frac{1-\eta^2}{\xi^2-\eta^2}}\begin{pmatrix} \xi \\ -\eta\sqrt{\frac{\xi^2-1}{1-\eta^2}}\cos\phi \\ -\eta\sqrt{\frac{\xi^2-1}{1-\eta^2}}\sin\phi \end{pmatrix} = \begin{pmatrix} \xi\sqrt{\frac{1-\eta^2}{\xi^2-\eta^2}} \\ -\eta\sqrt{\frac{\xi^2-1}{\xi^2-\eta^2}}\cos\phi \\ -\eta\sqrt{\frac{\xi^2-1}{\xi^2-\eta^2}}\sin\phi \end{pmatrix};$$

$$h_\phi = L\sqrt{(\xi^2-1)(1-\eta^2)}, \qquad \hat{\boldsymbol{\phi}} = \begin{pmatrix} 0 \\ -\sin\phi \\ \cos\phi \end{pmatrix}.$$

The gradient of a function f in prolate spheroidal coordinates is then given by

$$\nabla f = \frac{1}{L}\left(\hat{\boldsymbol{\xi}}\sqrt{\frac{\xi^2-1}{\xi^2-\eta^2}}\frac{\partial f}{\partial \xi} + \hat{\boldsymbol{\eta}}\sqrt{\frac{1-\eta^2}{\xi^2-\eta^2}}\frac{\partial f}{\partial \eta} + \hat{\boldsymbol{\phi}}\frac{1}{\sqrt{(\xi^2-1)(1-\eta^2)}}\frac{\partial f}{\partial \phi}\right).$$

The Laplacian of a function f is given by

$$\begin{aligned} \nabla^2 f &= \frac{1}{h_\xi h_\eta h_\phi}\left[\frac{\partial}{\partial \xi}\left(\frac{h_\eta h_\phi}{h_\xi}\frac{\partial f}{\partial \xi}\right) + \frac{\partial}{\partial \eta}\left(\frac{h_\xi h_\phi}{h_\eta}\frac{\partial f}{\partial \eta}\right) + \frac{\partial}{\partial \phi}\left(\frac{h_\xi h_\eta}{h_\phi}\frac{\partial f}{\partial \phi}\right)\right] \\ &= \frac{1}{L^2(\xi^2-\eta^2)}\left[\frac{\partial}{\partial \xi}\left((\xi^2-1)\frac{\partial f}{\partial \xi}\right) + \frac{\partial}{\partial \eta}\left((1-\eta^2)\frac{\partial f}{\partial \eta}\right) + \frac{\partial}{\partial \phi}\left(\frac{\xi^2-\eta^2}{(\xi^2-1)(1-\eta^2)}\frac{\partial f}{\partial \phi}\right)\right] \end{aligned} \tag{2.12}$$

2.3.2. Prolate spheroidal harmonics

The method of separation of variables will be applied to find the solution Ψ to the Laplace equation (2.9) in prolate spheroidal coordinates (ξ, η, ϕ). This method requires the partial differential equation to be linear and homogeneous with linear homogeneous boundary conditions.

The partial differential equation (PDE) in (2.9) expressed in prolate spheroidal coordinates using the expression of the Laplacian in (2.12) is given by

$$0 = \frac{1}{L^2(\xi^2-\eta^2)}\left[\frac{\partial}{\partial \xi}\left((\xi^2-1)\frac{\partial \Psi}{\partial \xi}\right) + \frac{\partial}{\partial \eta}\left((1-\eta^2)\frac{\partial \Psi}{\partial \eta}\right) + \frac{\partial}{\partial \phi}\left(\frac{\xi^2-\eta^2}{(\xi^2-1)(1-\eta^2)}\frac{\partial \Psi}{\partial \phi}\right)\right]. \tag{2.13}$$

This PDE is clearly linear in Ψ and homogeneous, which follows from the linearity of the partial derivatives and the lack of constant terms.

The boundary conditions have not been explicitly stated yet. Firstly, for a large distance from the origin, it is required that the potential goes to zero. At a large distance from the origin, L is relatively small compared to x, y and z, so from (2.11a) it follows that $\xi \to \infty$ with increasing distance from the origin. The corresponding boundary condition is given in (BC1).

Also, the scalar potential is periodic in ϕ with period 2π, because a rotation of the system of coordinates over 2π gives the original system. In other words, the scalar potential translated over a value $k \cdot 2\pi$ in the ϕ-coordinate must result in the same function value, where k can be any integer. This periodicity condition is formulated in (BC2). Furthermore, for physical reason it required that Ψ is bounded for $\xi > 1$.

The boundary conditions for (2.13) are as follows:

$$\Psi \to 0 \qquad \text{as } \xi \to \infty; \tag{BC1}$$

$$\Psi(\xi, \eta, \phi) = \Psi(\xi, \eta, \phi + k \cdot 2\pi) \qquad k \in \mathbb{Z}. \tag{BC2}$$

The conditions for separation of variables are thus satisfied.

The following substitution is made in (2.13):

$$\Psi(\xi, \eta, \phi) = \Xi(\xi)\mathrm{H}(\eta)\Phi(\phi). \tag{2.14}$$

Next, after dividing the result of the substitution by Ψ, the partial differential equation to solve reads

$$\frac{1}{\Xi}\frac{\mathrm{d}}{\mathrm{d}\xi}\left((\xi^2-1)\frac{\mathrm{d}\Xi}{\mathrm{d}\xi}\right)+\frac{1}{\mathrm{H}}\frac{\mathrm{d}}{\mathrm{d}\eta}\left((1-\eta^2)\frac{\mathrm{dH}}{\mathrm{d}\eta}\right)+\frac{\xi^2-\eta^2}{(\xi^2-1)(1-\eta^2)}\frac{1}{\Phi}\frac{\mathrm{d}^2\Phi}{\mathrm{d}\phi^2}=0$$
$$\frac{(\xi^2-1)(1-\eta^2)}{\xi^2-\eta^2}\left[\frac{1}{\Xi}\frac{\mathrm{d}}{\mathrm{d}\xi}\left((\xi^2-1)\frac{\mathrm{d}\Xi}{\mathrm{d}\xi}\right)+\frac{1}{\mathrm{H}}\frac{\mathrm{d}}{\mathrm{d}\eta}\left((1-\eta^2)\frac{\mathrm{dH}}{\mathrm{d}\eta}\right)\right]=-\frac{1}{\Phi}\frac{\mathrm{d}^2\Phi}{\mathrm{d}\phi^2}=\lambda.$$

The differential equation

$$\frac{\mathrm{d}^2\Phi}{\mathrm{d}\phi^2}=-\lambda\Phi$$

is investigated first. To have physical meaning, Φ is assumed to be a real function, so λ is assumed to be real too. If $\lambda<0$, then $\Phi(\phi)=A_\lambda \mathrm{e}^{\sqrt{-\lambda}\phi}+B_\lambda \mathrm{e}^{-\sqrt{-\lambda}\phi}$ solves the differential equation. However, since Ψ must be periodic in ϕ, Φ must be too. Only the trivial solution where $A=B=0$ solves the differential equation when $\lambda<0$.

For $\lambda=0$, the solution takes the form $\Phi(\phi)=A_\lambda\phi+B_\lambda$. Again, $A_\lambda=0$ because of the periodicity, but B_λ can take any value. Finally, if $\lambda>0$, the differential equation has a solution of the form $\Phi(\phi)=A_\lambda\sin(\sqrt{\lambda}\phi)+B_\lambda\cos(\sqrt{\lambda}\phi)$. It is noted that this solution is also valid if $\lambda=0$, since $\sin(0)=0$ and $\cos(0)=1$. The imposed periodicity condition requires that $\sqrt{\lambda}\in\mathbb{Z}$, since the sine and cosine functions are periodic with period 2π so the scaling factor $\sqrt{\lambda}$ over ϕ must be by an integer. The value $\sqrt{\lambda}$ is then written as $\sqrt{\lambda}=m$. It follows that

$$\Phi(\phi)=s_n^m\sin m\phi+c_n^m\cos m\phi.$$

The use of the superscripts and subscripts will become apparent later.

The inverse of the prefactor in the left-hand side of the partial differential equation can be rewritten as $\frac{\xi^2-\eta^2}{(\xi^2-1)(1-\eta^2)}=\frac{1}{\xi^2-1}+\frac{1}{1-\eta^2}$. The radial and azimuthal part of the differential equation can now be separated as follows:

$$\frac{1}{\Xi}\frac{\mathrm{d}}{\mathrm{d}\xi}\left((\xi^2-1)\frac{\mathrm{d}\Xi}{\mathrm{d}\xi}\right)-\frac{m^2}{\xi^2-1}=-\frac{1}{\mathrm{H}}\frac{\mathrm{d}}{\mathrm{d}\eta}\left((1-\eta^2)\frac{\mathrm{dH}}{\mathrm{d}\eta}\right)+\frac{m^2}{1-\eta^2}=\nu,$$

which results in the following two differential equations:

$$(1-\xi^2)\frac{\mathrm{d}^2\Xi}{\mathrm{d}\xi^2}-2\xi\frac{\mathrm{d}\Xi}{\mathrm{d}\xi}+\left(\nu-\frac{m^2}{1-\xi^2}\right)\Xi=0,\text{ and}$$
$$(1-\eta^2)\frac{\mathrm{d}^2\mathrm{H}}{\mathrm{d}\eta^2}-2\eta\frac{\mathrm{dH}}{\mathrm{d}\eta}+\left(\nu-\frac{m^2}{1-\eta^2}\right)\mathrm{H}=0.$$

These are both Legendre equations which are only stable if ν is of the form $n(n+1)$ [10, p. 338-339]. Legendre equations have solutions P_n^m and Q_n^m, the associated Legendre functions of the first and second kind respectively of the n-th degree and m-th order [9, p. 322].

The Legendre equation has singular points at ±1. The associated Legendre function $\mathrm{P}_n^m(x)$ of the first kind is bounded on the interval $x\in[-1,1]$, while the associated Legendre function of the second kind $\mathrm{Q}_n^m(x)$ is unbounded at $x=\pm1$. Using the definition from [9, p. 322], $\mathrm{P}_n^m(x)$ is defined for $|1-x^2|<2$ and $\mathrm{Q}_n^m(x)$ for $|x|>1$. The associated Legendre functions of the first and second kind are used as the azimuthal and radial functions from (2.14), respectively.

Now that the appropriate functions have been found, the scalar potential can be expressed as follows:

$$\Psi=\sum_{n=1}^{\infty}\sum_{m=0}^{n}\mathrm{Q}_n^m(\xi)\mathrm{P}_n^m(\eta)\left[c_n^m\cos(m\phi)+s_n^m\sin(m\phi)\right]. \tag{2.15}$$

The coefficients c_n^m and s_n^m are the multipole expansion coefficients for each degree and order. The zeroth degree relates to the monopole term, the first degree to the dipole term, etc. The orders relate to the orientation of each multipole.

Note that all s_m^0 are multiplied by 0 since $\sin m\phi = 0$ for $m = 0$. Moreover, c_0^0 represents the magnetic monopole, which has not been found in nature [1, p. 253]. Therefore, the sum over n begins at $n = 1$.

An expression for the magnetic field $\mathbf{B}$ in terms of the multipole coefficients c_n^m and s_n^m can be found by substituting the expression of the scalar potential Ψ given in (2.15) in (2.7). This gives

$$\begin{aligned}
\mathbf{B} = \mu_0 \mathbf{H} &= -\frac{\mu_0}{h_\xi}\frac{\partial \Psi}{\partial \xi}\hat{\boldsymbol{\xi}} - \frac{\mu_0}{h_\eta}\frac{\partial \Psi}{\partial \eta}\hat{\boldsymbol{\eta}} - \frac{\mu_0}{h_\phi}\frac{\partial \Psi}{\partial \phi}\hat{\boldsymbol{\phi}} \\
&= -\mu_0 \frac{\hat{\boldsymbol{\xi}}}{L}\sqrt{\frac{\xi^2-1}{\xi^2-\eta^2}} \sum_{n=1}^{\infty}\sum_{m=0}^{n} \mathrm{Q'}_n^m(\xi)\mathrm{P}_n^m(\eta)\,[c_n^m \cos(m\phi) + s_n^m \sin(m\phi)] \\
&\quad - \mu_0 \frac{\hat{\boldsymbol{\eta}}}{L}\sqrt{\frac{1-\eta^2}{\xi^2-\eta^2}} \sum_{n=1}^{\infty}\sum_{m=0}^{n} \mathrm{Q}_n^m(\xi)\mathrm{P'}_n^m(\eta)\,[c_n^m \cos(m\phi) + s_n^m \sin(m\phi)] \\
&\quad - \mu_0 \frac{\hat{\boldsymbol{\phi}}}{L}\frac{1}{\sqrt{(\xi^2-1)(1-\eta^2)}} \sum_{n=1}^{\infty}\sum_{m=0}^{n} m\mathrm{Q}_n^m(\xi)\mathrm{P}_n^m(\eta)\,[s_n^m \cos(m\phi) - c_n^m \sin(m\phi)]\,.
\end{aligned} \tag{2.16}$$

.

3

Forward Problem

In the previous chapter, the theoretical basis for the model was developed. The relevant magnetostatic theory and the prolate spheroidal coordinate system were investigated and expressions for the magnetic field and scalar potential in terms of prolate spheroidal harmonics were found. In this Chapter, the exact problem is formulated.

The research aim is to create a prediction model of the magnetic field underneath a source, based on field measurements above the source. The model will be split in two parts. Firstly, the magnetic source will be described based on the set of field measurements. The field underneath the ship will then be predicted based on the description of the magnetic source.

Interestingly, the two steps are each others' inverses: the first step takes field measurements and produces a description of the source, while the second step uses a description of the source to predict the magnetic field. These two will be called the forward and inverse problems.

3.1. Scalar potential

In the forward direction, the objective is to find the magnetic field vector from a known magnetic source at either one location or at a set of locations. One of the assumptions made in section 2.2 was that in the evaluated region no free currents exist. This means that the set of positions for which the field will be predicted, cannot contain positions that contain free currents and thus lie outside the encapsulating ellipsoid.

In Chapter 2 it was shown that any magnetic source inside an ellipsoid can be represented outside the ellipsoid by the complete set of multipole coefficients c_n^m and s_n^m. Since the far-field is adequately described by lower order terms, a sufficiently precise prediction may be achieved by terminating the infinite sum. The sum over the Legendre functions from (2.15) is terminated at degree $n = N$ and the order may not exceed $m = M$. Since the largest order must also be lesser than or equal to the degree n, the order m can take values $m = 0, \ldots, \min(n, M)$. The terminated scalar potential $\Psi_{N,M}$ is thus given by

$$\Psi_{N,M}(\xi, \eta, \phi) = \sum_{n=0}^{N} \sum_{m=0}^{\min(n,M)} \mathrm{Q}_n^m(\xi) \mathrm{P}_n^m(\eta) \left[c_n^m \cos(m\phi) + s_n^m \sin(m\phi) \right]. \tag{3.1}$$

The predicted scalar potential at a certain point (ξ, η, ϕ) is a linear combination of the multipole terms weighted with the coefficients. The forward problem is therefore linear in the coefficients c_n^m and s_n^m, and for J observations we can formulate the problem in matrix form:

$$\boldsymbol{\Psi}_{N,M}^{J} = \mathrm{A}_{\Psi} \boldsymbol{\beta}_{N,M}. \tag{3.2}$$

Here $\boldsymbol{\Psi}^{J}_{N,M}$ is the $J \times 1$ vector with J predicted observations of the scalar potential terminated at degree $n \leq N$ and order $m \leq M$.

The multipole coefficients are contained in the vector $\boldsymbol{\beta}_{N,M}$. As discussed in section 2.3.2, the monopole term c_0^0 and the terms s_n^0 are irrelevant for this problem and are therefore eliminated from this vector. The vector is given by

$$\boldsymbol{\beta}_{N,M} = \begin{pmatrix} c_1^0 & c_1^1 & s_1^1 & c_2^0 & c_2^1 & s_2^1 & c_2^2 & \cdots & c_N^M & s_N^M \end{pmatrix}^T . \tag{3.3}$$

The matrix A_Ψ is the matrix with the multipole terms. For each observation Ψ^j at (ξ_j, η_j, ϕ_j) with $j = 0, \ldots, J-1$, the matrix elements $\mathrm{A}^j_{c_n^m}$ and $\mathrm{A}^j_{s_n^m}$ as follows:

$$\begin{aligned}
\mathrm{A}^j_{c_n^m} &= \mathrm{Q}_n^m(\xi_j)\mathrm{P}_n^m(\eta_j)\cos(m\phi_j),\\
\mathrm{A}^j_{s_n^m} &= \mathrm{Q}_n^m(\xi_j)\mathrm{P}_n^m(\eta_j)\sin(m\phi_j).
\end{aligned}$$

The matrix A_Ψ is then

$$\mathrm{A}_\Psi = \begin{pmatrix}
\mathrm{A}^0_{c_1^0} & \cdots & \mathrm{A}^0_{c_n^m} & \mathrm{A}^0_{s_n^m} & \cdots & \mathrm{A}^0_{c_N^M} & \mathrm{A}^0_{s_N^M}\\
\vdots & & \vdots & \vdots & & \vdots & \vdots\\
\mathrm{A}^j_{c_1^0} & \cdots & \mathrm{A}^j_{c_n^m} & \mathrm{A}^j_{s_n^m} & \cdots & \mathrm{A}^j_{c_N^M} & \mathrm{A}^j_{s_N^M}\\
\vdots & & \vdots & \vdots & & \vdots & \vdots\\
\mathrm{A}^{J-1}_{c_1^0} & \cdots & \mathrm{A}^{J-1}_{c_n^m} & \mathrm{A}^{J-1}_{s_n^m} & \cdots & \mathrm{A}^{J-1}_{c_N^M} & \mathrm{A}^{J-1}_{s_N^M}
\end{pmatrix}. \tag{3.4}$$

The matrix has J rows. If $N \leq M$, then the termination of the sum at order M is inconsequential. Each degree n that is evaluated before termination, has a term with coefficient c_n^0 and for each order $0 < m \leq n$, the terms with coefficients c_n^m and s_n^m are added to the sum. The total number of coefficients is then given by $\sum_{n=1}^{N}(2n+1) = N(N+2)$. If $N > M$, then for every degree n that is larger than M, only $2M+1$ coefficients are added, totalling $M(M+2) + (2M+1)(N-M) = -M^2 + 2NM + N + M$. The total number of coefficients k is given by

$$k = \begin{cases} N(N+2), & N \leq M\\ -M^2 + 2NM + N + M, & N > M \end{cases}. \tag{3.5}$$

The matrix A_Ψ is therefore a $J \times k$-matrix.

3.2. Magnetic field

The same procedure can be followed for the magnetic field $\mathbf{B}$. The terminated prediction of the magnetic field is given by

$$\begin{aligned}
\mathbf{B}_{N,M} = &-\mu_0 \frac{\hat{\boldsymbol{\xi}}}{L}\sqrt{\frac{\xi^2-1}{\xi^2-\eta^2}} \sum_{n=0}^{N} \sum_{m=0}^{\min(n,M)} \mathrm{Q'}_n^m(\xi)\mathrm{P}_n^m(\eta)\left[c_n^m\cos(m\phi) + s_n^m\sin(m\phi)\right]\\
&-\mu_0 \frac{\hat{\boldsymbol{\eta}}}{L}\sqrt{\frac{1-\eta^2}{\xi^2-\eta^2}} \sum_{n=0}^{N} \sum_{m=0}^{\min(n,M)} \mathrm{Q}_n^m(\xi)\mathrm{P'}_n^m(\eta)\left[c_n^m\cos(m\phi) + s_n^m\sin(m\phi)\right]\\
&-\mu_0 \frac{\hat{\boldsymbol{\phi}}}{L}\frac{1}{\sqrt{(\xi^2-1)(1-\eta^2)}} \sum_{n=0}^{N} \sum_{m=0}^{\min(n,M)} m\mathrm{Q}_n^m(\xi)\mathrm{P}_n^m(\eta)\left[s_n^m\cos(m\phi) - c_n^m\sin(m\phi)\right].
\end{aligned} \tag{3.6}$$

However, since the $\mathbf{B}$-field is a vector field, each directional component is treated as a separate observation.

With the definition of the unit vectors, each matrix element for the cartesian directions can be calculated. They are given as follows:

$$\mathrm{A}^j_{x,c_n^m} = -\frac{\eta_j}{L}\frac{\xi^2-1}{\xi^2-\eta^2}\mathrm{Q'}^m_n(\xi_j)\mathrm{P}^m_n(\eta_j)\cos(m\phi_j) - \frac{\xi_j}{L}\frac{1-\eta^2}{\xi^2-\eta^2}\mathrm{Q}^m_n(\xi_j)\mathrm{P'}^m_n(\eta_j)\cos(m\phi_j)$$

$$\mathrm{A}^j_{x,s_n^m} = -\frac{\eta_j}{L}\frac{\xi^2-1}{\xi^2-\eta^2}\mathrm{Q'}^m_n(\xi_j)\mathrm{P}^m_n(\eta_j)\sin(m\phi_j) - \frac{\xi_j}{L}\frac{1-\eta^2}{\xi^2-\eta^2}\mathrm{Q}^m_n(\xi_j)\mathrm{P'}^m_n(\eta_j)\sin(m\phi_j)$$

$$\begin{aligned}\mathrm{A}^j_{y,c_n^m} = &-\frac{\xi_j}{L}\frac{\sqrt{(\xi^2-1)(1-\eta^2)}}{\xi^2-\eta^2}\cos\phi_j\mathrm{Q'}^m_n(\xi_j)\mathrm{P}^m_n(\eta_j)\cos(m\phi_j)\\ &+\frac{\eta_j}{L}\frac{\sqrt{(\xi^2-1)(1-\eta^2)}}{\xi^2-\eta^2}\cos\phi_j\mathrm{Q}^m_n(\xi_j)\mathrm{P'}^m_n(\eta_j)\cos(m\phi_j)\\ &-\frac{1}{L\sqrt{(\xi^2-1)(1-\eta^2)}}\sin\phi_j m\mathrm{Q}^m_n(\xi)\mathrm{P}^m_n(\eta)\sin(m\phi_j)\end{aligned}$$

$$\begin{aligned}\mathrm{A}^j_{y,s_n^m} = &-\frac{\xi_j}{L}\frac{\sqrt{(\xi^2-1)(1-\eta^2)}}{\xi^2-\eta^2}\cos\phi_j\mathrm{Q'}^m_n(\xi_j)\mathrm{P}^m_n(\eta_j)\sin(m\phi_j)\\ &+\frac{\eta_j}{L}\frac{\sqrt{(\xi^2-1)(1-\eta^2)}}{\xi^2-\eta^2}\cos\phi_j\mathrm{Q}^m_n(\xi_j)\mathrm{P'}^m_n(\eta_j)\sin(m\phi_j)\\ &+\frac{1}{L\sqrt{(\xi^2-1)(1-\eta^2)}}\sin\phi_j m\mathrm{Q}^m_n(\xi)\mathrm{P}^m_n(\eta)\cos(m\phi_j)\end{aligned}$$

$$\begin{aligned}\mathrm{A}^j_{z,c_n^m} = &-\frac{\xi_j}{L}\frac{\sqrt{(\xi^2-1)(1-\eta^2)}}{\xi^2-\eta^2}\sin\phi_j\mathrm{Q'}^m_n(\xi_j)\mathrm{P}^m_n(\eta_j)\cos(m\phi_j)\\ &+\frac{\eta_j}{L}\frac{\sqrt{(\xi^2-1)(1-\eta^2)}}{\xi^2-\eta^2}\sin\phi_j\mathrm{Q}^m_n(\xi_j)\mathrm{P'}^m_n(\eta_j)\cos(m\phi_j)\\ &+\frac{1}{L\sqrt{(\xi^2-1)(1-\eta^2)}}\cos\phi_j m\mathrm{Q}^m_n(\xi)\mathrm{P}^m_n(\eta)\sin(m\phi_j)\end{aligned}$$

$$\begin{aligned}\mathrm{A}^j_{z,s_n^m} = &-\frac{\xi_j}{L}\frac{\sqrt{(\xi^2-1)(1-\eta^2)}}{\xi^2-\eta^2}\sin\phi_j\mathrm{Q'}^m_n(\xi_j)\mathrm{P}^m_n(\eta_j)\sin(m\phi_j)\\ &+\frac{\eta_j}{L}\frac{\sqrt{(\xi^2-1)(1-\eta^2)}}{\xi^2-\eta^2}\sin\phi_j\mathrm{Q}^m_n(\xi_j)\mathrm{P'}^m_n(\eta_j)\sin(m\phi_j)\\ &-\frac{1}{L\sqrt{(\xi^2-1)(1-\eta^2)}}\cos\phi_j m\mathrm{Q}^m_n(\xi)\mathrm{P}^m_n(\eta)\cos(m\phi_j).\end{aligned}$$

The matrix $\mathrm{A}_{\mathbf{B}}$ is then given by

$$\mathrm{A}_{\mathbf{B}} = \mu_0\begin{pmatrix}\mathrm{A}^0_{x,c_1^0} & \cdots & \mathrm{A}^0_{x,c_n^m} & \mathrm{A}^0_{x,s_n^m} & \cdots & \mathrm{A}^0_{x,c_N^M} & \mathrm{A}^0_{x,s_N^M}\\ \mathrm{A}^0_{y,c_1^0} & \cdots & \mathrm{A}^0_{y,c_n^m} & \mathrm{A}^0_{y,s_n^m} & \cdots & \mathrm{A}^0_{y,c_N^M} & \mathrm{A}^0_{y,s_N^M}\\ \mathrm{A}^0_{z,c_1^0} & \cdots & \mathrm{A}^0_{z,c_n^m} & \mathrm{A}^0_{z,s_n^m} & \cdots & \mathrm{A}^0_{z,c_N^M} & \mathrm{A}^0_{z,s_N^M}\\ \vdots & & \vdots & \vdots & & \vdots & \vdots\\ \mathrm{A}^{J-1}_{z,c_1^0} & \cdots & \mathrm{A}^{J-1}_{z,c_n^m} & \mathrm{A}^{J-1}_{z,s_n^m} & \cdots & \mathrm{A}^{J-1}_{z,c_N^M} & \mathrm{A}^{J-1}_{z,s_N^M}\end{pmatrix} \tag{3.7}$$

and the problem in matrix form is formulated as

$$\mathbf{B}^J_{N,M} = \mathrm{A}_{\mathbf{B}}\boldsymbol{\beta}_{N,M}. \tag{3.8}$$

The matrix $\mathrm{A}_{\mathbf{B}}$ is a $3J \times k$-matrix, where k is given in (3.5). The vector of coefficients is the same vector as for the forward problem for the scalar potential given in (3.3). As noted earlier, the prediction vector $\mathbf{B}^J_{N,M}$ is defined as

$$\mathbf{B}^J_{N,M} = \begin{pmatrix} \mathrm{B}^0_x & \mathrm{B}^0_y & \mathrm{B}^0_z & \cdots & \mathrm{B}^j_x & \mathrm{B}^j_y & \mathrm{B}^j_z & \cdots & \mathrm{B}^{J-1}_x & \mathrm{B}^{J-1}_y & \mathrm{B}^{J-1}_z \end{pmatrix}^T . \tag{3.9}$$

Forward problem

Given is a set of J points (ξ_j, η_j, ϕ_j) in a prolate spheroidal coordinate system with a focal length $2L$ outside an ellipsoid encapsulating the source. Furthermore, the vector of specified multipole coefficients $\boldsymbol{\beta}_{N,M}$ is known.

Find the auxiliary field vector $\mathbf{B}^J_{N,M}$ for the J points as defined in (3.9) with the source specified by $\boldsymbol{\beta}_{N,M}$.

The magnetic field $\mathbf{B}$ in a plane at $z = 15$ m is visualised in Fig. 3.1 for a dipole with $c^0_1 = 1$ and all other coefficients set to 0 and $L = 50$ m. Plots of other multipole configurations are included in Appendix A in Fig. A.1 to A.5. For each configuration, each component and the magnitude of the total field are given in a contour plot.

