
MAG3D Documentation

Release 5.0

UBC-GIF

Oct 06, 2020

Contents

1	MAG3D package overview	3
1.1	Description	3
1.2	Program library content	4
1.3	Licensing	4
1.4	Installing	5
1.5	Highlights of changes from version 3.2	5
2	Background theory	7
2.1	Introduction	7
2.2	Forward modelling	7
2.3	Inversion methodology	9
2.4	Depth Weighting and Distance Weighting	11
2.5	Wavelet Compression of Sensitivity Matrix	12
3	Elements of the program MAG3D	15
3.1	Introduction	15
3.2	General files for MAG3D programs	15
4	Running the programs	21
4.1	Introduction	21
4.2	Input and output files	23
5	Examples	31
5.1	Create Model	31
5.2	Forward Modeling	33
5.3	Depth/Distance Weighting	33
5.4	Computing Sensitivities	34
5.5	Inversion	35
5.6	Examples	36
6	References	59

MAG3D is a program library for carrying out forward modelling and inversion of surface, borehole, and airborne magnetic data in 3D. The contents of this manual are as follows:

MAG3D package overview

1.1 Description

is a program library for carrying out forward modelling and inversion of surface, airborne, and/or borehole magnetic data in the presence of a three dimensional Earth. The program library carries out the following functions:

1. Forward modelling of the magnetic field anomaly response of a 3D volume of susceptibility contrast.
2. The model is specified in the mesh of rectangular cells, each with a constant value of susceptibility. Topography is included in the mesh. The magnetic response can be calculated anywhere within the model volume, including above the topography to simulate ground or airborne surveys. There is also a capability to simulate and invert data collected beneath the surface (e.g. borehole surveys) and combinations of ground and borehole surveys.
3. Assumptions:
 - This code assumes susceptibilities are small enough that the effects of self-demagnetization can be neglected.
 - Remanent magnetization is not directly accounted for, although anomaly projections can be included with the observations.
4. Inversion of surface, airborne, and/or borehole magnetic data to generate 3D models of susceptibility contrast:
 - The inversion is solved as an optimization problem with the simultaneous goals of (i) minimizing a model objective function and (ii) generating synthetic data that match observations to within a degree of misfit consistent with the statistics of those data.
 - To counteract the inherent lack of information about the distance between source and measurement, the formulation incorporates depth or distance weighting.
 - By minimizing the model objective function, distributions of subsurface susceptibility contrast are found that are both close to a reference model and smooth in three dimensions. The degree to which either of these two goals dominates is controlled by the user by incorporating prior geophysical or geological information into the inversion. Explicit prior information may also take the form of upper and lower bounds on the susceptibility contrast in any cell.

- The regularization parameter (controlling relative importance of objective function and misfit terms) is determined in one of three ways, depending upon how much is known about errors in the measured data.
 - Implementation of parallel computing architecture (OpenMP) allows the user to take full advantage of multi-core processors on a CPU. A cluster-based code using Message Passing Interface (MPI) is also available. Notes on computation speed are found at the end of this section.
5. The large size of 3D inversion problems is mitigated by the use of wavelet compression. Parameters controlling the implementation of this compression are available for advanced users.

The initial research underlying this program library was funded principally by the mineral industry consortium “Joint and Cooperative Inversion of Geophysical and Geological Data” (1991 - 1997) which was sponsored by NSERC and the following 11 companies: BHP Minerals, CRA Exploration, Cominco Exploration, Falconbridge, Hudson Bay Exploration and Development, INCO Exploration & Technical Services, Kennecott Exploration Company, Newmont Gold Company, Noranda Exploration, Placer Dome, and WMC.

The current improvements have been funded by the consortium “Potential fields and software for advanced inversion” (2012-2016) sponsored by Newmont, Teck, Glencore, BHP Billiton, Vale, Computational Geoscience Inc, Cameco, Barrick, Rio Tinto, and Anglo American.

