UNIVERSITY OF BIRMINGHAM

Bayesian inference from terrestrial gravimetry measurements of near-surface anomalies using a bespoke reversible-jump Markov chain Monte Carlo algorithm.

Anthony Rodgers

02/09/16

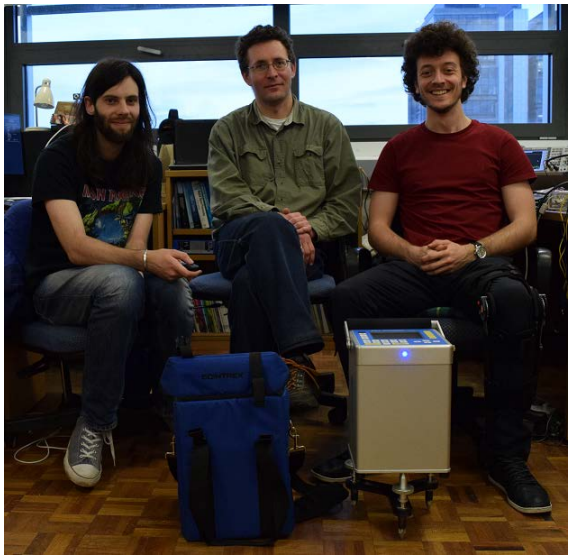
Abstract

This work describes a Bayesian algorithm developed to tackle the problem of inference from two-dimensional grids of terrestrial gravity measurements. Near-surface voids such as pipelines and tunnels were the motivating application. The forward models used to approximate the gravity signal due to these potentially complex underground anomalies were sums of simple geometrical shapes: the sphere, finite horizontal cylinder and cuboid. The model parameters of these shapes are related non-linearly to the gravity signal. The reversible-jump Markov chain Monte Carlo algorithm was used, allowing changes to the number of objects comprising the forward model. The natural parsimony of the algorithm was shown to be key for obtaining depth information without the need for arbitrary regularisation. Exploring the Bayesian posterior distribution in this way, spatial, geometrical and anomaly mass information can be obtained as outputs from the inference process, given prior information regarding the soil-anomaly density contrast. This was demonstrated both with synthetic noisy gravity and gravity gradient data and with field gravity data obtained using the Scintrex CG-5 commercial gravimeter. The methodology used to obtain field survey data using the CG-5 over multiple days is described, with discussion of the assignment of measurement uncertainty. A 134 point measurement grid was taken above two spatially separate concrete anomalies, for which volume and density information were known. The data was input into the Bayesian inference algorithm, the forward model parameters were successfully inferred within the total uncertainty.

To my parents, Patricia and David. To my girlfriend Debbie.

Acknowledgements

Many people made this work possible, perhaps too many to mention. The entire GG-ToP team, especially Emanuele Rocco and Rebecca Palmer for help in the early stages of my research regarding gravity modelling and Markov chain Monte Carlo theory. Phil Murgatroyd for his unwavering enthusiasm for gravity measurement campaigns. Kevin Ridley and Geoff De Villiers for interesting discussions on the gravity inference problem. Gareth Brown for his valuable input regarding the Markov chain Monte Carlo approach to the problem of inference. Dan Boddice and Yuriy Goncharenko for their incredible patience with me and perhaps to a greater extent the Scintrex CG-5. Without these two the CG-5 would surely have broken my spirit. My supervisor Phil Atkins, who made the experiments and general PhD life as easy as possible and who didn't raise an eyebrow to the question, 'can we rent eight concrete blocks and put them in the car-park?'.



Left to right: Dan Boddice, Yuriy Goncharenko, Me. Front: Scintrex CG-5, star of the show.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Obtaining and analysing gravity data and that of its derivatives has been widely practised for many years, providing a means of searching for minerals and other buried assets[1]. Due to the large scale of these targets, terrestrial gravity surveys have been employed to measure on the order of kilometres. However, with advances in instrument sensitivity made possible by recent developments in the field of quantum atom interferometry (AI)[2] it is now plausible to consider measuring gravity on smaller scales, for the detection of near-surface (< 10 m) anomalies. Meaning that gravity measurements can potentially aid in the detection and classification of relatively small man-made structures, such as pipelines, tunnels and mine-shafts, as a compliment to existing geophysical techniques[3]. Advances in other gravity instruments such as the Micro-electromechanical systems (MEMS) sensors[4] also promise faster, cheaper gravity surveys.

Gravity surveys provide a non-invasive means of target detection, which makes the method ideal for applications where invasive exploration can be potentially costly, both economically and socially, as in the detection of underground utility services[5]. Knowledge of the location of existing utilities is generally poor and there is a requirement to be able to detect and locate such assets in order that they be avoided during excavation, saving potential damage and disruption. The gravity method does not suffer from the problems of other near-surface geophysical techniques such as ground penetrating radar (GPR), which relies

on the transmission of electromagnetic waves into the attenuating sub-surface[6]. As a result of this, gravity surveys can be a useful complimentary data-set to obtain[7]. The limiting factors for gravity surveys are the target volume, depth and density contrast with respect to the surrounding material. Also, the time scales involved in gravity surveying are currently large in comparison to other geophysical techniques. However, near-surface anomalies with a sufficient volume and density contrast can in principle be detected using the gravity method, given sufficient surveying time and a gravity signal larger than the total survey uncertainty. Other natural targets could also be viable applications, such as near-surface sink-holes, which have the potential to cause disruption as shown in Figure 1.1. Detection of sink-holes prior to their collapse could help in preventing resource expense and potential injury.

Figure 1.1: A sink-hole causes disruption to a residential road in Dartford.

1.1 Geophysical Inference

When looking for underground targets of interest with measurements taken above the ground, we are trying to solve a problem of inference. In geophysical inference problems we aim to reconstruct a model of subsurface properties given a set of measurements that are taken above the surface. We can represent the model by a vector $\boldsymbol{\theta}$ and our data by a vector $\boldsymbol{d}$

given by,

$$\begin{aligned} \boldsymbol{d} &= [d_1, d_2, d_3, ..., d_{N_d}]^T \\ \boldsymbol{\theta} &= [\theta_1, \theta_2, \theta_3, ..., \theta_{N_\theta}]^T \end{aligned} \tag{1.1}$$

where we have a total number of model parameters N_θ, a total number of discrete measurement points N_d and T represents the matrix transpose. The measurement points are related to the model parameters via some set of functions[8],

$$\boldsymbol{d} = f(\boldsymbol{\theta}) \tag{1.2}$$

which define the forward model. The forward model functions are generally non-linear. Geophysical inference suffers from the problem of several models fitting the data equally well, often termed the problem of uniqueness[9]. The gravity problem is no exception to this, with inference being required from a finite, often small set of measurement data points[10]. We have a finite set of measurements with which to infer subsurface properties that in the real world often vary continuously in all directions. In practice we must represent the continuous functions of interest as a finite number of model parameters[11].

In order to address the uniqueness problem, assumptions need to be made about the forward model used for a particular inference problem. For the gravity inference problem, this leads to different approaches, depending on the particular goals of the inference. Two main approaches have been taken in the literature, that of the spatial domain and that of the wave number domain[12].

The wave number domain approach involves converting the gravity anomaly into a Fourier series, commonly using the Fast Fourier Transform[13]. This approach efficiently analyses large data sets. To achieve convergence high frequency components are often removed, leaving a smooth solution[14]. The datasets in which we are interested will be of more sharply defined, near-surface anomalies, limiting the usefulness of this approach. Algorithm development using the wave number domain approach is an active research area[15].

The spatial domain approach is characterised by some discretisation of the model space into a cell structure[16], with each cell given a constant, unknown density (vertical and horizontal prism structures have also been used[17]). Known as 'voxel modelling'[18], the densities of the cells are varied until a best fit to the data is found within some specified criterion. Such an approach is an effective linearisation of the problem, as the cell density is linearly related to the gravity measurement data. Often these methods are developed such that prior information is limited as far as possible[19]. However, depth weighting functions are employed to resolve the problem of the density distribution preferring to rise to the near-surface cells[20], giving erroneous values of anomaly depth. These methods generally involve the use of Tikhonov regularisation[21], which favours certain solutions that have desirable properties by introducing a regularisation parameter. The methods have had much success when applied to the search for density distributions of massive bodies and are ongoing areas of research[22][23][24]. By discretising the model space into voxels, the precision of estimates of target location is entirely dependent upon the density of the cells. In this research we are concerned primarily with the location and volume of near-surface targets. The discretisation of the near-surface into cells could be prohibitively computationally expensive. For example, a $20 \times 20 \times 10$ m model space with a voxel structure split into cells of diameter 0.5 m yields $32,000$ separate density values to be inferred. Even with the forward model formulation having a simple linear relation between the model parameters and the data, this method uses a large number of model parameters, and would provide us with relatively poor spatial resolution of our targets.

An alternative spatial approach to simplify the forward relation for the gravity inference problem is to assume the density and infer geometrical parameters of polygonal (2D) or polyhedral (3D) bodies[25]. This method is intuitively appealing for the targets under investigation here, as we may have reasonable prior information that can inform our choice of model. For example if we know a given gravity survey was carried out to search for near surface pipelines, it is reasonable to use gravity model of a finite cylinder. Making the assumption that the average ground density is spatially homogeneous and known to within

some uncertainty greatly reduces the number of model parameters required for inference, relative to the voxel model approach. However, this assumption could be suspect in the presence of near surface soil clutter. The drawback is that these geometrical model parameters are often related non-linearly to the gravity data.

In this research we formulate the gravity inference problem in a statistical Bayesian framework[26] where we interpret the data and model parameter vectors as random variables. We use forward models consisting of sums of simple shapes to approximate complex gravity measurement data. Figure 1.2 shows this idea graphically, where we have a forward model consisting of four cuboid model objects, each of which are defined by some number of model object parameters. The gravity signal of each of the model objects sum to produce a more complicated (relative to any single model object) gravity measurement grid. The challenge is to infer from our measurement grid the properties of an unknown forward model. We can

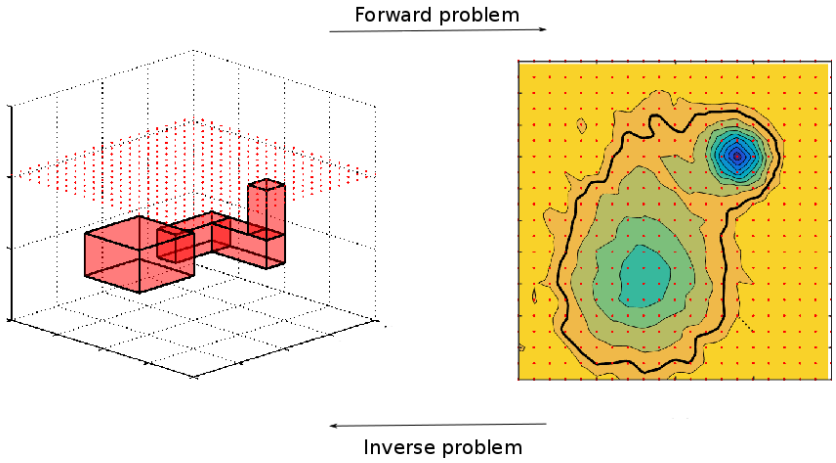


Figure 1.2: Illustration of the difference between the forward and inverse (or inference) problems.

use information available prior to data collection regarding the average ground density to limit the model space of the inference problem. This is equivalent to the regularisation of the previously mentioned approaches. We are concerned not only with the 'existence half' of the inference problem, where we wish to obtain a model that fits the data, but also the 'uniqueness half'[27], where we measure how much the model may vary whilst fitting the data, in order to understand the credibility of the model.

A bespoke reversible-jump Markov chain Monte Carlo algorithm is developed to explore the model space proportional to high probability regions of the Bayesian posterior distribution[28]. The model space has a variable number of parameters, allowing the complexity of the forward model (number of simple objects and hence model parameters) to be determined by the gravity measurement data and not defined as prior information. We show that information useful to end users regarding target location and structure can be inferred from the algorithm outputs, along with the corresponding uncertainty of the the models and parameter distributions.

Applications of the Bayesian inference framework specifically to the three dimensional gravity inference problem where the density is assumed as prior information are scarce. Work that has been conducted usually involves the voxel forward model [29][30]. More recently a Bayesian implementation has been shown, with methodology similar to this research for parameter estimation of a single buried cuboid void[31]. Other implementations have been demonstrated for inference of polygon models from single gravity measurement lines[32], employing a similar reversible-jump Markov chain Monte Carlo scheme to vary the model complexity of the polygon (number of sides).

Work has been carried out considering applications of gravity gradient measurements. Classical inference techniques (non-Bayesian) have been applied to the detection of anomalous density distributions in transport vehicles, in order to identify potential fissile materials[33].

1.2 Research Scope

This research was conduced as part of a multi disciplinary research group, the Gravity Gradient Technologies and Opportunities Programme (GG-Top). The aim of GG-ToP was to realise the first steps towards a portable quantum technology atom interferometer gravity gradiometer, robust and portable enough for field measurements. My role within the programme was to address the gravity inference problem, with a focus on possible civil engineering applications. My main contribution to the programme was development of the reversible

jump-Markov chain Monte Carlo Bayesian inference algorithm, which was demonstrated to be practically applicable using gravity measurements obtained using the Scintrex CG-5.

1.3 Thesis Structure

Initially we describe the gravity field, the generation of synthetic data and gravity measurement and corrections in Chapter 2. Then in Chapter 3 we explain the theoretical aspects of the Bayesian inference approach as applied to the gravity inference problem, along with the theory of the reversible-jump Markov chain Monte Carlo algorithm. Chapter 4 explains the practical implementation of the algorithm. Chapters 5 and 6 demonstrate the application of the algorithm to synthetic data and real world data respectively. We conclude the thesis with a discussion of the results shown and outline possible algorithm improvements and future work.

Chapter 2

Gravity: Generation, Measurement and Reduction

In this section we explain the theoretical aspects of the physical fields with which we are dealing with in this research. We also briefly outline the measurement of gravity, describing the workings of the Scintrex CG-5 and a future quantum technology atom interferometer gradiometer. We explain why such an instrument will be able to cancel environmental noise and consequently allow for faster, more precise terrestrial gravity gradient surveys. We also outline the analytical gravity models of basic shapes: the sphere, finite horizontal cylinder and cuboid. These simple shapes will be summed to construct more complicated forward models for inference from gravity and gravity gradient data in later chapters.

2.1 The Gravity Field

For two masses, m_1 and m_2 separated by a distance $\boldsymbol{r}_{12}$, the force incident on object two due to object one is given by *Newton's Law*,

$$\mathbf{F}_{12} = -\frac{Gm_1m_2}{|\boldsymbol{r}_{12}|^2}\hat{\boldsymbol{r}}_{12} \tag{2.1}$$

where $\hat{\boldsymbol{r}}_{12}$ is a unit vector (defined as $\frac{\boldsymbol{r}_2-\boldsymbol{r}_1}{|\boldsymbol{r}_2-\boldsymbol{r}_1|}$) and G is the universal gravitational constant ($G = 6.674 \times 10^{-11}$ m^{3}kg^{-1}s^{-2}). The acceleration of m_2 due to m_1 is then given by *Newton's Second Law*, dividing Equation 2.1 by m_2,

$$\mathbf{g} = -\frac{Gm_1}{|\boldsymbol{r}_{12}|^2}\hat{\boldsymbol{r}}_{12} \tag{2.2}$$

The gravitational field is conservative, meaning that the energy required to move a mass from one point to another is independent of the path taken between the two points. Any conservative field can be given as the gradient of some scalar field[34], in this case the gravitational potential U,

$$\mathbf{g} = \nabla U \tag{2.3}$$

where ∇ is the *gradient operator* or '*del*' and is given by $\nabla = \hat{\boldsymbol{i}}\frac{\partial}{\partial x} + \hat{\boldsymbol{j}}\frac{\partial}{\partial y} + \hat{\boldsymbol{k}}\frac{\partial}{\partial z}$.
The gravitational potential from an excess density distribution, ρ, in a volume, V, is given by[35],

$$U(\boldsymbol{r}) = -G\int_V \frac{\rho(\boldsymbol{r'})}{|\boldsymbol{r}-\boldsymbol{r'}|}dV \tag{2.4}$$

where $\mathbf{r}$ and $\mathbf{r}'$ represent the observation and integration points respectively. The gravity gradient tensor (also known as the *Eötvös* tensor) contains second derivatives of the gravitational potential and is defined as[36],

$$\nabla \otimes \mathbf{g} = \nabla \otimes (\nabla U) = \begin{pmatrix} U_{xx} & U_{xy} & U_{xz} \\ U_{yx} & U_{yy} & U_{yz} \\ U_{zx} & U_{zy} & U_{zz} \end{pmatrix} = \begin{pmatrix} \frac{\partial^2 U}{\partial x^2} & \frac{\partial^2 U}{\partial x \partial y} & \frac{\partial^2 U}{\partial x \partial z} \\ \frac{\partial^2 U}{\partial y \partial x} & \frac{\partial^2 U}{\partial y^2} & \frac{\partial^2 U}{\partial y \partial z} \\ \frac{\partial^2 U}{\partial z \partial x} & \frac{\partial^2 U}{\partial z \partial y} & \frac{\partial^2 U}{\partial z^2} \end{pmatrix} \tag{2.5}$$

where the symbol $\otimes$ represents the vector outer product. Outside of the source the gravity potential satisfies the Laplace equation,

$$\nabla.\nabla U(\boldsymbol{r}) = \nabla^2 U(\boldsymbol{r}) = 0 \tag{2.6}$$

The gravity tensor is of rank two with five independent elements, due to the symmetry about the leading diagonal and the fact that it is trace free[37]. Figure 2.1 shows the physical significance of the gravity gradient tensor components in Cartesian coordinates. In this research we

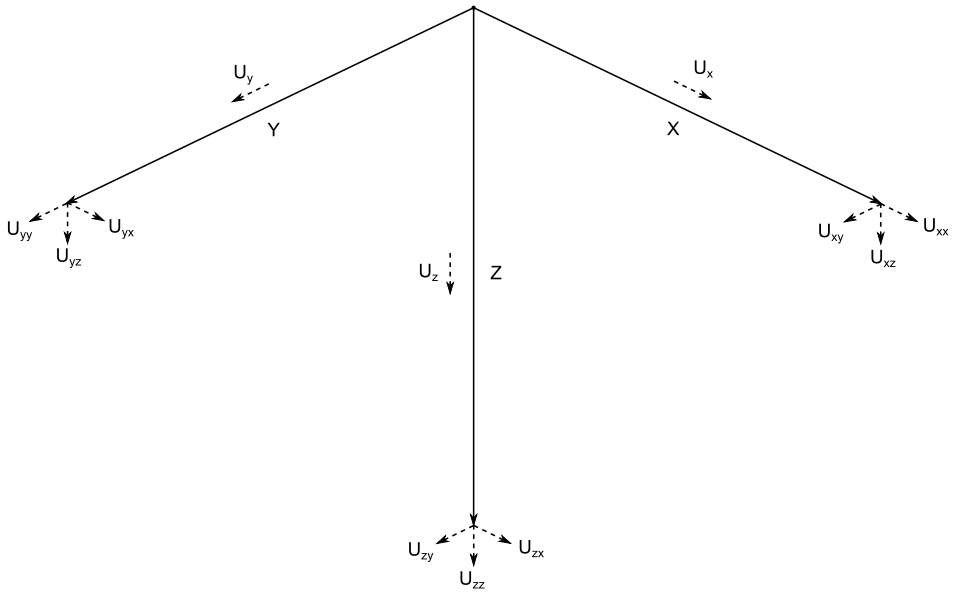


Figure 2.1: Illustration of the physical significance of the gravity tensor components in a Cartesian coordinate system.

are concerned only with the gravity (U_z) and gravity gradient (U_{zz}) field components as they are the easiest to accurately measure with current and future terrestrial field instruments (see Section 2.3). Investigation into the information content of other components is ongoing, and may be an area of future research as terrestrial instruments progress[38][39]. The SI units for gravity are ms^{-2}, much of the gravity community use μGal (10^{-8} ms^{-2}) named after Galileo Galilee. Another unit in use is the 'gravity unit' where 1 g.u = 10^{-6} ms^{-2}. Sometimes the 'nano-g' unit is also used where 1 ng = 10^{-9} g $\simeq 10^{-}8$ ms^{-2}. The gravity gradient SI unit is s^{-2}, the Eötvös (10^{-9} s^{-2}) is commonly used, named after Loránd Eötvös. In this research we use the respective SI units unless stated otherwise.

2.2 Synthetic Gravity Data

In this section we state the analytical expressions used to calculate the gravity and gravity gradient values of various simple shapes. Many of the target assets of interest in this research

such as pipelines are approximately cylindrical in shape, as such we are interested in modelling finite cylinders. We are also interested in underground structures such as mine-shafts and tunnels, these anomalies can be approximated by a cuboid model. We start with the simplest of the gravity forward models, that of a uniform density sphere. Models of arbitrary 3D gravity shapes have been developed and applied to problems such as inference of salt dome extent[40].

2.2.1 Sphere Forward Model

The gravitational attraction of a sphere of uniform density is the most fundamental gravity relation. It can be used as an approximation to any irregular body at a distance larger than the diameter of the body[41]. For a sphere centred at (x_0, y_0, z_0) the gravitational acceleration at measurement point (x, y, z) is[42],

$$\begin{aligned} U_z(x,y,z) &= \frac{4}{3} G\pi \frac{\Delta\rho R^3 z'}{r^3} \\ r &= \sqrt{x'^2 + y'^2 + z'^2}\,. \end{aligned} \tag{2.7}$$

where $x' = x - x_0$, R is the sphere radius and $\Delta\rho$ is the density difference between the sphere and the surrounding material. This simple relation provides an insight into the non-uniqueness of the gravity field. The density volume product defines the anomaly and neither can be determined individually.

Figure 2.2 shows the 5 independent *Eötvös* tensor components, along with the U_x, U_y and U_z components of a buried spherical void. We can see that the spatial position of the sphere is defined more precisely by the U_{zz} than by the U_z data. The numerical calculation of, for example, the U_{zz} component is achieved by differencing symbolically calculated acceleration at heights z_1 and z_2 (where $z_1 < z_2$) as given by,

$$\begin{aligned} U_{zz} &= \frac{U_{z_1} - U_{z_2}}{\Delta l} \\ \Delta l &= |z_2 - z_1| \end{aligned} \tag{2.8}$$

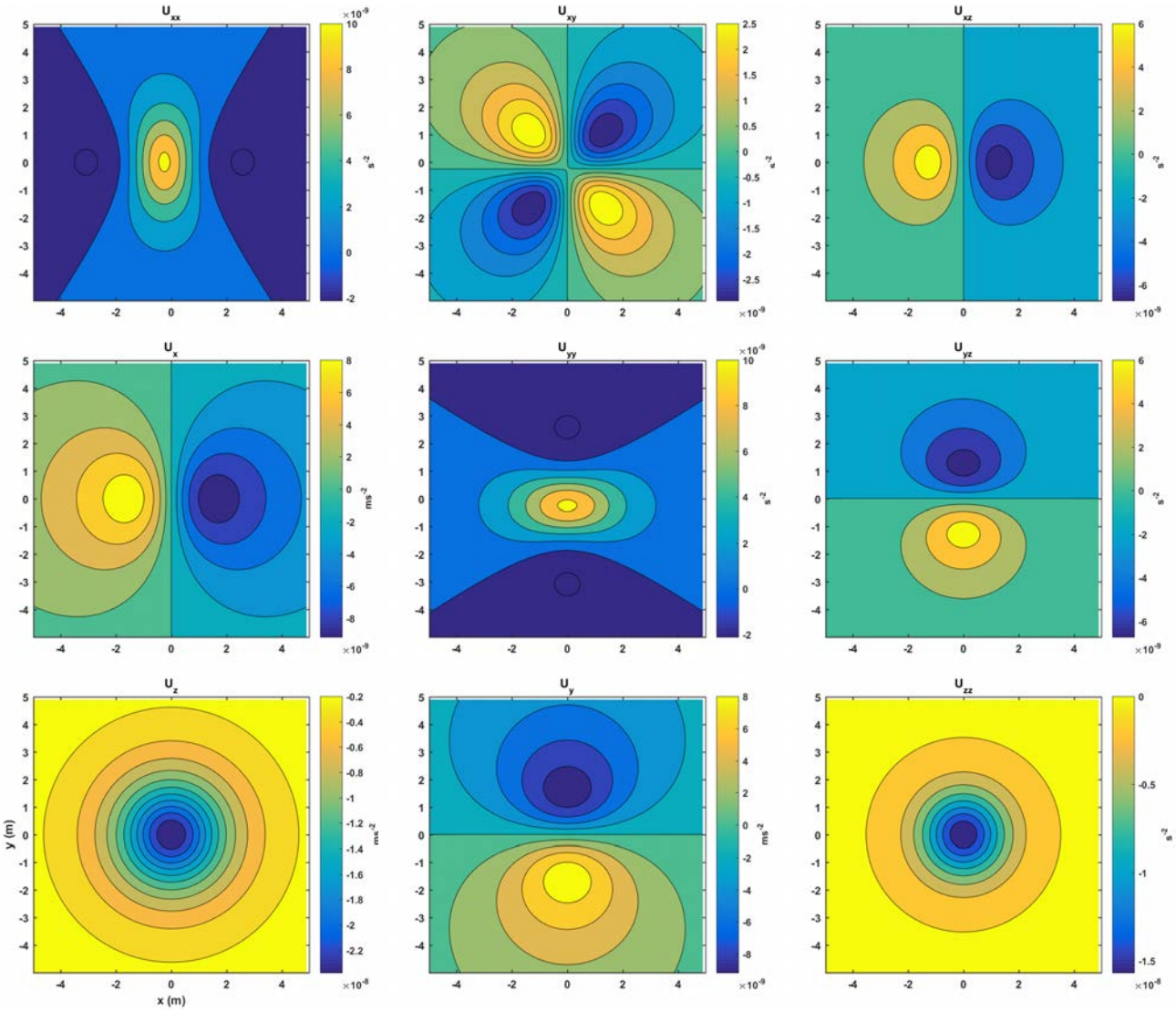


Figure 2.2: Gravity and gravity gradient signal produced by a buried spherical void. With density -2000 kgm^{-3}, depth 2 m and radius 0.5 m. (Bottom left, U_z, centre left U_x, bottom centre U_y, bottom right U_{zz})

The other numerical tensor components are calculated in the same way using the appropriate values, an example being,

$$U_{xz} = \frac{U_{x_a} - U_{x_2}}{\Delta l}$$
$$\Delta l = |z_2 - z_1| \tag{2.9}$$

explained in words, this is the U_x gravity acceleration as measured in two locations offset in the z direction.

Although the analytical expressions for the *Eötvös* tensor are easily derived for the sphere (see [37] for expressions), it is more challenging for increasingly complex geometries. Using the relations above we can numerically compute the tensor components, requiring two calculations of the forward model, for each tensor component.

2.2.2 Finite Cylinder Forward Model

The finite uniform density cylinder gravity model provides two extra degrees of freedom compared to the sphere model. A rotation parameter ψ, defines anti-clockwise rotation about the vertical axis, rotations about the x and y axes are assumed to be equal to zero. The finite length of the cylinder, L, is defined parallel to the y-axis.

For a measurement position (x, y, z) and cylinder centroid (x_0, y_0, z_0), the coordinates are translated and rotated about the cylinder centroid,

$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} \cos(\psi) & -\sin(\psi) \\ \sin(\psi) & \cos(\psi) \end{bmatrix} \begin{bmatrix} x - x_0 \\ y - y_0 \end{bmatrix} \tag{2.10}$$

then we can compute the vertical gravity acceleration, U_z, using[43],

$$U_z = G\pi \frac{\Delta\rho R^2 z'}{(x'^2 + z'^2)} \left[\frac{y' + L}{(x'^2 + (y' + L)^2 + z'^2)^{\frac{1}{2}}} - \frac{y' - L}{(x'^2 + (y' - L)^2 + z'^2)^{\frac{1}{2}}} \right] \tag{2.11}$$

with $z' = z - z_0$. Although not used in this research, of interest may be the infinite cylinder of uniform density, as given by[43],

$$U_z = 2G\pi \frac{\Delta\rho R^2 z'}{(x'^2 + z'^2)} \tag{2.12}$$

2.2.3 Cuboid Forward Model

The cuboid forward model adds a further degree of freedom to that of the finite cylinder, allowing the volume to vary independently in three spatial dimensions, as a result Equation

2.13[44] is somewhat more complex,

$$U_z = G\,\Delta\rho \left[x' \ln\left(y' + r\right) + y' \ln\left(x' + r\right) - z' \arctan \frac{x'\,y'}{z'\,r} \right] \Bigg|_{x_1}^{x_2} \Bigg|_{y_1}^{y_2} \Bigg|_{z_1}^{z_2} \qquad (2.13)$$
$$r = \sqrt{x'^2 + y'^2 + z'^2}\,.$$

where $x_1 = x_0 - \frac{l_x}{2}$, $x_2 = x_0 + \frac{l_x}{2}$ and x', y', z' are defined by,

$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} \cos(\psi) & -\sin(\psi) \\ \sin(\psi) & \cos(\psi) \end{bmatrix} \begin{bmatrix} x - x_0 \\ y - y_0 \end{bmatrix} + \begin{bmatrix} x_0 \\ y_0 \end{bmatrix} \qquad (2.14)$$

evaluated at each of the eight vertices (where x', y' and z' are offset when cycling through the vertex positions $(x_1, y_1, z_1), (x_2, y_1, z_1)$, etc.), the equation yields a total of 24 separate terms. Due to the evaluation of 12 log and 24 arc-tan functions, the calculation speed of the cuboid model is much slower when compared to the sphere or cylinder models[45]. The MATLAB function used to calculate the cuboid model can be seen in Appendix C. Figure 2.3 defines the cuboid vertices in the measurement coordinate system.

2.3 Measuring Gravity

The measurement of gravity is a very diverse field with many applications. In this research we are concerned specifically with terrestrial gravity measurements for small scale area surveys. However, gravity surveys are readily undertaken on land, underground, in the air, at sea and from satellites. With such a diversity of possible applications comes a wide variety of different instruments often specialised to a certain type of survey.

Measurements of gravity can be placed into two categories, relative and absolute. An absolute gravity measurement is made with direct observations of displacement or time. Relative gravity measurements give gravity differences between locations.

Throughout this research we have been working with two instruments in mind. The first being the Scintrex CG-5 (see Section 2.3.1), which is a relative gravimeter designed for ter-

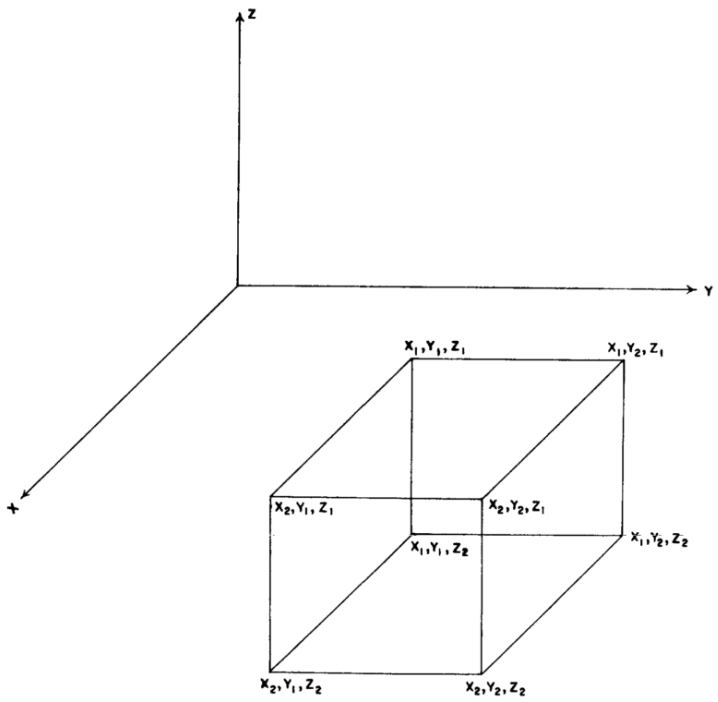


Figure 2.3: Cuboid model defined relative to the measurement coordinate axes $x - y - z$. Originally from [44].

restrial surveys. The other, which is yet to be realised as a completely field robust instrument (although progress has been made towards this goal[46][47]), is the quantum technology atom interferometer gradiometer. This is an absolute instrument promising increased accuracy and speed over current sensors[48] (see Section 2.3.2).

A terrestrial gravity survey is concerned with the measurement of U_z, the vertical derivative of the gravity potential. The average value of gravity at the Earth's surface is 9.81 ms^{-2}[49]. For near-surface anomalies we are looking for deviations from this value around one part in 10^9, requiring highly sophisticated instrumentation and rigorous survey techniques[50]. Surveys requiring this level of accuracy are often referred to as microgravity surveys. In order to reach this level of accuracy, information regarding the unwanted gravity effects at a given measurement point must be corrected for. The gravity data need to be 'reduced' to the signal of the anomalous density alone, commonly known as the Bouger anomaly (see Section 2.4)[49].

Microgravity surveys searching for relatively ($5 - 20$ m depth) near-surface voids have been conducted[51] with some success. Due to suppression of long wavelength (deep, regional)

anomalies, the measurement of the gravity gradient field U_{zz} is also of interest[52] for near surface anomalies, and has been used for many years[53].

2.3.1 Scintrex CG-5

The Scintrex CG-5 relative gravimeter is the current United Kingdom industry standard instrument for terrestrial gravity measurements. Figure 2.4 shows a picture of the CG-5 in the field, along with a schematic diagram of the general operating principle. A test mass

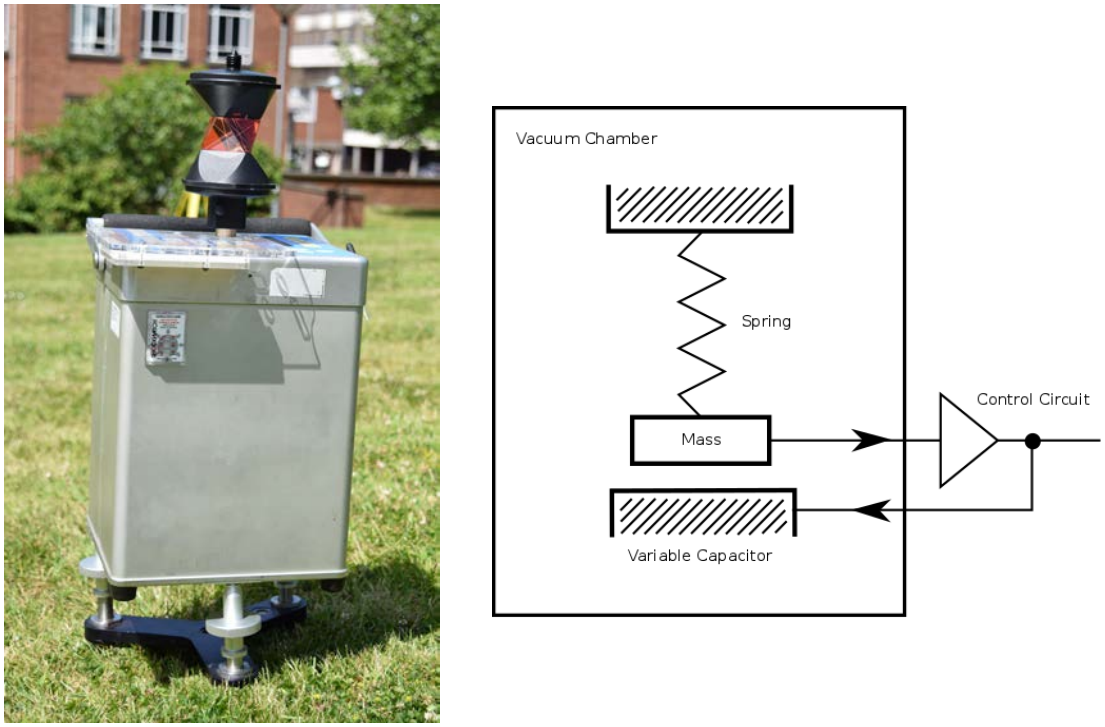


Figure 2.4: Left: Scintrex CG-5 with Leica 360 degree prism positioned directly above the internal sensor head. Right: Schematic of CG-5 operation. Using a zero length spring, correcting the position of a suspended mass to a null position using a variable capacitor control circuit.

is suspended from a zero length quartz spring[54]. The mass is continuously returned to a null position by an electronic control circuit, this restoring force is equivalent to the gravity acceleration after accounting for corrections due to temperature fluctuations and instrument tilt. The zero length spring has the property that if the test mass were to be removed, the force would be zero and the spring would have zero extension. This property is possible by

manufacturing the spring to spiral inwards, so that it would have zero length with no added mass[37]. The choice of quartz spring is preferable to metal because of its excellent elastic properties, low thermal response, and is approximately non-magnetic. However, the spring does relax over time, causing the instrument to drift. The drift is rated at $< 20 \times 10^{-8}$ ms^{-2} per day when static, increasing above this value when the instrument is moved. The Scintrex CG-5 has a resolution of 1×10^{-8} ms^{-2} and a survey precision of $< 5 \times 10^{-8}$ ms^{-2}[55]. It can automatically correct for temperature, tilt, tide and instrument drift. It measures at a rate of 6 Hz, with a single measurement of up to 256 seconds possible (longer measurements would be affected by the instrument drift).

Figure 2.5 shows a typical sixty second raw data measurement output by the CG-5. This particular measurement was taken as part of a gravity survey used later in this research (see Chapter 6). The low frequency, high amplitude variations are due to microseisms caused by

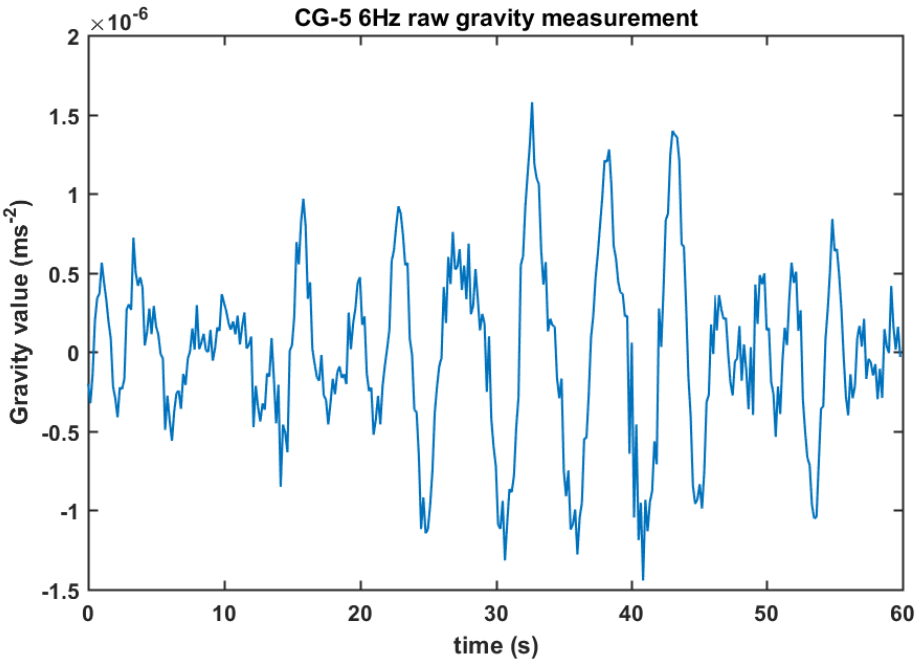


Figure 2.5: A typical raw gravity reading from the CG-5 taken in Birmingham United Kingdom. The low frequency ($0.1 - 0.3$ Hz) high amplitude variations due to the signal from ocean waves is visible.

ocean waves hitting the coast[56]. The amplitude of these microseisms is often the limiting factor for precise gravity surveys in the UK, requiring long measurement times to average out to a required uncertainty level. The high frequency noise is due to wind and other vibrational

effects.

2.3.2 Quantum Technology Atom Interferometer

The use of atom interferometry (AI) to measure gravity was first demonstrated by Chu[57] *et al* in 1991. An AI essentially exchanges the roles of light and matter from the more conventional laser interferometer. Laser cooled atoms (typically to micro-Kelvin level)[58] traverse the interferometer arms and light pulses are effectively used as the interferometer mirrors and beam splitters. Exploiting the wave-nature of atoms in this way allows for a resolution far higher than that achievable with laser interferometers as the de Broglie wavelength of atoms is small in comparison to wavelengths of light typically used in laser interferometers.

The $\frac{\pi}{2} - \pi - \frac{\pi}{2}$ interferometer set-up, analogous to the Mach-Zehnder laser interferometer is shown in figure 2.6[59]. Stimulated Raman transitions between hyperfine ground states

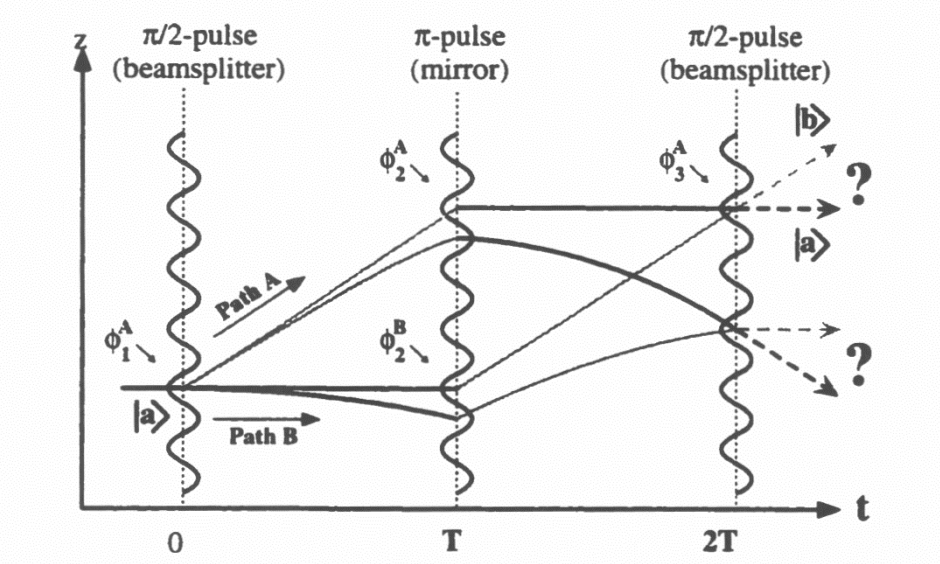


Figure 2.6: Originally from [59]. The $\frac{\pi}{2} - \pi - \frac{\pi}{2}$ interferometer set-up The straight paths represent the atom trajectories if gravity were not present (no path length difference), the curved lines represent the path difference caused by gravity.

are used to excite the atoms. The first $\frac{\pi}{2}$ transition is analogous to a beam splitter, atoms are driven into a superposition of hyperfine states $\mid a\rangle$ and $\mid b\rangle$, they subsequently drift apart spatially. A time T later the π pulse toggles the states ($\mid a\rangle \rightarrow \mid b\rangle$ and $\mid b\rangle \rightarrow \mid a\rangle$)

meaning the superposition of states drift back together. A time T later the states overlap, but only interfere when the final $\frac{\pi}{2}$ pulse is applied. The fraction of atoms in one state is then detected, this results in an oscillatory function of the path difference, which is dependent on local gravity[60]. This phase shift is given by,

$$\Delta\phi = k_{\text{eff}} g T^2 \tag{2.15}$$

where k_{eff} is the effective wave number of the counter propagating Raman beams, T is the time between Raman pulses and g is the acceleration due to gravity. There is a squared dependence on the time between Raman pulses T, the longer the atom clouds can be left in free-fall the greater the accuracy of the gravity reading. To form a gradiometer, two of the interferometers described above are aligned and interrogated simultaneously. This allows for common mode noise rejection, making the device less sensitive to vibrational noise than mechanical devices. Practically this means that the vertical gravity gradient U_{zz} can be measured to a high precision ($< 4 \times 10^{-9}$ s$^{-2}/\sqrt{\text{Hz}}$[61]). Other Eötvös tensor components that require simultaneous interrogation in axes orthogonal to the vertical are measured with less accuracy, as the time T is necessarily significantly shorter due to the horizontal alignment of Raman beams (limited by the beam width).

2.4 Gravity Corrections

Gravity data can only be compared to synthetic models when the so called gravity anomaly is calculated. That is, the theoretical gravity value at a given point is computed and compared with the observed gravity measurement[62]. The difference between the observed gravity and theoretical gravity at multiple points in space gives the two or three dimensional gravity anomaly. As the only measurements presented in this research were taken using the Scintrex CG-5 gravimeter, we address the corrections with it in mind. The corrections are split into time varying (temporal) and spatially varying. The methodology of a small scale gravity survey is discussed in Chapter 6, where these corrections are implemented.

2.4.1 Temporal Corrections

The largest natural variation of gravity is due to the Earth tide, which can produce variations of up to 300×10^{-8} ms^{-2} per day. The tidal gravity variations are caused primarily by the gravitational attraction of the Moon (the Sun is also significant) and from the deformation of the ground as a result. The tidal signals are periodic and can be accurately (to $< 1 \times 10^{-8}$ ms^{-2} precision) predicted. The Longman tidal model[63] is used by the Scintrex for automatic tidal correction, it requires input values of instrument latitude, longitude and Universal Coordinate Time (UTC).

After removing the tidal signal, the instrument drift can be removed by taking multiple readings on a survey day at a chosen base station location (see Section 6.1.1). Figure 2.7 shows the base station readings with corresponding linear quadratic and cubic least squares fits for CG-5 raw analogue-to-digital converter data. We can see that the drift is approximately linear, as the goodness of fit does not increase substantially with higher order polynomials. Figure 2.8 shows the base station readings of the Scintrex CG-5 corrected data. There is non-linearity in this data, possibly due to an incorrect initial setting of the drift parameter or to unknown affects of the built in black-box seismic filter. For comparisons between the raw corrected data and CG-5 corrected data we remove a second order drift correction from the gravity values. By obtaining regular base station measurements, other temporal corrections can be removed as part of the drift correction, such as changes in atmospheric pressure[64]. However, the corrections due to atmospheric pressure changes are less than the survey precision of the Scintrex.

2.4.2 Spatial Corrections

The spatial corrections that need to be applied to gravity data are illustrated in Figure 2.9. These corrections can be summarised by[66],

$$g_{\text{corr}} = g_{\text{obs}} - g_{\text{N}} - g_{\text{FA}} + g_{\text{BC}} \tag{2.16}$$

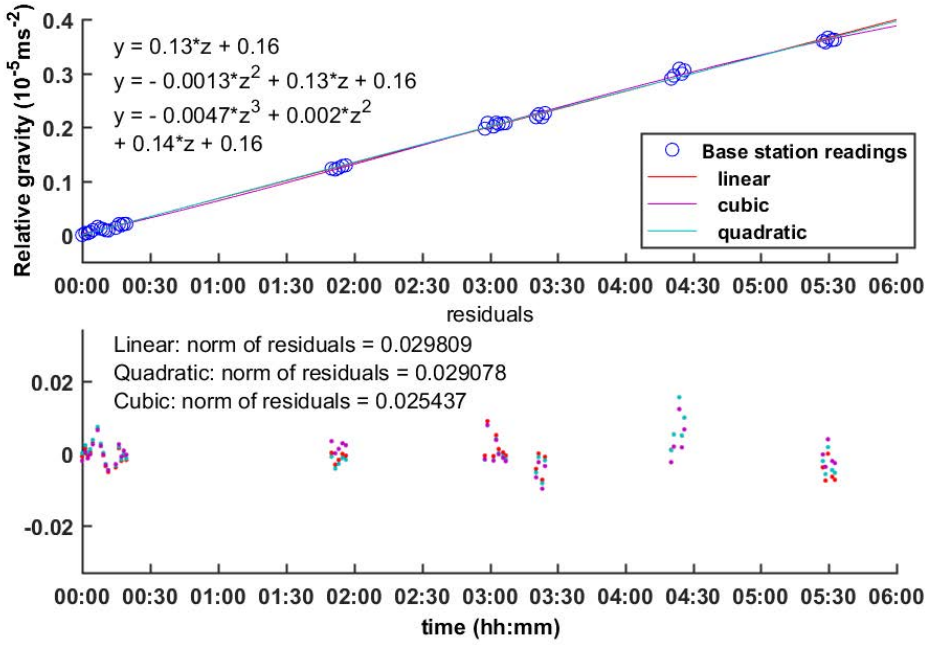


Figure 2.7: A collection of base station readings taken using the CG-5 during a survey day. Raw analogue to digital converter measurement data is shown.

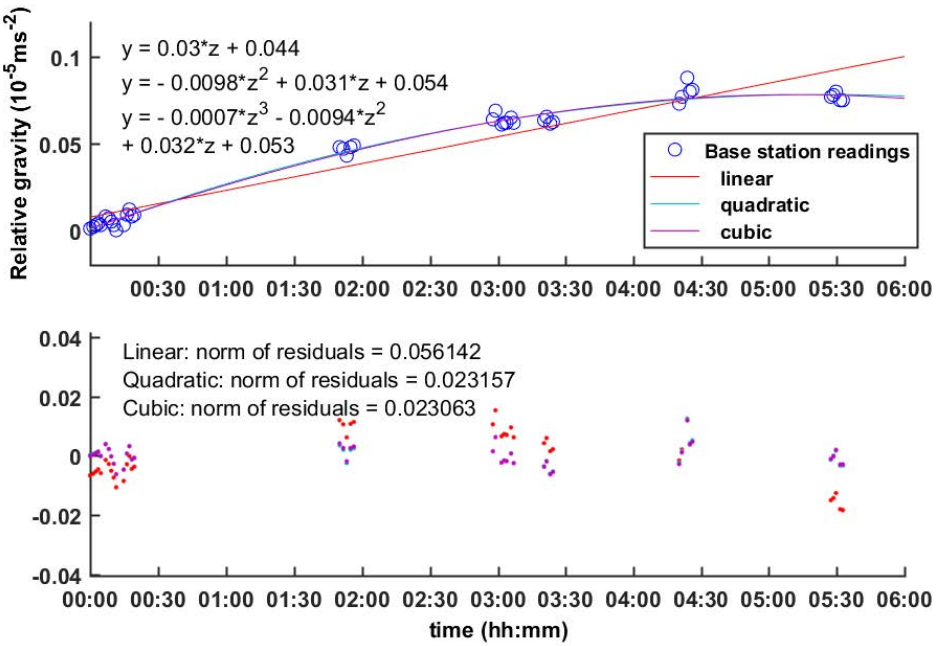


Figure 2.8: A collection of base station readings taken using the CG-5 during a survey day. Scintrex corrected measurement data is shown. Some non-linearity is introduced by the CG-5 corrections.

where we have; the observed gravity g_{obs}, the normal gravity g_{N}, the free air anomaly g_{FA} and the Bouger correction, g_{BC}. The spatial corrections are usually made with respect to the

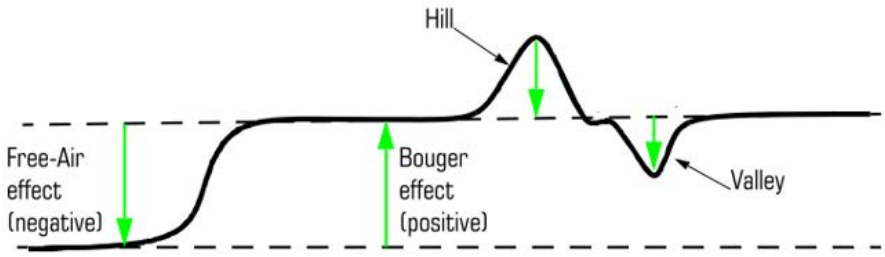


Figure 2.9: Illustration of the free-air and Bouger corrections. Adapted from [65].

geoid (the equipotential surface represented by average sea level[37]). However, for relative surveys not tied to absolute measurements, we can define our height difference relative to an arbitrary measurement point.

Latitude Variation

The Earth is not a perfect sphere, due to it's rotation it is deformed into an ellipsoid. The average gravity value at the surface varies with latitude ϕ, using the International Gravity Formula[65],

$$g_{\mathrm{N}} = g_{\mathrm{eq}}(1 + c_2 \sin^2 \phi - c_4 \sin^2 2\phi) \tag{2.17}$$

with $g_{\mathrm{eq}} = 9.780237$ ms^{-2} , $c_2 = 0.0053024$, $c_4 = 0.0000058$, giving g_{N} in units of mGal(10^{-5} ms^{-2}). These coefficients depend only on the Earth mass, ellipticity, rate of rotation and radius.

Free Air Correction

The strength of the Earth's gravity field decreases as measurement height, Δh, increases. The free air correction is the mean vertical gradient above the surface of the Earth. It is given by[67],

$$g_{\mathrm{FA}} = 0.3086 \times \Delta h \tag{2.18}$$

in units of mGal per metre. A height difference of one centimetre gives a free air correction value of 3×10^{-8} ms^{-2}. Meaning that the height of the Scintrex sensor needs to be known to within 0.01 m for measurement uncertainty $< 3 \times 10^{-8}$ ms^{-2}

Bouger Correction

When we change elevation on the Earth following the topography, there is a difference in mass between the relative measurement heights, Δh. This either produces a positive correction for extra mass (higher measurement point) or a negative correction for a mass deficit (lower measurement point). The simple infinite slab relation is used,

$$g_{\mathrm{BC}} = 2\pi G\rho\Delta h \tag{2.19}$$

The density, ρ used for the Bouger correction is usually the estimated near-surface ground density of a given survey area. Estimates can usually be obtained from nearby borehole log data, using the British Geological Survey web applications[68]. If these logs are unavailable, the Nettleton method can be used to determine the ground density using gravity readings at different heights on a slope, then minimising the correlation of the gravity measurements with the topography[69]. This method is dependent upon measurement uncertainty, and often requires extra data points to be taken away from the measurement area so as to avoid the effect of any anomalies present. A typical density value of near-surface soil (1800 kgm^{-3})[43] was used throughout this research.

Summary

In this chapter we have given an overview of the gravity and gravity gradient fields. The gravity models for the: sphere, cylinder and cuboid were stated for later use in the Bayesian inference algorithm (see Section 5). The workings of the Scintrex CG-5 relative gravimeter were discussed, along with the quantum atom interferometer gravity gradiometer. Finally, the corrections required to reduce gravity data to the Bouger anomaly were explained, which will be used later (Chapter 6) to correct survey data collected with the Scintrex CG-5. In the next chapter we overview the Bayesian approach to geophysical inference problems.

Chapter 3

Inference from Gravity Data

The Bayesian inference framework has seen increasing application to geophysical inference problems in the last few decades, as computing power has increased allowing for powerful search techniques such as Markov chain Monte Carlo to explore the Bayesian posterior distribution[70]. We skip over the philosophical nature of the disagreements between Frequentist and Bayesian schools of statistical thought (for those interested there was a lively debate in the literature of which [71][72][73][74] are just a sample). In some special cases the Bayesian approach produces solutions that are equivalent to Tikhonov regularisation, except that the Bayesian solution is always a probability distribution, whereas Tikhonov regularisation gives point estimates[75]. We begin the chapter with an overview of Bayes' theorem and progress through to the global optimisation technique used to make draws from the Bayesian posterior, the reversible-jump-Markov chain Monte Carlo algorithm. We keep the notation completely general, as the Bayesian framework can be applied to any problem of inference.

3.1 Bayesian Inference

Bayesian inference was first proposed by Reverend Thomas Bayes, in a paper read posthumously to the Royal Statistical Society in 1763[76]. The main distinction of the Bayesian paradigm from that of the classical or frequentist statistical approach is that all unknown quantities are treated as random variables (a variable used to represent the outcome of a

random experiment[77]), namely the unknown parameters of interest and the measurement data before observation. Bayesian inference explicitly incorporates all prior knowledge (data independent) of the model parameters and combines it with information contained in the measurement data to update our state of knowledge regarding our model parameters.
From the definition of conditional probabilities we have, for a vector of parameters $\boldsymbol{\theta}$ and a vector of data values $\boldsymbol{d}$ given by 1.1[78],

$$p(\boldsymbol{d}, \boldsymbol{\theta}) = p(\boldsymbol{d} \mid \boldsymbol{\theta})p(\boldsymbol{\theta}) = p(\boldsymbol{\theta} \mid \boldsymbol{d})p(\boldsymbol{d}) \tag{3.1}$$

The notation used here reads as follows, $p(\boldsymbol{d} \mid \boldsymbol{\theta})$ is the probability of $\boldsymbol{d}$ *given that* $\boldsymbol{\theta}$ is true. Rearranging gives the central equation of Bayesian inference, Bayes' Theorem,

$$p(\boldsymbol{\theta} \mid \boldsymbol{d}, k, \mathcal{I}) = \frac{p(\boldsymbol{d} \mid \boldsymbol{\theta}, k, \mathcal{I})p(\boldsymbol{\theta} \mid k, \mathcal{I})}{p(\boldsymbol{d} \mid k, \mathcal{I})} \tag{3.2}$$

where $p(\boldsymbol{\theta} \mid \boldsymbol{d}, k, \mathcal{I})$ is the posterior distribution, which represents the probability distribution of a vector of unknowns $\boldsymbol{\theta}$ given the data vector $\boldsymbol{d}$ the number of model objects k and the prior information $\mathcal{I}$; $p(\boldsymbol{d} \mid \boldsymbol{\theta}, k, \mathcal{I})$ is the likelihood function, a measure of the likelihood of observing data $\boldsymbol{d}$ given specific $\boldsymbol{\theta}$; $p(\boldsymbol{\theta} \mid k, \mathcal{I})$ the prior distribution, which contains all knowledge of the parameters before any data acquisition and $p(\boldsymbol{d} \mid k, \mathcal{I})$ the normalisation constant, which can be shown to be[79],

$$p(\boldsymbol{d} \mid k, \mathcal{I}) = \int p(\boldsymbol{d} \mid \boldsymbol{\theta}, k, \mathcal{I})p(\boldsymbol{\theta} \mid k, \mathcal{I})d\boldsymbol{\theta} \tag{3.3}$$

for the case of continuous $\boldsymbol{\theta}$. This constant or evidence term ensures that the posterior distribution integrates to one, as any valid probability distribution should. We can say that the posterior, up to a normalising constant is,

$$p(\boldsymbol{\theta} \mid \boldsymbol{d}, k, \mathcal{I}) \propto p(\boldsymbol{d} \mid \boldsymbol{\theta}, k, \mathcal{I})p(\boldsymbol{\theta} \mid k, \mathcal{I}) \tag{3.4}$$

The posterior probability distribution is the complete solution to the inference problem, as it contains all the available information about the parameter vector. However, for problems of high dimensionality (*i.e.* high number of parameters), accessing the information contained within the distribution can be difficult, as the normalizing integral can become mathematically intractable.

3.1.1 Likelihood Function

The likelihood function takes into account the information contained in the data points via a misfit function. The multivariate Gaussian distribution is usually chosen based upon the assumption of normally distributed measurements, uncorrelated with zero mean[80]. It is defined by,

$$\begin{aligned} p(\boldsymbol{d} \mid \boldsymbol{\theta}, k, \mathcal{I}) &= \frac{1}{\sqrt{(2\pi)^{N_d}|\boldsymbol{\Sigma}|}} \exp\left[-\frac{1}{2}\left(\boldsymbol{d} - f(\boldsymbol{\theta})\right)^T \boldsymbol{\Sigma}^{-1}\left(\boldsymbol{d} - f(\boldsymbol{\theta})\right)\right] \\ \boldsymbol{\Sigma} &= \mathbf{I}\left[\boldsymbol{\sigma}_{\mathbf{d}}^{\mathbf{2}} + \sigma_{\mathbf{m}}^{\mathbf{2}}\right] \end{aligned} \tag{3.5}$$

where $f(\boldsymbol{\theta})$ represents our forward model and N_d is the number of data points. The covariance matrix $\boldsymbol{\Sigma}$ contains the total uncertainty, which is defined as the sum of contributions from the data uncertainty $\boldsymbol{\sigma}_d$ and the theoretical model uncertainty σ_m. The data uncertainty is generally assumed known from experiment, with the model uncertainty an unknown parameter to be found. It represents the inevitable uncertainty associated with finite models that are necessarily an approximation to a more complex reality. The likelihood function has a relation to the method of least squares. For uniform prior distributions

3.1.2 Prior Probability Distribution

The prior probability distribution contains all information known about our model parameters prior to taking the data measurements. There is some controversy over the assignment of the prior distribution[81] as subjective choices can affect inference outputs, as we shall see

in practice in Chapter 6. As such, all information of prior distribution parameters should be made available and the inference outputs interpreted with respect to the priors used[82].

Often it is the case that we do not wish to bias our algorithm without good reason to do so. In situations where we seemingly have null prior information a uniform prior distribution can be used which only requires us to define minimum and maximum bounding values. These bounding values can be set to represent physical plausibility. An example is that of a cuboid side length, which must take a positive value (lower bound equal to zero) and reasonably be no larger than the measurement grid (upper bound equal to corresponding size of measurement grid).

Careful consideration should be given to each model parameter's prior probability distribution, in this research we use either the uniform distribution, Gaussian distribution or Gamma distribution.