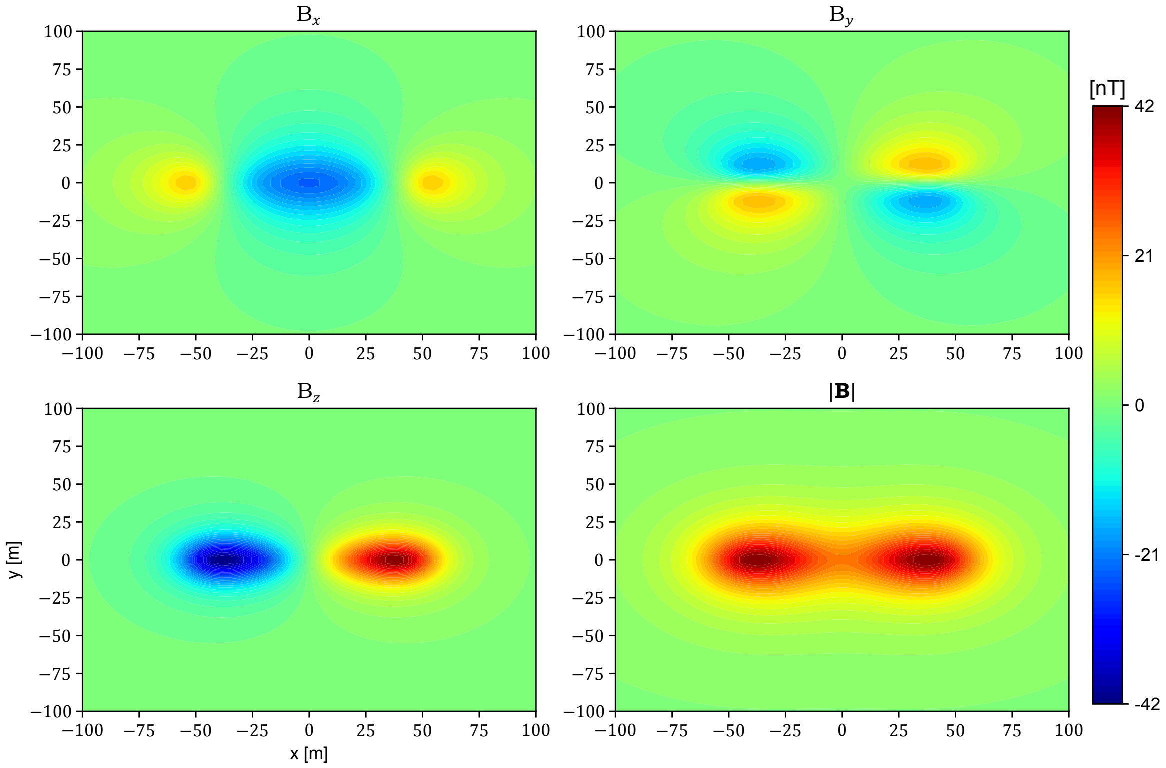


Figure 3.1: Contour plot of the x-, y- and z-components and the field strength of the magnetic field $\mathbf{B}$ at $z = 15$ m and -100 m $\leq x, y \leq 100$ m generated by a source with $c^0_1 = 1$ and all other coefficients set to 0 and $L = 50$ m.

4

Inverse Problem

4.1. Definition of the inverse problem

The objective of the inverse problem is to find the coefficients that describe the source adequately from field measurements. For the purpose of generality, the coordinates at which the field measurements are taken are given in cartesian coordinates. The extra step of finding the optimal parameters defining a prolate spheroidal coordinate system is therefore added to the problem. These parameters are the centre point (x_0, y_0, z_0) and the focal length L. It is assumed that the foci of the prolate spheroidal coordinate system are positioned on the x-axis of the cartesian coordinate system, thus there is no parameter specifying the orientation of prolate spheroidal system needed.

The set of field measurements $\{\mathbf{B}_j | j = 0, \dots, J-1\}$ is a set of vectors. The same transformation as in (3.9) is used such that the vector $\mathbf{B}^J$ is the vector with all field measurements as components and is defined by

$$\mathbf{B}^J = \begin{pmatrix} \mathrm{B}_x^0 & \mathrm{B}_y^0 & \mathrm{B}_z^0 & \cdots & \mathrm{B}_x^j & \mathrm{B}_y^j & \mathrm{B}_z^j & \cdots & \mathrm{B}_x^{J-1} & \mathrm{B}_y^{J-1} & \mathrm{B}_z^{J-1} \end{pmatrix}^T. \tag{4.1}$$

The objective is to find an accurate representation of the source given by the vector $\boldsymbol{\beta}$ containing the set of predicted coefficients.

Moreover, since the number of coefficients describing the source is unknown, the number of coefficients for which the source is adequately described by the model in the regions that are of interest needs to be determined. For now it is assumed that the optimal number of coefficients is k, given by (3.5) for a certain N and M. The modelled field is then represented by $\mathbf{B}$.

Inverse problem

Given is a set of J positions (x_j, y_j, z_j) outside an ellipsoid encapsulating the magnetic source and a collection of magnetic field measurements $\mathbf{B}_j$.

Find the parameters L, x_0, y_0 and z_0 and the vector $\boldsymbol{\beta}$ of multipole coefficients c_m^n and s_m^n as defined in (3.3).

4.2. Well-posedness

While the forward problem formulated in the previous chapter can be solved with relatively straightforward methods, the inverse problem cannot. The reason for this can be found in the nature of inverse problems: only observations are known and all conditions contributing to these observations must be found. It might, for example, not be known whether a solution exists and, if it does exist, whether it is unique.

These concepts relate to the quality of a problem, which is referred to as the well-posedness of a problem. For any inverse problem of the form

$$\mathrm{A}\boldsymbol{\beta} = \mathbf{B}^J \tag{4.2}$$

where $\mathrm{A} \in \mathbb{R}^{3J \times k}$, $\boldsymbol{\beta} \in \mathbb{R}^k$ and $\mathbf{B}^J \in \mathbb{R}^{3J}$, the mathematician Hadamard qualified it to be well-posed if [11, p. 18]:

1. the solution exists;
2. the solution in unique;
3. $\boldsymbol{\beta}_i \to \boldsymbol{\beta}$ if $\mathbf{B}_i^J \to \mathbf{B}^J$, $\mathrm{A}\boldsymbol{\beta}_i = \mathbf{B}_i^J$ and $\mathrm{A}\boldsymbol{\beta} = \mathbf{B}^J$.

For a linear problem such as (4.2), the first two conditions imply that the matrix A is invertible. More generally, these first two conditions ensure that a problem has a satisfying and unambiguous solution. The third condition is equivalent to the condition that the solution is stable. This ensures that if the inputs are changed slightly, the outputs will not change extremely.

If one of the Hadamard conditions is not satisfied, a problem is called ill-posed. In that case, finding the solution tends to be difficult, but additional conditions can be imposed on the problem to help approximate the solution.

An inverse as in (4.2) can be classified as one of three cases [11, p. 4]. In the first case $3J = k$. This results in a square matrix A. If the columns of the matrix are linearly independent, the matrix is invertible and the problem has the unique solution $\mathrm{A}^{-1}\mathbf{B}^J$. This would be the ideal case. However, due to uncertainties in the measurement data, errors are easily propagated to the solution. Also, the positions of measurements must be chosen such that the resulting matrix has linearly independent columns. Otherwise, the solution is either non-existent or non-unique.

In the second case $3J > k$, which gives an overdetermined system. A solution generally does not exist, but a 'closest' solution might be determined. When the measurements contain errors, regression methods (see Chapter 5) are often employed to find such a closest solution.

In the final case, the system is underdetermined ($3J < k$). The matrix A is then necessarily linearly dependent and if a solution exists, it is non-unique.

The problem described in this thesis is considered ill-posed. Due to possible random errors in the measurement data, the solution $\boldsymbol{\beta}$ must be approximated for a given prolate spheroidal coordinate system. Furthermore, the measured magnetic field might be described in different prolate spheroidal coordinate systems, with each a different solution $\boldsymbol{\beta}$. Different methods are employed to resolve the ill-posedness.

4.3. Overview of methods

In the next chapters, the methods used to solve this inverse problem are discussed. Firstly, the regression method and regularisation is discussed in Chapter 5. This step is used to find stable solutions of the linear system of equations from (4.2). Such a linear system requires a known number of coefficients k. However, since k is unknown in practice, an upper bound on the largest number of coefficients is sought in Chapter 6. Moreover, the focal length L of the coordinate system in Section 2.3.1 is not known a priori. These parameters are found by sampling them at random and estimating the optimal values. This is also discussed in Chapter 6.

These methods produce a set of possible models, from which one is to be selected. This is discussed in Chapter 7. A complete overview of the incorporation of these methods into the model is given in Chapter 8.

5

Regression Method

This chapter is concerned with the regression method. Regression is an umbrella term used to refer to a class of methods that attempt to find the relation between a dependent variable and a set of independent variables. The dependent variable for the problem at hand is the vector of magnetic field measurements $\mathbf{B}^J$. The independent variables for this problem are the individual multipole fields, represented by the elements of the matrix $\mathrm{A}_{\mathbf{B}}$ (from now on, simply referred to as A). Finally the relation between the dependent variable and the independent variables is given by the unknown parameters, which are for this problem the multipole coefficients c_n^m and s_n^m in $\boldsymbol{\beta}$.

5.1. Residuals

The difference between the predicted field $\hat{\mathbf{B}}(x, y, z)$ and the measured field $\mathbf{B}(x, y, z)$ at each position is called a residual. The relation between the field measurements and the predicted multipole coefficients is given by

$$\mathbf{B}^J = \hat{\mathbf{B}}^J + \boldsymbol{\epsilon} = \mathrm{A}\boldsymbol{\beta} + \boldsymbol{\epsilon}. \tag{5.1}$$

Here, $\boldsymbol{\epsilon}$ is the $J \times 1$ vector of residuals. Each component of $\boldsymbol{\epsilon}$ is assumed to be normally distributed with mean 0 and standard deviation σ_j. Furthermore, the residuals are assumed to be caused by two sources: the measurement uncertainty in the field measurements and the uncertainty in the position of the measurement.

The residuals due to the error in the field measurement are assumed to be normally distributed with standard deviations σ_{B_x}, σ_{B_y} and σ_{B_z}. However, as will be discussed in Chapter 9, common magnetometers have equal noise levels for each component of the magnetic field. Therefore, it can be assumed that $\sigma_{\mathrm{B}_x} = \sigma_{\mathrm{B}_y} = \sigma_{\mathrm{B}_z}$, which will be referred to as σ_{B}[1]. Then

$$\epsilon_{\mathrm{B},1}, \epsilon_{\mathrm{B},2}, \dots, \epsilon_{\mathrm{B},3J} \overset{\text{i.i.d.}}{\sim} N(0, \sigma_{\mathrm{B}}^2),$$

where $\epsilon_{\mathrm{B},j}$ represents the residual in the j-th element of $\mathbf{B}^J$ due to the measurement error by the magnetometer.

While the measurement error directly affects the measured value of the magnetic field, the positioning error needs to be propagated to the magnetic field to see the effect: a wrong measurement of the position causes predictions to be made for the wrong points in space. There are two different paths to propagate the positioning error: estimation of the error caused by a wrong position through interpolation

---

[1] The standard deviation of the measurement of each component of the magnetic field σ_{B} is not to be confused with the standard deviation of the magnitude of $\mathbf{B} = (\mathrm{B}_x, \mathrm{B}_y, \mathrm{B}_z)$. The standard deviation of the magnitude is not used in this thesis.

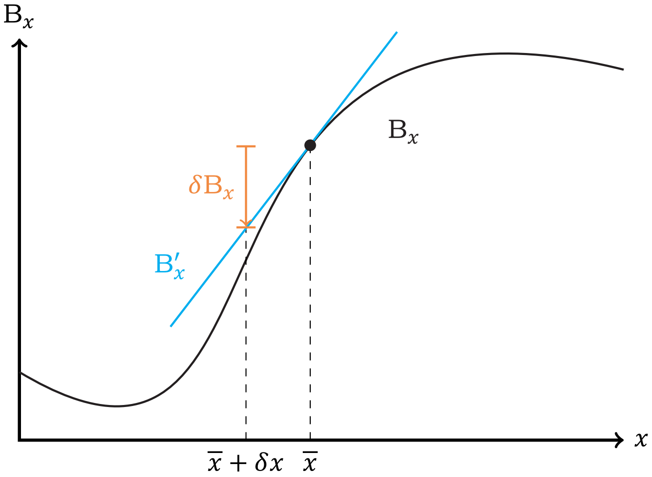


Figure 5.1: Sketch of the propagation of error δx in the measurement of B_x at position $\bar{x}$ using the calculus approach. The function B_x is given in a certain yz-plane and is represented by the black line. The tangent of B_x in point $\bar{x}$ is represented by the cyan line and the propagated error δB_x is given in orange.

of the measurements or estimation of the errors in each matrix element. The former is chosen, because it is simpler.

An example is given to illustrate this method: instead of having measured B_x at a certain position with coordinate $\bar{x}$, the measurement was performed at coordinate $\bar{x}+\delta x$. This results in an error in the measurement equal to

$$\delta B_x = B_x(\bar{x}+\delta x) - B_x(\bar{x}) \approx \delta x \frac{\partial}{\partial x} B_x(\bar{x}).$$

An indicative sketch of this error propagation method is given in fig. 5.1.

This method of propagation of the positional errors is called the *calculus approach*, which can be used for multi-variable functions [12, p. 43]. For the j-th element of $\mathbf{B}^J$, the residual ϵ_j has standard error

$$\sigma_j^2 = \sigma_x^2 \left(\frac{\partial B_j}{\partial x}\right)^2_{x_j,y_j,z_j} + \sigma_y^2 \left(\frac{\partial B_j}{\partial y}\right)^2_{x_j,y_j,z_j} + \sigma_z^2 \left(\frac{\partial B_j}{\partial z}\right)^2_{x_j,y_j,z_j} + \sigma_B^2, \tag{5.2}$$

where B_j is either B_x, B_y or B_z, dependent on which of these the j-th element represents.

If the measurements are performed in a grid in the xy-plane with regular spacing, then the numerical derivative can be used to find the partial derivatives in (5.2). The central difference formulae [13, p. 27] are given by

$$\frac{\partial B_j}{\partial x} \approx \frac{B_j(x+h_x) - B_j(x-h_x)}{2h}, \tag{5.3}$$
$$\frac{\partial B_j}{\partial y} \approx \frac{B_j(y+h_y) - B_j(y-h_y)}{2h}, \tag{5.4}$$

where h_x and h_y are the grid spacing in their respective directions.

Since the grid does not extend into the z-direction, this method cannot be used directly to approximate the partial derivatives with respect to z. However, Maxwell's equations can be used to find these partial derivatives. Firstly, (2.4) produces

$$\frac{\partial B_z}{\partial z} = -\frac{\partial B_x}{\partial x} - \frac{\partial B_y}{\partial y}. \tag{5.5}$$

In regions without magnetised materials and free currents, (2.6) can be reduced to

$$\nabla \times \mathbf{B} = \left(\frac{\partial \mathrm{B}_y}{\partial z} - \frac{\partial \mathrm{B}_z}{\partial y}\right)\hat{\mathbf{x}} - \left(\frac{\partial \mathrm{B}_z}{\partial x} - \frac{\partial \mathrm{B}_x}{\partial z}\right)\hat{\mathbf{y}} - \left(\frac{\partial \mathrm{B}_x}{\partial y} - \frac{\partial \mathrm{B}_y}{\partial x}\right)\hat{\mathbf{z}} = \mathbf{0}.$$

The partial derivatives with respect z can be obtained from the previous relation:

$$\frac{\partial \mathrm{B}_x}{\partial z} = \frac{\partial \mathrm{B}_z}{\partial x} \tag{5.6}$$

$$\frac{\partial \mathrm{B}_y}{\partial z} = \frac{\partial \mathrm{B}_z}{\partial y}. \tag{5.7}$$

The last obstacle that stands in the way of using the numerical derivatives defined above is that the calculation would require the true magnetic field values at the given positions. This might result in a more complex implementation of the methods in this chapter. However, it will be argued that the numerical derivatives based on the measurements instead of the true values are a good enough approximation. To this end, the conditions

$$h_x \gg \sigma_x \quad \text{and} \quad h_y \gg \sigma_y$$

are imposed. These conditions imply that the error in the partial derivatives of the the central difference formulae in (5.3) and (5.4) is mainly due to the size of h_x and h_y and not the error in the position of the measurement of $\mathbf{B}$.

In conclusion, the partial derivatives of each component of the magnetic field $\mathbf{B}$ with respect to the cartesian coordinates can be numerically approximated from a grid of field measurements. Then, the standard errors σ_j^2 in each element B_j of the vector with magnetometer measurement components $\mathbf{B}^J$ can be calculated using (5.2).

5.2. Least squares regression

Since this problem is linear, a linear regression method can be used to find the coefficients. A very common method is the method of ordinary least squares (OLS) which seeks to minimise the residuals [14, p. 564-572]. The goal is to find the vector $\boldsymbol{\beta}$ of given length k for which the function

$$S(\boldsymbol{\beta}) = \|\boldsymbol{\epsilon}\|^2 = \left\|\mathbf{B}^J - \mathrm{A}\boldsymbol{\beta}\right\|^2 = \left(\mathbf{B}^J - \mathrm{A}\boldsymbol{\beta}\right)^T \left(\mathbf{B}^J - \mathrm{A}\boldsymbol{\beta}\right)$$

is minimal. The function $S(\boldsymbol{\beta})$ is minimised by

$$\boldsymbol{\beta} = \left(\mathrm{A}^T\mathrm{A}\right)^{-1}\mathrm{A}^T\mathbf{B}^J$$

provided that A has independent columns. If A is itself invertible, then $\left(\mathrm{A}^T\mathrm{A}\right)^{-1}\mathrm{A}^T = \mathrm{A}^{-1}$. This means that the inverse problem is either solvable using OLS or is ill-posed.

The method of ordinary least squares is, however, too simplistic: this method is biased towards those points with large standard errors in the residuals. By simply minimising the norm of the residuals, the most gain can be obtained from minimising the residuals with large standard deviations, since these tend to have larger values. A weighted least squares fit resolves this problem.

In this method, each residual is normalised as follows

$$\tilde{\epsilon}_j = \frac{\epsilon_j}{\sigma_j} = \frac{\mathrm{B}_j - \hat{\mathrm{B}}_j}{\sigma_j}.$$

The objective function then becomes

$$S(\boldsymbol{\beta}) = \|\tilde{\boldsymbol{\epsilon}}\|^2 = \left(\mathbf{B}^J - \mathrm{A}\boldsymbol{\beta}\right)^T \mathrm{W}\left(\mathbf{B}^J - \mathrm{A}\boldsymbol{\beta}\right),$$

where the matrix W is a diagonal matrix with

$$\mathrm{W}_{jj} = \frac{1}{\sigma_j^2}.$$

This matrix is referred to as the weight matrix.

The method of ordinary least squares and weighted least squares are prone to overfitting. Overfitting can occur when some of the independent variables have no predictive relation with the dependent variable. If more variables are fit than can actually predict the dependent variable, overfitting occurs. This results in a smaller deviation from 0 by the residuals. Since both ordinary and weighted least squares minimise the residuals, these methods are susceptible to overfitting.

An example of overfitting is when linearly dependent data with errors is fit to a quadratic function. Especially for a small sample size, the coefficient describing the quadratic dependence will likely be unequal to zero. In this example, this is a result of the errors and not of an underlying quadratic dependence.

For the problem in this thesis, this happens inevitably: the coefficients that describe the source are unknown and the set of independent variables will therefore necessarily contain variables with no predictive power. The result is that the magnetic field caused by the source is well described in the measurement points, but other points and especially points in other planes might not be well described by the coefficients.

5.3. Regularisation

As mentioned in the previous section, a solution of any inverse problem is either ill-posed or solvable by the method of ordinary (or weighted) least squares. Moreover, these least squares regression methods are susceptible to overfitting. Adding new information to the problem could help solve the ill-posedness or reduce the error due to overfitting. This process is called regularisation.

In the illustrative example of overfitting in the previous section where a quadratic function was fit on linearly dependent data, eliminating the coefficient describing the quadratic dependence would have resolved overfitting. This process is called *variable selection* and can help resolve not only overfitting but also the ill-posedness of an underdetermined system.

For an underdetermined system, the columns of the linear operator (in matrix form) are not independent. Therefore, the uniqueness of the solution required for well-posedness is not achieved. Variable selection reduces the number of columns in the operator and could force uniqueness.

To do this, a regularising function $R(\boldsymbol{\beta})$ is added to the least squares objective function. This function imposes extra information on the solution. The new objective function is

$$S(\boldsymbol{\beta}) = \left(\mathbf{B}^J - \mathrm{A}\boldsymbol{\beta}\right)^T \mathrm{W} \left(\mathbf{B}^J - \mathrm{A}\boldsymbol{\beta}\right) + \alpha R(\boldsymbol{\beta}), \tag{5.8}$$

where the hyperparameter $\alpha \geq 0$ controls the strength of regularisation. Also, the residuals have been weighted by matrix W. If $\alpha = 0$, then the objective function is the same as for WLS (or OLS if the matrix W is the identity matrix). For increasing α, the regularisation function becomes increasingly more dominant.

Usually, the regularisation function takes the form of a norm on the vector space $\mathbb{R}^k$. Two common functions are the L_1- and the squared L_2-norm given by

$$||\boldsymbol{\beta}||_1 = \sum_{i=0}^{k-1} |\beta_i| \tag{5.9a}$$

$$||\boldsymbol{\beta}||_2^2 = \sum_{i=0}^{k-1} |\beta_i|^2. \tag{5.9b}$$

The former is called *Least absolute shrinkage and selection operator* (Lasso) regularisation and the latter is referred to as ridge regularisation. Both methods penalise large fit coefficients, so both tend to set insignificantly contributing coefficients to (approximately) zero.

However, the two methods operate slightly differently. On the one hand, ridge regularisation prefers a model with all medium coefficients over a model with some small and some large coefficients. This is due to the squaring of the absolute value. On the other hand, Lasso regularisation does not favour a small variation in coefficients and tends to set coefficients to exactly zero if they do not contribute significantly.

The regularisation term is added to the objective function to combat overfitting, which might be achieved by either of the two mentioned regularisation functions. Previous research has suggested that the use of Lasso yields better results than ridge regularisation in the context of magnetic ranging [15]. Therefore, it is chosen to use Lasso regularisation.

5.4. Least angle regression

The least angle regression method (LARS) was proposed by [16]. As it will be shown later, LARS is a useful method to implement Lasso regularisation. LARS is a forward method for finding solutions of linear problems. The algorithm works by building up a working solution, or current solution, until a termination condition is reached.

Initially, all coefficients are set to zero and at each step, one coefficient is altered. The set $\mathcal{A}$ contains the indices of the coefficients that have entered the solution and is called the active set.

At each step, the correlation $\hat{c}_i$ of the residuals of the current solution with each column $\mathbf{a}_i$ in the matrix A corresponding to coefficient β_i is calculated. In this section, the matrix A is the weighted matrix, meaning that each row j was divided by the standard deviation σ_j of the measurement of the j-th element in $\mathbf{B}^J$.

The vector of correlations is given by

$$\hat{\mathbf{c}} = \mathrm{A}^T \left(\mathbf{B}^J - \hat{\mathbf{B}}_{\mathcal{A}}\right),$$

where $\hat{\mathbf{B}}_{\mathcal{A}}$ is the working solution, belonging to the active set $\mathcal{A}$. It is noted that this method normally requires the matrix columns to be standardised, such that the mean of each column is 0 and the standard deviation is 1.

However, the removal of the mean of each column is not advisable in the context of magnetic multipoles. LARS was developed in a more statistical context, where an assumption is made that an entire *population* was sampled to produce the measurements. This assumption does not hold in the context of selective sampling of multipoles in a certain plane. Perhaps this assumption would hold if the entire space (or maybe a symmetric region around the source) were to be sampled. Therefore, the mean is not removed from each column, but the columns are scaled such that the standard deviation of the elements around their mean is 0.

The matrix column with the largest correlation with the residuals is then selected and its coefficient is increased until the new residuals have as much correlation with this column as with one of the columns outside the active set. This second coefficient now enters the active set. At this point, instead of continuing in the direction of just the first selected coefficient, the next direction is chosen such that the direction is of equal angle with all columns corresponding to coefficients in the active set. This is referred to as the *least angle direction*. Hence least *angle* regression.

At each step, a new coefficient is added to the active set and all coefficients are increased such that the angles of the corresponding matrix columns are equiangular with the least angle direction. The algorithm from [16, p. 413-414] is given in Algorithm 5.1. Here, the vector

$$\mathbf{u}_{\mathcal{A}} = \mathrm{A}_{\mathcal{A}}\mathbf{w}_{\mathcal{A}}$$

is the equiangular vector in the active column space of unit length. The vector $\mathbf{w}_{\mathcal{A}}$ is a vector of multipole coefficients.

The new solution $\hat{\mathbf{B}}_{\mathcal{A}+}$ is the result of adding a vector $\lambda\mathbf{u}_{\mathcal{A}}$ to the previous solution $\hat{\mathbf{B}}_{\mathcal{A}}$. The scalar value λ determines the size of the multipole coefficients added in the step. The value of λ must be chosen such that a new column becomes active. This choice is described in [16, p. 414] and is reflected in the algorithm.

```
Input: $\mathrm{A}, \mathbf{B}^J$
begin
    $\mathcal{A} = \emptyset$                                         // Active set
    $\hat{\mathbf{B}}_\emptyset = \mathbf{0}$                                      // Current estimate
    while $\mathcal{A}^C \neq \emptyset$ do
        compute $\hat{\mathbf{c}} = \mathrm{A}^T\left(\mathbf{B}^J - \hat{\mathbf{B}}_{\mathcal{A}}\right)$          // Vector of current correlations
        let $\hat{C} = \max_i\{|\hat{c}_i|\}$ and $\mathcal{A} = \{i : |\hat{c}_i| = \hat{C}\}$     // Greatest absolute current
                                                          // correlation and corresponding set
        for each $i \in \mathcal{A}$ let $s_i = \mathrm{sign}\{\hat{c}_i\}$

        $\mathrm{A}_{\mathcal{A}} = (\cdots s_i\mathbf{a}_i \cdots)_{i\in\mathcal{A}}$          // Matrix containing columns of active set
        $g_{\mathcal{A}} = \mathrm{A}_{\mathcal{A}}^T\mathrm{A}_{\mathcal{A}}$ and $H_{\mathcal{A}} = \left(\mathbf{1}_{\mathcal{A}}^T g_{\mathcal{A}}^{-1}\mathbf{1}_{\mathcal{A}}\right)^{-1/2}$, where $\mathbf{1}_{\mathcal{A}}$ is a vector of 1's with length $|\mathcal{A}|$
        $\mathbf{u}_{\mathcal{A}} = \mathrm{A}_{\mathcal{A}}\mathbf{w}_{\mathcal{A}}$ with $\mathbf{w}_{\mathcal{A}} = H_{\mathcal{A}} g_{\mathcal{A}}^{-1}\mathbf{1}_{\mathcal{A}}$
        $\mathbf{h} = \mathrm{A}^T\mathbf{u}_{\mathcal{A}}$
        $\hat{\mathbf{B}}_{\mathcal{A}+} = \hat{\mathbf{B}}_{\mathcal{A}} + \lambda\mathbf{u}_{\mathcal{A}}$ where $\lambda = \min^{+}_{i\in\mathcal{A}^C}\left\{\frac{\hat{C}-\hat{c}_i}{H_{\mathcal{A}}-h_i}, \frac{\hat{C}+\hat{c}_i}{H_{\mathcal{A}}+h_i}\right\}$      // Update solution
```

Algorithm 5.1: Least angle regression

The k-th solution is given by

$$\begin{aligned}\hat{\mathbf{B}}_k &= \lambda_1\mathbf{u}_1 + \cdots + \lambda_k\mathbf{u}_k \\ &= \lambda_1\mathrm{A}_1\mathbf{w}_1 + \cdots + \lambda_k\mathrm{A}_k\mathbf{w}_k.\end{aligned}$$

This is equivalent adding up all vectors w_i with weights λ_i, where elements of inactive columns are set to 0 and then multiplying with A.

The result of this method is a set of $k+1$ solutions from each step in the process, where k is the total number of coefficients. The final solution is the WLS (or OLS) solution. Which of these produced solutions should be chosen is discussed in Chapter 7.

5.5. Lasso-LARS

Under the about to be introduced *Lasso modification* of the original LARS algorithm, which was also introduced by [16, p. 417], the modified algorithm yields all Lasso solutions. Therefore, a modified implementation of the LARS algorithm can be used to find solutions under Lasso regularisation efficiently.

For the mathematical details, one is directed towards section 3 of [16]. Suppose that after a LARS step the working solution $\hat{\mathbf{B}}$ is acquired and that this solution is also a Lasso solution with $\hat{\mathbf{B}} = \mathrm{A}\hat{\boldsymbol{\beta}}$. It can be proven that the sign of $\hat{\beta}_j$ is equal to the sign of $\hat{c}_j$. This not restriction is enforced in the original LARS algorithm. If, however, this restriction is enforced, then all solutions that the modified algorithm yields, are also Lasso solution.