1.2 Program library content

1.2.1 Executable programs

This package consists of five major programs:

- PFWEIGHT: calculates the depth/distance weighting function
- MAGFOR3D: performs forward modelling
- MAGSEN3D: calculates the sensitivity matrix
- MAGPRE3D: multiplies the sensitivity file by the model to get the predicted data
- MAGINV3D: performs 3D magnetic inversion

1.2.2 Graphical user interfaces

GUI-based utilities for these codes include respective viewers for the data and models. They are only available on Windows platforms and can be freely downloaded through the UBC-GIF website:

- [GM_DATA_VIEWER](#): a utility for viewing raw surface or airborne data (not borehole data), error distributions, and for comparing observed to predicted data directly or as difference maps.
- [MeshTools3D](#): a utility for displaying resulting 3D models as volume renderings. Susceptibility volumes can be sliced in any direction, or isosurface renderings can be generated.
- [GUI](#): a GUI to run MAG3D v5.0 on either Linux or Windows. **NOTE: The download does not contain the inversion/modelling codes.**

1.3 Licensing

A **constrained educational version** of the program is available with the [IAG](#) package (please visit [UBC-GIF website](#) for details). The educational version is fully functional so that users can learn how to carry out effective and efficient

3D inversions of magnetic data. **However, RESEARCH OR COMMERCIAL USE IS NOT POSSIBLE because the educational version only allows a limited number of data and model cells.**

Licensing for an unconstrained academic version is available - see the [Licensing policy document](#).

NOTE: All academic licenses will be **time-limited to one year**. You can re-apply after that time. This ensures that everyone is using the most recent versions of codes.

Licensing for commercial use is managed by third party distributors. Details are in the [Licensing policy document](#).

1.4 Installing

There is no automatic installer currently available for this package. Please follow the following steps in order to use the software:

1. Extract all files provided from the given zip-based archive and place them all together in a new folder such as
2. Add this directory as new path to your environment variables.

Two additional notes about installation:

- Do not store anything in the “bin” directory other than executable applications and Graphical User Interface applications (GUIs).
- A Message Pass Interface (MPI) version is available for Linux upon and the installation instructions will accompany the code.

1.5 Highlights of changes from version 3.2

The principal upgrades, described below, allow the new code to take advantage of current multi-core computers and also provide greater flexibility to incorporate the geological information.

Improvements since version :

1. A new projected gradient algorithm is used to implement hard constraints.
2. Fully parallelized computational capability (for both sensitivity matrix calculations and inversion calculations).
3. A facility to have active and inactive (i.e. fixed) cells.
4. Bounds are specified through two separate files, rather than one two-column file.
5. Additional flexibility for incorporating the reference model in the model objective function facilitates the generation of smooth models when borehole constraints are incorporated.
6. The `maginv3d.log` file has been simplified and detailed information on the inversion can be found in the `maginv3d.out` file.
7. An alternative version of the software compatible with Message Pass Interface (MPI) is available for Linux.
8. Backward compatibility: The new version has changed the input file format and the bounds file. Data, mesh, model, and topographic file formats have not changed.
9. The depth weighting function and sensitivity are computed separately.

1.5.1 Notes on computation speed

- For large problems, MAG3D is significantly faster than the previous single processor inversion because of the parallelization for computing the sensitivity matrix computation and inversion calculations. Using multiple threads for running the parallelized version resulted in sensitivity matrix calculation speedup proportional to the number of threads. The increase in speed for the inversion was less pronounced, but still substantial.
- It is strongly recommended to use multi-core processors for running the weighting and sensitivity calculation. The calculation of the sensitivity matrix (\mathbf{G}) is directly proportional to the number of data. The parallelized calculation of the n rows of \mathbf{G} is split between p processors. By default, all available processors are used. There is a feature to limit p to a user-defined number of processors.
- In the parallelized inversion calculation, $\mathbf{G}^T \mathbf{G}$ is multiplied by a vector, therefore each parallel process uses only a sub-matrix of \mathbf{G} and then the calculations are summed. Since there is significant communication between the CPUs, the speedup is less than a direct proportionality to the number of processors. However when running the same inversion under MPI environment on multiple computers the advantage is that a single computer does not have to store the entire sensitivity matrix.
- For incorporating bound information, the implementation of the projected gradient algorithm in version MAG3D is primarily that it results in a significantly faster solution than the logarithmic barrier technique used in earlier versions. An added benefit is the ability to reach the bounds given by the user.

2.1 Introduction

This manual presents theoretical background, numerical examples, and explanation for implementing the program library MAG3D. This suite of algorithms, developed at the UBC-Geophysical Inversion Facility, is used to invert magnetic responses over a three-dimensional susceptibility distribution. The manual is designed so that a geophysicist who is familiar with the magnetic experiment, but who is not necessarily versed in the details of inverse theory, can use the codes and invert his or her data.