For some parameter θ, the uniform prior distribution is defined,

$$p(\theta \mid k, \mathcal{I}, \theta_{\max}, \theta_{\min}) = \begin{cases} \frac{1}{\theta_{\max} - \theta_{\min}}, & \text{if } \theta_{\min} \leq \theta \leq \theta_{\max} \\ 0, & \text{otherwise} \end{cases} \tag{3.6}$$

The uniform distribution is an example of a 'proper prior', meaning that it integrates to unity. The uniform distribution gives zero outside the minimum and maximum limits, putting a hard limit on the size of the model space.

The Gaussian distribution can be used when we have more information about a given model parameter. For example if we know that the gravity anomaly density contrast is 2000 kgm^{-3} with a standard deviation uncertainty of $\pm$100 kgm^{-3} then we would use the appropriate Gaussian as our prior distribution with mean $\mu = 2000$ and standard deviation $\sigma = 100$. The univariate Gaussian is defined as,

$$p(\theta \mid k, \mathcal{I}, \mu, \sigma) = \frac{1}{\sigma\sqrt{2\pi}} \exp\left(\frac{-(\theta - \mu)^2}{2\sigma^2}\right) \tag{3.7}$$

The Gaussian distribution with a large standard deviation can be used for situations where we do not wish to limit the model space abruptly, as with the uniform distribution.
The Gamma distribution, is defined for positive θ as[83],

$$p(\theta \mid k, \mathcal{I}, a, b) = \frac{1}{b^a \Gamma(a)} \theta^{a-1} \exp(\frac{-\theta}{b}) \tag{3.8}$$

where a and b are the distribution shape and scale parameters respectively. The Gamma distribution can be used for parameters with a known constraint on the sign. Examples being parameters related to volume, such as radius and length. Figure 3.1 shows plots of the Gamma distribution with various shape and scale parameters. Gamma distributions with

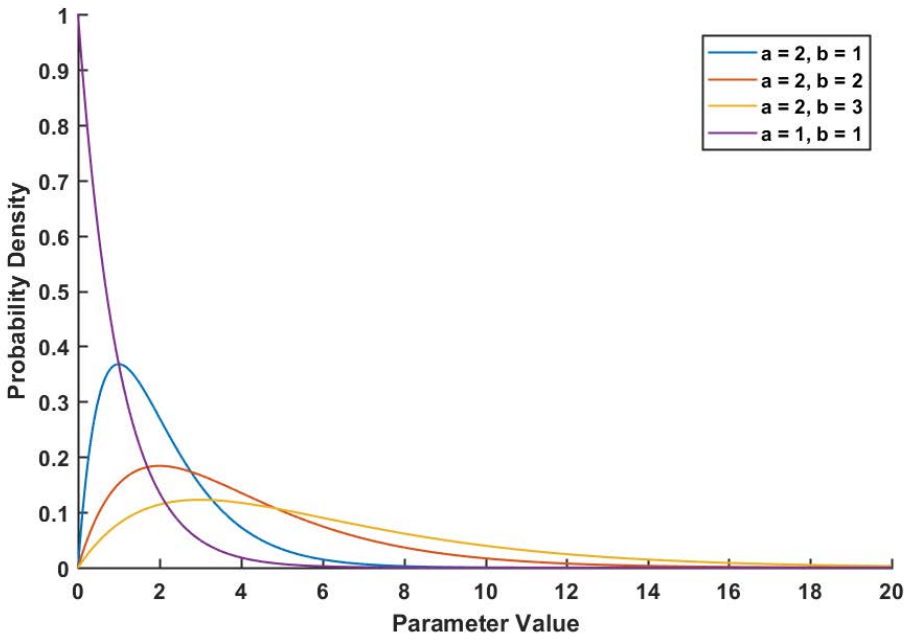


Figure 3.1: Gamma distributions for different shape (a) and scale (b) parameters. The distribution mode is located at $(a-1)b$.

shape $a = 2$ and various scale values are used in this research. The total prior distribution value is calculated by multiplying all of the individual parameter priors together,

$$p(\boldsymbol{\theta} \mid k, \mathcal{I}, \boldsymbol{h}) = \prod_{n=1}^{N_\theta} p(\theta_n \mid k, \mathcal{I}, \boldsymbol{h}) \tag{3.9}$$

The parameters that define the prior distributions are known as hyper-parameters These parameters are defined prior to any measurements taking place.

3.2 Markov chain Monte Carlo

Monte Carlo (probabilistic) methods were first used to study the problem of neutron diffusion in fissile material for the development of the atomic bomb during the Second World War. After successfully tackling this probabilistic problem, it was realised that Monte Carlo methods could be used to solve deterministic problems such as the evaluation of high-dimensional integrals.[70]

A Markov chain $X = [X_0, X_1, X_2, ...]$ is a trajectory where the next quantity (X_{t+1}) is determined probabilistically from the previous quantity (X_t) only, through some proposal distribution $q(X_{t+1} \mid X_t)$ (assuming that the proposal distribution $q(.)$ has no dependence on time, t), with no memory of how the previous quantity was arrived at[84]. This process can be used to sample a target, invariant distribution, which will be the equilibrium distribution of the Markov chain, to which the process will converge after many iterations[85]. In our case the equilibrium distribution will be the Bayesian posterior probability distribution the extent of which is defined by the model parameter prior distributions.

In order to ensure that a Markov chain will converge to the desired distribution, some conditions must be adhered to. A Markov chain needs to be *homogeneous*, *irreducible* and *aperiodic*[86]. A Markov chain is said to be homogeneous if the transition probabilities do not depend on the time step, that is they do not change with time. A Markov chain is irreducible when it is possible to access all states of the distribution from any other state. That is, there are no jumps that can be made to a state from which the chain cannot return. An aperiodic chain is when every state can recur at each time step[87].

Due to the intractability of the normalisation constant in problems of high dimensionality, the Bayesian posterior distribution often cannot be independently directly sampled. An exhaustive search of the model space may be initially appealing, but as the dimensionality

of the model space increases such sampling methods become too computationally intensive. The approach of importance sampling is used, where model parameters that adhere closely to the data and prior information are sampled more frequently[88].

Markov chain Monte Carlo methods have found widespread use in recent years in geophysics, often to evaluate high dimensional Bayesian posterior distributions (although there are non-Bayesian applications), an extensive overview of the history of Markov chain Monte Carlo methods in geophysics is given in by Sambridge *et al*[70]. The most readily used algorithm that ensures the equilibrium distribution will indeed sample the desired target distribution is the Metropolis-Hastings algorithm[89], described in the following section.

3.2.1 Metropolis-Hastings algorithm

The Metropolis algorithm was first proposed by Nicholas Metropolis et al. in 1953[90], it was used to calculate the properties of chemical substances. Hastings[91] generalized the algorithm for non-symmetrical proposal distributions.

The algorithm starts at a random position in the model space, $\boldsymbol{\theta}$. At each step a proposal is made to move from $\boldsymbol{\theta}$ to a new point in the model space $\boldsymbol{\theta}'$. This new model is chosen probabilistically from a defined proposal distribution $q(\boldsymbol{\theta}' \mid \boldsymbol{\theta})$ that depends only on the previous model. The new model is accepted with probability[92],

$$\alpha = \min\left[1, \frac{p(\boldsymbol{\theta}' \mid \boldsymbol{d}, k, \mathcal{I})}{p(\boldsymbol{\theta} \mid \boldsymbol{d}, k, \mathcal{I})} \cdot \frac{q(\boldsymbol{\theta} \mid \boldsymbol{\theta}')}{q(\boldsymbol{\theta}' \mid \boldsymbol{\theta})}\right] \tag{3.10}$$

where the fraction on the left represents the ratio of a proposed posterior point and the current posterior point. The fraction on the right represents the ratio of the proposal probabilities (which equals one if the proposal distribution is symmetric). If the new model is accepted, the next iteration will start with the model $\boldsymbol{\theta}'$, if the proposal is rejected the next iteration of the process will start from $\boldsymbol{\theta}$. A proposal that increases the posterior distribution will always be accepted, this ensures that the random walk is targeted to areas of high probability within the posterior distribution. However, if a proposal decreases the posterior

distribution probability, it is either accepted or rejected based on the generation of a uniform random number[93], $U[0,1]$, accepting the proposal if $u < \alpha$ and rejecting if $u > \alpha$. Being able to accept moves that reduce the probability goes some way to ensuring that the algorithm does not get stuck in local high probability regions. Due to the ratio of posterior distributions (see Equation 3.2), the evidence term does not require calculation as it cancels. The output of the algorithm is made up of the collection of models obtained at the end of each step, from these models inferences can be made about the individual parameters of interest. Figure 3.2 shows a representation of a Metropolis-Hastings update. The initial guess

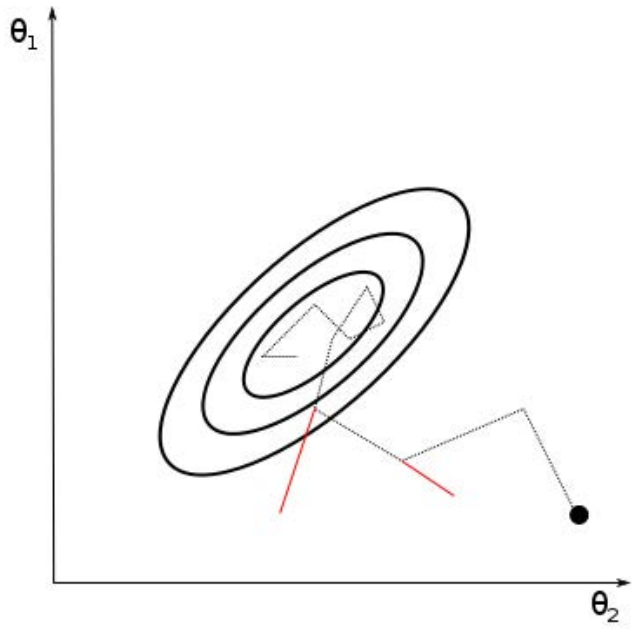


Figure 3.2: Schematic representation of the Metropolis-Hastings algorithm for two arbitrary parameters. The sampler starts far away from the high probability region, successive iterations progress towards the high probability region with rejected iterations shown in red.

is far from the high probability region, successive guesses move towards the high probability region. From here the chain will explore the high probability region proportional to the stationary distribution probability. For finite length chains, this initial exploration of the low probability region is often removed from the final models as a *burn-in* period (see Section 4.1.5)

3.2.2 Reversible-Jump Markov chain Monte Carlo

There are a class of problems where the number of unknowns is something we do not know, sometimes known as 'non-parametric' Bayesian inference. The final generalisation of the Metropolis Hastings algorithm was made by Green [94] in his analysis of change point detection in the rate of coal mining disasters, where he was interested in finding the amount of change points and their locations.

The reversible-jump Markov chain Monte Carlo algorithm extends the use of the Metropolis-Hastings rule to cases where the proposal distribution is not only able to move to a point within the current model, but between models too. The power of the reversible-jump Markov chain Monte Carlo algorithm (sometimes called the Metropolis-Hastings-Green algorithm) is that we are no longer required to assign the length of the model parameter vector as prior information, the measurement data effectively determines the model complexity required to best fit the data within the total uncertainty. Equation 3.11 shows the modification made by Green to the Metropolis-Hastings acceptance ratio,

$$\alpha = \min\left[1, \frac{p(\boldsymbol{\theta}' \mid \boldsymbol{d}, k', \mathcal{I})}{p(\boldsymbol{\theta} \mid \boldsymbol{d}, k, \mathcal{I})} \cdot \frac{q(\boldsymbol{\theta} \mid \boldsymbol{\theta}')}{q(\boldsymbol{\theta}' \mid \boldsymbol{\theta})} \cdot \mid \boldsymbol{J} \mid\right] \tag{3.11}$$

the only difference compared to Equation 3.10 is the addition of the 'Jacobian' term, defined as,

$$\mid \boldsymbol{J} \mid = \frac{\partial(\boldsymbol{\theta}', \boldsymbol{u}^{k'})}{\partial(\boldsymbol{\theta}, \boldsymbol{u}^{k})} \tag{3.12}$$

the role of which is to account for scale changes in the transformation between models $\boldsymbol{\theta}$ and $\boldsymbol{\theta}'$. Using vectors of random numbers $\boldsymbol{u}^{k'}$ and $\boldsymbol{u}^{k}$ of length r and r' respectively, to match the dimension of the two models being compared[95] such that $r + k = r' + k'$. The Jacobian term for a regular Metropolis-Hastings update is simply equal to one.

Green notes that for nested models where the only dimension change necessary is the addition of deletion of model parameters, the reversible jump formalism is not required. This is due to the deletion being essentially equivalent to setting the parameters to zero[96]. We use such models in this research, where gravity anomalies consisting of multiple model objects

are simply summed in the forward model.

This leads to a special case of the reversible-jump Markov chain Monte Carlo algorithm that provides the ability to explore different models without the added complication of calculating the Jacobian[93]. Such processes are called 'Birth-Death' Markov chain Monte Carlo[97]. The birth-death Markov chain Monte Carlo scheme works by suggesting only moves that are reversible when transitioning between models.

For a birth move, a number of parameters (multiple parameters constitute one model object) are added to the model, their starting values drawn from their respective prior probability distributions. With the condition that the number of model objects is less than some maximum specified as prior information, $k < k_{\max}$.

For the death move an object is chosen at random and deleted from the model, with the condition that $k > 1$.

The birth move increases the complexity of the model, the death move decreases the complexity. It is intuitive to see why this process is deemed to be a natural Ockham's razor, as the more parameters there are to explain a model the lower the prior probability (as given by Equation 3.9) becomes.

The simple implementation of the birth-death Markov chain Monte Carlo algorithm has seen it used many times in the geophysical literature, first by Malinverno [28] [11]. It is usually applied to problems where the number of layers of a given Earth model are unknown, and a layer can be added or deleted via a birth-death proposal[98].

A common criticism of the reversible-jump method is that we could simply obtain similar results by varying the number of model objects and completing multiple runs with stationary k, essentially many separate standard Markov chain Monte Carlo runs. This may seem appealing, however, for runs with a high number of model objects, the computational time will become prohibitive. Further to this, simply defining multiple model objects does not mean that they will contribute to the total model in any meaningful way, an object that is superfluous to the fitting of the data will be marginalised, making assessment of convergence difficult (see Section 4.1.4). It is more efficient to simply delete the objects that are not

required to fit the given dataset, as is done in the reversible-jump Markov chain Monte Carlo algorithm.

Summary

In this section we have given an overview of Bayesian inference and it's application to geophysical inference problems. The work horse of Bayesian inference for complex models, Markov chain Monte Carlo, was discussed. The importance sampling method of Metropolis-Hastings which guarantees convergence to the Bayesian posterior distribution was explained, along with the extension into determination of the model complexity using a special case of the reversible-jump Markov chain Monte Carlo algorithm, the birth-death Markov chain Monte Carlo. In the following chapter we discuss the implementation of these ideas into an algorithm to tackle the gravity inference problem.

Chapter 4

The Reversible-jump Markov chain Monte Carlo Algorithm

An overview of Bayesian theory and Monte Carlo methods was described in Sections 3.1 and 3.2 respectively. In this chapter we detail the algorithm produced to make draws from the Bayesian posterior probability distribution for the gravity inference problem, where the number of unknowns, or model complexity, is itself unknown. The basic outline of the algorithm is described in Section 4.1. Explanations of adaptive Markov chain Monte Carlo and simulated annealing, which are employed to address the practical problem of chain mixing are discussed in later sections. Methods to identify convergence failure are described in Section 4.1.4.

4.1 Algorithm implementation

The reversible-jump Markov chain Monte Carlo algorithm developed in this research was coded in MATLAB (see appendix C for the main code body). We put the theory described in the previous chapter into practice for the gravity inference problem. The flow chart shown in Figure 4.1 describes the basic algorithm outline. The algorithm presented is due to an accumulation of experience with running reversible-jump Markov chain Monte Carlo for the detection of near surface gravity anomalies over the course of the research. It represents the

best method found for ensuring reasonable convergence for the three forward model types. The user decides on initial values for the following parameters,

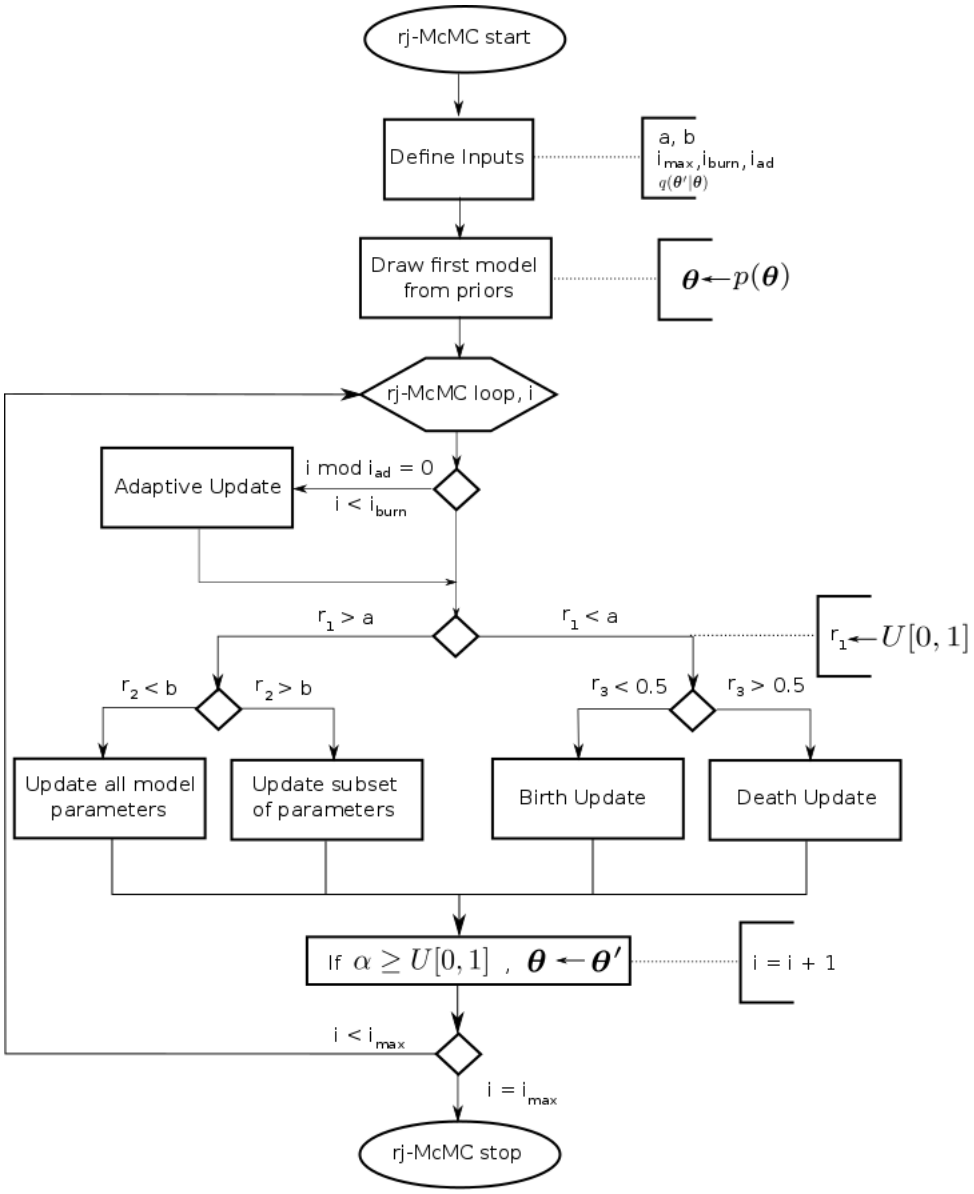


Figure 4.1: Flow chart of basic code structure. There are four possible update schemes. The initial inputs are defined by the user.

- i_{max}: The maximum number of iterations for the reversible-jump Markov chain Monte Carlo run.
- i_{burn}: The number of iterations defined as the burn-in period.
- i_{ad}: When i mod $i_{\text{ad}} = 0$, the proposal distributions are updated (see section 4.1.2)

- a, b: Determine the relative frequency of different updates.

In this research, a and b are both set to 0.5. Meaning that we have equal probability for all four updates to take place at each iteration. This is arbitrary and can be altered to improve chain mixing. The burn-in period is set as some percentage of the total number of iterations. We set $i_{\text{burn}} = 50,000$ here, or $0.25 \times i_{\text{max}}$. The adaptive proposal update is carried out every i_{ad} iterations up to the end of the defined burn-in period. Section 4.1.2 describes the details of this update.

Now we consider the four separate update schemes and the reasoning behind their inclusion, from left to right as shown in Figure 4.1.

The all parameter update is self explanatory, we take our current parameter vector and propose new values from the respective parameter proposal distributions.

When updating a subset of the model parameters we randomly select a pre-defined parameter block from a random model object (which is dependent upon the forward model being used). If we currently have k model objects, we generate a uniform random number $U[1, k]$ to select the object which we wish to update. Then we randomly select the parameter block of this model object which we will update. Here we used two parameter blocks, one for parameters that are generally well defined by a gravity anomaly; the x-y centroid and rotation (if applicable). The second block consisting of volume, depth and density parameters, all of which are in general highly correlated. The number of parameter blocks and their relative update frequencies are at the discretion of the user. When we go towards the extreme case of only updating a single parameter at each iteration, it is easy to see how many more iterations will be required for each parameter to converge. In this research we update the two parameter blocks with equal frequency. However, it may be useful to update the highly correlated parameters more often than those expected to be well defined by the dataset[87]. It is noted that all other parameters are held constant during a block update.

These two updates are standard Metropolis-Hastings schemes where the number of model objects stays the same. We have two separate updates for constant k so that the forward model objects have the flexibility to be updated one at a time and all together. Updating

all objects together is advantageous as it allows all of the model parameters to be updated at once. However, if we only allowed this update we would have no way of knowing which objects are having a large affect on the acceptance percentages and hence mixing of the sampler. To address this issue, a block update for a single object is used which allows tracking of the acceptance percentage of parameters within a particular object, and using the adaptive Markov chain Monte Carlo scheme outlined in Section 4.1.2 we can improve the mixing (see Section 4.1.4) of individual objects.

The next two updates are reversible-jump updates, where the number of model objects k changes. For a birth update, we add a new object to our parameter vector, whilst holding all other parameters constant, so long as $k < k_{\max}$. The initial parameters of our new object are drawn at random from the prior probability distribution.

For a death update, we randomly select a model object, $U[1, k]$ and delete it from the parameter vector, ensuring that $k > 1$.

After an update has been completed, the acceptance ratio is computed (from Equation 3.10),

$$\log(\alpha) = \frac{\log(p(\boldsymbol{\theta}' \mid \boldsymbol{d}, k', \mathcal{I})) - \log(p(\boldsymbol{\theta} \mid \boldsymbol{d}, k, \mathcal{I}))}{1 + t} \tag{4.1}$$

and accepted if $\log(\alpha) > \log(U[0, 1])$. Where t is the simulated annealing temperature (see Section 4.1.3). Logarithms are used to ensure the values stay within a computable range. If an iteration is accepted, the parameter values are updated accordingly, if rejected the parameters stay the same. The parameter values from all accepted iterations are saved together in a parameter matrix. The values saved after the burn-in are regarded as converged parameters, unless failure of convergence is found using the diagnostics discussed in Section 4.1.4.

4.1.1 Proposal Distributions

The choice of proposal distributions has a significant impact on the quality of Markov chain Monte Carlo mixing and hence convergence. Usually a normal distribution centred on the

current model value is used to make a new model proposal. Figure 4.2 shows a comparison between a normal distribution and the Cauchy distribution. The Cauchy distribution has

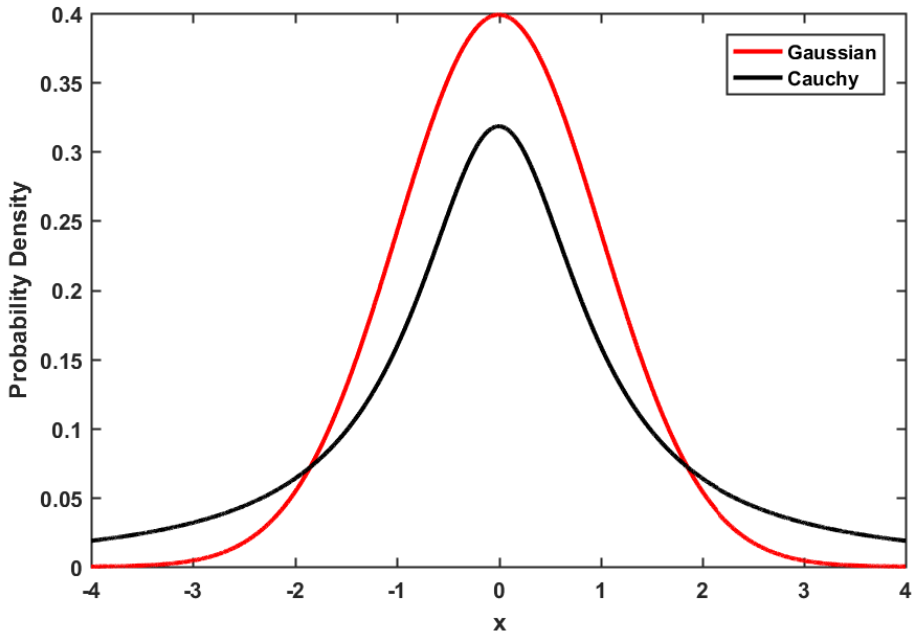


Figure 4.2: Comparison of the Normal distribution and Cauchy distribution with $\gamma = 1$ and $x_0 = 0$.

higher chance of proposing values further away from the mean value than the normal distribution. In this research, it was deemed advantageous to have an increased chance of large steps in the model space for the spatial and geometric parameters of a model object, allowing for large jumps between model parameter values. As such, the Cauchy distribution was used as the proposal distribution for all parameters, it is defined by,

$$f(x|x_0, \gamma) = \frac{1}{\pi\gamma\left[1 + \left(\frac{x-x_0}{\gamma}\right)^2\right]} \tag{4.2}$$

where γ is the half-width half maximum and x_0 is the peak centroid. The Cauchy distribution is equivalent to a student-t distribution with $\nu = 1$. The initial values of the proposal distributions width for unknown parameters need to be defined by the user, values used here are defined depending on the particular model being used. As a rule, parameters that are expected to mix and converge well are given a smaller proposal scaling value than for

parameters which we expect to be highly correlated and to mix poorly and converge slowly.

4.1.2 Adaptive Markov chain Monte Carlo

Usually, the proposal distribution for each parameter is set at the start of a Markov chain Monte Carlo run and left unchanged throughout. A system of trial and error is used to choose the proposal distribution widths which seem to provide the best mixing of the chain (see Section 4.1.4). This trial and error can be difficult, and the rigidity of this approach can mean that the acceptance percentages of the updates are far from the optimum of 0.23 for high dimensional distributions[99].

Adaptive Markov chain Monte Carlo[100] updates the proposal distribution widths based on past information from the Markov chain. As a result the technique is non-Markovian.

By keeping track of the sum of the total attempted iterations and the accepted iterations we can easily calculate the acceptance percentage. For a given number of iterations, i_{ad}, if the acceptance percentage is lower than some minimum bound ad_{min}, the proposal distribution standard deviation values are scaled by some factor ad_{dec} where $\text{ad}_{\text{dec}} < 1$. Smaller jumps in the parameter space will result in a larger acceptance percent. If the acceptance is larger than ad_{max} the values are scaled by ad_{inc} where $\text{ad}_{\text{inc}} > 1$. The assignment of these values is somewhat arbitrary, although it is recommended that the scaling parameters are symmetrical. In this research we used; $i_{\text{ad}} = 1000$, $\text{ad}_{\text{dec}} = 0.9$, $\text{ad}_{\text{inc}} = 1.1$, $\text{ad}_{\text{max}} = 0.4$ and $\text{ad}_{\text{min}} = 0.1$. So, at every thousandth iteration the respective acceptance percentages are calculated. If a given acceptance percent is less than 10% the proposal width is decreased by 10%. If the acceptance percent is greater than 40% the proposal widths are increased by 10%.

As described in Section 4.1, we have one update which alters all of our parameters and one which updates only a subset or block of the parameter vector. We use both of these updates to alter different aspects of the proposal distributions.

After assessing the acceptance percentage of the all parameter update, a scale factor ad_{all} is adjusted as explained above. This value should be initialised at a value less than one. As we

are updating many parameters, the overall proposal scaling needs to be smaller to ensure a reasonable acceptance percentage is obtained. The block update allows us to alter the proposal distributions for these parameters only. This gives flexibility as different model objects and parameter blocks will require different proposal distributions to mix well. However, more iterations are required when sampling only a sub-set of model parameters at each iteration. As we have proposal distributions that are specific to our individual model objects, we need to ensure that the proposal distributions are created and deleted accordingly with the birth and death updates. For a death move we deleted the corresponding proposal distribution values. For a birth move we assign the same proposal distributions as for the $k = 1$ object (which can never be deleted).

The adaptive Markov chain Monte Carlo scheme does nothing to alter the rate at which birth-death updates are accepted, as there is no specific proposal distribution associated with these updates. It is not clear that the reversible-jump Markov chain Monte Carlo birth-death moves should follow the same acceptance percent rules as for standard fixed dimension MH moves. As is the case with many reversible jump algorithms, the dimension jumps create a large perturbation of the likelihood function and as a result generally have a very low acceptance percentage ($<1\%$) [101]. As the adaptive process is non-Markovian, we only allow updates of the proposal distribution during the pre-defined burn-in period (see Section 4.1.4). After this period the proposal distribution widths are fixed at their current values.

4.1.3 Simulated Annealing

Simulated annealing increases the chance of updates being accepted early in a run, by altering the Metropolis-Hastings acceptance criterion[102]. Iterative cooling returns the acceptance probability to the standard Metropolis Hastings acceptance as the temperature approaches zero as given by Equation 4.1. A simple exponential cooling schedule was used, for current

iteration i the temperature is given by,

$$t_i = t_{i-1} C^i \tag{4.3}$$

Figure 4.3 shows different schedules for various constant values. In this work we use a short

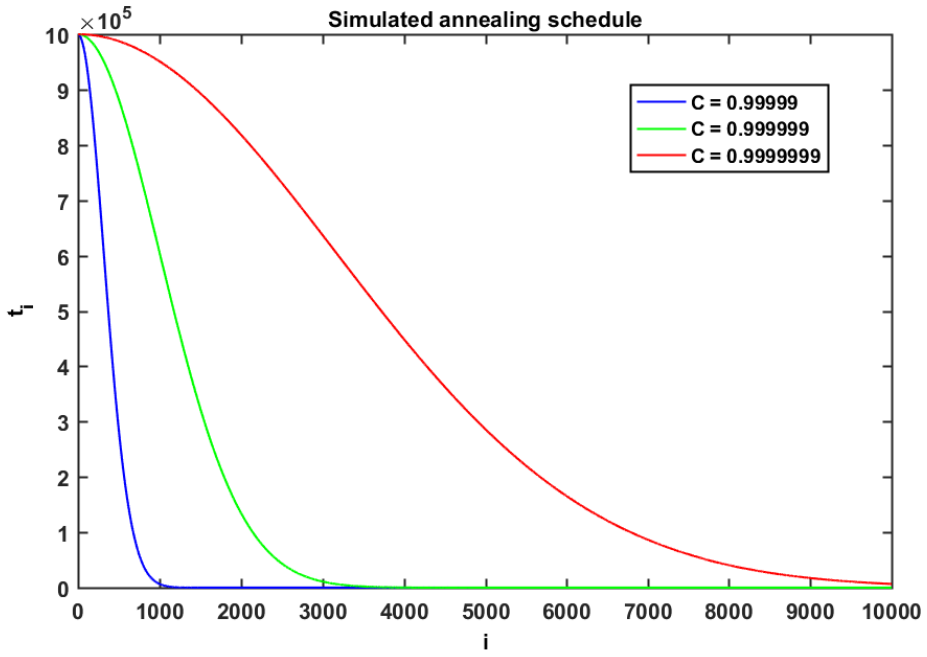


Figure 4.3: Different simulated annealing schedules for various C values.

simulated annealing schedule with $C = 0.99999$ and $t_0 = 1000000$ so that after around a thousand or so iterations we return to the correct Metropolis-Hastings acceptance. This initial period allows exploration of different k values that otherwise might not be explored due to the low acceptance rate of birth-death updates.

4.1.4 Assessing Convergence

When implementing any Markov chain Monte Carlo algorithm, it is crucial to consider the questions of convergence and whether the Bayesian posterior distribution has been fully explored. These are active research questions in the statistics community, and as such there is no generic rule to decide when a particular run has sufficiently explored the posterior distribution or has converged, indeed the answers to these questions may be much harder

when considering the reversible-jump Markov chain Monte Carlo algorithm as we can no longer directly compare parameters between individual runs due to the model switching problem[103][104].

A well implemented Markov chain Monte Carlo algorithm should show what is known as good mixing. Mixing is an indicator that draws from the Bayesian posterior distribution are independent. Figure 4.4 shows a comparison of good and poor parameter mixing. Good parameter mixing leads to faster algorithm convergence. As the process starts at a random

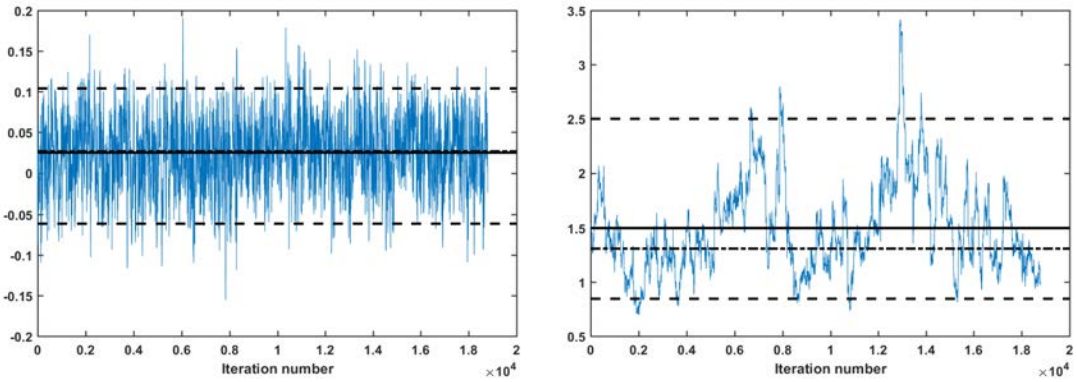


Figure 4.4: An example of good mixing (left) and poor parameter mixing (right) for some arbitrary model parameter.

point in the model space there is a 'burn-in' period due to the very likely scenario of initializing the algorithm in a low probability region of the model space. To account for this, early samples are rejected until a criterion (the standard deviation of the data error vector $\boldsymbol{e} = \boldsymbol{d} - \boldsymbol{\theta}$ becomes smaller than the expected standard deviation of the measurement errors) is met[28] or the process is simply run for so many iterations that the effect of the burn in period is negligible. Alternatively, the method used in this research is to initialise a run with a set burn-in period. The determination of the length of the burn-in is based on trial and error, running the algorithm with a given measurement set and forward model multiple times.

Although simulated annealing and adaptive Markov chain Monte Carlo, described above, can help chain mixing and hence speed up convergence, they cannot guarantee convergence. For this reason we are required to run many realisations of the reversible-jump Markov chain Monte Carlo process from multiple random starting points in the model space. Once the burn in period has been removed, the problem of assessing convergence needs to be addressed[105].

There are many possible diagnostics that can be used. However, most cannot be used in this research as they rely on parameters maintaining their meaning across models and chains[103]. The chosen diagnostics of Geweke (see below) and manual autocorrelation inspection are used as they can be applied to any marginal posterior distribution.

Geweke Diagnostic

The Geweke diagnostic is defined by Equation 4.4[106][85].

$$z_G = \frac{\mu_a - \mu_b}{\sqrt{\sigma_a^2 + \sigma_b^2}} \rightarrow \mathcal{N}(0,1) \tag{4.4}$$

A parameter's marginal posterior is split into two windows. The first window encompassing the first 10% of the iterations, the second spanning the final 50% of iterations. If the chain is stationary the values of the mean and variance at the start and end should be similar. As the number of iterations approaches infinity, the Geweke diagnostic approaches a normal distribution if the chain has converged. So any values obtained that stray far from the normal distribution are warnings of failed convergence. The diagnostic can only give warnings of potentially failed convergence, it cannot prove convergence. Figure 4.5 outlines how the Geweke diagnostic is calculated for an arbitrary model parameter output chain. The Geweke diagnostic gave a value of 4.47 for this chain. We set a threshold of 3 throughout this research, above this value chains are automatically rejected. Depending on the importance of the dataset this value can be altered as required. The Geweke diagnostic can also be used to determine when the burn-in period has ended, if we imagine removing progressively more of the chain from the left hand side of the figure, we would eventually obtain a Geweke diagnostic within our accepted threshold.

Auto-correlation

Autocorrelation is a measure of how much the value of a parameter posterior probability distribution correlates to other values of itself at different points in time. The autocorrelation

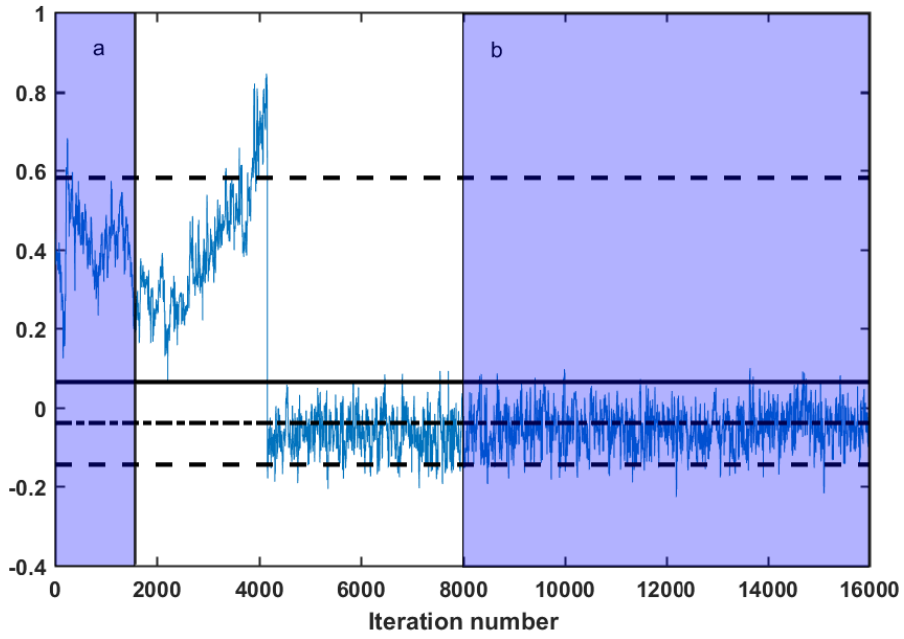


Figure 4.5: Geweke diagnostic for an arbitrary parameter output chain.

of a parameter marginal posterior distribution can provide information on slow mixing and potentially slow convergence. A low autocorrelation is desirable as it is a good indicator that the draws from the posterior probability distribution are independent. For the trace displayed in Figure 4.5 the autocorrelation plot is calculated using the MATLAB function *xcorr*. Arbitrary bounds of ± 0.2 were added to aid the visual inspection, correlation values outside of these bounds usually indicates poor mixing and slow model parameter convergence. The autocorrelation is not used to automatically reject reversible-jump Markov chain Monte Carlo runs in this research, as some of the highly correlated parameters of the non-linear gravity forward model are expected to mix slowly. Instead it is used simply as a visual aid to understand the status of mixing and convergence of different model parameters.

4.1.5 Algorithm Outputs

Figure 4.7 shows a plot of a parameter trace, not corrected for the burn-in period. The red outline shows the initial simulated annealing period where the parameter undergoes large changes due to the altered Metropolis-Hastings acceptance ratio. The orange section shows the extent of some pre-defined burn-in period where adaptive Markov chain Monte Carlo is

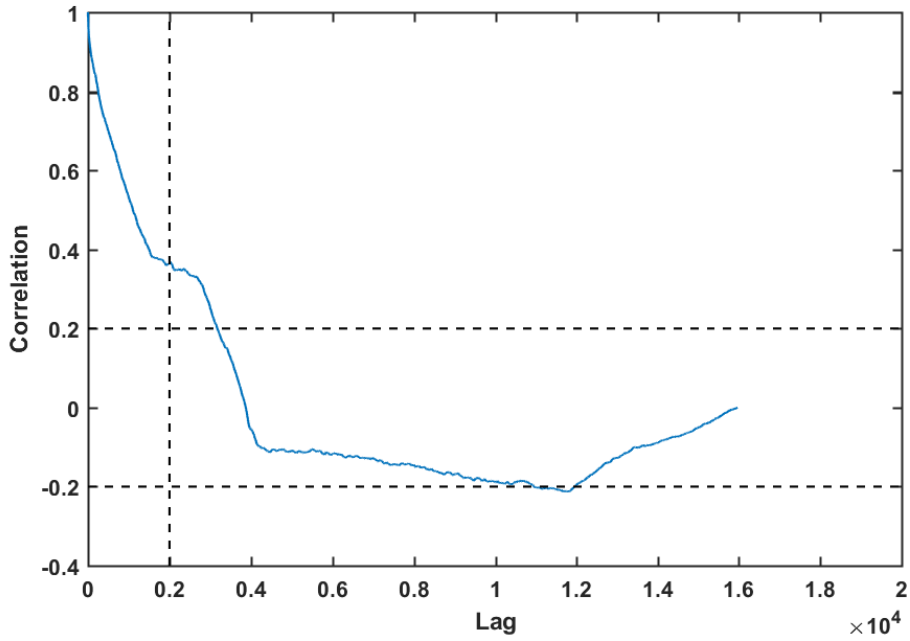


Figure 4.6: The autocorrelation plot for the trace in Figure 4.5. The initial chain values skewed the autocorrelation. The vertical dashed line represents the end of the burn-in period.

permitted. The green section is taken as the 'converged' output distribution for the model parameter. We can plot this converged distribution as a histogram with appropriate summary

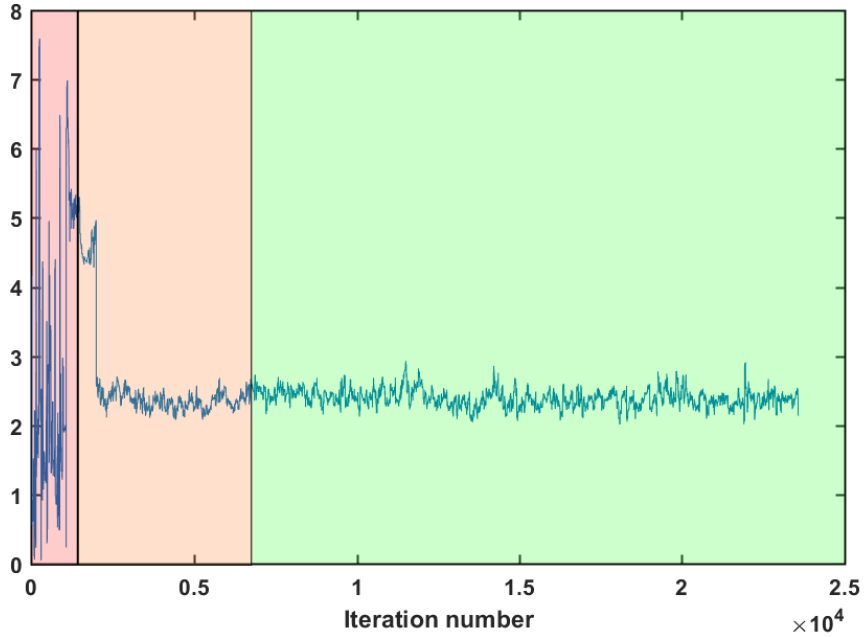


Figure 4.7: Trace plot showing the stages of a an arbitrary model parameter output chain. Red area is simulated annealing, orange is burn-in and green is accepted iterations.

statistics of: mean, mode and 95% credibility intervals (there is a 95% probability that the

parameter value lies between these bounds) as shown in Figure 4.8. Figure 4.9 shows the

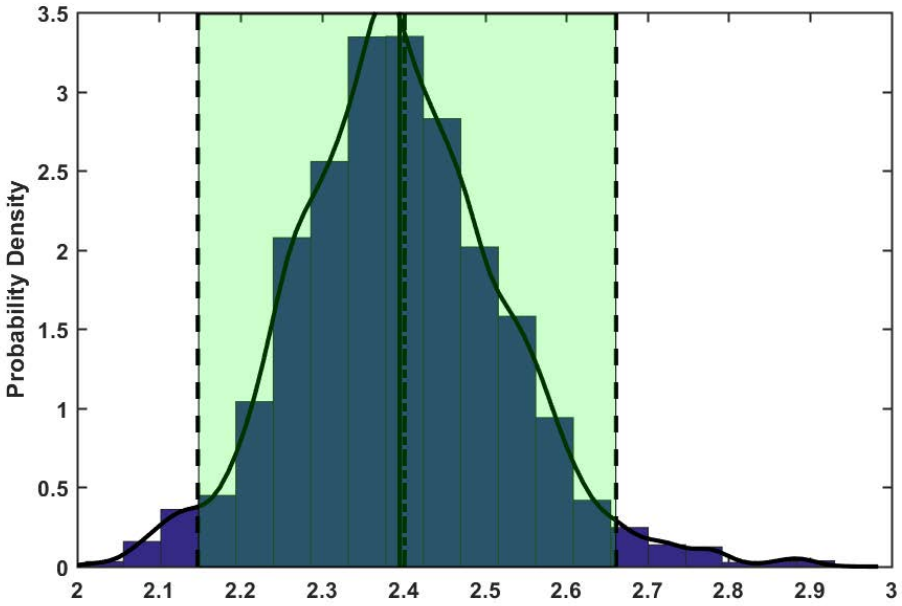


Figure 4.8: A histogram of the accepted region of Figure 4.7 trace. The 95% credibility interval is shown by vertical dashed lines, the mean by a solid line and the mode by a dot-dash line.

evolution of the number of model objects. The initial period allows for an exploration of the model space. The parameter quickly converges to a stationary value, where new objects are rarely accepted and when they are accepted, they are swiftly deleted.

Summary

In this chapter we outlined the coding implementation of the reversible-jump Markov chain Monte Carlo algorithm. The four different types of updates were discussed: all parameters, model object parameter blocks, birth and death. The key methods used to ensure good parameter mixing were described. Simulated annealing was shown to allow an early exploration of low probability areas of the model space before returning the acceptance ratio to the usual Metropolis-Hastings rule. The adaptive Markov chain Monte Carlo scheme was outlined, showing how the proposal distributions for the all parameter and model object parameter block updates are changed by retaining some limited memory of the chain (updating the ac-

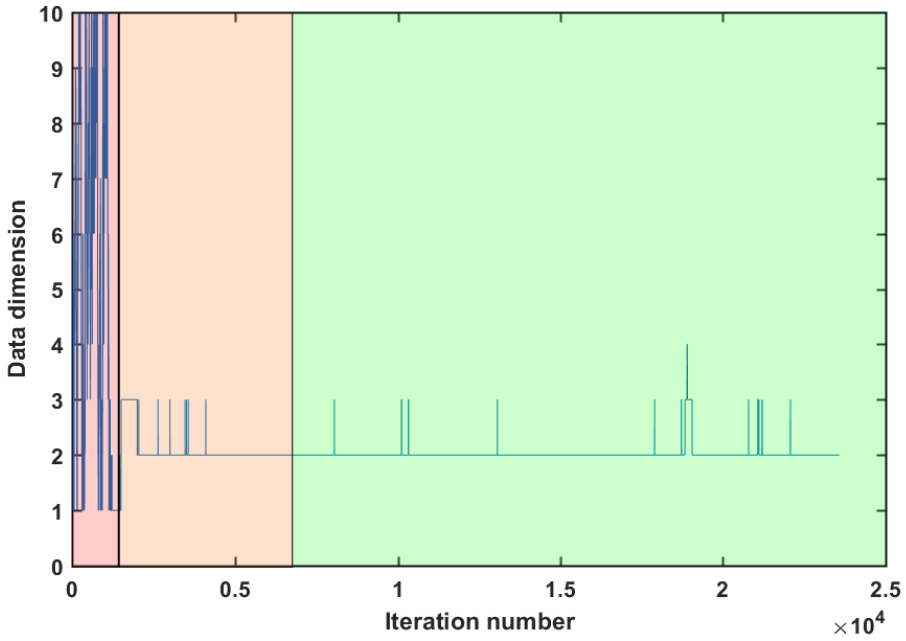


Figure 4.9: The evolution of the number of model objects parameter k over one reversible-jump Markov chain Monte Carlo run.

ceptance percentages). The adaptive scheme was halted at a set burn-in period; after which the marginal posterior distributions were deemed to have converged unless shown otherwise by the Geweke diagnostic.

Chapter 5

Synthetic Data Examples

Here we apply the reversible-jump Markov chain Monte Carlo algorithm to a number of synthetic targets of interest contaminated with added Gaussian noise. The first is a large buried cuboid void, modelled from the Scintrex CG-5 survey outlined in Appendix A.2. We imagine a situation where we have prior information that we are looking for a cuboid, and thus use the cuboid forward model. We also imagine a situation where we are unsure of the shape of the anomaly and hence use the simplest geometry - that of a sphere. We briefly investigate the change in inference outputs due to a reduction in data point density.
The other model considered is that of a near-surface gas pipeline in Section 5.2. Again we consider the case where we know the true model (cylinder) and the case when we do not. This anomaly provides an interesting 'semi-infinite' scenario whereby the gravity signal extends past the measurement grid bounds. For both targets, we compare the inference for synthetic gravity measurement data with uncertainty comparable to that of the Scintrex CG-5, and the vertical gravity gradient as we would expect to collect with a quantum technology atom interferometer gradiometer.

5.1 The Buried Void Problem

An obvious target for which gravity measurements have direct application is that of a buried void. Here we describe the application of the reversible-jump Markov chain Monte Carlo

algorithm to simulated noise corrupted data of a near surface nuclear bunker of centroid depth -2.3 m and dimensions $5.5 \times 2.25 \times 2.25$ m. As mentioned previously, this is a synthetic representation of an anomaly that was measured using the Scintrex CG-5 but for which only one measurement line was obtained due to unforeseen circumstances and time constraints. The gravity measurement data collected is shown in Appendix A.2. Good agreement was found with the forward model generated using the known anomaly location and geometric properties, assuming an average density of limestone. We can be confident that the synthetic data generated here is at least partially supported by experimental measurement. A full gravity survey would contain further complications; the bunker had an entry shaft that protruded from the ground approximately half a metre, measurements at this location would not be available, producing a gap in the measurement grid.

The noise corrupted synthetic data is shown in Figure 5.1. The model that produced the data is shown in Figure 5.2. The measurement points are spaced by 0.5 m on a square 10×10 m grid, giving a total of 441 data points. The data points are taken in a flat plane with $z = 0.25$ m, which is the approximate distance between the Scintrex CG-5 test mass and ground level. It will be useful to define some approximate measure of the signal strength within the data grid, so that we can quickly compare the signal strength for different synthetic gravity anomalies. Although the signal-to-noise ratio is usually defined as the ratio of the signal power to the noise power, for the rest of this work we define the signal-to-noise ratio for a gravity anomaly as,

$$\mathrm{SNR} = \frac{\max(|\boldsymbol{d}|)}{\langle \sigma_d \rangle} \tag{5.1}$$

which is the maximum value of the measurement vector, divided by the average measurement uncertainty, $\langle \sigma_d \rangle$. This gives an intuitive comparison between the measurement uncertainty and the peak signal of the gravity anomaly, whether negative or positive. We are assuming that the maximum signal is relative to a zero background. For this simple void anomaly we obtain a value of 17.4 for synthetic gravity measurements with uncertainty comparable to the single gravity line obtained.

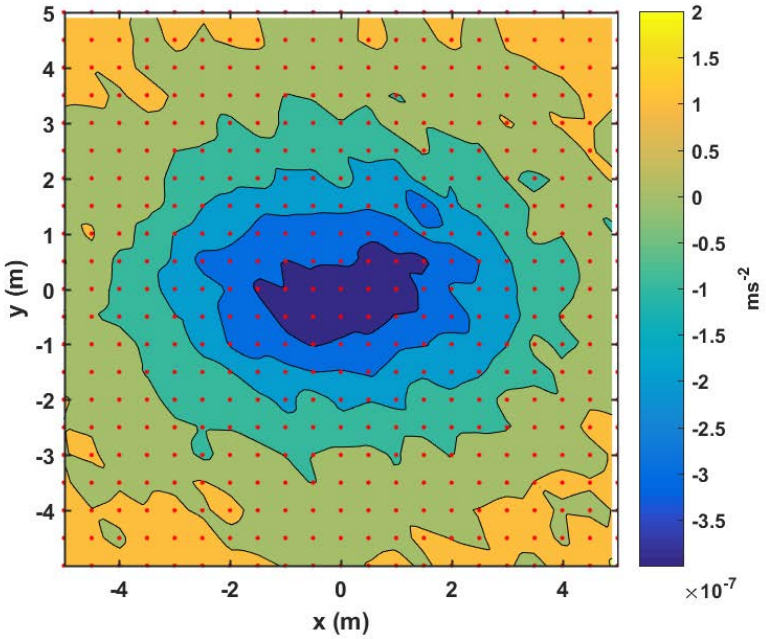


Figure 5.1: Synthetic data of buried near surface void. The data is offset by the mean value. Gaussian noise with $\sigma = 3 \times 10^{-8}$ ms^{-2} was added to the data. A measurement point spacing of 0.5 m was used (red dots). There are a total of 441 data points.

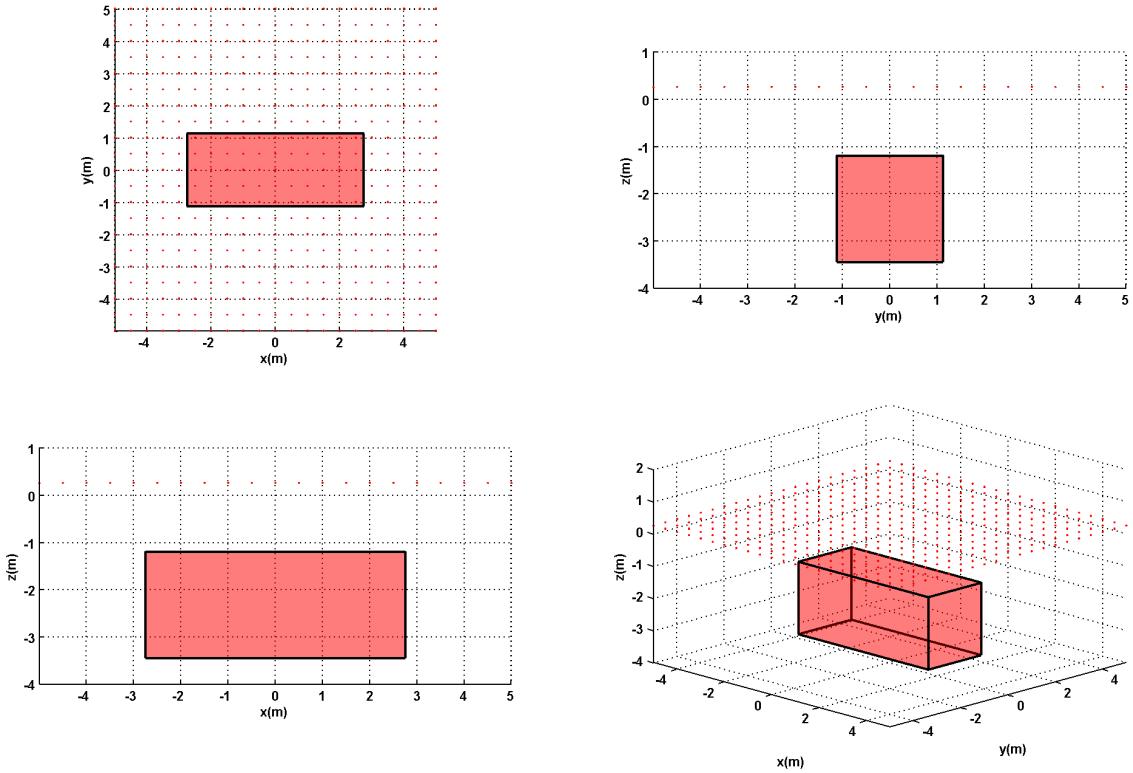


Figure 5.2: Nuclear bunker modelled as a single void. The model density, $\Delta\rho$, was set as -2700 kgm^{-3}. Red dots are the measurement point locations.

5.1.1 Cuboid Forward Model

Depending on the circumstances of a given gravity survey, the level and quality of prior information may vary quite substantially. Initially, let us assume that we know the geometry of the buried void to be that of a cuboid, or multiple cuboids. With such strong prior knowledge of our model, we are now engaged in a parameter estimation problem. We wish to find the marginal Bayesian posterior distributions of the geometrical and locational properties of any cuboids present in the data-set.

We define our forward model as a sum of contributions from individual cuboids (model objects), with the gravitational acceleration due to each being given by Equation 2.13. The forward model vector is then,

$$\boldsymbol{U_z}\left(\boldsymbol{\theta}\right) = \left(\sum_{k=1}^{k_{\max}} \boldsymbol{U}_{\boldsymbol{z}_k} - \left\langle \sum_{k=1}^{k_{\max}} \boldsymbol{U}_{\boldsymbol{z}_k} \right\rangle\right) + \eta \tag{5.2}$$

for a number of objects k, with gravity offset parameter η. Removing the mean from the forward model ensures that the offset parameter will be close to zero. Note that we have made a slight re-parametrisation to Equation 2.13. The z_0 parameter is redefined as $z_0 = z_0 - \frac{\ell_z}{2}$, meaning that the anomaly cannot exist above the $z = 0$ plane.

The likelihood function (Equation 3.5) is then given by,

$$\begin{aligned} p(\boldsymbol{d} \mid \boldsymbol{\theta}, k, \mathcal{I}) &= \frac{1}{\sqrt{(2\pi)^{N_d}|\boldsymbol{\Sigma}|}} \exp\left[-\frac{1}{2}\left(\boldsymbol{d} - \boldsymbol{U_z}\right)\right)^T \boldsymbol{\Sigma}^{-1}\left(\boldsymbol{d} - \boldsymbol{U_z}\right)\right) \\ \boldsymbol{\Sigma} &= \mathbf{I}\left[\boldsymbol{\sigma}_{\mathbf{d}}^{\mathbf{2}} + \sigma_{\mathbf{m}}^{\mathbf{2}}\right] \end{aligned} \tag{5.3}$$

where $\boldsymbol{d}$ is the measurement vector, $\boldsymbol{\sigma_d}$ is the measurement uncertainty vector and σ_m is the model uncertainty parameter. In-total the model consists of $8k + 2$ separate parameters, as the model dimension k is indirectly assessed by the reversible-jump algorithm (hence it does not require an explicit parameter). The prior probability distributions of the model parameters are shown graphically in Figure 5.3, with Table 5.1 showing the distribution types and their respective hyper-parameters. We are assuming prior knowledge of the average ground

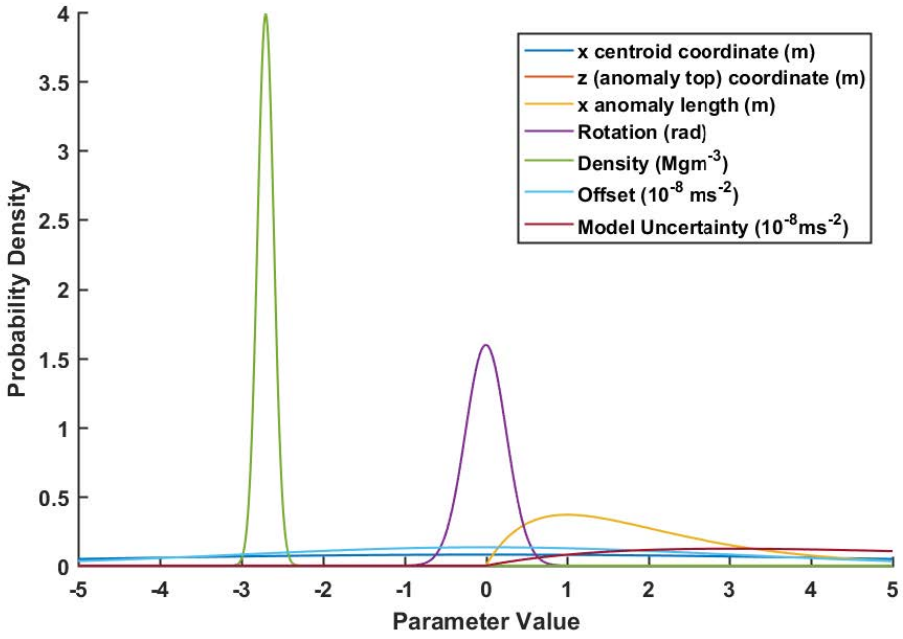


Figure 5.3: Prior distributions of key model parameters. Note that the y-centroid has the same distribution as the x-centroid, similarly for the x,y and z lengths.

density, $\Delta\rho$. It is usually feasible to estimate the ground density from local borehole measurements, as outlined in Appendix A.2. The rotation parameter, ψ, is tightly defined so that large jumps in degenerate values are avoided. A rotation of $n\pi$ radians, where n is any integer yields exactly the same model. Such degeneracy is only a problem in so far as the stationarity of parameters is used as an indication of reversible-jump Markov chain Monte Carlo chain convergence. The $x-y$ centroid values; x_0, y_0, are given Gaussian prior probabilities linked to the maximum values of the measurement coordinates. Such an assignment assumes that the measurement grid is rectangular and has been shifted to the coordinate origin. Any prior information based on measurement position is completely dependent on the data-set being analysed, and should be altered accordingly. The depth to the top of the anomaly, z_0 is given a Gamma distribution. Ensuring that any model object cannot be above the $z = 0$ plane (which is equal to 'ground-level' here, but in general this will not be true for non-uniform z measurement positions). The three length parameters; ℓ_x, ℓ_y, ℓ_z are always positive, as a result Gamma distributions are used. Depending on the type of anomaly being investigated, it may be appropriate to scale the Gamma distribution scaling hyper-parameter of the ℓ_z

Gamma Prior Distribution				Normal Prior Distribution			
Parameter	Unit	a (shape)	b (scaling)	Parameter	Unit	μ	σ
z_0	m	2	1	x_0	m	0	$\max(\boldsymbol{x})$
ℓ_x	m	2	1	y_0	m	0	$\max(\boldsymbol{y})$
ℓ_y	m	2	1	ψ	rad	0	0.25
ℓ_z	m	2	1	η	10^{-8} ms^{-2}	0	$\langle\boldsymbol{\sigma_d}\rangle$
σ_m	10^{-8} ms^{-2}	2	$\langle\boldsymbol{\sigma_d}\rangle$	$\Delta\rho$	Mgm^{-3}	-2.7	0.1

Uniform Prior Distribution			
Parameter	Unit	Minimum	Maximum
k	Dimensionless	1	10

Table 5.1: List of prior distributions used for all model parameters.

parameter prior distribution accordingly; as we may suspect that the height of an anomaly will not exceed more than a few metres for man made structures. The gravity offset, η is assigned a Gaussian distribution, scaled relative to the average of the measurement uncertainty vector. As we have already subtracted the mean of the model, we know that the offset parameter will be approximately zero. The model uncertainty, σ_m, will take values greater than zero. A reasonable scaling value is again the average of the measurement uncertainty vector, meaning that we expect it to be of the order of our measurement uncertainty. The number of model objects, k, is assigned a uniform prior between a minimum of one and a maximum of 10. As we have no information of this parameter beforehand, the maximum bound is set for practical reasons of computational time.