The restriction is implemented in the algorithm as follows: along the path of an increasing γ, if one of the coefficients changes sign, then the current step is stopped at the value of γ where the sign change occurs and the corresponding index is removed from the active set. The sign change would break the restriction that the coefficients and the corresponding correlations are of equal sign, since the correlations do not change sign within one step of the algorithm [16, p. 417].

6

Hyperparameter Optimisation

Before the previously discussed Lasso-LARS algorithm can be applied, certain parameter values must be set. These are referred to as *hyperparameters*. For the Lasso-LARS process, 6 hyperparameters are discerned: the focal length L and the centre point (x_0, y_0, z_0) of the prolate spheroidal coordinate system and the termination order M and degree N in (3.6). While the values of M and N can be selected based on the measurement setup, the focal length and centre point are more difficult to estimate.

6.1. Number of coefficients

Up until now, the problem was posed with the goal to find a vector $\boldsymbol{\beta}_{N,M}$ of length k defined in (3.5) by a given maximum order M and degree N. Both M and N constrain the level of detail that can be modelled. For example, if $N = 1$, then only a dipole field can be modelled from given field measurements. This constraint raises the question of what might be the highest possible level of detail that can be modelled given the locations of measurements (expressed in M and N).

Finding such upper bounds M_{ub} and N_{ub} will reduce overfitting since a limit will be put on the number of coefficients. Also, the size of the problem will be constrained, which will reduce computation time.

Firstly, the measurement grid considered is a rectangular grid at a certain z-coordinate. The grid is specified by the grid spacings δx and δy, the number of grid points in each direction, $\mathrm{num}(x)$ and $\mathrm{num}(y)$, and the bottom left coordinate of the grid $(\tilde{x}, \tilde{y})$. Fig. 6.1 shows a schematic representation of a rectangular grid with $\mathrm{num}(x) = 8$ and $\mathrm{num}(y) = 6$.

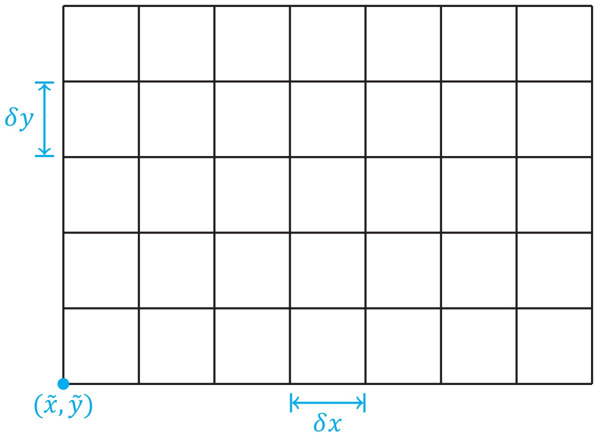


Figure 6.1: Schematic representation of a rectangular measurement grid. The grid spacings δx and δy and the bottom left coordinate $(\tilde{x}, \tilde{y})$ are indicated.

The resolution in the y-direction is related to the order m, because the scalar potential Φ, in (3.1), is essentially a sum of sines and cosines with argument $m\phi$. Sampling a harmonic function without aliasing requires the sampling rate to be twice the highest frequency in the domain. In other words, having $\mathrm{num}(y)$ samples implies that the highest measurable order equals $\mathrm{num}(y)/2$.

However, because the magnetic field, which is the gradient of the scalar potential, is sampled, not one but three measurements are taken at a given grid point. These three measurements are not independent, so only one ambiguity in the phase is removed. The highest measurable order is therefore $\mathrm{num}(y)$.

Central in this thesis are the benefits of performing measurements using a drone. These benefits include the relative ease for taking more measurements. It is, therefore, theoretically possible to extend the measurement grid to an arbitrarily large dimension. In the y-direction, this would result in a large number of distinguishable orders m. However, for larger distances from the source, the associated Legendre functions with higher orders and degrees diminish more rapidly. At a certain point, due to measurement errors, an added row of measurements will not yield the desired increased resolution. Therefore, the highest measurable order must be lower than the theoretical value $\mathrm{num}(y)$ when the grid extends far beyond the ship.

The upper bound on the highest distinguishable order M_{ub} is chosen to be the number of grid points within the ship's dimension in the athwart direction:

$$M_{\text{ub}} = \mathrm{num}(y)_{\text{ship}}.$$

The highest distinguishable degree on the other hand, is related to the coordinate η and therefore to the resolution in the x-direction. The associated Legendre function of the first kind shows oscillations, resembling harmonic functions. This approximation is only valid away from the poles at $\eta = -1, 1$. A similar argumentation as for the resolution in the y-direction can be followed.

The number of oscillations for the associated Legendre functions of the first kind are not only dependent on n, but also on m. The number of oscillations ν equals [17, p. 3]

$$\nu = n - m + 1 - \delta_m, \tag{6.1}$$

where $\delta_m = 1$ if $m = 0$, otherwise $\delta_m = 0$. Instead of setting the largest value of ν that can be distinguished equal to the number of measurement grid points within twice the ship's length, it is set to only the number of grid points just within the length of the ship. This is due to the oscillatory behaviour only being present away form the poles and therefore away from the ends of the ship.

Furthermore, for simplicity, the largest measurable degree is approximated independent of the order. The result is the upper bound of the distinguishable degrees to be set to

$$N_{\text{ub}} = \mathrm{num}(x)_{\text{ship}}. \tag{6.2}$$

It is noted that larger values of N_{ub} and M_{ub} might result in a less robust solution, since the number of coefficients is larger. Especially for measurement grids with small spacings, these upper bounds can be rather large. Another effect of large upper bounds is that more computational power is required, due to the fact that the problem has become larger.

In this thesis, a rectangular grid is considered, but the use of drones would make any flight path possible. If, for example, the measurements were spaced on an ellipsoidal grid, the values of N_{ub} and M_{ub} can be chosen differently. All measurements taken in the athwart direction might contribute to the largest order m that can be perceived. The same is suggested for the measurements in the longitudinal direction, which contribute to the highest degree n that can be observed.

6.2. Optimising coordinate system

The regression methods described in Chapter 5 seek to solve the equation

$$\mathbf{B}^J = \mathrm{A}\boldsymbol{\beta}$$

for unknown $\boldsymbol{\beta}$. The length of this vector is already investigated in the previous section. So this section focuses on the other hyperparameters that, besides the coordinates, determine A: the focus length L and centre point (x_0, y_0, z_0) of the prolate spheroidal coordinate system.

The matrix A is computed from (3.7) for a given prolate spheroidal coordinate system. However, since the parameters defining an optimal prolate spheroidal coordinate system are not directly measurable, these parameters need to be estimated. This transformation to a prolate spheroidal coordinate system affects the elements of matrix A.

The transformations in (2.10) and (2.11) give the relations between coordinates in a cartesian coordinate system centred at $(0, 0, 0)$ and in a prolate spheroidal system with foci at $x = \pm L$. As discussed before, the prolate spheroidal harmonics can describe the magnetic field outside an ellipsoid encapsulating the source. Therefore, the prolate spheroidal coordinate system must be centred on that encapsulating ellipsoid. This centre point is (x_0, y_0, z_0). Therefore, the coordinates of the measurements must be translated such that $(0, 0, 0)$ becomes the ellipsoid's centre:

$$
\begin{aligned}
x &= \overline{x} - x_0,\\
y &= \overline{y} - y_0,\\
z &= \overline{z} - z_0,
\end{aligned}
$$

where the bar indicates that the coordinates are the cartesian coordinates before translation.

The rotation of the coordinate system is not considered in this thesis but could easily be included in the transformations. This would also require a more complex propagation of positioning errors. The measurement grid in this thesis is aligned such that the x-axis is aligned with the longitudinal direction of the ship.

After the cartesian coordinates have been centred, the coordinates can be transformed to the prolate spheroidal system. An important constraint is considered: L must be chosen such that the smallest encapsulating ellipsoid with radial parameter ξ_{ship} does not cross the planes in which the field is measured and predicted. This is illustrated as follows: imagine a prolate spheroidal system with $2L$ much smaller than the length of the ship. The smallest encapsulating ellipsoid approaches spherical dimensions. For such a small value of L measurements and predictions can then only be made far away from the ship's centre. Such a model would not be of much assistance against naval mines.

This constraint is quantified as follows: take height h at which a prediction or measurement is desired and define l to be the length of the ship. Given L, the value of ξ_0 must be large enough to enclose the ship, which gives rise to the constraint:

$$
\begin{aligned}
\xi_0 &> \frac{1}{2}\left(\sqrt{\left(1+\frac{l}{2L}\right)^2} + \sqrt{\left(1-\frac{l}{2L}\right)^2}\right)\\
&= \frac{1}{2}\left|1+\frac{l}{2L}\right| + \frac{1}{2}\left|1-\frac{l}{2L}\right| = \frac{l}{2L}.
\end{aligned}
$$

On the other hand, ξ_0 is constrained by the horizontal measurement plane at height h:

$$
\begin{aligned}
\xi_0 &< \frac{1}{2L}\left(\sqrt{L^2+h^2} + \sqrt{L^2+h^2}\right)\\
&= \frac{1}{L}\sqrt{L^2+h^2}.
\end{aligned}
$$

Combining these two constraints, gives

$$
\begin{aligned}
\frac{l^2}{4} &< L^2 + h^2\\
L &> \sqrt{\frac{l^2}{4} - h^2}.
\end{aligned}
$$

With this constraint in mind, it seems easy to choose a value for L much larger than $l/2$ to be safe. However, doing so might require more multipoles to describe the measured field adequately, which makes the model unnecessarily complex.

Since the parameters L, x_0, y_0 and z_0 cannot be directly measured, they must be estimated. A Monte Carlo method is chosen for this estimation: for each parameter a region is determined in its parameter space in which the optimal value is likely to be, after which random samples are drawn from this region. For each candidate (L, x_0, y_0, z_0), the matrix A is computed and the Lasso-LARS algorithm is run to generate a set of solutions $\boldsymbol{\beta}$. The selection of a model with a certain (L, x_0, y_0, z_0) and $\boldsymbol{\beta}$ amongst all generated models is discussed in Chapter 7.

The randomised search for the optimal parameter values is considered to be computationally more efficient than the well-known grid searches. For the latter a grid of parameter combinations is created. Just as for the random sampling, the matrix A is computed and the Lasso-LARS algorithm is run.

Since the dependence of the success of each model on the accuracy of a parameter is unknown, a grid wastes computational power. For example, if say the focus length is not needed to be known as accurately as the parameter x_0 to adequately describe the magnetic field, a higher resolution in the x_0 space is desirable. However, since the resolution of each parameter is chosen before knowledge is gathered on the desired precision, a medium resolution in both the L- and x_0-parameter spaces is chosen instead of a low resolution in the L-space and a high resolution in the x_0-space. The random sampling of parameters from a uniform distribution, results in a more flexible resolution.

7

Model Selection

The Lasso-LARS method described in section 5.5 produces a set of coefficient vectors of which each vector is a solution of the regularised cost function (5.8) for different values of the regularisation parameter α, where the regularisation function is the Lasso regularisation function from (5.9a). This is then done for each of the sampled parameters as described in 6.2. Then which of the computed models is the best model? The answer to this question depends on the measure that is chosen. The process of choosing the best model with a certain measure is referred to as *model selection*.

Common measures of the goodness of fit describe the residuals, since small residuals indicate that the model describes the training data well. However, as discussed previously, this might play into overfitting. A regularisation parameter α needs to be chosen such that the chance that the model is overfit, is minimised. Measures such as the mean square error (MSE) and R^2 will naturally select low values of α, because for α close to 0, the solution resembles the OLS solution, which minimises the residuals per definition. A different measure must be chosen.

The division of data into training data and test data moves the focus to the predictive power of the model and is called *cross validation*. The model is fit to the training data, but the quality of the model is scored using the test data. If the the model performs well, then the model is said to be a good predictor. The division, however, must be such that the test data is a good representation of the whole population, or, in the context of naval mine threats, the space below a ship. For the problem in this thesis, the model selection must be based on field measurements taken by a drone, which can, therefore, not be representative of the field underneath the ship. Since a model is sought that is a good representation of the actual source and thus the field underneath the ship, cross validation is not a viable selection method.

In this thesis, two types of model selection are discussed. The first method selects models based on Akaike's information criterion, while the second takes a weighted average of the coordinate parameters L, x_0, y_0 and z_0 of 'good' models, after which the Lasso-LARS algorithm is run again to find the multipole coefficients belonging to this averaged coordinate system.

7.1. Akaike's information criterion

A method that does take the complexity of the model into account is model selection based on *Akaike's Information Criterion* (AIC). The AIC rewards models with small residuals using the likelihood function and penalises models with a large number of coefficients. This results in a balance between the simplicity of the model and the goodness of fit.

The AIC is based on the minimisation of the Kullback-Leibler information, which is defined as [18, p. 51]

$$I(f,g) = \int f(x) \ln\left(\frac{f(x)}{g(x|\gamma)}\right) \mathrm{d}x$$

for two probability distributions f and g. The function $f(x)$ is the 'full reality' or truth that is modelled by $g(x|\gamma)$, where x is the data and γ the parameters. This measure can be interpreted as the information lost when f is modelled by g, or the distance between f and g. Model selection should therefore aim to minimise the distance between the selected model and f.

However, since the computation of I requires full knowledge of $f(x)$, which is unknown, the Kullback-Leibler information I cannot be used directly. A relative measure can be defined that is equal to

$$-\mathbb{E}_f[\ln(g(x|\gamma))] = -\int f(x)\ln(g(x|\gamma))\mathrm{d}x,$$

since the remainder is a constant that depends on $f(x)$. The derivation of the AIC is beyond the scope of this thesis, but the results are presented. Hirotugu Akaike showed that the expectation above can be estimated without bias by

$$\ln\left(\mathcal{L}\left(\hat{\gamma}|\text{data}\right)\right) - k, \tag{7.1}$$

where $\mathcal{L}$ is the likelihood function of for model g and k is the number of estimable parameters in the model [18, p. 51]. Akaike defined his criterion by multiplying this estimator with -2 for 'historical reasons':

$$\text{AIC} = -2\ln\left(\mathcal{L}\left(\hat{\gamma}|\text{data}\right)\right) + 2k. \tag{7.2}$$

To find an expression for the AIC, the likelihood function $\mathcal{L}$ must be determined. Each observation in itself is treated as a random variable with mean determined by $\mathrm{A}\boldsymbol{\beta}$ and standard deviation σ_j following (5.1), where σ_j is given by (5.2). As a useful shorthand, γ will be used to refer to the coordinate parameters L, x_0, y_0 and z_0 of the system from now on. The data, which are the coordinates at which magnetic field measurements were taken and the measurements, are represented by the set of coordinates $C = \{C_j \in \mathbb{R}^3 : j = 0, 1, ... J-1\}$ and $\mathbf{B}^J$. The likelihood function is therefore written as $\mathcal{L}(\gamma, \boldsymbol{\beta}|C, \mathbf{B}^J)$.

Since each field measurement is assumed to be normally distributed around its calculated expected value, the probability density function of each measurement B_j^J at a coordinate C_j is given by

$$f_j(C_j, \mathrm{B}_j^J|\gamma, \boldsymbol{\beta}) = \frac{1}{\sqrt{2\pi\sigma^2}}\mathrm{e}^{\left(\mathrm{B}_j^J - \mathrm{A}_j\boldsymbol{\beta}\right)^2/2\sigma_j^2},$$

where A_j is the j-th row of matrix A, which is calculated from C and γ. If each row and the corresponding measurement are normalised by weight $1/\sigma_j$, then the probability density function can be simplified as follows:

$$f_j(C_j, \mathrm{B}_j^J|\gamma, \boldsymbol{\beta}) = \frac{1}{\sqrt{2\pi}}\mathrm{e}^{\left(\mathrm{B}_j^J - \mathrm{A}_j\boldsymbol{\beta}\right)^2/2},$$

The likelihood function of the data given a certain coefficient vector $\boldsymbol{\beta}$ and coordinate parameters γ is expressed as follows:

$$\mathcal{L}(\gamma, \boldsymbol{\beta}|C, \mathbf{B}^J) = \prod_{j=0}^{J} f_j(C_j, \mathrm{B}_j^J|\gamma, \boldsymbol{\beta}) = (2\pi)^{-3J/2}\mathrm{e}^{-||\mathbf{B}^J - \mathrm{A}\boldsymbol{\beta}||^2/2}.$$

With the expression for the likelihood function above, the full AIC is given by

$$\text{AIC}(\gamma, \boldsymbol{\beta}) = 3J\ln(2\pi) + ||\mathbf{B}^J - \mathrm{A}\boldsymbol{\beta}||^2 + 2k. \tag{7.3}$$

Since the objective is to minimise the AIC, the constant terms can be omitted, which results in the following expression for the AIC for Lasso:

$$\text{AIC}(\gamma, \boldsymbol{\beta}) = ||\mathbf{B}^J - \mathrm{A}\boldsymbol{\beta}||^2 + 2k, \tag{7.4}$$

and if the residuals are not normalised:

$$\mathrm{AIC}(\gamma, \boldsymbol{\beta}) = \left(\mathbf{B}^J - \mathrm{A}\boldsymbol{\beta}\right)^T \mathrm{W} \left(\mathbf{B}^J - \mathrm{A}\boldsymbol{\beta}\right) + 2k,$$

where W is the diagonal matrix with $1/\sigma_j^2$ on its diagonal and k is the number of coefficients in $\boldsymbol{\beta}$ unequal to 0. The vector of multipole coefficients is selected as follows:

$$\hat{\beta} = \operatorname{argmin}_{\beta} \mathrm{AIC}(\beta). \tag{7.5}$$

In the strict form, all parameters in the model should be counted to find k [18, p. 62]. These include each σ_j and L, x_0, y_0 and z_0. However, since the constant terms in the AIC can be omitted, the number of coefficients in β unequal to 0 suffices [18, p. 71].

The AIC should be interpreted as a measure of the relative likelihood of each model. This means that if the model with the lowest AIC is selected, this model is most probable to describe the underlying phenomenon.

7.2. Akaike weights

If unlimited computational power would be available, models for all combinations of parameters could be generated and scored. In this case, the assumption that the truly best model is contained in the set of computed models would be correct. However, this assumption should be considered wrong for limited computation time. For trying to find the optimal values for the four parameters describing the coordinate system, a grid of only 10 values in each direction results in a total of $N = 10^4$ models.

In this thesis, besides traditional model selection based on the lowest AIC value, a model selection method using a weighted average of parameters is proposed. A comparison of the results from the two methods will be given later.

The AIC value belonging to $\hat{\beta}$ from (7.5), which is the lowest computed AIC, is referred to as $\mathrm{AIC}_{\mathrm{min}}$. The relative measure of the Kullback-Leibler information, which is the corrected log-likelihood of a model, is estimated by (7.1) and can be used as a weight of that model. This value equals $-\mathrm{AIC}/2$. For each model i with AIC value AIC_i, a normalised weight w_i is defined by [18, p. 75]

$$w_i = \frac{\mathrm{e}^{-\mathrm{AIC}_i/2}}{\sum_r \mathrm{e}^{-\mathrm{AIC}_r/2}} = \frac{\mathrm{e}^{(\mathrm{AIC}_{\mathrm{min}} - \mathrm{AIC}_i)/2}}{\sum_r \mathrm{e}^{(\mathrm{AIC}_{\mathrm{min}} - \mathrm{AIC}_r)/2}}, \tag{7.6}$$

where the sum in the denominator is the sum over all weighted models. The weights are normalised, meaning that they sum to 1. These weights can be used to compute a weighted average of the coordinate parameters L, x_0, y_0 and z_0 of the system. For the parameter L, the averaged parameter is calculated by

$$\overline{L} = w_1 L_1 + w_2 L_2 + \cdots + w_i L_i + \cdots + w_R L_R, \tag{7.7}$$

where R is the index of the last model that is used for the averaging. The same is done for parameters $\overline{x_0}$, $\overline{y_0}$ and $\overline{z_0}$.

Interestingly, one could also introduce an estimated standard error in these averaged parameters. This could be used to draw conclusions on the confidence in the found parameter values. Such a weighted standard error is calculated as follows:

$$\sigma_{\overline{L}} = \sqrt{\sum_{r=1}^{R} w_i \left(L_i - \overline{L}\right)^2} \tag{7.8}$$

and similarly for the other hyperparameters $\overline{x_0}$, $\overline{y_0}$ and $\overline{z_0}$.

For certain large values of the AIC, the corresponding model should be considered ‘wrong’ and can be discarded on that basis. These ‘wrong’ models have a negligible weight. Assuming that these values do not influence the weighted average, the sum over the all models can be limited to those with low AIC values and therefore non-negligible weights. There are multiple ways to do this.

One could opt for a limit based on an absolute criterion (e.g. the best 20 models are averaged), or a relative criterion (e.g. the best 5% is averaged). However, both of these methods do not take the influence of the model in the averaging process into account. In other words, these models leave room for either including or excluding too many 'wrong' models.

It is, therefore, considered to select models based on their AIC-value or more specifically the difference between their AIC-value and AIC_{min}. The value of

$$\mathrm{e}^{(\text{AIC}_{\text{min}} - \text{AIC}_i)/2}$$

is the ratio of w_i and the weight corresponding to AIC_{min}, which has the highest weight. It is chosen to omit models with ratio of weights smaller than 0.01, since these would have only limited effect on the calculated average. The difference in AIC-value with AIC_{min} is calculated by

$$\text{AIC}_i - \text{AIC}_{\text{min}} = -2\ln(0.01) \approx 9.2. \tag{7.9}$$

For clarity, it is stated that this averaging process only averages the hyperparameters L, x_0, y_0 and z_0 and not the multipole coefficient vector $\boldsymbol{\beta}$. To find the corresponding $\boldsymbol{\beta}$, the Lasso-LARS algorithm is run again for the averaged parameters.

8

Model Implementation and Evaluation Methods

In this chapter, the theoretical knowledge from the previous is chapters is combined to discuss the design of the complete model and the implementation in python. Also, several methods that will be used to analyse the performance and stability of the model will be discussed.

8.1. Model design

Firstly, to answer the question on how a model that translates measurements of the magnetic field above a source to underneath the source can be designed and implemented, the different aspects discussed in the previous chapters are combined into one prediction model. As discussed in Chapter 4, the input of the model consists of a set of coordinates and a set magnetic field measurements at each coordinate.

From the grid of coordinates, the upper bounds for the largest order and degree of the associated Legendre functions to be evaluated is first computed using (6.1) and (6.2). If limited computer power necessitates it, lower values can be chosen.

The next steps are repeated until satisfaction. An initial guess of the possible range for values of the parameter vector $\gamma = (L, x_0, y_0, z_0)$ is made. A number of parameter candidates is sampled from a uniformly distributed parameter space of γ. For each candidate γ_i, an instance of the original set of coordinates is translated to have centre $(x_0, y_0, z_0)_i$ and then transformed to a prolate spheroidal coordinate system with focal length L. The matrix $\mathrm{A}_{\mathbf{B}}$ as defined in (3.7) is then constructed for each candidate.

The computed matrix is, however, not ready to be used in the Lasso-LARS algorithm. First, the residuals are standardised by dividing each row of the matrix and its corresponding magnetic field component by the computed standard deviation σ_j as given in (5.2). Moreover, each column is standardised by dividing each element with the standard deviation of the values in the row.

The Lasso-LARS algorithm (see Section 5.5) can now be used to find every Lasso solution $\boldsymbol{\beta}$ for a candidate γ_i. For each of these solutions, Akaike's information criterion is computed from (7.4) and the Lasso solution $\hat{\boldsymbol{\beta}}$ with the lowest AIC is selected as in (7.5). This is done for each candidate γ_i. The result is a set of vectors $\hat{\boldsymbol{\beta}}_i$ with multipole coefficients giving the best description of the magnetic field as measured by a drone, where 'best' is defined as 'to have the lowest AIC'. Each combination of γ_i and $\hat{\boldsymbol{\beta}}_i$ is one model, which is referred to as $(\gamma, \hat{\boldsymbol{\beta}})_i$. Then, for every candidate and corresponding coefficient vector $(\gamma, \hat{\boldsymbol{\beta}})_i$, the AIC values are compared to find the best combination of parameters.

This process of searching a certain partition of the parameter space for the optimal combination of parameters with corresponding optimal vector of multipole coefficients can be repeated until a satisfactory solution is found. No hard and fast rule has been decided upon, but a formulation of one might be of interest for further research. A criterion to be considered for example is that the parameter space must have been sampled in such a manner that the 'best' solution is surrounded by multiple other 'good' solutions.

From a final search of the parameter space, a weighted average of each parameter L, x_0, y_0 and z_0 is computed as discussed in Section 7.2. The Lasso-LARS algorithm is then used again to find the optimal coefficient vector $\hat{\boldsymbol{\beta}}$ belonging to these averaged parameters to find a more stable solution.

The result until thus far is a best solution from the sampled candidates $(\gamma, \hat{\boldsymbol{\beta}})_{\text{min}}$ and an 'averaged' solution $(\gamma, \hat{\boldsymbol{\beta}})_{\text{avg}}$. Either one of these solutions can be chosen to predict the magnetic field underneath the source.

This prediction can be made as follows. The coordinates in which the magnetic field is to be predicted are first transformed to prolate spheroidal coordinates specified by γ. These are then used to compute the matrix $\mathrm{A}_{\mathbf{B}}$, after which matrix multiplication with $\hat{\boldsymbol{\beta}}$ as in (3.8) gives a prediction of the magnetic field based on the measurements performed by a drone above the source. It is important that either this new matrix $\mathrm{A}_{\mathbf{B}}$ is scaled exactly the same as the matrix used in the inversion problem or that the multipole coefficients are scaled back.

8.2. Model implementation

An implementation of the model was made in Python. The most important modules that were used are SciPy (which contains NumPy and Matplotlib) and scikit-learn. SciPy facilitates easy access to complex functions such as the associated Legendre functions of both the first and second kind. Moreover, its submodule NumPy is one of the most used modules in Python and it enables efficient computations using n-dimensional arrays. Matplotlib on the other hand is used to make plots.

Scikit-learn is a widely used Python module for machine learning. This module was used for all machine learning steps, but some adaptations were made to fit the application for this thesis.

Three importand objects were created to implement the model: `MatrixBuilder`, `RandomizedSearchAIC` and `LassoLarsAIC`. The object `MatrixBuilder` is needed to compute the matrix $\mathrm{A}_{\mathbf{B}}$ at runtime. Scikit-learn offers `Pipeline` objects. As the name suggests, these objects can be used to simply the different steps in the fit process such that the inputs enter the pipeline and at the other end, a fitted model is returned. It is desirable to let parameter candidates be sampled at runtime, so for abstraction of the code, the inputs of the fit process should consist of the coordinates, the measurements, values for N and M and the parameter candidate L, x_0, y_0 and z_0. This allows for the sampling to be done outside of the pipeline (and in the `RandomizedSearchAIC` object). The `MatrixBuilder` object uses these inputs to construct matrix $\mathrm{A}_{\mathbf{B}}$.

The other steps in the pipeline are the scaling of the rows and columns of the matrix, but most importantly the actual fit using the Lasso-LARS algorithm. The scikit-learn module offers an implementation of this algorithm. However, a critical assumption underlying the code makes this implementation inadequate for the use in this thesis. TheThe scikit-learn implementation of the Lasso-LARS with build-in model selection using the AIC, assumes that the measurement data is identically distributed around mean 0 with standard deviation σ. However, the measurement data used in this thesis is obviously not standardised in this way. While the standard deviations of the measurement errors are set to 1, the mean of all the measurements cannot be set to 0. Therefore, a custom implementation `LassoLarsAIC` was introduced.

The last important object that was introduced for this thesis is `RandomizedSearchAIC`. In machine learning, the term *hyperparameters* refers to those parameters of a model that are set prior to fitting the model to training data. For example the value of α that controls the strength of regularisation in (5.8) is considered a hyperparameter. In the scope of `LassoLarsAIC`, which finds a vector of multipole coefficients, the parameter candidates (L, x_0, y_0, z_0) are hyperparameters, since these are set before the training of the model. One of the objects that is offered by the scikit-learn module and aid in finding these hyperparameters, is `RandomizedSearchCV`. This object facilitates random sampling of hyperparameters from a parameter space, but does not select based on an AIC comparison, but by comparing goodness of fit measures such as R^2 or mean squared errors. Therefore, this object cannot be employed in the implementation. Additionally, cross validation can be used in this selection process.