A magnetic experiment involves measuring the anomalous magnetic field produced by magnetically susceptible materials beneath the surface, which have been magnetized by the earth's main magnetic field. The material with susceptibility $\kappa(x, y, z)$ is magnetized when the earth's main field impinges upon the subsurface formation with flux intensity \mathbf{B}_o . The magnetized material gives rise to a magnetic field, \mathbf{B}_a , which is superimposed on the inducing field to produce a total, or resultant, field. By measuring the resultant field and removing the inducing field from the measurements through numerical processing, one obtains the distribution of the anomalous field due to the susceptible material. In this program library we make the assumption that no remanent magnetization is present and restrict our attention to induced magnetization.

The data from a typical magnetic survey are a set of magnetic field measurements acquired over a 2D grid above the surface or along a number of boreholes within the volume of interest. These data are first processed to yield an estimate of the anomalous field due to the susceptible material in the area. The goal of the magnetic inversion is to obtain, from the extracted anomaly data, quantitative information about the distribution of the magnetic susceptibility in the ground. Thus, it is assumed that the input data to the inversion program is the extracted residual anomaly and the programs in the library are developed accordingly.

2.2 Forward modelling

2.2.1 General formulation

For a given inducing field \mathbf{B}_o , the magnetization \mathbf{J} depends upon the susceptibility through a differential equation. In most mineral exploration cases, the actual susceptibility is very small. Thus, we use the first order approximation,

where the magnetization is proportional to the susceptibility and is given by the product of susceptibility and the inducing magnetic field \mathbf{H}_o ,

$$\mathbf{J} = \kappa \mathbf{H}_o, \quad (2.1)$$

where $\mathbf{H}_o = \mathbf{B}_o / \mu_o$ and μ_o is the free-space magnetic permeability. This ignores the self-demagnetization effect by which the secondary field reduces the total inducing field within the susceptible region and results in a weaker magnetization than that given by equation (2.1).

The anomalous field produced by the distribution of magnetization \mathbf{J} given by the following integral equation with a dyadic Green's function:

$$\mathbf{B}_a(\mathbf{r}) = \frac{\mu_0}{4\pi} \int_V \nabla \nabla \frac{1}{|\mathbf{r} - \mathbf{r}_o|} \cdot \mathbf{J} dv, \quad (2.2)$$

where \mathbf{r} is the position of the observation point and V represents the volume of magnetization at its position, \mathbf{r}_o . The above equation is valid for observation locations above the earth's surface. We assume the magnetic permeability is μ_o at borehole locations to allow equation (2.2) to be valid everywhere.

When the susceptibility is constant within a volume of source region, the above equation can be written in matrix form as:

$$\mathbf{B}_a = \mu_o \begin{pmatrix} T_{11} & T_{12} & T_{13} \\ T_{21} & T_{22} & T_{23} \\ T_{31} & T_{32} & T_{33} \end{pmatrix} \kappa \mathbf{H}_o \equiv \mu_o \kappa \mathbf{T} \mathbf{H}_o. \quad (2.3)$$

The tensor \mathbf{T}_{ij} is given by

$$\mathbf{T}_{ij} = \frac{1}{4\pi} \int_V \frac{\partial}{\partial x_i} \frac{\partial}{\partial x_j} \frac{1}{|\mathbf{r} - \mathbf{r}_o|} dv, \text{ for } i = 1, 3; j = 1, 3, \quad (2.4)$$

and where x_1, x_2 , and x_3 represent x -, y -, and z -directions, respectively. The expressions of \mathbf{T}_{ij} for a cuboidal source volume can be found in [Bhattacharyya64] and [Sharma66]. Since \mathbf{T} is symmetric and its trace is equal to -1 when the observation is inside the cell and is 0 when the observation is outside the cell, only five independent elements need to be calculated.

Once \mathbf{T} is formed, the magnetic anomaly \mathbf{B}_a and its projection onto any direction of measurement are easily obtained by the inner product with the directional vector. The projection of the field \mathbf{B}_a onto different directions yields different anomalies commonly obtained in the magnetic survey. For instance, the vertical anomaly is simply B_{az} , the vertical component of \mathbf{B}_a , whereas the total field anomaly is, to first order, the projection of \mathbf{B}_a onto the direction of the inducing field \mathbf{B}_o .