The choice of prior probability distributions used is subjective. Hopefully we have given plausible reasons for the choices made here. Explicitly stating the priors in this way allows interpretation of inference results to be seen in the context of the available model space.

The importance of the parameter prior distributions is magnified here as we readily make draws from the prior distributions during the reversible-jump Markov chain Monte Carlo birth update. We can see that the prior probability distributions favour the generation of small model objects with length parameters of the order of 1 m. Birth steps generally create a large change to the forward model, and as a result are accepted very rarely. So it is advantageous to have a high probability to generate smaller model objects that are more likely to be accepted. The initial Cauchy proposal distribution (see section 4.1.1) scaling

parameters for the model parameters are defined as shown in Table 5.2. These parameters were chosen by trial and error. However, it makes little difference when using an adaptive Markov chain Monte Carlo scheme to tune the scaling values over the burn-in period. The

x_0	y_0	z_0	ℓ_x	ℓ_y	ℓ_z	ψ	$\Delta\rho$	σ_m	η
0.05	0.05	0.1	0.1	0.1	0.1	0.05	0.1	0.05	0.05

Table 5.2: List of proposal distribution scaling parameters for the Cauchy distribution.

scaling parameter for the reversible-jump Markov chain Monte Carlo update in which all parameters are changed simultaneously is initiated as 0.5. This parameter is required, as the more parameters that are updated simultaneously the lower the chance of acceptance becomes. As more parameters are changed simultaneously there is a higher chance of the proposed model changing dramatically with respect to the current model.

Having defined the forward model, prior probability distributions and parameter proposal distributions, we can proceed as outlined in Chapter 4. We run the reversible-jump Markov chain Monte Carlo algorithm for 200,000 iterations, a number chosen based on trial and error of initial runs.

We employ both the adaptive Markov chain Monte Carlo scheme and simulated annealing outlined in Sections 4.1.2 and 4.1.3 respectively. The simulated annealing parameters were set as, $t_0 = 1 \times 10^6$ and $C = 0.99999$. Meaning that after approximately one thousand iterations the acceptance ratio returns to the correct Metropolis-Hastings ratio.

The adaptive Markov chain Monte Carlo scheme was updated at every thousandth iteration, if the acceptance percentage of a parameter block was greater than 0.4 the proposal scaling was multiplied by 1.1. If the ratio was less than 0.1 the proposal scaling was multiplied by 0.9. As explained in Section 4.1, we have two possible standard Metropolis-Hastings updates, one where all parameters are updated together, and one where we select a group of parameters from one object to update in isolation. Two groups of parameters were defined, group one consisted of x_0, y_0 and ψ, group two consisted of the rest of the object parameters. These groups were chosen because the group one parameters are often well defined by gravity measurements, whilst the parameters in group two are highly correlated. After a

pre-defined burn-in period of 50,000 iterations (25% of the total number of iterations), the adaptive scheme is halted and the proposal distributions are fixed at their current values. For this particular implementation, a chain of 200,000 iterations took approximately fifteen minutes to run (on a standard laptop computer).

After the algorithm has finished, the marginal posterior distributions of all the model parameters are checked for obvious convergence failure using the Geweke diagnostic as described in section 4.1.4. The number of model objects k is also tested with the diagnostic. If the Geweke diagnostic gave a value greater than 3 for any parameter, we take a close look at the parameter trace and rejected the run if it was clear that convergence had failed. Aided by trace plots and autocorrelation plots. For this particular run, the Geweke diagnostic values are displayed in Table 5.3. Figures 5.5 and 5.6 show the output trace plots and histograms

x_0	y_0	z_0	ℓ_x	ℓ_y	ℓ_z	ψ	$\Delta\rho$	σ_m	η
0.006	0.07	0.059	-0.39	-1.08	0.86	-0.08	-0.24	-0.1	0.05

Table 5.3: Geweke diagnostic values for all model parameters.

of the separate model parameter posterior distributions. It is useful to show both the histograms and the trace side by side, as the trace gives information on the quality of mixing for a given parameter that a histogram alone does not. The object number, k is not shown as it is effectively constant at $k = 1$; meaning that the run has found the simplest model to fit the measurement data. Each trace shows the distribution with mean values (solid line), maximum a posteriori probability (MAP, dash-dotted line), 95% credibility interval (dashed line) and the actual parameter value (red line). The histograms also show the parameter prior probability distribution (blue line). We can see that the prior distributions have no effect on the posterior distributions, except for the density parameter which follows the prior distribution, telling us that the measurement data contains no information to improve upon our prior knowledge. All parameters were found within the 95% credibility intervals, except the for the model uncertainty. For this synthetic case we know that the model uncertainty is actually zero. However, for real measurement data this would not be the case. Analysing the parameter trace plots it is clear that we have correlation between the: z_0, ℓ_x, ℓ_y and

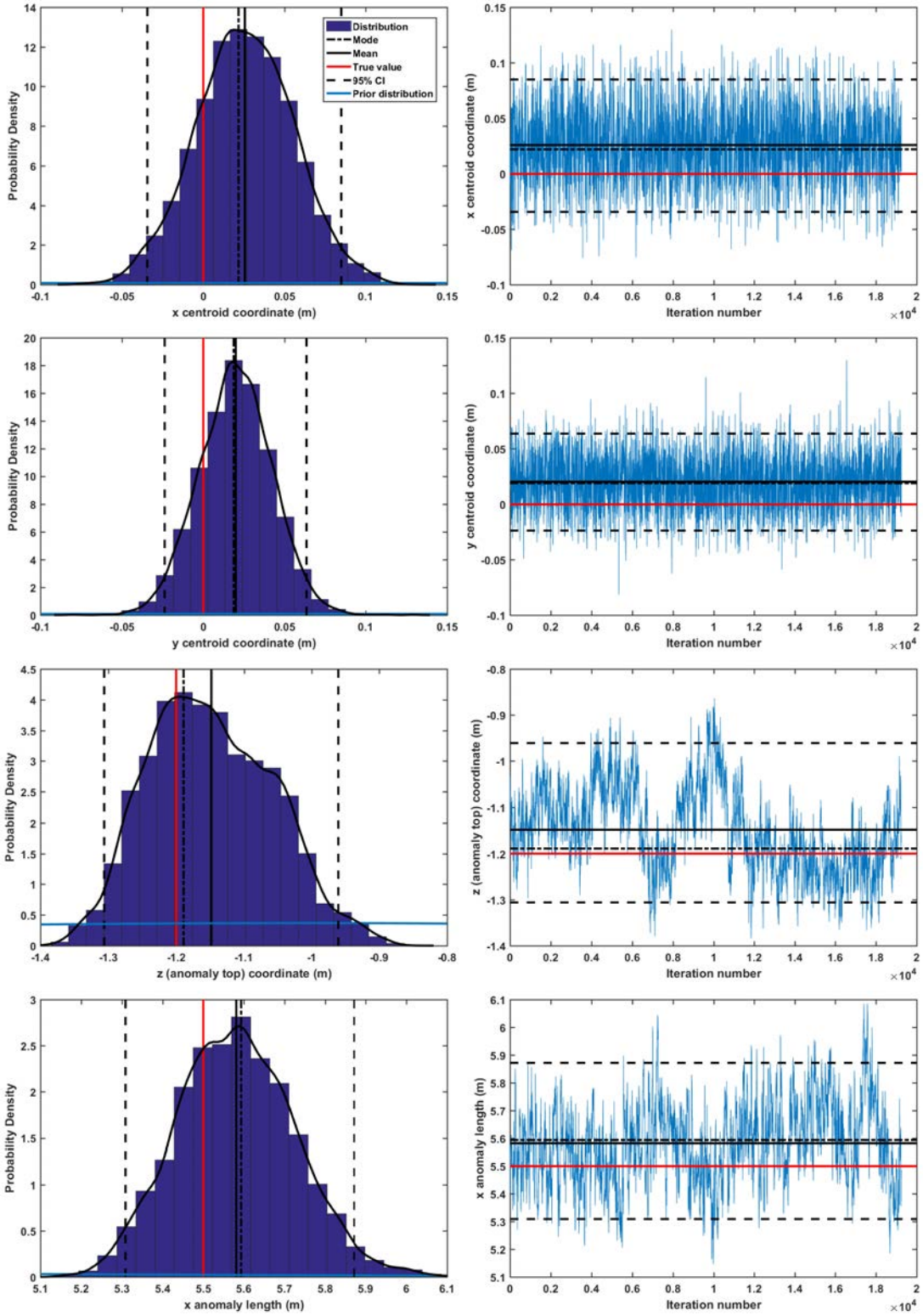


Figure 5.4: Histogram and trace plots of the algorithm output for the x_0, y_0, z_0, ℓ_x model parameters (from top to bottom).

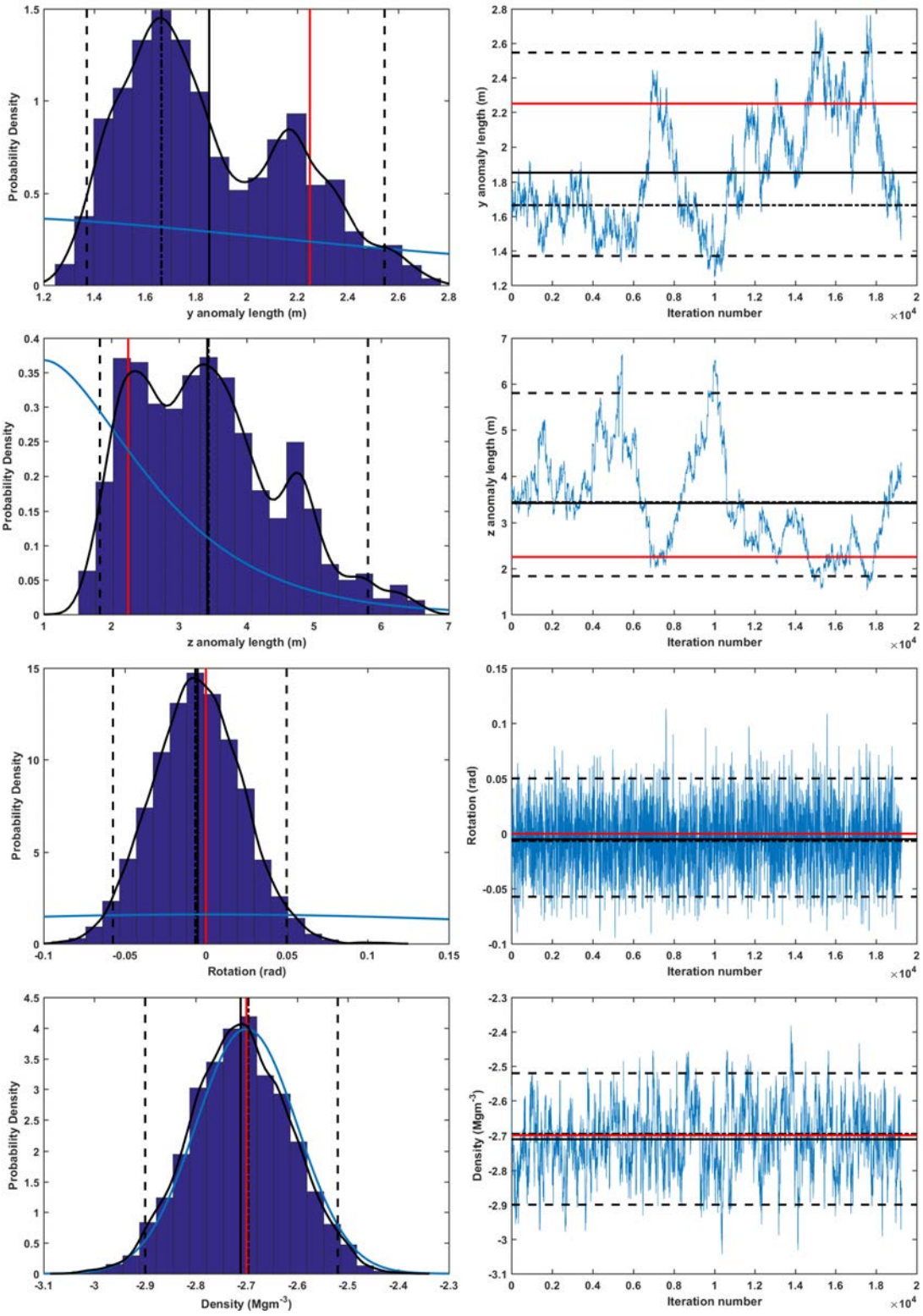


Figure 5.5: Histogram and trace plots of the $\ell_y, \ell_z, \psi, \Delta\rho$ model parameters (from top to bottom).

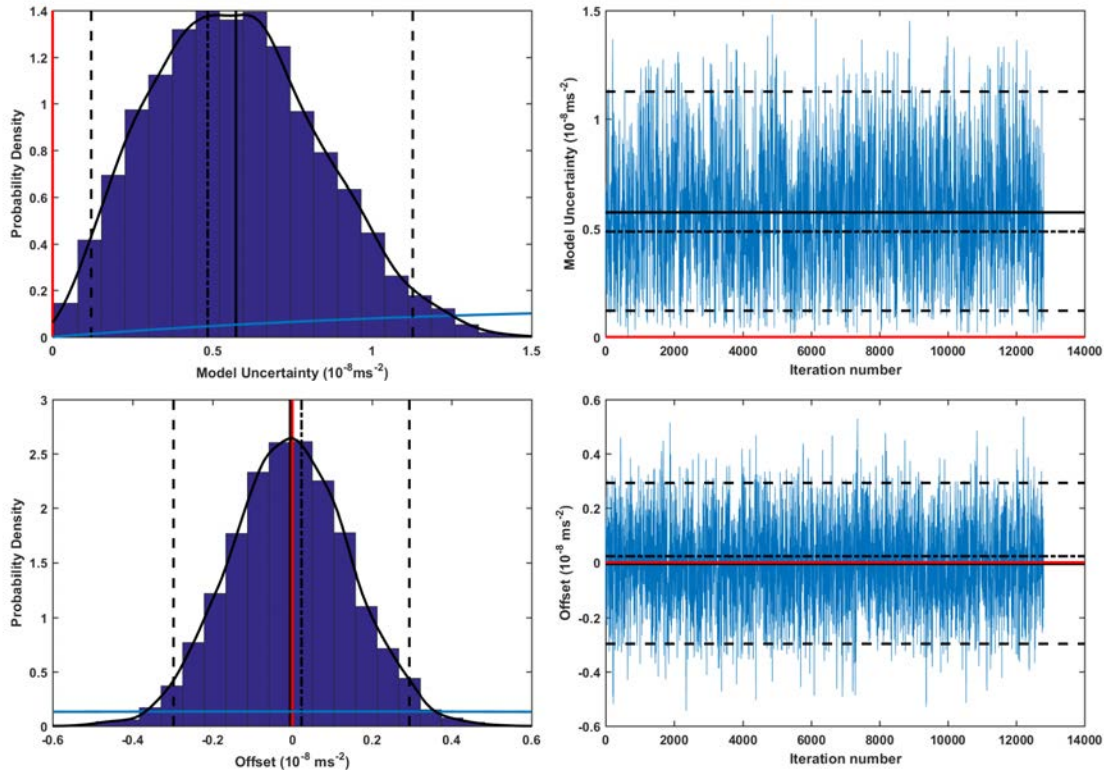


Figure 5.6: Histogram and trace plots of the model uncertainty, σ_m and gravity offset, ν, parameters (from top to bottom).

ℓ_z parameters (all defined in parameter block two). The ℓ_y and ℓ_z parameters are highly correlated and show signs of poor mixing and hence slow convergence. Figure 5.7 shows a scatter plot of the two parameters output vectors. The correlation value obtained using the MATLAB *corr* function was -0.91. There is correlation between all three volume parameters, due to the unavoidable ambiguity between them. The smallest side parameter (ℓ_y) is less well defined than the longer side length due to the finite measurement grid spacing. The forward model is least sensitive (relative to the other length parameters) to the height of the void ℓ_z. Due to the re-parametrisation described earlier, any change in ℓ_z adds volume to the cuboid at its lowest point, which is furthest from the measurement grid. As a consequence, altering the value of ℓ_z has a small affect on the forward model values when contrasted with a comparable change to either the ℓ_x or ℓ_y parameters.

We can combine the marginal posterior distributions of the three length parameters; ℓ_x, ℓ_y and ℓ_z to obtain the marginal distribution for the object volume. Also, we can rescale the z_0

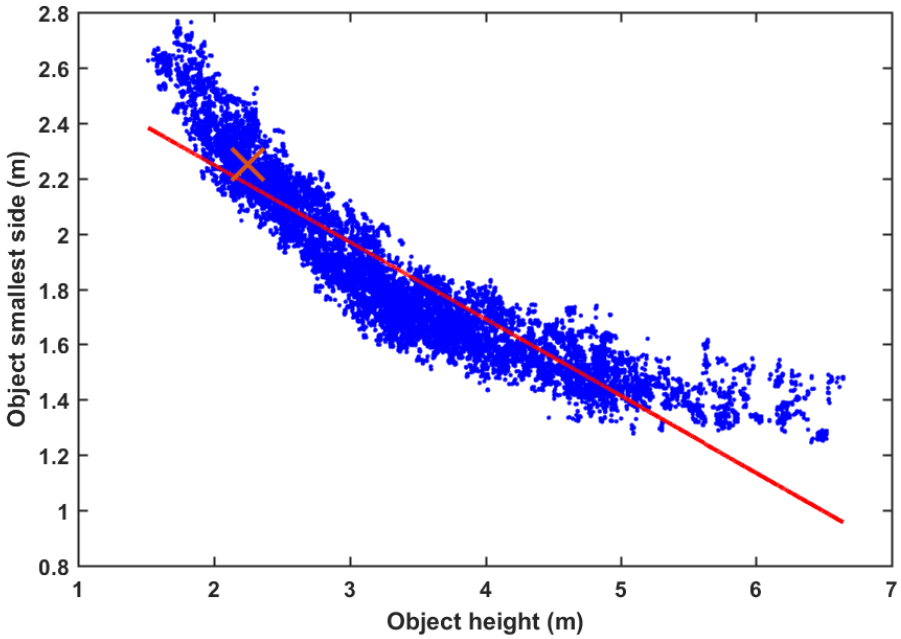


Figure 5.7: Correlation between ℓ_y and ℓ_z parameters. Red line shows linear fit, with the orange cross indicating the actual parameter values.

parameter to represent the depth to centroid instead of the depth to top. Figure 5.8 shows these combined posteriors. As we might expect, correlation between model volume and depth is clearly shown. Comparison of the centroid depth and depth to the model object top as seen in Figure 5.5 shows that the depth to the top is a more well defined parametrisation than the centroid depth alone. Figure 5.8 also shows the combined volume and density of the object which gives the object mass. In this case the value is negative, as we are dealing with a void. As previously discussed, it is difficult in a reversible-jump Markov chain Monte Carlo algorithm to compare parameters between runs due to the problem of converging to models of varying complexity. To sidestep this problem, we can investigate combinations of parameters that are consistent between individual reversible-jump Markov chain Monte Carlo runs. One such combination is the model total mass,

$$\boldsymbol{M}_{\text{total}} = \sum_{k=1}^{k_{\max}} \boldsymbol{\Delta\rho}_k \boldsymbol{V}_k \tag{5.4}$$

where we sum the mass of each model object. Figure 5.9 shows the correlation between the object centroid depth and the object model mass. A correlation coefficient of 0.99 was

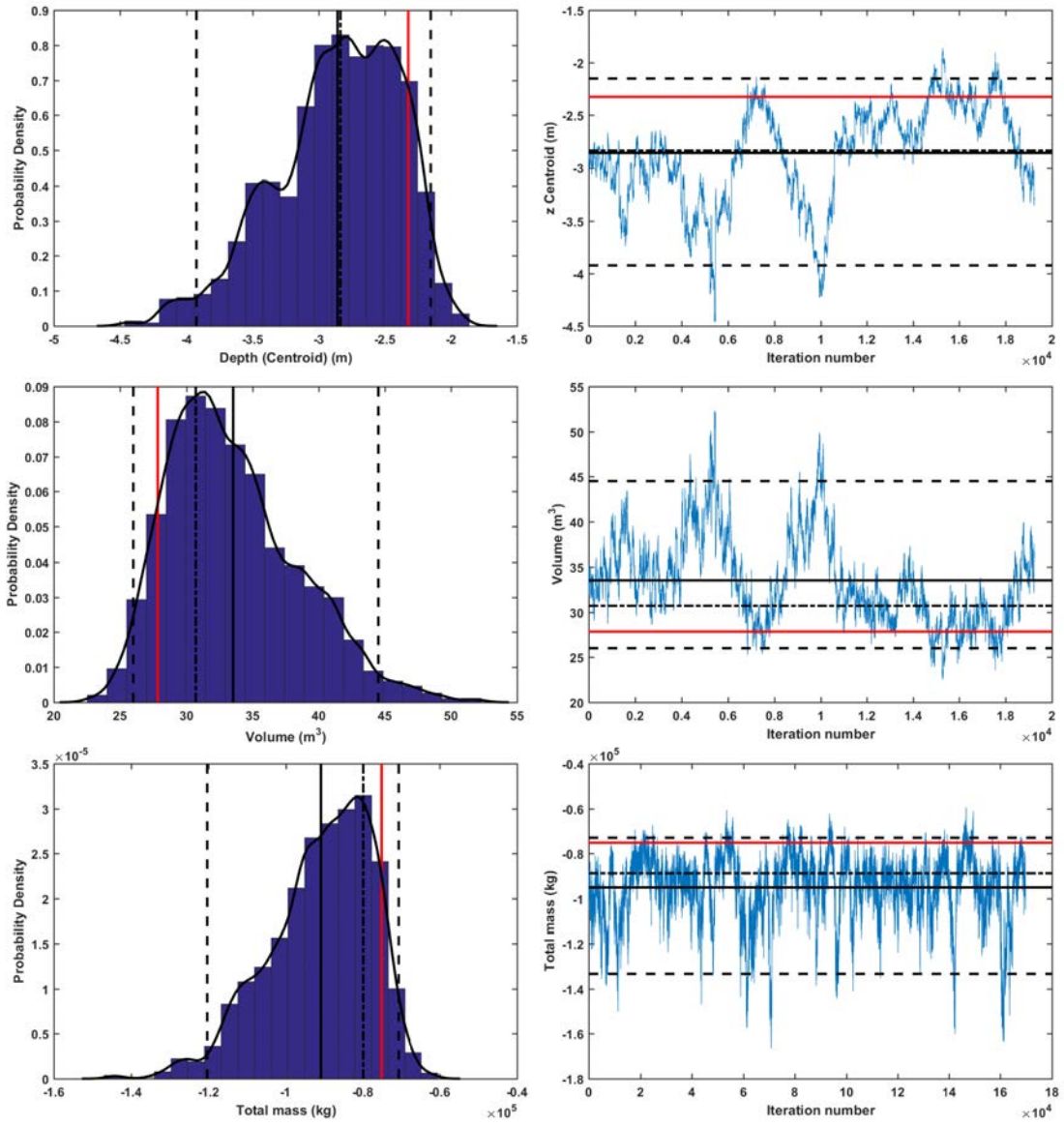


Figure 5.8: Combined length parameter posteriors show the object volume. A re-parametrisation of the z_0 and ℓ_z parameters give the depth.

found using the MATLAB function *corr*. It is clear from Figure 5.9 that the high probability regions of the Bayesian posterior distribution are situated on a thin line in the model space. Although the histograms and trace plots above give us useful information on the convergence and uncertainty of parameters or combinations of parameters, it is not clear how this information can be used to aid decision making by a potential end-user of the algorithm. In an attempt to display the information obtained in a more intuitive way, one hundred random model samples are drawn, each plot in 3D with one percent transparency. A comparable

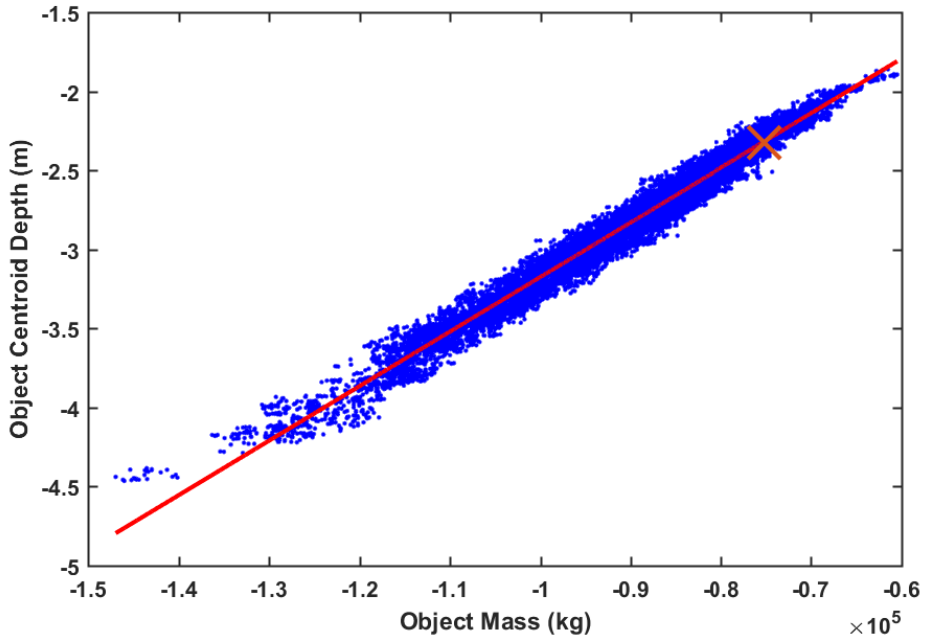


Figure 5.9: Scatter plot of centroid depth and object mass. Red line shows linear fit, with the orange cross indicating the actual parameter value.

technique was first proposed in [107], where similar plots are called 'probability of excavation maps'. These maps were constructed by splitting the model space into cells, and increasing each cells transparency relative to the number of posterior models encompassed by each cell. Here we do not claim that we are displaying a probability of excavation, as we only plot a small subset of the total number of accepted models. However, such plots are still useful for displaying the models spatially, with darker areas of the plot being proportional to higher probability models. We only make one hundred draws due to the limitations of MATLAB's transparency rendering. It should be noted that such plots do not render correctly using versions of MATLAB later than 2014b, as after this version major changes were made to graphics rendering. Figure 5.10 shows various views of such a plot, with the actual anomaly location shown in red. It is clear that the object is well resolved in the xy plane, along with the depth to the top of the object. The data contain less information regarding the height of the buried object. The usefulness of such a plot will become clearer in later sections, where we deal with more complex multi-object models.

Above, we have tried to focus on displaying the Bayesian posterior information in terms of

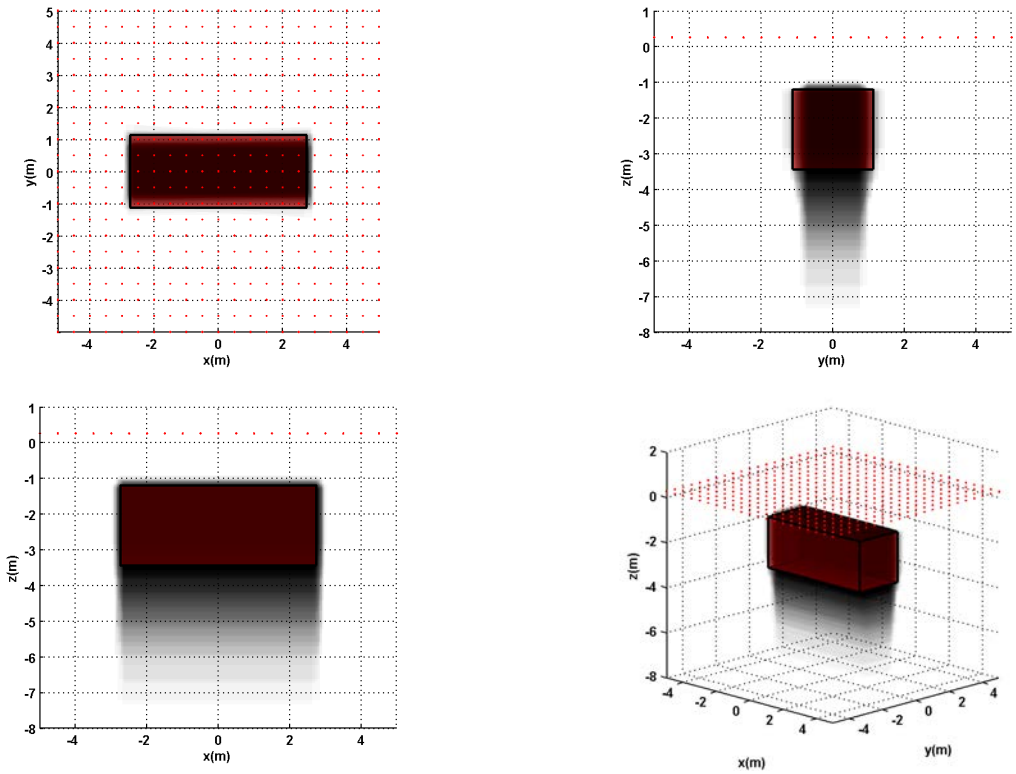


Figure 5.10: Plots of one hundred posterior models each of one percent transparency. The actual model is shown in red.

the marginal distributions, as we believe that it is the best way to display the uncertainty inherent in any inference problem. Any attempt to give a single 'correct' solution to the inference problem is in danger of giving an over confident interpretation of the inference process. However, in the interest of completeness, Figure 5.11 shows the forward models of the mean and maximum a-posteriori (MAP) models, as compared to the synthetic data. As with any Markov chain Monte Carlo sampler, we must run multiple separate chains to ensure that we fully explore the Bayesian posterior distribution, as one run alone could get stuck in a local minima due to the necessarily finite number of iterations. Now that we have summarised the outputs for one reversible-jump Markov chain Monte Carlo run, we now focus on interpreting and displaying the outputs of many separate runs.

Ten independent runs were performed, Figure 5.12 shows the distribution of the number of model objects, k. This is not corrected for burn in, as the sampler finds the simplest solution ($k = 1$ in this case) within a few thousand iterations. During a reversible-jump Markov chain

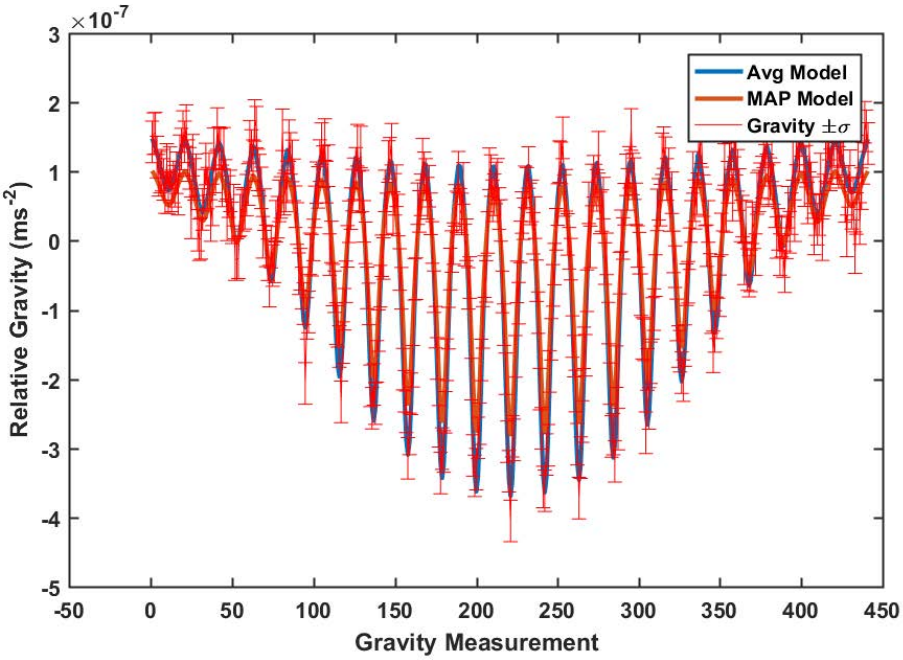


Figure 5.11: Forward models calculated using point estimates. In this instance the average model values fit the data-set more closely than the maximum posterior model values.

Monte Carlo run, early iterations usually fit the data using a complex model (large k). As the sampler proceeds, model objects that are superfluous become marginalised. The objects' contribution to the forward model decreases, by increasing its depth or decreasing its volume, or both. At some stage, a death update will delete these marginalised objects.

Of the ten separate runs, one run was deemed not to have converged satisfactorily, alerted to by multiple Geweke diagnostic values larger than the threshold. As previously stated, we can use the total mass (Equation 5.4) to summarise the output of multiple chains, without regard for the parameter switching problems of the reversible-jump Markov chain Monte Carlo process. We can also combine the traces of parameters whose meanings stay the same between models of varying k; the model uncertainty parameter σ_m, the gravity offset parameter η and the total mass, M. Figure 5.13 shows the plots of these parameters, combined from the successfully converged data-sets. We see that the individual runs converge to the same model parameter values, as we would expect given that all of the runs converged to one model object. The total mass of the void is found within the 95% credibility interval, having considerable spread in favour of a larger total mass.

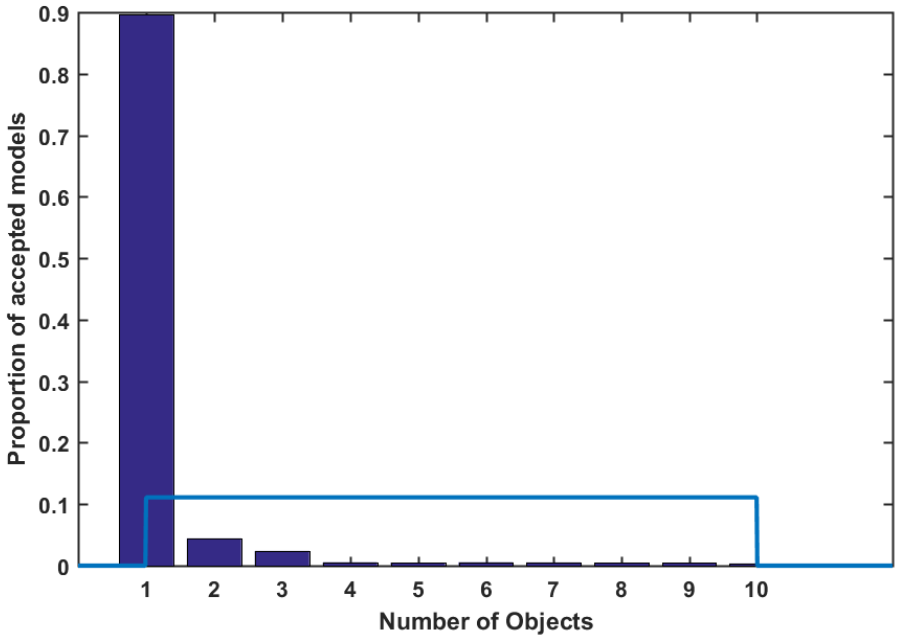


Figure 5.12: Number of model objects, k. Data not corrected for burn-in. The simplest model $k = 1$ is found for all converged runs. Prior probability distribution for the number of model objects is uniform.

This initial analysis of a single cuboid model has demonstrated that the algorithm is capable of finding the parameters of the buried void within the uncertainty.

5.1.2 Sphere Forward Model

Now we consider the more general case where we do not know the exact forward model relation. In such a case, we have little choice but to resort to the simplest gravity forward model, that of a uniform density sphere. The sphere model can be used as a reasonable approximation to any compact anomaly measured at a distance of a few diameters of the anomaly[41]. We use Equation 5.2 as our forward model, where U_z is given by equation 2.7. The sphere model contains $5k + 2$ model parameters. Three less parameters per object than the cuboid model. However, more objects may be required to fit a given data-set relative to the cuboid model which has more degrees of freedom. The prior distributions used are shown in Figure 5.14, the respective hyper parameters are displayed in Table 5.4. The scaling factors for the initial Cauchy proposal distributions are shown in Table 5.5. We have similar reasoning for the assignment of prior distributions here as we did in the previous section.

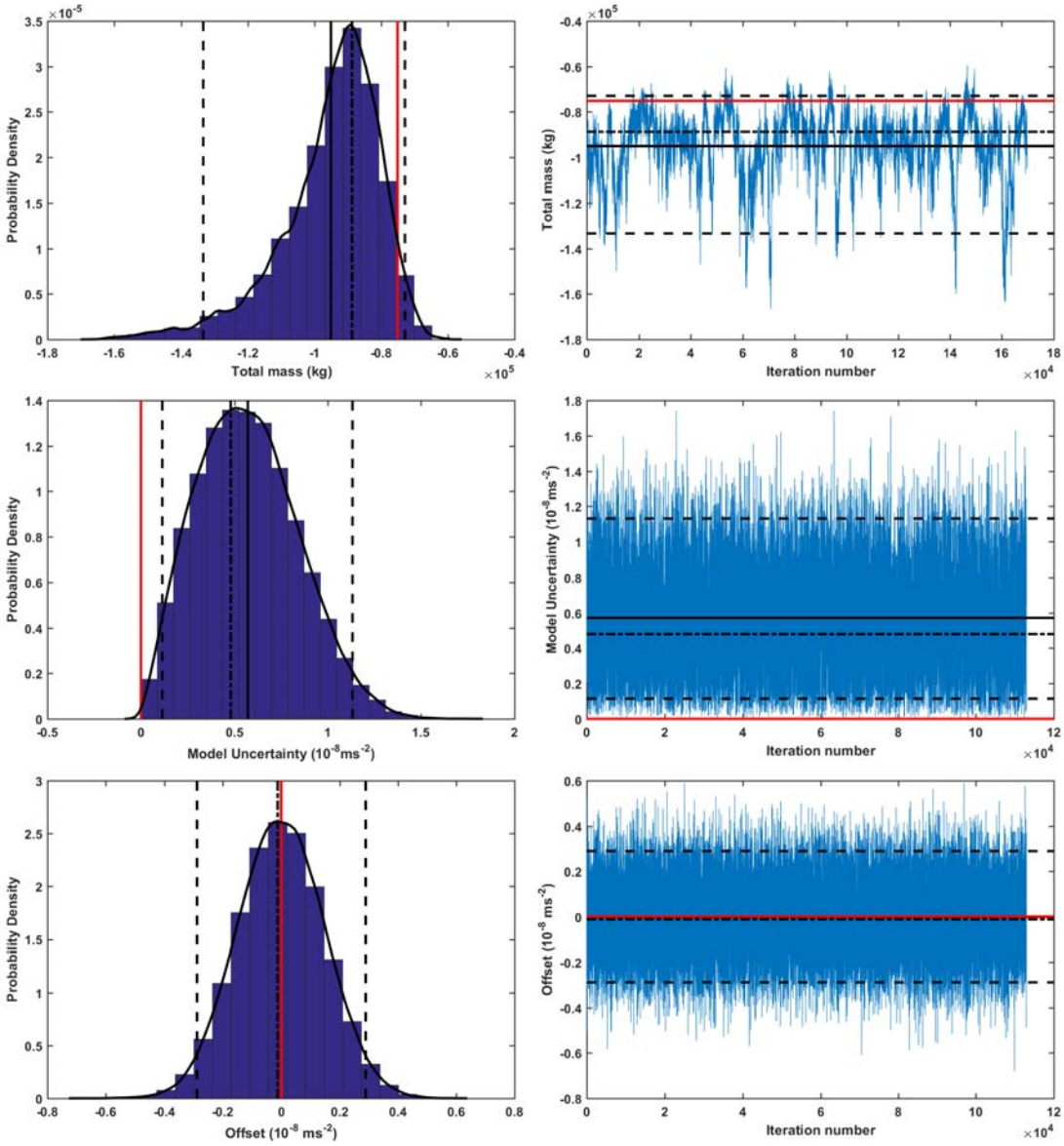


Figure 5.13: The model uncertainty, gravity offset and total model mass retain the same meaning between models of differing k value. The marginal posterior distributions of these parameters can be combined from multiple runs.

The sphere radius, R, is given a Gamma distribution peaked at 0.5 as this is of the order of our expected anomaly scale.

When using the sphere forward model we must acknowledge that we are no longer interested in direct parameter estimation. At least not of all of the sphere parameters. We are more concerned with the combined effect of multiple model objects. A single sphere cannot give us meaningful knowledge of a complex anomaly structure. We need to interpret the model objects together. Let us begin by showing the spatial plot of ten combined reversible-jump

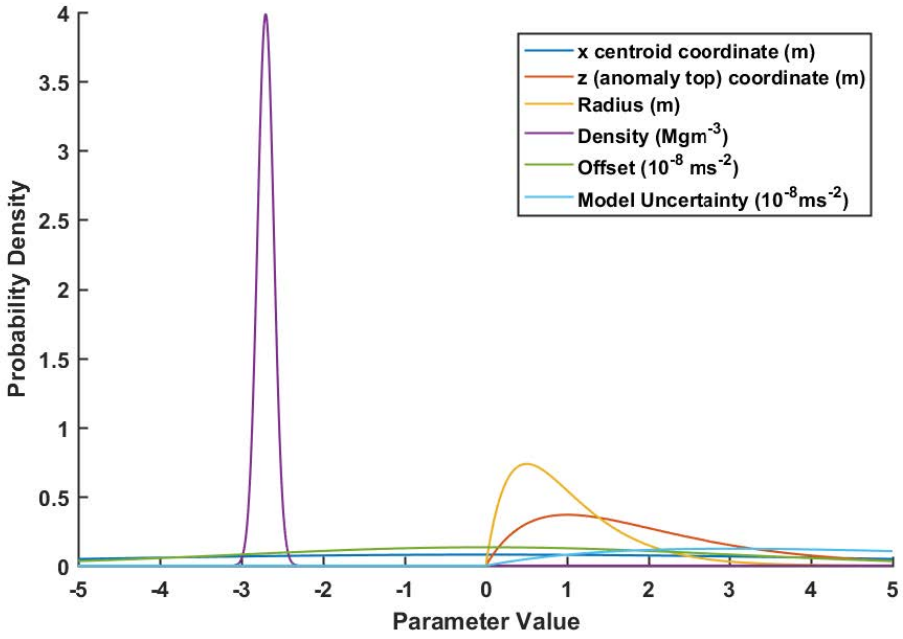


Figure 5.14: Prior probability distributions used for the sphere forward model.

Gamma Prior Distribution

Parameter	Unit	a (shape)	b (scale)
z_0	m	2	1
R	m	2	0.5
σ_m	10^{-8} ms^{-2}	2	$\langle \boldsymbol{\sigma_d} \rangle$

Gaussian Prior Distribution

Parameter	Unit	μ	σ
x_0	m	0	max x
y_0	m	0	max y
η	10^{-8} ms^{-2}	0	$\langle \boldsymbol{\sigma_d} \rangle$
$\Delta\rho$	Mgm^{-3}	-2.7	0.1

Uniform Prior Distribution

Parameter	Unit	Minimum	Maximum
k	Dimensionless	1	10

Table 5.4: List of prior distributions used for all sphere model parameters.

Markov chain Monte Carlo runs, eight of which were deemed to have converged successfully using the Geweke diagnostic. Figure 5.15 shows the spatial distribution of the sphere objects. We can see that the algorithm converged to a model consisting of two spheres with which to fit the anomaly, as confirmed by Figure 5.16. The algorithm is finding the simplest model to describe the data within the total uncertainty. We shall see later on that data with smaller uncertainty leads to a more complex model (a model consisting of more objects). One parameter that we would expect to fit with the known model parameters is the object depth. Taking one of the converged runs, Figure 5.17 shows the trace and histogram plot for the depth of object one. We can see that the anomaly depth is found within the 95%

x_0	y_0	z_0	R	$\Delta\rho$	σ_m	η
0.05	0.05	0.1	0.5	0.1	0.05	0.05

Table 5.5: List of proposal distribution scaling parameters for the Cauchy distribution.

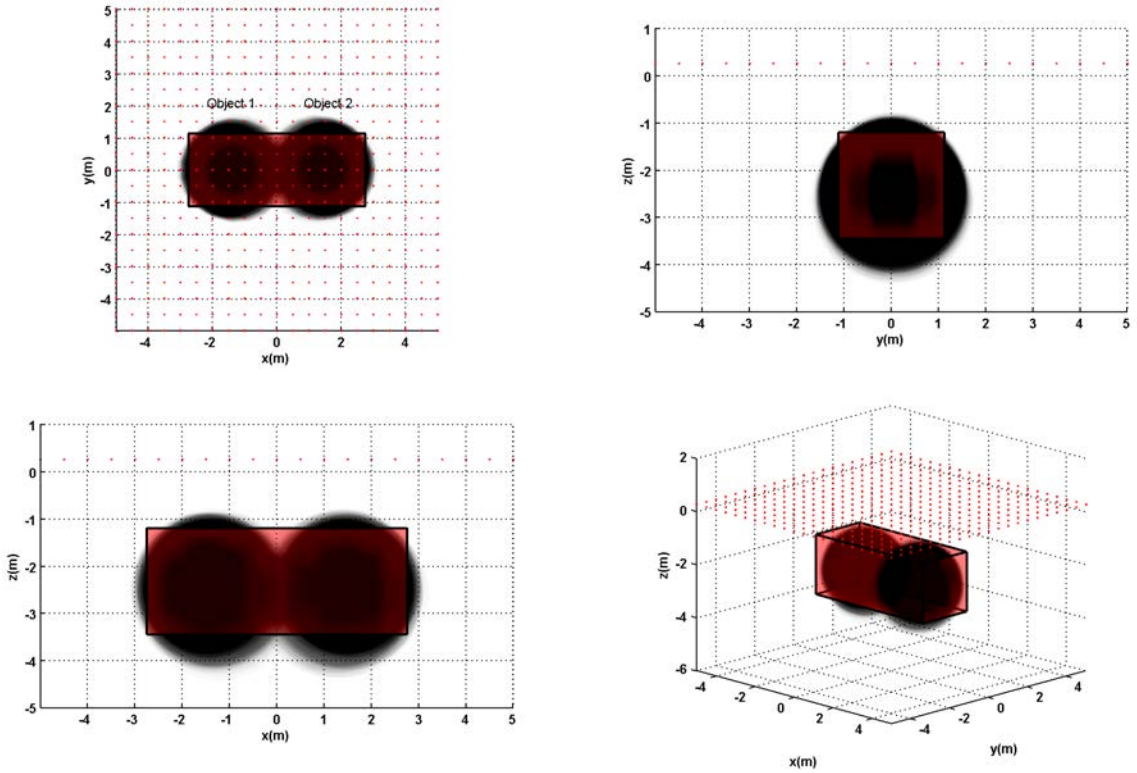


Figure 5.15: Spatial plot of ten combined runs of the reversible-jump Markov chain Monte Carlo algorithm using the sphere forward model.

credibility interval. Now that our output model consists of multiple objects, we not only have to be concerned with correlation between object parameters as we saw with the cuboid model, but indeed correlation between objects themselves. It is clear in this scenario that the x centroid parameter of both of the objects may well be correlated, as the objects 'wobble' about some average position. Figure 5.18 confirms this, with a calculated correlation value of 0.81. As before, we can summarise the parameters whose interpretation does not change with k, the total mass, σ_m and η. Figure 5.19 shows their respective trace and histogram plots. A direct comparison of the total mass obtained using the sphere model and that obtained using the cuboid model (Figure 5.13) shows that the sphere model fits the total mass more closely than the cuboid model. A result that is perhaps somewhat unexpected given that the

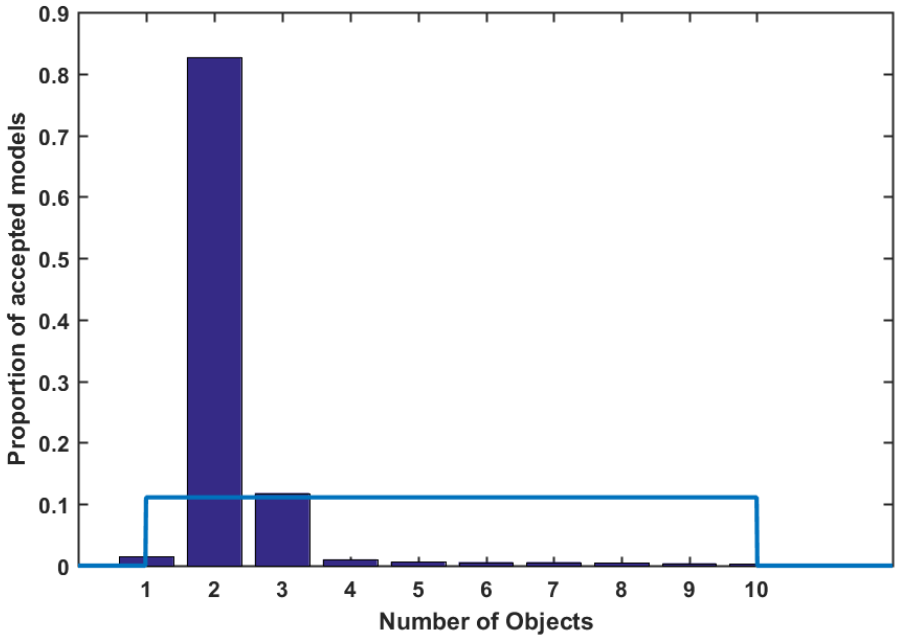


Figure 5.16: Combined number of objects, k for eight converged reversible-jump Markov chain Monte Carlo runs. Two objects are favoured to fit the data-set.

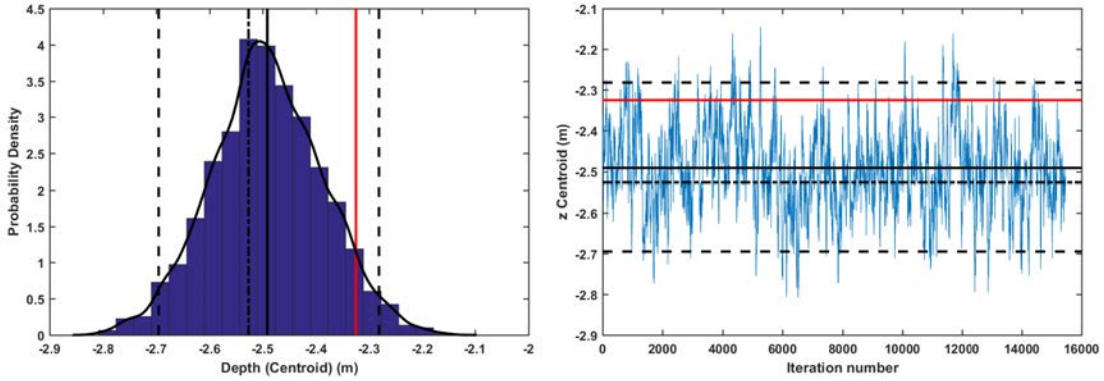


Figure 5.17: The centroid depth of sphere 1. Matches the known anomaly centroid (red line) within the 95% credibility interval (dot-dash lines).

synthetic data was generated using a cuboid model. This result may be due to the simpler nature of the sphere model; it does not have the same problems of within object parameter correlation between length parameters as was observed for the cuboid model (see Figure 5.7) as its volume is determined solely by the radius parameter. An obvious trade off with using the sphere model is the inability to fit the exact geometrical parameters of the anomaly. However, the need to obtain such precise geometrical information from a small scale gravity survey may be unimportant compared to the model parameters of object depth and total

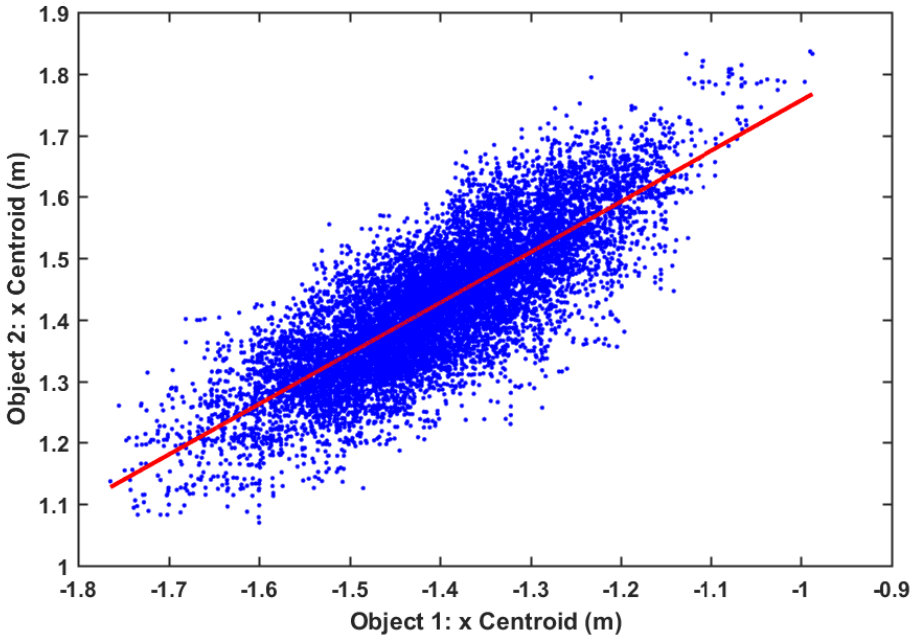


Figure 5.18: The x centroid values of the two objects are correlated.

mass, which were inferred within the calculated credibility intervals.

The sphere model does have a number of advantages over the cuboid model, the obvious being that the model itself is much simpler and can be computed more quickly than the cuboid model. It also does not require extra angle parameters to be added in order to fit anomalies which are not at constant depth, as would be required to extend the cuboid model to cope with such a scenario. Finally it is also potentially easier to extend to more complex models where ground density is non-uniform.

5.1.3 Decreased Data Density

The number of measurement points used for the previous data-set (411) was large in terms of what is currently realistically possible with the gravity method, especially at the measurement uncertainty level of 3×10^{-8} ms^{-2}, without the survey extending into multiple weeks. We briefly consider an increase in the grid spacing to from 0.5 m to 1 m. This reduces the number of data points to 121, a more realistic amount to commit to such a small area. The exact values for these data points are used for the previous analysis, by simply deleting every second measurement row and column. Doing this we can be sure that we are comparing

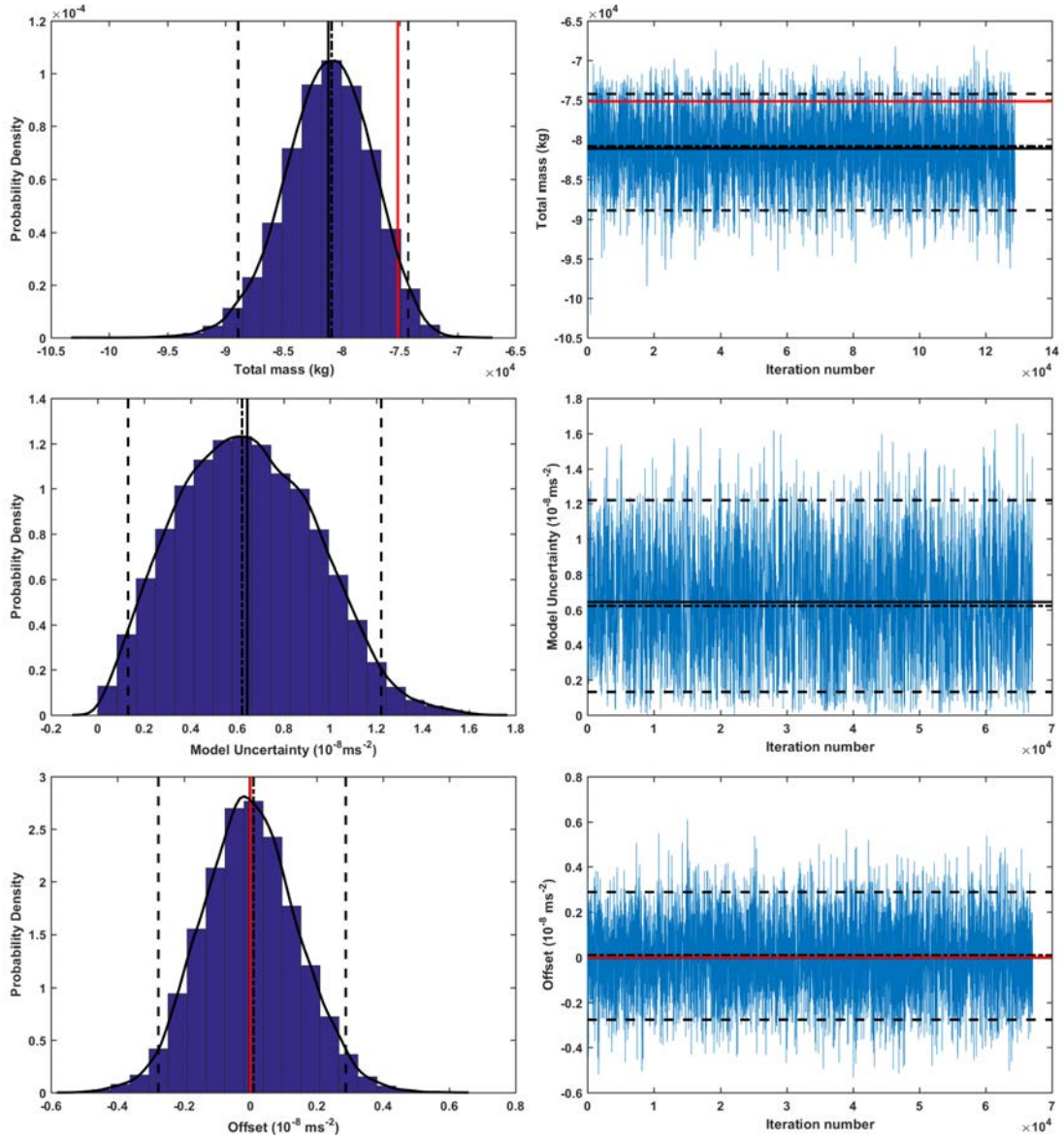


Figure 5.19: The model uncertainty, offset and total mass retain the same meaning between models of differing k value. The marginal posterior distributions of these parameters can be combined from multiple runs.

like for like, the same survey but with less data points. Figure 5.20 shows the reduced data grid. Using the same prior probability distributions as for the previous cuboid model, five reversible-jump Markov chain Monte Carlo runs were completed, all of which converged. We truncate the individual analysis to show the combined three dimensional spatial plot, Figure 5.22. As expected, when compared to Figure 5.10, we have increased uncertainty in the location and extent of the anomaly due to the reduced measurement point density. The lower uncertainty obtained when taking 320 more measurements is minimal and not worth

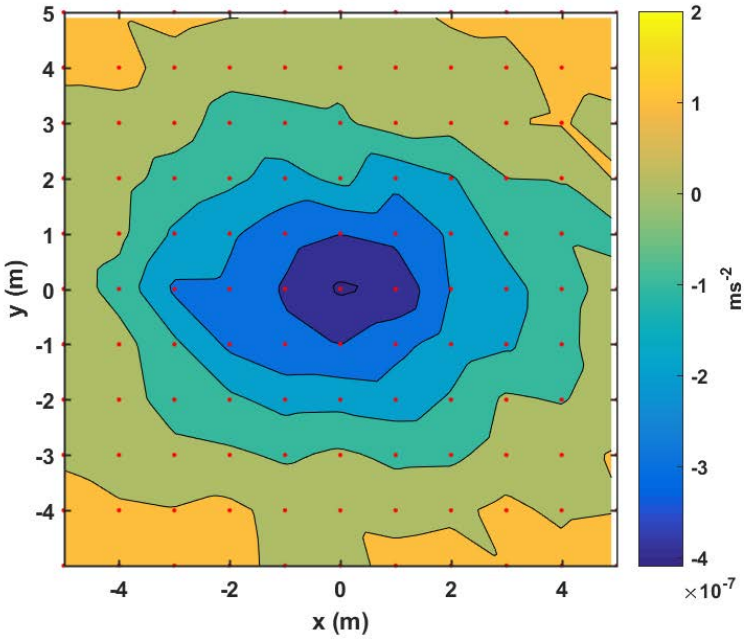


Figure 5.20: Synthetic gravity data (121 points) generated for the nuclear bunker anomaly. Gaussian noise with $\sigma = 3 \times 10^{-8}\text{ms}^{-2}$ was added to the data.