The custom `RandomizedSearchAIC` object implements the selection of hyperparameters on the basis of their AIC. Moreover, the best hyperparameters are weighted and averaged to produce $(\gamma, \hat{\boldsymbol{\beta}})_{\text{avg}}$. This is all handled by this object.

By using scikit-learn's API [19], the high level of efficiency offered by the module was achieved. Lastly, the most important code used in this included in Appendix C.

8.3. Evaluation of the model

Firstly, a comparison of the true field and the predicted field underwater is to be made. The measure that is chosen is the root mean squared error (RMSE), which is defined as follows:

$$\text{RMSE} = \sqrt{\frac{\sum_{j=0}^{J-1} \left(\mathrm{B}_j - \hat{\mathrm{B}}_j\right)}{J}}, \tag{8.1}$$

where B_j and $\hat{\mathrm{B}}_j$ are elements of the vector $\mathbf{B}^J$ as defined in (4.1) which represents the true and predicted magnetic fields respectively. The RMSE can be interpreted as the expected difference between the true field and modelled field at any point.

Not only does the question arise whether or not the model described above can successfully predict the field underneath ships, but also how sensitive the model is to noise levels. Three questions are formulated in this respect:

1. How does the noise level affect the success of the magnetic field prediction underneath sources?
2. What is the influence of the noise level on the number of selected coefficients $k_{>0}$[1]?
3. How is the stability of the multipole coefficients affected by the noise level?

To answer these question, 8 levels are chosen for the errors in positioning and 9 are chosen for the errors in the field measurements, ranging from no noise to what can be considered as very noisy. For each of the 72 combinations of noise levels, the coefficient vector $\hat{\boldsymbol{\beta}}$ is estimated for 50 realisations of noisy data. The parameters $\gamma = (L, x_0, y_0, z_0)$ are the same for each attempt.

After computing the RMSE for each realisation of noisy data, a mean RMSE value can be computed for each of the 72 noise levels. A comparison of these means might give insights into the sensitivity to noise.

Moreover, the number of coefficients that are unequal to 0 in the found solution of each realisation of noisy data is averaged for each combination of noise levels. Next, the standard deviation of this $k_{>0}$ is also calculated.

To analyse the stability of the multipole coefficients, the deviation from the mean of the first three coefficients is computed for each combination of noise levels. A small deviation indicates stability, since the value of coefficients in not dependent on the random noise.

Furthermore, the number of samples taken in the sample space is to be studied. Ultimately, the necessary resolution in the parameter space determines the number of samples of (L, x_0, y_0, z_0) that are taken, which determines the required computational power for a large part. The following question is posed to study the necessary resolution:

4. What is the influence of variations in the hyperparamters L, x_0, y_0 and z_0 on the success of the predictions?

---

[1] $k_{>0}$ denotes the number of multipole coefficients that are unequal to 0.

To answer this question, the hyperparameters L, x_0, y_0 and z_0 are set to the optimal values as determined by the algorithm designed in this thesis. Then, each hyperparameter is varied within a range, while the others are kept constant. The algorithm is used to produce a vector of multipole coefficients $\boldsymbol{\beta}$ for each variation. By plotting the RMSE values of the predictions made by each model, the effect of variations in each parameter around the found optimum can be studied. The measurement data for this analysis is the simulated data for test case 1, which will be discussed in the next chapter.

9

Data Simulation

The model designed in the previous chapters will be tested, to conclude to the success of signature measurements taken by a drone. However, since no data set of magnetic field measurements taken by a drone of a ship are publicly accessible, these measurements will be simulated. Three test cases are simulated.

One of the supervisors of this thesis has generated three sets of $(\mathrm{B}_x, \mathrm{B}_y, \mathrm{B}_z)$ values on a grid above a source. The magnetic field values are generated by a prolate spheroidal harmonic expansion from an actual signature measurement under a ship with an unknown centre point (x_0, y_0, z_0) and focal length L. The field measurements under the ship, were only provided after the model was implemented to ensure a true verification of the model and these measurements were used for nothing else. Only small errors in the code were fixed after receiving the verification data. Plots of the true magnetic field underwater are included in Chapter 10.

The measurements were generated on a grid with $\mathrm{num}(x) = 61$ and $\mathrm{num}(y) = 31$ and grid spacings $\delta x \approx 2.278$ m and $\delta y \approx 2.277$ m. The grid lies in the horizontal plane 10 m above the water. The three data sets are visualised in Fig. 9.1 to 9.3.

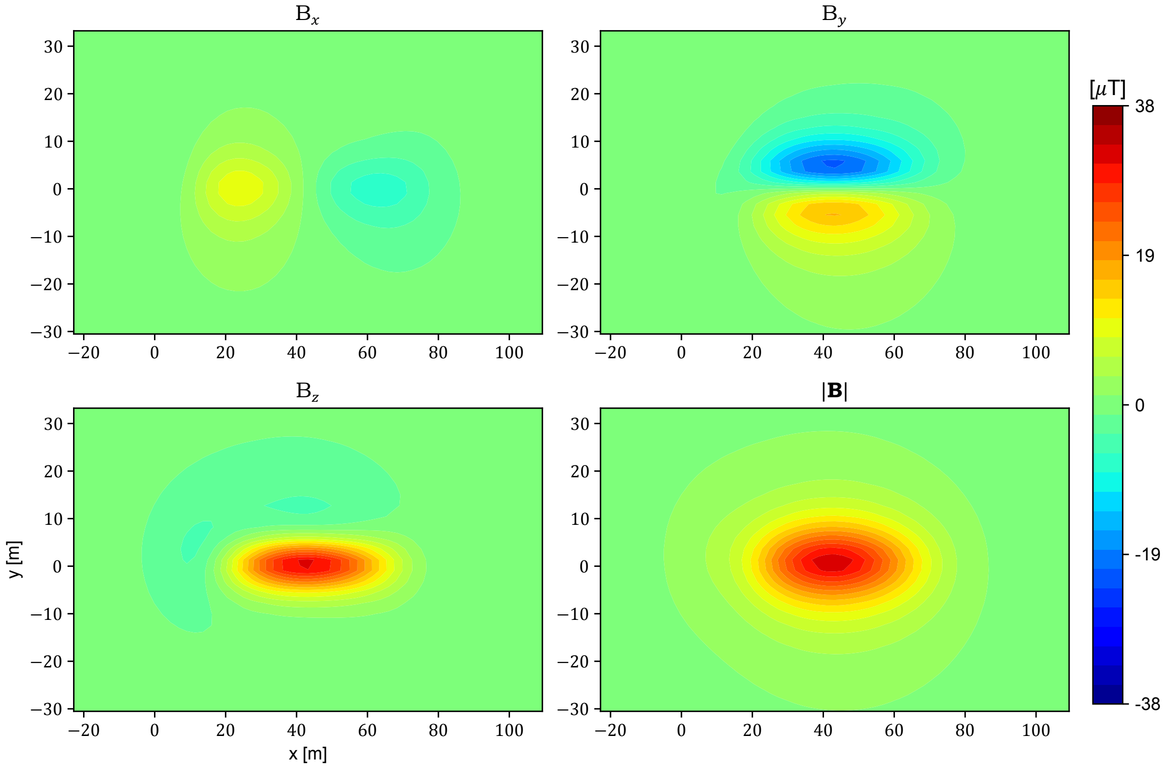


Figure 9.1: Visualisation of test case 1 without noise. From top left to bottom right, the subplots show contour plots of the three components of the magnetic field B_x, B_y and B_z and the magnitude of the magnetic field $|\mathbf{B}|$.

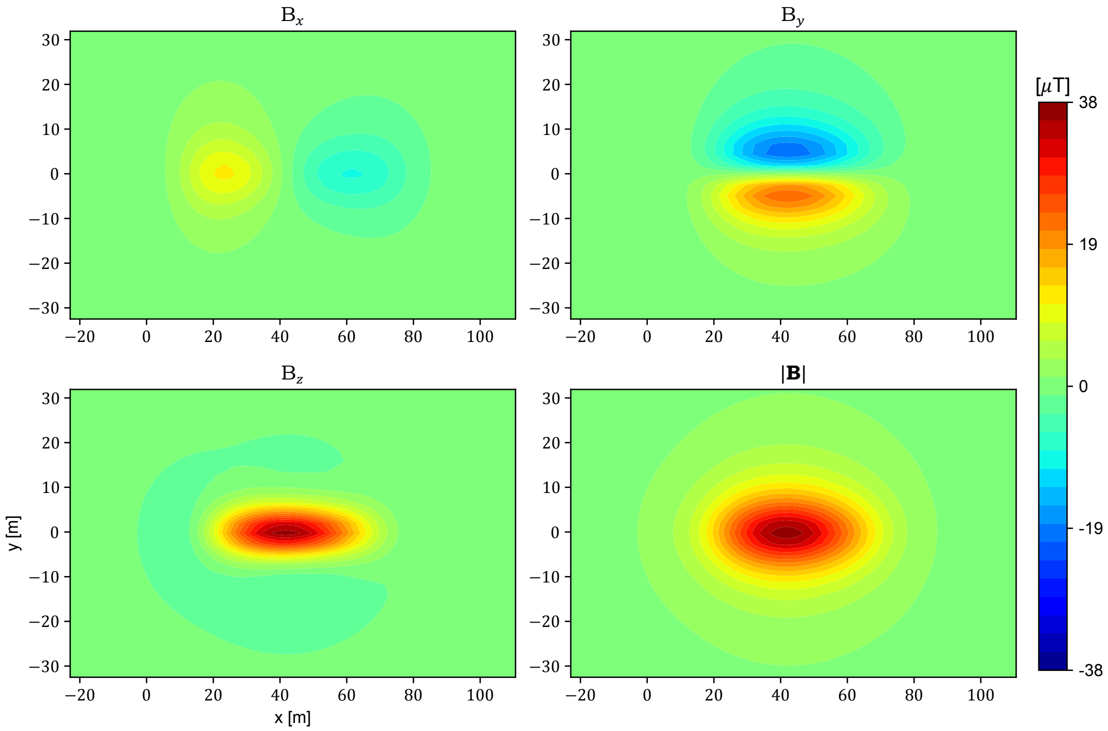


Figure 9.2: Visualisation of test case 2 without noise. From top left to bottom right, the subplots show contour plots of the three components of the magnetic field B_x, B_y and B_z and the magnitude of the magnetic field $|\mathbf{B}|$.

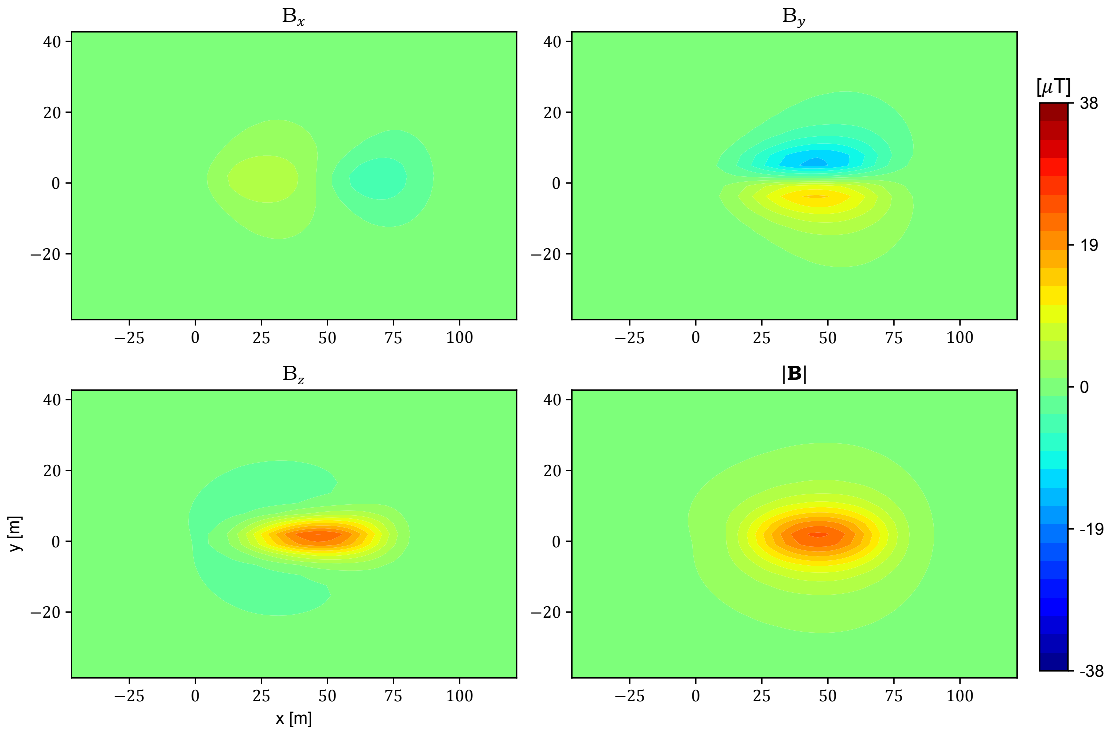


Figure 9.3: Visualisation of test case 3 without noise. From top left to bottom right, the subplots show contour plots of the three components of the magnetic field B_x, B_y and B_z and the magnitude of the magnetic field $|\mathbf{B}|$.

The result is three sets of simulated data without noise. To simulate actual measurements, noise is added to each measurement of the magnetic field and to the measurement of the position.

For the magnetic field measurements, the shape and size of the noise is determined by the sensor specifications. Typical commercially available magnetometers have errors caused by multiple sources. These sources include errors caused by the scaling of the measured field to an output voltage, a zero offset, and temperature drift.

The errors are dominated by the scaling error, which lies in the order of 0.1% of the measurement (0.1% for the Sensys FGM3D [20]; 0.5% for the Bartington Mag-13 [21]). The scaling error is also called the scale factor accuracy and is caused by an error in the linear translation of the measured magnetic field value to an output voltage.

The zero offset, on the other hand can be in the order of ± 5 nT. For the sake of simplicity, all errors in the measurements of the magnetic field are assumed to be the size of the scaling errors, since these are the dominant source of errors.

Positioning is imagined to be done by GPS. Normally, the location of the GPS receiver is calculated based on satellite signals. Typical high-end GPS receivers for drones reportedly yield positioning accuracies of 1.5 m in the horizontal direction and 3.0 m in the vertical direction [22]. These errors are of the same order as the grid spacings and are therefore too large.

Higher accuracies are possible with the introduction of a base station. The most modern method that uses a base station is called real-time kinematic positioning (RTK)[1]. In addition to positioning using GPS by the drone, a base station located nearby transmits corrections based on more accurate GPS information. The increased precision is significant: the positioning accuracy in the horizontal direction can be 1 cm and in the vertical direction 2 cm (1 cm and 2 cm for the D-RTK 2 by DJI [22]; 0.7 cm and

---

[1] An older method is Differential GPS (DGPS), which can reach a precision level of roughly 40 cm.

1.4 cm for the Reach M2 by Emlid while moving [23]). These values are the standard deviations of the errors.

Realistic measurements are simulated by adding noise to each measurement. The shape and size of which are chosen from typical sensors described above. For the magnetic field measurements, the errors are taken from a normal distribution with mean 0 nT and standard deviation 100 nT, since this is approximately 0.2% of the measurement scale.

The coordinates of each measurement are distorted by adding noise to each coordinate. The errors in the x- and y-coordinates are generated with a standard deviation of $\frac{0.7}{\sqrt{2}} \approx 0.5$ cm. Combining these two coordinates, amounts to an error distribution in the horizontal plane with a standard deviation of 0.7 cm. The errors in the z-coordinate are generated with a standard deviation of 1.4 cm.

10

Results

10.1. Field prediction

For test case 1, the initial search for the parameters L, x_0, y_0 and z_0 was performed with 2000 samples taken from uniform distributions where

$$
\begin{aligned}
20\text{ m} &\le L \le 80\text{ m},\\
20\text{ m} &\le x_0 \le 60\text{ m},\\
-15\text{ m} &\le y_0 \le 15\text{ m},\\
-5\text{ m} &\le z_0 \le 10\text{ m}.
\end{aligned}
$$

Figure 10.1 shows the AIC values plotted against each parameter and in figure 10.2 these plots were zoomed in to show more detail.

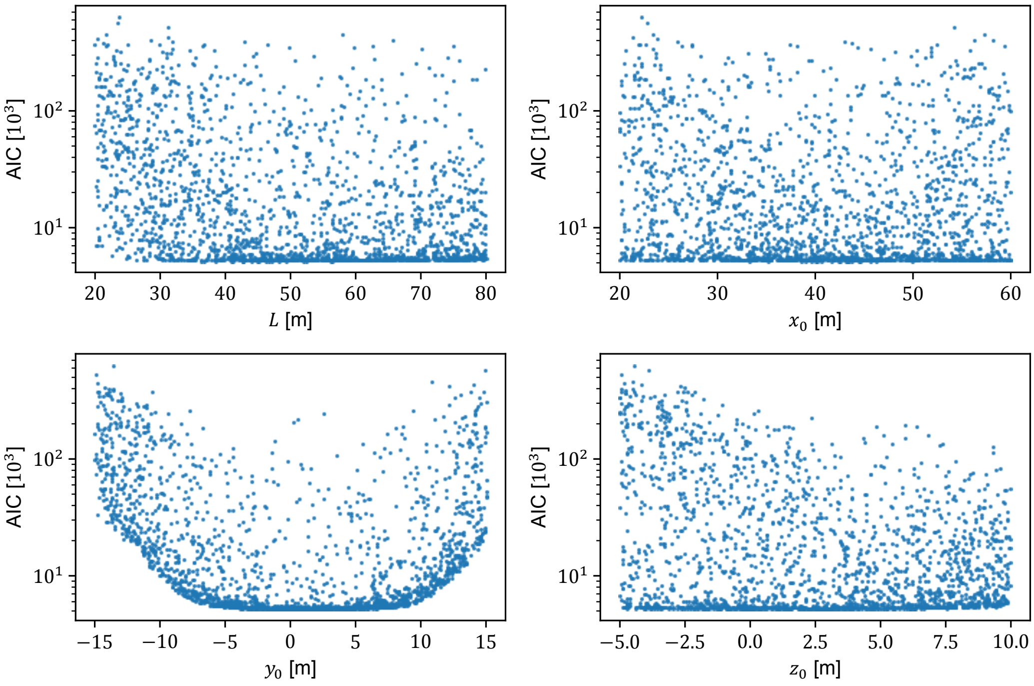


Figure 10.1: Scatter plot of the AIC values against the parameter candidates of L, x_0, y_0 and z_0 for the first search attempt. The vertical axis has a logarithmic scale.

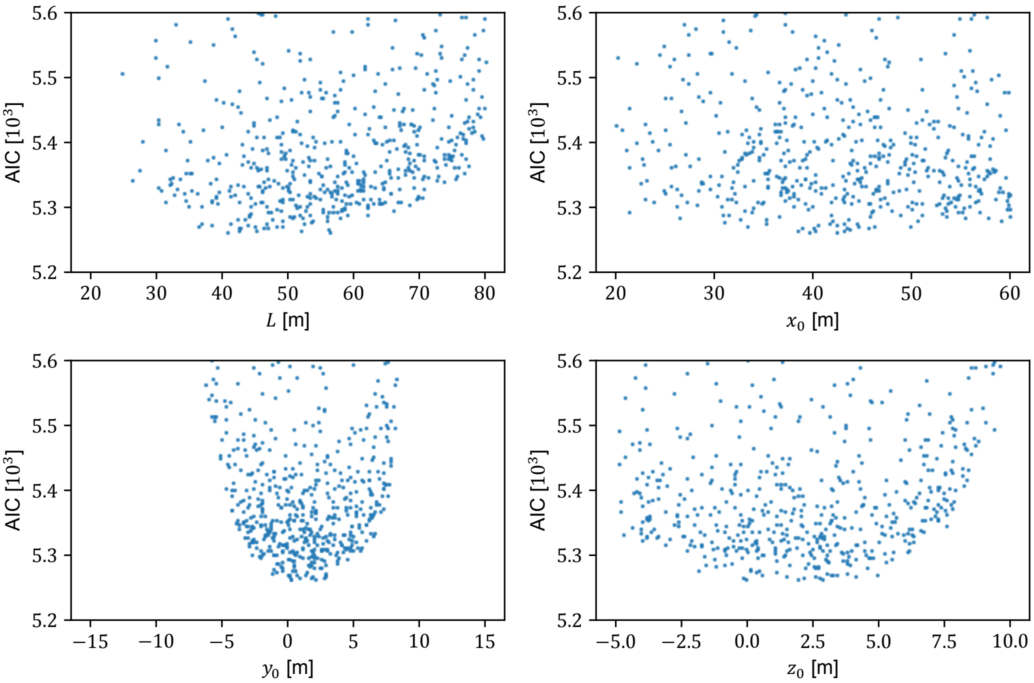


Figure 10.2: Scatter plot of the AIC values against the parameter candidates of L, x_0, y_0 and z_0 for the first search attempt, only showing AIC values lower than 5600.

Figure 10.1 shows clearly that within certain ranges, the AIC is small, but that the AIC becomes very large outside these boundaries. A second search was then performed within these ranges:

$$33\text{ m} \leq L \leq 60\text{ m},$$
$$38\text{ m} \leq x_0 \leq 50\text{ m},$$
$$-1\text{ m} \leq y_0 \leq 4\text{ m},$$
$$-1\text{ m} \leq z_0 \leq 4\text{ m}.$$

The search was performed with 2000 candidates. The results of the second search are presented in figure 10.3.

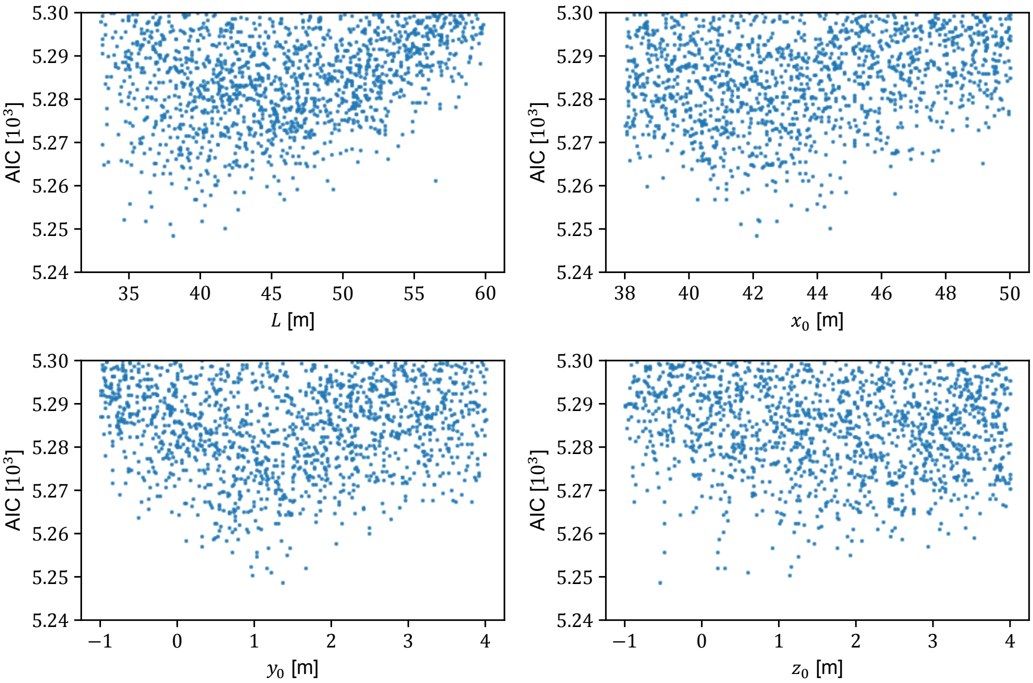


Figure 10.3: Scatter plot of the AIC values against the parameter candidates of L, x_0, y_0 and z_0 for the second search attempt, zoomed in to show AIC values under 5300.

Next, a final search of was performed closely around the found best values. From figure 10.3 the following bounds were chosen:

$$\begin{aligned} 33\text{ m} &\leq L \leq 42\text{ m}, \\ 41\text{ m} &\leq x_0 \leq 44\text{ m}, \\ 0.75\text{ m} &\leq y_0 \leq 2\text{ m}, \\ -0.5\text{ m} &\leq z_0 \leq 2\text{ m}. \end{aligned}$$

This final search was performed with 5000 candidates and, as before, plots of the AIC values against each parameter candidate are shown in figure 10.4.

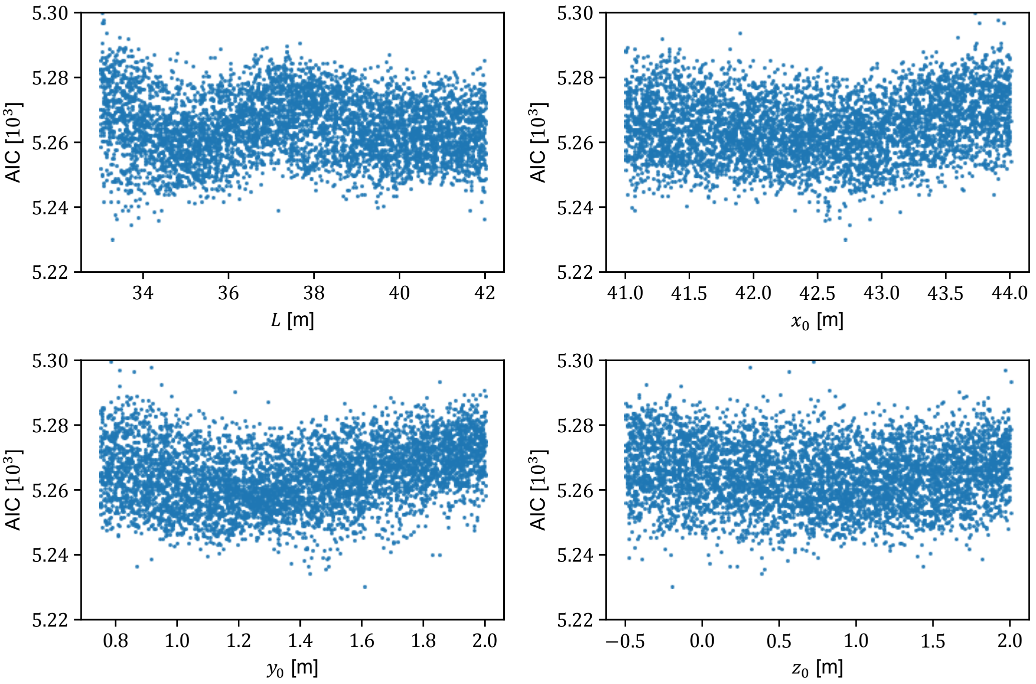


Figure 10.4: Scatter plot of the AIC values against the parameter candidates of L, x_0, y_0 and z_0 for the final search attempt, zoomed in to show AIC values under 5300.

The search for $(\gamma, \hat{\boldsymbol{\beta}})_{\text{min}}$, has concluded. The condition set in (7.9) was met by 13 candidates. Their AIC values, the weights w, the number of coefficients $k_{>0}$ and normalised mean squared errors (MSE)[1] are presented in Table 10.1. Moreover, the weights were used to produce the weighted averages such as in (7.7). These values are also given in table 10.1 with their standard errors calculated from (7.8). These averaged parameters were used to produce a set of coefficients.

AIC	L [m]	x_0 [m]	y_0 [m]	z_0 [m]	w	$k_{>0}$	MSE
5230.3	33.28	42.71	1.61	−0.19	0.713	40	1.0034
5234.6	33.71	42.74	1.43	0.39	0.083	49	1.0007
5235.9	34.34	42.55	1.48	0.40	0.042	45	1.0025
5236.4	33.36	42.59	1.42	0.23	0.033	45	1.0026
5236.6	41.95	42.91	0.87	1.43	0.030	51	1.0003
5236.8	33.89	42.67	1.49	0.17	0.027	49	1.0011
5237.7	33.36	42.57	1.37	0.04	0.018	46	1.0025
5238.5	34.27	43.14	1.41	0.55	0.012	49	1.0015
5239.0	41.62	42.67	0.91	1.82	0.009	49	1.0016
5239.1	37.16	41.07	1.48	−0.39	0.009	48	1.0020
5239.2	33.66	42.49	1.43	0.38	0.008	48	1.0020
5239.3	35.53	42.57	1.49	−0.47	0.008	48	1.0020
5239.3	34.42	42.32	1.47	0.75	0.008	45	1.0032
5228.6	33.78 ± 1.69	42.69 ± 0.18	1.54 ± 0.15	-0.01 ± 0.39	-	40	1.0030

Table 10.1: AIC and parameter values for the 13 parameter candidates that met the averaging requirement. The weights, $k_{>0}$ and MSE values are also given. The last row shows these values for the averaged parameters and the model that resulted from these values.