2.2.2 Borehole data

In a borehole experiment, the three components are measured in the directions of local coordinate axes (l_1, l_2, l_3) defined according to the borehole orientation. Assuming that the borehole dip θ is measured downward from the horizontal surface and azimuth φ is measured eastward from the North; a commonly used convention has the l_3 -axis pointing downward along borehole, l_1 -axis pointing perpendicular to the borehole in the direction of the azimuth. The l_2 -axis completes the right-handed coordinate system and is 90° clockwise from the azimuth and perpendicular to the borehole. Based upon the above definition the rotation matrix that transforms three components of a vector in the global coordinate system to the components in the local coordinates is given by

$$\mathbf{R} = \begin{pmatrix} \cos \varphi \sin \theta & \sin \varphi \sin \theta & -\cos \theta \\ -\sin \varphi & \cos \varphi & 0 \\ \cos \varphi \cos \theta & \sin \varphi \cos \theta & \sin \theta \end{pmatrix} \quad (2.5)$$

If a vector is defined in local coordinates as $(l_1, l_2, l_3)^T$, and in global coordinates as $(g_1, g_2, g_3)^T$, then the following two relations hold:

$$\begin{aligned}(l_1, l_2, l_3)^T &= \mathbf{R}(g_1, g_2, g_3)^T \\ (g_1, g_2, g_3)^T &= \mathbf{R}^T(l_1, l_2, l_3)^T\end{aligned}$$

The rotation matrix \mathbf{R} therefore allows measured components in local coordinates to be rotated into global coordinate, or the components of the regional field to be rotated into local coordinates for use in regional removal.

2.2.3 Numerical implementation of forward modelling

We divide the region of interest into a set of 3D prismatic cells by using a 3D orthogonal mesh and assume a constant susceptibility within each cell. By equation (2.1), we have a uniform magnetization within each cell and its field anomaly can be calculated using equations (2.3) and (2.6). The actual anomaly that would be measured at an observation point is the sum of field produced by all cells having a non-zero susceptibility value. The calculation involves the evaluation of equation (2.3) in a 3D rectangular domain define by each cell. The program that performs this calculation is MAGSEN3D. As input parameters, the coordinates of the observation points and the inclination and declination of the anomaly direction must be specified for each datum. For generality, each component in a multi-component data set is specified as a separate datum with its own location and direction of projection.

2.3 Inversion methodology

Let the set of extracted anomaly data be $\mathbf{d} = (d_1, d_2, \dots, d_N)^T$ and the susceptibility of cells in the model be $\kappa = (\kappa_1, \kappa_2, \dots, \kappa_M)^T$. The two are related by the sensitivity matrix

$$\mathbf{d} = \mathbf{G}\kappa. \quad (2.6)$$

The matrix has elements g_{ij} which quantify the contribution to the i^{th} datum due to a unit susceptibility in the j^{th} cell. The program performs the calculation of the sensitivity matrix, which is to be used by the subsequent inversion. The sensitivity matrix provides the forward mapping from the model to the data during the entire inverse process. We will discuss its efficient representation via the wavelet transform in a separate section.

The first question that arises in the inversion of magnetic data concerns definition of the “model”. We choose magnetic susceptibility κ as the model for since the anomalous field is directly proportional to the susceptibility. The inverse problem is formulated as an optimization problem where a global objective function, ϕ , is minimized subject to the constraints in equation (2.6). The global objective functions consists of two components: a model objective function, ϕ_m , and a data misfit function, ϕ_d , such that

$$\begin{aligned}\min \phi &= \phi_d + \beta \phi_m \\ \text{s. t. } \kappa^l &\leq \kappa \leq \kappa^u,\end{aligned}$$

where β is a trade off parameter that controls the relative importance of the model smoothness through the model objective function and data misfit function. When the standard deviations of data errors are known, the acceptable misfit is given by the expected value ϕ_d and we will search for the value of β via an L-curve criterion [Hansen00] that produces the expected misfit. Otherwise, a user-defined β value is used. Bound are imposed through the projected gradient method so that the recovered model lies between imposed lower (κ^l) and upper (κ^u) bounds.

We next discuss the construction of a model objective function which, when minimized, produces a model that is geophysically interpretable. This function gives the flexibility to incorporate as little or as much information as possible. At the minimum, it drives the solution towards a reference model κ_o and requires that the model be relatively smooth in the three spatial directions. Here we adopt a right handed Cartesian coordinate system with positive north and positive down. Let the model objective function be

$$\begin{aligned}\phi_m(\kappa) = & \alpha_s \int_V w_s \{w(\mathbf{r})[\kappa(\mathbf{r}) - \kappa_o]\}^2 dv + \alpha_x \int_V w_x \left\{ \frac{\partial w(\mathbf{r})[\kappa(\mathbf{r}) - \kappa_o]}{\partial x} \right\}^2 dv \\ & + \alpha_y \int_V w_y \left\{ \frac{\partial w(\mathbf{r})[\kappa(\mathbf{r}) - \kappa_o]}{\partial y} \right\}^2 dv + \alpha_z \int_V w_z \left\{ \frac{\partial w(\mathbf{r})[\kappa(\mathbf{r}) - \kappa_o]}{\partial z} \right\}^2 dv,\end{aligned}$$