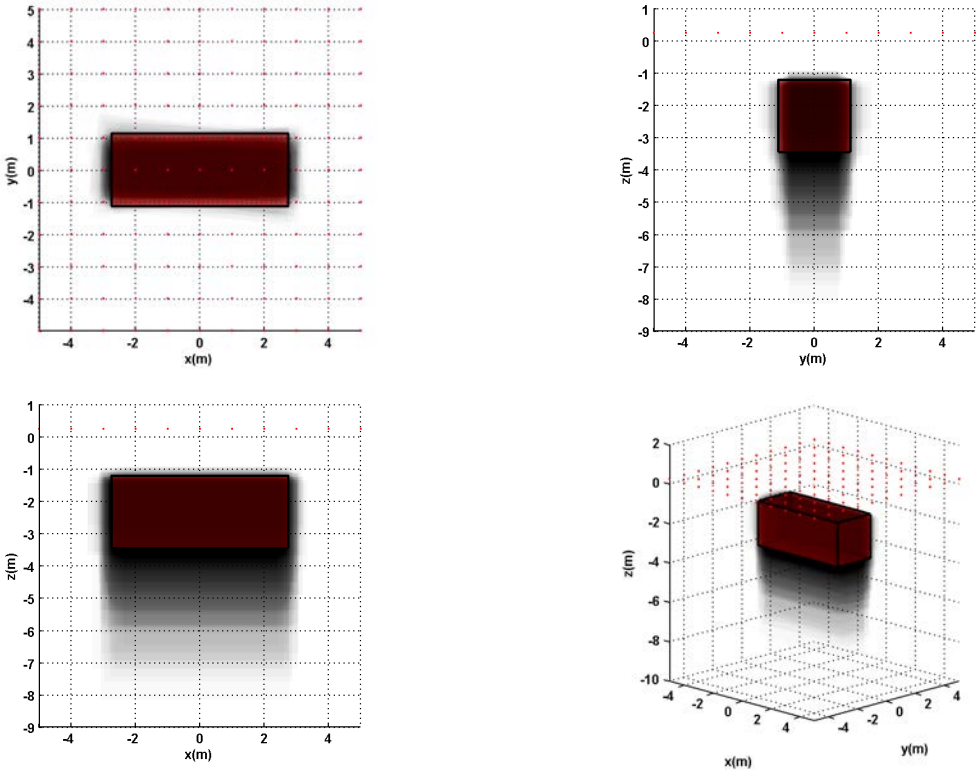


Figure 5.21: Spatial plot of five combined runs of the reversible-jump Markov chain Monte Carlo algorithm, fitting a cuboid model to the reduced data-set.

the considerable added survey effort that would be required.

The corresponding analysis for the sphere forward model results in the spatial distribution shown in Figure 5.22. When compared to Figure 5.15 we can see that the spheres are spatially

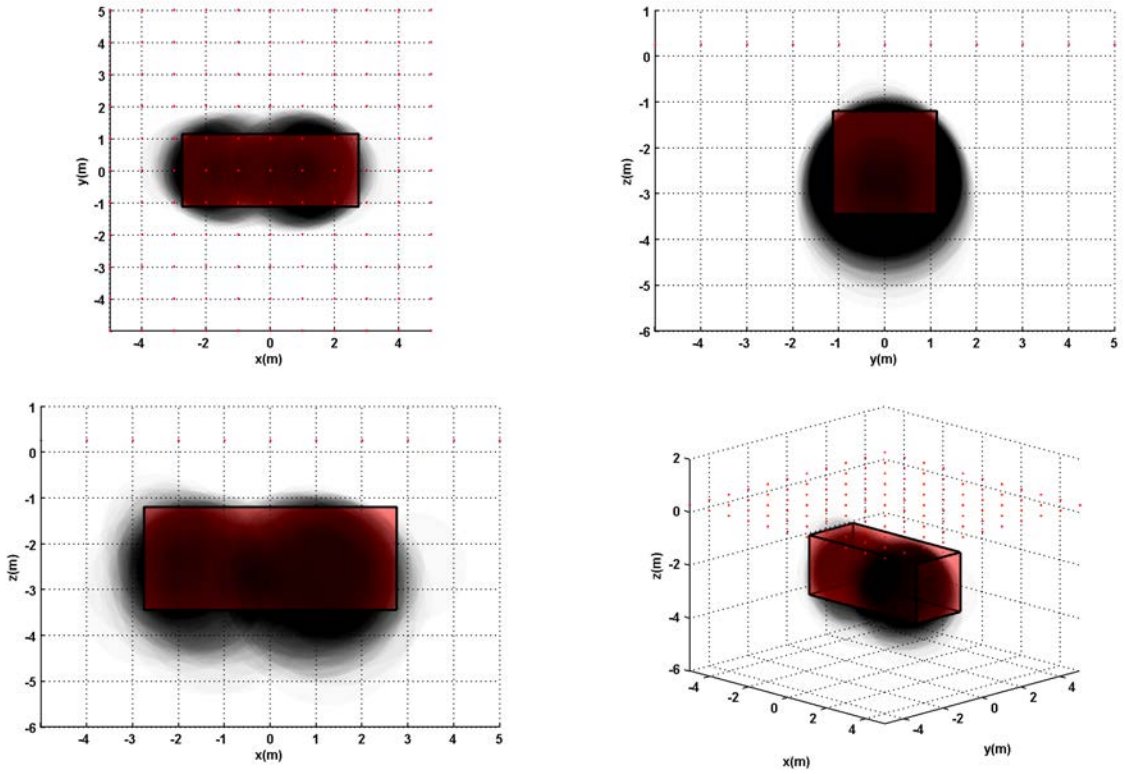


Figure 5.22: Spatial plot of five combined runs of the reversible-jump Markov chain Monte Carlo algorithm, fitting a sphere model to the reduced data-set.

more diffuse. The spheres are no longer symmetrical in terms of position and volume. Due to the decreased number of measurement points, the gravity values at each point (relative to the dense measurement grid) have more influence on the inference algorithm. If we have an outlier in a low density measurement grid, it has a more pronounced affect on the inference than the same outlier would have in a high density measurement grid.

5.1.4 Inference from Synthetic Gravity Gradient Data

We now take a brief look at the results that we might expect if measuring the nuclear bunker anomaly using an atom interferometer gravity gradiometer. We assume a lower cloud measurement position the same as that of the Scintrex CG-5 ($z_a = 0.25$ m) and a cloud

spacing of 0.5 m, giving $z_b = 0.75$ m. Which gives an effective gradient measurement position equal to 0.5 m. Figure 5.23 shows the synthetic data. The signal-to-noise ratio (Equation 5.1) for this data-set is 48.

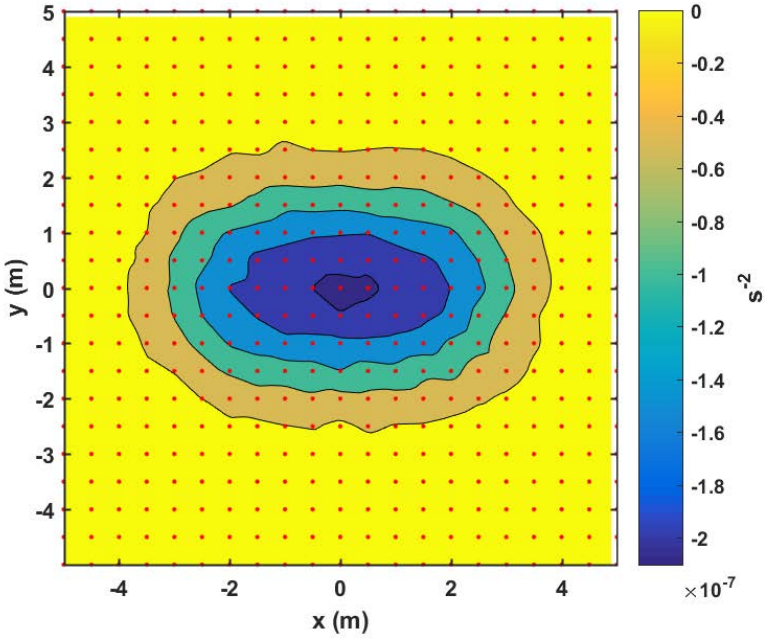


Figure 5.23: Synthetic gravity gradient data generated for the nuclear bunker anomaly. Gaussian noise with $\sigma = 0.5 \times 10^{-8}$ s^{-2} (5 E) was added to the data.

Cuboid Forward Model

We use the forward model as defined by Equation 2.13 taking U_{zz} as given by Equation 2.8. The proposal distribution for the offset η and model uncertainty σ_m are reduced by a factor of 10 from those used previously for gravity data. This is due to the change of scale of gravity gradient measurements. Five runs, each of 200,000 iterations were carried out. Figure 5.24 shows the spatial combination of the five runs, the high signal-to-noise ratio of the data-set results in a very close fit to the actual model. As the signal-to-noise ratio is very high for this data-set, the reversible-jump Markov chain Monte Carlo algorithm is being limited by the uncertainty on the density contrast parameter. In reality we would not expect to obtain a data-set with such a high signal-to-noise ratio, as other effects not accounted for in the synthetic model generation will increase our experimental uncertainty. Issues such

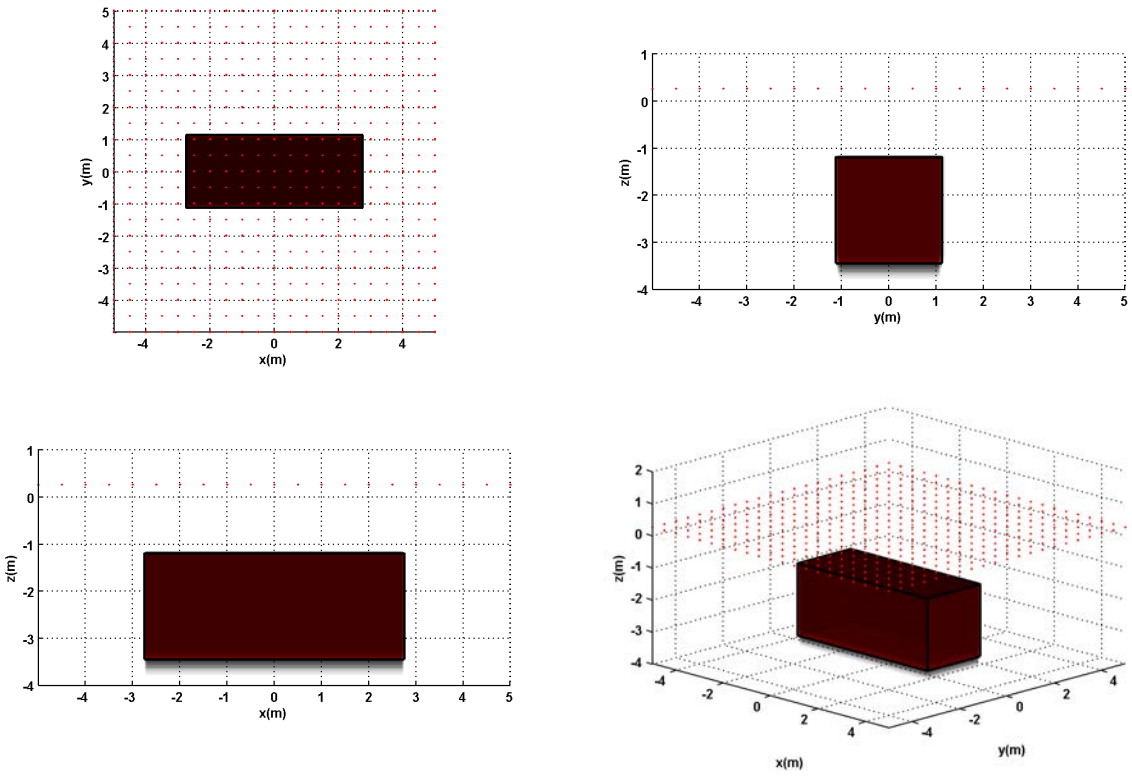


Figure 5.24: Spatial distribution of 100 accepted models. The large signal-to-noise ratio leads to a very close fit.

as near-surface soil density variation, accurate location of the instrument and accuracy of terrain corrections will combine to increase the overall survey uncertainty. The extent to which these will affect a quantum atom interferometer gradiometer in the field is at present unknown.

Sphere Forward Model

We can now apply the sphere forward model to the synthetic gravity gradient data-set. Figure 5.25 shows that the much higher signal to noise ratio of the gravity gradient data allows the forward model to become more complex, in order to fit the data more accurately. We carried out five combined runs each of 200,000 iterations. Each run converged to three model objects. We have shown in this section the capabilities of the reversible-jump Markov chain Monte Carlo algorithm for classifying a near surface anomaly whose gravity signal is large enough to be detected by the Scintrex CG-5. We have partial confirmation of the synthetic model

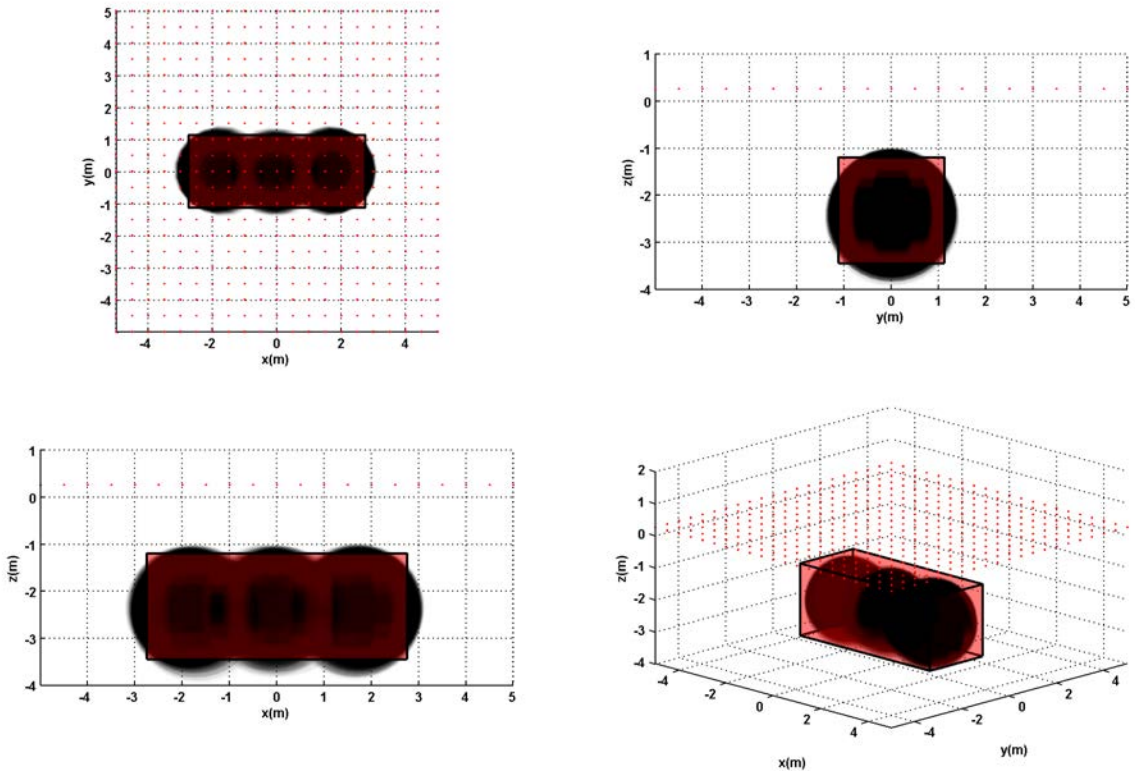


Figure 5.25: Spatial distribution of 100 accepted models. The large signal-to-noise ratio leads to a more complex model ($k = 3$).

from a line of data obtained over the Broadway Tower nuclear bunker as shown in Appendix A.2. We have seen parsimonious behaviour of the reversible-jump Markov chain Monte Carlo algorithm, converging consistently to the simplest model solution. The issues associated with assessing convergence have been addressed using the Geweke diagnostic. The parameter switching problem was sidestepped by summarising the total mass, M, and displaying the spatial information in 3D plots of the model objects. We investigated the difference in the inference outputs due to an increased and more realistic measurement survey spacing of 1 m. The corresponding gravity gradient signal of the void was analysed, which had a much larger signal to noise ratio than the gravity signal (48 to 17.4) . We saw how the increased signal strength allowed the model to become more complex in the case of the sphere ($k = 2 \rightarrow k = 3$) relative to the gravity data. Showing that the model complexity is determined by the data. Essentially a built in Ockham's Razor[108].

5.2 The 'Dog-leg' Pipeline Problem

The underground bunker modelled in the previous section was a simple anomaly, as it could be described by a single model object, assuming the object forward model was known a-priori. Now we tackle a problem where none of the simple shape models can describe the data using a single object. Such a problem could arise in civil engineering, with a curved or 'dog-leg' pipeline. The model used here is an approximation of a near surface gas pipeline. The cylinder model was used (Equation 2.11) to generate the model shown in Figure 5.26, with the corresponding gravity data shown in Figure 5.27. The cylinder properties were: $R = 0.6$ m,

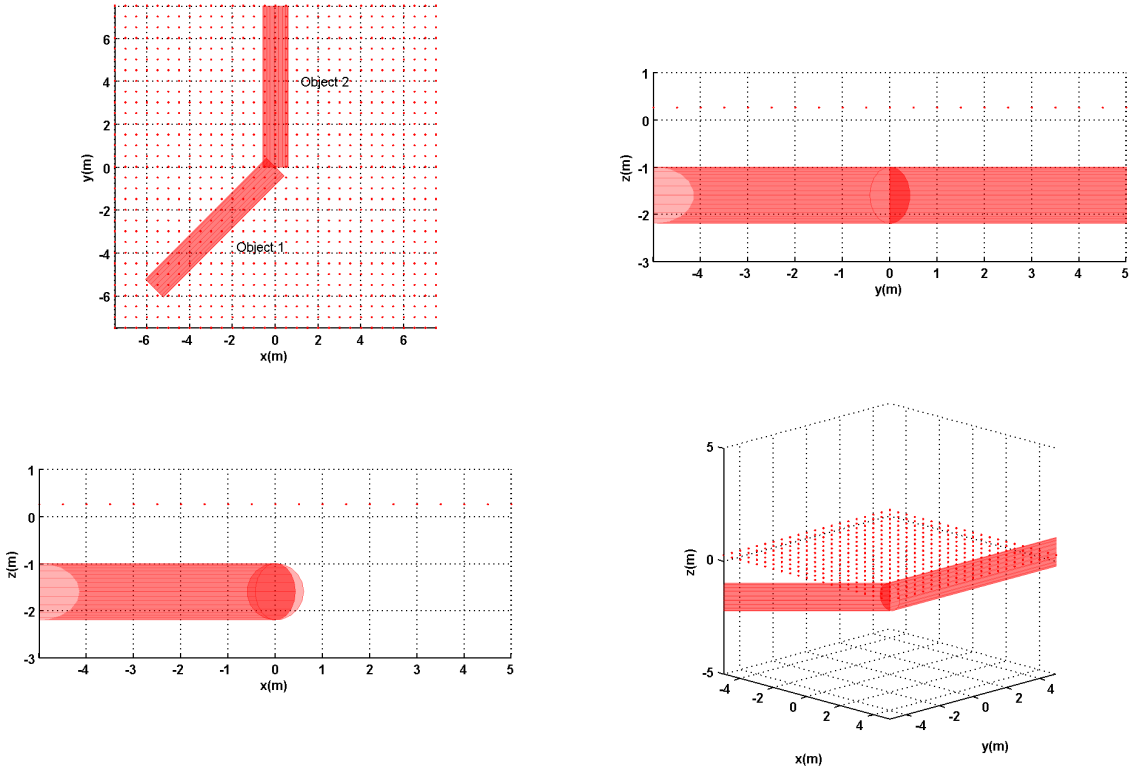


Figure 5.26: Model of curved pipeline. Consisting of two, effectively semi-infinite cylinders of radius 0.6 m, buried at a depth of 1 m to the anomaly top (1.6 m to centroid). Object one is rotated an angle of 45° with respect to object two.

$\Delta\rho = -2000$ kgm^{-3}. The measurement points are spaced by 0.5 m on a square 15×15 m grid. The measurement points are taken in a flat plane with $z = 0.25$ m. The signal-to-noise ratio for this anomaly was found to be 5.6. Significantly lower than the signal strength that we were dealing with in the previous section. A further complication with the data-set is the

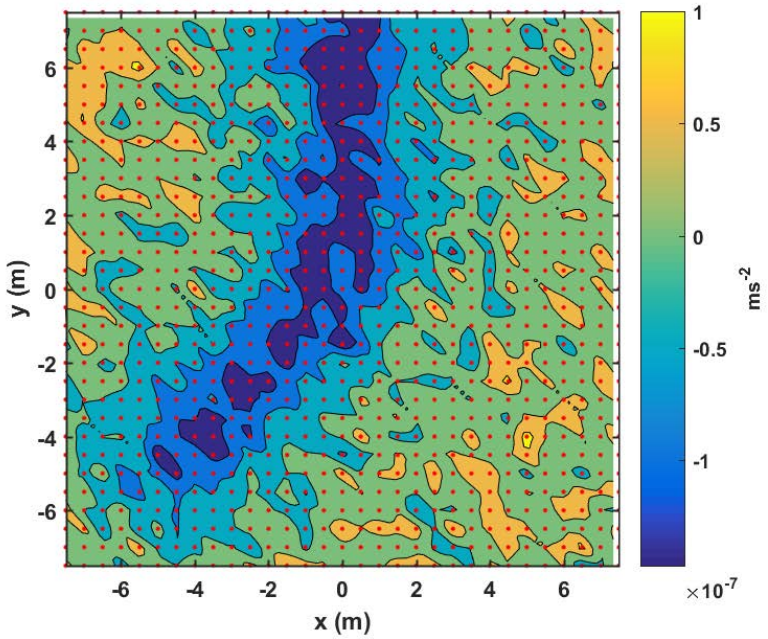


Figure 5.27: Synthetic gravity gradient data generated for the pipeline anomaly. Gaussian noise with $\sigma = 3 \times 10^{-8}$ ms^{-2} was added to the data.

semi-infinite nature of the anomaly. The anomaly is larger than our measurement grid, which may in general be the case for real world small scale gravity surveys. First, let us consider the case where we have prior information regarding the appropriate forward model to use in the reversible-jump Markov chain Monte Carlo algorithm, that of the finite cylinder.

5.2.1 Cylinder Forward Model

Defining our forward model from Equations 2.11 and 5.2, we proceed as before. Prior and proposal distributions are defined, see Figure 5.28 and Tables 5.6 and 5.7. We now have $7k + 2$ parameters defining our forward model. Similar reasoning is used for the assignment of prior distributions to previous problems. One difference here is that the length parameter ℓ_y is defined with a Gamma scale parameter of four. Essentially stating that we expect the anomalies to be relatively long in one direction, as is reasonable for a suspected pipeline.
Ten runs of the reversible-jump Markov chain Monte Carlo algorithm were carried out, of which 8 were deemed to have converged. Selecting a converged run at random, we can assess the output parameters. Figure 5.29 shows a random selection of one hundred accepted models

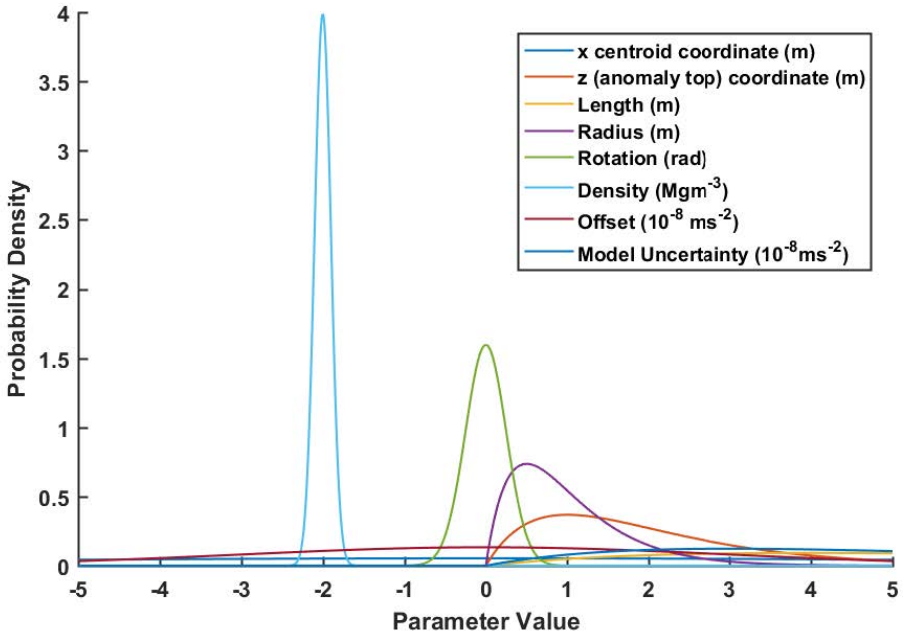


Figure 5.28: Prior probability distributions used for the cylinder forward model.

Gamma Prior Distribution

Parameter	Unit	a (shape)	b (scale)
z_0	m	2	1
R	m	2	0.5
ℓ_y	m	2	4
σ_m	10^{-8} ms^{-2}	2	$\langle \boldsymbol{\sigma_d} \rangle$

Gaussian Prior Distribution

Parameter	Unit	μ	σ
x_0	m	0	max($\boldsymbol{x}$)
y_0	m	0	max($\boldsymbol{y}$)
ψ	m	0	0.25
η	10^{-8} ms^{-2}	0	$\langle \boldsymbol{\sigma_d} \rangle$
$\Delta\rho$	Mgm^{-3}	-2	0.1

Uniform Prior Distribution

Parameter	Unit	Minimum	Maximum
k	Dimensionless	1	10

Table 5.6: List of prior distributions used for all model parameters.

of the run. Considering the signal-to-noise ratio of the data-set, the spatial fit seems reasonable. We can see that there is some ambiguity around the meeting point of the two objects. So we expect the two objects to have correlated parameters, as we encountered previously with the sphere model fit to the nuclear bunker anomaly. Figures 5.31 and 5.33 show the marginal posterior distributions of objects one and two respectively. For completeness Figure 5.34 shows the marginal posterior of the error and offset parameters. The parameters of object one, which is encompassed by the measurement grid, are all found within the marginal distribution credibility intervals. The credibility interval widths for the depth and radius are

x_0	y_0	z_0	R	ℓ_y	ψ	$\Delta\rho$	σ_m	η
0.05	0.05	0.1	0.05	0.1	0.05	0.1	0.05	0.05

Table 5.7: List of proposal distribution scaling parameters for the Cauchy distribution.

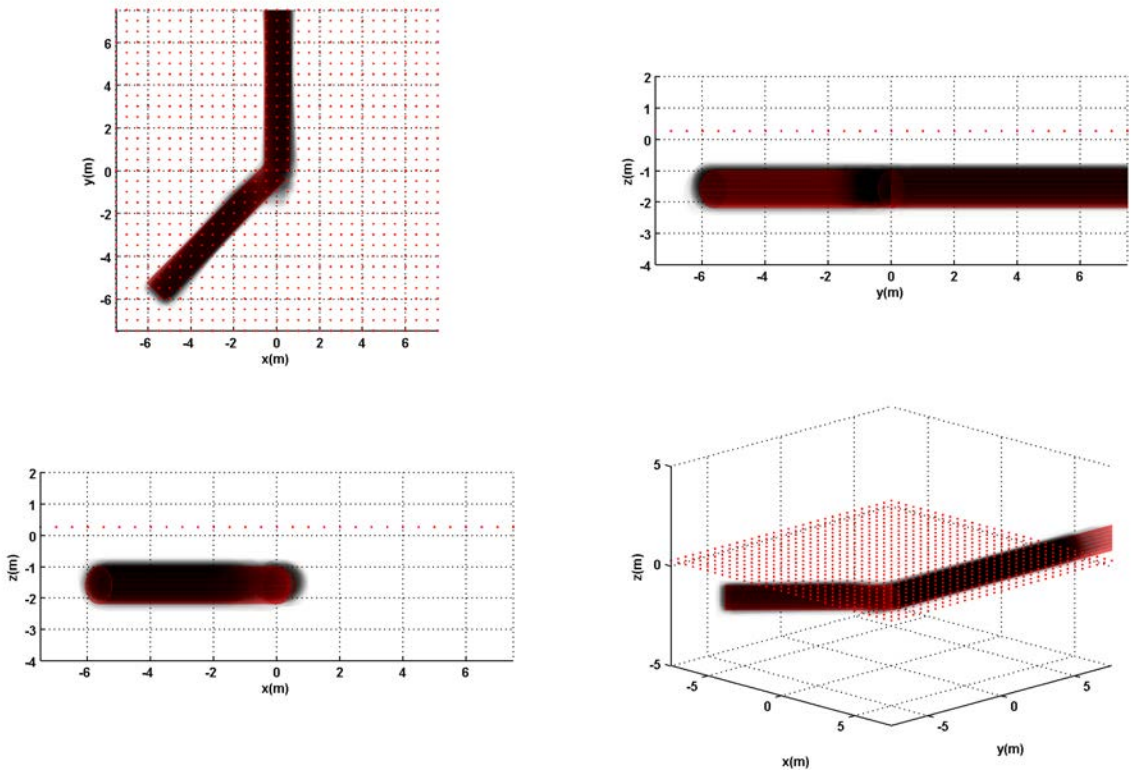


Figure 5.29: Converged run for dog-leg pipeline anomaly.

small, only 0.4 m and 0.13 m respectively. Object two, which is semi-infinite with respect to the measurement grid is well defined for all parameters except for y_0 and ℓ_y, where an infinite number of solutions exist. This ambiguity causes problems for the Geweke diagnostic, which relies on stationarity to assess convergence. Parameters that can undergo a random walk together whilst still fitting the data equally well have no stationary property. This problem of semi-infinite anomalies does not have a simple solution without raising other issues. We may consider a uniform distribution for our centroid values that equals the extent of the measurement grid, effectively forcing the object to be within the grid. This would solve the current problem, but would create other issues by limiting the model space. Anomalies that lie just outside of the measurement grid could not be investigated even though their gravity signal could be non-negligible.

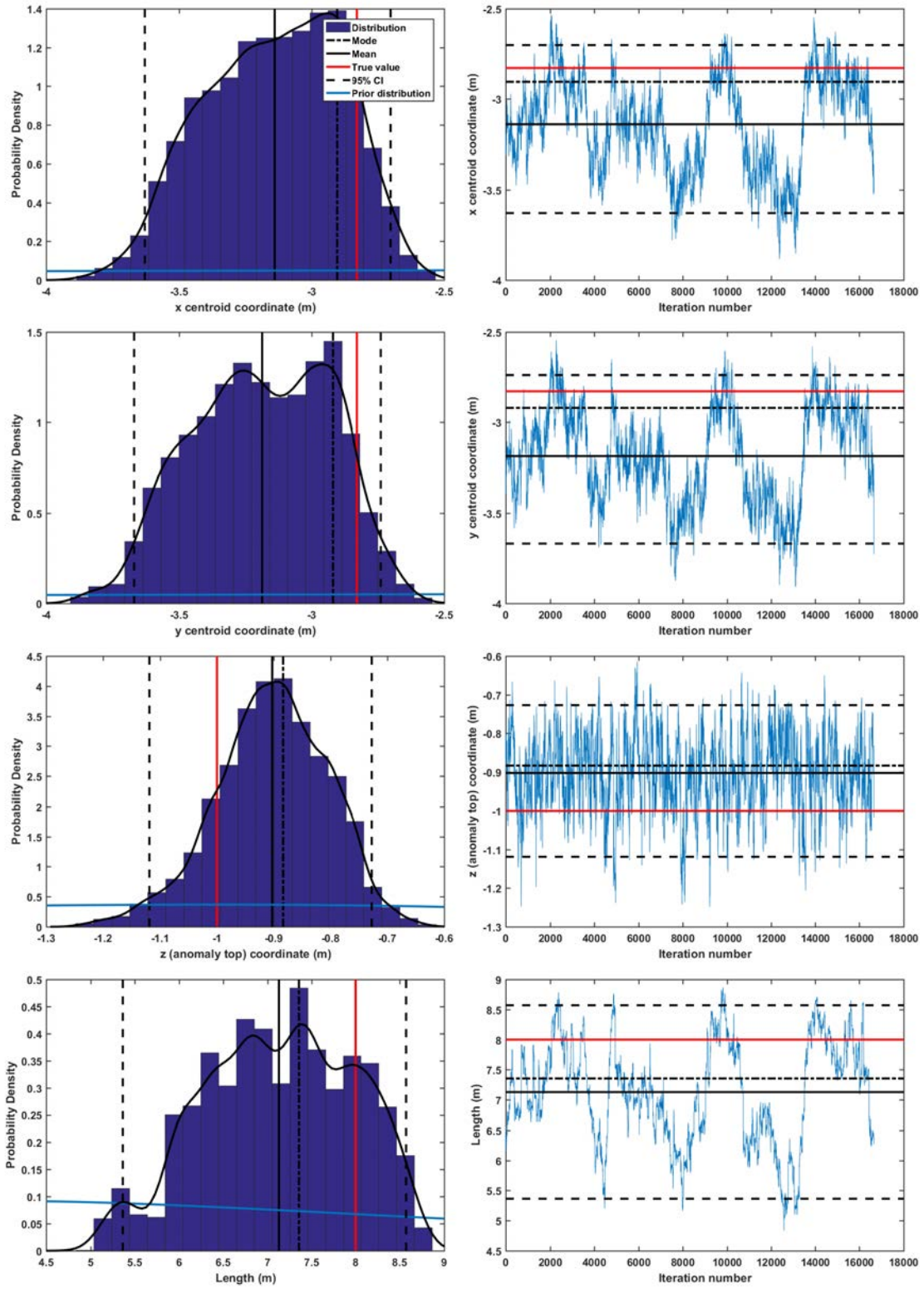


Figure 5.30: Marginal posterior distributions for object one. Histogram and trace plots of parameters x_0, y_0, z_0 and ℓ_y are shown.

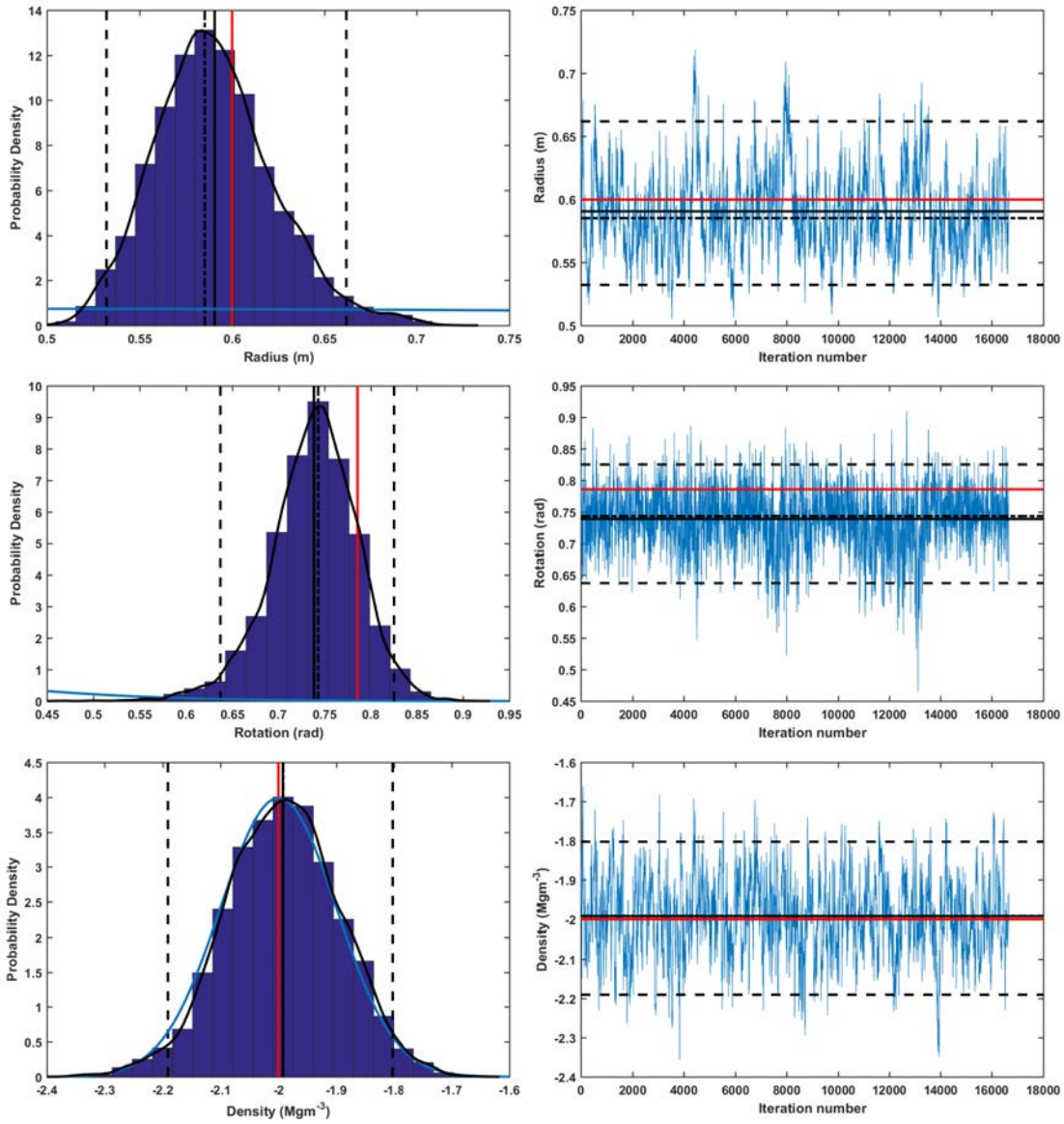


Figure 5.31: Histogram and trace plots of object one parameters: R, ψ and $\Delta\rho$.

Figure 5.35 shows the combined k value for the eight converged runs. We see that the algorithm again converges to the simplest solution to describe the measurements within the uncertainty. As before, we combine the total mass of the multiple runs. Figure 5.36 shows that with a semi-infinite anomaly we can no longer rely on the total mass to be a descriptive distribution. The ambiguity due to the object length means that there is also ambiguity in the total mass. Further to this, with a semi-infinite object we cannot say exactly what the total mass parameter should be, as we do not know the total anomaly volume affecting our gravity measurement grid. As with the bunker anomaly, we can also approach the problem

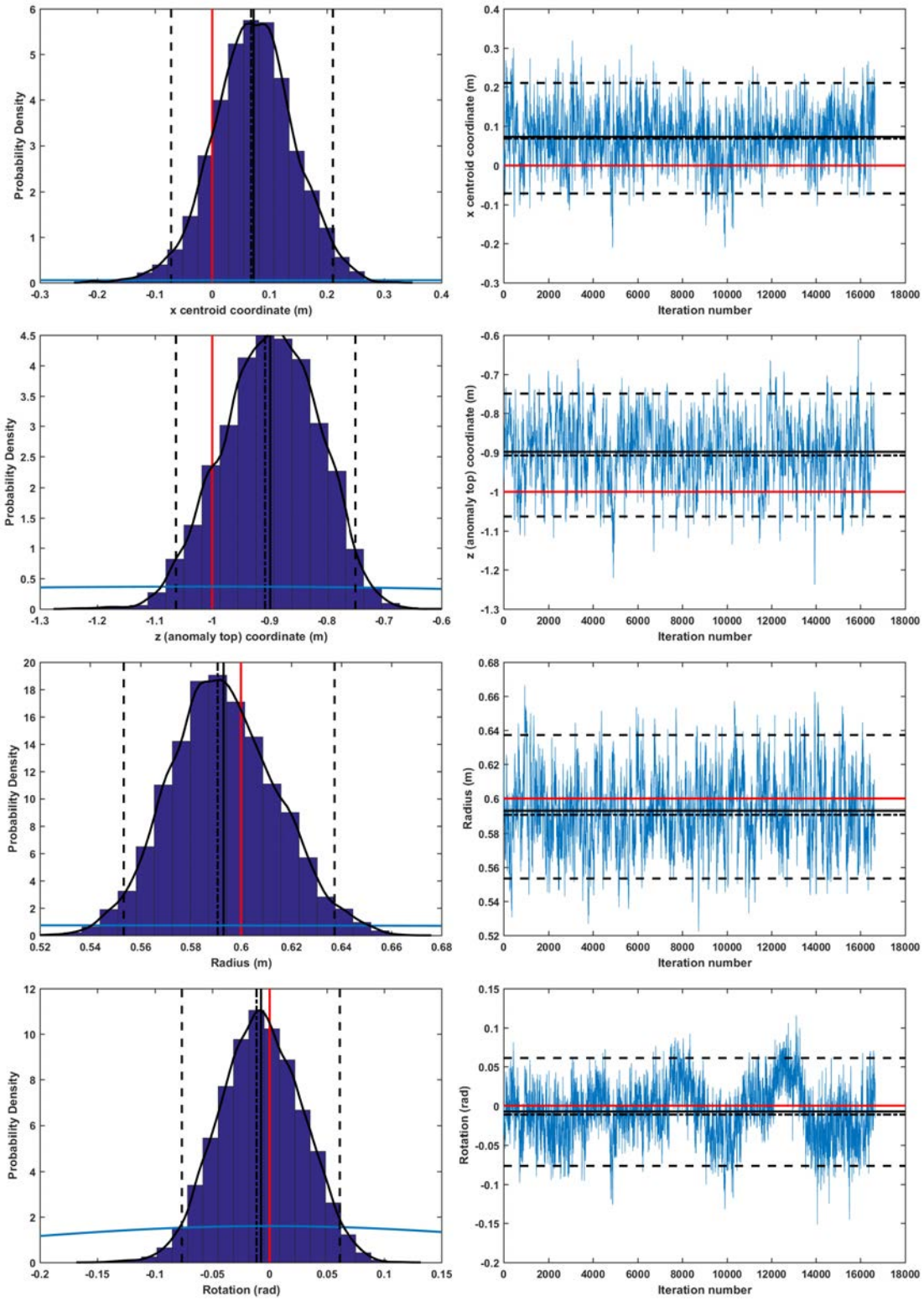


Figure 5.32: Marginal posterior distributions for object two. Object parameters x_0, z_0, R and ψ are shown. The y_0 parameter is omitted as it has no correct value due to the semi-infinite anomaly.

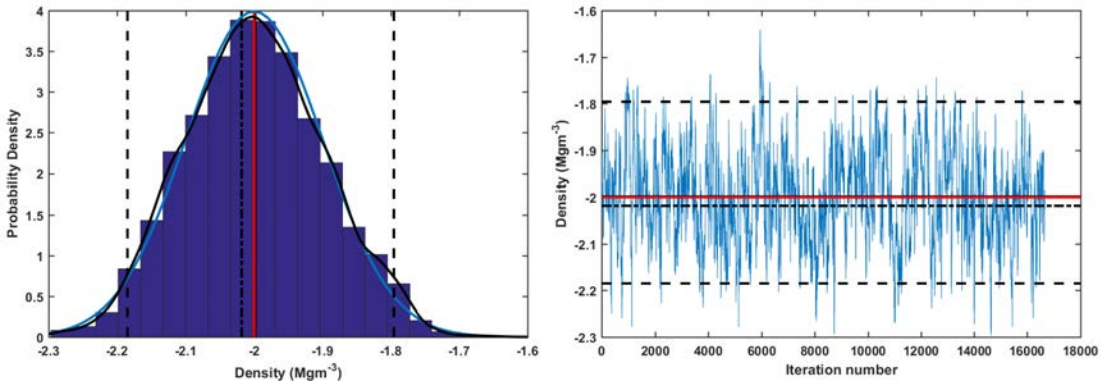


Figure 5.33: Marginal posterior distributions for object two continued. The ℓ_y parameter is omitted as it has no correct value due to the semi-infinite anomaly.

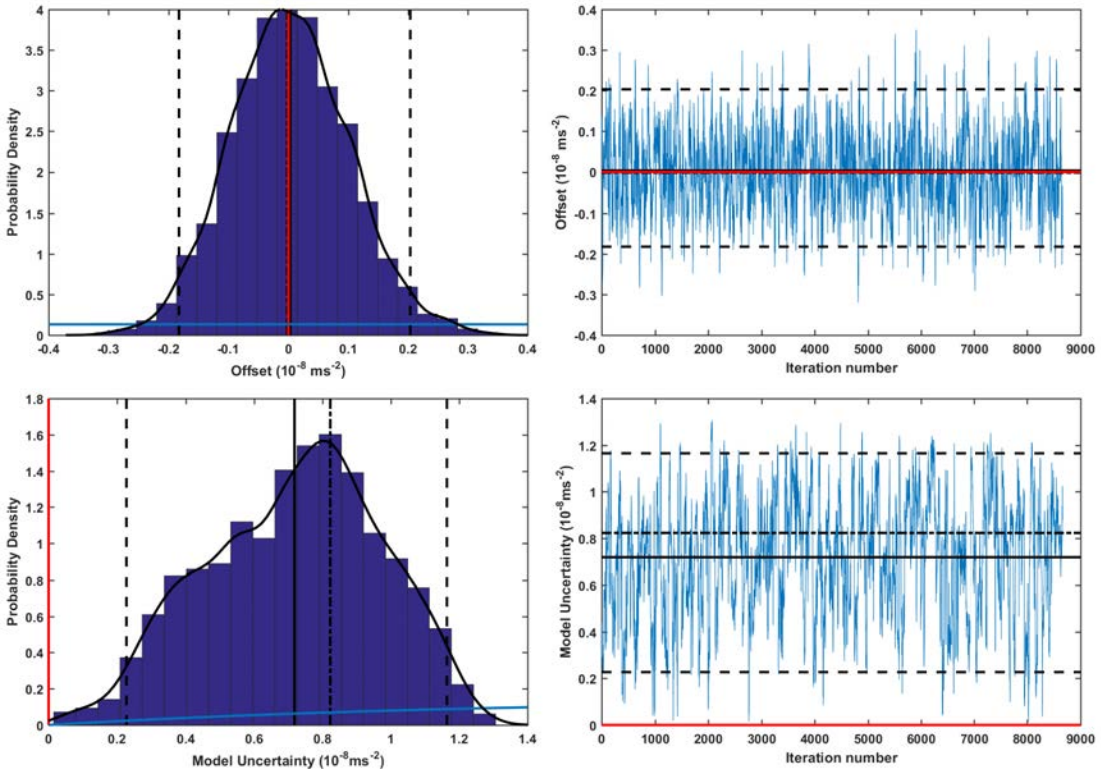


Figure 5.34: Marginal posterior distributions for the global parameters: σ_m (model uncertainty) and η (gravity offset).

from a state of increased ignorance by using the sphere forward model. However, we may beforehand expect a less than satisfactory fit, as the limits of the sphere approximation (see Appendix B) are violated by this anomaly.

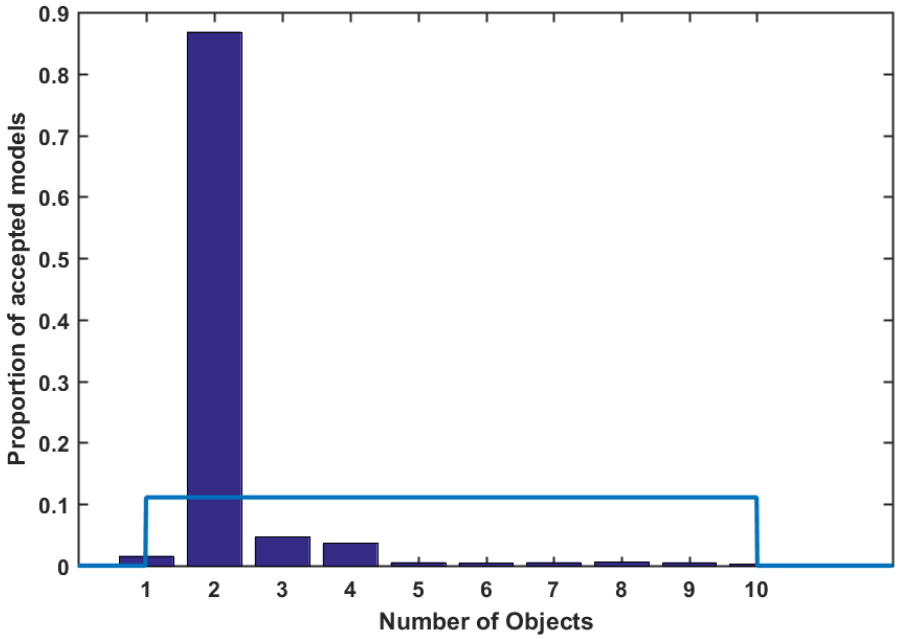


Figure 5.35: k parameter for the eight converged reversible-jump Markov chain Monte Carlo runs.

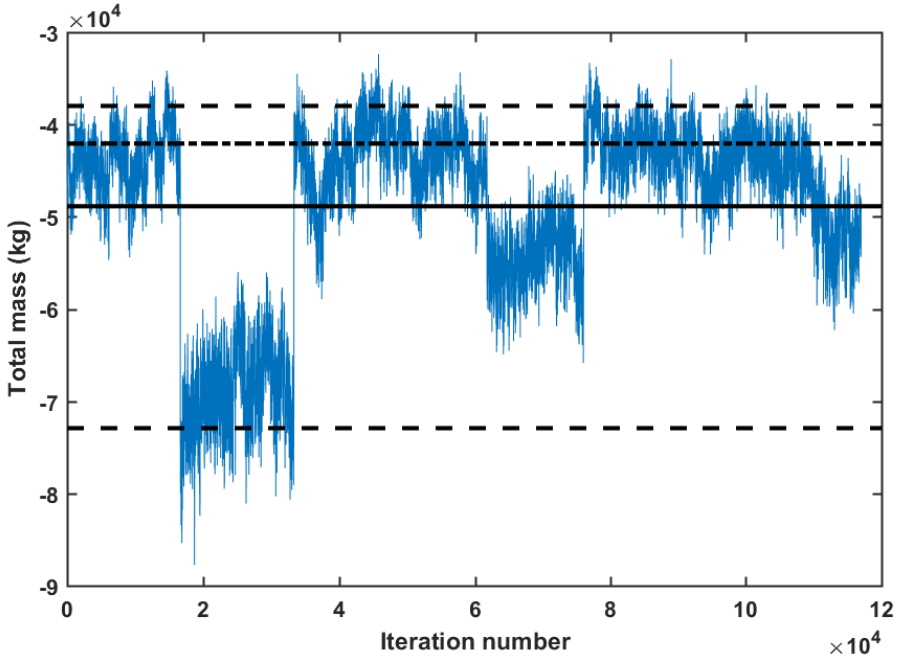


Figure 5.36: Total mass of the eight converged reversible-jump Markov chain Monte Carlo runs

5.2.2 Sphere Forward Model

We apply the sphere forward model to the dog-leg pipeline data-set described in the previous section. The prior distributions and initial Cauchy proposal distribution widths are set the

same as those used in Section 5.1.2. Appendix B explains the approximation of spheres to fit a cylindrical anomaly.

Ten runs (of 200,000 iterations each) of the reversible-jump Markov chain Monte Carlo algorithm were completed, of which nine passed the convergence test. Figure 5.37 shows the combined k value of the nine converged runs. Four objects are clearly favoured. Figure 5.38 shows the corresponding spatial distribution of the four objects. Starting from an initial

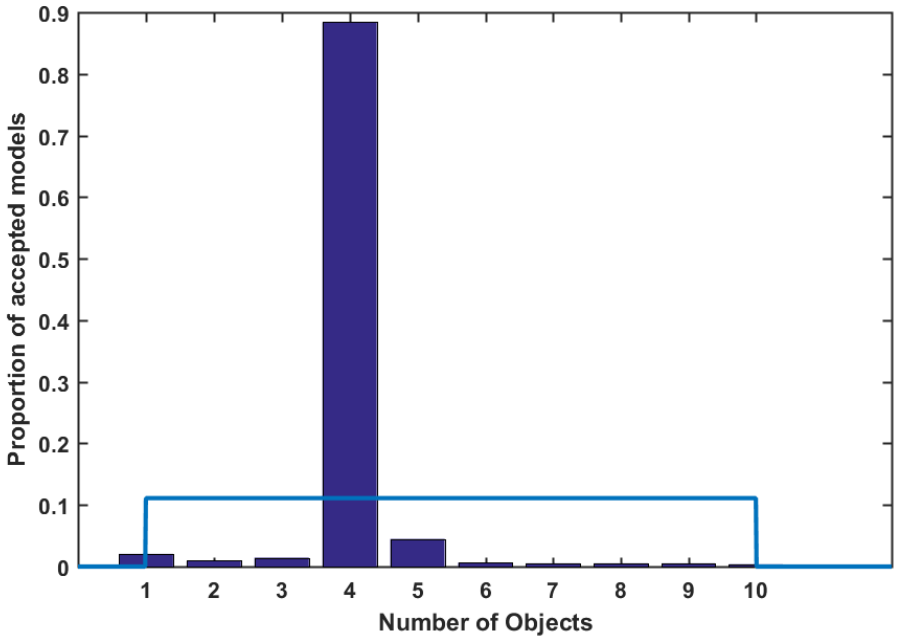


Figure 5.37: Number of objects for the nine converged runs.

situation of ignorance regarding the forward model, we should expect less information in our Bayesian posterior distribution. The sphere model requires a total of twenty two parameters to fit the gravity data-set within the total uncertainty; compared to sixteen parameters of the cylinder model. The x_0, y_0 values of the spheres follow the cylinder centroid well. The radii of the spheres is not directly comparable to the radii of the cylinders. However, we would expect the depth of the sphere to correspond with the depth of the cylinders. Figure 5.39 shows the marginal posterior distribution of the depth of one of the four sphere model objects. The model parameters that are independent of k are shown in Figure 5.40. We see that due to the discrete nature of the sphere anomaly, the total mass posterior distribution once again converges to a stationary distribution for separate runs, as we no longer have the

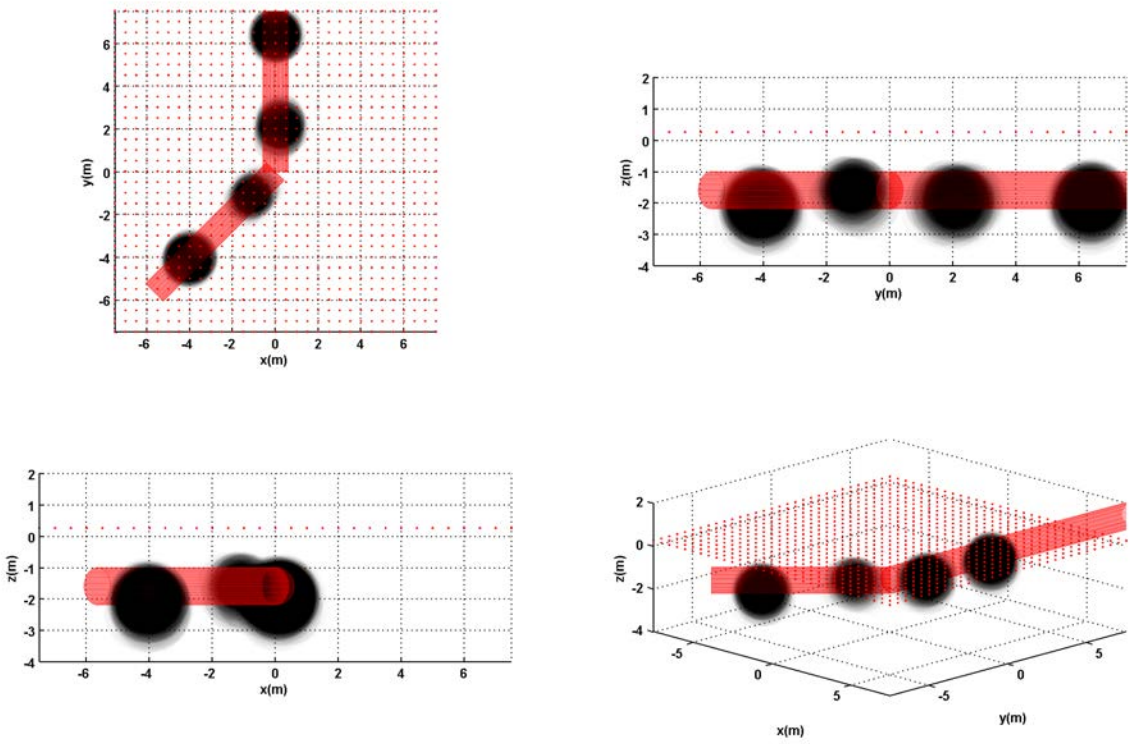


Figure 5.38: Converged combined runs for dog-leg pipeline anomaly fit using the sphere forward model.

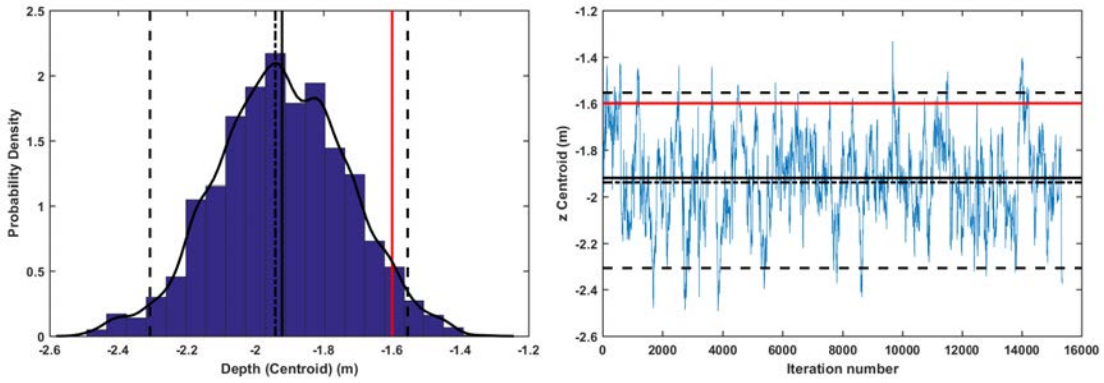


Figure 5.39: Centroid depth of one of the spheres.

problem of model parameter degeneracy that we encountered with the cylinder model. This can be seen as an advantage of using a simple sphere forward model to fit a gravity data-set of a complex anomaly.

The drawback to using the sphere model is clear, the user is left to interpret the spatial distribution further. It is not clear if we are dealing with four independent anomalies or if the discrete anomalies can be linked into a large continuous anomaly. The only way to obtain

more information from our Bayesian posterior distribution is to incorporate more specific prior information (as is provided by using the cylinder model) or using the gravity data-set as part of a larger joint-inference algorithm, combining inference of multiple geophysical data-sets simultaneously[109].