---

[1] Both the number of coefficients $k_{>0}$ and the MSE value are used in the computation of the AIC value in (7.4). Note that $||\mathbf{B}^J - \mathrm{A}\boldsymbol{\beta}||$, which is used in the AIC, is equal to $J \cdot$ MSE, with J the number of measurements.

Next, a prediction was made for the magnetic field under the ship. The predictions are made for a horizontal rectangular grid at 9 m underwater. The predictions were made using both $(\gamma, \hat{\boldsymbol{\beta}})_{\text{min}}$ and $(\gamma, \hat{\boldsymbol{\beta}})_{\text{avg}}$ and are shown in 10.5 to 10.8. The differences between the true and modelled field are plot in figure 10.9 and 10.10 of $(\gamma, \hat{\boldsymbol{\beta}})_{\text{min}}$ and $(\gamma, \hat{\boldsymbol{\beta}})_{\text{avg}}$, respectively.

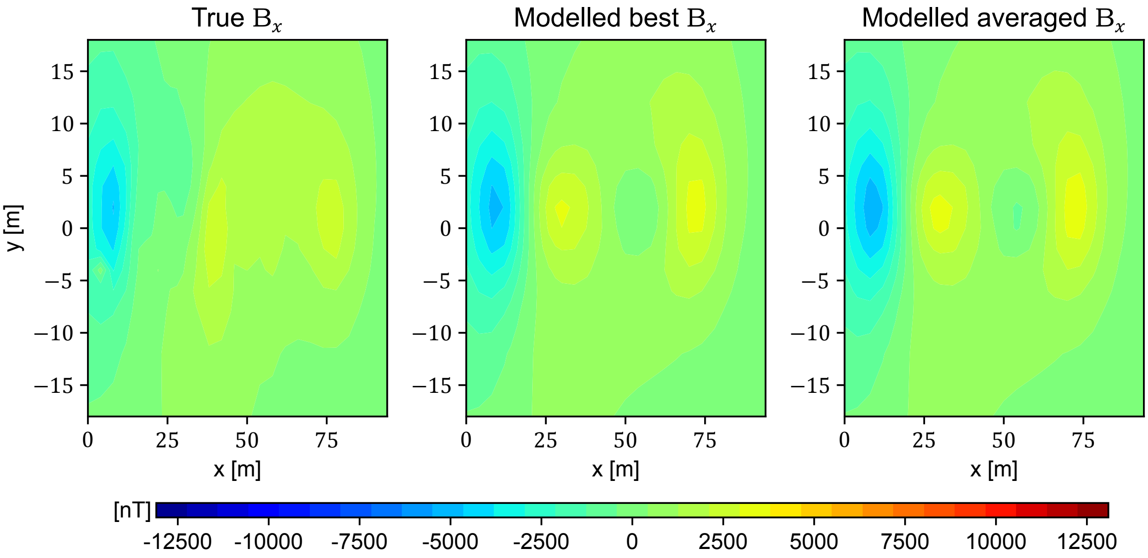


Figure 10.5: Plot of the true and modelled magnetic field component B_x underwater. The first plot shows the true field, the second shows the prediction by $(\gamma, \hat{\boldsymbol{\beta}})_{\text{min}}$ and the last plot shows the prediction by $(\gamma, \hat{\boldsymbol{\beta}})_{\text{avg}}$.

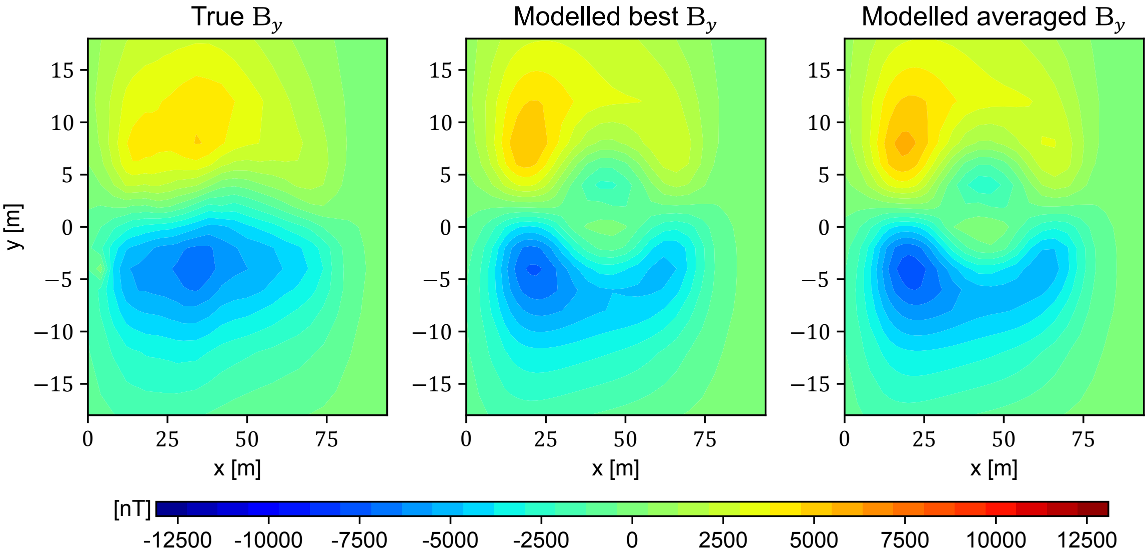


Figure 10.6: Plot of the true and modelled magnetic field component B_y underwater. The first plot shows the true field, the second shows the prediction by $(\gamma, \hat{\boldsymbol{\beta}})_{\text{min}}$ and the last plot shows the prediction by $(\gamma, \hat{\boldsymbol{\beta}})_{\text{avg}}$.

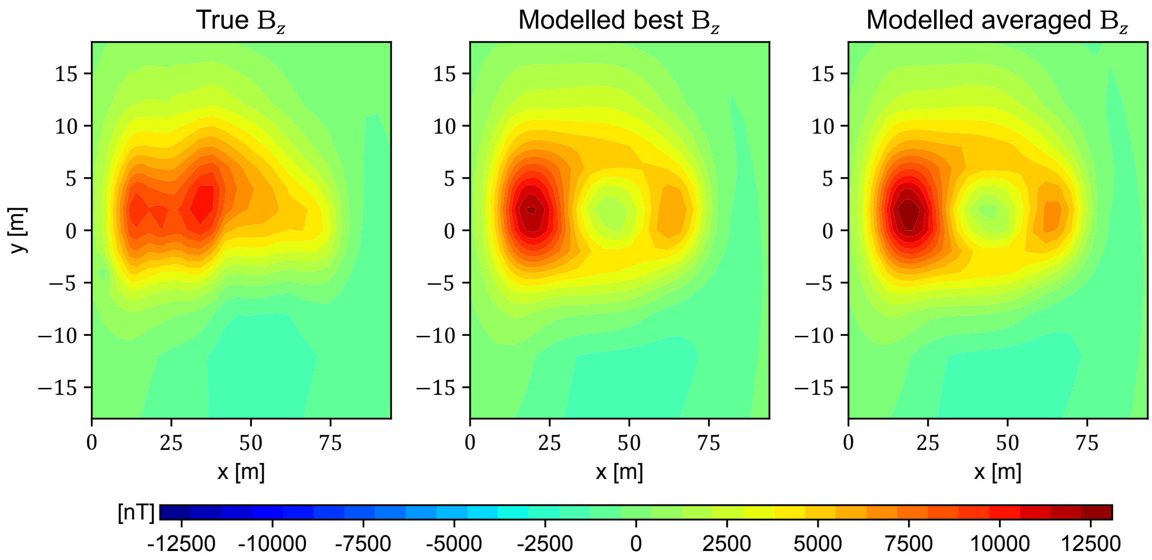


Figure 10.7: Plot of the true and modelled magnetic field component B_z underwater. The first plot shows the true field, the second shows the prediction by $(\gamma, \hat{\boldsymbol{\beta}})_{\text{min}}$ and the last plot shows the prediction by $(\gamma, \hat{\boldsymbol{\beta}})_{\text{avg}}$.

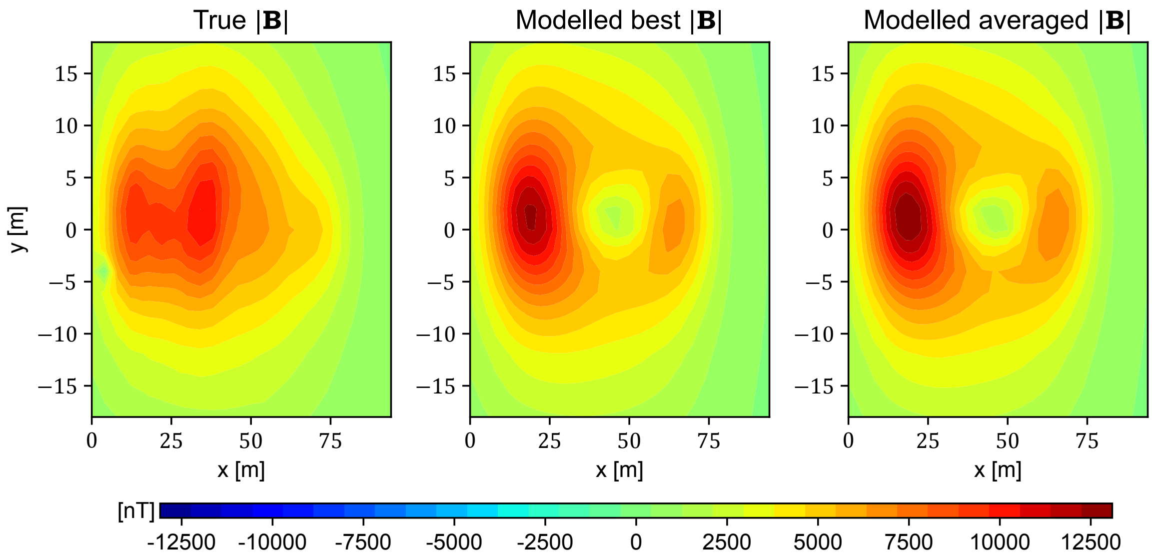


Figure 10.8: Plot of the true and modelled magnetic field strength $|\mathbf{B}|$ underwater. The first plot shows the true field, the second shows the prediction by $(\gamma, \hat{\boldsymbol{\beta}})_{\text{min}}$ and the last plot shows the prediction by $(\gamma, \hat{\boldsymbol{\beta}})_{\text{avg}}$.

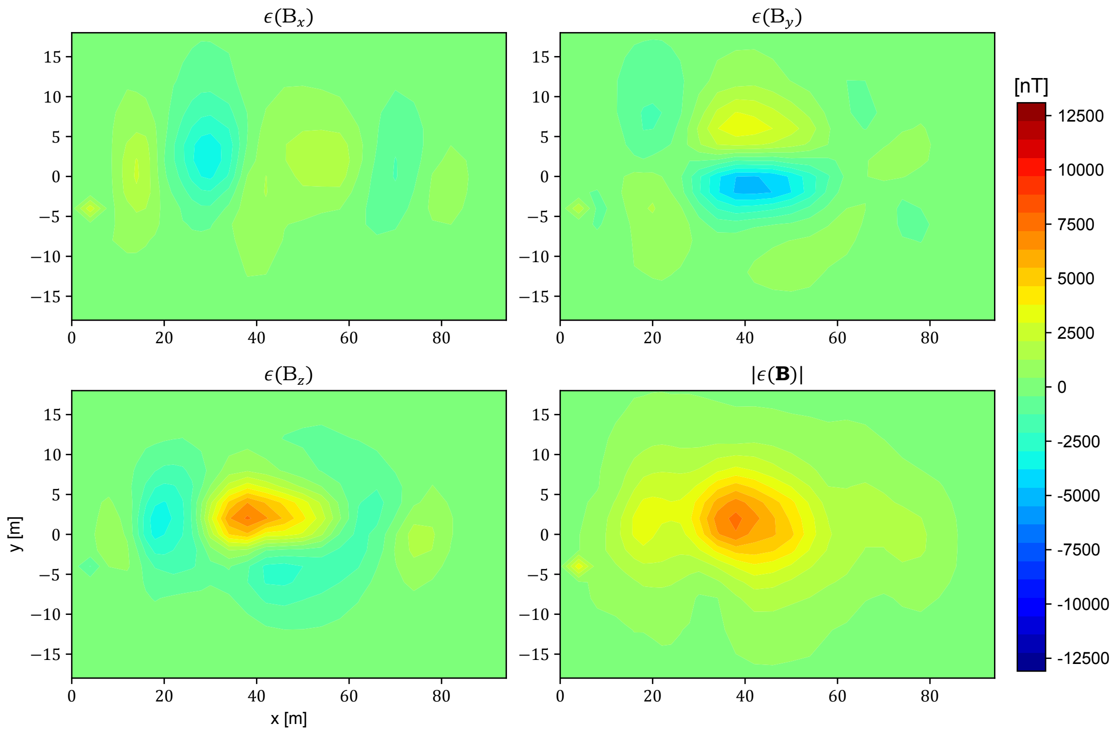


Figure 10.9: The error in the predicted field using $(\gamma, \hat{\boldsymbol{\beta}})_{\mathrm{min}}$. The first three plots show the error in each component and the last plot shows the magnitude of the difference vector.

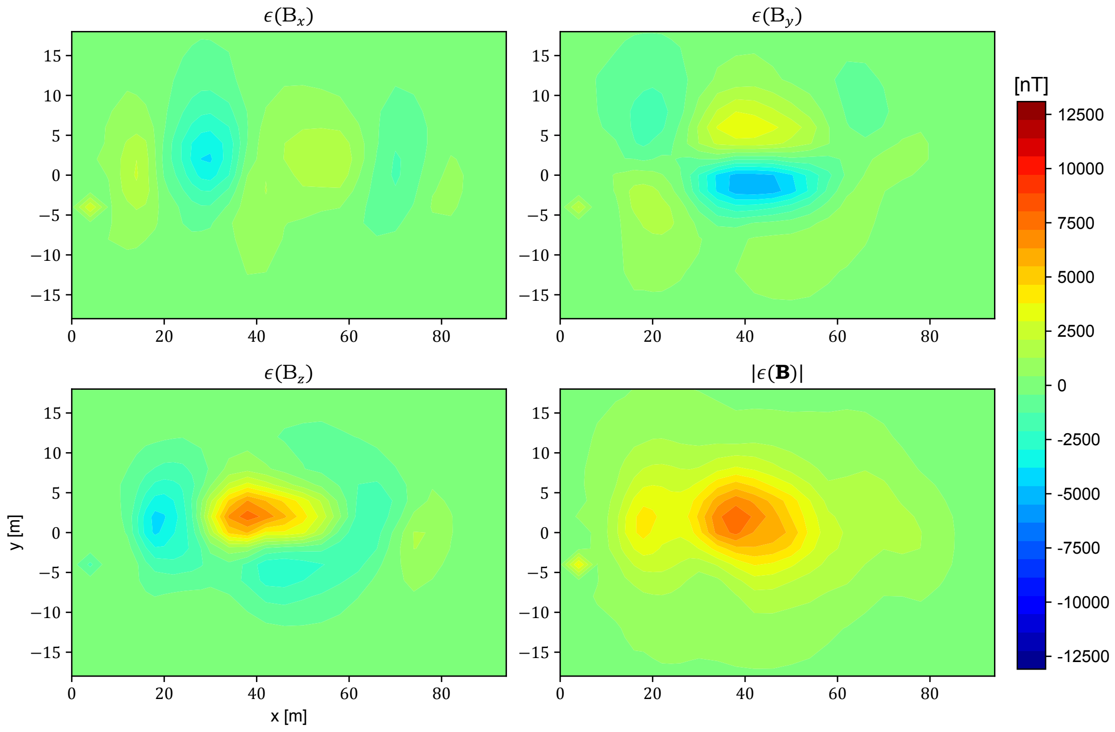


Figure 10.10: The error in the predicted field using $(\gamma, \hat{\boldsymbol{\beta}})_{\mathrm{avg}}$. The first three plots show the error in each component and the last plot shows the magnitude of the difference vector.

From the visualisations of the true magnetic field and the predicted fields in Fig. 10.5 to 10.7, it becomes clear that the two predictions resemble the true magnetic field. However, the predictions differ especially in the B_z components. This can also be observed in Fig. 10.9 and 10.10, which show the residuals. The residuals are clearly not random, but show a loose pattern. This is likely caused by one or two wrong coefficients.

The results of test cases 2 and 3 are presented in Appendix B. The searches for hyperparameters L, x_0, y_0 and z_0 were performed in a similar manner is described above. However, the final searches were performed with only 1000 samples from the parameter spaces.

The results are similar to those from test case 1, supporting the generality of these results.

10.2. Influence of noise

The results of the analysis of the effect of noise on the success of predictions are given in Fig. 10.11 and Fig. 10.12. The data for test case 1 was used.

The first figure shows the computed mean of the root mean squared errors (RMSE) in each component of the magnetic field of 50 realisation of noisy data at different noise levels, as described in Section 8.3. This figure also shows the mean of the root mean square magnitude of the errors. The noise in the positional measurements σ_{GPS} is added for 8 different values:

$$\sigma_{\text{GPS}} = 0,\ 0.1,\ 0.5,\ 1,\ 2,\ 5,\ 10 \text{ cm}.$$

Errors in the x- and y-coordinates were added with standard deviation $\sigma_{\text{GPS}}/\sqrt{2}$ and in the vertical direction this was $2\sigma_{\text{GPS}}$, as this ratio seems to be typical.

The errors in the magnetic field measurements were generated with mean 0 and standard deviation σ_B. A total of 9 different values for σ_B[2] were used:

$$\sigma_B = 0.001,\ 1,\ 10,\ 20,\ 50,\ 100,\ 200,\ 500,\ 1000 \text{ nT}.$$

The value $\sigma_B = 0.001$ nT was included since this was the precision of the measurements as supplied for this thesis.

The parameters defining the prolate spheroidal coordinate system were kept constant throughout this analysis. Their values were chosen based on the results of Section 10.1:

$$\begin{aligned} L &= 33.8 \text{ m}, \\ x_0 &= 42.7 \text{ m}, \\ y_0 &= 1.5 \text{ m}, \\ z_0 &= 0.0 \text{ m}. \end{aligned}$$

---

[2] Here, σ_B is defined as in Section 5.1. Thus σ_B is the standard deviation of the components B_x, B_y and B_z.

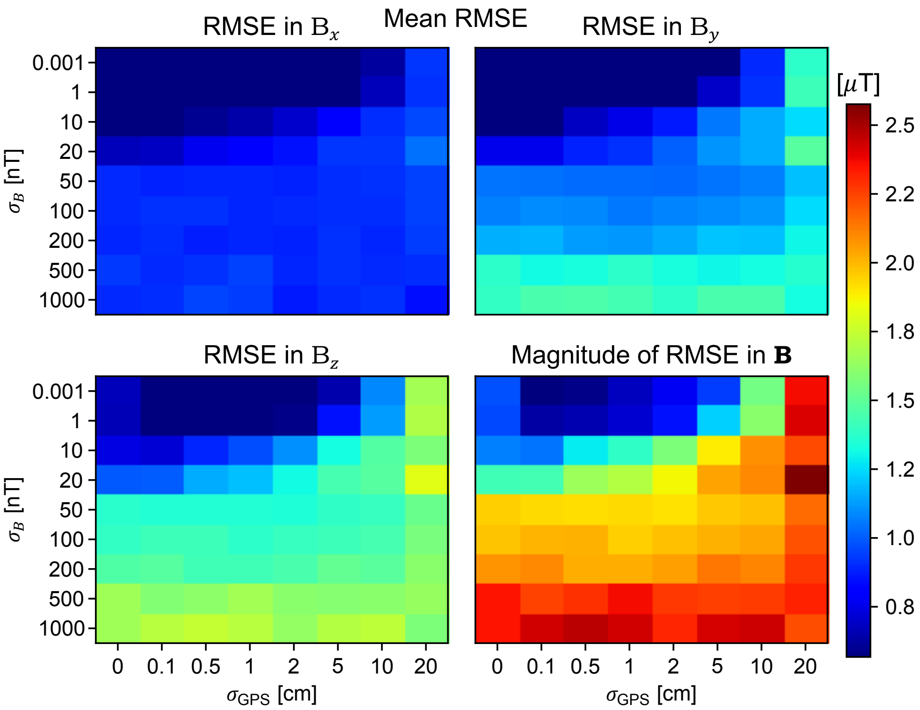


Figure 10.11: Four plots of the mean value of the root mean square error (RMSE) of each realisation of noisy data at different noise levels. The noise level in the position of measurements σ_{GPS} was increased over the horizontal axis, while the noise level of the measurements of the magnetic field σ_B was increased over the vertical axis. The fourth plot shows the mean of the root mean square magnitude of errors.

Figure 10.11 shows that the RMSE of the predictions is strongly dependent on the level of noise. While the errors in the predictions all seem to be magnitudes larger than the added noise, the model seems to predict the magnetic field rather well for very low levels of noise. Larger errors can be found when σ_B exceeds 50 nT and σ_{GPS} exceeds 10 cm.

For each of the combination of noise levels, the mean of the number of found multipole coefficients $k_{>0}$ was also computed. Moreover, the standard deviation from the mean was computed for each combination of noise levels. These two values are plot in figure 10.12.

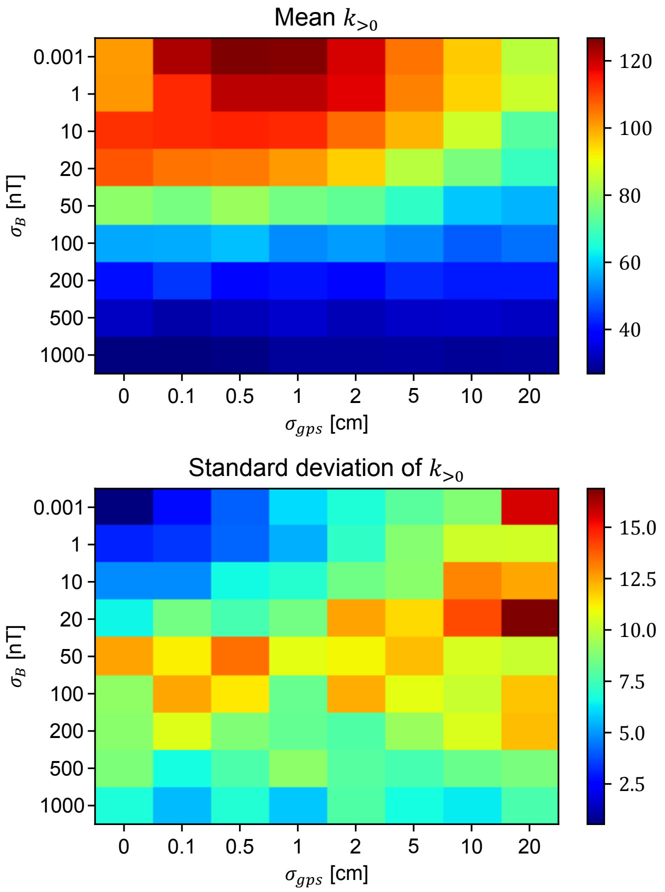


Figure 10.12: Two plots of the mean value of $k_{>0}$ and its standard deviation. The value $k_{>0}$ was computed for each realisation of noisy data at different noise levels. The noise level in the position of measurements σ_{GPS} was increased over the horizontal axis, while the noise level of the measurements of the magnetic field σ_B was increased over the vertical axis. The fourth plot shows the mean of the root mean square magnitude of errors.

Figure 10.12 shows that a large number of coefficients are found for the lowest values of σ_B. For increasing σ_B, less and less coefficients are selected. Interestingly, the plot of the standard deviation from the mean $k_{>0}$ shows that especially for $\sigma_B = 50$ nT, the number of coefficients found varies widely. This effect increases for larger values of σ_{GPS}. For the largest values σ_B and approximately for 0.5 cm $\leq \sigma_{\text{GPS}} \leq 5$ cm, the number of found coefficients varies less and less between the selected models.

The results of the analysis of the stability of the first three multipole coefficients, which describe the dipole field, are visualised in Fig. 10.13. The deviation from the mean coefficient value of each multipole coefficient was computed for every combination of noise levels.

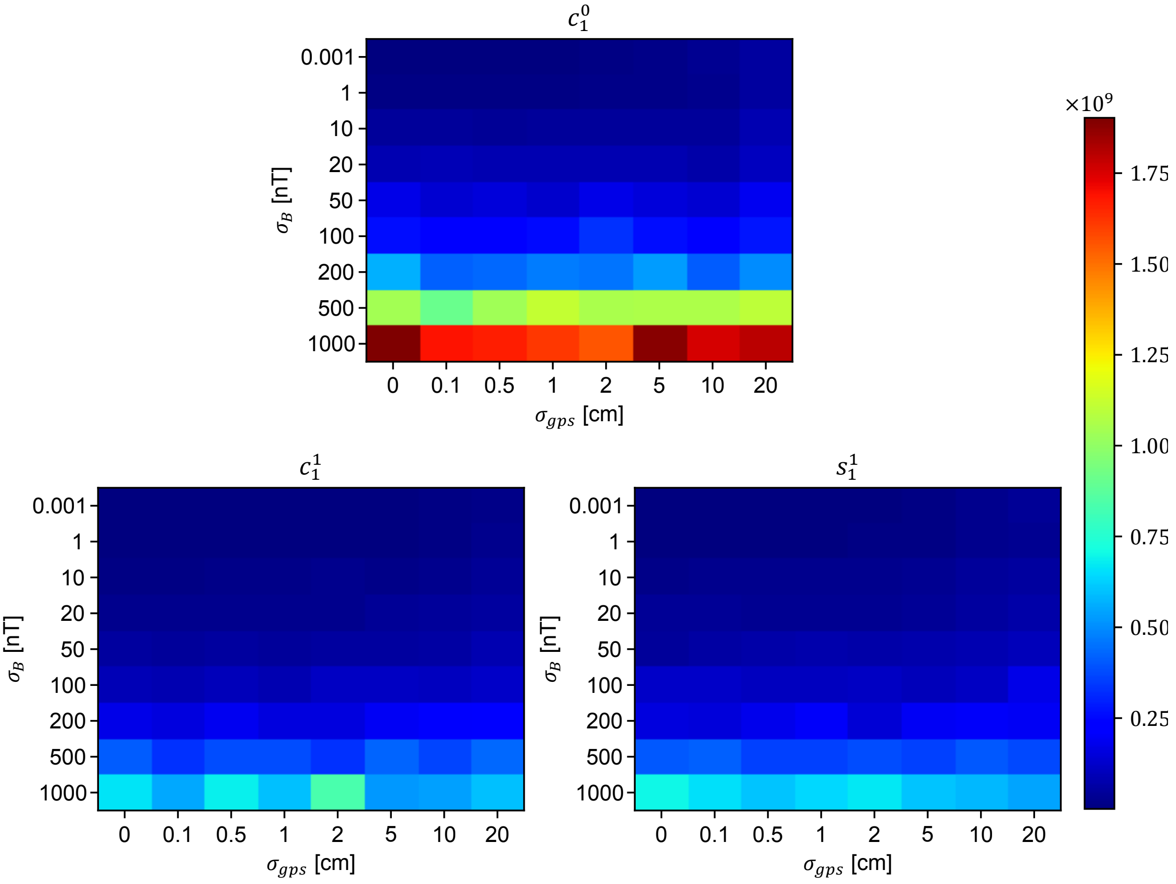


Figure 10.13: Plot of the standard deviation from the mean coefficient value for the first three multipole coefficients (c_1^0, c_1^1 and s_1^1) at different noise levels σ_B and σ_{gps}.

Fig. 10.13 shows clearly that the stability of the first three coefficients decreases for increased noise. It seems that the stability is mostly dependent on the noise level in the measurements of the magnetic field σ_B. The mean values of the coefficients for the noise level $(\sigma_B, \sigma_{\text{gps}}) = (0.001\text{ nT}, 0\text{ nT})$ are as follows:

$$\begin{aligned} c_0^1 &= -3.43 \times 10^{10}, \\ c_1^1 &= \ \ 1.02 \times 10^{10}, \\ s_1^1 &= \ \ 2.79 \times 10^{10}. \end{aligned}$$

10.3. Variation of coordinate parameters

The results of the analysis of the influence of variations in the hyperparameters L, x_0, y_0 and z_0 are visualised in Fig. 10.14 and 10.15. The values of x_0, y_0 and z_0 were varied between ± 3 m of the optimal hyperparameters found in Section 10.1. The value of L was varied within a larger range (-3 m and $+10$ m of the found optimal value), since the final range of L, within which the optimal value was searched in Section 10.1, was significantly larger than the final ranges of the other hyperparameters. For each plot, the one of the hyperparameters was varied, while the other values were kept constant. Moreover, the varied parameter was incremented linearly within the range in 300 steps, except for L which was increased in 600 steps.