where the functions w_s , w_x , w_y and w_z are spatially dependent, while α_s , α_x , α_y and α_z are coefficients, which affect the relative importance of different components in the objective function. The reference model is given as κ_o and $w(\mathbf{r})$ is a generalized depth weighting function. The purpose of this function is to counteract the geometrical decay of the sensitivity with the distance from the observation location so that the recovered susceptibility is not concentrated near the observation locations. It should be noted that although traditionally the depth weighting is applied through the model objective function, practically applies it to the sensitivity matrix prior to compression, increasing the effectiveness of the wavelet transform. The details of the depth weighting function will be discussed in the next section.

The objective function in equation (2.7) has the flexibility to incorporate many types of prior knowledge into the inversion. The reference model may be a general background model that is estimated from previous investigations or it will be a zero model. The reference model would generally be included in the first component of the objective function but it can be removed, if desired, from the remaining terms; often we are more confident in specifying the value of the model at a particular point than in supplying an estimate of the gradient. The choice of whether or not to include κ_o in the derivative terms can have significant effect on the recovered model as shown through the synthetic example (section [RefModSection]). The relative closeness of the final model to the reference model at any location is controlled by the function w_s . For example, if the interpreter has high confidence in the reference model at a particular region, he can specify w_s to have increased amplitude there compared to other regions of the model, thus favouring a model near the reference model in those locations. The weighting functions w_x , w_y , and w_z can be designed to enhance or attenuate gradients in various regions in the model domain. If geology suggests a rapid transition zone in the model, then a decreased weighting on particular derivatives of the model will allow for higher gradients there and thus provide a more geologic model that fits the data.

Numerically, the model objective function in equation eq:mof is discretized onto the mesh defining the susceptibility model using a finite difference approximation. This yields:

$$\begin{aligned}\phi_m(\kappa) &= (\kappa - \kappa_o)^T (\alpha_s \mathbf{W}_s^T \mathbf{W}_s + \alpha_x \mathbf{W}_x^T \mathbf{W}_x + \alpha_y \mathbf{W}_y^T \mathbf{W}_y + \alpha_z \mathbf{W}_z^T \mathbf{W}_z) (\kappa - \kappa_o), \\ &\equiv (\kappa - \kappa_o)^T \mathbf{W}_m^T \mathbf{W}_m (\kappa - \kappa_o), \\ &= \|\mathbf{W}_m (\kappa - \kappa_o)\|^2,\end{aligned}$$

where \mathbf{m} and \mathbf{m}_o are M -length vectors representing the recovered and reference models, respectively. Similarly, there is an option to remove the reference model from the spatial derivatives in equation (2.7) such that

$$\begin{aligned}\phi_m(\kappa) &= (\kappa - \kappa_o)^T (\alpha_s \mathbf{W}_s^T \mathbf{W}_s) (\kappa - \kappa_o) + \kappa^T (\alpha_x \mathbf{W}_x^T \mathbf{W}_x + \alpha_y \mathbf{W}_y^T \mathbf{W}_y + \alpha_z \mathbf{W}_z^T \mathbf{W}_z) \kappa, \\ &\equiv (\kappa - \kappa_o)^T \mathbf{W}_s^T \mathbf{W}_s (\kappa - \kappa_o) + \kappa^T \mathbf{W}_m^T \mathbf{W}_m \kappa, \\ &= \|\mathbf{W}_s (\kappa - \kappa_o) + \mathbf{W}_m \kappa\|^2.\end{aligned}$$

In the previous two equations, the individual matrices \mathbf{W}_s , \mathbf{W}_x , \mathbf{W}_y , and \mathbf{W}_z are straight forward to calculated once the model mesh and the weighting functions $w(\mathbf{r})$ and w_s , w_x , w_y , w_z are defined. The cumulative matrix $\mathbf{W}_m^T \mathbf{W}_m$ is then formed for the chosen configuration.

The next step in setting up the inversion is to define a measure of how well the observed data are reproduced. Here we use the l_2 -norm measure

$$\phi_d = \|\mathbf{W}_d (\mathbf{G}\kappa - \mathbf{d})\|^2. \quad (2.7)$$

For the work here, we assume that the contaminating noise on the data is independent and Gaussian with zero mean. Specifying \mathbf{W}_d to be a diagonal matrix whose i^{th} element is $1/\sigma_i$, where σ_i is the standard deviation of the i^{th} datum

makes ϕ_d a chi-squared distribution with N degrees of freedom. The optimal data misfit for data contaminated with independent, Gaussian noise has an expected value of $E[\chi^2] = N$, providing a target misfit for the inversion. We now have the components to solve the inversion as defined in equation (2.7).