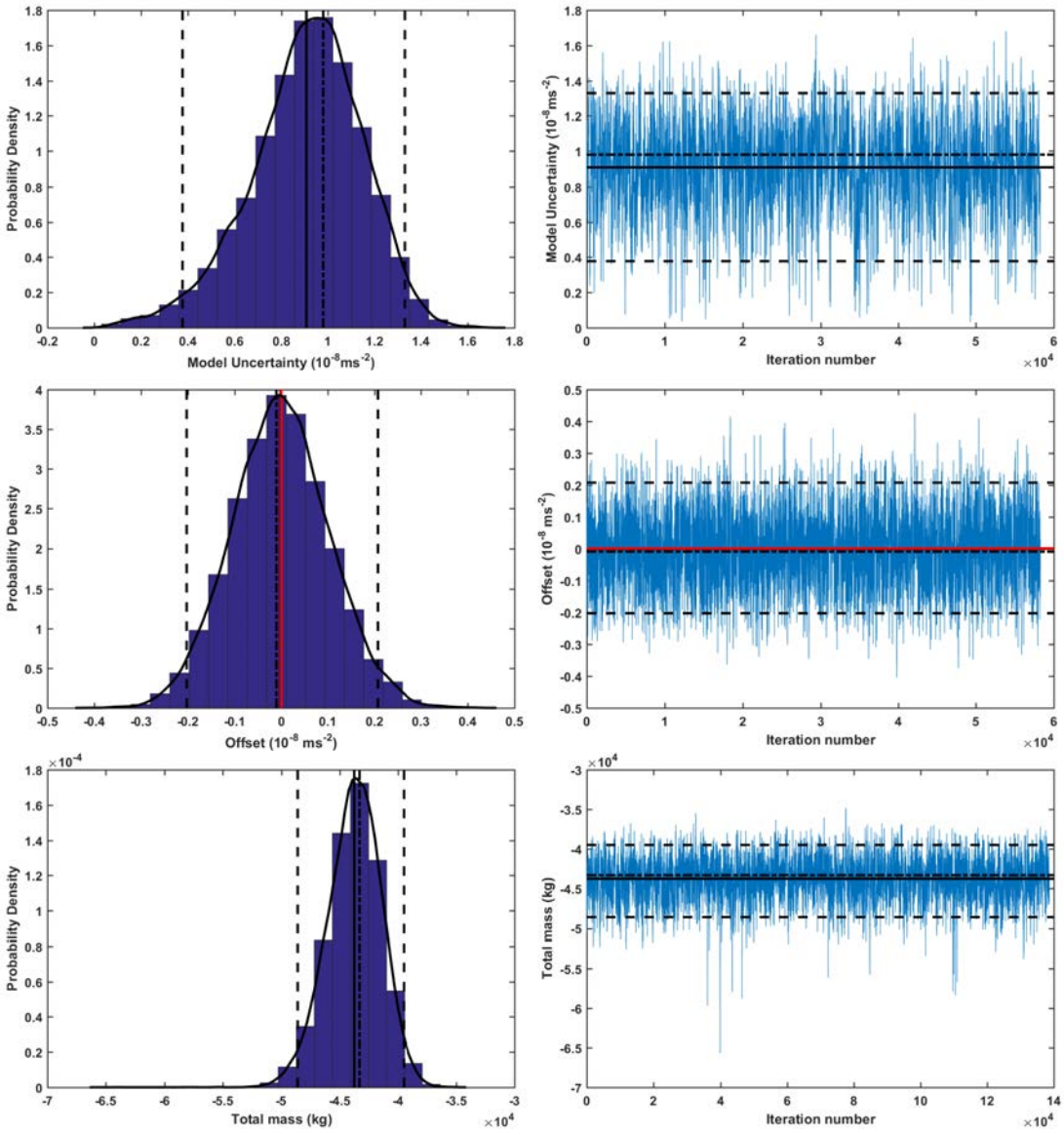


Figure 5.40: Marginal posterior distributions for the global parameters, model uncertainty offset and total mass.

5.2.3 Inference from Synthetic Gravity Gradient Data

We briefly discuss the reversible-jump Markov chain Monte Carlo algorithm outputs for synthetic U_{zz} measurements of the dog-leg pipeline anomaly. Figure 5.41 shows the U_{zz} synthetic gravity data. Using the same measurement positions as described in Section 5.1.4. The signal-to-noise ratio for this anomaly was found to be 18.4. For the cylinder model we

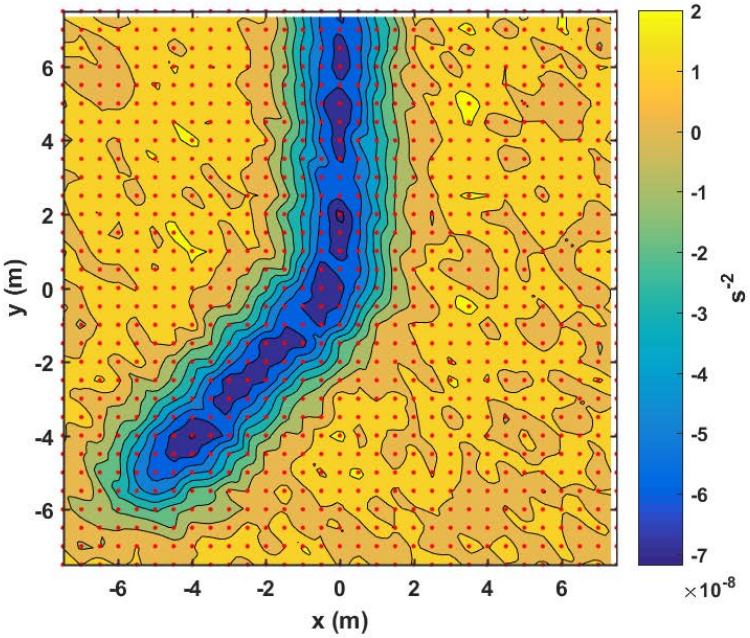


Figure 5.41: U_{zz} data for the dog-leg pipeline anomaly. Data-set consists of 441 measurement points, with added Gaussian noise with $\sigma_d = 0.5 \times 10^{-8}$ s^{-2}.

simply get back tighter marginal posterior distributions on the model parameters due to the increased signal-to-noise ratio. So we move straight to the sphere forward model. Figure 5.42 shows the converged spatial distribution of one reversible-jump Markov chain Monte Carlo run of 500,000 iterations. More iterations are required to reach convergence due to the increased number of model objects. We see that the complexity of the model has effectively doubled, with a total of eight spheres combining to best fit the gravity gradient data-set. As a consequence, the sphere centroid depth more closely approximates that of the cylinder anomaly. Figure 5.43 shows the total mass of the model objects.

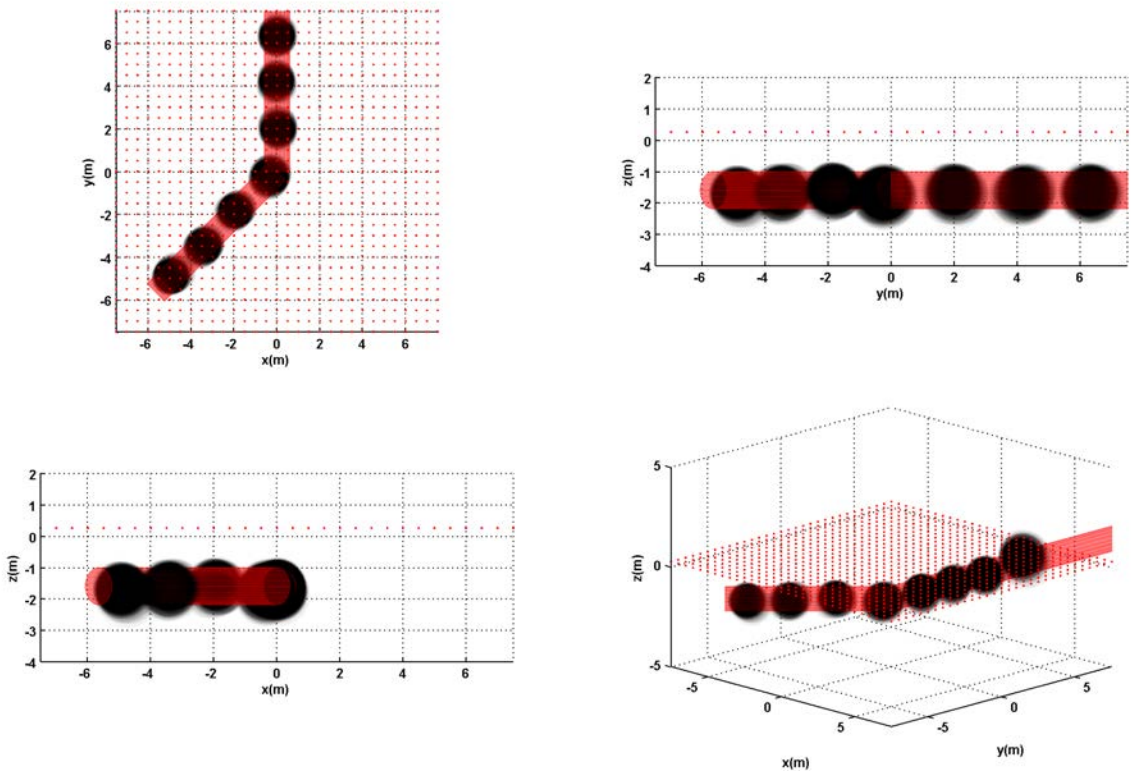


Figure 5.42: Spatial plot of the eight sphere objects. Double the amount of objects used to fit the lower signal-to-noise gravity data.

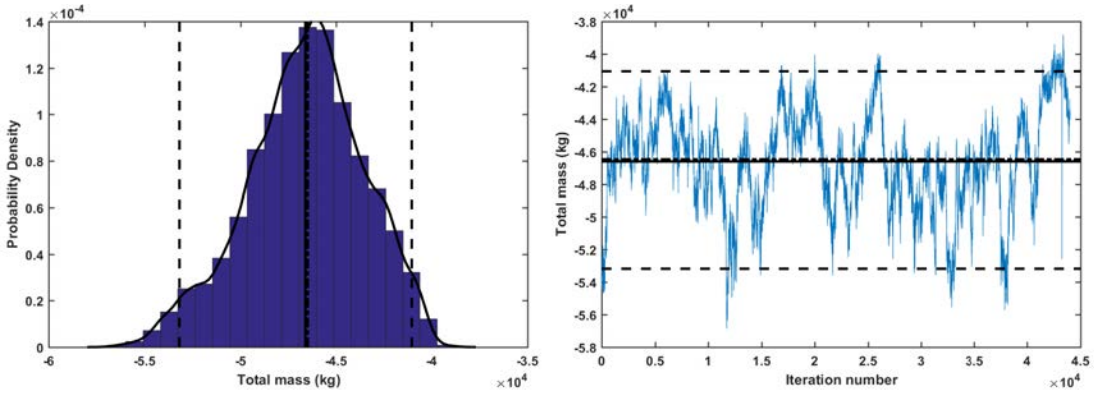


Figure 5.43: The total mass for one run of 500,000 iterations. The margnal distribution is more unstable with the large number of objects. Perhaps more iterations are needed.

Summary

In this section we have seen the application of the reversible-jump Markov chain Monte Carlo algorithm to two synthetic noise corrupted data-sets. The simple void data-set was partially confirmed by experiment (appendix A.2) and the dog-leg gas pipeline was simply a reasonable civil engineering application. We have shown that given sufficient prior information regarding

the average ground density and forward model type, we can remove sufficient ambiguity in the interpretation to achieve good agreement with the synthetic model truth. We demonstrate the usefulness of the birth-death moves to find the appropriate model complexity given the total measurement uncertainty. The three dimensional spatial plots show an intuitive method of displaying the algorithm outputs, combined with the total mass parameter which allows comparison between models of varying complexity. The synthetic gravity gradient data for the two anomalies provided a much larger signal to noise ratio than the corresponding gravity data, as as such the complexity of the model solutions for both cases were increased. The multiple sphere forward model was shown to be a good approximation to both synthetic anomalies investigated.

Chapter 6

Where are the concrete blocks?

In order to demonstrate the practical use of the Bayesian inference algorithm, an experiment to obtain gravity measurements of known anomalies was devised. Having knowledge of the anomalies present in the data allows for a direct assessment of the outputs of the Bayesian inference algorithm and thus its usefulness when applied to data with unknown anomalies. This chapter explains the experimental process of small scale terrestrial gravity measurements with a Scintrex CG-5 gravimeter, including survey method, data quality controls and data reduction techniques. After these techniques have been applied, the data is ready for input into the reversible-jump Markov chain Monte Carlo algorithm. The inference algorithm is run for a number of different measurement uncertainty values, with corresponding results discussed.

6.1 Experimental Set-up

The experimental set-up is shown in Figure 6.3. The location of the experiment was chosen for reasons of practicality, a rarely used car park close to the University offices, naturally sheltered from wind, footfall and road traffic. The car-park is located at 52.453965 N, -1.927668 W. Eight separate concrete blocks were stacked into two distinct gravity anomalies of four blocks each. Each block weighed 1600 ± 30 kg with dimensions of $2.3 \times 0.5 \times 0.7$ m. The two stacks of blocks were of dimension $2.3 \times 1 \times 1.4$ m. Using a scaffold tower adjusted

to the same height as the blocks, measurements were taken using a Scintrex CG-5 relative gravimeter in a grid above the two anomalies. The CG-5 was accurately located (±0.002 m

Figure 6.1: Photograph of the experimental set-up used. Showing the Scintrex CG-5 on top of the scaffold tower, with the Leica 360 degree prism on top, next to the two concrete anomalies.

in horizontal plane, ±0.01 m vertically) using a Leica TS-15 by logging the location of the 360 degree prism positioned above the CG-5 sensor location as shown in figure 6.2. The location is only logged once the instrument has been levelled to within the threshold of 10 arc seconds. The actual position of the concrete blocks was also mapped using the Leica station, so that it was possible to know where our measurement points were in relation to the anomalies. The measurement points relative to the concrete blocks are shown in Figure 6.3. We offset the Easting and Northing values to an arbitrary measurement point. Here, the middle point of the survey, 508, was chosen. A rotation of -27 degrees was applied to the measurement grid, making it approximately parallel to the xy axes. The height was scaled to the minimum height measurement. This offset and rotation process allows for easier integration into the reversible-jump Markov chain Monte Carlo algorithm. The original coordinates can be obtained by reversing the rotation and offset process.

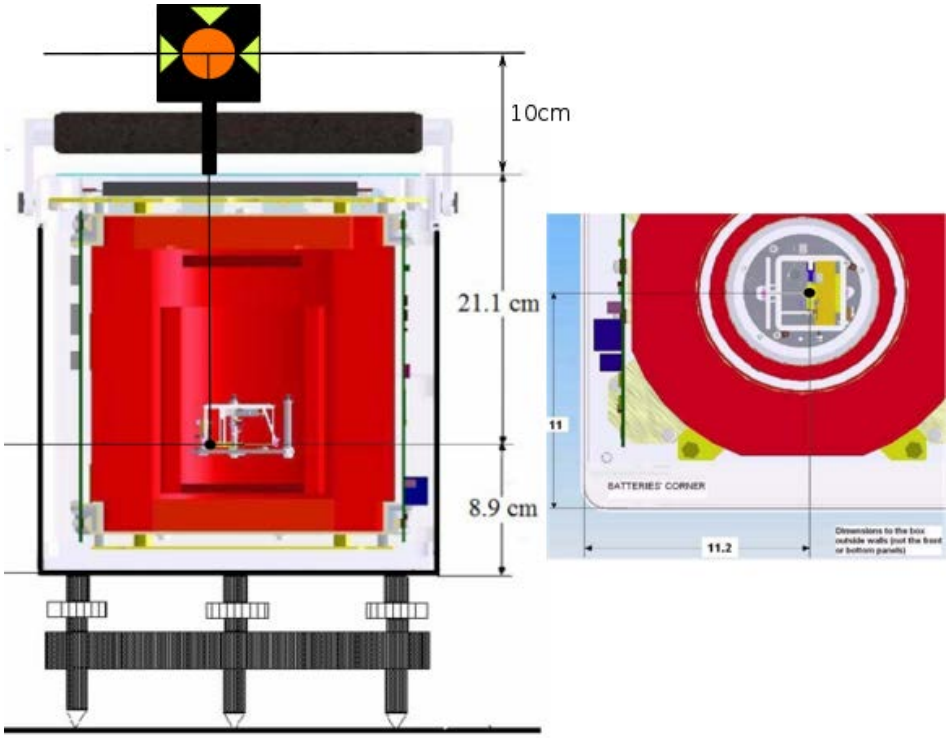


Figure 6.2: Location of the CG-5 sensor and Leica 360 degree prism. Adapted from [50] and [55].

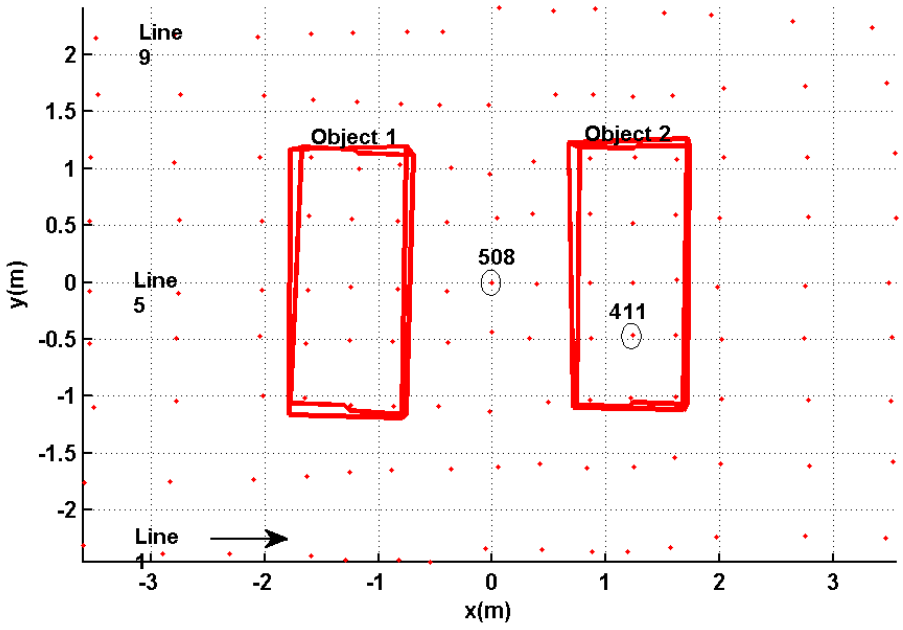


Figure 6.3: Schematic birds eye view of the concrete block set-up, with data lines shown along with the survey midpoint, point 508. Point 411 was chosen as a repeat point for each measurement day to check for consistency. The concrete block edges were found using the Leica total station, along with the measurement points (red dots).

6.1.1 Survey Procedure

As previously mentioned, the Scintrex CG-5 is often used for large scale terrestrial gravity measurements, looking for targets with a much larger gravity signal than the small near surface targets which are of interest here. To detect small anomalies we are pushing the CG-5 to its limits and as a result we require careful experimental procedure. Commercial companies such as RSK do carry out small scale surveys using the CG-5 in the UK, usually looking for sink-holes or other potentially dangerous gravity anomalies. These surveys can take many days or weeks to complete[110]. Most of the problems associated with terrestrial gravity measurements can be eliminated or limited by good survey practice[111]. Here we explain our general survey procedure, arrived at after many small scale surveys with the CG-5. To ensure the best chance of a successful survey, thorough planning is required before any gravity measurements take place. The survey site should be visited beforehand to attempt to foresee issues that may arise during the survey. Basic questions such as the location and extent of the measurement grid, and grid spacing should be addressed. The extent of a measurement grid is a trade off between available time and required accuracy. Horizontal spatial resolution of a survey is the minimum separation that permits recognition of nearby individual anomalies As such it is an important consideration as it determines the smallest anomalies that can be identified in a given survey. The horizontal spacing can be estimated beforehand using rough prior knowledge of the anomaly of interest.

A suitable location for the Leica base-station should be identified, ideally one which has line of sight to all required measurement points. The exposure of the site should be considered, as particularly windy conditions may require the use of wind defenders (see Appendix A.1). Finally, any instruments to be used on site should be recently calibrated as per their user manual instructions. After this initial planning phase, the measurement campaign can proceed.

The first day of measurements at a given site requires the Leica total station to be set up, with its position referenced to two GPS points, or control points. These control point locations are usually chosen to be orthogonal with respect to the total station location. Both the

total station and the GPS should be levelled on tripods, and their locations set by placing a survey peg in the ground directly below. On the first measurement day, the Leica Viva GS08 GPS logs the position of the control points over a number of hours. Once the initial set-up has been completed, the total station can be set-up relatively quickly between days by logging the positions of the control points[112].

When arriving on site, the Scintrex should be levelled and set measuring as soon as possible. This initial settling period usually lasted for approximately thirty minutes, during which time other survey tasks were carried out. This is required to ensure the instrument recovers sufficiently from the transportation and that it adjusts to the environmental conditions of the site. Whilst in this settling period, the Scintrex GPS attachment was used to log the site location and ensure that the internal clock was synchronised.

For each measurement, the instrument was placed carefully on the tripod and levelled using the tripod screws. A threshold of 10 arc sec (~ 50 μrad) is recommended[113] to achieve a measurement precision of 1×10^{-8} ms^{-2}. The Scintrex CG-5 screen outputs real-time information relating to the quality of a given measurement, which can be used as a quality control reference in the field. The xy tilt is displayed in arc seconds. When the instrument is initially levelled, the on screen tilt values should be monitored to ensure that there is no tilt drift from level. Often, tilt drift is unavoidable, especially on unstable ground. The best course of action in this situation is to re-level the instrument after each individual measurement. Care should be taken when stepping away from the instrument after levelling. For certain ground types the deformation under the operator's weight will cause the instrument to deviate from level. The Scintrex CG-5 allows a time to be set that delays the start of a measurement. This delay gives time to move a reasonable distance away from the instrument, to avoid imparting unwanted signals to the measurement. Ideally, if available the remote control should be used to initialise a measurement.

The measurement standard deviation is also displayed. This is highly dependent upon environmental conditions, large micro-seismic noise leads to large standard deviation values. There can be short spikes in micro-seismic noise which can cause measurements to have es-

pecially large standard deviations. At the end of a measurement or set of measurements the standard deviation values should be logged and extra measurements taken where spikes were identified.

The individual measurement time is somewhat dependent upon field conditions. The CG-5 allows a maximum single measurement time of 256 seconds. After some experimentation, it was decided that multiple shorter measurements were often the most efficient to collect. At each measurement point we aim to collect five-one minute measurements that fall within the quality controls defined for a given survey. The gravity value for each point is then taken as the average of the measurements that meet the quality controls.

The Scintrex CG-5 can suffer from what is known as a 'tare'. This is where the magnitude of the gravity reading changes abruptly for reasons that are not obvious. We speculate that abnormally large shocks when placing the instrument on the tripod or vigorous movement between measurement points could make tares more likely. Gentle handling of the instrument at all times may help with tare prevention. To detect a tare in the field, a set of measurements need to be observed and the mean values logged. An abnormal mean value with respect to the rest of the measurement set is usually a sign of a tare.

As the instrument suffers from mechanical drift, a base-station needs to be set up on site on which multiple measurements are taken throughout the day. The base-station can be an arbitrary point, however a few practical considerations can help improve the speed of a survey. The base-station should be placed in a quiet area, where the ground is stable and levelling the instrument will be a quick task. Ideally the location should not be too far from the measurement grid, to avoid unnecessary transportation of the instrument over large distances. The Scintrex CG-5 claims a linear drift coefficient, even so, here we opted to take base-station readings approximately every hour[41].

Once the gravity measurements have been collected as detailed above. The data are corrected as outlined in section 2.4 for each individual survey day. We can then combine the measurement points from different days and proceed with analysis of our survey uncertainty, as outlined in the next section.

6.1.2 Measurement Consistency and Uncertainty

Data point uncertainty is input into the Bayesian inference through the covariance matrix in the likelihood function (see Equation 3.5). Determination of the Scintrex CG-5 uncertainty is a non-trivial exercise, especially within the small measurement survey framework[111]. A small scale survey carried out over multiple days brings in areas of uncertainty that cannot be effectively determined by only accounting for the scatter of five separate measurements on each measurement point. Effects such as regular movement of the instrument between measurement points and location uncertainty and repeatability need to be taken into account when considering what number should be placed on the measurement uncertainty. In this experiment we make use of the base-station measurements to determine both the instrument drift and the measurement consistency. On each survey day the base-station is visited multiple times in order to allow the removal of the instrument drift in post-processing. If we had a noise-free instrument with perfect data reduction and experimental practice, we would expect our corrected base-station values to all be equally zero. In reality we have some distribution about zero as shown in Figure 6.4. The histogram shows base-station readings taken on the same day as measurement line five (see Figure 6.3). Although the number of base-station readings is small, we approximate a Gaussian distribution. We plot both the CG-5 corrected data and our own corrections of the raw analogue-to-digital converter (ADC) data values. We can see that they give very similar results. The base-station measurement standard deviation (σ_d) on a given day is taken as a base level of data uncertainty, as it is unreasonable to assume that any of our actual measurement points could possibly be known with more certainty than our base-station for which we have multiple measurements taken at different times. Figure 6.5 shows the mid line of gravity measurements with the calculated base-station standard deviation uncertainty of $\pm 2\sigma_d$. The raw corrected data and Scintrex corrected data are the same within the given uncertainty. We can clearly see the gravity signal due to the two concrete structures. There is some overlap of the signal in-between the blocks, as the signal does not drop to the background values at the start and end of the measurement line. The purple line shows the expected gravity signal with mean removed,

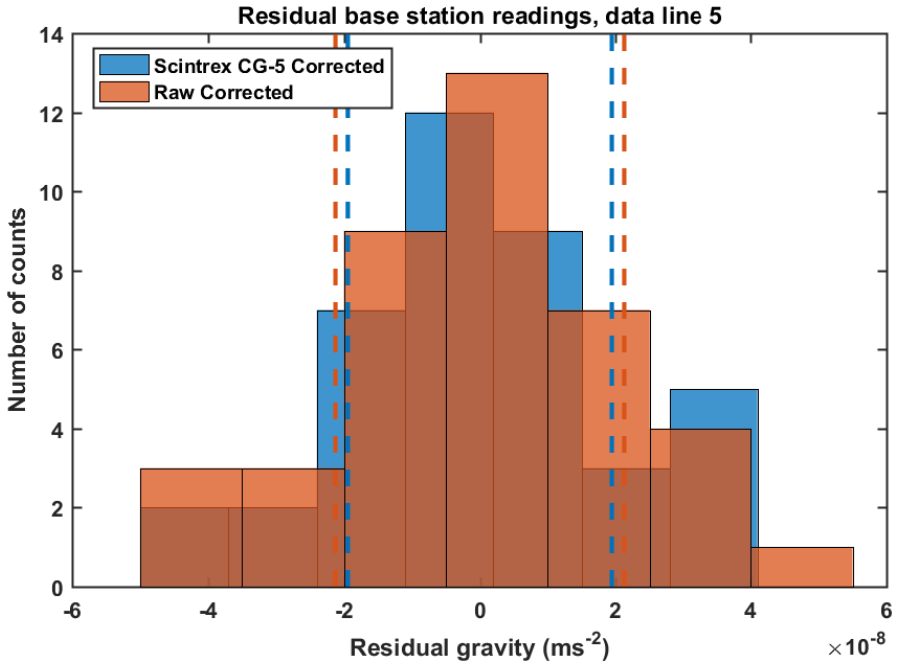


Figure 6.4: Histogram of the base-station measurement residuals. The measurements approximate to a normal distribution with a standard deviation of 2×10^{-8} ms^{-2}. Both the Scintrex corrected data and raw corrected data values are shown.

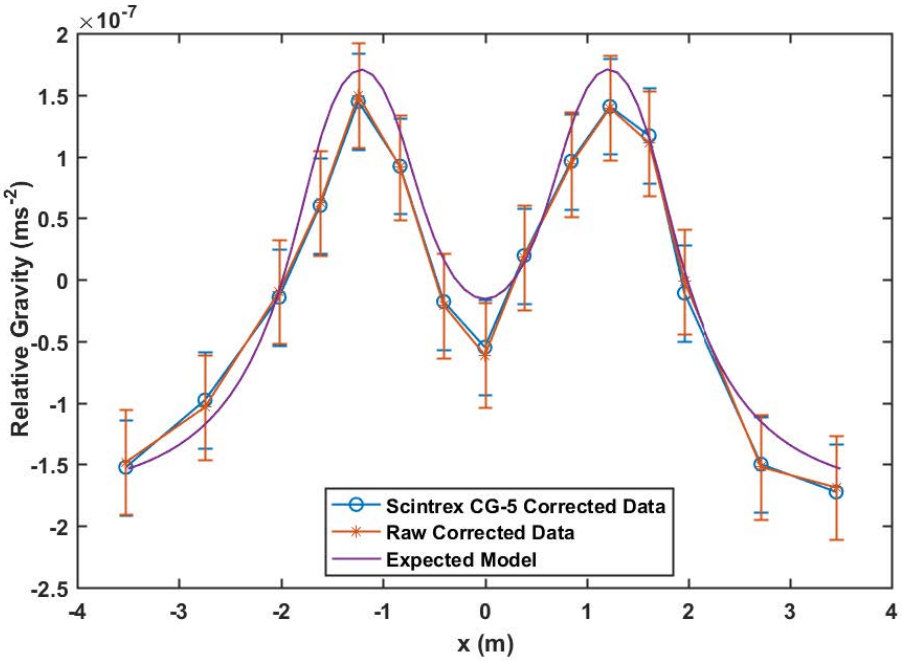


Figure 6.5: Line 5 (see Figure 6.3) of the gravity grid data with mean removed. Uncertainty $\pm 2\sigma_d$, from Figure 6.4. The purple line shows the expected gravity signal from the concrete anomalies.

based on our measurement locations and the known density and geometrical properties of the concrete structures. A regional linear trend was fit and removed from the data-set to

give the flat background shown.

The measurement grid was taken over multiple days, with point 411 being repeated on each day to check for consistency. Figure 6.6 shows the residual gravity values with the associated base-station uncertainty calculated as for Figure 6.4. All measurements of point 411 made on different days match within the $\pm 2\sigma_d$ uncertainty. Which demonstrates that the methodology of using the base-station reading standard deviation for a given survey day as the blanket measurement uncertainty is reasonable. It also shows that our data taken between days are comparable and can be combined into a larger, grid data-set. We see that data lines; two, seven and eight have large uncertainty values using this method. These were days where environmental conditions were poor, with gusts of wind causing vibrations of the instrument and adversely affecting tilt. Also, rain causing the ground to become unstable, making levelling of the instrument difficult. The difference in uncertainty values between the raw corrected

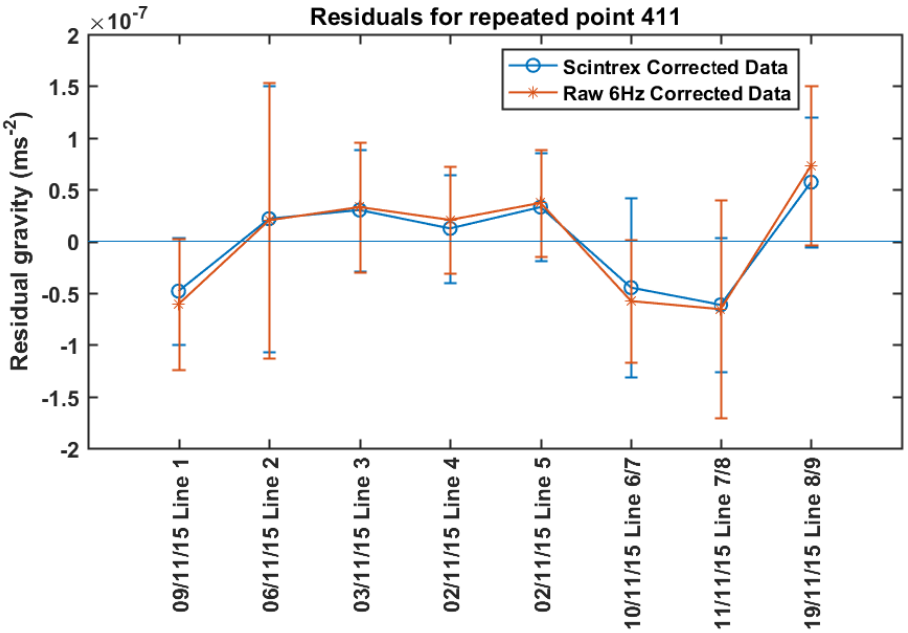


Figure 6.6: Point 411 was repeated on each measurement day in order to check for consistency between days. Shown are the residual corrected gravity values with two times the base-station standard deviation (σ_d) uncertainty. All measurement points agree within the uncertainty bounds. The size of the uncertainty bounds vary, due to discrepancies in the correction procedures for raw and Scintrex CG-5 corrected data.

data and the Scintrex corrected data may be caused by the Scintrex using a box car seismic noise reduction filter[55]. The exact workings of this process (and many others relating to the

Scintrex) are not explained in the CG-5 manual, hence the reason we opted to correct the raw data manually. An alternative characterisation of the survey uncertainty is to combine all of the base-station corrected values for each survey day into one distribution as shown in Figure 6.7. This yields a standard deviation value of $\sim 3.7 \times 10^{-8}$ ms^{-2}. This value gives an overall survey uncertainty, similar to the blanket uncertainty values that we assumed for synthetic anomalies investigated in Chapter 5. These uncertainty values match well with

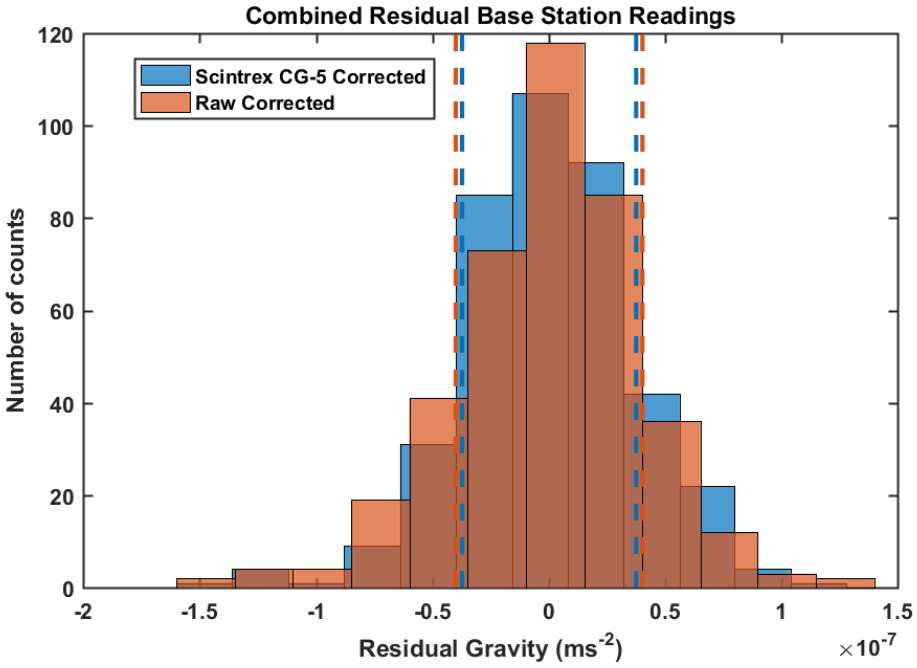


Figure 6.7: A combination of all base-station readings from all survey days. The larger amount of measurements provides a better approximation to a Gaussian distribution. The standard deviation is approximately 3.7×10^{-8} ms^{-2} for both the raw and Scintrex corrected data.

other gravity surveys carried out using the Scintrex CG-5 ($< 5\ \mu$ Gal repeatability)[110]. The CG-5 operation manual gives the uncertainty on a given measurement as the standard deviation divided by the square root of the measurement duration in seconds[55],

$$ERR = \frac{STD}{\sqrt{DUR}} \tag{6.1}$$

This could be used as an alternative data uncertainty estimate although it does not account for complications arising due to frequent movement of the instrument, as the above base-

station methods do.

6.2 Experimental Results

The final correction to our gravity measurement values is to remove any regional linear trend present in the data-set. Assumed here to be a flat plane so as to avoid fitting any anomalies present. The gravity values are fit using the *MATLAB polyfit* (which gives the best fit in a least-squares sense) function and the resulting fitted function is evaluated at the measurement point positions and subtracted from the gravity data. Figure 6.8 shows the regional trend that was subtracted from the corrected gravity data. The assumption of a simple flat regional trend may have to be examined more closely for gravity surveys of larger spatial extent or surveys that are close to buildings or other sources of potential distortions in the gravity data. The regional trend across the site was substantial, 30×10^{-8} ms^{-2} over a dis-

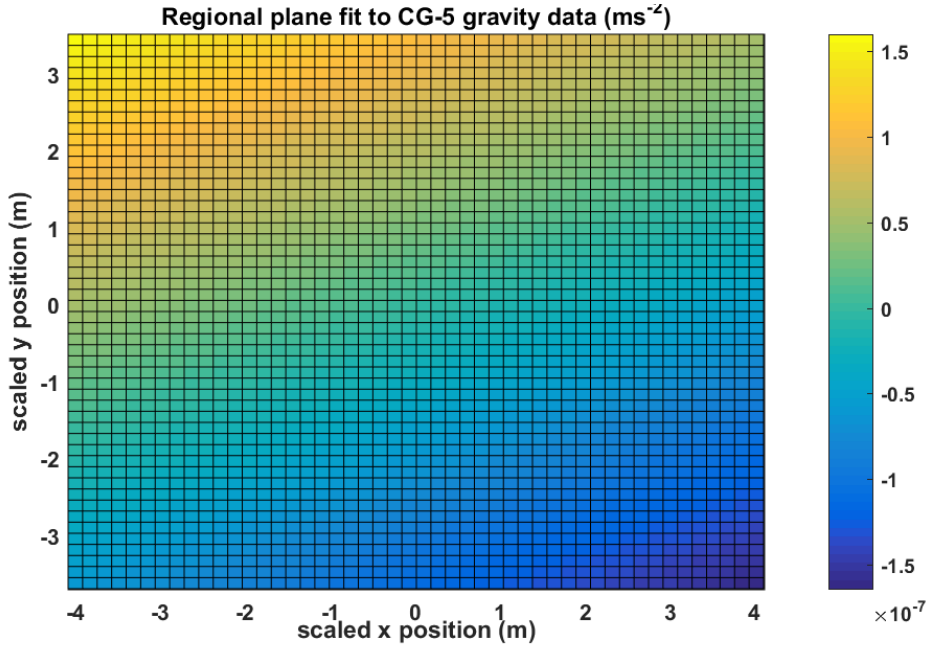


Figure 6.8: The regional gravity correction shows a trend of 30×10^{-8} ms^{-2} from NW to SE.

tance of just under ten metres. Without removing this beforehand or accounting for it in the modelling process the inference algorithm would give erroneous results.

Figure 6.9 shows the 134 final gravity values obtained, Figure 6.10 shows the corresponding

measurement uncertainties, which were taken as the standard deviation of the base-station readings for a given day as demonstrated in Figure 6.4. We can see that the concrete structures are visible in the data-set. The uncertainty contour shows different measurement uncertainty correlating to different measurement days. It is clear that line 5 provided the best conditions and therefore gave the smallest uncertainty value. Line two and six gave much larger uncertainty values.

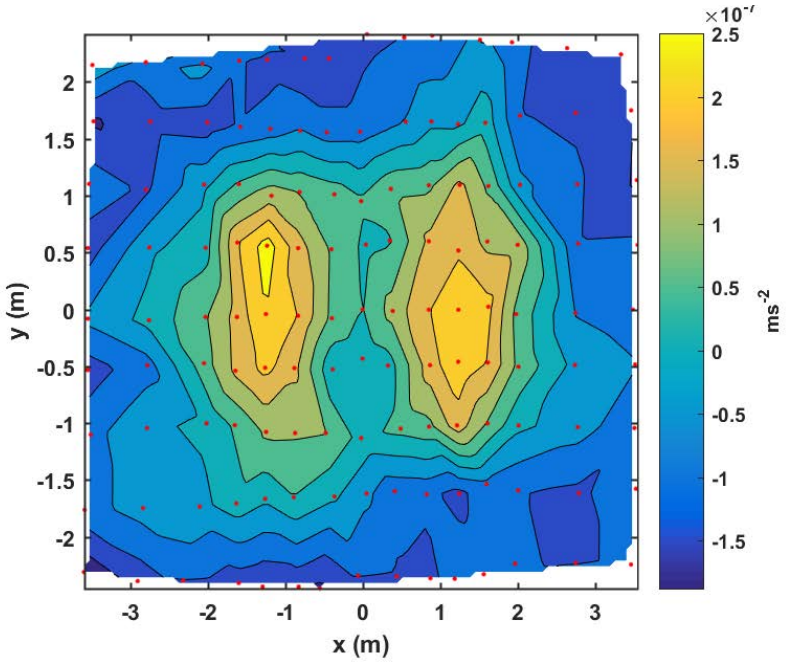


Figure 6.9: Contour plot of the 134 gravity measurements (red dots) taken above the two concrete structures. The regional linear trend has been removed.

6.3 Inference from Gravity Survey Data

Having obtained the final survey data-set, we can input the measurement: coordinates, gravity values and uncertainty values into the reversible-jump Markov chain Monte Carlo algorithm. Firstly we assume prior knowledge of the forward model, that of multiple cuboids, as used for the Nuclear Bunker synthetic data. We then investigate the effect of assigning different measurement uncertainty values, and removing the model uncertainty parameter from the forward model. Finally we investigate the applicability of the sphere forward model to the data-set.

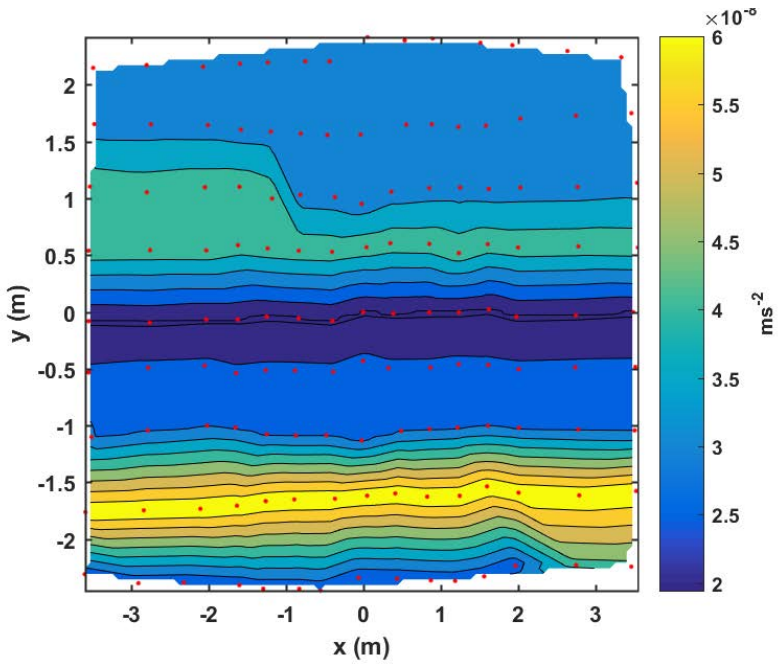


Figure 6.10: Standard deviation uncertainty (derived from base-station measurements) of the 134 gravity measurements (red dots). Linear trends show how the measurement uncertainty varies between survey days with environmental conditions.

6.3.1 Cuboid Forward Model

The data in Figure 6.9 with uncertainty as in Figure 6.10 was input into the reversible-jump Markov chain Monte Carlo algorithm using a cuboid forward model. The prior probability distributions used were the same as used in Section 5.1.1 for the Nuclear Bunker anomaly. Similarly for the Cauchy proposal distribution widths.

As we are now dealing with actual gravity data, we commit more computational time to the problem. Twenty five independent runs were carried out, each of 200,000 iterations. All of the runs passed the Geweke convergence test. Let us first analyse the combined distribution of the number of objects, k. Figure 6.11 shows that we have obtained what initially my seem a somewhat surprising result, one object is favoured over the known value of two. Figure 6.20 shows the spatial distribution of these converged models. The slope of the measurement points is apparent from the yz and xz viewpoints. It is clear that the one object model dominates the accepted solutions, however there is evidence of the true solution of two objects, visible in the xz and yz plots. The main cause of this result is the model uncertainty

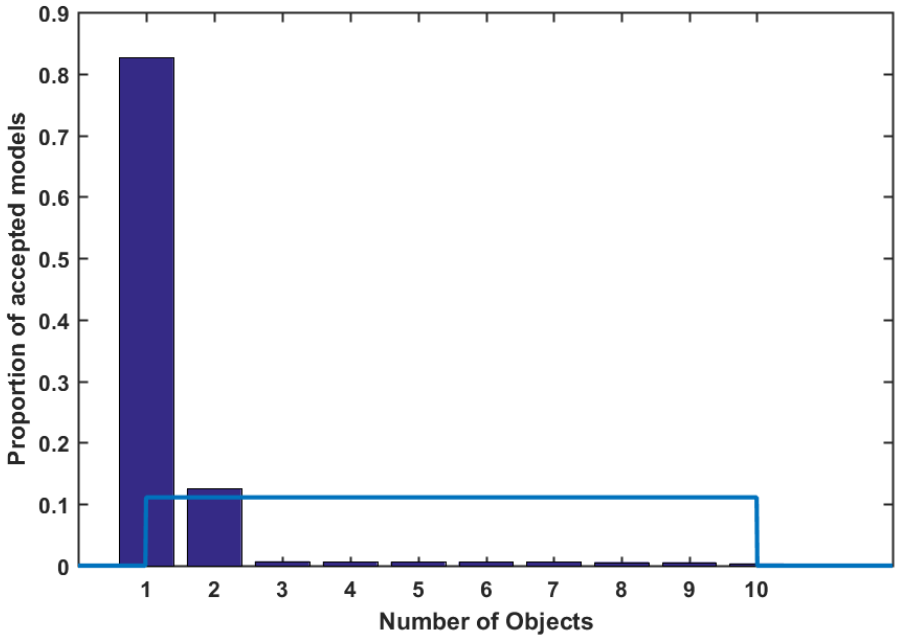


Figure 6.11: Number of objects k, combined from twenty five reversible-jump Markov chain Monte Carlo runs.

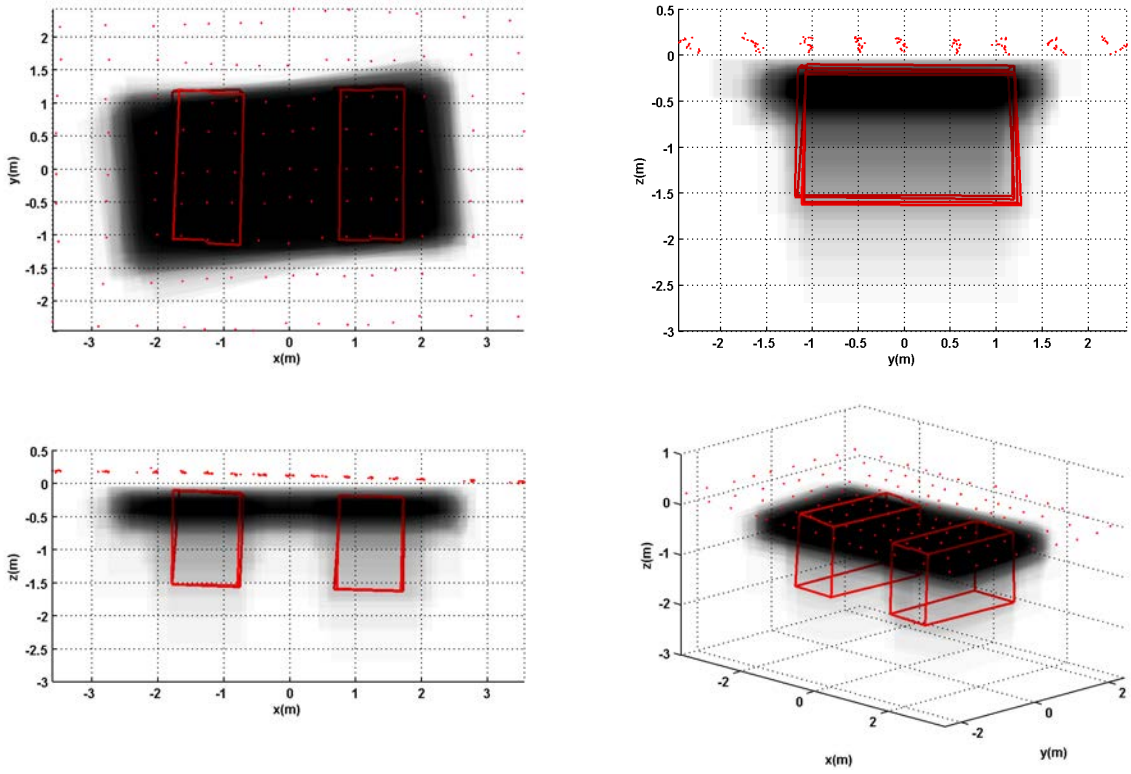


Figure 6.12: Combined twenty five reversible-jump Markov chain Monte Carlo runs, cuboid forward model with CG-5 concrete data. Measured anomaly position outlined in red.

parameter, σ_m. The one object model is found when the model uncertainty is large, the two object model is found when the model uncertainty is small. Figure 6.14 shows the histogram and trace of the model uncertainty parameter, combined from all runs. We see that the

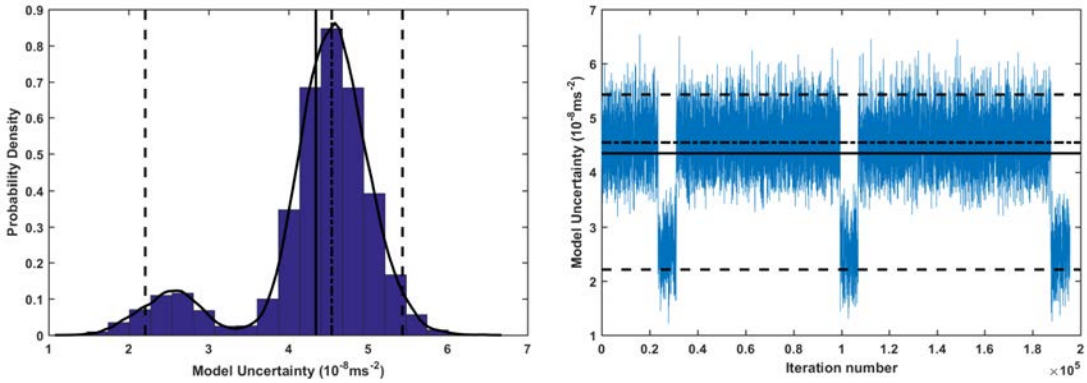


Figure 6.13: Histogram and trace plots of the model uncertainty parameter, σ_m. Evidence of a multi-modal posterior distribution.

model uncertainty posterior is dual peaked. Three of the runs converged to a lower model uncertainty value. The prior probability difference of the two converged uncertainty values is negligible. The prior probability difference of two objects compared to one is significantly larger (due to the total prior being a multiplication of each parameter prior). So it is unsurprising, upon further analysis, that our algorithm favours the simpler one object solution. We do not analyse the parameters of this one object model, as we have no comparison with true model parameters. However, we would expect, as before, for the total mass of the model object to be close to that of the actual anomalies. Figure 6.14 shows the combined total mass from all runs. The total mass meets the approximate actual value within the 95% credibility interval. The one object model appears to approximate more closely the total mass than the two object model.

Figure 6.16 shows the marginal Bayesian posterior distributions for one run (chosen at random) that converged to the one object model. The parameters of the one object model mix well and have to be included as a possible solution given the data-set and prior information available. The corresponding histograms for the two object model are shown in Figure 6.18 (for object two as defined by Figure 6.3, similar results are obtained for object one). The parameters defining object two all converge to the approximate real world values within

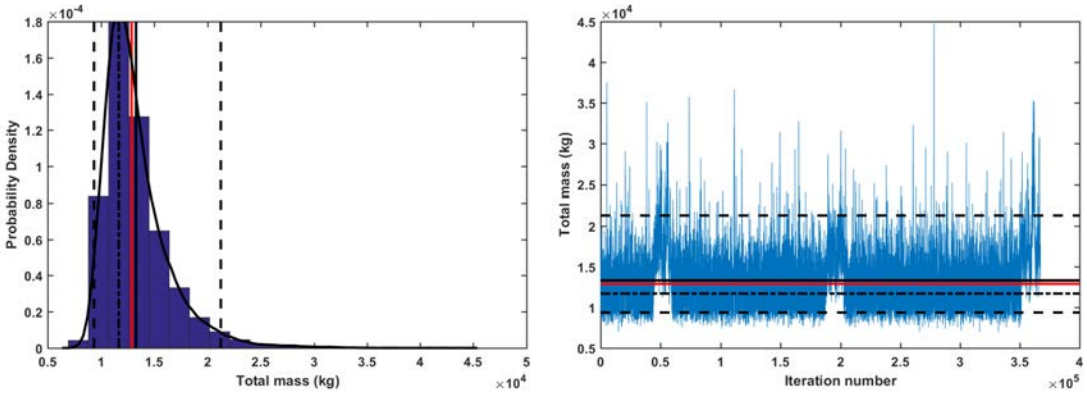


Figure 6.14: Histogram and trace plots of the combined Bayesian posterior distribution for the total mass of 25 runs.

the 95% credibility intervals. The key estimates of the marginal posterior distributions are summarised in Table 6.1 for both objects. We see from the trace plots the same correlation

Object 1							
Parameter	Unit	Actual Value	Mean	MAP	95% LCI	95% UCI	CI Width
x_0	m	-1.25	-1.31	-1.3	-1.41	-1.22	0.19
y_0	m	-0.05	-0.06	-0.06	-0.18	0.05	0.23
$z_{0_{top}}$	m	-0.12	-0.15	-0.14	-0.26	-0.04	0.22
ℓ_x	m	1	1.2	1.14	0.86	1.6	0.74
ℓ_y	m	2.3	2.14	2.1	1.8	2.5	0.77
ℓ_z	m	1.4	1.96	1.5	0.8	5	4.2
ψ	rad	0	0.06	0.1	-0.2	0.33	0.5
$\Delta\rho$	Mgm^{-3}	2	2.01	2.04	1.8	2.2	0.4
Object 2							
Parameter	Unit	Actual Value	Mean	MAP	95% LCI	95% UCI	CI Width
x_0	m	1.25	1.28	1.3	1.2	1.37	0.18
y_0	m	0	0.03	0.06	-0.1	0.13	0.22
$z_{0_{top}}$	m	-0.19	-0.21	-0.19	-0.3	-0.1	0.22
ℓ_x	m	1	1.2	1.2	0.87	1.5	0.68
ℓ_y	m	2.3	2.4	2.4	2.1	2.8	0.72
ℓ_z	m	1.4	1.6	1.3	0.8	3	2.17
ψ	rad	0	-0.1	-0.13	-0.3	0.13	0.43
$\Delta\rho$	Mgm^{-3}	2	2.01	2	1.8	2.2	0.38

Table 6.1: Summary of key posterior distribution parameters for both objects.

between the object shortest side and depth as seen with the nuclear bunker synthetic data analysis (see Section 5.1.1). The uncertainty of the ℓ_z parameters are the largest of all, the mixing of this parameter is quite poor relative to the other parameters. This is due to ℓ_z

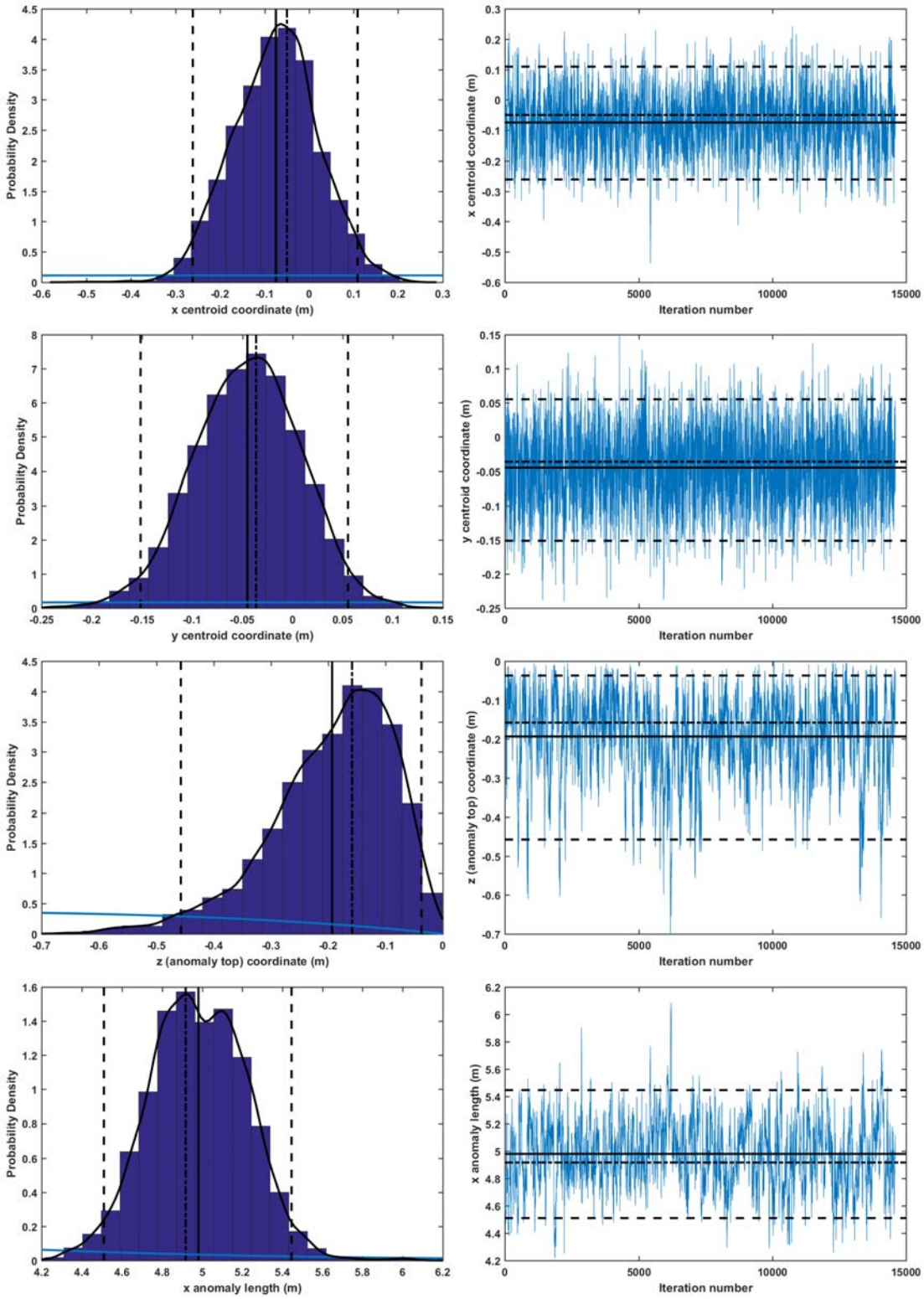


Figure 6.15: Histogram and trace plots of marginal Bayesian posterior distributions for the one object model. Parameters x_0, y_0, z_0 and ℓ_x are shown (from top to bottom).

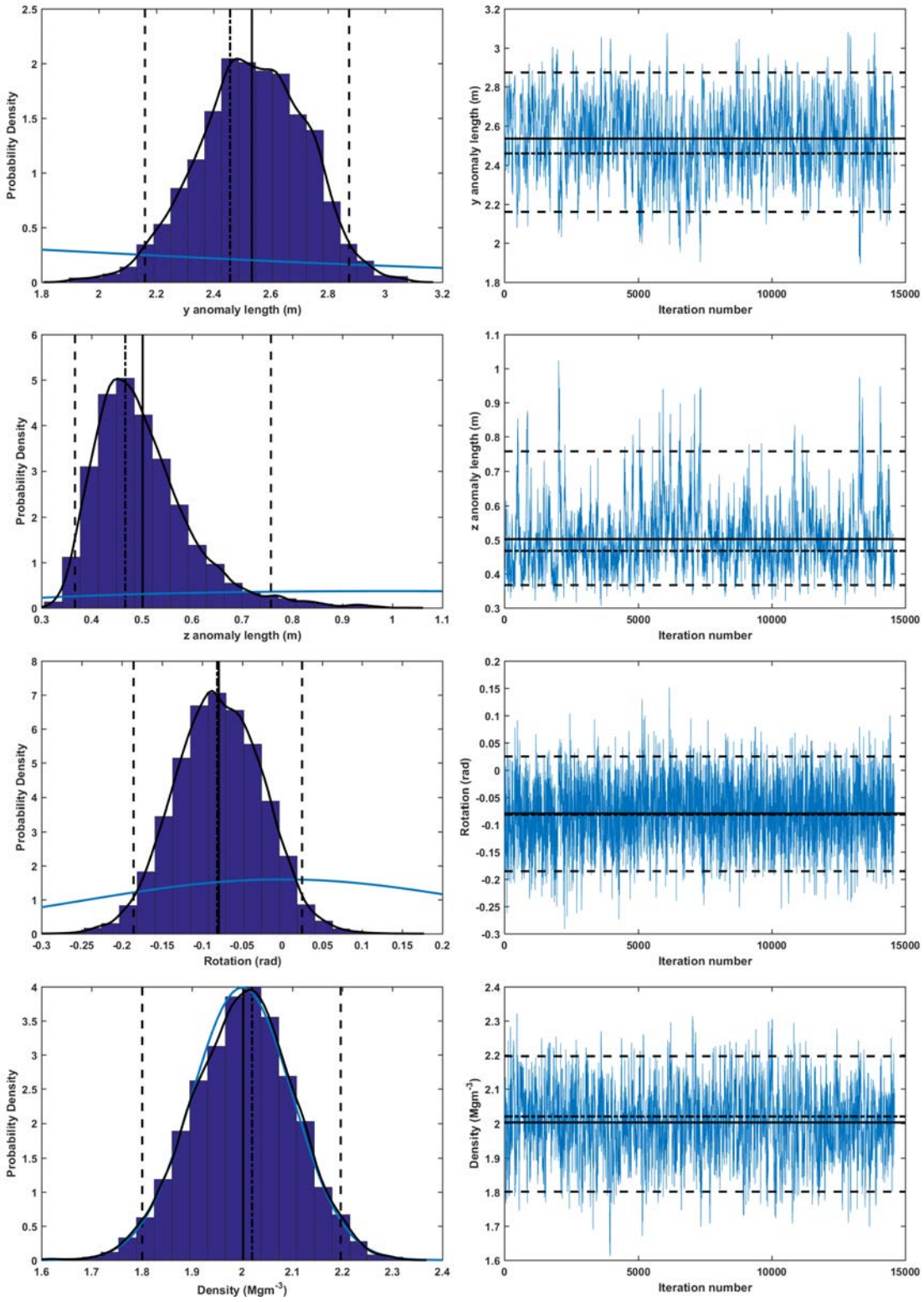


Figure 6.16: Histogram and trace plots of marginal Bayesian posterior distributions for the one object model. Parameters ℓ_y, ℓ_z, ψ and $\Delta\rho$ are shown (from top to bottom).