A plot of the AIC for the fit on the measurement data above the ship versus the varied parameter is given in Fig. 10.14. The RMSE value of the predictions made by a model with a varied parameter against that varied parameter is given in Fig. 10.15.

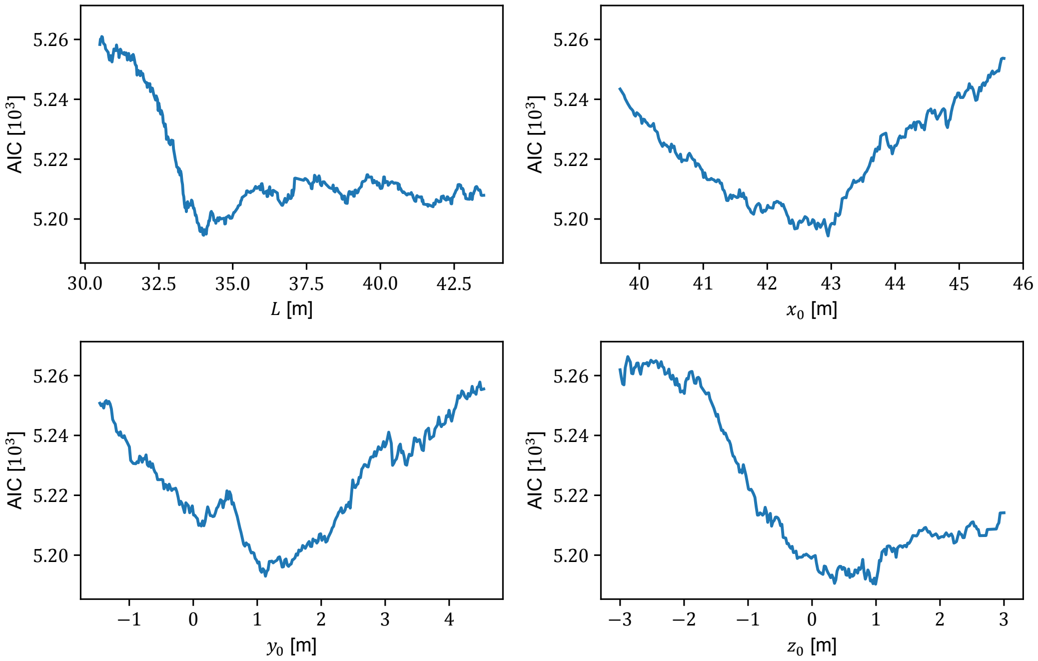


Figure 10.14: Plot of the AIC for varying hyperparameters L, x_0, y_0 and z_0. In each plot only one parameter is varying while the others are kept at their found optimal values of Section 10.1.

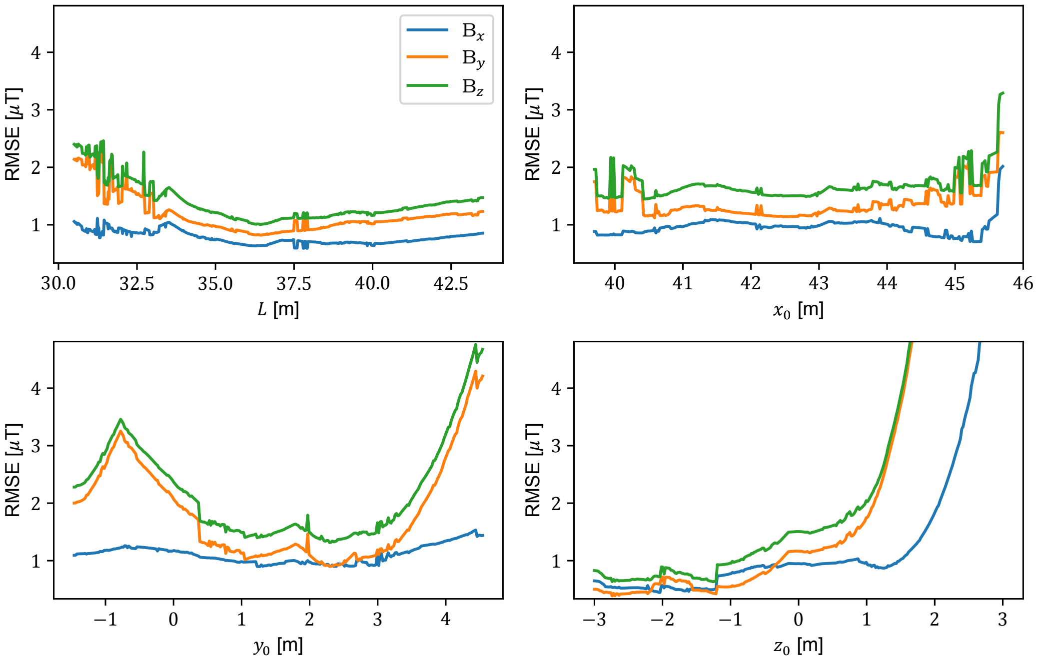


Figure 10.15: Plot of the RMSE for varying hyperparameters L, x_0, y_0 and z_0. In each plot only one parameter is varying while the others are kept at their found optimal values of Section 10.1. The blue line is the RMSE in B_x, the orange line shows the RMSE in B_y and the green line in B_z.

The value of the AIC plotted in Fig. 10.14 shows clear optimal values close to the values found in Section 10.1, which is to be expected. The RMSE values, which were plotted in Fig. 10.15, show less clear optimal parameter values. In light of the RMSE in the predictions, the hyperparameter L should be chosen larger than 35 m, while the AIC is clearly lowest for approximately 34 m. The success of the predictions seems to be most strongly dependent on variations in y_0 and z_0. Variations in the y-coordinate of the centre point of the prolate spheroidal coordinate system y_0 affect the AIC and RMSE values similarly, but the valley is larger in the RMSE plot. The plot of the AIC for varying z_0 shows that models with $z_0 < 0$ m tend to be not selected, while values of $z_0 > 1$ m appear to result in large RMSE values.

11

Discussion and Recommendations

11.1. Discussion of results

In this thesis, an algorithm was designed to develop a prediction model for the magnetic field under ships based on field measurements taken by a drone above that ship. The results of the fit and predictions of models that were produced by this algorithm were presented in Chapter 10.

Comparisons of the true magnetic fields and the predictions for different sets of simulated data have shown that this algorithm generates adequate predictions. However, the modelled fields show signs of missing coefficients: the residuals are not random but show patterns. Several causes can be identified.

Firstly, the purpose of the AIC is to select between models based on a compromise between their goodness of fit and their complexity. If a multipole term shows little significance in the measurement plane above the ship, then it is likely to be omitted. However, this omitted term might play an important role in the magnetic field underwater. This difference is due to the different rates of decay of associated Legendre functions of the first kind: higher degrees decay faster for increasing distance from the source. Therefore, higher order terms are needed when the magnetic field is described closer to the source. The danger of selecting wrong coefficients is inherent to basing a model on measurements taken in a different subspace than where the predictions are to be made.

Secondly, since the sum of multipole terms is terminated at degree $N = 17$ and order $M = 6$, a simple explanation is that the degree or the order of the missing multipoles is higher than N or M, respectively. The pattern of the residuals can be used to identify a missing multipole term: the pattern of the residuals of test case 1 resemble the pattern of either the quadrupole term c_2^0 or the octopole term c_3^0, which certainly have lower degrees and orders than their respective termination values. Therefore, this explanation seems unlikely. However, if the centre point is chosen incorrectly, this identification is not as straightforward.

From the analysis on the effect of noise on the model, some interesting conclusions can be drawn. Firstly, a rather large range of noise levels in the position measurements result in small root mean squared error values. However, a tipping point around $\sigma_{\text{GPS}} \approx 10$ cm can be identified, since for larger values of σ_{GPS}, the RMSE values increase drastically. Most real time kinematic positioning (RTK) systems, provide errors far below this point, but using standard GPS or even differential GPS (DGPS) gives noise levels beyond this tipping point. This suggests that while an RTK system is necessary in this context, financial resources may be directed elsewhere, since not the most advanced system is needed.

On the other hand, the noise level of the magnetic field measurements σ_B has shown to strongly influence the quality of the predictions made by the model. However, a more gradual increase in the RMSE values is observed. This is especially true within the range 20 nT $\leq \sigma_B \leq 200$ nT.

In this range another interesting development takes place: the standard deviation from the mean number of selected coefficients $k_{>0}$ rises and falls. A large standard deviation indicates that the values of $k_{>0}$ differ much between realisations of the noisy data. This is troublesome since the values of the coefficients should not depend on the noise. This behaviour indicates that the model is susceptible to overfitting in this range.

Outside this region the number of coefficients found seems more stable. This is especially the case for low levels of noise in both the position and field measurements, which is to be expected. However, for larger values of σ_B, this stability is rather surprising. A possible explanation is that for these high amounts of noise, the resolution in the field measurements is very low, allowing only the lowest few coefficients to be found, such as the dipole and quadrupole coefficients. This explanation is supported by the behaviour of the mean $k_{>0}$: these values seem to decrease for increasing levels of noise.

Finally, the stability of the first three (dipole) coefficients was investigated for different noise levels. The results show that the stability is almost exclusively dependent on the noise level of the magnetic field measurements σ_B.

The analysis on the effect of variations in the hyperparameters L, x_0, y_0 and z_0 yields several interesting points. The main motivation for performing this analysis was to investigate the advisable density of samples in the parameter space. The number of candidate parameter combinations must be chosen such that at least one of those candidates is likely to be located near the global minimum of the root mean square error plots. Apparently, the quality of predictions is most stongly dependent on the values of y_0 and z_0: both have shown a clear (local) minimum in the RMSE. The density of samples in the parameter space is advised to be at least 2 samples per m^4. It seems that a density that is half of what is advised, is likely to result in a favourable combination of parameters to be sampled. However, if the available computational power allows it, a higher density is advised.

Most interestingly, the plots of the AIC and RMSE values for varied hyperparameters, show restrictions should be imposed on the sampled parameter space. This is most visible for the plots where z_0 was varied: the AIC tends to be lowest for values larger than -1 m, while the RMSE tends to be lowest for values smaller than 1 m. That the AIC shows a different preference than selection based on the RMSE values, can be explained as follows. The AIC value is based on the goodness of fit in a plane above the ship. This horizontal plane must be outside of the encapsulating ellipsoid as described in Section 6.2. If z_0 is chosen too far above the water level (negative z_0), then one of the assumptions behind the model is violated, since the magnetic field is attempted to be described within this ellipsoid. Therefore, the AIC increases when z_0 is chosen too close to the measurement plane.

Oppositely, the plane underwater where predictions are made must also be outside the encapsulating ellipsoid. If z_0 is chosen too far below the water level, then predictions are attempted to be made within the ellipsoid, which is also a violation. Therefore, values of z_0 far below the water level (positive z_0) are expected to result in wrong predictions, which is reflected by a large RMSE of the field underwater. Restrictions on combinations of L and z_0 might be imposed to ensure that a generated model is able to describe the magnetic field in certain planes.

Aside from possible shortcomings of the designed algorithm, some results might also be attributed to the manner in which the data was simulated. As described in Chapter 9, the test cases were generated by performing a fit of prolate spheroidal harmonic functions on actual magnetic field measurements under a ship. The fitted multipole coefficients were then used to generate field measurements above the ship. The algorithm designed in this thesis is more complex than the methods used to generate the test cases. Differences between these methods might contribute to the some findings. Verification of the results of this thesis using actual drone measurements, is therefore advised.

Lastly, the test cases used in this thesis were made to resemble measurements around an undegaussed ship. However, an important possible application of the algorithm is the verification of the degaussing system. The same principles and assumptions behind the algorithm apply to degaussed ships. However, since the measurement errors are relatively larger when the magnetic field is weak, different success rates might be found. Therefore, future research should verify these findings with both degaussed and undegaussed ships.

11.2. Recommendations

All in all, it seems that the designed algorithm produces models that are able to predict the magnetic field under ships adequately. However, further improvements are necessary to apply the algorithm in practice. Some improvements are suggested in the following.

11.2.1. Modified AIC

The implementation of the AIC is first discussed. The most important motivation for using the AIC to perform model selection was that the AIC gives a balance between model complexity (number of coefficients) and the goodness of fit to combat overfitting. Overfitting was, however, not avoided for the practical noise levels. Perhaps the model might perform well for low budget, low requirement applications.

A modified version of Akaike's information criterion (AICc) introduces a correction term in the AIC that is quadratic in k and is often used for small sample sizes. For the problem in this thesis, the difference between the AIC and AICc is typically around 1. This difference is small, but it could be interesting to compare its behaviour with the uncorrected AIC.

11.2.2. Improved error propagation and normalisation

Another possibility to improve the model is to use a different method of error propagation. In this thesis, the errors in the position were propagated by numerical differentiation, the quality of which is dependent on several factors including the grid spacing and the errors in the measurements of the magnetic field. Also, this method is insensitive to changes in the magnetic field in between the grid points. To solve this, error propagation on the side of matrix A might be attempted.

While this improves the normalisation of the rows of A, there is also more room for improvement in the normalisation of the columns of this matrix. As discussed in Chapter 5, the LARS algorithm requires the columns of the matrix to have standard deviation equal to 1. This method normalises based on the numeric values of each column element. However, normalisation based on the expected intensity of each multipole term might prevent bias against certain terms more effectively. The size of each column element is related to the degree of its harmonic, but it is also related to the position of the measurement.

Since the LARS algorithm requires standardisation of the columns in the prescribed way, an additional step in the process either before or after fitting the coefficients can solve its shortcommings. For example, a selection of coefficients can help against overfitting. This selection could be based on the size of the coefficients after normalisation based on energies as suggested by [24].

11.2.3. Complex measurement planes

Furthermore, this thesis considered a simple horizontal measurement plane, but a more complex flight path can be investigated in further research. For example, by adding a second, smaller plane at a different height, a cross-validation-like approach to model selection could be taken. The effect of taking measurements on an ellipsoidal surface might also be interesting.

11.2.4. Hyperparameters

Improvement is not only sought in the direction of model selection, but also in the optimisation of the hyperparameters specifying the prolate spheroidal coordinate system centred on the ship. In Section 6.2, a restriction was given on the value of the focal length L. Further restrictions on the parameter space should be investigated and their implementation into the algorithm considered.

11.2.5. Rotation of the coordinate system

Lastly, it is suggest that future research also includes the rotation of the coordinate system, i.e. the assumption that the foci of the prolate spheroidal coordinate system are not located on the x-axis of the cartesian coordinate system. This was not included this thesis due to the manner in which the data was simulated, but this is an important hyperparameter in practice.

12

Conclusions

This thesis presents a structured way to approach the designing of an algorithm that produces a prediction model of the magnetic field based on measurements taken in one region around a source. More specifically, the algorithm designed in this thesis has proven to be able to make adequate predictions of the magnetic field underwater based on measurements taken by a drones. The magnetic field measurements were simulated as if they were made by commercially available sensors. Therefore, these methods can be applied with the current state of technology.

Analyses of the influence of noise showed that the success of predictions is strongly dependent on the noise level in measurements of the magnetic field. The noise level in the measurements of positions appeared to diminish the success of predictions less, but a tipping point could be identified at approximately $\sigma_{\text{GPS}} = 10$ cm.

The analysis of the influence of variations in the hyperparameters L, x_0, y_0 and z_0, which specify the prolate spheroidal coordinate system, showed that restrictions on the sampled parameter space should be implemented into the algorithm. Moreover, the parameter space of (L, x_0, y_0, z_0) is advised to be sampled at at least 2 samples per m^4.

A

Examples of magnetic field from multipoles

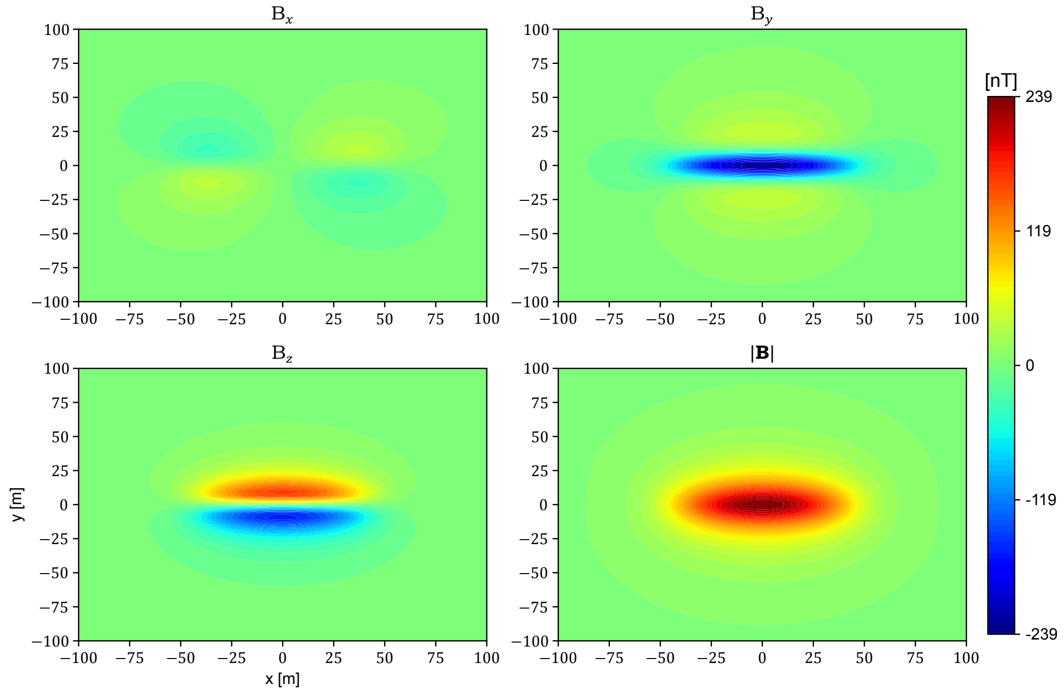


Figure A.1: Contour plot of the x-, y- and z-components and the field strength of the magnetic field $\mathbf{B}$ at $z = 15$ m and -100 m $\le x, y \le 100$ m generated by a source with $c_1^1 = 1$ and all other coefficients set to 0 and $L = 50$ m.

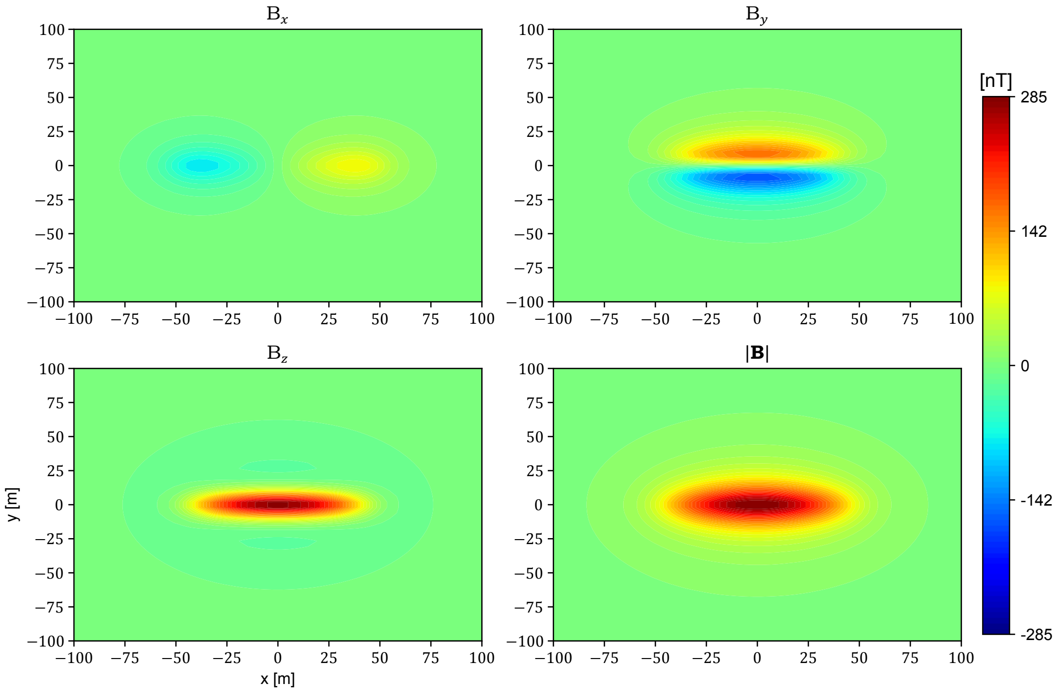


Figure A.2: Contour plot of the x-, y- and z-components and the field strength of the magnetic field $\mathbf{B}$ at $z = 15$ m and -100 m $\leq x, y \leq 100$ m generated by a source with $s_1^1 = 1$ and all other coefficients set to 0 and $L = 50$ m.

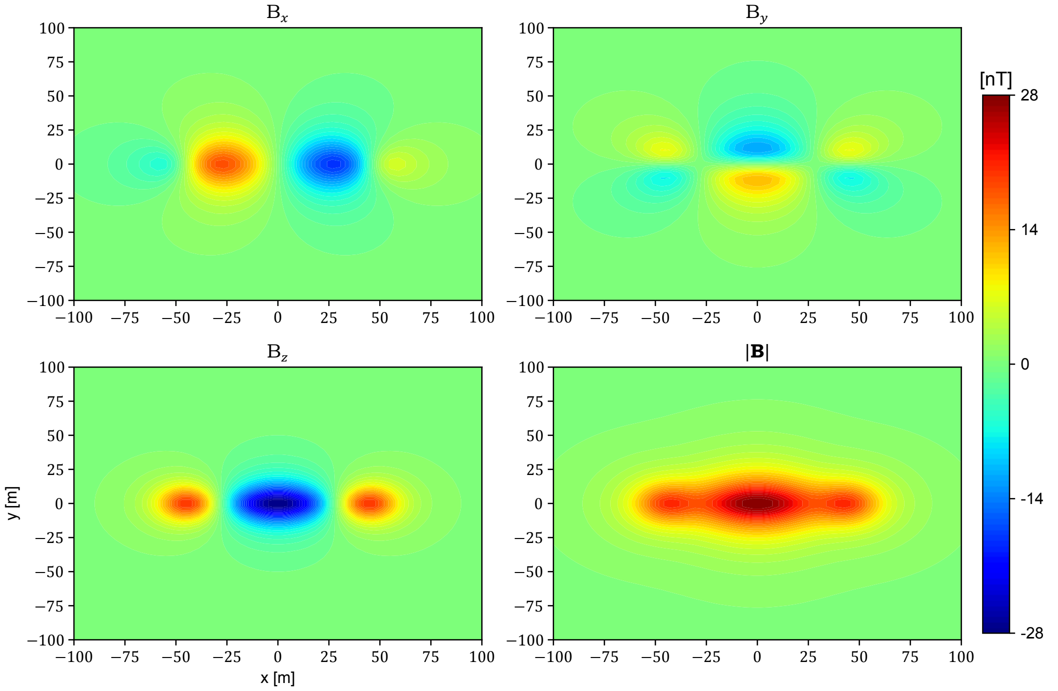


Figure A.3: Contour plot of the x-, y- and z-components and the field strength of the magnetic field $\mathbf{B}$ at $z = 15$ m and -100 m $\leq x, y \leq 100$ m generated by a source with $c_2^0 = 1$ and all other coefficients set to 0 and $L = 50$ m.

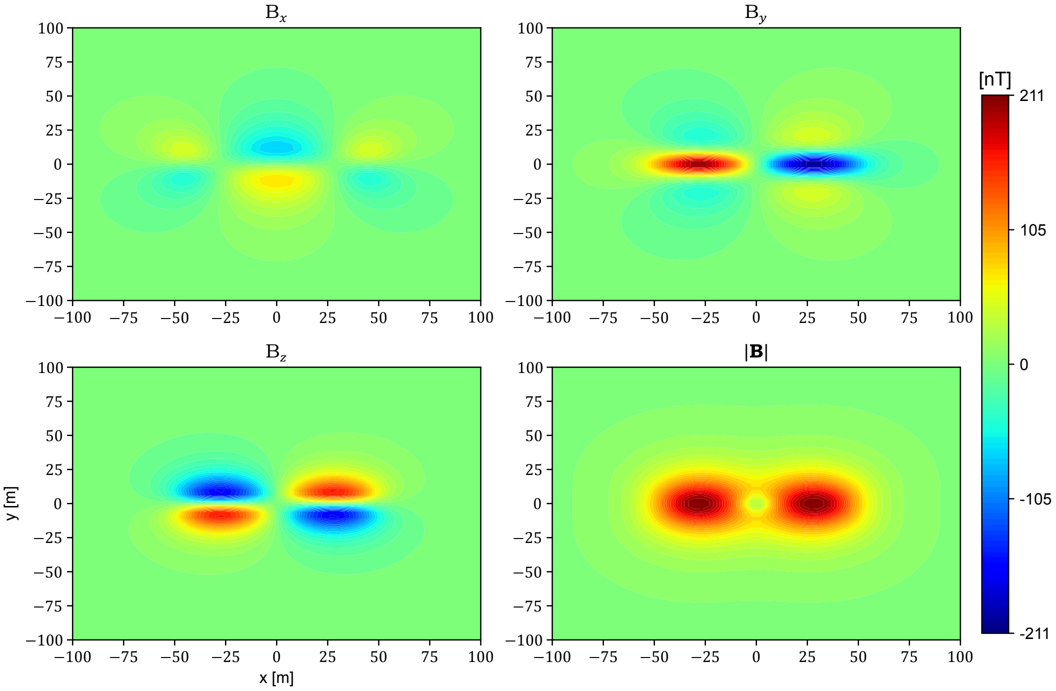


Figure A.4: Contour plot of the x-, y- and z-components and the field strength of the magnetic field $\mathbf{B}$ at $z = 15$ m and -100 m $\leq x, y \leq 100$ m generated by a source with $c_2^1 = 1$ and all other coefficients set to 0 and $L = 50$ m.

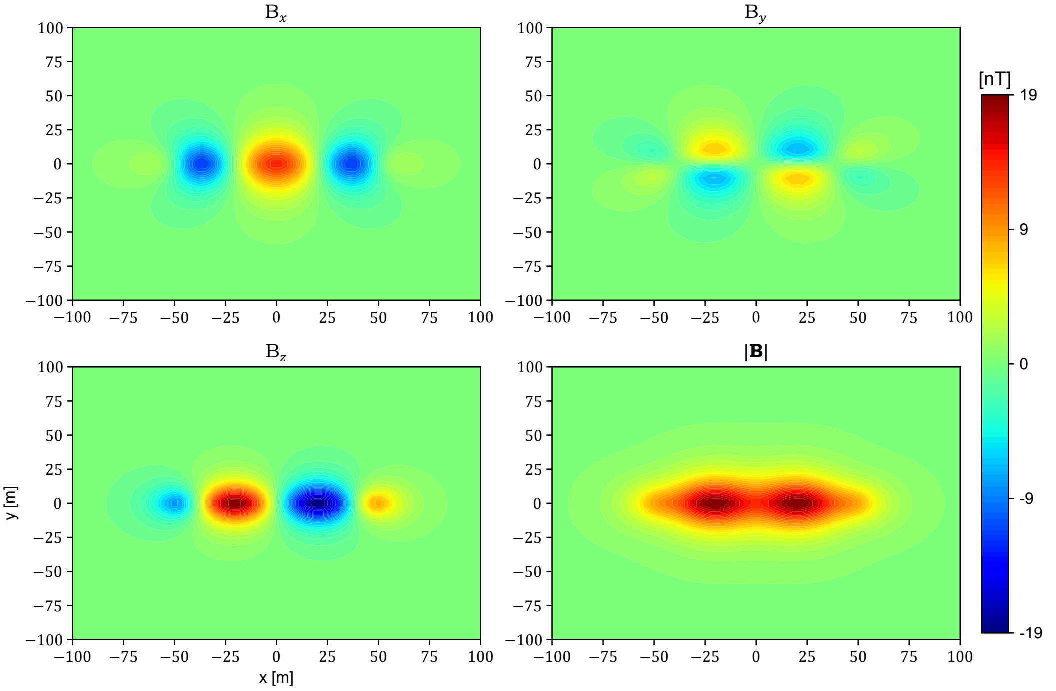


Figure A.5: Contour plot of the x-, y- and z-components and the field strength of the magnetic field $\mathbf{B}$ at $z = 15$ m and -100 m $\leq x, y \leq 100$ m generated by a source with $c_3^0 = 1$ and all other coefficients set to 0 and $L = 50$ m.