To solve the optimization problem when constraints are imposed we use the projected gradients method [Cala-maiMore87][Vogel02]. This technique forces the gradient in the Krylov sub-space minimization (in other words a step during the conjugate gradient process) to zero if the proposed step would make a model parameter exceed the bound constraints. The result is a model that reaches the bounds, but does not exceed them. This method is computationally faster than the log-barrier method because (1) model parameters on the bounds are neglected for the next iteration and (2) the log-barrier method requires the calculation of a barrier term. Previous versions of MAG3D used the logarithmic barrier method [Wright97][NocedalWright99].

The weighting function is generated by the program that is in turn given as input to the sensitivity generation program MAGSEN3D. This gives the user full flexibility in using customized weighting functions. This program allows user to specify whether to use a generalized depth weighting or a distance-based weighting that is useful in regions of largely varying topography. Distance weighting must be used when borehole data are present.

2.4 Depth Weighting and Distance Weighting

It is a well-known fact that static magnetic data have no inherent depth resolution. A numerical consequence of this is that when an inversion is performed, which minimizes $\int m(\mathbf{r})^2 dv$, subject to fitting the data, the constructed susceptibility is concentrated close to the observation locations. This is a direct manifestation of the kernel's decay with the distance between the cell and observation locations. Because of the rapidly diminishing amplitude, the kernels of magnetic data are not sufficient to generate a function that possess significant structure at locations that are far away from observations. In order to overcome this, the inversion requires a weighting to counteract this natural decay. Intuitively, such a weighting will be the inverse of the approximate geometrical decay. This gives cells at all locations equal probability to enter into the solution with a non-zero susceptibility.

2.4.1 Depth weighting for surface or airborne data

The sensitivity decays predominantly as a function of depth for surface data. Numerical experiments indicate that a function of the form $(z + z_o)^{-3}$ closely approximates the kernel's decay directly under the observation point provided that a reasonable value is chosen for z_o . The value of 3 in the exponent is consistent with the fact that, to first order, a cuboidal cell acts like a dipole source whose field decays as inverse distance cubed. The value of z_o can be obtained by matching the function $1/(z + z_o)^3$ with the field produced at an observation point by a column of cells. Thus we use a depth weighting function of the form

$$w(\mathbf{r}_j) = \left[\frac{1}{\Delta z_j} \int_{\Delta z_{ij}} \frac{dz}{(z + z_o)^\alpha} \right]^{1/2}, \quad j = 1, \dots, M. \quad (2.8)$$

For the inversion of surface data, where $\alpha = 3$, \mathbf{r}_j is used to identify the j^{th} cell, and Δz_j is its thickness. This weighting function is normalized so that the maximum value is unity. Numerical tests indicate that when this weighting is used, the susceptibility model constructed by minimizing the model objective function in equation (2.7), subject to fitting the data, places the recovered anomaly at approximately the correct depth.

If the data set involves highly variable observation heights the normal depth weighting function might not be most suitable. Distance weighting used for borehole data may be more appropriate as explained in the next section.

2.4.2 Distance weighting for borehole data

For data sets that contain borehole measurements, the sensitivities do not have a predominant decay direction, therefore a weighting function that varies in three dimensions is needed. We generalize the depth weighting used in surface data

inversion to form such a 3D weighting function called distance weighting:

$$w(\mathbf{r}_j) = \frac{1}{\sqrt{\Delta V_j}} \left\{ \sum_{i=1}^N \left[\int_{\Delta V_j} \frac{dv}{(R_{ij} + R_o)^\alpha} \right]^2 \right\}^{1/4}, \quad j = 1, \dots, M, \quad (2.9)$$

where $\alpha = 3$, V_j is the volume of j^{th} cell, R_{ij} is the distance between a point within the source volume and the i^{th} observation, and R_o is a small constant used to ensure that the integral is well-defined (chosen to be a quarter of the smallest cell dimension). This weighting function is also normalized to have a maximum value of unity. For inversion of borehole data, it is necessary to use this more general weighting. This weighting function is also advantageous if surface data with highly variable observation heights are inverted.