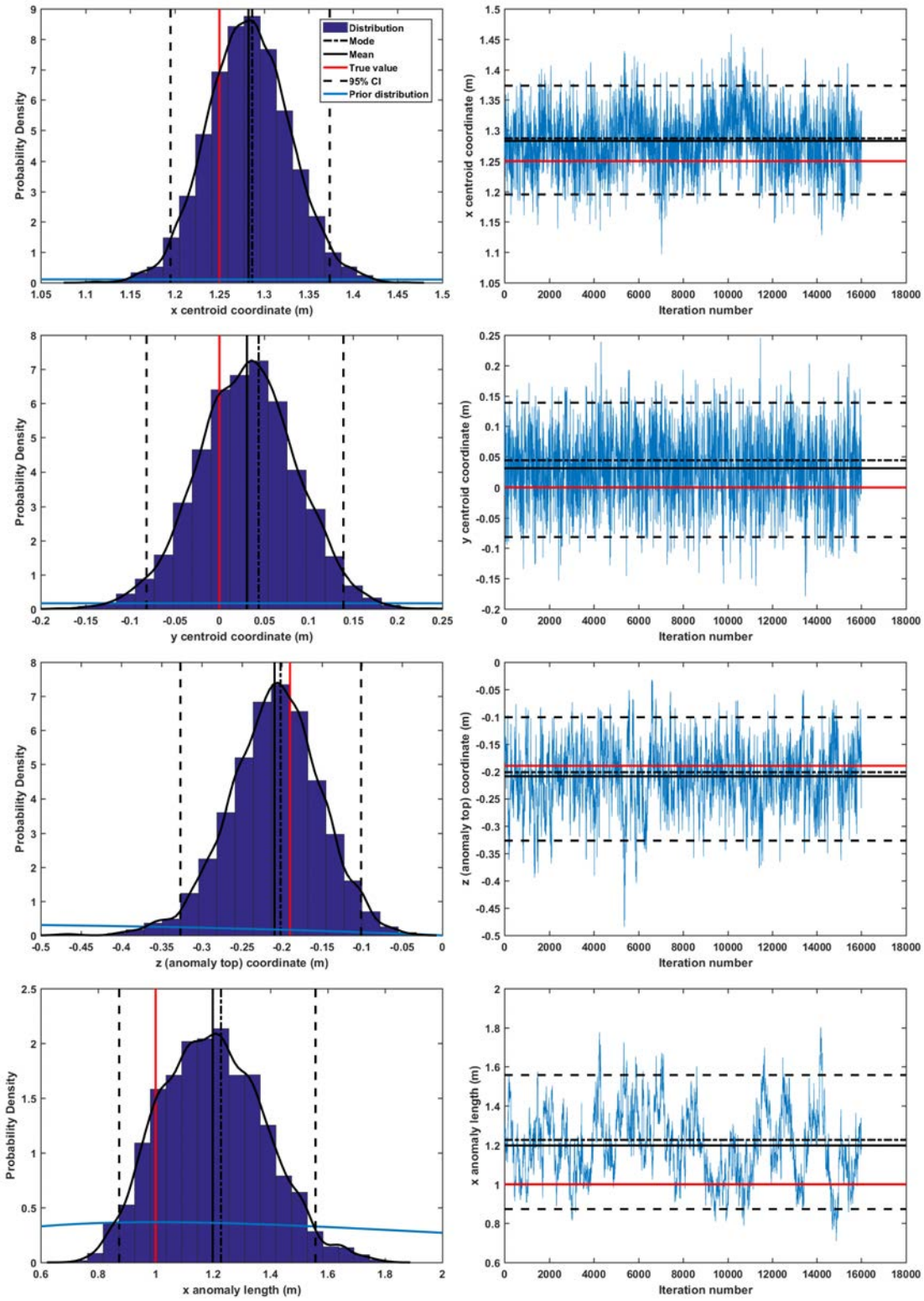


Figure 6.17: Histogram and trace plots of marginal Bayesian posterior distributions for object two of the two object model. Parameters x_0, y_0, z_0 and ℓ_x are shown (from top to bottom).

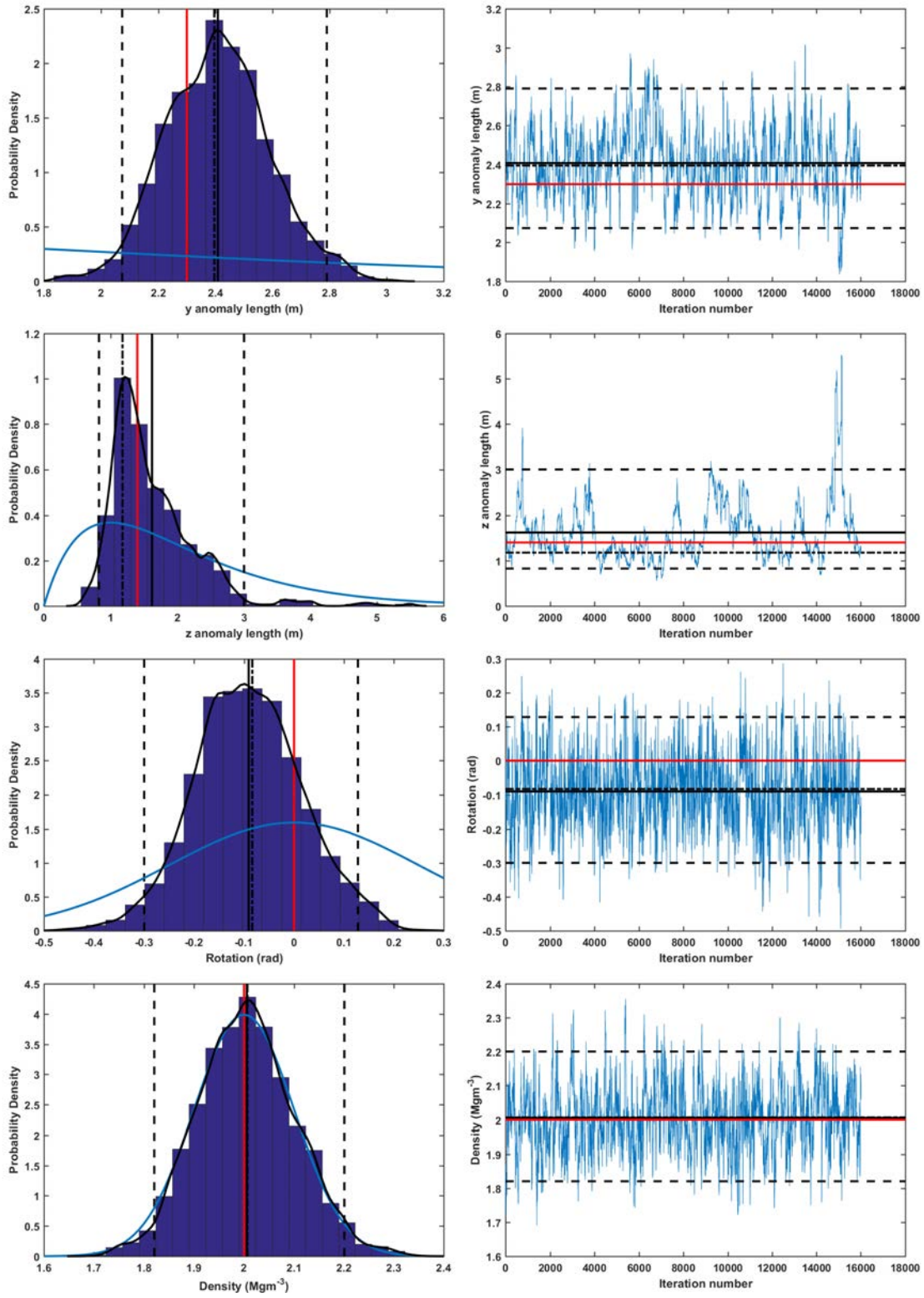


Figure 6.18: Histogram and trace plots of marginal Bayesian posterior distributions for object two of the two object model. Parameters ℓ_y, ℓ_z, ψ and $\Delta\rho$ are shown (from top to bottom).

having little affect on the forward model as its value changes. Looking at the width of the credibility intervals, displayed in Table 6.1, we can see that the parameters of object two are generally marginally more tightly defined than those of object one. Now we consider the role of the model uncertainty parameter. The gravity measurements were taken over multiple days in a variety of environmental conditions. It is not clear that the assignment of one model uncertainty parameter spanning multiple measurement days is a physically reasonable thing to do. So, let us proceed to run the reversible-jump Markov chain Monte Carlo algorithm with the model uncertainty parameter set equal to zero. With $\sigma_m = 0$, we are assuming that our total uncertainty is contained within the data uncertainty σ_d. Such an assumption may not be viable for future high precision quantum gravity instruments, as the total uncertainty for such an instrument will be dominated by other environmental factors such as anomalous density variations in the near-surface.

The gravity measurement data was re-run with $\sigma_m = 0$, a total of twenty five separate runs were completed of which five runs failed the convergence diagnostic test. When compared to the twenty five runs with σ_m as a free parameter, of which no runs failed, we see that the model uncertainty parameter provides greater flexibility for the algorithm to converge to a solution more quickly than is possible with set measurement uncertainty alone.

Based on the previous runs we would expect all of our converged data-sets to converge to the two object model. Figure 6.19 shows the number of objects combined from the twenty converged runs. As expected, we see more complex models favoured by the algorithm now that the total uncertainty is reduced. The increase for models with nine and ten objects is due to the tendency of the algorithm to favour complex models initially and over time delete those that are no longer required to fit the measurement data. The spatial distribution of the models is shown in Figure 6.20. The algorithm is no longer able to increase the uncertainty through the σ_m parameter to permit the single object solution. As a result we converge to the two object model for every separate run. The posterior distributions of parameters independent of k are shown in Figure 6.21. The total mass parameter encompasses the actual value within the 95% credibility interval. The shift towards larger total mass is possibly due to the

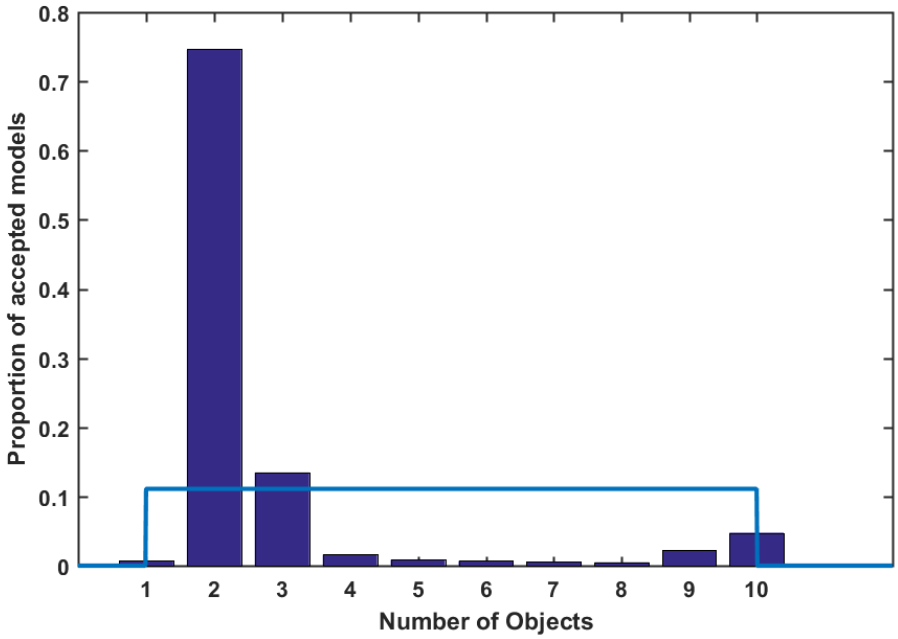


Figure 6.19: The number of objects k, combined from twenty runs with $\sigma_m = 0$.

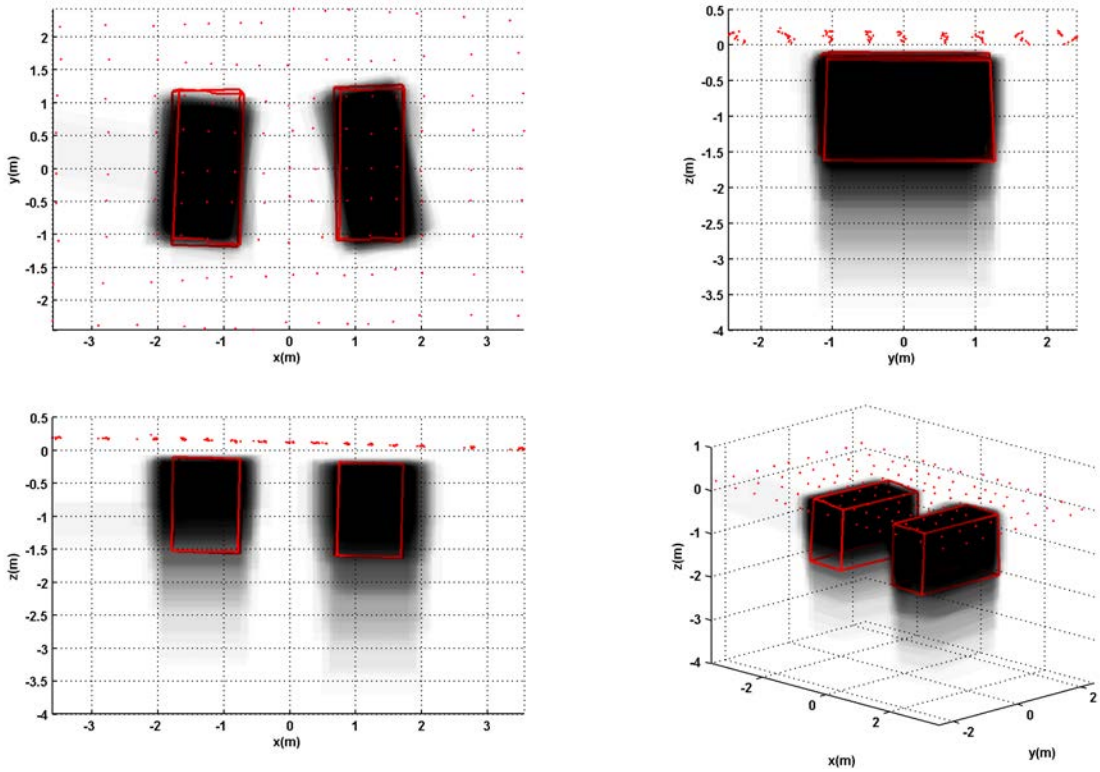


Figure 6.20: Combined twenty reversible-jump Markov chain Monte Carlo runs, cuboid forward model with $\sigma_m = 0$.

combined ambiguity of two objects summing, such that the total mass is inevitably larger than that found with just one object. We now look at the Bayesian posterior distributions

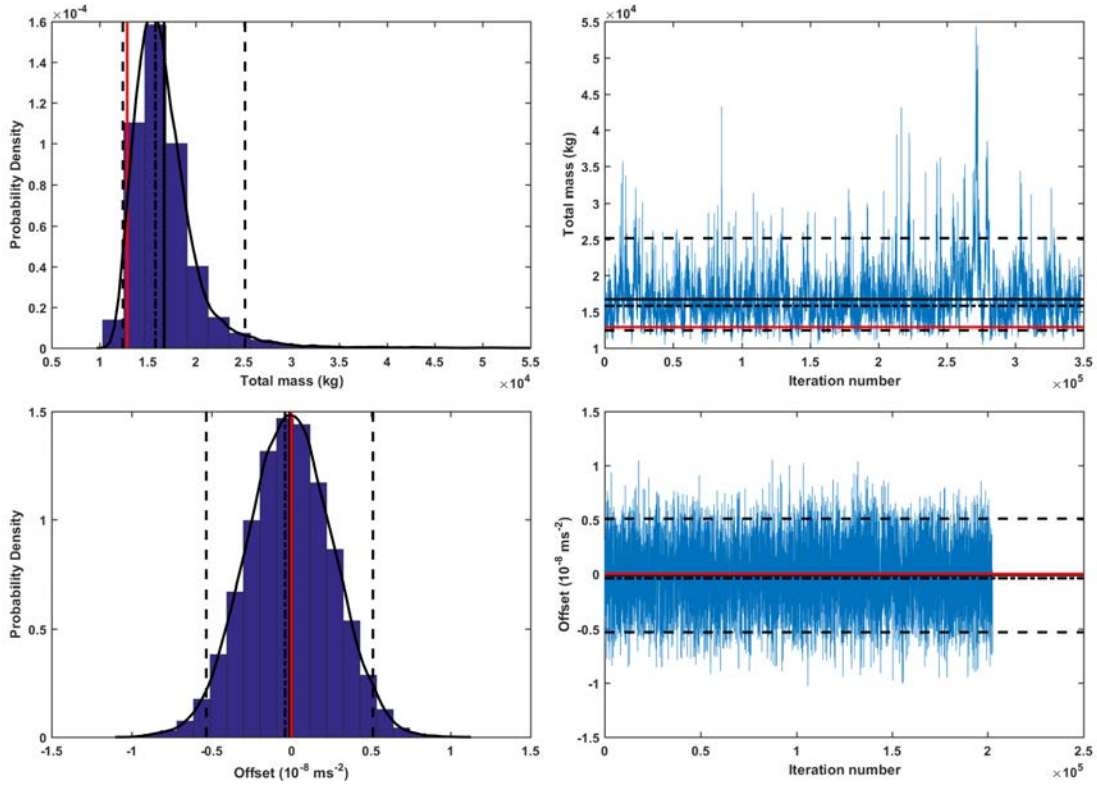


Figure 6.21: Combined Bayesian posterior distributions of twenty converged runs for the total mass and gravity offset parameters (when $\sigma_m = 0$).

for object two of one converged run chosen at random, so that we can compare our analysis with the previous runs, where σ_m was a free parameter. Figure 6.23 shows the histograms and trace plots, along with the actual values and prior probability distributions. Table 6.2 summarises the pertinent estimates from the Bayesian posterior distributions of both objects.

We see that due to the reduced total uncertainty, the credibility interval widths decrease slightly overall when compared with Table 6.1. We still find all of the actual parameter values within the credibility intervals. Again we see the theme of object two being marginally more tightly defined than object one. This may be due to the asymmetric measurement uncertainties (see Figure 6.10), some measurement points on the left of the grid have a larger uncertainty value than the corresponding points on the right side, due to different numbers of measurements being carried out on different days.

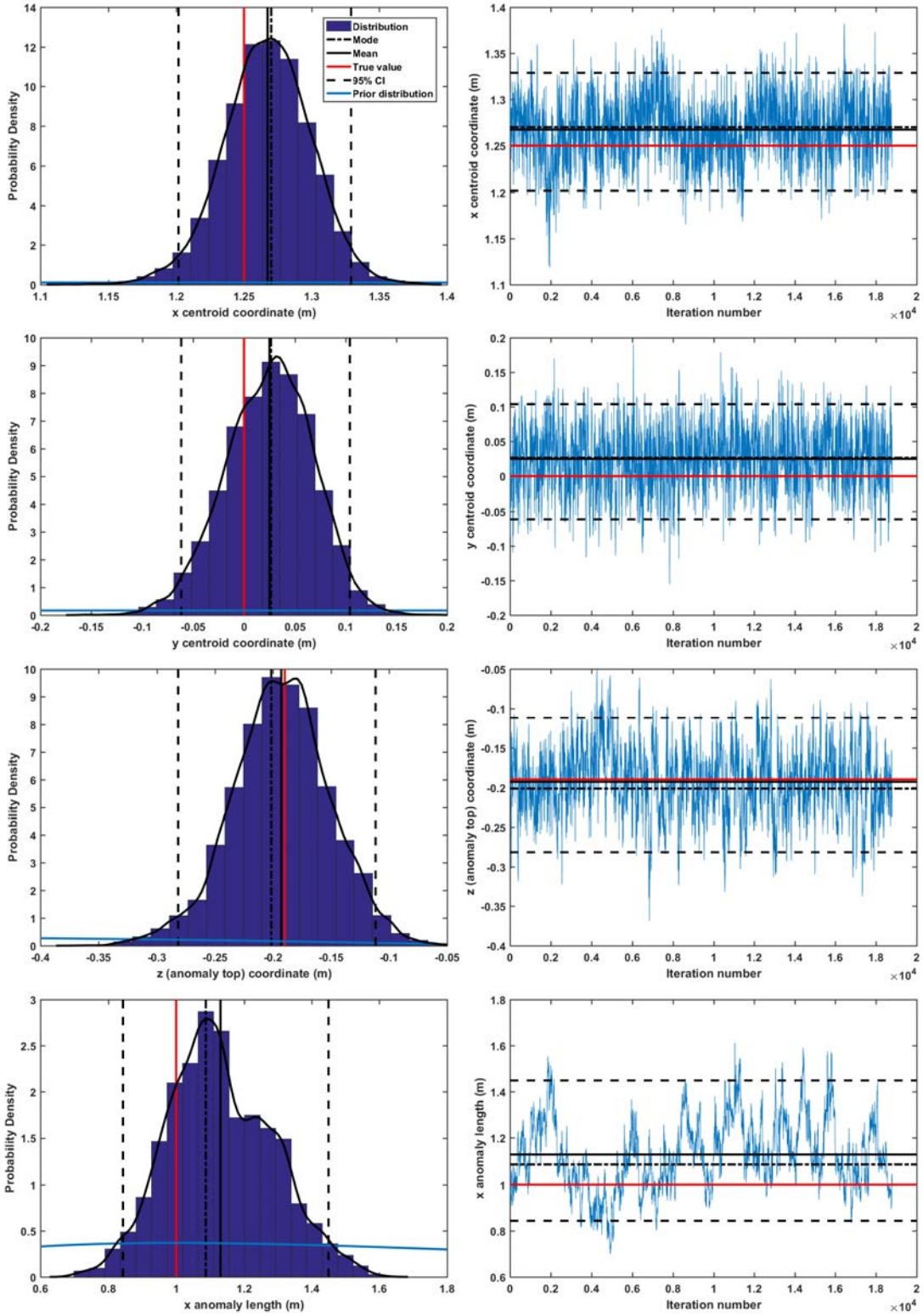


Figure 6.22: Histogram and trace plots of marginal Bayesian posterior distributions for object two of the two object model with $\sigma_m = 0$. Parameters x_0, y_0, z_0 and ℓ_x are shown (from top to bottom).

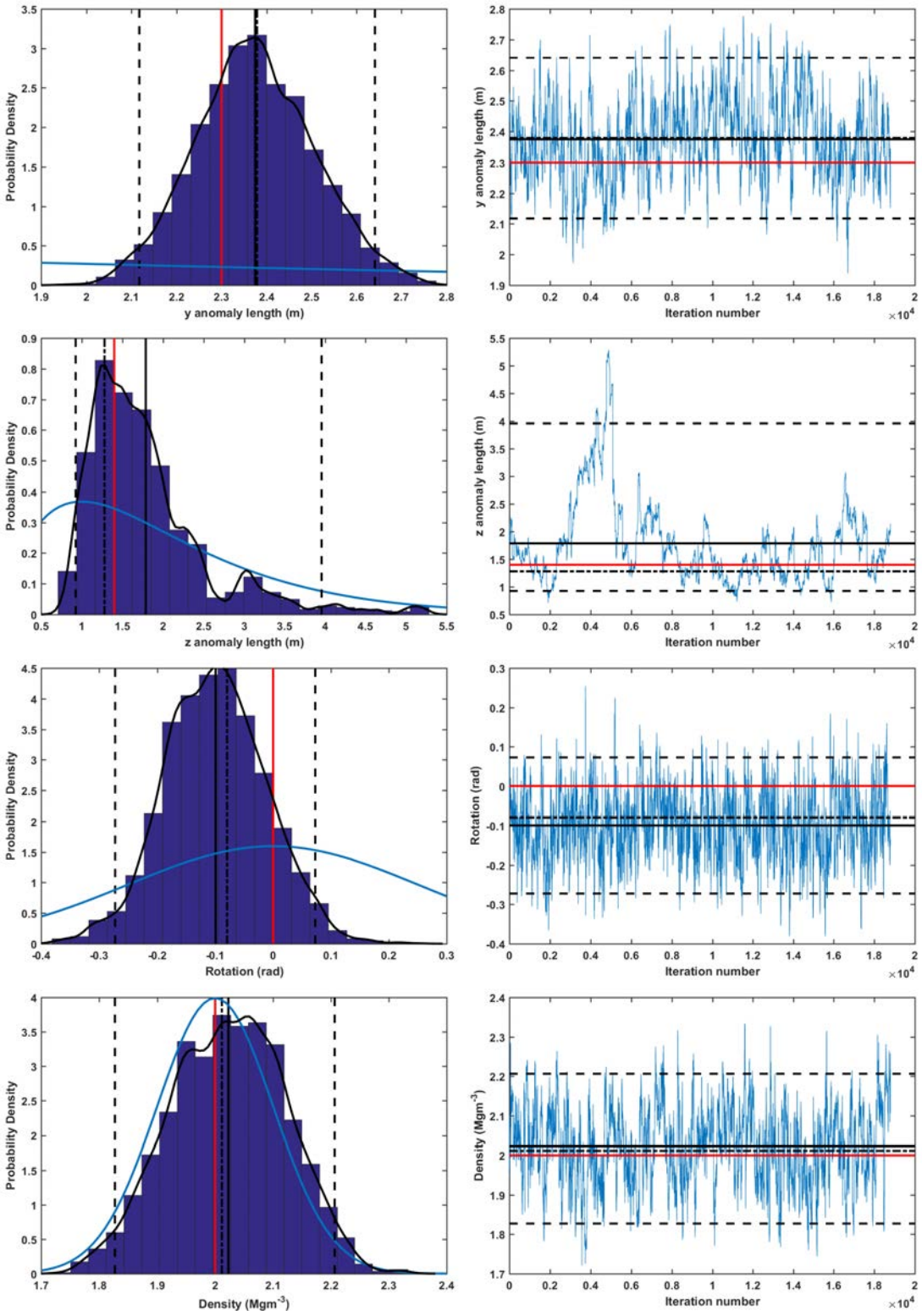


Figure 6.23: Histogram and trace plots of marginal Bayesian posterior distributions for object two of the two object model with $\sigma_m = 0$. Parameters ℓ_y, ℓ_z, ψ and $\Delta\rho$ are shown (from top to bottom).

Object 1							
Parameter	Unit	Actual Value	Mean	MAP	95% LCI	95% UCI	CI Width
x_0	m	-1.25	-1.31	-1.32	-1.38	-1.23	0.13
y_0	m	-0.05	-0.05	-0.04	-0.14	0.03	0.17
$z_{0_{top}}$	m	-0.12	-0.15	-0.14	-0.23	-0.06	0.17
ℓ_x	m	1	1.3	1.22	1	1.6	0.58
ℓ_y	m	2.3	2.13	2.1	1.84	2.4	0.58
ℓ_z	m	1.4	1.43	1.36	0.8	2.4	1.6
ψ	rad	0	0.08	0.08	-0.14	0.3	0.44
$\Delta\rho$	M)	2	2.03	2.02	1.84	2.2	0.37
Object 2							
Parameter	Unit	Actual Value	Mean	MAP	95% LCI	95% UCI	CI Width
x_0	m	1.25	1.27	1.28	1.2	1.33	0.12
y_0	m	0	0.03	0.04	-0.05	0.12	0.16
$z_{0_{top}}$	m	-0.19	-0.19	-0.19	-0.27	-0.12	0.15
ℓ_x	m	1	1.2	1.11	0.93	1.6	0.53
ℓ_y	m	2.3	2.4	2.4	2.17	2.6	0.47
ℓ_z	m	1.4	1.48	1.5	0.9	2.5	1.65
ψ	rad	0	-0.11	-0.12	-0.3	0.06	0.33
$\Delta\rho$	Mg	2	2.02	2.02	1.84	2.2	0.36

Table 6.2: Posterior distribution summaries for the two object model with $\sigma_m = 0$.

We briefly discuss a final set of reversible-jump Markov chain Monte Carlo runs, where a blanket measurement uncertainty given by the collection of all base-station readings (see Figure 6.7) is used. A data uncertainty, $\sigma_d = 3.7 \times 10^{-8}$ ms^{-2} was found using this method, we apply this to all of our measurement points. Again we set the model uncertainty, $\sigma_m = 0$. The algorithm was run twenty five times, with only one run failing to converge. Figure 6.24 shows the object number k, combined for all converged models and figure **??** shows the corresponding spatial distribution. We see that the number of model objects k, is more evenly split, slightly in favour of two objects.

We have a situation where, when we define our data uncertainty separately for different measurement days, we have convergence to a complex ($k = 2$) model solution. The mean data uncertainty value for this case is 3.46×10^{-8} ms^{-2}. Which is essentially the same as the blanket data uncertainty of 3.7×10^{-8} ms^{-2}, which produces a more even split between the models with $k = 1$ and $k = 2$. The reason for this discrepancy change is that our most important measurement points (points with lowest data uncertainty) are effectively given

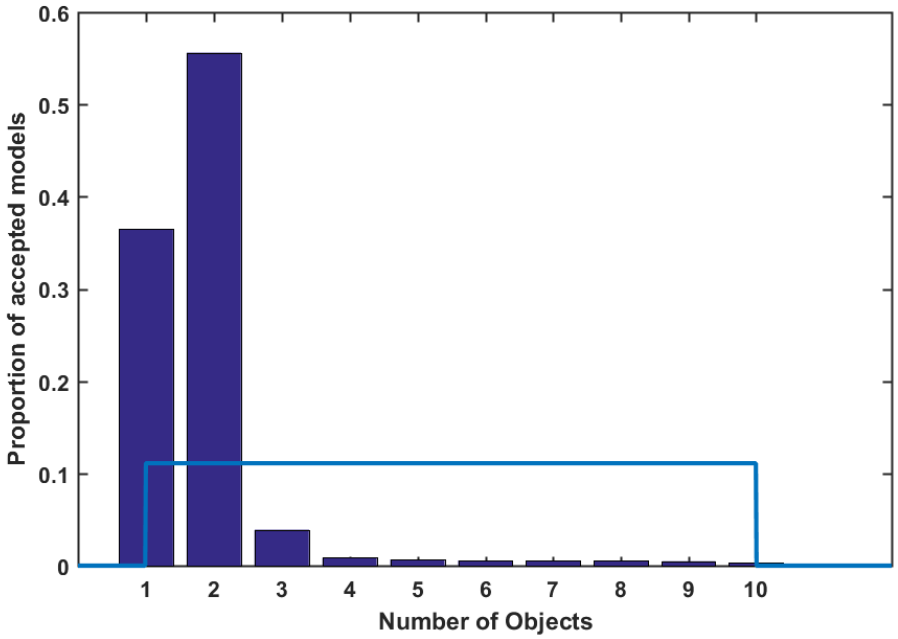


Figure 6.24: The number of objects k, combined from twenty four runs with $\sigma_m = 0$.

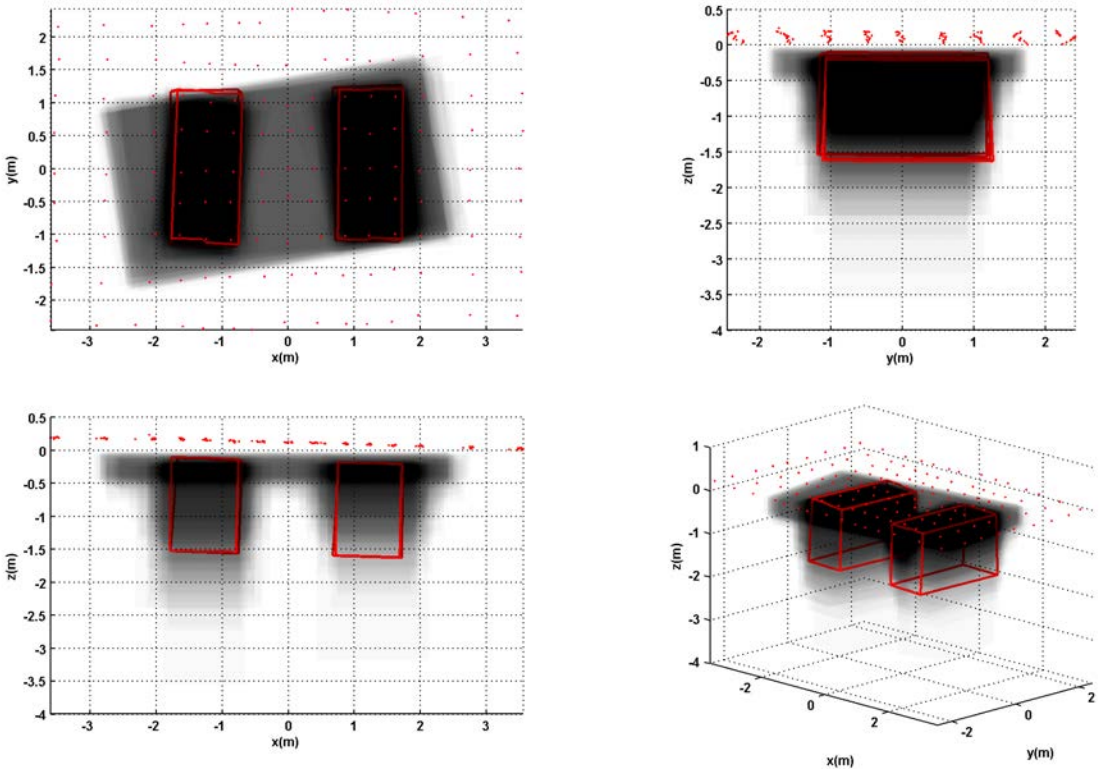


Figure 6.25: Combined twenty four rj-McMC runs, cuboid forward model with uniform measurement uncertainty.

the same weight in the inference process when using a blanket data uncertainty. Looking at Figure 6.10 it is clear that measurement line 5 (the central line parallel to the x axis) gives by far our lowest data uncertainty at, 2×10^{-8} ms^{-2}. Without this low data uncertainty for arguably the most important measurement line (peak signal) our reversible-jump Markov chain Monte Carlo outputs become more ambiguous with regard to the number of model objects.

6.3.2 The Prior Effect

As previously mentioned, an often cited weakness of the Bayesian formulation of inference problems is that of sensitivity to seemingly arbitrary (or subjective) prior probability distributions. From the Bayesian perspective, sensitivity to prior distributions is simply an indication that the data do not contain sufficient information to increase our knowledge of a given parameter. In Chapter 5 and in the examples given in this chapter, we suggested prior distributions for the model parameters with some rationale to justify our choices. Other authors may, and indeed inevitably will, come to different conclusions regarding what are to be deemed reasonable prior probability distributions for a given gravity survey. The possible set of priors that may be chosen is infinite, so we focus on some of the more likely assignments that could be made.

The last example analysed, that of the concrete block experiment, with a blanket measurement uncertainty of 3.7×10^{-8} ms^{-2} will be re-analysed here using different prior distributions for comparison. This data-set was chosen for comparison as there was some ambiguity in the convergence between models with $k = 1$ and $k = 2$ for multiple runs (see Figure 6.24). So changing the prior distributions may have an effect on the relative frequencies of converging to each model; which will not be noticeable for the individual runs. We ignore the extreme case of choosing priors such that the actual parameter values are not contained within the model space, limiting ourselves to prior distributions that could be justifiably used.

Firstly we consider the case where we use the same distributions as previously given for the cuboid model, but we tweak the hyper-parameters slightly of certain distributions. The

model parameters: z_o, ℓ_x, ℓ_y and ℓ_z have their priors altered from Gamma distributions with shape parameter $a = 2$ and scale parameter $b = 1$ to shape parameter $a = 2$ and scale parameter $b = 2$. We know from Section 3.1.2 that this change shifts the peak of the probability distribution from one to two. This is a very slight change that we are making to the prior distribution, we would hope that such a minimal change might have little effect on the output of multiple runs. We expect that there may be a bias imparted (relative to the previous prior probability distributions) in favour of the $k = 1$ model. This is due to the fact that we will on average be generating deeper and larger cuboids for each birth proposal and we can imagine that larger and deeper cuboids will have a higher chance to eventually converge to the larger cuboid, corresponding to the $k = 1$ model.

We run the reversible-jump Markov chain Monte Carlo algorithm with the altered prior probability distributions for twenty five separate chains. Figure 6.25 shows the number of model objects of the combined twenty four runs that were deemed to have converged, passing the Geweke diagnostic. The expectation of a slight change in convergence between the $k = 1$

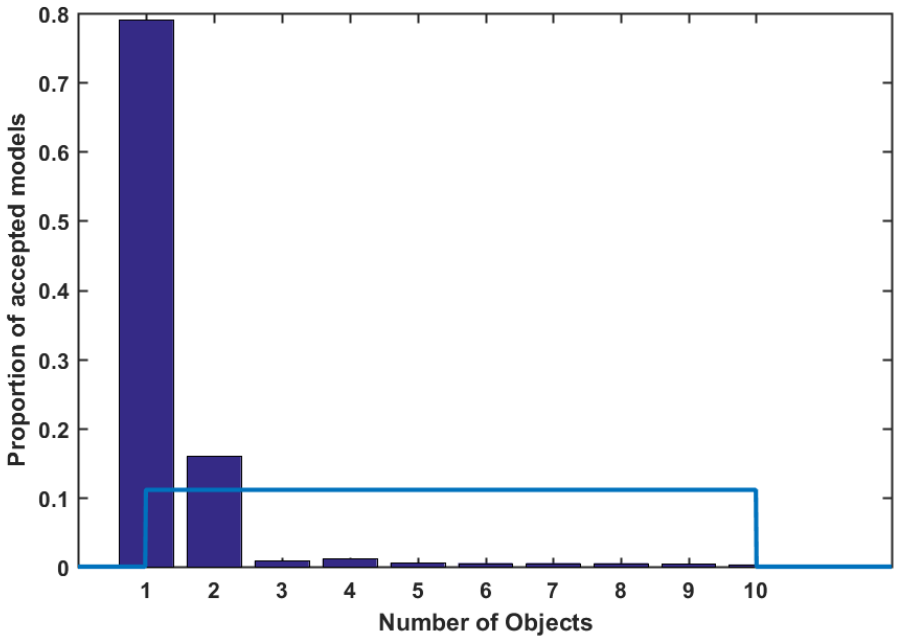


Figure 6.26: The number of objects k, combined from twenty four runs. With $\sigma_m = 0$ and altered prior probability distributions.

and $k = 2$ models was underestimated. All of the runs, except three, converged to the one

object model. As compared with the previous set of runs, for which nine runs converged to the one object model. This result is troubling for the reversible-jump algorithm, at least for it's applicability to this particular data-set. We can examine the survey data more closely in an attempt to explain the extreme change in inference outputs caused by a slight change in the prior probability distributions.

The data-set we have collected is an atypical gravity survey in many respects. The instrument was positioned directly on top of the gravity anomalies, giving a more highly peaked signal than we may expect from an anomaly buried underground, such as the bunker and pipeline anomalies discussed in Chapter 5. Obviously measuring directly on top of the anomalies about which we wish to infer information is not a realistic survey scenario. Also, the positioning of the two anomalies with their centroids aligned in the y-axis and offset by a small distance in the x-axis such that the gravity signals overlap, is again atypical for real world applications. We were also using the blanket measurement uncertainty for this comparison, which as we argued previously adversely effects the data-set due to the uncertainty of measurement line five increasing by a factor of two. As a result our data-set is less informative, and less able to 'combat' the change in the prior probability distributions.

It is usually the case in the geophysical literature that prior probability distributions are set as uniform for each model parameter, defined by some predetermined bounds[82][81]. There are a number of advantages to using uniform prior distributions. They are trivial to compute, taking up minimal computational resources in comparison to the Gamma and Gaussian distributions. They are also easy to visualise. If we say that we have used a uniform prior distribution between zero and five, this is more intuitive than saying that we have used a Gamma distribution with scale parameter two and shape parameter one.

In some cases it may also be advantageous to define uniform priors for certain parameters. As discussed earlier for the cylinder model (see Section 5.2.1); a uniform prior limiting an object centroid to be within the measurement grid can be advantageous in certain situations, such as the semi-infinite anomaly problem. There are always trade-offs regarding prior distribution choice (and choice of model), and defining the centroid to be within the measurement

grid trades a certain amount of flexibility offered by the Gaussian prior distribution, used for the x_0 and y_0 parameters throughout the research.

To provide another comparative analysis, we assign uniform prior distributions to our model parameters as shown in Table 6.3 The prior distributions for the density parameter $\Delta\rho$ and

Uniform			
Parameter	Unit	Lower Bound	Upper Bound
x_0	m	min($\boldsymbol{x}$)	max($\boldsymbol{x}$)
y_0	m	min($\boldsymbol{y}$)	max($\boldsymbol{y}$)
z_0	m	-10	0
ℓ_x	m	0	max($\boldsymbol{x}$)
ℓ_y	m	0	max($\boldsymbol{y}$)
ℓ_z	m	0	max(max($\boldsymbol{x}$), max($\boldsymbol{y}$))
ψ	deg	-10	190

Table 6.3: List of uniform prior distributions used for all model parameters.

offset parameter η are kept the same as before (both Gaussian distributions). The x_0 and y_0 prior bounds are set as per the measurement grid bounds. The object depth is set at a maximum of ten metres as we do not expect to be able to resolve anomalies at a depth greater than this. The ℓ_x and ℓ_y length parameters of the cuboid have their maximum prior bounds set based on the half-length of the measurement grid. As the measurement grid is offset so that our zero point is the central point, we simply take the maximum measurement position values for the x and y axes respectively. The ℓ_z parameter is defined between zero and the maximum of the xy measurement length. For an actual survey, the lower bounds could be raised from zero, as evidently we do not expect to be able to resolve objects that have such small volumes as allowed by these prior distributions.

Using the prior distributions set as above, twenty five runs of the reversible-jump Markov chain Monte Carlo algorithm were completed of which two runs failed convergence. Before we analyse the results, it is worthwhile considering the key differences of the prior distributions used here to those used initially Figure 6.26 shows the difference between the ℓ_x parameter prior distributions. Assigning the uniform distribution cuts off the higher values that were possible when using the Gamma distribution. This will result in birth steps generating objects from the uniform distribution that are smaller in volume than some of those that

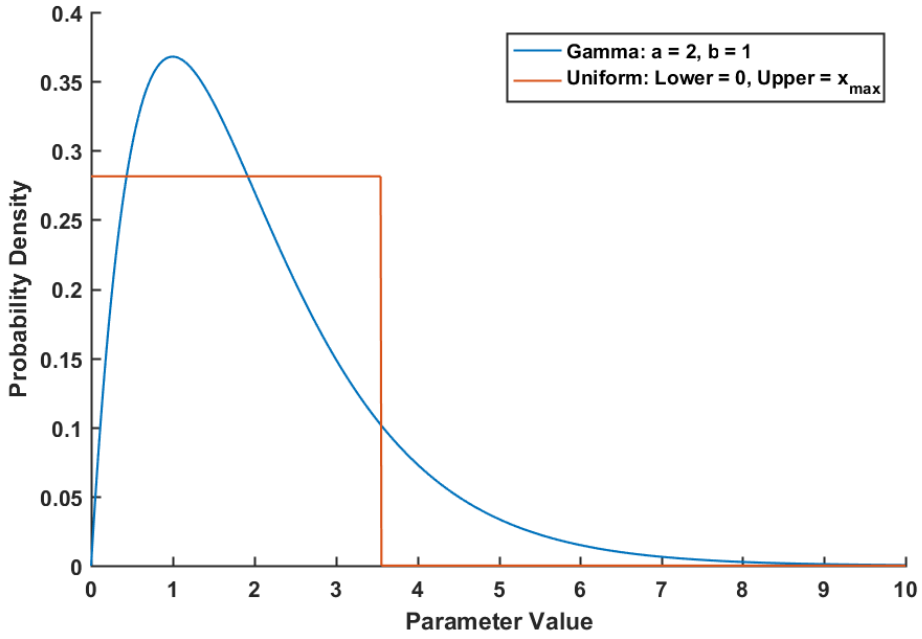


Figure 6.27: The relative proportion of converged model objects using uniform priors.

are drawn from the Gamma distribution. We should now expect an increase of convergence towards the two object model, as the objects generated from the prior are more likely to be close in terms of volume to the two actual anomalies. Also, setting the prior distributions in this way makes the maximum length of an anomaly in the survey around 3.5 m which is smaller than the longest side length of the one object anomaly cuboid. Again, twenty five runs of the reversible-jump Markov chain Monte Carlo algorithm were undertaken. Figure 6.27 shows the converged number of model objects. As expected we only see the two object model in the outputs, as the increased prior information has prevented the algorithm from converging to the one object model.

We analyse the data-set again with uniform priors, this time with the cuboid length upper bounds set equal to their respective axis length ($\ell_x = \max(\boldsymbol{x}) - \min(\boldsymbol{x})$), effectively doubling the previous values. Here we are stating that our measurement grid could contain objects up to but no larger than our measurement grid area. This would leave no space for background gravity data collection, which should ideally be obtained in a gravity survey. Again, depending on the survey scenario, this may or may not be a reasonable assignment. The rest of the prior distributions are kept the same. From the strong convergence seen previously with the

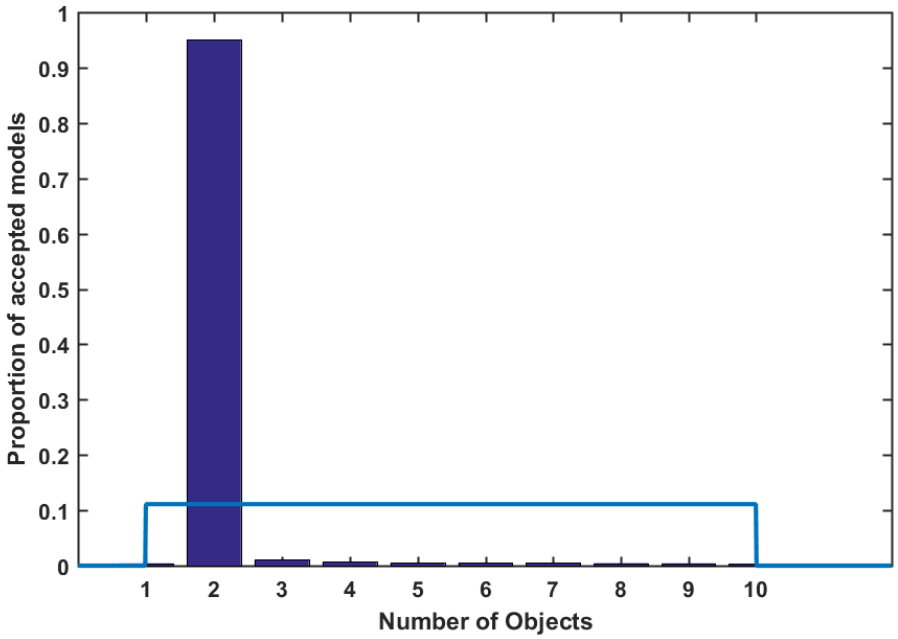


Figure 6.28: The number of objects k, combined from twenty three runs. With $\sigma_m = 0$ and altered prior probability distributions.

constrained uniform priors, we expect a deviation towards models favouring the one object forward model, as we will on average generate objects of a greater spatial extent. Figure 6.28 shows the number of model parameters k for twenty five runs of which four failed to converge. The twenty one runs gave an approximately even distribution between the one and two object models.

We have seen that, although the prior distributions do not have an affect on individual model parameters (except for density), they have a strong indirect effect through the reversible-jump birth update. However, this should be mitigated as the uncertainty associated with the measurement points decreases.

6.3.3 Sphere Forward Model

We can also apply our sphere forward model to the measurement data. We begin again with the case of data uncertainty as per figure 6.10, with the model uncertainty as an unknown parameter to be found. The prior and proposal distributions used were the same as those given in Section 5.1.2. We remember that for the cuboid model, the output of the algorithm

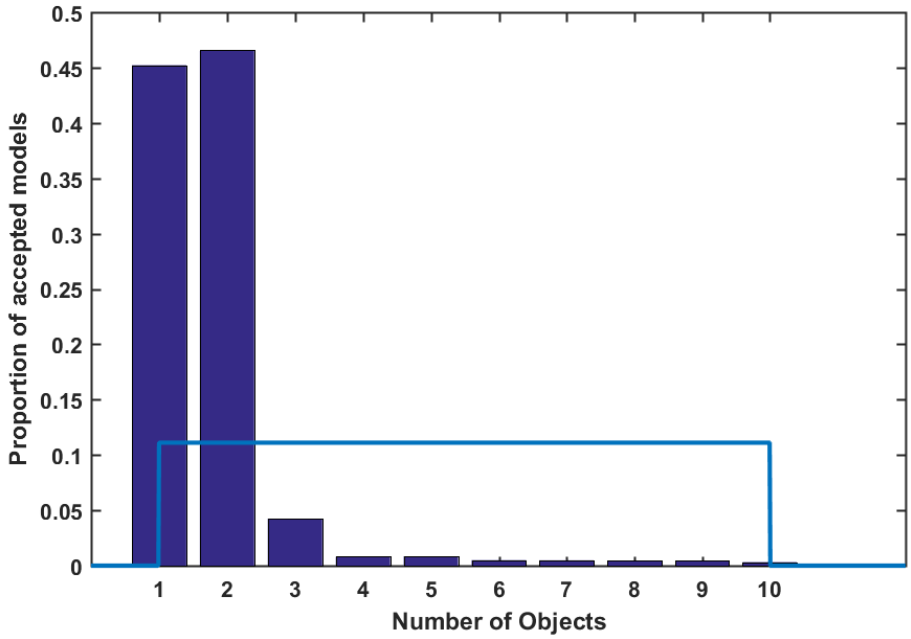


Figure 6.29: The relative proportion of converged model objects using larger uniform priors for the ℓ_x and ℓ_y parameters.

for this data uncertainty scenario was favourable towards a $k = 1$ model. We carried out 25 separate runs of the reversible-jump Markov chain Monte Carlo algorithm of which all were deemed to have converged. Figure 6.29 shows the number of objects k for all converged runs. Due to the discrete nature (it only has radius with no length parameter) of the sphere forward model the runs converge to $k = 2$ every time. However, as the measurements were taken so close to the anomalies, the sphere approximation is not accurate. The three-dimensional spatial plot shown in Figure 6.30 confirms this. The total mass of the two objects combined for all runs is given in Figure 6.31, along with plots of the gravity offset and model uncertainty. The marginal posterior distribution of the total mass does not fit the known value within the credibility interval. This is not surprising as the sphere model is not a good approximation to the very near surface concrete anomalies.

We made twenty five more runs, each of 200,000 iterations, with $\sigma_m = 0$. As with the cuboid model we expect that the algorithm will provide a more complex model with which to describe the data, now that we assume a model uncertainty of zero. Ten of the runs failed the convergence diagnostics indicating, as we saw previously that the model uncertainty param-

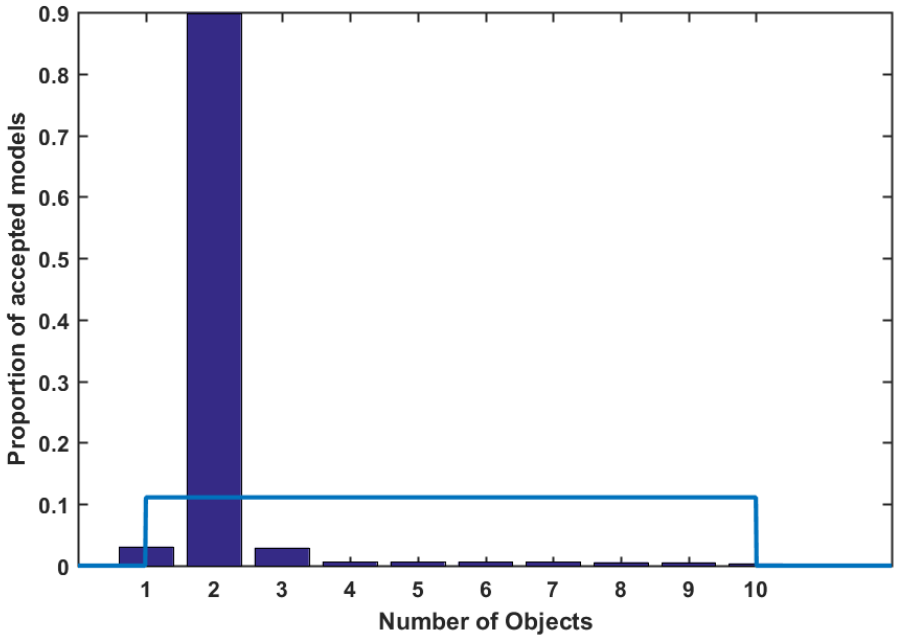


Figure 6.30: The number of objects k, for the sphere forward model. Combined from 25 runs.

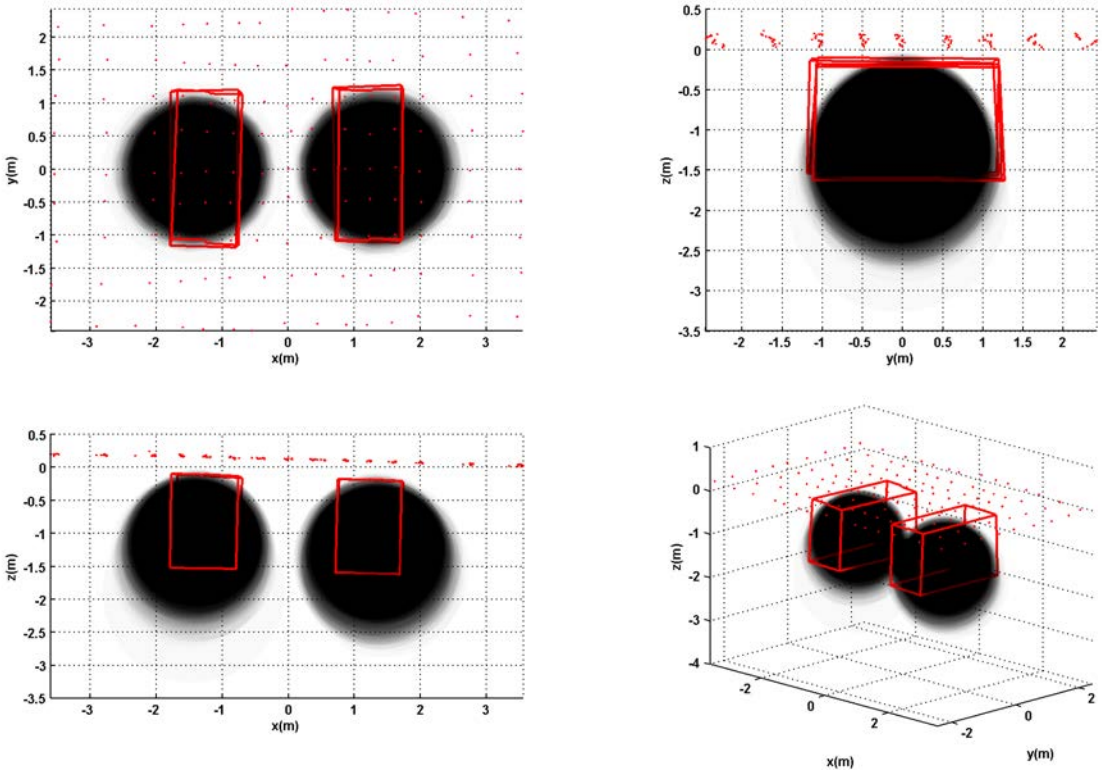


Figure 6.31: Spatial distribution of twenty five combined reversible-jump Markov chain Monte Carlo runs for the sphere forward model.

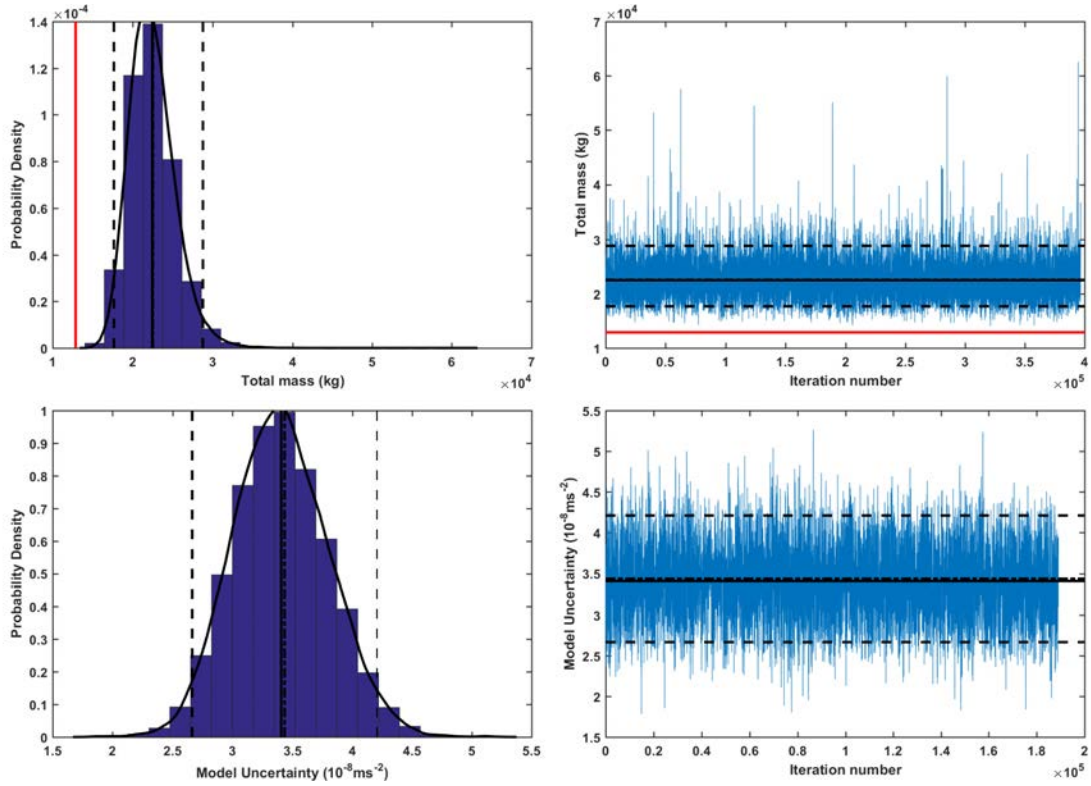


Figure 6.32: Histogram and trace of total mass and model uncertainty for 25 combined runs.

eter provides flexibility which aids convergence. The number of model objects k is shown in Figure 6.32 with the corresponding three-dimensional spatial plot shown in Figure 6.33. The parameters that are independent of the number of model objects are shown in Figure 6.34. As we expect, the complexity of the model is increased due to the smaller total uncertainty. Object two is more well defined by the gravity data than object one, as we saw with the cuboid model. The total mass of the models is closer to the actual value, but still not within the 95% credibility intervals.

6.3.4 The Prior Effect

The sensitivity to prior distributions was demonstrated with the cuboid model in Section 6.3.2. We perform a similar analysis here for the sphere model. We proceed as with the analysis in the previous section, keeping the data uncertainty the same and setting the model uncertainty parameter $\sigma_m = 0$. This set-up was chosen as there was the most ambiguity in

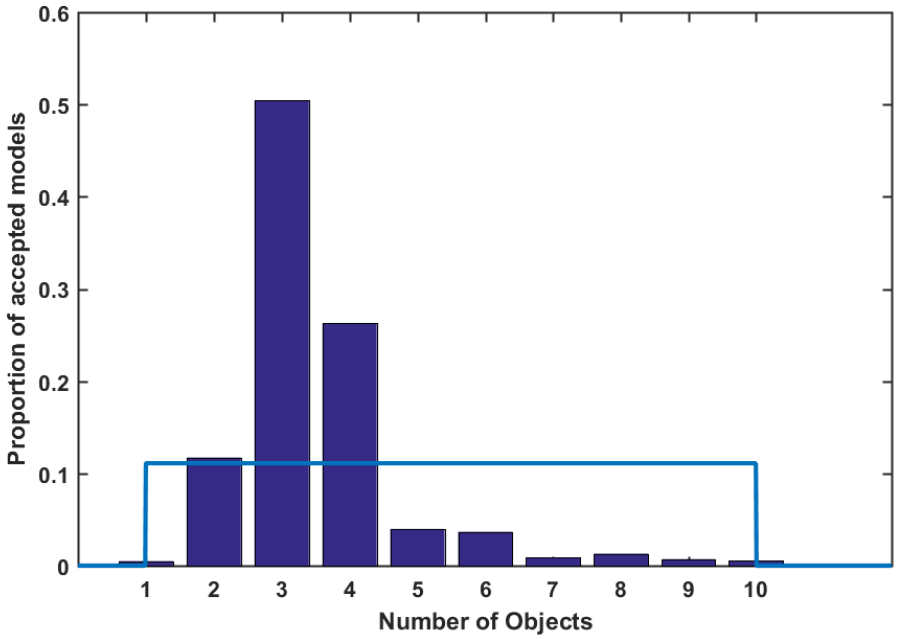


Figure 6.33: The number of objects k, for the sphere forward model. Combined from 15 coverged runs.