B

Results for test cases 2 and 3

B.1. Test case 2

The results of the search for the best model for test case 2 are presented in this section. The search was performed with 1000 samples drawn in a comparable subspace of the parameter space as for test case 1. Fig. B.1 to B.4 show the true and modelled magnetic field components and magnitude. Predictions from both the model with the lowest AIC value and the model with averaged hyperparameters are visualised. Fig. B.5 and B.6 show the residuals of these predictions. The values for the hyperparameters defining the prolate spheroidal coordinate system are as follows:

$$\begin{aligned} L_{\text{min}} &= 37.97 \text{ m}, & L_{\text{avg}} &= 36.78 \text{ m}, \\ x_{0,\text{min}} &= 43.01 \text{ m}, & x_{0,\text{avg}} &= 42.91 \text{ m}, \\ y_{0,\text{min}} &= -0.14 \text{ m}, & y_{0,\text{avg}} &= -0.31 \text{ m}, \\ z_{0,\text{min}} &= 0.70 \text{ m}, & z_{0,\text{avg}} &= 1.20 \text{ m}. \end{aligned}$$

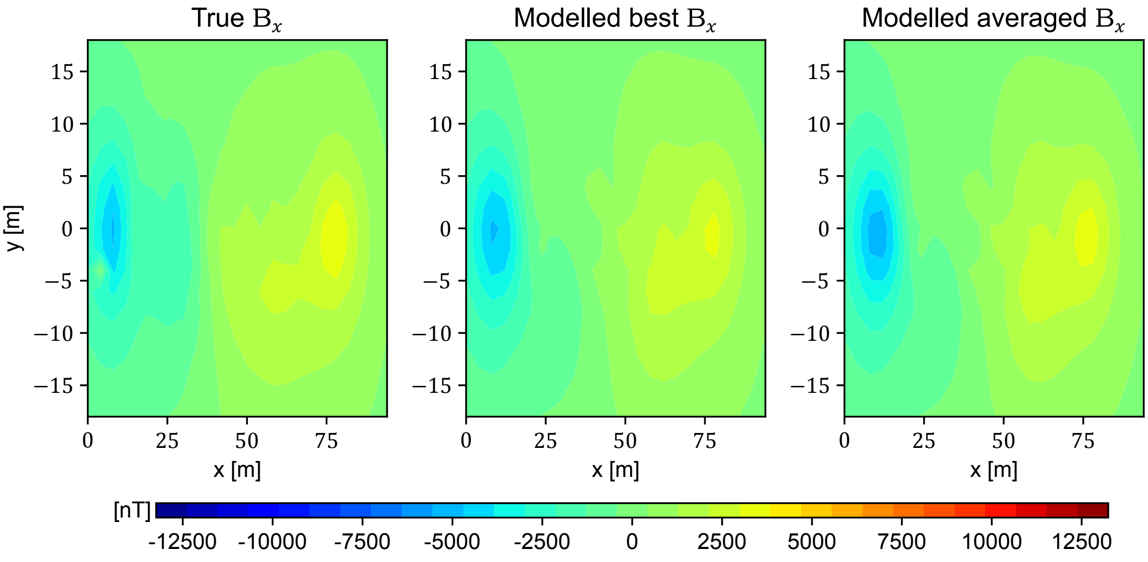


Figure B.1: Plot of the true and modelled magnetic field component B_x underwater. The first plot shows the true field, the second shows the prediction by $(\gamma, \hat{\boldsymbol{\beta}})_{\text{min}}$ and the last plot shows the prediction by $(\gamma, \hat{\boldsymbol{\beta}})_{\text{avg}}$.

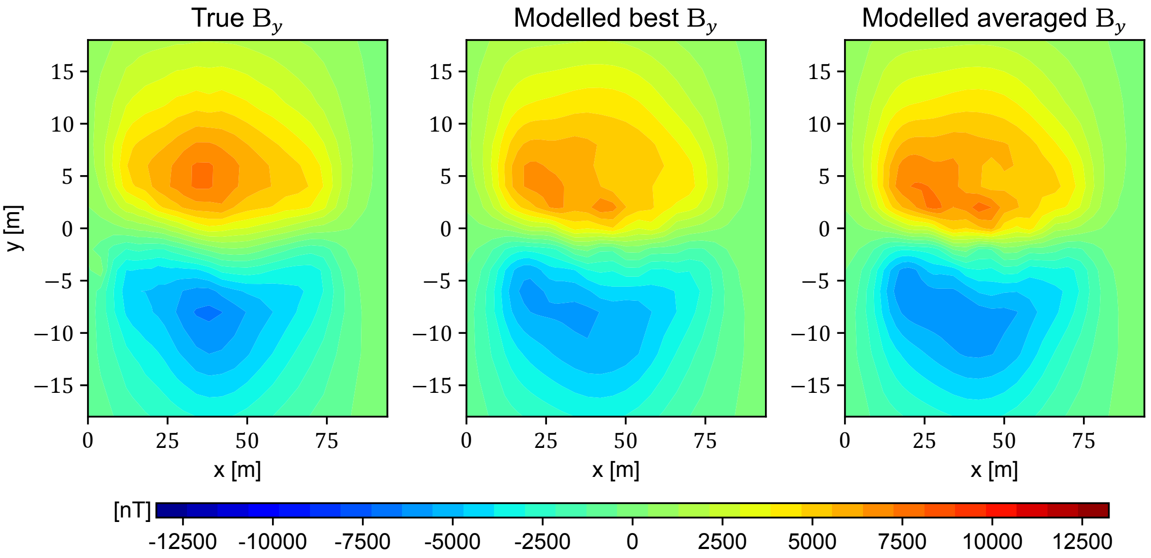

True B_y Modelled best B_y Modelled averaged B_y

Figure B.2: Plot of the true and modelled magnetic field component B_y underwater. The first plot shows the true field, the second shows the prediction by $(\gamma, \hat{\boldsymbol{\beta}})_{\text{min}}$ and the last plot shows the prediction by $(\gamma, \hat{\boldsymbol{\beta}})_{\text{avg}}$.

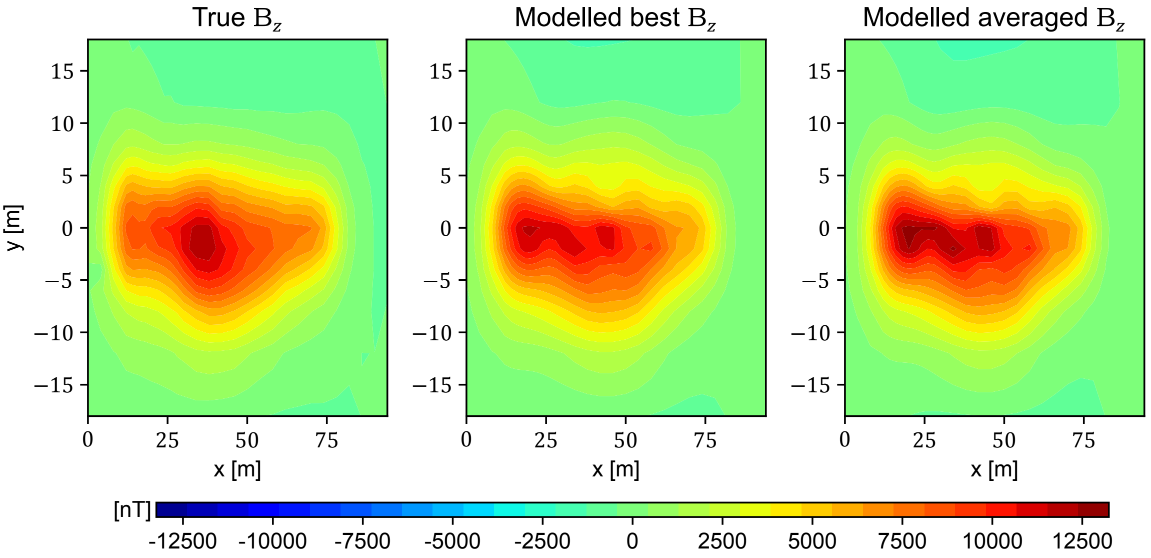

True B_z Modelled best B_z Modelled averaged B_z

Figure B.3: Plot of the true and modelled magnetic field component B_z underwater. The first plot shows the true field, the second shows the prediction by $(\gamma, \hat{\boldsymbol{\beta}})_{\text{min}}$ and the last plot shows the prediction by $(\gamma, \hat{\boldsymbol{\beta}})_{\text{avg}}$.

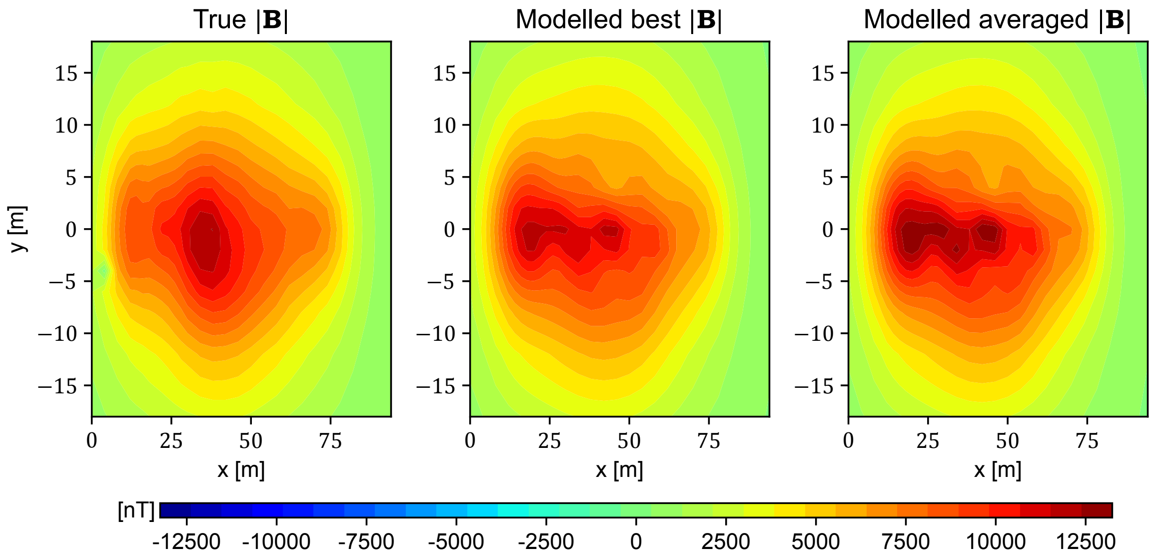


Figure B.4: Plot of the true and modelled magnetic field strength $|\mathbf{B}|$ underwater. The first plot shows the true field, the second shows the prediction by $(\gamma, \hat{\boldsymbol{\beta}})_{\text{min}}$ and the last plot shows the prediction by $(\gamma, \hat{\boldsymbol{\beta}})_{\text{avg}}$.

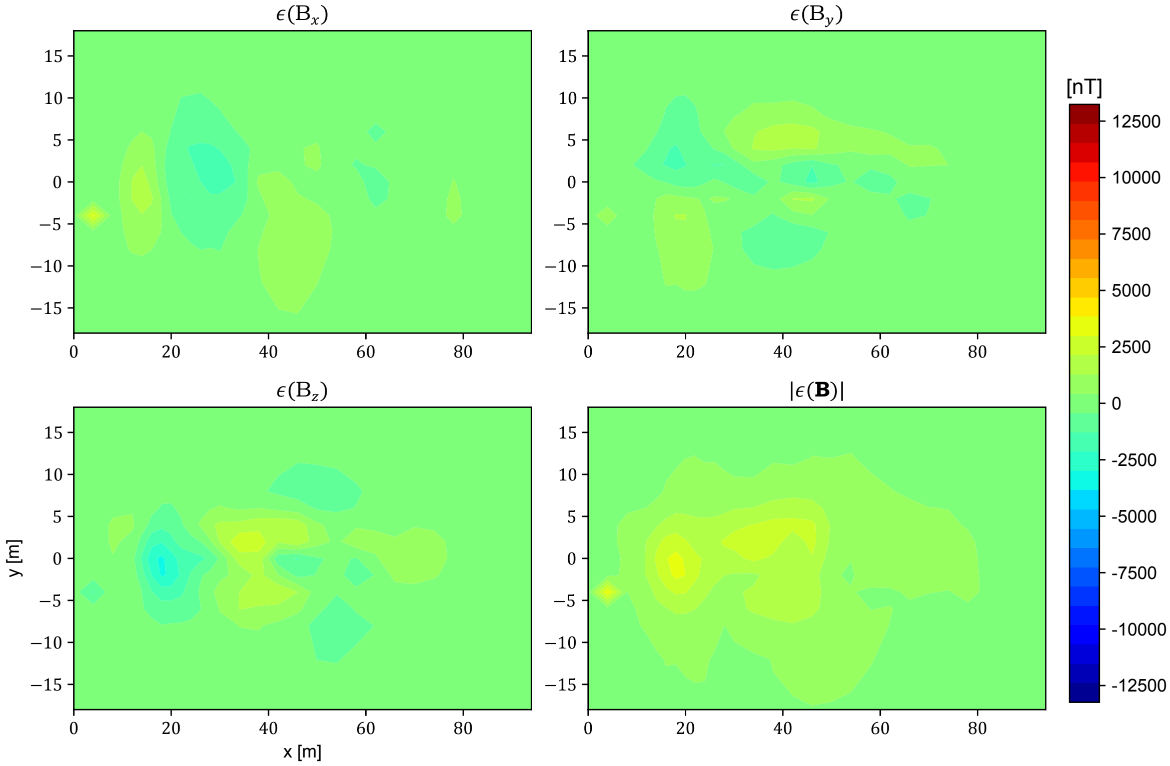


Figure B.5: The error in the predicted field using $(\gamma, \hat{\boldsymbol{\beta}})_{\text{min}}$. The first three plots show the error in each component and the last plot shows the magnitude of the difference vector.

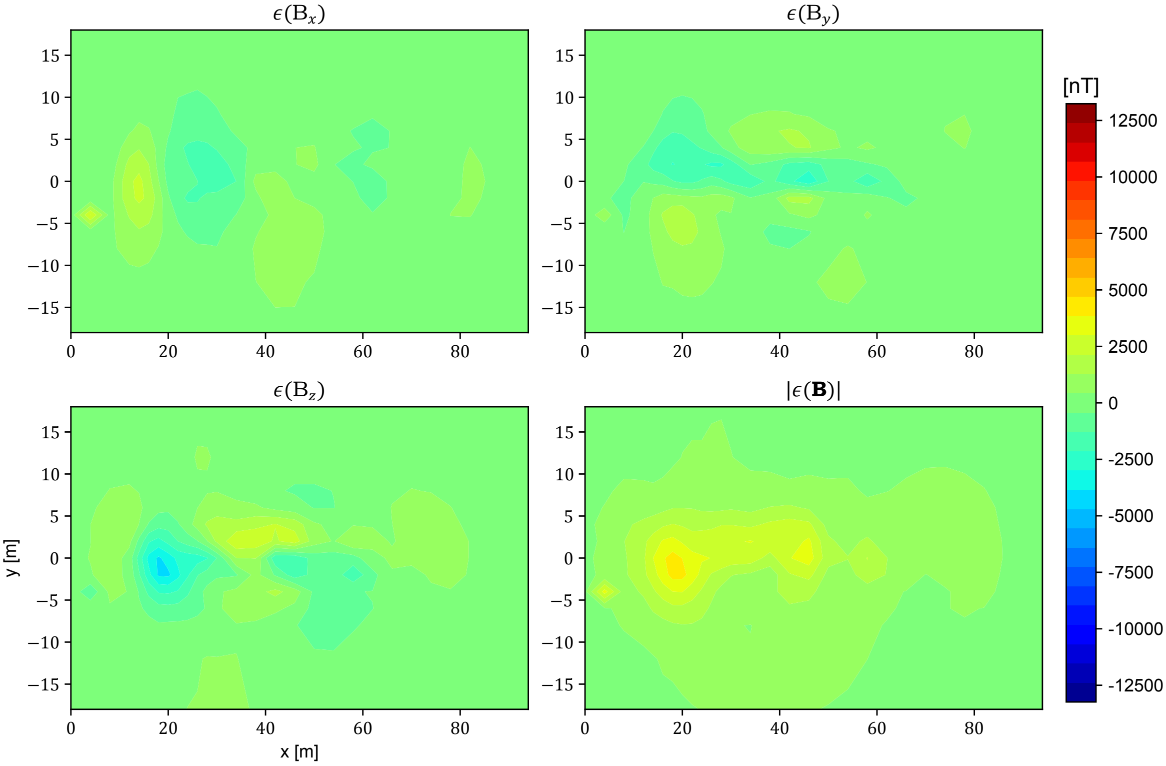


Figure B.6: The error in the predicted field using $(\gamma, \hat{\boldsymbol{\beta}})_{\text{avg}}$. The first three plots show the error in each component and the last plot shows the magnitude of the difference vector.

B.2. Test case 3

The results of the search for the best model for test case 3 are presented in this section. Again, the search was performed with 1000 samples drawn in a comparable subspace of the parameter space as for the other test cases. Fig. B.3, B.7, B.8 and B.10 show the true and modelled magnetic field components and magnitude. Predictions from both the model with the lowest AIC value and the model with averaged hyperparameters are visualised. Fig. B.11 and B.12 show the residuals of these predictions. The values for the hyperparameters defining the prolate spheroidal coordinate system are as follows:

$$\begin{aligned} L_{\text{min}} &= 53.04\ \text{m}, & L_{\text{avg}} &= 56.68\ \text{m}, \\ x_{0,\text{min}} &= 44.47\ \text{m}, & x_{0,\text{avg}} &= 42.30\ \text{m}, \\ y_{0,\text{min}} &= 2.28\ \text{m}, & y_{0,\text{avg}} &= 1.99\ \text{m}, \\ z_{0,\text{min}} &= -0.12\ \text{m}, & z_{0,\text{avg}} &= 0.47\ \text{m}. \end{aligned}$$

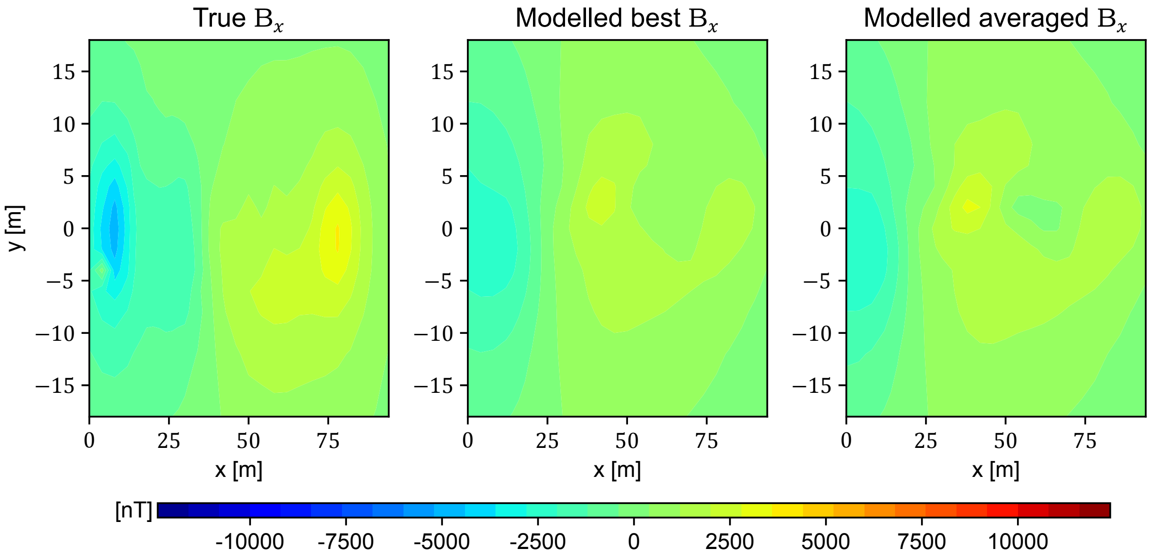


Figure B.7: Plot of the true and modelled magnetic field component B_x underwater. The first plot shows the true field, the second shows the prediction by $(\gamma, \hat{\boldsymbol{\beta}})_{\text{min}}$ and the last plot shows the prediction by $(\gamma, \hat{\boldsymbol{\beta}})_{\text{avg}}$.

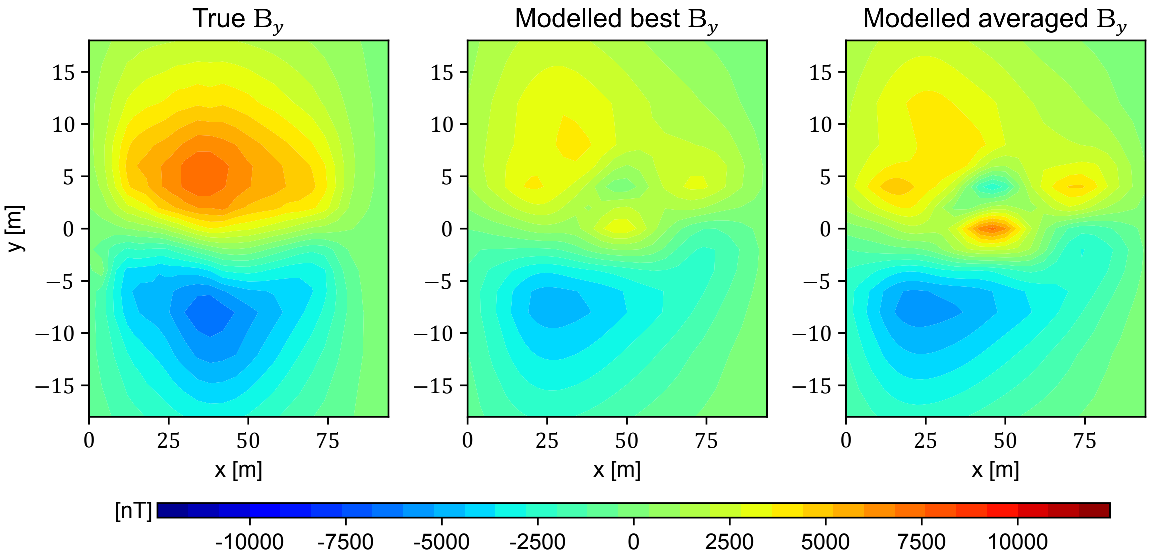


Figure B.8: Plot of the true and modelled magnetic field component B_y underwater. The first plot shows the true field, the second shows the prediction by $(\gamma, \hat{\boldsymbol{\beta}})_{\text{min}}$ and the last plot shows the prediction by $(\gamma, \hat{\boldsymbol{\beta}})_{\text{avg}}$.

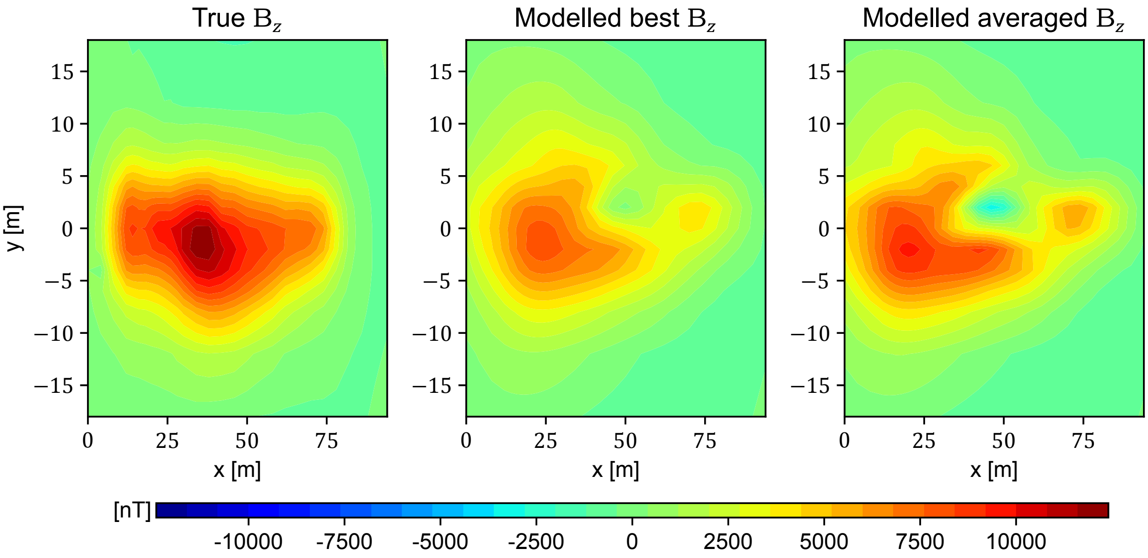


Figure B.9: Plot of the true and modelled magnetic field component B_z underwater. The first plot shows the true field, the second shows the prediction by $(\gamma, \hat{\boldsymbol{\beta}})_{\text{min}}$ and the last plot shows the prediction by $(\gamma, \hat{\boldsymbol{\beta}})_{\text{avg}}$.

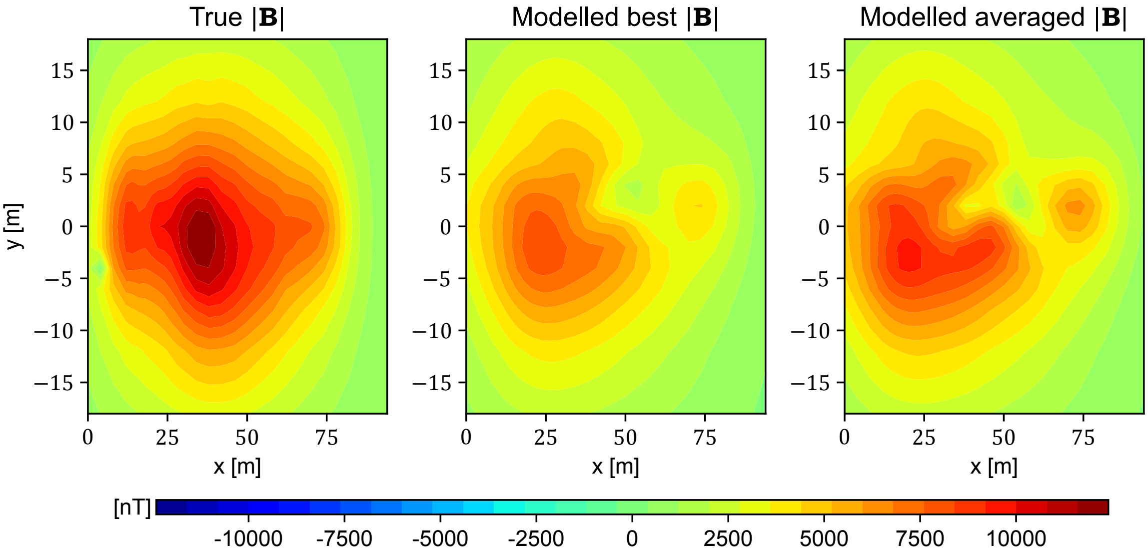


Figure B.10: Plot of the true and modelled magnetic field strength $|\mathbf{B}|$ underwater. The first plot shows the true field, the second shows the prediction by $(\gamma, \hat{\boldsymbol{\beta}})_{\text{min}}$ and the last plot shows the prediction by $(\gamma, \hat{\boldsymbol{\beta}})_{\text{avg}}$.

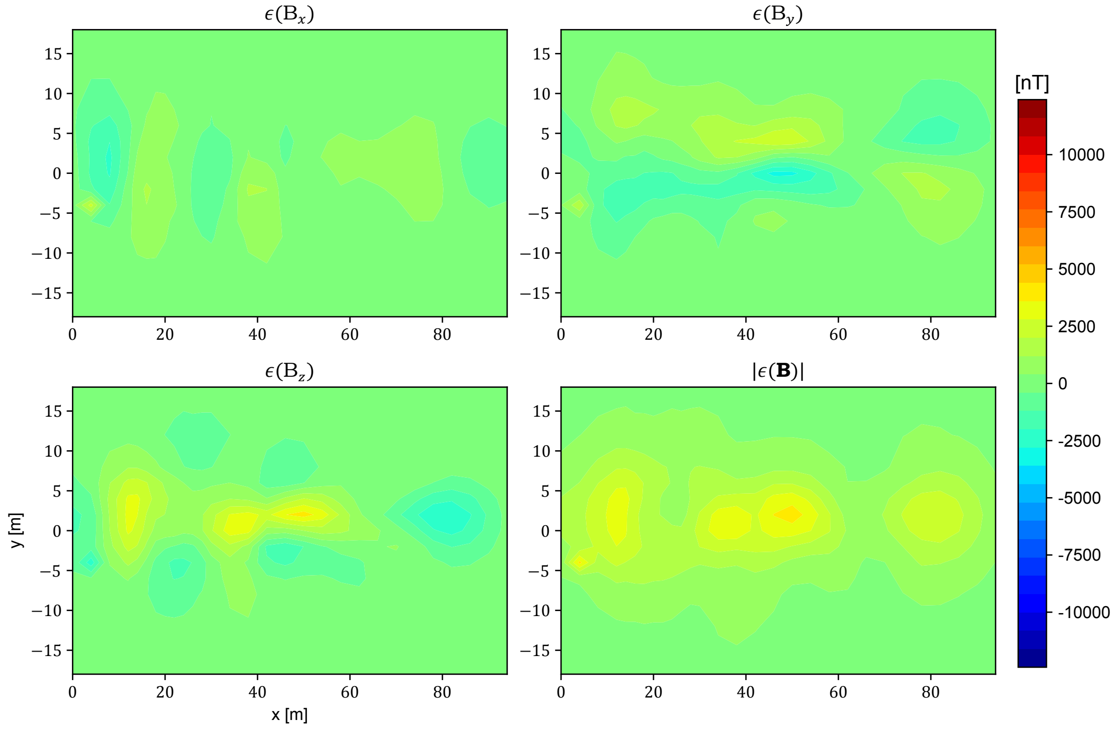


Figure B.11: The error in the predicted field using $(\gamma, \hat{\boldsymbol{\beta}})_{\mathrm{min}}$. The first three plots show the error in each component and the last plot shows the magnitude of the difference vector.