2.5 Wavelet Compression of Sensitivity Matrix

The two major obstacles to the solution of a large-scale magnetic inversion problem are the large amount of memory required for storing the sensitivity matrix and the CPU time required for the application of the sensitivity matrix to model vectors. This program library overcomes these difficulties by forming a sparse representation of the sensitivity matrix using a wavelet transform based on compactly supported, orthonormal wavelets. For more details, the users are referred to [LiOldenburg03][LiOldenburg10]. Here, we give a brief description of the method necessary for the use of the MAG3D library.

Each row of the sensitivity matrix in a 3D magnetic inversion can be treated as a 3D image and a 3D wavelet transform can be applied to it. By the properties of the wavelet transform, most transform coefficients are nearly or identically zero. When coefficients of small magnitudes are discarded (the process of thresholding), the remaining coefficients still contain much of the necessary information to reconstruct the sensitivity accurately. These retained coefficients form a sparse representation of the sensitivity in the wavelet domain. The need to store only these large coefficients means that the memory requirement is reduced. Further, the multiplication of the sensitivity with a vector can be carried out by a sparse multiplication in the wavelet domain. This greatly reduces the CPU time. Since the matrix-vector multiplication constitutes the core computation of the inversion, the CPU time for the inverse solution is reduced accordingly. The use of this approach increases the size of solvable problems by nearly two orders of magnitude.

Let \mathbf{G} be the sensitivity matrix and \mathcal{W} be the symbolic matrix-representation of the 3D wavelet transform. Then applying the transform to each row of \mathbf{G} and forming a new matrix consisting of rows of transformed sensitivity is equivalent to the following operation:

$$\tilde{\mathbf{G}} = \mathbf{G}\mathcal{W}^T, \quad (2.10)$$

where $\tilde{\mathbf{G}}$ is the transformed matrix. The thresholding is applied to individual rows of \mathbf{G} by the following rule to form the sparse representation $\tilde{\mathbf{G}}^S$,

$$\tilde{g}_{ij}^s = \begin{cases} \tilde{g}_{ij} & \text{if } |\tilde{g}_{ij}| \geq \delta_i \\ 0 & \text{if } |\tilde{g}_{ij}| < \delta_i \end{cases}, \quad i = 1, \dots, N, \quad (2.11)$$

where δ_i is the threshold level, and \tilde{g}_{ij} and \tilde{g}_{ij}^s are the elements of $\tilde{\mathbf{G}}$ and $\tilde{\mathbf{G}}^S$, respectively. The threshold level δ_i are determined according to the allowable error of the reconstructed sensitivity, which is measured by the ratio of norm of the error in each row to the norm of that row, $r_i(\delta_i)$. It can be evaluated directly in the wavelet domain by the following expression:

$$r_i(\delta_i) = \sqrt{\frac{\sum_{|\tilde{g}_{ij}| < \delta_i} \tilde{g}_{ij}^2}{\sum_j \tilde{g}_{ij}^2}}, \quad i = 1, \dots, N, \quad (2.12)$$

Here the numerator is the norm of the discarded coefficients and the denominator is the norm of all coefficients. The threshold level δ_{i_o} is calculated on a representative row, i_o . This threshold is then used to define a relative threshold $\epsilon = \delta_{i_o} / \max_j |\tilde{g}_{ij}|$. The absolute threshold level for each row is obtained by

$$\delta_i = \epsilon \max_j |\tilde{g}_{ij}|, \quad i = 1, \dots, N. \quad (2.13)$$

The program that implements this compression procedure is MAGSEN3D. The user is asked to specify the relative error r^* and the program will determine the relative threshold level δ_i . Usually a value of a few percent is appropriate for r^* . When both surface and borehole data are present, two different relative threshold levels are calculated by choosing a representative row for surface data and another for borehole data. For experienced users, the program also allows the direct input of the relative threshold level.

Elements of the program MAG3D

3.1 Introduction

The program library consists of the programs:

1. **MAGFOR3D**: performs forward modelling.
2. **PFWEIGHT**: calculates depth or distance weighting function.
3. **MAGSEN3D**: calculates sensitivity for the inversion.
4. **MAGINV3D**: performs 3D inversion of magnetic data
5. **MAGPRE3D**: multiplies the sensitivity file by the model to get the predicted data. This rarely used utility multiplies a model by the sensitivity matrix in to produce the predicted data. This program is included so that users who are not familiar with the wavelet transform and the structure of can utilize the available sensitivity matrix to carry out model studies.

Each of the above programs requires input files and the specification of parameters in order to run. However, some files are used by a number of programs. Before detailing the procedures for running each of the above programs, we first present information about these general files.