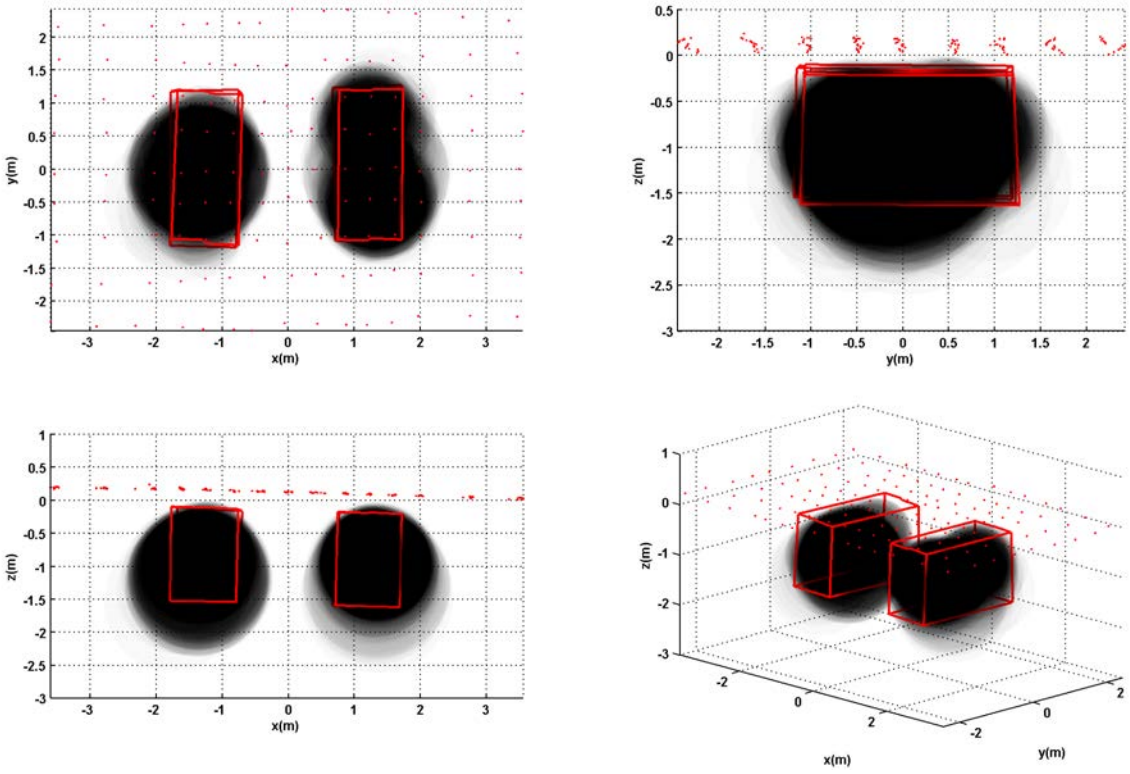


Figure 6.34: Spatial distribution of twenty five combined reversible-jump Markov chain Monte Carlo runs, sphere forward model with $\sigma_m = 0$.

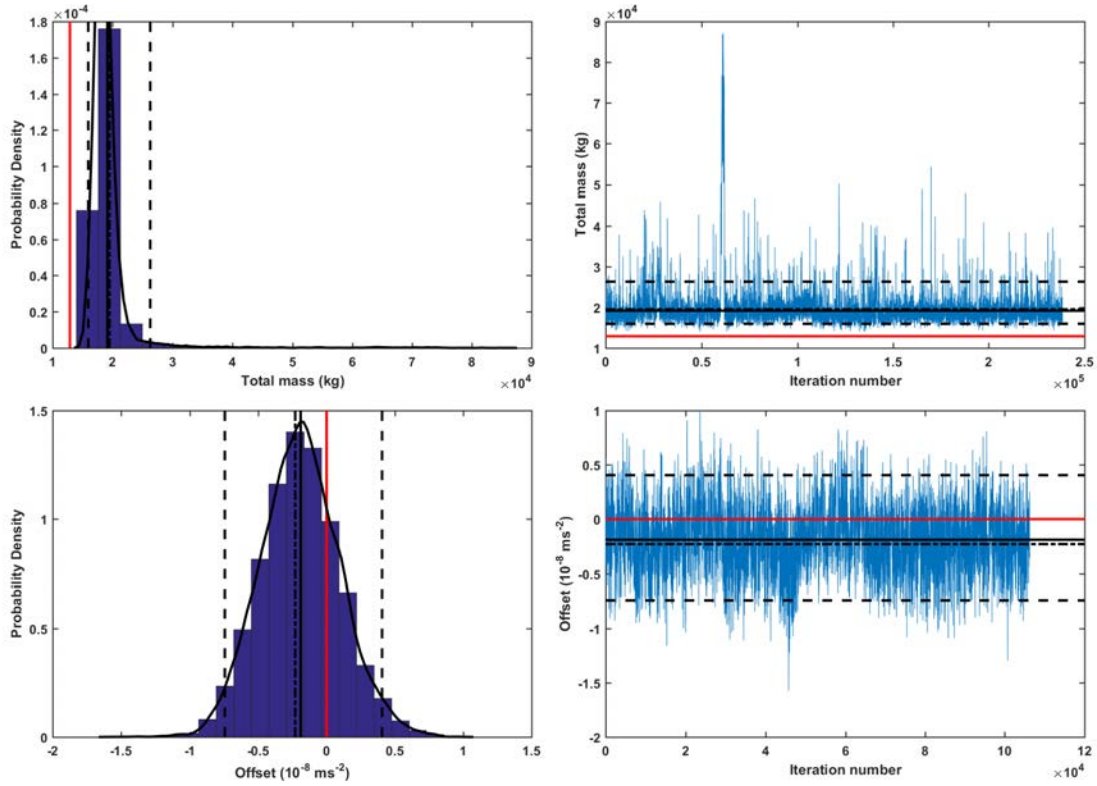


Figure 6.35: Combined Bayesian posterior distributions for the total mass and offset.

the model object number k.

From here we alter the prior distributions only slightly, as we did with the cuboid model. Focusing on the depth and radius parameters only, we increase their respective Gamma distribution scale parameters from 1 and 0.5 to 2 and 1 respectively. As a result, the reversible-jump Markov chain Monte Carlo birth update will propose on average model objects that are larger in volume and deeper than before. We would then expect a shift towards the $k = 2$ model, if the sphere model is sensitive to the prior distributions in a similar way to the cuboid model.

Twenty five runs of the reversible-jump Markov chain Monte Carlo algorithm were made, with twenty two deemed to have converged, a much higher rate of convergence than was found for the smaller prior distributions. Figure 6.35 shows the number of model parameters for the combined converged runs. As expected, a shift towards the two sphere model was made relative to Figure 6.32, due to on average producing larger spheres through the reversible-jump birth update.

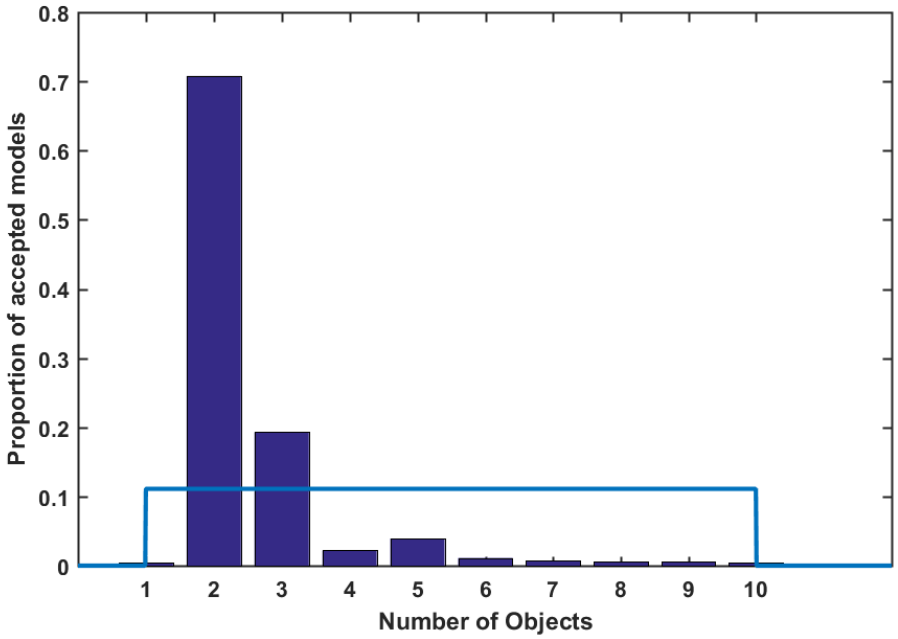


Figure 6.36: The relative proportion of converged model objects using altered sphere parameter prior distributions.

Summary

In this chapter we outlined the experimental procedure used to carry out a small scale gravity survey consisting of 134 measurements using the Scintrex CG-5 gravimeter. The determination of the survey uncertainty was a non-trivial task, due to unknown influence of moving the instrument between measurement points. Approximations of the uncertainty were made using the distributions of base-station readings. The data-set collected matched the expected model of the anomalies (for which the density and volume were known) within the calculated measurement uncertainty. Demonstrating a potential method of comparing gravity instruments in a survey environment.

The reversible-jump Markov chain Monte Carlo algorithm was applied to the measurement data with various realisations of the data uncertainty. As with the synthetic data examples, we saw how the algorithm favours the simplest solution to a given data-set, converging mostly to $k = 1$ when inferring the model uncertainty parameter σ_m for the cuboid forward model. With the measurement uncertainties as per Figure 6.10 and the model uncertainty set to zero, the reversible-jump Markov chain Monte Carlo algorithm always converged to $k = 2$.

For these two model objects, the individual marginal posterior distributions found the known parameter values within the credibility intervals. The mid line (line 5, from Figures 6.5 and 6.3) of the data-set was shown to be by far the most important, due to it's low measurement uncertainty value. When the algorithm was run with a blanket measurement uncertainty of 3.7×10^{-8} ms^{-2} (from Figure 6.7) the algorithm converged to a distribution of one and two object models.

The sphere model was also applied to the data-set and due to the discrete nature of the model, two objects were always favoured. Again, models with a larger number of objects were favoured when the model uncertainty was set equal to zero.

A comparative analysis into the impact that the prior distributions have on the convergence to models of different complexity was made. The general rule being that prior distributions that allow for large volume objects will bias the convergence towards the largest, and often the simplest model with which to fit the data-set. It is thought that the effects of altering the prior probability distributions will be minimised when faced with a less atypical data-set, where objects are not usually identical in mass and symmetrically positioned. Any effect due to the prior distribution should be further minimised as the measurement uncertainty decreases with the future use of quantum gravity instruments.

Chapter 7

Conclusions and further work

This research concerned the question of inference of near-surface structure from terrestrial gravity and gravity gradient data. Given a set of experimental measurements and appropriate prior knowledge, what information can we infer about any gravity anomalies present? Such questions arise in civil engineering applications when searching for, often complex in shape, anomalies of interest such as pipelines, mine-shafts and tunnels. These applications require information to be inferred about the anomaly structural extent and depth but also of importance is the overall uncertainty associated with the inference process outputs. Only with knowledge of this uncertainty can qualified decisions be made from the inference process.

A bespoke reversible-jump Markov chain Monte Carlo algorithm was developed in MATLAB (see Appendix C) to address the gravity inference problem. The Bayesian inference algorithm explores the model space and returns an ensemble of models that fit the data with a high likelihood, given any prior information we may have. The ensemble of models are draws from the Bayesian posterior distribution, which contains all of the information available for our inference problem. Once we have obtained this distribution, we calculated summary statistics for individual model parameters, such as the posterior mean, maximum and 95% credibility intervals. These statistics allowed us to produce outputs which would make intuitive sense to an end user concerned with the probability of finding an anomaly at a given location.

Three main forward models were the focus of the research; that of the finite horizontal cylinder (representing pipeline anomalies), that of the cuboid (representing tunnel systems) and

that of the sphere representing ignorance of the model required. These were chosen for their direct application to anomalies of interest. The work was focussed on simple anomalies that can be described in terms of multiples of the cylinder or parallelepiped geometries. For this reason it was deemed necessary to allow the algorithm the freedom to explore models consisting of a changeable number of parameters, through the 'birth' and 'death' moves of the reversible-jump framework (as discussed in Section 3.2.2). The research was carried out with future quantum atom interferometer gravity gradiometer instruments in mind, although without any actual datasets to analyse. The quantum instrument, whose workings were briefly described in Section 2.3.2 will allow the cancellation of environmental noise which could increase the measurement accuracy by an order of magnitude for near-surface anomalies. Where appropriate throughout the thesis, comparisons of synthetic data between the current generation of spring based gravity instruments and the future quantum gradiometer were made. As soon as a quantum gradiometer is available for field measurements, comparisons can be made with datasets taken during this research.

The problem of ambiguity inherent in all potential field data taken at the surface could only be addressed by introducing suitable prior information. Bayes' theorem provided a natural method of doing this via the parameter prior probability distributions. Here, the density contrast parameter was assumed known to some level of uncertainty, information represented by a Gaussian prior distribution. This, combined with the varying level of prior information provided by the choice of forward model, combine to limit the model space to physically plausible regions. Although we are accounting for the uncertainty in the average ground density contrast, we assumed that the spatial variation in the near-surface ground density is negligible. It is acknowledged that this may not be the case in reality. The extent of soil density variation in the near-surface is a potential area for further study and may be one of the limiting factors for inference from high accuracy measurements made by a future quantum atom interferometer gravity gradiometer due to the amplification of short wavelength (near-surface) anomalies.

The Bayesian reversible-jump Markov chain Monte Carlo algorithm was demonstrated to

work well with noisy synthetic data in a range of anomaly scenarios. Chapter 4 discussed the algorithm workings in detail. Much experimentation with various Markov chain Monte Carlo schemes was conducted to arrive at the algorithm described here; as we required an algorithm that could perform well for different forward models, ensuring reasonable mixing and convergence whilst keeping computational time as short as possible.

The simple example of a single buried void was investigated in Section 5.1, the geometry and dimensions of which were the same as the nuclear bunker anomaly explained in Appendix A.2. The data were generated using knowledge obtained from the site, and from the one line of gravity data we managed to obtain. A Gaussian noise with $\sigma = 3 \times 10^{-8}$ ms^{-2} was added to the synthetic data. This value represents an uncertainty that we might expect from a typical small scale gravity survey carried out with the Scintrex CG-5.

Firstly, the reversible-jump Markov chain Monte Carlo algorithm was run using the cuboid forward model with a maximum of $k = 10$ model objects. Combining ten runs, we were able to show convergence to the simplest forward model to explain the data with $k = 1$. As the cuboid forward model uses eight parameters, a total of ten parameters were used including the model error and offset to describe the gravity data. Convergence of the separate runs was determined by the Geweke diagnostic (as outlined in Section 4.1.4). The converged marginal parameter posterior distributions were displayed in trace and histogram form. A more intuitive representation of the spatial information contained in the marginal posterior distributions was given by three dimensional plots of one hundred posterior models, each of equal transparency. Such plots give a sense of the positional uncertainty of model objects relative to measurement points, with darker regions corresponding to higher posterior probability areas. These plots, or a more robust form of, could be used to convey pertinent information to end users, perhaps through augmented reality applications. We explored the correlation between the cuboid model parameters. It was found that the shortest side length and the height were highly correlated. The gravity data are least sensitive to the height of the anomaly, and the smallest side length is less well defined by the data than the longer length.

We also approached the problem from a position of decreased prior information by using the sphere model. With the same uniform prior over the number of model parameters k, ten runs of the reversible-jump Markov chain Monte Carlo algorithm converged to two objects with which to describe the data. As the sphere model contains five parameters per object the total number of parameters for the converged model was twelve, compared to the ten used by the single cuboid model. We observed correlation of parameters between the two objects, namely the x centroid coordinates for this example. The sphere depths matched the known depth of the cuboid anomaly within the credibility intervals, showing that useful information can be obtained from the inference algorithm when using a simpler forward model.

It was recognised that the key output that could be deduced from the separate marginal distributions was the total mass of the anomaly. This combination of density and volume of all model objects allows comparison of models of differing type and complexity. This, combined with the three dimensional spatial plots, sidesteps the label switching issue associated with reversible-jump Markov chain Monte Carlo. The total mass of the two object sphere model was shown to approximate the actual mass of the anomaly more accurately than the one object cuboid model, although the latter was still found within the credibility intervals.

A comparison with the corresponding gravity gradient data for this buried void anomaly was made. The synthetic data was generated as outlined in Section 2.2.3, with an added Gaussian noise of 0.5×10^{-8} s^{-2}. A value deemed reasonable for initial surveys with a field robust atom interferometer gradiometer instrument. Due to the increased signal-to-noise ratio of the gravity gradient data the marginal posterior probability distributions had smaller credibility intervals for the cuboid model, relative to inference from the gravity data. Again, the simplest $k = 1$ number of objects was found. When using the sphere model with the gravity gradient data, we saw an increase from $k = 2$ for the gravity data to $k = 3$. This result shows the natural parsimony of the Bayesian reversible-jump Markov chain Monte Carlo algorithm, which finds the simplest model with which to fit the data within the total uncertainty. A consequence of the natural parsimony is the ability to obtain depth information without the need for arbitrary regularisation associated with other methods.

In Section 5.2 we analysed a more structurally complex 'dogleg' pipeline anomaly with the reversible-jump Markov chain Monte Carlo algorithm. The anomaly was representative of a gas pipeline, which can have diameters of around 1.2 metres. The data consisted of two cylinder segments, which were rotated at an angle of 45 degrees. One of the segments was fully contained within the measurement grid, the other was semi-infinite. Semi-infinite anomalies may be the general case for small scale field gravity surveys where the extent of the measurement grid is generally limited. The data were generated using the cylinder forward model (Section 2.2.2), with a Gaussian uncertainty added as for the single void anomaly. Assuming knowledge of the cylinder forward model first, the reversible-jump Markov chain Monte Carlo algorithm was run for ten separate chains. The simplest solution of two cylinders was found with close fits of the cylinder depth and radius to the known values. Each cylinder uses seven separate parameters, giving a total of sixteen parameters used to fit the data. As with the sphere model the two separate cylinder object parameters were correlated, with the position of the vertex being a source of ambiguity. The issues raised by the semi-infinite nature of the anomaly were discussed. The infinite degeneracy of the cylinder length and centroid position caused problems when combining the total mass posterior distribution between separate runs. Different runs converged to different values of cylinder length, making interpretation of the total mass difficult.

The sphere model was also applied to this problem. Ten runs of the algorithm showed a preference for four objects ($k = 4$) to explain the data. Four spheres gives a total of twenty two model parameters. The problems raised by the semi-infinite anomaly are handled well by the sphere model and the total mass converges to the same value for separate runs. The trade off is decreased knowledge of the spatial extent of the anomaly as well as the depth being less well resolved.

Synthetic gravity gradient data was generated and fit using the sphere model, where we found a preference for $k = 8$ model objects due to the increased signal-to-noise ratio. With forty two parameters now describing the model, the number of iterations had to be increased in order to obtain satisfactory convergence.

The synthetic data generated for the two anomaly scenarios was idealised; regularly spaced measurements taken on a flat plane, only corrupted by Gaussian uncertainty with an equal standard deviation for each measurement point. In Chapter 6 the experimental process involved in the acquisition of real world gravity data using the Scintrex CG-5 was described. The outcome of the survey conducted over multiple days, was a dataset that was irregularly spaced, with variable data uncertainties between measurement points.

The anomaly was constructed above ground, where we set-up two stacks of concrete blocks of known dimensions and density. As we measured about 1.5 m above the ground, we limited the possible effect of soil noise variation and other complications. We applied the standard gravity corrections outlined in Sections 2.4 and 6.1.2. The data uncertainty was determined by the spread of the base station values taken on a given survey day. The data set obtained matched that predicted by synthetic data well, within the data uncertainty. Indicating a potential method of comparing gravity survey instruments, or testing operator competency. The data was input into the reversible-jump Markov chain Monte Carlo algorithm. Various measurement uncertainty values were used, along with setting the model uncertainty parameter equal to zero. The outcome of the various different runs was a similar to what we might have expected based on our initial synthetic investigations.

The total uncertainty was found to be crucial in determining the number of model objects k. When inferring the extra model uncertainty parameter, we saw a preference for one cuboid object to fit the dataset, the simplest model within the large total uncertainty. When setting this value to zero, we saw a clear preference for models with $k = 2$.

The model uncertainty parameter gives advantages and disadvantages to the inference algorithm. It was consistently seen that multiple runs with the parameter tended to meet the convergence diagnostics more readily than equivalent runs where it was set equal to zero. However, it also makes it easier for the algorithm to give what is perhaps an over-simplified model, by simply increasing the model uncertainty to higher values. Whether this parameter is even appropriate to apply to a measurement survey taken over multiple days is a reasonable question to consider. Perhaps it should be seen as a useful descriptive hyper-parameter

which has no real physical meaning, akin to the number of objects k. More stringent prior constraints on the model error could be enforced, but it is difficult to see a situation where such constraints could be justified, especially for gravity surveys.

The sphere model was also applied to the real world data, but it's appropriateness for this dataset was limited by the very shallow anomalies. With the model uncertainty as an unknown parameter, the sphere model converged to $k = 2$, due to the discrete nature of the sphere. However, the total mass estimate was very poor when compared with the approximate known value. When setting the model uncertainty to zero, we found a preference for two, three or four anomalies. Giving a closer, but still incorrect, estimate of the total mass.

The prior effect was investigated for the cuboid and sphere models as applied to the real world data. It was found that even small changes made to the prior distributions could cause a large change in the model complexity to which the algorithm converged. This was due to a combination of the atypical data obtained in the concrete block experiment (identical anomalies aligned symmetrically) and the large uncertainty of the gravity data. Data with lower uncertainty should reduce this problem, although the extent to which it will be a significant problem for real underground anomalies is unknown. This problem could be tackled with more complex reversible-jump updates. Split-merge moves[114] could be used to try to aid transitions between one large object and two small objects (and vice versa). A split would be accomplished by splitting a large object and creating two smaller objects with parameters related to the initial object (maintaining position and volume would increase the probability of acceptance). And a merge would merge two nearby objects into one larger object, whose parameters are similarly related to the two initial objects.

The general applicability of the reversible-jump Markov chain Monte Carlo algorithm was shown by fitting a range of synthetic anomalies using different forward models. The algorithm in its current form could feasibly be used for initial interpretation of a coarse preliminary gravity survey of a site. With inference using the sphere model showing areas of interest which may require further measurements and investigation. To become a reasonable tool for practical inference the computational efficiency of the algorithm needs to be increased to

allow on site inference as data is collected. Currently it takes around twenty minutes for a run of 200,000 iterations, this can rise substantially for models with many objects which may require even more iterations to reach convergence (as seen in Section 5.2.3). The room for improvement of the computation time is significant if the algorithm is migrated to a lower level coding language.

Linked to the algorithm efficiency, other problems associated with the reversible-jump Markov chain Monte Carlo algorithm need to be addressed in future work. Due to the very small acceptance ratio for the birth and death moves, we essentially waste half of the iterations attempting an update that in all likelihood will fail. A possible solution to this would be to implement the reversible-jump Markov chain Monte Carlo algorithm, and when the model object number k reaches a stationary value, convert the algorithm to the standard Markov chain Monte Carlo to complete the run, thus avoiding the problem of wasted iterations.

It may have been noted that the algorithm varies the average ground density in the forward model, when we have already assumed an average ground density for the Bouger correction of our measurement data (as detailed in Section 2.4.2). The Bouger corrections used for the concrete block data were minimal as the CG-5 sensor was far from the ground and the relative height difference between the minimum and maximum measurement height was only approximately 0.25 m. For a complete treatment the Bouger correction should be incorporated into the forward model of the algorithm to accurately represent the change of density on the individual measurement values.

Future work should be focussed around the application of the inference algorithm to real world data of underground anomalies taken using both the Scintrex CG-5 and the quantum atom interferometer gradiometer (attempts were made in this research to obtain such datasets, outlined in Appendix A, which were unsuccessful). With a two dimensional measurement grid over a known buried anomaly, we will be able to assess how well the available prior information on ground density can fit the measurement data. We should also be able to investigate the extent to which the simple shape forward models can cope with issue of spatially correlated soil density variations. The forward models may have to be made more

complex, to incorporate an average density parameter that is a function of the object centroid position in the subsurface. With the hope of further improving the inference outputs, other geophysical data-sets such as ground penetrating radar, magnetometry and borehole logs could be incorporated into the current inference algorithm via the parameter prior distributions or through a 'joint-inversion' approach which infers model parameters from multiple geophysical data-sets simultaneously[115][109].

Appendix A

CG-5 Field Data

During the research project there were multiple field trials carried out using the Scintrex CG-5 that were unsuccessful or incomplete. It may be of use to describe these experiments briefly in order that they may be repeated in the future with a quantum instrument, or indeed with the Scintrex CG-5. All of the experiments listed lend themselves to the Bayesian inference process outlined in this research.

A.1 University Utility Tunnels

An early measurement campaign was undertaken above utility tunnels at the University of Birmingham. Figure A.1 shows the top of the tunnels during their installation. We can see that they are reasonably close to the surface (less than 2 m) and they are approximately 2 m wide. We know from University plans that they are also approximately 2 m deep to the centre. The tunnels are encased in reinforced concrete about half a metre thick. An experiment with multiple Scintrex CG-5's was carried out, figure A.2 shows the three instruments measuring beneath bespoke 'wind-shields' (upturned recycling bins to the trained eye). Unfortunately, due to insufficient quality controls, the data obtained (see Figure A.3) was unusable for inference purposes. The instrument tilt was too high, as the instruments were not given enough time to settle on the tripod before initialising a measurement. There appears to be some linear feature in the dataset, of the order of the expected signal given the anomaly

Figure A.1: Photograph of the utility tunnels during their installation. Photograph c/o David Chapman.

Figure A.2: Three Scintrex CG-5 instruments protected from wind noise by shielding.

information. This could be a promising survey site with which to test new instruments.

A.2 Broadway Tower Nuclear Bunker

A measurement campaign was conducted at the Broadway tower nuclear bunker in Worcestershire, UK. We were able to measure the location, depth and dimensions of the bunker,

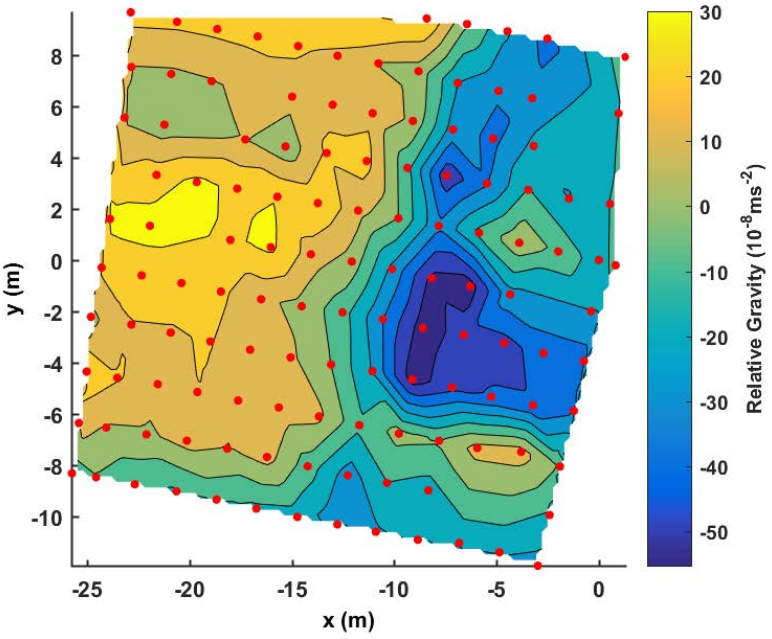


Figure A.3: Plot of gravity data obtained at the University of Birmingham utility tunnels. Poor quality control adherence rendered the dataset unusable.

allowing us to accurately model the expected gravity signal, given an assumption on the ground density. The centroid of the anomaly was approximately 2.5 m below the surface, with dimensions of $5.5 \times 2.25 \times 2.25$ m. Figure A.4 shows the site, a 10 m line of gravity measurements was taken bisecting the anomaly running parallel with the shortest side. Figure A.5 shows the dataset obtained along with various models fit to the data with the known dimensions of the anomaly. The gravity signal obtained was larger than expected when modelling with a soil density contrast of -2000 kgm^{-3}. This result may have a bearing on the detection of near surface man made anomalies with the gravity method. It is believed that the construction method used to build the bunker ('cut and cover') leaves the soil around the anomaly with a lower density than the surrounding undisturbed soil. As such there is a 'halo' of low density soil around and above the anomaly. More investigation is required to assess the validity of this theory, involving the analysis of suitable soil samples to see if the density variation is a reality. In figure A.5, the purple line shows the expected gravity signal due to a single cuboid void with an average ground density of 2000 kgm^{-3}. The yellow and orange lines are models of increasing density, the density may lie somewhere between

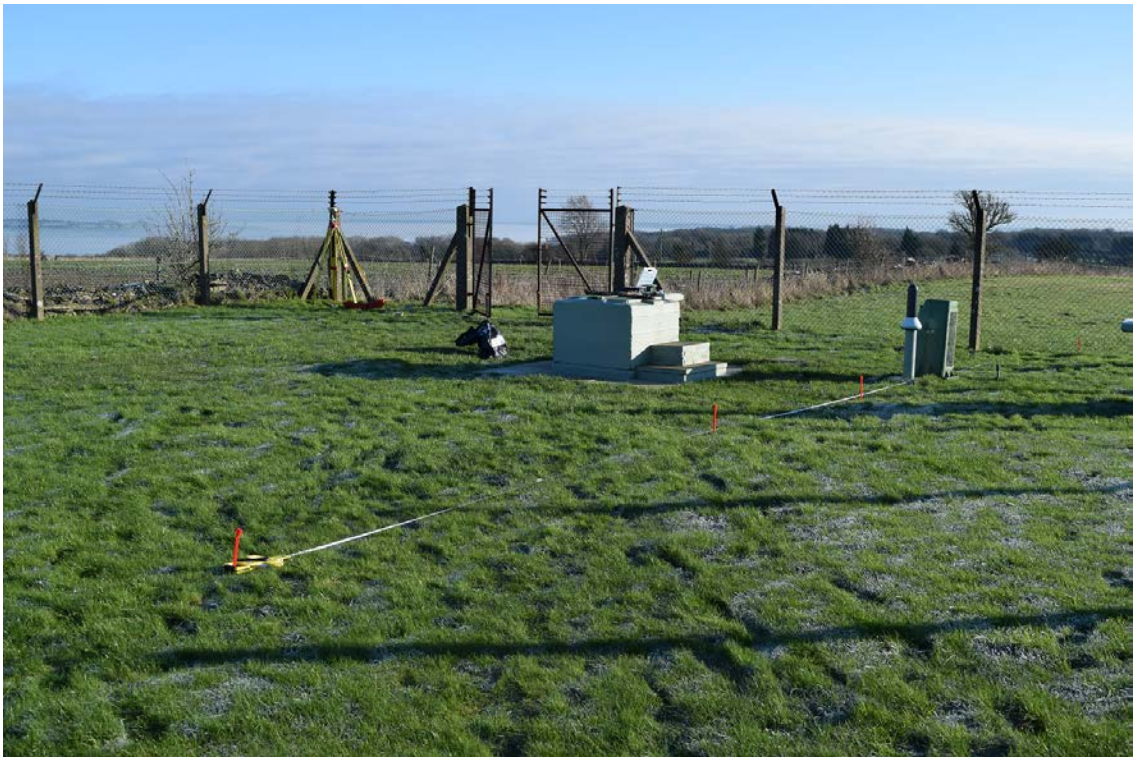


Figure A.4: Picture of the fenced off nuclear bunker site at Broadway tower. A line of gravity data was taken in a line parallel with the centre of the picture.

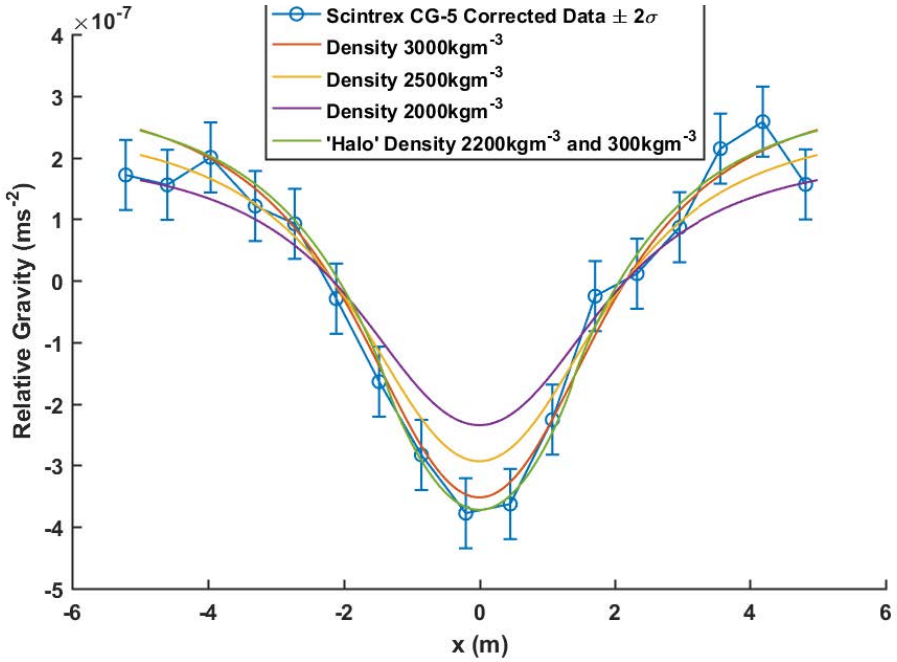


Figure A.5: Scintrex CG-5 dataset obtained over the nuclear bunker anomaly. The yellow line represents a model accounting only for the known anomaly. The orange model represents the known anomaly plus a 'halo' of low density soil.

-2500 kgm^{-3} and -3000 kgm^{-3}. The green line represents a model with an outer halo, extending past the anomaly half a meter in the x and y directions and to the surface in the z direction. Given the addition of gravity anomalies it is possible to model the density of the actual void as -2200 kgm^{-3} and the halo as -300 kgm^{-3}. This halo model is illustrated in figure A.6. Borehole records obtained from the British Geological Survey (BGS) website [68]

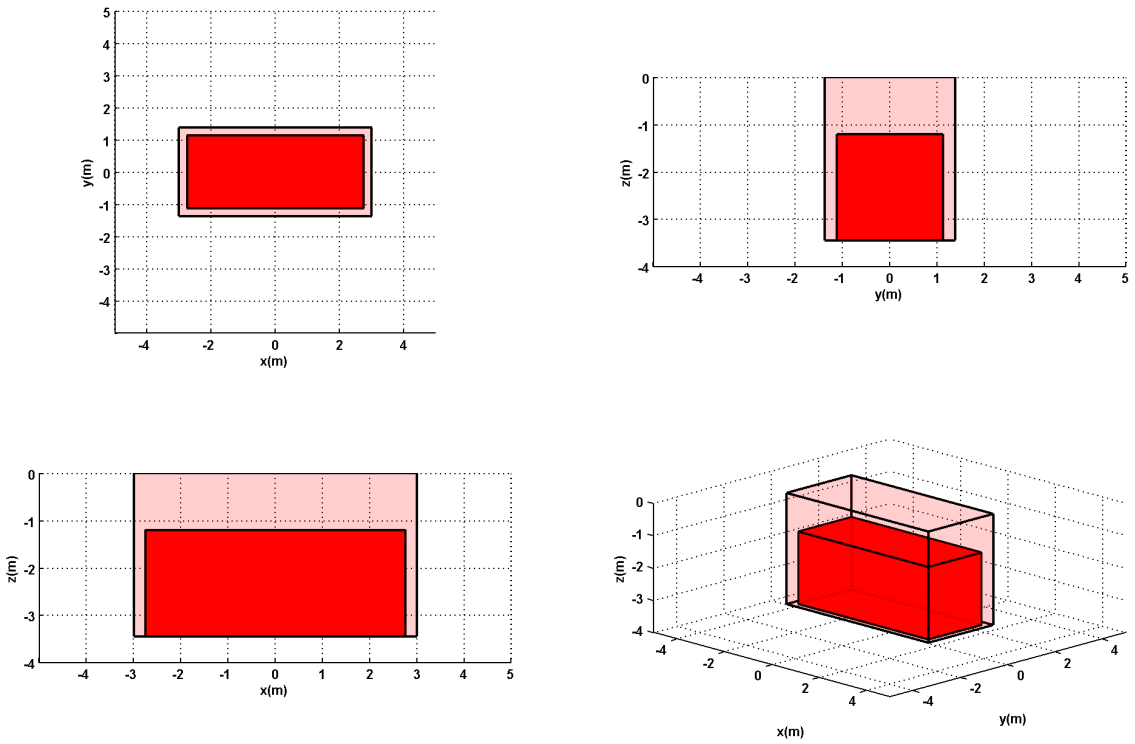


Figure A.6: 'Halo' model for the Broadway Tower nuclear bunker anomaly. The known anomaly is modelled with a density of -2200 kgm^{-3}. The halo is modelled 0.5 m larger in the x and y directions, extending up to ground level in the z direction, with a density of -300 kgm^{-3}. Giving an effective anomaly density of -2500 kgm^{-3}. The green line in Figure A.5 shows the gravity signal.

(BGS ID: 309546 : BGS Reference: SP13NW16), show that the near surface is composed of limestone with only 0.3 m of top soil at the nearby borehole site. If this holds for our site, the density could be anywhere within the reasonable range for limestone, $1930 - 2900$ kgm^{-3} with an average of 2550 kgm^{-3}[43]. Further support for this comes from another BGS online tool[116] which gives the site a soil thickness rating of 'shallow'. Meaning that; 'A thin soil profile is likely. Digging the Parent Material beneath the soil will be extremely difficult

at a depth of 0.5 m (or possibly less)'. More measurements would be required to obtain a grid sufficient for parameter estimation using the reversible-jump Markov chain Monte Carlo algorithm, without having to make simplifying assumptions to force the model to become two dimensional. It may be feasible to alter the Bayesian algorithm to search for the density

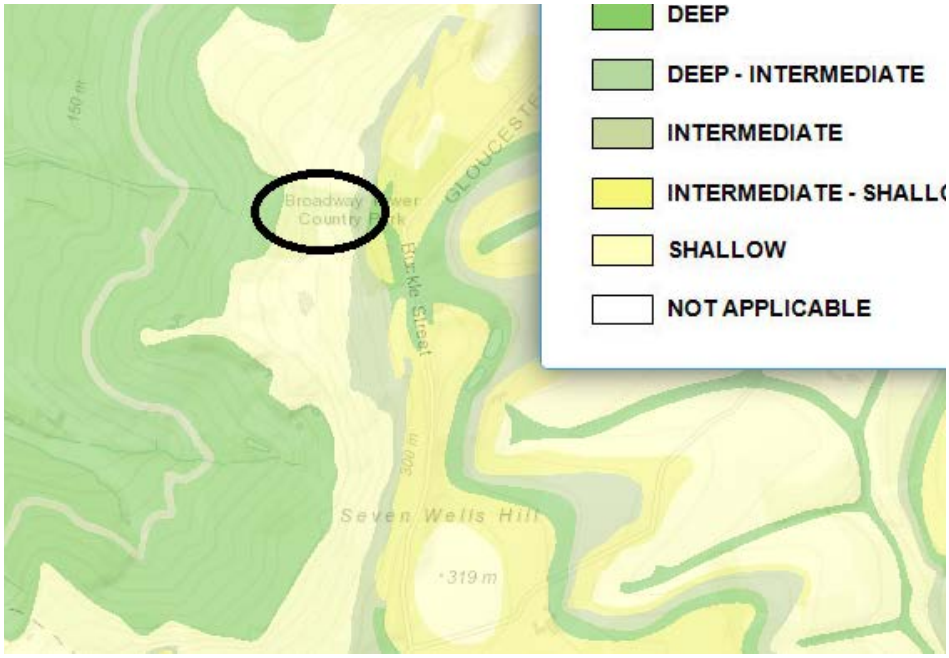


Figure A.7: Soil depth at the Broadway Tower nuclear bunker site (black oval).

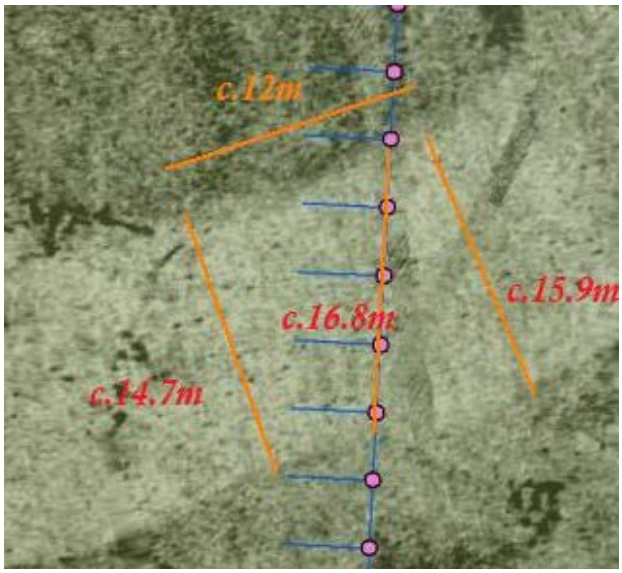
values given our prior information of the anomaly location to best find the value of the halo density. However a 2D grid of gravity data would be required to make robust conclusions. Section 5.1 uses the dimensions of this anomaly to demonstrate the Bayesian algorithm for a simple void, ignoring the speculated halo effect and taking the average ground density as -2700 kgm^{-3}.

A.3 Woodhenge Ditch

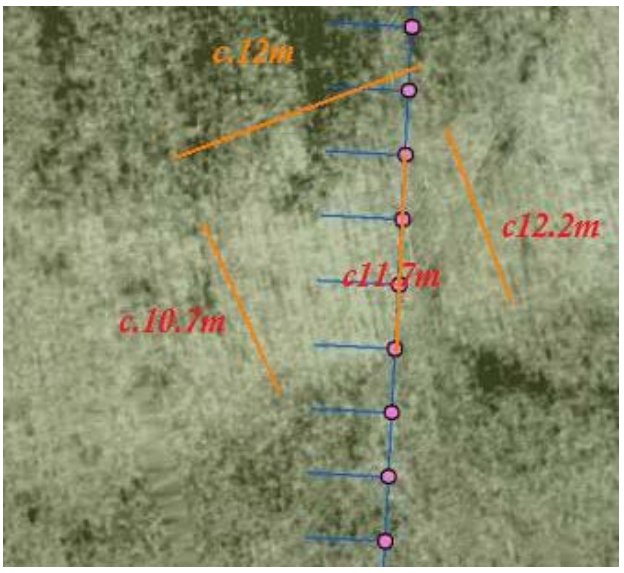
The Woodhenge National trust site in Durrington has been extensively explored with various near-surface geophysics techniques. It is located near the famous Stonehenge site, Figure A.8 shows the proximity. Ground penetrating radar scans revealed a previously unknown ditch, cut into the chalk bedrock. Time-slices are shown in Figure A.9. This particular

Figure A.8: Location of measurements (red marker) taken at Durrington walls (marked). A few kilometres away from Stonehenge.

GPR survey was unable to determine the depth of the anomaly. Previous experience of such



(a) GPR time-slice 0.25 m depth. (image c/o Eamonn Baldwin)



(b) GPR time-slice 2.5 m depth. (image c/o Eamonn Baldwin)

Figure A.9: GPR time-slice of Woodhenge ditch. Approximate gravity station locations are shown in pink. Gravity data is shown in figure A.10

anomalies suggests that the depth of the ditch is approximately $5 - 6$ m. Taking a line of gravity measurements across the anomaly we obtained the dataset shown in Figure A.10.

The purple fit represents an estimated model of a cuboid, making use of the GPR data and

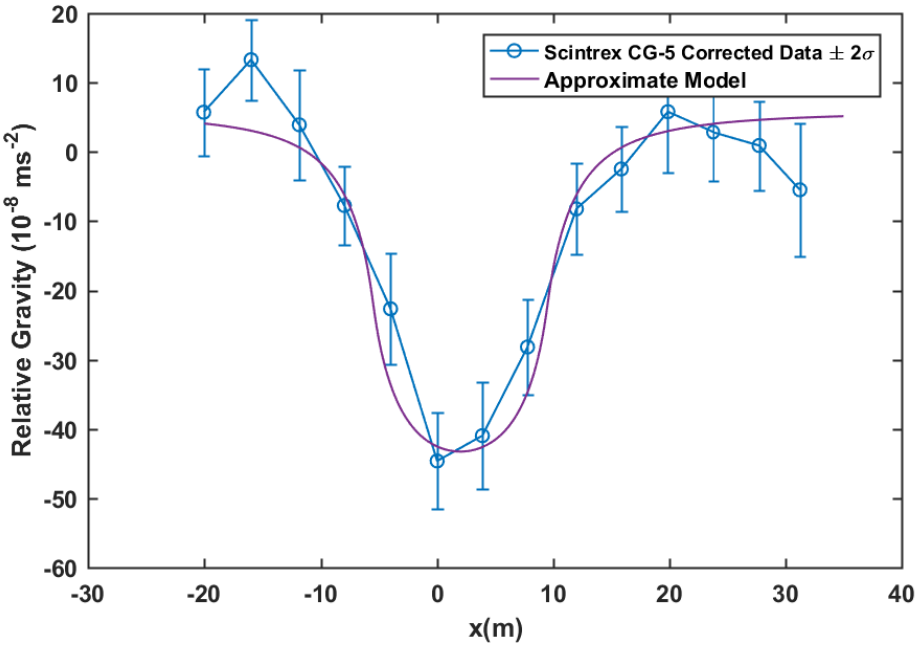


Figure A.10: Gravity data obtained using the Scintrex CG-5. Purple model represents a reasonable estimate of the anomaly.

past experience. The density difference between chalk and top-soil was taken as -300 kgm^{-3}, the width of the anomaly as 14 m with depth 5 m. The cuboid model is an approximation to what is clearly a ditch with sloping banks. Time constraints meant that further data lines could not be obtained, this site could be the focus of future surveys. A more detailed forward model may need to be implemented in order to model the sloped sides of the ditch.

Appendix B

Sphere Model Approximation

Here we equate the volume of a chain of spheres to that of a cylinder, comparing the gravity signals to show that the sphere model provides a good approximation to the cylinder model at distances greater than the diameter. Figure B.1 shows the model of multiple spheres inside a cylinder.

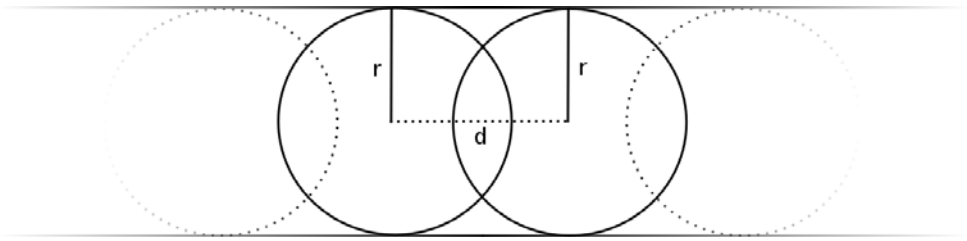


Figure B.1: Equating the volume of multiple spheres to that of a cylinder segment.

We equate the volume of a sphere and cylinder of length equal to the distance d. So, $\frac{4}{3}\pi r^3 = \pi r^2 d \rightarrow d = \frac{4}{3}r$. Using this result, a chain of spheres can approximate a cylinder anomaly. This result may be useful for developing forward models of realistic pipelines, that are inevitably not perfectly cylindrical but suffer from bends and kinks.

When using this approximation to fit a cylinder as in Section 5.2.2, the model was essentially finding the maximum distance between spheres d such that the model meets the measurement data within the given uncertainty. Figure B.2 shows contours of a buried cylinder and buried spheres at the same depth (5 m). The cylinder has a radius $R = 0.6$. The spheres are separated by a distance of 3.2 m. The radius of the spheres increases to make up the volume

deficit, a value $R = 0.97$ gives a good approximation of the cylinder data.

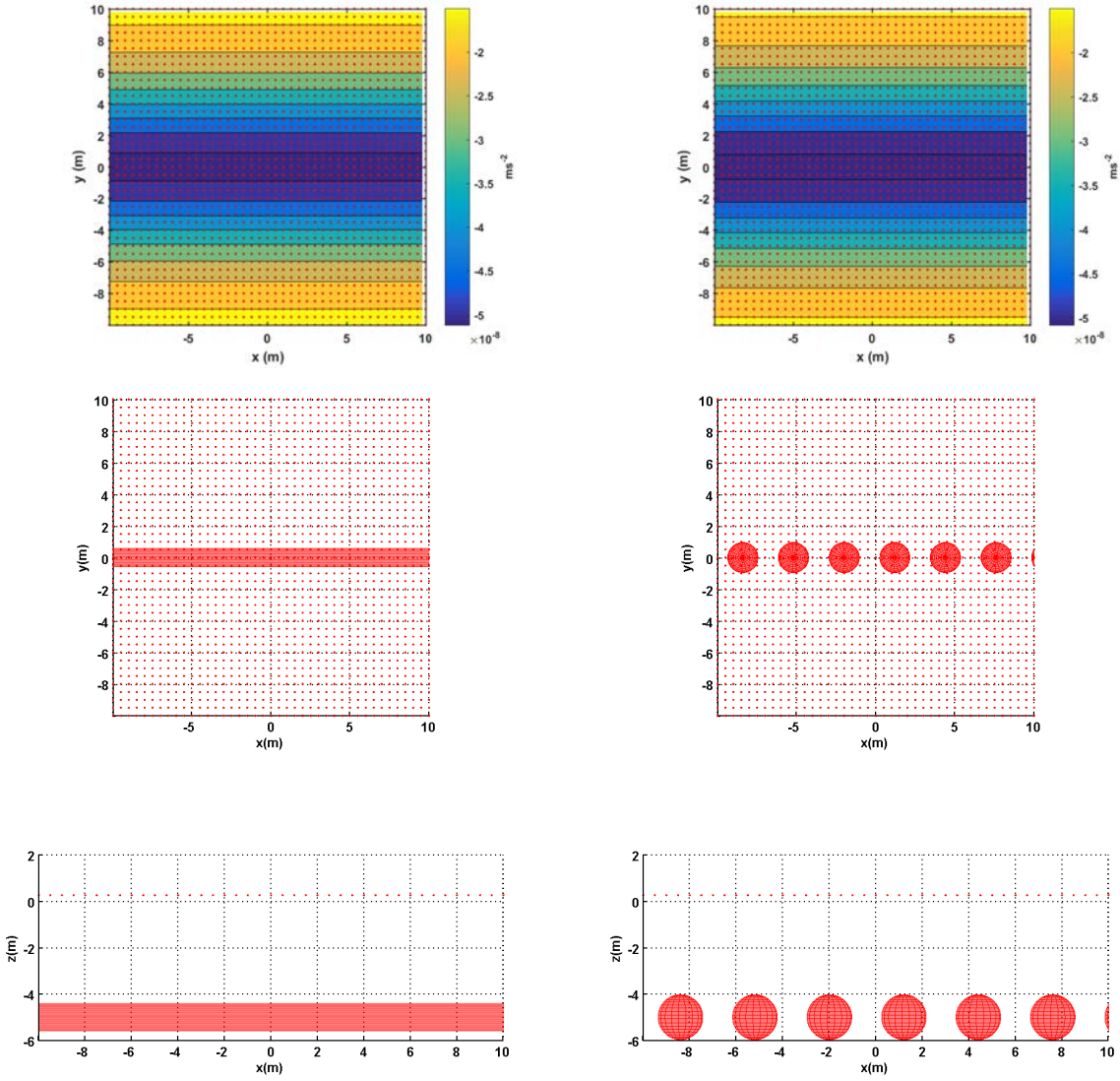


Figure B.2: Comparison of models of a cylinder and the approximation with multiple spheres.

The approximation becomes less accurate with increasing distance d and as the depth of the anomaly approaches the measurement plane ($z = 0$).

Appendix C

MATLAB Code

The code was in a constant state of flux with analysis of various different datasets. It is simple to switch between inference from gravity and gravity gradient data, likewise for the three forward model types. Shown here are the main functions used for inference from gravity data using a cuboid model.