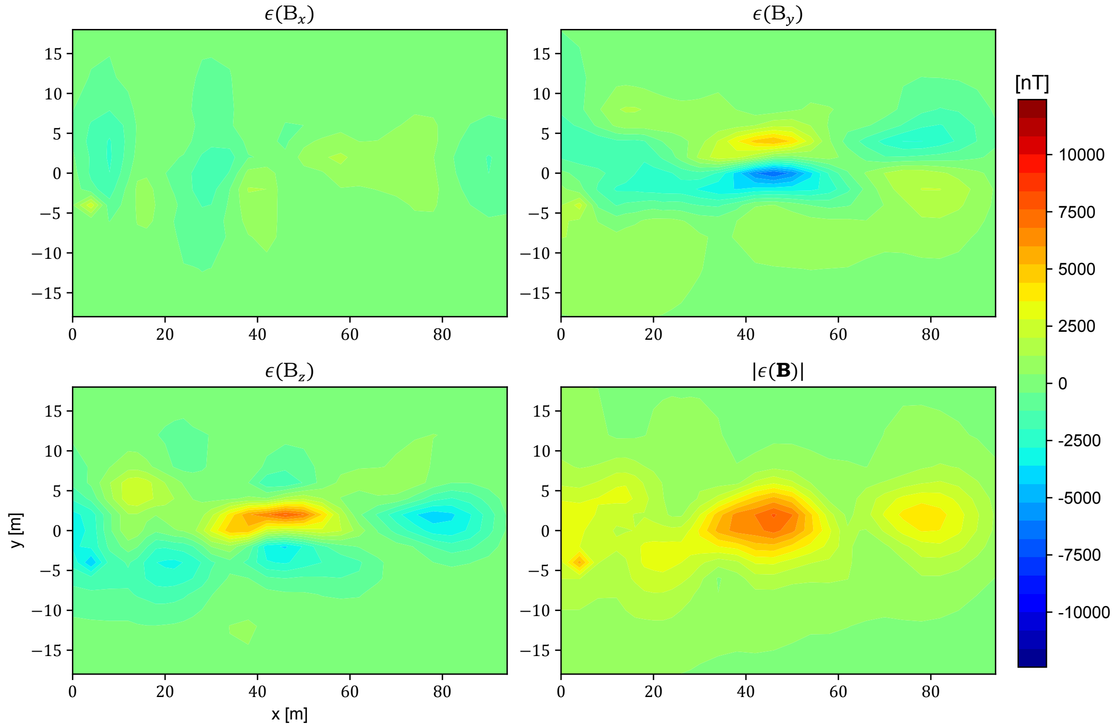


Figure B.12: The error in the predicted field using $(\gamma, \hat{\boldsymbol{\beta}})_{\mathrm{avg}}$. The first three plots show the error in each component and the last plot shows the magnitude of the difference vector.

C

Python code

The code used for this thesis can be separated into two sections: the module code and the program itself. The module code contains all reusable functions needed for coordinate transformations, computations of the matrix A defined in (3.7) and AIC calculations. This section also contains the created classes added to be used in the scikit-learn enviroment. Only the module code is included in this report. Other code can be made availible upon request.

```
import numpy as np
import scipy.special as scp
from sklearn.model_selection import ParameterSampler, RandomizedSearchCV
from sklearn.base import BaseEstimator, TransformerMixin, clone
from sklearn.exceptions import FitFailedWarning
from joblib import Parallel, delayed
import warnings
import traceback
from collections import defaultdict
from functools import partial
import time
from sklearn.linear_model import LassoLarsIC, lars_path, LassoLars
from sklearn.linear_model.base import LinearModel

# physical constants
mu_0 = 1.25663706212e–6 # [N/A^2] vacuum permeability

def pro2car(xi:float,eta:float,phi:float,L:float=1):
    '''
    Translates prolate spheroidal coordinates to cartesian coordinates

    Paramters:
        xi (float): First coordinate; xi > 1
        eta (float): Second coordinate; –1 < eta < 1
        phi (float): Third coordinate; 0<=phi<2*pi
        L (float): Focus distance to origin
    Returns:
        x,y,z (float, float, float): Cartesian coordinates
    '''
    return L*xi*eta,\
```

```
        L*np.sqrt((xi**2-1)*(1-eta**2))*np.cos(phi),\
        L*np.sqrt((xi**2-1)*(1-eta**2))*np.sin(phi)


def car2pro(x:float,y:float,z:float,L:float=1):
    '''
    Translates cartesian coordinates to prolate spheroidal coordinates

    Paramters:
        x,y,z: Cartesian coordinates
        L (float): Focus distance to origin
    Returns:
        xi,eta,phi (float, float, float): Prolate spheroidal coordinates
    '''
    return \
        (np.sqrt((L-x)**2+y**2+z**2)+np.sqrt((L+x)**2+y**2+z**2))/(2*L),\
        (np.sqrt((L+x)**2+y**2+z**2)-np.sqrt((L-x)**2+y**2+z**2))/(2*L),\
        np.arctan2(z,y)

def multipolematrix(N:int, xi:list, eta:list, phi:list):
    '''
    Returns the Jx(N^2+N) matrix of multipole coefficients for prolate
    spheroidal expansion.

    Paramters:
        N (int): Maximum degree of evaluated legendre functions
        coords (list(numpy.array)): list of numpy arrays (or tuples) with
            (xi,eta,phi) in prolate spheroidal coordinates

    Returns:
        result: J x N(N+2) matrix
    '''

    #Create empty matrix
    J = len(xi)
    shape = (J, N**2+2*N)
    result = np.zeros(shape)    #first operation must be addition, then
                                #multiplication

    for j in range(J):       #loop over observations
        P = scp.lpmn(N,N,eta)[0]
        Q = scp.lqmn(N,N,xi)[0]
        i = 0   #index of coefficient
        for n in range(N+1):    #loop over degrees
            if(n==0):   #monopole term is ignored
                continue
            else:
                for m in range(n+1):  #loop over orders from 0 to n
                    result[j,i] = P[:,m,n]*Q[:,m,n]*np.cos(m*phi)
                    i += 1
                    if m > 0:
```

```
                        result[j,i] = P[:,m,n]*Q[:,m,n]*np.sin(m*phi)
                        i += 1
    return np.asmatrix(result)

def Hmultipolematrix(N:int, M:int, xi:list, eta:list, phi:list,
                     L:float, d:list):

    #Create empty matrix
    J = len(xi)
    k = N**2+2*N if N<=M else -M**2+2*N*M+N+M
    shape = (J, k)
    h = np.zeros(shape)

    p = np.asarray([scp.lpmn(N,N,e) for e in eta])
    P = p[:,0,:,:]
    dP= p[:,1,:,:]
    q = np.asarray([scp.lqmn(N,N,x) for x in xi])
    Q = q[:,0,:,:]
    dQ= q[:,1,:,:]
    del p,q

    d0 = (d==0); eta0 = eta[d0];xi0 = xi[d0];phi0=phi[d0]
    d1 = (d==1); eta1 = eta[d1];xi1 = xi[d1];phi1=phi[d1]
    d2 = (d==2); eta2 = eta[d2];xi2 = xi[d2];phi2=phi[d2]

    i = 0   #index of coefficient
    for n in range(N+1):    #loop over degrees
        if(n==0):   #monopole term is ignored
            continue
        else:
            for m in range(min(n+1,M+1)):  #loop over orders from 0 to min(n,M)
                h[d0,i] = -eta0/L*(xi0**2-1)/(xi0**2-eta0**2)*P[d0,m,n] \
                        *dQ[d0,m,n]*np.cos(m*phi0)\
                    - xi0/L*(1-eta0**2)/(xi0**2-eta0**2)*dP[d0,m,n]*Q[d0,m,n] \
                        *np.cos(m*phi0)

                h[d1,i] = -xi1/L*np.sqrt((xi1**2-1)*(1-eta1**2))/(xi1**2-eta1**2) \
                        *np.cos(phi1)*P[d1,m,n]*dQ[d1,m,n]*np.cos(m*phi1) \
                    + eta1/L*np.sqrt((xi1**2-1)*(1-eta1**2))/(xi1**2-eta1**2) \
                        *np.cos(phi1)*dP[d1,m,n]*Q[d1,m,n]*np.cos(m*phi1) \
                    - 1/(L*np.sqrt((xi1**2-1)*(1-eta1**2)))*np.sin(phi1) \
                        *m*Q[d1,m,n]*P[d1,m,n]*np.sin(m*phi1)

                h[d2,i] = -xi2/L*np.sqrt((xi2**2-1)*(1-eta2**2))/(xi2**2-eta2**2) \
                        *np.sin(phi2)*P[d2,m,n]*dQ[d2,m,n]*np.cos(m*phi2) \
                    + eta2/L*np.sqrt((xi2**2-1)*(1-eta2**2))/(xi2**2-eta2**2) \
                        *np.sin(phi2)*dP[d2,m,n]*Q[d2,m,n]*np.cos(m*phi2) \
                    + 1/(L*np.sqrt((xi2**2-1)*(1-eta2**2)))*np.cos(phi2) \
                        *m*Q[d2,m,n]*P[d2,m,n]*np.sin(m*phi2)
                i += 1
                if m > 0:
```

```
                    h[d0,i] = -eta0/L*(xi0**2-1)/(xi0**2-eta0**2)*P[d0,m,n] \
                            *dQ[d0,m,n]*np.sin(m*phi0) \
                        - xi0/L*(1-eta0**2)/(xi0**2-eta0**2)*dP[d0,m,n] \
                            *Q[d0,m,n]*np.sin(m*phi0)

                    h[d1,i] = -xi1/L*np.sqrt((xi1**2-1) \
                             *(1-eta1**2))/(xi1**2-eta1**2)*np.cos(phi1) \
                             *P[d1,m,n]*dQ[d1,m,n]*np.sin(m*phi1) \
                        + eta1/L*np.sqrt((xi1**2-1)*(1-eta1**2))/(xi1**2-eta1**2)\
                            *np.cos(phi1)*dP[d1,m,n]*Q[d1,m,n]*np.sin(m*phi1) \
                        + 1/(L*np.sqrt((xi1**2-1)*(1-eta1**2)))*np.sin(phi1) \
                            *m*Q[d1,m,n]*P[d1,m,n]*np.cos(m*phi1)

                    h[d2,i] = -xi2/L*np.sqrt((xi2**2-1)*(1-eta2**2))/ \
                            (xi2**2-eta2**2)*np.sin(phi2)*P[d2,m,n] \
                            *dQ[d2,m,n]*np.sin(m*phi2) \
                        + eta2/L*np.sqrt((xi2**2-1)*(1-eta2**2))/ \
                            (xi2**2-eta2**2)*np.sin(phi2)*dP[d2,m,n] \
                            *Q[d2,m,n]*np.sin(m*phi2) \
                        - 1/(L*np.sqrt((xi2**2-1)*(1-eta2**2)))*np.cos(phi2) \
                            *m*Q[d2,m,n]*P[d2,m,n]*np.cos(m*phi2)
                    i += 1
    return np.asmatrix(h)

def Bmatrix(coords,dx,dy,dz,L,rot,N,M,k):
    x = coords[:,0] - dx     # translation
    y = coords[:,1] - dy     # translation
    xr = x*np.cos(rot)-y*np.sin(rot)
    yr = y*np.cos(rot)+x*np.sin(rot)
    x = xr
    y = yr
    z = coords[:,2] - dz
    d = coords[:,3]
    (xi, eta, phi) = car2pro(x,y,z,L)
    h = Hmultipolematrix(N,M,xi.flatten(),eta.flatten(),phi.flatten(),L,d)
    b = mu_0 * h
    return b


def pseudoinverse(matrix: np.matrix):
    return (matrix.H*matrix).I*matrix.H

def routeeta(start: float, stop: float, num = 50, xi = 10, phi = 0,
             incllast = True):
    '''
    For constant xi and phi, a list of 'steps' prolate spheroidal coordinates
    is returned starting from -1 <= start < stop <= 1
    '''
    return (np.ones(num)*xi,np.linspace(start, stop, num, endpoint = incllast),
            np.ones(num)*phi)
```

```
class MatrixBuilder(BaseEstimator, TransformerMixin):
    def __init__(self,dx,dy,dz,L,rot,N,M,k):
        self.dx= dx
        self.dy= dy
        self.dy= dy
        self.dz= dz
        self.L = L
        self.rot=rot
        self.N = N
        self.M = M
        self.k = k

    def fit(self,X,y=None):
        return self

    def transform(self,X):
        return Bmatrix(X,self.dx,self.dy,self.dz,self.L,
                       self.rot,self.N,self.M,self.k)

class PSHTransformer(BaseEstimator, TransformerMixin):
    def __init__(self,N,M,xi0=np.sqrt(2)):
        self.N = N
        self.M = M
        self.xi0=xi0

    def fit(self,X,y=None):
        k = (self.N**2+2*self.N if self.N<=self.M
             else -self.M**2+2*self.N*self.M+self.N+self.M)

        Q,dQ = scp.lqmn(self.N,self.N,self.xi0)

        self.w = np.zeros(k)
        i = 0
        for n in range(self.N+1):
            if n == 0:
                continue
            for m in range(min(n+1,self.M+1)):
                w = (-(2*np.pi*(2 if m==0 else 1))/(2*n+1)
                     * np.math.factorial(n+m)/np.math.factorial(n-m)
                     * (self.xi0**2-1) * Q[m,n] * dQ[m,n]
                    )
                self.w[i] = w
                i += 1
                if m > 0:
                    self.w[i] = w
                    i += 1

        return self

    def transform(self,X):
```

```
        return np.asarray(X) / self.w[np.newaxis,:]

    def inverse_transform(self,X):
        return np.asarray


class Standardiser(BaseEstimator,TransformerMixin):
    def __init__(self, w):
        self.w = w

    def fit(self,X,y=None):
        return self

    def transform(self,X):
        ws = self.w[:,np.newaxis]
        r = np.asarray(X) * ws
        return r



class StdScaler(BaseEstimator,TransformerMixin):
    def __init__(self):
        pass

    def fit(self,X,y=None):
        self.scale_ = np.std(X+np.mean(X,axis=0),axis=0)
        return self

    def transform(self,X):
        return np.asarray(X) / self.scale_

def AIC(y,X,coef):
    '''
    Calculate Akaike's Information Criterion (AIC)

    Parameters:
        y: 1D array of targets
        X: 2D array of data
        coef: 1D array of coefficients

    Returns:
        AIC: Akaike's Information Criterion (scalar)
    '''

    y = np.asarray(y).reshape(-1) # reshape y to 1D array
    R = y - np.dot(X, coef)  # residuals

    # find number of coefficients unequal to zero
    k = np.sum((np.abs(coef)>np.finfo(coef.dtype).eps))

    return np.sum(R**2) + 2*k
```

```
class RandomizedSearchAIC(RandomizedSearchCV):
    ’’’
    Based on the RandomizedSearchCV from the sckit−learn module. This
    randomized search uses the Akaike Information Criterion as a criterion for
    model selection.
    ’’’
    def __init__(self, estimator, param_distributions, *, n_iter=10,
                 scoring=None, n_jobs=None, refit=True,
                 cv=[(slice(None), slice(None))], verbose=0,
                 pre_dispatch=’2*n_jobs’,
                 random_state=None, error_score=np.nan,
                 return_train_score=False):
        self.param_distributions = param_distributions
        self.n_iter = n_iter
        self.random_state = random_state
        self.verbose = verbose
        super().__init__(
            estimator=estimator, scoring=scoring,
            param_distributions=param_distributions,
            n_jobs=n_jobs, refit=refit, cv=cv, verbose=verbose,
            pre_dispatch=pre_dispatch, error_score=error_score,
            return_train_score=return_train_score, n_iter=n_iter)

    def _run_search(self, evaluate_candidates):
        ’’’
        Search n_iter candidates from param_distributions
        ’’’
        evaluate_candidates(ParameterSampler(
            self.param_distributions, self.n_iter,
            random_state=self.random_state))

    def _fit_and_score(estimator, X, y, parameters):
        error_score = 1e6
        if parameters is not None:
            estimator = estimator.set_params(**parameters)

        start_time = time.time()
        try:
            estimator.fit(X,y)
        except Exception as e:
            fit_time = time.time() − start_time
            score_time = 0.0

            test_score = error_score
            coef_ = np.zeros(191) #TODO: make varibale
            est = clone(estimator)
            warnings.warn(”Estimator␣fit␣failed.␣The␣score␣on␣this␣train−test”
                          ”␣partition␣for␣these␣parameters␣will␣be␣set␣to␣{0}.␣”
                          ”Details:␣\n%s”.format(error_score,
                                                  traceback.format_exc()),
```

```
                          FitFailedWarning)
            print(e)
        else:
            fit_time = time.time() - start_time
            test_score = AIC(y,
                             estimator.steps[2][1].transform(
                              estimator.steps[1][1].transform(
                                      estimator.steps[0][1].transform(X))),
                              estimator.steps[3][1].coef_)
            score_time = time.time() - start_time - fit_time
            coef_ = estimator.steps[3][1].coef_
            est = estimator
        return [test_score, fit_time, score_time, coef_, est]

    def _format_results(self, candidate_params, out):
        n_candidates = len(candidate_params)
        (AIC_vals, fit_time_vals, score_times_vals,
         coefficients, estimators) = zip(*out)
        results = {}
        results['AIC'] = np.asarray(AIC_vals)
        results['fit_time'] = np.asarray(fit_time_vals)
        results['score_times'] = np.asarray(score_times_vals)
        results['params'] = candidate_params
        results['coef'] = np.asarray(coefficients).T
        results['estimator'] = list(estimators)

        param_results = defaultdict(partial(np.array,np.zeros(n_candidates)))

        for cand_idx, params in enumerate(candidate_params):
            for key, value in params.items():
                param_results['param_{:s}'.format(key)][cand_idx] = value

        results.update(param_results)

        return results


    def fit(self, X, y=None, *, groups=None, **fit_params):
        '''
        Cross-validation and multimetric selection are not implemented (yet).
        '''
        estimator = self.estimator
        base_estimator = clone(self.estimator)

        parallel = Parallel(n_jobs=self.n_jobs,
                            pre_dispatch=self.pre_dispatch)

        results={}
        with parallel:
            all_candidate_params = []
            all_out = []
```

```
            def evaluate_candidates(candidate_params):
                candidate_params = list(candidate_params)
                n_candidates = len(candidate_params)

                if self.verbose > 0:
                    print(”Fitting␣{}␣candidates.”.format(n_candidates))

                out = parallel(
                        delayed(RandomizedSearchAIC._fit_and_score)(
                                    clone(base_estimator),
                                    X,y,params)
                               for (cand_idx,params) in enumerate(
                                       candidate_params))

                all_candidate_params.extend(candidate_params)
                all_out.extend(out)

                nonlocal results
                results = self._format_results(all_candidate_params, all_out)
                return results

            self._run_search(evaluate_candidates)

        self.results_ = results
        if self.refit:
            self.best_index_ = results[’AIC’].argmin()
            self.best_score_ = results[’AIC’][self.best_index_]
            self.best_params_= results[’params’][self.best_index_]

            self.best_estimator_ = clone(base_estimator).set_params(
                    ∗∗self.best_params_)
            self.best_estimator_.fit(X,y)

            self.average(X,y,results)


        return self

    def average(self,X,y,results):
        weights = np.exp((self.best_score_−results[’AIC’])/2)
        mask = results[’AIC’] − self.best_score_ <= 9.2 # smallest weight is ~0.01


        norm = np.sum(weights[mask])
        self.avg_params_ = {}
        self.avg_sigmas_ = {}
        for key,_ in self.best_params_.items():
            self.avg_params_[key] = np.sum(
                    results[’param_{:s}’.format(key)][mask]∗weights[mask]
                    ) / norm
```

```python
            self.avg_sigmas_[key] = np.sum(
                    (results['param_{:s}'.format(key)][mask]
                    -self.avg_params_[key])**2*weights[mask]
                    ) / norm
        self.weights_ = weights[mask] / norm
        self.avg_estimator_ = clone(self.estimator).set_params(
                **self.avg_params_)
        self.avg_estimator_.fit(X,y)
        self.avg_score_ = AIC(y,
                              self.avg_estimator_.steps[2][1].transform(
                               self.avg_estimator_.steps[1][1].transform(
                                      self.avg_estimator_.steps[0][1]
                                      .transform(X))),
                              self.avg_estimator_.steps[3][1].coef_)

class LassoLarsAIC(LassoLarsIC,LassoLars):
    def __init__(self, *, fit_intercept=True, verbose=False,
                 normalize=True, precompute='auto', max_iter=500,
                 eps=np.finfo(np.float).eps, copy_X=True, positive=False):
        super().__init__('aic', fit_intercept, verbose,
                 normalize, precompute, max_iter,
                 eps, copy_X, positive)


    def fit(self, X, y, copy_X=None):
        """Fit the model using X, y as training data.
        Parameters
        ----------
        X : array-like of shape (n_samples, n_features)
            training data.
        y : array-like of shape (n_samples,)
            target values. Will be cast to X's dtype if necessary
        copy_X : bool, default=None
            If provided, this parameter will override the choice
            of copy_X made at instance creation.
            If ``True``, X will be copied; else, it may be overwritten.
        Returns
        -------
        self : object
            returns an instance of self.
        """
        if copy_X is None:
            copy_X = self.copy_X
#        X, y = self._validate_data(X, y, y_numeric=True)

        X, y, Xmean, ymean, Xstd = LinearModel._preprocess_data(
            X, y, self.fit_intercept, self.normalize, copy_X)
        max_iter = self.max_iter

        Gram = self.precompute
```

```
alphas_, _, coef_path_, self.n_iter_ = lars_path(
    X, y, Gram=Gram, copy_X=copy_X, copy_Gram=True, alpha_min=0.0,
    method='lasso', verbose=self.verbose, max_iter=max_iter,
    eps=self.eps, return_n_iter=True, positive=self.positive)

n_samples = X.shape[0]

if self.criterion == 'aic':
    K = 2  # AIC
elif self.criterion == 'bic':
    K = np.log(n_samples)  # BIC
else:
    raise ValueError('criterion□should□be□either□bic□or□aic')

R = y[:, np.newaxis] − np.dot(X, coef_path_)  # residuals

df = np.zeros(coef_path_.shape[1], dtype=np.int)  # Degrees of freedom
for k, coef in enumerate(coef_path_.T):
    mask = np.abs(coef) > np.finfo(coef.dtype).eps
    if not np.any(mask):
        continue
    df[k] = np.sum(mask)

self.alphas_ = alphas_
# Eqns. 2.15−−16 in (Zou et al, 2007)
self.criterion_ = np.sum(R**2, axis=0) + K * df
n_best = np.argmin(self.criterion_)

self.alpha_ = alphas_[n_best]
self.coef_ = coef_path_[:, n_best]
self._set_intercept(Xmean, ymean, Xstd)
return self
```

Bibliography

[1] David J. Griffiths. *Introduction to Electrodynamics*. Pearson Education, 2013.

[2] NOS. Helderse vissers halen enorme dieptemijn naar boven. *NOS*, April 2020. URL `https://nos.nl/artikel/2331584-helderse-vissers-halen-enorme-dieptemijn-naar-boven.html`.

[3] Xu Hailin. Pla conducts anti-underwater mine warfare drill to protect maritime sovereignty. *Global Times*, June 2018. URL `http://www.globaltimes.cn/content/1107348.shtml`.

[4] John J. Holmes. *Exploitation of A Ship's Magnetic Field Signatures*. Morgan & Claypool, 2006.

[5] John J. Holmes. *Reduction of Ship's Magnetic Field Signatures*. Morgan & Claypool, 2008.

[6] Mahdi Isa, Hasril Nain, Nik Hassanuddin, Abdul Rauf Abdul Manap, Abdul Manap, Roslan Slamatt, and Mohd Hambali Anuar. An overview of ship magnetic signature and silencing technologies. *Defence S&T Technical Bulletin*, 12 2019. doi: 10.13140/RG.2.2.14643.58401.

[7] SAM Electronics GbbH. Mobile magnetic and multi-influence range - advanced degaussing and ranging system, 2010. URL `https://www.l-3mps.com/products/datasheet/SAM/1.024.pdf`. Accessed July 2020.

[8] B. Ginzburg, A. Sheinker, N. Salomonski, A. Yaniv, A. Noiman, A Naim, and M. Wolf. Estimation of a vessels magnetic signature using magnetic moment modeling method. In *MARALEC*, 2019.

[9] Milton Abramowitz and Irene A. Stegun, editors. *Handbook of mathematical functions*. United States National Bureau of Standards, 1972.

[10] Richard Harberman. *Applied Partial Differential Equations: with Fourier Series and Boundary Value Problems*. Pearson, 2005.

[11] Yanfei Wang, Anatoly G. Yagola, and Changchun Yang, editors. *Optimization and Regularization for Computational Inverse Problems*. Springer, 2010.

[12] Ifan G. Hughes and Thomas P.A. Hase. *Measurements and their Uncertainties*. Oxford University Press, 2010.

[13] C. Vuik, F.J. Vermolen, M.B. van Gijzen, and M.J. Vuik. *Numerical Methods for Ordinary Differential Equations*. Delft Academic Press, 2016.

[14] John A. Rice. *Mathematical Statistics and Data Analysis*. Brooks/Cole, 2007.

[15] Mads Stormo Nilsson. Modelling of civilian ships' ferromagnetic signature. Technical report, Norwegian Defence Research Establishment, May 2016.

[16] Bradley Efron, Trevor Hastie, Iain Johnstone, and Robert Tibshirani. Least angle regression. *The Annals of Statistics*, 32(2):407–451, 2004.

[17] Stig A.V. Synnes, P.A. Brodtkorb, and Eugene S.A.M. Lepelaars. Representing the ship magnetic field using prolate spheroidal harmonics – a comparative study of methods. In *Electromagnetic Silencing Symposium*, 2007.

[18] Kenneth P. Burnham and David R. Anderson. *Model selection and multimodel inference*. Springer-Verlag, 2002.

[19] Lars Buitinck, Gilles Louppe, Mathieu Blondel, Fabian Pedregosa, Andreas Mueller, Olivier Grisel, Vlad Niculae, Peter Prettenhofer, Alexandre Gramfort, Jaques Grobler, Robert Layton, Jake VanderPlas, Arnaud Joly, Brian Holt, and Gaël Varoquaux. API design for machine learning software: experiences from the scikit-learn project. In *ECML PKDD Workshop: Languages for Data Mining and Machine Learning*, pages 108–122, 2013.

[20] Sensys Magnetometers & Survey Solutions. Fgm3d. URL `https://sensysmagnetometer.com/products/sensors-recorder/fluxgates/fgm3d-magnetometer/`. Accessed August 2020.

[21] Bartington Instruments. Mag-13®. URL `https://www.bartington.com/wp-content/uploads/pdfs/datasheets/Mag-13_DS3143.pdf`. Accessed August 2020.

[22] DJI. D-rtk 2, 2020. URL `https://www.dji.com/nl/d-rtk-2/info`. Accessed August 2020.

[23] Emlid. Reach m2, 2020. URL `https://store.emlid.com/product/reachm2/`. Accessed August 2020.

[24] A. V. Kildishev and J. A. Nyenhuis. External magnetic characterization of marine vehicles. In *OCEANS 2000 MTS/IEEE Conference and Exhibition. Conference Proceedings (Cat. No.00CH37158)*, volume 2, pages 1145–1147 vol.2, 2000.