Main body of code where the user defined parameters are set. The chosen dataset is loaded.

```
%% Bayesian Inference from Gravity and Gravity Gradient Data
%-------------------------------------------------------------------------%
% created by Anthony Rodgers 2015
% if you wish to use or publish this code please contact the author
% adr328@gmail.com, +447799018559
%-------------------------------------------------------------------------%
%% Import dataset
load('concrete_line_quadratic_reasonableqc_12sd_15arcsec_5rej_deloutliers_...
consistency_combined')
final_table_data = final_table_data_combined;
labels = final_table_data(:,1);
x_coords = final_table_data(:,2);
y_coords = final_table_data(:,3);
z_coords = final_table_data(:,4);
% Scintrex data
gravity = (final_table_data(:,17) - mean(final_table_data(:,17))).*10^-8;
% Scintrex BS error
error_sensor = (final_table_data(:,13));
min_height = min(z_coords);
z_coords = z_coords - min_height;
[x_coords,y_coords,z_coords] = coordinate_rotation(0, 0,...
    deg2rad(-27), x_coords,y_coords,z_coords);
% Subtract data mean
gravity = gravity - mean(gravity);
no_datapoints = size(gravity,1);
%% Set MCMC Parameters
save_name = 'final_concrete_data_SPHERE_infererror_';
density_mean = 2;
density_std = 0.1;
density_prop_scaling = 1;
mcmc_iterations = 200000; % Number of iterations performed-
gravity_derivative = 1; % 1 for gz 2 for gzz
% 1 for cylinder 2 for parallelepiped 3 for spheres
forward_model_type = 3;
infer_error = 1;
adaptive_mcmc = 1;
% Decrease, increase, lower bound, upper bound
adaptive_mcmc_params = [0.9,1.1,0.1,0.4];
% Number of iterations before adaptive MCMC update
adaptive_mcmc_counter = 1000;
sim_anneal = 1;
% Use Laplace likelihood instead of Gaussian
laplace_likelihood = 0;
% Heavy tailed proposal distribution
cauchy_prop = 1;
% Subtract mean from McMC forward model
subtract_mean = 1;
% 1 in x or 1 in x-1 chance of a RJ update
mh_upper_rand = 2;
% Start value and constant value
simulated_annealing = [1000000,0.99999];
% Factor of data discarded
burn_in_discard_percent = 0.25;
no_min_objects = 1;
if forward_model_type == 3
    no_max_objects = 10; % Max number of allowed objects for TDMCMC
else
    no_max_objects = 10;
```

```
end
proposal_scaling = 0.5;
ai_separation = 0.5; % Distance between atom clouds
no_chains = [15,25];
suspect_convergence_limit = 3;

%% Setting prior and proposals
x_min = min(x_coords);
x_max = max(x_coords);
y_min = min(y_coords);
y_max = max(y_coords);
x_size = x_max - x_min;
y_size = y_max - y_min;
max_size = max([x_size,y_size]);
min_size = min([x_size,y_size]);
min_size_capped_depth = min([min_size,10]);
min_size_capped_zlength = min([min_size,5]);
total_min = min([x_min,y_min]);
total_max = max([x_max,y_max]);
if gravity_derivative == 1
    blanket_proposal_scaling = 0.1;
else
    blanket_proposal_scaling = 0.1;
end
%% Proposals
clear proposal_std prior_distributions
if forward_model_type == 2
    % Parallelepiped
    param_block_1 = [1,2,7]; % Position params
    param_block_2 = [3,4,5,6,8]; % Volume density depth
    proposal_std(:,1) = 0.5.*blanket_proposal_scaling;
    proposal_std(:,2) = 0.5.*blanket_proposal_scaling;
    proposal_std(:,3) = 1.*blanket_proposal_scaling;
    proposal_std(:,4) = 1.*blanket_proposal_scaling;
    proposal_std(:,5) = 1.*blanket_proposal_scaling;
    proposal_std(:,6) = 1.*blanket_proposal_scaling;
    proposal_std(:,7) = 0.5.*blanket_proposal_scaling;
    if gravity_derivative == 1
        proposal_std(:,8) = 1.*blanket_proposal_scaling*density_prop_scaling;
    elseif gravity_derivative == 2
        proposal_std(:,8) = 1.*blanket_proposal_scaling*density_prop_scaling;
    end
    if subtract_mean == 1
        if gravity_derivative == 1
            proposal_std_offset = 0.5.*blanket_proposal_scaling;
            proposal_std_error = 0.5.*blanket_proposal_scaling;
        elseif gravity_derivative == 2
            proposal_std_offset = 0.05.*blanket_proposal_scaling;
            proposal_std_error = 0.05.*blanket_proposal_scaling;
        end
    elseif subtract_mean == 0
        if gravity_derivative == 1
            proposal_std_offset = 0.5.*blanket_proposal_scaling;
            proposal_std_error = 0.5.*blanket_proposal_scaling;
        elseif gravity_derivative == 2
            proposal_std_offset = 0.05.*blanket_proposal_scaling;
            proposal_std_error = 0.05.*blanket_proposal_scaling;
        end
    end
    % Priors
    prior_distributions(1) = makedist('Normal','mu',0,'sigma',x_max);
    prior_distributions(2) = makedist('Normal','mu',0,'sigma',y_max);
    prior_distributions(3) = makedist('Gamma','a',2,'b',1);
    prior_distributions(4) = makedist('Gamma','a',2,'b',1);
    prior_distributions(5) = makedist('Gamma','a',2,'b',1);
    prior_distributions(6) = makedist('Gamma','a',2,'b',1);
    prior_distributions(7) = makedist('Normal','mu',0,'sigma',0.25);
    if infer_error == 1
        prior_distribution_error = makedist('Gamma','a',2,'b'...
        ,mean(error_sensor));
    else
        prior_distribution_error = makedist('Normal','mu',0,...
        'sigma',0);
    end
    prior_distributions(8) = makedist('Normal','mu',density_mean,...
    'sigma',density_std);
    if subtract_mean == 1
        prior_distribution_offset = makedist('Normal','mu',0,'sigma',...
        mean(error_sensor));
    else
        prior_distribution_offset = makedist('Normal','mu',0,'sigma',...
        ((max(gravity) - min(gravity))/2).*10^8);
    end
elseif forward_model_type == 1
    % Cylinder
    param_block_1 = [1,2,6];
    param_block_2 = [3,4,5,7];
    proposal_std(:,1) = 0.5.*blanket_proposal_scaling;
    proposal_std(:,2) = 0.5.*blanket_proposal_scaling;
    proposal_std(:,3) = 1.*blanket_proposal_scaling;
    proposal_std(:,4) = 1.*blanket_proposal_scaling;
    proposal_std(:,5) = 0.5.*blanket_proposal_scaling;
    proposal_std(:,6) = 0.5.*blanket_proposal_scaling;
    if gravity_derivative == 1
        proposal_std(:,7) = 1.*blanket_proposal_scaling*density_prop_scaling;
```

```
154     elseif gravity_derivative == 2
155         proposal_std(:,7) = 1.*blanket_proposal_scaling*density_prop_scaling;
156     end
157     if gravity_derivative == 1
158         proposal_std_offset = 0.5.*blanket_proposal_scaling;
159         proposal_std_error = 0.5.*blanket_proposal_scaling;
160     elseif gravity_derivative == 2
161         proposal_std_offset = 0.05.*blanket_proposal_scaling;
162         proposal_std_error = 0.05.*blanket_proposal_scaling;
163     end
164     % Priors
165     prior_distributions(1) = makedist('Normal','mu',0,'sigma',x_max);
166     prior_distributions(2) = makedist('Normal','mu',0,'sigma',y_max);
167     prior_distributions(3) = makedist('Gamma','a',2,'b',1); % Depth
168     prior_distributions(6) = makedist('Normal','mu',0,'sigma',0.25); % Rotation
169     if infer_error == 1
170         prior_distribution_error = makedist('Gamma','a',2,'b',mean(error_sensor));
171     else
172         prior_distribution_error = makedist('Normal','mu',0,'sigma',0);
173     end
174     prior_distributions(4) = makedist('Gamma','a',2,'b',4); % Length
175     prior_distributions(5) = makedist('Gamma','a',2,'b',0.5); % Radius
176     prior_distributions(7) = makedist('Normal','mu',density_mean,'sigma',...
177     density_std); % Density
178     if subtract_mean == 1
179         prior_distribution_offset = makedist('Normal','mu',0,'sigma',...
180         mean(error_sensor));
181     else
182         prior_distribution_offset = makedist('Normal','mu',0,'sigma',...
183         ((max(gravity) - min(gravity))/2).*10^8);
184     end
185 elseif forward_model_type == 3
186     % Sphere
187     param_block_1 = [1,2];
188     param_block_2 = [3,4,5];
189     proposal_std(:,1) = 0.5.*blanket_proposal_scaling;
190     proposal_std(:,2) = 0.5.*blanket_proposal_scaling;
191     proposal_std(:,3) = 1.*blanket_proposal_scaling;
192     proposal_std(:,4) = 0.5.*blanket_proposal_scaling;
193     if gravity_derivative == 1
194         proposal_std(:,5) = 1.*blanket_proposal_scaling*density_prop_scaling;
195     elseif gravity_derivative == 2
196         proposal_std(:,5) = 1.*blanket_proposal_scaling*density_prop_scaling;
197     end
198     if gravity_derivative == 1
199         proposal_std_offset = 0.5.*blanket_proposal_scaling;
200         proposal_std_error = 0.5.*blanket_proposal_scaling;
201     elseif gravity_derivative == 2
202         proposal_std_offset = 0.05.*blanket_proposal_scaling;
203         proposal_std_error = 0.05.*blanket_proposal_scaling;
204     end
205     % Priors
206     prior_distributions(1) = makedist('Normal','mu',0,'sigma',x_max);
207     prior_distributions(2) = makedist('Normal','mu',0,'sigma',y_max);
208     prior_distributions(3) = makedist('Gamma','a',2,'b',1); % Depth
209     if infer_error == 1
210         prior_distribution_error = makedist('Gamma','a',2,'b',mean(error_sensor));
211     else
212         prior_distribution_error = makedist('Normal','mu',0,'sigma',0);
213     end
214     prior_distributions(4) = makedist('Gamma','a',2,'b',0.5); % Radius
215     prior_distributions(5) = makedist('Normal','mu',density_mean,'sigma',...
216     density_std); % Density
217     if subtract_mean == 1
218         prior_distribution_offset = makedist('Normal','mu',0,'sigma',...
219         mean(error_sensor));
220     else
221         prior_distribution_offset = makedist('Normal','mu',0,'sigma',...
222         ((max(gravity) - min(gravity))/2).*10^8);
223     end
224 end
225
226 %% MCMC Process
227 for no_chain = no_chains(1):no_chains(2)
228     tic
229     [all_log_prior,all_log_likelihood,all_log_alpha,accepted_log_prior,...
230     accepted_log_likelihood,accepted_log_alpha,output_objects, output_objects_block,...
231     output_error, output_offset, output_objects_padded,output_objects_converged, ...
232     output_objects_block_converged,output_error_converged, output_offset_converged,...
233     output_objects_padded_converged,output_no_objects,no_objects_output_converged,...
234     output_data_dimension,output_data_dimension_converged]...
235         = mcmc_process(gravity, error_sensor,...
236         proposal_std, proposal_std_error, proposal_std_offset,...
237         prior_distributions,prior_distribution_error,prior_distribution_offset,...
238         no_max_objects, no_min_objects, param_block_1, param_block_2, ...
239         x_coords, y_coords, z_coords, mcmc_iterations,simulated_annealing,...
240         gravity_derivative,ai_separation,forward_model_type,infer_error,...
241         adaptive_mcmc_counter,adaptive_mcmc,sim_anneal,burn_in_discard_percent,...
242         laplace_likelihood,subtract_mean, mh_upper_rand,adaptive_mcmc_params,...
243         cauchy_prop,proposal_scaling);
244     output_time_elapsed = toc/(60*60);
245     disp(no_chain)
```

Function that carries out the reversible-jump Markov chain Monte Carlo inference.

```
function [all_log_prior,all_log_likelihood,all_log_alpha,accepted_log_prior,...
accepted_log_likelihood,accepted_log_alpha,mcmc_output_total...
,mcmc_output_block_total,output_error_total,output_offset_total,...
mcmc_output_padded,mcmc_output_converged,mcmc_output_block_converged...
,output_error_converged, output_offset_converged,mcmc_output_padded_converged, ...
no_objects_output,no_objects_output_converged,data_dimension,data_dimension_converged] ...
    = mcmc_process(gravity_data, error_sensor,...
    proposal_std_unknowns, proposal_std_error, proposal_std_offset,...
    prior_distributions,prior_distribution_error,prior_distribution_offset,...
    no_max_objects, no_min_objects,param_block_1,param_block_2, ...
    x_coords, y_coords, z_coords, mcmc_iterations,simulated_annealing,...
    gravity_derivative,ai_separation,forward_model_type,infer_error,...
    adaptive_mcmc_counter,adaptive_mcmc,sim_anneal,...
    burn_in_discard_percent,laplace,subtract_mean,...
    mh_upper_rand,adaptive_mcmc_params,cauchy_prop,proposal_scaling)
%% Defining output parameters
% Randomising the initial number of objects
no_object_params = size(prior_distributions,2);
no_objects = randi([no_min_objects,no_max_objects]);
% Accepted iteration count
accept_count_object = zeros(1,no_max_objects);
accept_count_object_block = zeros(no_max_objects,2);
accept_count_all_object_params = 0;
accept_count_offset_params = 0;
accept_count_error_params = 0;
accept_count_rj_birth = 0;
accept_count_rj_death = 0;
% Iteration count
iteration_count_object = zeros(1,no_max_objects);
iteration_count_object_parameter_block = zeros(no_max_objects,2);
iteration_count_all_object_params = 0;
iteration_count_offset_params = 0;
iteration_count_error_params = 0;
iteration_count_rj_birth = 0;
iteration_count_rj_death = 0;
% Count number of objects at each iteration
no_objects_output = zeros(1,no_max_objects);
no_objects_output_converged = zeros(1,no_max_objects);
% rj mcmc parameters
mcmc_output_total = cell(1,no_object_params*no_max_objects);
mcmc_output_converged = cell(1,no_object_params*no_max_objects);
mcmc_output_padded = cell(1,no_object_params*no_max_objects);
mcmc_output_padded_converged = cell(1,no_object_params*no_max_objects);
mcmc_output_block_total = cell(1,no_object_params*no_max_objects);
mcmc_output_block_converged = cell(1,no_object_params*no_max_objects);
% Global parameters
output_error_total = zeros(1,1);
output_offset_total = zeros(1,1);
output_error_converged = zeros(1,1);
output_offset_converged = zeros(1,1);
% Likelihood and prior
accepted_log_likelihood = zeros(1,1);
accepted_log_prior = zeros(1,1);
accepted_log_alpha = zeros(1,1);
all_log_likelihood = zeros(1,1);
all_log_prior = zeros(1,1);
all_log_alpha = zeros(1,1);
% Simulated annealing parameters
if sim_anneal == 1
    t = simulated_annealing(1);
    C = simulated_annealing(2);
else
    t = 0;
end
% Adaptive MCMC parameters
adaptive_dec = adaptive_mcmc_params(1);
adaptive_inc = adaptive_mcmc_params(2);
adaptive_lower_bound = adaptive_mcmc_params(3);
adaptive_upper_bound = adaptive_mcmc_params(4);
% Dimension parameter
data_dimension = zeros(1,1);
data_dimension_converged = zeros(1,1);
%% Randomly select a starting point for the inference process
% Randomising the initial object parameters
unknowns = zeros(no_objects,no_object_params);
prior = zeros(no_objects,no_object_params);
for i = 1:no_objects
    for j = 1:no_object_params
        unknowns(i,j) = random(prior_distributions(j));
        prior(i,j) = pdf(prior_distributions(j),unknowns(i,j));
    end
end
% Make depth negative
unknowns(:,3) = -unknowns(:,3);
offset = random(prior_distribution_offset);
prior_offset = pdf(prior_distribution_offset,offset);
if infer_error == 1
    error = random(prior_distribution_error);
    prior_error = pdf(prior_distribution_error,error);
else
    error = 0;
```

```
92      prior_error = 1;
93  end
94  log_prior = log(prod(prod(prior)).*prior_offset.*prior_error);
95  % Initialise proposal std values
96  default_proposal_std = proposal_std_unknowns(1,:);
97  proposal_std_unknowns = repmat(default_proposal_std,size(unknowns,1),1);
98  % Calculating initial forward model
99  if gravity_derivative == 1
100     if forward_model_type == 1
101         forward_model = cylinder_model_gz(x_coords, y_coords, z_coords,unknowns);
102     elseif forward_model_type == 2
103         forward_model = cuboid_model_gz(x_coords, y_coords, z_coords,unknowns);
104     elseif forward_model_type == 3
105         forward_model = sphere_model_gz(x_coords, y_coords, z_coords,unknowns);
106     end
107 elseif gravity_derivative == 2
108     if forward_model_type == 1
109         forward_model = cylinder_model_gzz(x_coords, y_coords, z_coords,unknowns,...
110         ai_separation);
111     elseif forward_model_type == 2
112         forward_model = cuboid_model_gzz(x_coords, y_coords, z_coords,unknowns,...
113         ai_separation);
114     elseif forward_model_type == 3
115         forward_model = sphere_model_gzz(x_coords, y_coords, z_coords,unknowns,...
116         ai_separation);
117     end
118 end
119 % Subtract the model mean as grav data is arbitrarily scaled
120 if subtract_mean == 1
121     forward_model = (forward_model - mean(forward_model)) + offset.*10^-8;
122 else
123     forward_model = forward_model + offset.*10^-8;
124 end
125 % Calculating initial likelihood
126 if laplace == 1
127     log_likelihood = log_laplace_model(gravity_data, forward_model, error_sensor,...
128      error, gravity_derivative);
129 elseif laplace == 0
130     log_likelihood = log_gaussian_model(gravity_data, forward_model, error_sensor...
131     , error,gravity_derivative);
132 end
133
134 for k = 1:mcmc_iterations
135     %% Setting simulated annealing temperature
136     if sim_anneal == 1
137         % Simple exponential simulated annealing schedule
138         t = t*(C^k);
139         if t < 0.001
140             % Set t = 0 when to is close to 0
141             t = 0;
142         end
143     else
144     end
145     %% Display pertinent outputs and adaptive mcmc
146     if k > 0 && k < mcmc_iterations*burn_in_discard_percent && ...
147     rem(k,adaptive_mcmc_counter) == 0
148         % Display values of interest
149         accept_ratio_all_object_params = accept_count_all_object_params./...
150         (iteration_count_all_object_params);
151         accept_ratio_offset_params = accept_count_offset_params./...
152         (iteration_count_offset_params);
153         accept_ratio_error_params = accept_count_error_params./...
154         (iteration_count_error_params);
155         accept_ratio_object_block = accept_count_object_block...
156         /iteration_count_object_parameter_block;
157         accept_ratio_object = (accept_count_object)./(iteration_count_object);
158         disp('proposal_scaling: '),disp(proposal_scaling);
159         disp('proposal std: '),disp(proposal_std_unknowns);
160         disp('proposal std offset: '),disp(proposal_std_offset);
161         if infer_error == 1
162             disp('proposal std error: '),disp(proposal_std_error);
163         else
164         end
165         disp('accept_ratio_object_total: '),disp(accept_ratio_all_object_params)
166         disp('accept_ratio_offset_param: '),disp(accept_ratio_offset_params)
167         disp('accept_ratio_error_param: '),disp(accept_ratio_error_params)
168         disp('accept_ratio_object: '),disp(accept_ratio_object)
169         disp('accept_ratio_object_block: '),disp(accept_ratio_object_block)
170         if adaptive_mcmc == 1
171             %% Adaptive proposal distribution for object param blocks update
172             for i = 1:size(unknowns,1)
173                 if accept_ratio_object_block(i,1) < adaptive_lower_bound
174                     proposal_std_unknowns(i,param_block_1) =  ...
175                     proposal_std_unknowns(i,param_block_1).*adaptive_dec;
176                 elseif accept_ratio_object_block(i,1) > adaptive_upper_bound
177                     proposal_std_unknowns(i,param_block_1) =  ...
178                     proposal_std_unknowns(i,param_block_1).*adaptive_inc;
179                 end
180                 if accept_ratio_object_block(i,2) < adaptive_lower_bound
181                     proposal_std_unknowns(i,param_block_2) =  ...
182                     proposal_std_unknowns(i,param_block_2).*adaptive_dec;
183                 elseif accept_ratio_object_block(i,2) > adaptive_upper_bound
184                     proposal_std_unknowns(i,param_block_2) =  ...
185                     proposal_std_unknowns(i,param_block_2).*adaptive_inc;
186                 end
```

```
187             end
188             %% Adaptive proposal distribution for all object update
189             if accept_ratio_all_object_params < adaptive_lower_bound
190                 proposal_scaling = proposal_scaling.*adaptive_dec;
191                 proposal_std_offset = proposal_std_offset.*adaptive_dec;
192                 if infer_error == 1
193                     proposal_std_error = proposal_std_error.*adaptive_dec;
194                 else
195                 end
196             elseif accept_ratio_all_object_params > adaptive_upper_bound
197                 proposal_scaling = proposal_scaling.*adaptive_inc;
198                 proposal_std_offset = proposal_std_offset.*adaptive_inc;
199                 if infer_error == 1
200                     proposal_std_error = proposal_std_error.*adaptive_inc;
201                 else
202                 end
203             end
204         end
205         %% Reset acceptance percentages
206         accept_count_object_block = zeros(no_max_objects,2);
207         accept_count_object = zeros(1,no_max_objects);
208         accept_count_all_object_params = 0;
209         accept_count_offset_params = 0;
210         accept_count_rj_birth = 0;
211         accept_count_rj_death = 0;
212         %% Reset iteration count
213         iteration_count_object_parameter_block = zeros(no_max_objects,2);
214         iteration_count_object = zeros(1,no_max_objects);
215         iteration_count_all_object_params = 0;
216         iteration_count_offset_params = 0;
217         iteration_count_rj_birth = 0;
218         iteration_count_rj_death = 0;
219
220         disp('unknowns: '),disp(unknowns);
221         disp('offset: '),disp(offset);
222         disp('error: '),disp(error);
223         if t > 0
224             disp('simulated annealing')
225         end
226         disp('Burn-in Period')
227         disp(k)
228     elseif k > mcmc_iterations*burn_in_discard_percent && ...
229     rem(k,adaptive_mcmc_counter) == 0
230         if infer_error == 1
231             disp('error: '),disp(error)
232         end
233         disp('offset: '),disp(offset)
234         disp('unknowns: '),disp(unknowns);
235         disp('proposed unknowns: '),disp(proposed_unknowns);
236         disp(k)
237     end
238     % Only MH moves allowed if rjmcmc 'off'
239     random_number = randi(100);
240     if (no_max_objects - no_min_objects) == 0
241         move_type = 2;
242     else
243         if random_number <= 50
244             move_type = 1;
245         elseif random_number >= 51
246             move_type = 2;
247         end
248     end
249     %% reversible jump moves
250     if move_type == 1
251         % Select birth/death move
252         b_d = randi(2);
253         if b_d == 1
254             % Birth move
255             iteration_count_rj_birth = iteration_count_rj_birth + 1;
256             if size(unknowns,1) < no_max_objects
257                 for j = 1:no_object_params
258                     random_proposed_unknowns(1,j) = random(prior_distributions(j));
259                 end
260                 random_proposed_unknowns(1,3) = -random_proposed_unknowns(1,3);
261                 proposed_unknowns = [unknowns;random_proposed_unknowns];
262             else
263                 continue
264             end
265         elseif b_d == 2
266             % Death move
267             iteration_count_rj_death = iteration_count_rj_death + 1;
268             if size(unknowns,1) > no_min_objects
269                 del_object = randi(size(unknowns,1));
270                 if del_object == 1
271                     proposed_unknowns = unknowns(2:end,:);
272                 elseif del_object == size(unknowns,1)
273                     proposed_unknowns = unknowns(1:end-1,:);
274                 else
275                     proposed_unknowns = [unknowns(1:del_object-1,:);...
276                     unknowns(del_object+1:end,:)];
277                 end
278             else
279                 continue
280             end
```

```
281         end
282         %% reversible jump moves save
283         % Calculating new forward model
284         if gravity_derivative == 1
285             if forward_model_type == 1
286                 proposed_forward_model = cylinder_model_gz(x_coords, y_coords,...
287                  z_coords, proposed_unknowns);
288             elseif forward_model_type == 2
289                 proposed_forward_model = cuboid_model_gz(x_coords, y_coords,...
290                 z_coords, proposed_unknowns);
291             elseif forward_model_type == 3
292                 proposed_forward_model = sphere_model_gz(x_coords, y_coords,...
293                  z_coords,proposed_unknowns);
294             end
295         elseif gravity_derivative == 2
296             if forward_model_type == 1
297                 proposed_forward_model = cylinder_model_gzz(x_coords, y_coords,...
298                 z_coords, proposed_unknowns,ai_separation);
299             elseif forward_model_type == 2
300                 proposed_forward_model = cuboid_model_gzz(x_coords, y_coords,...
301                  z_coords, proposed_unknowns,ai_separation);
302             elseif forward_model_type == 3
303                 proposed_forward_model = sphere_model_gzz(x_coords, y_coords,...
304                  z_coords,proposed_unknowns,ai_separation);
305             end
306         elseif gravity_derivative == 3
307         end
308         % Subtract the mean and add the offset value
309         if subtract_mean == 1
310             proposed_forward_model = (proposed_forward_model - ...
311 mean(proposed_forward_model)) + offset.*10^-8;
312         else
313             proposed_forward_model = proposed_forward_model + offset.*10^-8;
314         end
315         % Calculate the prior values
316         for i = 1:size(proposed_unknowns,1)
317             for j = 1:no_object_params
318                 if j == 3 && proposed_unknowns(i,j) > 0
319                     proposed_prior(i,j) = 0;
320                 elseif j == 3
321                     proposed_prior(i,j) = pdf(prior_distributions(j),...
322                     abs(proposed_unknowns(i,j)));
323                 else
324                     proposed_prior(i,j) = pdf(prior_distributions(j),...
325                     proposed_unknowns(i,j));
326                 end
327             end
328         end
329         proposed_log_prior = log(prod(prod(proposed_prior)).*prior_offset.*prior_error);
330         % Calculating the proposed likelihood
331         if laplace == 1
332             proposed_log_likelihood = log_laplace_model(gravity_data, ...
333             proposed_forward_model, error_sensor, error, gravity_derivative);
334         elseif laplace == 0
335             proposed_log_likelihood = log_gaussian_model(gravity_data...
336             , proposed_forward_model, error_sensor, error, gravity_derivative);
337         end
338         % Calculate ratio of current and previous posterior distributions
339         log_alpha = ((proposed_log_likelihood + proposed_log_prior) -...
340          (log_likelihood + log_prior));
341         log_alpha = log_alpha/(1 + t);
342         random_no = log(rand(1));
343         % Save likelihood, prior and alpha values
344         all_log_likelihood = [all_log_likelihood;proposed_log_likelihood];
345         all_log_prior = [all_log_prior;proposed_log_prior];
346         all_log_alpha = [all_log_alpha;log_alpha];
347         % Save and update all params
348         if (log_alpha > random_no) && (isinf(log_alpha) == 0) && ...
349         (isnan(log_alpha) == 0)
350             % Birth/death move
351             if move_type == 1
352                 if b_d == 1
353                     % The proposal distribution needs to follow object changes
354                     proposal_std_unknowns = [proposal_std_unknowns;...
355                     ones(1,size(proposal_std_unknowns,2))...
356                     .*proposal_std_unknowns(1,:)];
357                     accept_count_rj_birth = accept_count_rj_birth + 1;
358                 elseif b_d == 2
359                     % The proposal distribution needs to follow object changes
360                     if del_object == 1 %&& size(proposed_unknowns,1) > 1
361                         proposal_std_unknowns = proposal_std_unknowns(2:end,:);
362                     elseif del_object == size(proposal_std_unknowns,1)
363                         proposal_std_unknowns = proposal_std_unknowns(1:end-1,:);
364                     else
365                         proposal_std_unknowns = [...
366                         proposal_std_unknowns(1:del_object-1,:);...
367                         proposal_std_unknowns(del_object+1:end,:)];
368                     end
369                     accept_count_rj_death = accept_count_rj_death + 1;
370                 end
371             end
372             % Set accept changes
373             unknowns = proposed_unknowns;
374             log_prior = proposed_log_prior;
```

```
375             log_likelihood = proposed_log_likelihood;
376             % Save number of objects
377             no_objects_output(1,size(unknowns,1)) = ...
378             no_objects_output(1,size(unknowns,1)) + 1;
379             data_dimension = [data_dimension; size(unknowns,1)];
380             if k > mcmc_iterations*burn_in_discard_percent
381                 data_dimension_converged = [data_dimension_converged; ...
382                 size(unknowns,1)];
383                 no_objects_output_converged(1,size(unknowns,1)) = ...
384                 no_objects_output_converged(1,size(unknowns,1)) + 1;
385             end
386             % Save all parameters
387             for i = 1:no_object_params
388                 for j = 1:size(unknowns,1)
389                     mcmc_output_total{1,i+(j*no_object_params-...
390                     no_object_params)} = [mcmc_output_total{...
391                     1,i+(j*no_object_params-no_object_params)}; ...
392                     proposed_unknowns(j,i)];
393                     if k > mcmc_iterations*burn_in_discard_percent
394                         mcmc_output_converged{1,i+(j*no_object_params...
395                         -no_object_params)} = [mcmc_output_converged{...
396                         1,i+(j*no_object_params-no_object_params)};...
397                          proposed_unknowns(j,i)];
398                     end
399                 end
400             end
401             unknowns_padded_save = [reshape(unknowns',1,no_object_params*...
402             size(unknowns,1)),zeros(1,size(mcmc_output_total,2)-...
403             size(reshape(unknowns,1,no_object_params*size(unknowns,1)),2))];
404             unknowns_padded = padarray(unknowns_padded_save,[0,no_object_params*...
405             no_max_objects-size(unknowns_padded_save,2)]);
406             for i = 1:no_object_params*no_max_objects
407                     mcmc_output_padded{1,i} = [mcmc_output_padded{1,i}; ...
408                     unknowns_padded(1,i)];
409                     if k > mcmc_iterations*burn_in_discard_percent
410                         mcmc_output_padded_converged{1,i} = ...
411                         [mcmc_output_padded_converged{1,i};unknowns_padded(1,i)];
412                     end
413             end
414             % Save key values
415             accepted_log_likelihood = [accepted_log_likelihood;log_likelihood];
416             accepted_log_prior = [accepted_log_prior;log_prior];
417             accepted_log_alpha = [accepted_log_alpha;log_alpha];
418         else
419
420         end
421     end
422     %% Regular MH move
423     if move_type == 2
424         random_number = randi(100);
425         if random_number < 50
426             mh_move_type = 1; % Single object block update
427         elseif random_number >= 50 && random_number <= 100
428             mh_move_type = 2; % All object update
429         elseif random_number > 100
430             mh_move_type = 3; % Global params
431         end
432         proposed_unknowns = unknowns;
433         % Update one object parameter block
434         if mh_move_type == 1
435             rand_object = randi(size(unknowns,1));
436             % Set relative chance to sample depth volume and density..
437             skew_rand_parameter_block = randi(2);
438             if skew_rand_parameter_block == 1
439                 rand_parameter_block = 1;
440             elseif skew_rand_parameter_block > 1;
441                 rand_parameter_block = 2;
442             end
443             iteration_count_object(1,rand_object) =...
444              iteration_count_object(1,rand_object) + 1;
445             iteration_count_object_parameter_block(...
446             rand_object,rand_parameter_block) = ...
447             iteration_count_object_parameter_block(rand_object,...
448             rand_parameter_block) + 1;
449             if cauchy_prop == 1
450                 if rand_parameter_block == 1
451                     for i = param_block_1
452                         new_param = trnd(1).*proposal_std_unknowns(rand_object,i);
453                         proposed_unknowns(rand_object,i) = ...
454                         proposed_unknowns(rand_object,i) ...
455                         + new_param;
456                     end
457                 elseif rand_parameter_block == 2
458                     for i = param_block_2
459                         new_param = trnd(1).*proposal_std_unknowns(rand_object,i);
460                         proposed_unknowns(rand_object,i) = ...
461                         proposed_unknowns(rand_object,i) ...
462                         + new_param;
463                     end
464                 end
465             else
466                 if rand_parameter_block == 1
467                     for i = param_block_1
468                         new_param = normrnd(zeros(size(i,2)),...
469                         proposal_std_unknowns(rand_object,i));
```

```
470                         proposed_unknowns(rand_object,i) = ...
471                         proposed_unknowns(rand_object,i) ...
472                         + new_param;
473                     end
474                 elseif rand_parameter_block == 2
475                     for i = param_block_2
476                         new_param = normrnd(zeros(size(i,2)),...
477                         proposal_std_unknowns(rand_object,i));
478                         proposed_unknowns(rand_object,i) = ...
479                         proposed_unknowns(rand_object,i) ...
480                         + new_param;
481                     end
482                 end
483             end
484             proposed_offset = offset;
485             proposed_error = error;
486         elseif mh_move_type == 2
487             %% Update all parameters at once
488             iteration_count_all_object_params = iteration_count_all_object_params...
489              + 1;
490             if cauchy_prop == 1
491                 for i = 1:no_object_params
492                     for j = 1:size(unknowns,1)
493                         new_param = trnd(1).*proposal_std_unknowns(j,i)...
494                         .*proposal_scaling;
495                         proposed_unknowns(j,i) = proposed_unknowns(j,i) ...
496                         + new_param;
497                     end
498                 end
499             else
500                 proposed_unknowns = normrnd(unknowns,proposal_std_unknowns.*...
501                 proposal_scaling,[size(unknowns,1),no_object_params]);
502             end
503             if cauchy_prop == 1
504                 proposed_offset = offset + trnd(1).*proposal_std_offset;
505             else
506                 proposed_offset = normrnd(offset,proposal_std_offset);
507             end
508             if infer_error == 1
509                 if cauchy_prop == 1
510                     proposed_error = error + trnd(1).*proposal_std_error;
511                 else
512                     proposed_error = normrnd(error,proposal_std_error);
513                 end
514             else
515                 proposed_error = error;
516             end
517         elseif mh_move_type == 3
518             %% Update global parameters only
519             proposed_unknowns = unknowns;
520             if infer_error == 1
521                 e_o = randi(2);
522             else
523                 e_o = 1;
524             end
525             if e_o == 1
526                 % Proposed offset
527                 if cauchy_prop == 1
528                     proposed_offset = offset + trnd(1).*proposal_std_offset;
529                 else
530                     proposed_offset = normrnd(offset,proposal_std_offset);
531                 end
532                 proposed_error = error;
533                 iteration_count_offset_params = iteration_count_offset_params + 1;
534             elseif e_o == 2
535                 % Proposed error
536                 if infer_error == 1
537                     if cauchy_prop == 1
538                         proposed_error = error + trnd(1).*proposal_std_error;
539                     else
540                         proposed_error = normrnd(error,proposal_std_error);
541                     end
542                 else
543                     proposed_error = error;
544                 end
545                 proposed_offset = offset;
546                 iteration_count_error_params = iteration_count_error_params + 1;
547             end
548 
549         end
550         % Calculate forward model
551         if gravity_derivative == 1
552             if forward_model_type == 1
553                 proposed_forward_model = cylinder_model_gz(x_coords, y_coords, ...
554                 z_coords, proposed_unknowns);
555             elseif forward_model_type == 2
556                 proposed_forward_model = cuboid_model_gz(x_coords, y_coords, ...
557                 z_coords, proposed_unknowns);
558             elseif forward_model_type == 3
559                 proposed_forward_model = sphere_model_gz(x_coords, y_coords,...
560                  z_coords, proposed_unknowns);
561             end
562         elseif gravity_derivative == 2
563             if forward_model_type == 1
564                 proposed_forward_model = cylinder_model_gzz(x_coords, y_coords,...
```

```
565                  z_coords, proposed_unknowns,ai_separation);
566              elseif forward_model_type == 2
567                  proposed_forward_model = cuboid_model_gzz(x_coords, y_coords,...
568                   z_coords,proposed_unknowns,ai_separation);
569              elseif forward_model_type == 3
570                  proposed_forward_model = sphere_model_gzz(x_coords, y_coords,...
571                   z_coords, proposed_unknowns,ai_separation);
572              end
573          end
574          % Remove mean from model and add offset
575          if subtract_mean == 1
576              proposed_forward_model = (proposed_forward_model - ...
577              mean(proposed_forward_model)) + proposed_offset.*10^-8;
578          else
579              proposed_forward_model = proposed_forward_model + ...
580              proposed_offset.*10^-8;
581          end
582          % Calculate prior
583          for i = 1:size(proposed_unknowns,1)
584              for j = 1:no_object_params
585                  if j == 3 && proposed_unknowns(i,j) > 0
586                      proposed_prior(i,j) = 0;
587                  elseif j == 3
588                      proposed_prior(i,j) = pdf(prior_distributions(j),...
589                      abs(proposed_unknowns(i,j)));
590                  else
591                      proposed_prior(i,j) = pdf(prior_distributions(j),...
592                      proposed_unknowns(i,j));
593                  end
594              end
595          end
596          proposed_prior_offset = pdf(prior_distribution_offset,proposed_offset);
597          proposed_prior_error = pdf(prior_distribution_error,proposed_error);
598          if isnan(proposed_prior_error)
599              proposed_prior_error = 1;
600          end
601          proposed_log_prior = log(prod(prod(proposed_prior)).*...
602          proposed_prior_offset.* proposed_prior_error);
603          % Calculate likelihood value
604          if laplace == 1
605              proposed_log_likelihood = log_laplace_model(gravity_data, ...
606              proposed_forward_model, error_sensor, proposed_error,gravity_derivative);
607          elseif laplace == 0
608              proposed_log_likelihood = log_gaussian_model(gravity_data, ...
609              proposed_forward_model, error_sensor, proposed_error,gravity_derivative);
610          end
611          %% Calculate ratio of current and previous posterior distributions
612          log_alpha = ((proposed_log_likelihood + proposed_log_prior) - ...
613          (log_likelihood + log_prior));
614          log_alpha = log_alpha/(1 + t);
615          random_no = log(rand(1));
616          % Save likelihood prior and alpha values
617          all_log_likelihood = [all_log_likelihood;proposed_log_likelihood];
618          all_log_prior = [all_log_prior;proposed_log_prior];
619          all_log_alpha = [all_log_alpha;log_alpha];
620          % Save and update params
621          if (log_alpha > random_no) && (isinf(log_alpha) == 0) &&...
622           (isnan(log_alpha) == 0)
623              % Set new values
624              unknowns = proposed_unknowns;
625              error = proposed_error;
626              offset = proposed_offset;
627              prior_offset = proposed_prior_offset;
628              prior_error = proposed_prior_error;
629              log_prior = proposed_log_prior;
630              log_likelihood = proposed_log_likelihood;
631              % Save dimension parameters
632              no_objects_output(1,size(unknowns,1)) =...
633               no_objects_output(1,size(unknowns,1)) + 1;
634              data_dimension = [data_dimension; size(unknowns,1)];
635              if k > mcmc_iterations*burn_in_discard_percent
636                  no_objects_output_converged(1,size(unknowns,1)) = ...
637                  no_objects_output_converged(1,size(unknowns,1)) + 1;
638                  data_dimension_converged = ...
639                  [data_dimension_converged; size(unknowns,1)];
640              end
641              % Save data for each proposal type
642              if mh_move_type == 1
643                  % Increase accepted number
644                  accept_count_object_block(rand_object,rand_parameter_block) = ...
645                  accept_count_object_block(rand_object,rand_parameter_block) + 1;
646                  accept_count_object(1,rand_object) = ...
647                  accept_count_object(1,rand_object) + 1;
648                  % Save parameter block only
649                  if rand_parameter_block == 1
650                      param_block_save = param_block_1;
651                  elseif rand_parameter_block == 2
652                      param_block_save = param_block_2;
653                  end
654                  for i = param_block_save
655                      mcmc_output_block_total{1,...
656                      i+(rand_object*no_object_params-no_object_params)} = ...
657                      [mcmc_output_total{1,i+(rand_object*...
658                      no_object_params-no_object_params)};...
659                       proposed_unknowns(rand_object,i)];
```

```
660                     if k > mcmc_iterations*burn_in_discard_percent
661                         mcmc_output_block_converged{1,...
662                         i+(rand_object*no_object_params-no_object_params)} = ...
663                         [mcmc_output_converged{1,i+(rand_object...
664                         *no_object_params-no_object_params)}...
665                         ; proposed_unknowns(rand_object,i)];
666                     end
667                 end
668                 % Save all parameters
669                 for i = 1:no_object_params
670                     for j = 1:size(unknowns,1)
671                         mcmc_output_total{1,i+(j*...
672                         no_object_params-no_object_params)} = ...
673                         [mcmc_output_total{1,i+(j*...
674                         no_object_params-no_object_params)};...
675                          proposed_unknowns(j,i)];
676                         if k > mcmc_iterations*burn_in_discard_percent
677                             mcmc_output_converged{1,i+...
678                             (j*no_object_params-no_object_params)} = ...
679                             [mcmc_output_converged{1,i+...
680                             (j*no_object_params-no_object_params)};...
681                              proposed_unknowns(j,i)];
682                         end
683                     end
684                 end
685                 % Save padded with zeros
686                 unknowns_padded_save = [reshape(unknowns',1,...
687                 no_object_params*size(unknowns,1)),zeros(1,size(mcmc_output_total,2)-...
688                 size(reshape(unknowns,1, no_object_params*size(unknowns,1)),2))];
689                 unknowns_padded = padarray(unknowns_padded_save,[0,no_object_params*...
690                 no_max_objects-size(unknowns_padded_save,2)]);
691                 for i = 1:no_object_params*no_max_objects
692                     mcmc_output_padded{1,i} = [mcmc_output_padded{1,i}; ...
693                     unknowns_padded(1,i)];
694                     if k > mcmc_iterations*burn_in_discard_percent
695                         mcmc_output_padded_converged{1,i} = ...
696                         [mcmc_output_padded_converged{1,i};...
697                          unknowns_padded(1,i)];
698                     end
699                 end
700             elseif mh_move_type == 2
701                 % Increase accept number
702                 accept_count_all_object_params = accept_count_all_object_params + 1;
703                 % Save all parameters
704                 for i = 1:no_object_params
705                     for j = 1:size(unknowns,1)
706                         mcmc_output_total{1,i+(j*no_object_params-...
707                         no_object_params)} = ...
708                         [mcmc_output_total{1,i+(j*no_object_params-...
709                         no_object_params)};...
710                          proposed_unknowns(j,i)];
711                         if k > mcmc_iterations*burn_in_discard_percent
712                             mcmc_output_converged{1,i+(j*no_object_params-...
713                             no_object_params)} = ...
714                             [mcmc_output_converged{1,i+(j*no_object_params-...
715                             no_object_params)};...
716                              proposed_unknowns(j,i)];
717                         end
718                     end
719                 end
720                 % Save padded with zeros
721                 unknowns_padded_save = [reshape(unknowns',1,...
722                 no_object_params*size(unknowns,1)),...
723                     zeros(1,size(mcmc_output_total,2)-size(reshape(unknowns,1,...
724                     no_object_params*size(unknowns,1)),2))];
725                 unknowns_padded = padarray(unknowns_padded_save,...
726                 [0,no_object_params*no_max_objects-size(unknowns_padded_save,2)]);
727                 for i = 1:no_object_params*no_max_objects
728                     mcmc_output_padded{1,i} = [mcmc_output_padded{1,i};...
729                      unknowns_padded(1,i)];
730                     if k > mcmc_iterations*burn_in_discard_percent
731                         mcmc_output_padded_converged{1,i} = ...
732                         [mcmc_output_padded_converged{1,i};unknowns_padded(1,i)];
733                     end
734                 end
735                 % Save offset
736                 output_offset_total = [output_offset_total;offset];
737                 if k > mcmc_iterations*burn_in_discard_percent
738                     output_offset_converged = [output_offset_converged; offset];
739                 end
740                 % Save error
741                 accept_count_error_params = accept_count_error_params + 1;
742                 if infer_error == 1
743                     output_error_total = [output_error_total; error];
744                     if k > mcmc_iterations*burn_in_discard_percent
745                         output_error_converged = [output_error_converged; error];
746                     end
747                 else
748                     error = 0;
749                 end
750             elseif mh_move_type == 3
751                 % Increase accept number
752                 if e_o == 1
753                     accept_count_offset_params = accept_count_offset_params + 1;
754                     % Save offset
```

```
                    output_offset_total = [output_offset_total;offset];
                    if k > mcmc_iterations*burn_in_discard_percent
                        output_offset_converged = [output_offset_converged; offset];
                    end
                elseif e_o == 2
                    % Save error
                    accept_count_error_params = accept_count_error_params + 1;
                    if infer_error == 1
                        output_error_total = [output_error_total; error];
                        if k > mcmc_iterations*burn_in_discard_percent
                            output_error_converged = [output_error_converged; error];
                        end
                    else
                        error = 0;
                    end
                end
            end
            % Save key values
            accepted_log_likelihood = [accepted_log_likelihood;log_likelihood];
            accepted_log_prior = [accepted_log_prior;log_prior];
            accepted_log_alpha = [accepted_log_alpha;log_alpha];
        else
        end
    else
    end
    clear prior proposed_prior
end
end
```

Function to calculate the cuboid U_z forward model.

```
function [gz_final] = cuboid_model_gz(x,y,z,unknowns)
%% INPUT: x,y,z measurement coordinate vectors
%        unknowns [x centroid,y centroid, z top, ...
%                  x length, y length, z length, ...
%                  z rot, density]
% OUTPUT: gz vector
% Multiple objects are calculated at once
% Defined in RH coordinate system, -z is into soil.
% e.g. unknowns [0,0,-1,2,2,2,0,-2];
% Gives anomaly 2m deep to centroid.
centroid = [unknowns(:,1),unknowns(:,2),unknowns(:,3)];
l = [unknowns(:,4),unknowns(:,5),unknowns(:,6)];
centroid(:,3) = centroid(:,3) - l(:,3)./2;
psi = unknowns(:,7);
density = unknowns(:,8);
G = 6.67384*10^-11;
no_anomalies = size(psi,1);
%% Define cuboid vertices
const = G.*density.*1000;
gz_total = zeros(size(x,1),no_anomalies);
%% Define positive/negative sum contributions
pos_neg = [-1,-1,-1;...
    -1,1,-1;...
    1,-1,-1;...
    1,1,-1;...
    -1,-1,1;...
    -1,1,1;...
    1,-1,1;...
    1,1,1;...
    ];
%% Shift coords relative to object
for i = 1:no_anomalies
    %% Making coordinate centre the same as object centre
    x_trans = x - centroid(i,1);
    y_trans = y - centroid(i,2);
    z_trans = z - centroid(i,3);
    %% Rotating the coordinate system relative to object
    x_rot = (x_trans.*cos(psi(i)) - y_trans.*sin(psi(i))) + centroid(i,1);
    y_rot = (x_trans.*sin(psi(i)) + y_trans.*cos(psi(i))) + centroid(i,2);
    z_rot = z_trans + centroid(i,3);
    %% Define cuboid bounds/vertices
    x_1 = centroid(i,1) - l(i,1)/2;
    y_1 = centroid(i,2) - l(i,2)/2;
    z_1 = centroid(i,3) - l(i,3)/2;
    x_2 = centroid(i,1) + l(i,1)/2;
    y_2 = centroid(i,2) + l(i,2)/2;
    z_2 = centroid(i,3) + l(i,3)/2;
    vertices = [x_1,y_1,z_1;...
        x_1,y_2,z_1;...
        x_2,y_1,z_1;...
        x_2,y_2,z_1;...
        x_1,y_1,z_2;...
        x_1,y_2,z_2;...
        x_2,y_1,z_2;...
        x_2,y_2,z_2;...
        ];
    %% Calculate gravity contribution from cuboid
    gz = zeros(size(x,1),8,no_anomalies);
    for m = 1:size(vertices)
```

```
60          x_dash = x_rot - vertices(m,1);
61          y_dash = y_rot - vertices(m,2);
62          z_dash = z_rot - vertices(m,3);
63          r = sqrt(x_dash.^2 + y_dash.^2 + z_dash.^2);
64          gz(:,i,m) = prod(pos_neg(m,:),2).*((x_dash.*log(r + y_dash) + ...
65              y_dash.*log(r + x_dash) -...
66              z_dash.*atan((x_dash.*y_dash)./(r.*z_dash))));
67      end
68      gz_total(:,i) = const(i).*sum(gz(:,i,:),3);
69  end
70  gz_final = sum(gz_total,2);
71
72  end
```

Function to calculate the likelihood value.

```
1  function [log_gaussian] = log_gaussian_model(data,model_data,sensor_error...
2  ,other_error,gravity_derivative)
3  if gravity_derivative == 1
4      sensor_error = sensor_error.*10^-8;
5      other_error = other_error.*10^-8;
6  elseif gravity_derivative == 2
7      sensor_error = sensor_error.*10^-8;
8      other_error = other_error.*10^-8;
9  end
10 data = data(:);
11 model_data = model_data(:);
12 sensor_error = sensor_error(:);
13 other_error = other_error(:);
14 theta = (data - model_data);
15 % Calculate leading diagonal of the covariance matrix
16 covariance = sensor_error.^2 + other_error.^2;
17 log_det_covariance_matrix = sum(log(covariance));
18 log_const = size(data,1)*log(2*pi);
19 %% Calculation of log of exponential term
20 inverse_covariancematrix = 1./(covariance);
21 log_gaussian = -0.5*(sum(theta.*inverse_covariancematrix.*theta))...
22  - 0.5*(log_det_covariance_matrix + log_const);
23 % log_gaussian = log(mvnpdf(model_data,data,diag(covariance)));
24 end
```

Bibliography

[1] William J. Hinze, Ralph R. B. von Frese, and Afif H. Saad. *Gravity and Magnetic Exploration*. Cambridge University Press, 2013. Cambridge Books Online.

[2] a Peters, K Y Chung, and S Chu. High-precision gravity measurements using atom interferometry. *Metrologia*, 38(1):25–61, 2003.

[3] M E Everett. *Near-Surface Applied Geophysics*. Cambridge University Press, 2013.

[4] R P Middlemiss, A. Samarelli, D. J. Paul, J. Hough, S. Rowan, and G.D. Hammond. The First Measurement of the Earth Tides with a MEMS Gravimeter. *Nature*, 531(1):614, 2016.

[5] N. Metje, D. N. Chapman, C. D F Rogers, and K. Bongs. Seeing through the ground: The potential of gravity gradient as a complementary technology. *Advances in Civil Engineering*, 2011, 2011.

[6] H.M. Jol. *Ground Penetrating Radar Theory and Applications*. Elsevier Science, 2008.

[7] Jaroslava Panisova, Igor Murín, Roman Pašteka, Jana Haličková, Peter Brunčák, Vladimír Pohánka, Juraj Papčo, and Peter Milo. Geophysical fingerprints of shallow cultural structures from microgravity and GPR measurements in the Church of St. George, Svätý Jur, Slovakia. *Journal of Applied Geophysics*, 127:102–111, 2016.

[8] Klaus Mosegaard. Quest for consistency, symmetry, and simplicity — The legacy of Albert Tarantola.

[9] Mrinal K. Sen and Paul L. Stoffa. Bayesian inference, Gibbs' sampler and uncertainty estimation in geophysical inversion. *Geophysical Prospecting*, 44(2):313–350, 1996.

[10] D. C. Skeels. Ambiguity in Gravity Interpretation. *Geophysics*, 12(1):43–56, 1947.

[11] a Malinverno. A Bayesian criterion for simlicity in inverse problem parameterisation. *Geophys. J. Int.*, 140:267–285, 2000.

[12] A. Sumanovac and F L. Ritsema. System Architecture for 3D Gravity Modelling. *Geologia Croatica*, 49(2):145–153, 1996.

[13] R L Parker. The rapid calculation of potential anomalies. *Geophys. JR Astron. Soc*, 31(4):447–455, 1972.

[14] Douglas W. Oldenburg. The inversion and interpretation of gravity anomalies. *GEOPHYSICS*, 39(4):526–536, 1974.

[15] Juan Feng, Xiaohong Meng, Zhaoxi Chen, and Sheng Zhang. Three-dimensional density interface inversion of gravity anomalies in the spectral domain. *Journal of Geophysics and Engineering*, 11(3):035001, 2014.

[16] R. BOTEZATU, M. VISARION, F. SCURTU, and G. CUCU. Approximation of the gravitational attraction of geological bodies*. *Geophysical Prospecting*, 19(2):218–227, 1971.

[17] Lindrith Cordell and Roland G. Henderson. Iterative three dimensional solution of gravity anomaly data using a digital computer. *GEOPHYSICS*, 33(4):596–601, 1968.

[18] Mark Jessell. Three-dimensional geological modelling of potential-field data. *Computers & Geosciences*, 27(4):455–465, 2001.

[19] Michael Commer. Three-dimensional gravity modelling and focusing inversion using rectangular meshes. *Geophysical Prospecting*, 59(5):966–979, 2011.

[20] Y. Li and D.W. Oldenburg. 3-D inversion of gravity data. *Geophysics*, 63(1):109–119, 1998.

[21] A.N. Tikhonov and V.I.A. Arsenin. *Solutions of ill-posed problems*. Scripta series in mathematics. Winston, 1977.

[22] Zhu Wen Wang, Shi Xu, Yin Ping Liu, and Jing Hua Liu. Extrapolated Tikhonov method and inversion of 3D density images of gravity data. *Applied Geophysics*, 11(2):139–148, 2014.

[23] Michael S. Zhdanov. New advances in regularized inversion of gravity and electromagnetic data. *Geophysical Prospecting*, 57(4):463–478, 2009.

[24] Hendra Grandis and Darharta Dahrin. Constrained Two-Dimensional Inversion of Gravity Data. *Journal of Mathematical and Fundamental Sciences*, 46(1):1–13, 2014.

[25] L. BORSTING PEDERSEN. Interpretation of potential field data a generalized inverse approach*. *Geophysical Prospecting*, 25(2):199–230, 1977.

[26] A. Tarantola. *Inverse Problem Theory and Methods for Model Parameter Estimation*. Other titles in applied mathematics. Society for Industrial and Applied Mathematics, 2005.

[27] G. E. Backus. Bayesian inference in geomagnetism. *Geophysical Journal International*, 92(1):125–142, 1988.

[28] Alberto Malinverno. Parsimonious Bayesian Markov chain Monte Carlo inversion in a nonlinear geophysical problem. *Geophysical Journal International*, 151(3):675–688, 2002.

[29] Seokhoon Oh and Baeksoo Suh. Bayesian Inversion for Cavity Investigation. *http://dx.doi.org/10.1080/12269328.2010.10541313*, 2012.

[30] Politecnico D I Milano. Bayesian interpretation of gravity data with geological prior information. 2013.

[31] Gareth Brown. Bayesian mass anomaly estimation with measurements of gravity. In *ISP 2015 The 2nd IET International Conference on Intelligent Signal Processing*. IET, 2015.

[32] Xiaolin Luo. Constraining the shape of a gravity anomalous body using reversible jump Markov chain Monte Carlo. *Geophysical Journal International*, 180(3):1067–1079, mar 2010.

[33] Barry Kirkendall. *Nonlinear model appraisal in gravity gradiometry imaging: properties and modeling of the resolution matrix*. PhD thesis.

[34] R J Blakely. *Potential Theory in Gravity and Magnetic Applications*. Stanford-Cambridge program. Cambridge University Press, 1996.

[35] Majid Beiki. Analytic signals of gravity gradient tensor and their application to estimate source location. *Geophysics*, 75(6):159–174, 2010.

[36] P. Novák and E. W. Grafarend. The effect of topographical and atmospheric masses on spaceborne gravimetric and gradiometric data. *Studia Geophysica et Geodaetica*, 50(4):549–582, 2006.

[37] W J Hinze, R R B von Frese, R Von Frese, and A H Saad. *Gravity and Magnetic Exploration: Principles, Practices, and Applications*. Cambridge University Press, 2013.

[38] A Veryaskin and W McRae. On combined gravity gradient components modelling for applied geophysics. *Journal of Geophysics and Engineering*, 348:1–17, 2008.

[39] a H Saad. Understanding gravity gradients — a tutorial. *The Leading Edge*, 25(8):942–949, 2006.

[40] Manik Talwani and Maurice Ewing. Rapid computation of gravitational attraction of three dimensional bodies of arbitrary shape. *GEOPHYSICS*, 25(1):203–225, 1960.

[41] L T Long and R D Kaufmann. *Acquisition and Analysis of Terrestrial Gravity Data*. Cambridge University Press, 2013.

[42] D H Griffiths and R F King. *Applied Geophysics for Geologists and Engineers: The Elements of Geophysical Prospecting*. Pergamon International Library of Science, Technology, Engin. Elsevier Science, 2013.

[43] W M Telford, L P Geldart, and R E Sheriff. *Applied Geophysics*. Cambridge University Press, 1990.

[44] Buddhadeb Banerjee and S. P. Das Gupta. Gravitational attraction of a rectangular parallelepiped. *Geophysics*, 42(5):1053–1055, 1977.

[45] B. Heck and K. Seitz. A comparison of the tesseroid, prism and point-mass approaches for mass reductions in gravity field modelling. *Journal of Geodesy*, 81(2):121–136, 2007.

[46] Xinan Wu. Gravity Gradient Survey with a Mobile Atom Interferometer. *Thesis*, (March):160, 2009.

[47] A Senger. A Mobile Atom Interferometer for High-Precision Measurements of Local Gravity. 2011.

[48] Alexander Niggebaum. *Towards mobile quantum sensors for gravity surveys.* PhD thesis, 2015.

[49] W Jacoby and P L Smilde. *Gravity Interpretation: Fundamentals and Application of Gravity Inversion and Geological Interpretation*. Earth and Environmental Science. Springer Berlin Heidelberg, 2009.

[50] I P Escobar. Adjustment of Gravity Observations towards a Microgal Precision. *Int. J. ldots*, 2013(January):98–107, 2013.

[51] Dwain K. Butler. Microgravimetric and gravity gradient techniques for detection of subsurface cavities. *Geophysics*, 49(7):1084 – 1096, 1984.

[52] Anestis J Romaides, James C Battis, Roger W Sands, Alan Zorn, Donald O Benson, and Daniel J DiFrancesco. A comparison of gravimetric techniques for measuring subsurface void signals. *Journal of Physics D: Applied Physics*, 34(3):433–443, 2001.

[53] Z. J. Fajklewicz. Gravity vertical gradient measurements for the detection of small geologic and anthropogenic forms. *Geophysics*, 41(5):1016–1030, 1976.

[54] L J B LaCoste. The zero-length spring gravity meter. *The Leading Edge*, (July):20–21, 1988.

[55] SCINTREX. CG-5 Scintrex Autograv System OPERATION MANUAL. page 308, 2006.

[56] H H Essen, F Kruger, T Dahm, and I Grevemeyer. On the generation of secondary microseisms observed in northern and central Europe. *J. Geophys. Res: Solid Earth*, 108(B10):2506–2520, 2003.

[57] M. Kasevich and S. Chu. Measurement of the gravitational acceleration of an atom with a light-pulse atom interferometer. *Applied Physics B Photophysics and Laser Chemistry*, 54(5):321–332, 1992.

[58] C.J. Foot. *Atomic physics*. Oxford master series in physics. Oxford University Press, 2005.

[59] a Peters, K Y Chung, and S Chu. High-precision gravity measurements using atom interferometry. *Metrologia*, 38(1):25–61, 2003.

[60] M. Kasevich. *Atom Interferometry in an Atomic Fountain*. PhD thesis, Stanford University, 1992.

[61] J. M. McGuirk, G. T. Foster, J. B. Fixler, M. J. Snadden, and M. A. Kasevich. Sensitive absolute-gravity gradiometry using atom interferometry. *Phys. Rev. A*, 65:033608, Feb 2002.

[62] Martin Lederer. Accuracy of the relative gravity measurement. *Acta Geodynamica et Geomaterialia*, 6(3):383–390, 2009.

[63] I. M. Longman. Formulas for computing the tidal accelerations due to the moon and the sun. *Journal of Geophysical Research*, 64(12):2351, 1959.

[64] J. B. Merriam. Atmospheric pressure and gravity. *Geophysical Journal International*, 109(3):488–500, 1992.

[65] H.O Seigel. A Guide to high precision land Gravimeter surveys. (905):1–132, 1995.

[66] T. R. LaFehr. Standardization in gravity reduction. *Geophysics*, 56(8):1170, 1991.

[67] K. Sundaralingam. Free-air correction in gravity anomaly interpretation. *Physics of the Earth and Planetary Interiors*, 49(1-2):1–2, 1987.

[68] British Geological Survey. Online tool: Uk geology. `http://mapapps.bgs.ac.uk/geologyofbritain/home.html`. [Online; Accessed 18/08/16].

[69] L L Nettleton. DETERMINATION OF DENSITY FOR REDUCTION OF GRAVIMETER OBSERVATIONS. *Geophysics*, 4(3):176–183, 1939.

[70] Malcolm Sambridge and Klaus Mosegaard. Monte Carlo Methods in Geophysical Inverse Problems. *Earth*, 40(September):1009, 2002.

[71] Colin Howson and Peter Urbach. Bayesian reasoning in science. *Nature*, 350(6317):371–374, 1991.

[72] John a Scales, Roel Snieder, and Sven Treitel. To Bayes or not to Bayes ? *Geophysics*, 62(4):1045–1046, 1997.

[73] Andrew Gelman and Cosma Rohilla Shalizi. Philosophy and the practice of Bayesian statistics. *British Journal of Mathematical and Statistical Psychology*, 66(1):8–38, 2013.

[74] Andrew Gelman and Christian P. Robert. "Not Only Defended But Also Applied": The Perceived Absurdity of Bayesian Inference. *The American Statistician*, 67(1):1–5, 2013.

[75] R C Aster, B Borchers, and C H Thurber. *Parameter Estimation and Inverse Problems*. Academic Press. Academic Press, 2013.

[76] Mr. Bayes and Mr. Price. An essay towards solving a problem in the doctrine of chances. by the late rev. mr. bayes, f. r. s. communicated by mr. price, in a letter to john canton, a. m. f. r. s. *Philosophical Transactions*, 53:370–418, 1763.

[77] D Sivia and J Skilling. *Data Analysis: A Bayesian Tutorial*. Oxford science publications. OUP Oxford, 2006.

[78] Tadeusz J. Ulrych, Mauricio D. Sacchi, and Alan Woodbury. A Bayes tour of inversion: A tutorial. *http://dx.doi.org/10.1190/1.1444923*, 2012.

[79] A O'Hagan and M G Kendall. *Kendall's Advanced Theory of Statistics: Bayesian inference. Vol. 2B*. Kendall's advanced theory of statistics. Arnold, 2000.

[80] Klaus Mosegaard and Albert Tarantola. Monte Carlo sampling of solutions to inverse problems. *Journal of Geophysical Research: Solid Earth*, 100(B7):12431–12447, 1995.

[81] John A. Scales and Luis Tenorio. Prior information and uncertainty in inverse problems. *Geophysics*, 66:389–397, 2000.

[82] M Sambridge, T Bodin, K Gallagher, and H Tkalcic. Transdimensional inference in the geosciences. *Philosophical transactions. Series A, Mathematical, physical, and engineering sciences*, 371(1984):20110547, 2013.

[83] G. D'Agostini. *Bayesian Reasoning in Data Analysis: A Critical Introduction*. World Scientific, 2003.

[84] Charles Geyer. Practical Markov Chain Monte Carlo. *STATISTICAL SCIENCE*, 7(4):473–483, 1992.

[85] D Gamerman. *Markov Chain Monte Carlo: Stochastic Simulation for Bayesian Inference*. Chapman & Hall/CRC Texts in Statistical Science. Taylor & Francis, 1997.

[86] Mrinal K. Sen and Paul L. Stoffa. *Global Optimization Methods in Geophysical Inversion*. Cambridge University Press, second edition, 2013. Cambridge Books Online.

[87] W R Gilks, S Richardson, and D Spiegelhalter. *Markov Chain Monte Carlo in Practice*. Chapman & Hall/CRC Interdisciplinary Statistics. Taylor & Francis, 1995.

[88] Klaus Mosegaard. Resolution analysis of general inverse problems through inverse Monte Carlo sampling. *Inverse Problems*, 14:405–426, 1999.

[89] Luke Tierney. Markov Chains for Exploring Posterior Distributions. *Annals of Statistics*, 22(4):1701–1762, 1994.

[90] Nicholas Metropolis, Arianna W. Rosenbluth, Marshall N. Rosenbluth, Augusta H. Teller, and Edward Teller. Equation of state calculations by fast computing machines. *Journal Chemical Physics*, 21(6):1087–1092, 1953.

[91] W K Hastings. Monte Carlo sampling methods using Markov chains and their applications. *Biometrika Vol*, 57(1):97–109, 1970.

[92] Siddhartha Chib and Edward Greenberg. Understanding the Metropolis-Hastings Algorithm. *The American Statistician*, 49(4):327–335, 1995.

[93] Malcolm Sambridge, K. Gallagher, A. Jackson, and P. Rickwood. Trans-dimensional inverse problems, model comparison and the evidence. *Geophysical Journal International*, 167(2):528–542, 2006.

[94] Peter J. Green. Reversible jump markov chain monte carlo computation and bayesian model determination. *Biometrika*, 82(4):711–732, 1995.

[95] Kerry Gallagher, Karl Charvin, Soren Nielsen, Malcolm Sambridge, and John Stephenson. Markov chain Monte Carlo (MCMC) sampling methods to determine optimal models, model resolution and model choice for Earth Science problems. *Marine and Petroleum Geology*, 26(4):525–535, 2009.

[96] P.J. Green. Trans-dimensional Markov chain Monte Carlo. *Highly structured stochastic systems*, 27:179–198, 2003.

[97] Mathematical Statistics. Bayesian Analysis of Mixture Models with an Unknown Number of Components- An Alternative to Reversible Jump Methods Author (s): Matthew Stephens Source : The Annals of Statistics , Vol . 28 , No . 1 (Feb ., 2000), pp . 40-74 Published by : Institute of Mathematical Statistics Stable URL : http://www.jstor.org/stable/2673981 Accessed : 10-08-2016 08 : 55 UTC. 28(1):40–74, 2000.

[98] Burke J. Minsley. A trans-dimensional Bayesian Markov chain Monte Carlo algorithm for model assessment using frequency-domain electromagnetic data. *Geophysical Journal International*, 187(1):252–272, 2011.

[99] A. Gelman, G. O. Roberts, and W. R. Gilks. Efficient Metropolis Jumping Rules, 1996.

[100] Heikki Haario. An adaptive Metropolis algorithm. *Bernoulli*, 7(2):223–242, 2001.

[101] S P Brooks, P Giudici, and G O Roberts. Efficient construction of reversible jump Markov chain Monte Carlo proposal distributions. *J. R. Statist. Soc. B*, 65(1):3–55, 2003.

[102] Seok Hoon Oh and Byung D. Kwon. Geostatistical approach to bayesian inversion of geophysical data: Markov chain Monte Carlo method. *Earth, Planets and Space*, 53(8):777–791, 2001.

[103] S Brooks, A Gelman, G Jones, and X L Meng. *Handbook of Markov Chain Monte Carlo*. Chapman & Hall/CRC Handbooks of Modern Statistical Methods. CRC Press, 2011.

[104] Y Fan and S A Sisson. Reversible jump Markov chain Monte Carlo. *Biometrika*, 82(October):711–732, 2010.

[105] A Gelman, J B Carlin, H S Stern, D B Dunson, A Vehtari, and D B Rubin. *Bayesian Data Analysis, Third Edition*. Chapman & Hall/CRC Texts in Statistical Science. Taylor & Francis, 2013.

[106] John Geweke. Evaluating the accuracy of sampling-based approaches to the calculation of posterior moments. Staff Report 148, Federal Reserve Bank of Minneapolis, 1991.

[107] Gareth Brown. A Bayesian method for imaging voids using Gravimetry. In *4th IMA Conference on Mathematics in Defence*. IMA, 2015.

[108] Bayesian Ockham 's Razor, Author Analysis, William H Jefferys, and James O Berger. Ockham's Razor and Bayesian Analysis. *Source: American Scientist*, 80188128(1):64–72.

[109] Alistair Reid, Simon O'Callaghan, Edwin V. Bonilla, Lachlan McCalman, Tim Rawling, and Fabio Ramos. Bayesian joint inversions for the exploration of earth resources. *IJCAI International Joint Conference on Artificial Intelligence*, pages 2877–2884, 2013.

[110] George Tuckwell, T Grossey, S Owen, and P Stearns. The use of microgravity to detect small distributed voids and low-density ground. *Quarterly Journal of Engineering Geology and Hydrogeology*, 41(3):371–380, 2008.

[111] R. Reudink, R. Klees, O. Francis, J. Kusche, R. Schlesinger, A. Shabanloui, N. Sneeuw, and L. Timmen. High tilt susceptibility of the Scintrex CG-5 relative gravimeters. *Journal of Geodesy*, 88(6):617–622, 2014.

[112] Leica Geosystems. Leica Viva TS11 TS15 User Manual. Technical report.

[113] David Crossley, Jacques Hinderer, and Umberto Riccardi. The measurement of surface gravity. *Reports on progress in physics. Physical Society (Great Britain)*, 76(4):046101, 2013.

[114] Peter Green. On Bayesian Analysis of Mixtures with an Unknown Number of Components. *Journal Of The Royal Statistical Society Series B-Methodological*, 59(4):731–792, 1997.

[115] Lachlan McCalman, Simon T O'Callaghan, Alistair Reid, Darren Shen, Simon Carter, Lars Krieger, Graeme R Beardsmore, Edwin V Bonilla, and Fabio T Ramos. Dis-

tributed bayesian geophysical inversions. *Thirty-Ninth Stanford Geothermal Workshop*, pages 1–11, 2014.

[116] British Geological Survey. Online tool: Uk soil depth. `http://mapapps2.bgs.ac.uk/ukso/home.html?layer=SPMThickness`. [Online; Accessed 18/08/16].