3.2 General files for MAG3D programs

There are seven general files which are used in MAG3D. All are in ASCII text format. Input files can have any user-defined name. *Program output files have restricted file names that will be over-written if already in the directory.* Also the filename extensions are not important. Many prefer to use the filename convention so that files are more easily read and edited in the Windows environment. File and file locations may have spaces in the name or path, but it is discouraged. The file name (absolute or relative path) must be 500 characters or less in length. The files contain components of the inversion:

3.2.1 Mesh file

The mesh defines the model region and has the following structure:

NE	NN	NZ	
E_o	N_o	Z_o	
ΔE_1	ΔE_2	\dots	ΔE_{NE}
ΔN_1	ΔN_2	\dots	ΔN_{NN}
ΔZ_1	ΔZ_2	\dots	ΔZ_{NZ}

- NE : Number of cells in the East direction.
- NN : Number of cells in the North direction
- NZ : Number of cells in the vertical direction
- E_o, N_o, Z_o : Coordinates, in meters, of the southwest top corner, specified in (Easting, Northing, Elevation). The elevation can be relative to a reference elevation other than the sea level, but it needs to be consistent with the elevation used to specify the locations, observations, and topography files.
- ΔE_n : n^{th} cell width in the easting direction (ordered W to E).
- ΔN_n : n^{th} cell width in the northing direction (ordered S to N).
- ΔZ_n : n^{th} cell thickness (ordered top to bottom).

The mesh can be designed in accordance with the area of interest and the spacing of the data available in the area. In general, the mesh consists of a core region which is directly beneath the area of available data, and a padding zone surrounding this core mesh. Within the core mesh, the size of the cells should be comparable with the spacing of the data. There is no restriction on the relative position of data location and nodal points in horizontal direction. The cell width in this area is usually uniform. Beyond the core region, the mesh should be padded with cells that increase (typically no more than 40% of the previous length).

The vertical position of the mesh is specified in elevation. This is to accommodate the inversion of a data set acquired over a topographic surface. When there is strong topographic relief, which the user wishes to incorporate it into the inversion, special care should be taken to design the mesh. A conceptually simple approach is first to design a rectangular mesh whose top (specified by Z_o) is just below the highest elevation point, and then to strip off cells that are above the topographic surface. This is the approach taken in . The number of cells to be stripped off in each column is determined by the user-supplied topography file. Only the remaining cells will be used in the forward modelling or included in the inversion as model parameters.

Example

This example shows a mesh that consists of 26 cells in easting, 27 cells in the northing, and 23 cells in the vertical directions. The top of the mesh is located at 0 m of elevation and the southwest corner is at -350 m easting and -400 m northing. The cells in the core portion of the mesh are all 50 m \times 50 m \times 25 m. There are three cells in the padding zone in every direction except the top of the core mesh.

3.2.2 Topography file

This file is used to define the surface topography of a mesh/model by the elevation at different locations. Lines starting with ! are comments. The topography file has the following general structure:

26	27	23				
-350	-400	0				
200	100	50	21*50.0	50	100	200
200	100	50	20*50.0	50	100	200
20*25.0	50	100	200			

!	comment	
npt		
E ₁	N ₁	ELEV ₁
E ₂	N ₂	ELEV ₂
⋮	⋮	⋮
E _n	N _n	ELEV _n

Parameter definitions:

- npt: Number of points defining the topographic surface.
- E_i: Easting of the i^{th} point on the surface.
- N_i: Northing of the i^{th} point on the surface.
- ELEV_i: Elevation (metres) of the i^{th} point on the profile.

The lines in this file can be in any order as long as the total number is equal to npt. The topographic data need not be supplied on a regular grid. GIF inversion codes assume a set of scattered points for generality and use a triangulation-based interpolation to determine the surface elevation above each column of cells. To ensure the accurate discretization of the topography, it is important that the topographic data be supplied over the entire area above the model and that the supplied elevation data points are not too sparse.

NOTE 2: Only the cells completely below the (interpolated) topographic surface are kept. The cells above or at the topographic surface are removed from the model, although these must still be included in the as if they are a part of the model. For input model files these cells can be assigned any value. The recovered model produced by inversion program also includes the cells that are excluded from the model, but these cells will have unrealistic values and be set to -100.

Example:

The following is an example of a topography file:

3.2.3 Magnetic observations file

This file is used to specify the inducing field parameters, anomaly type, observation locations, and the observed magnetic anomalies with estimated standard deviation. The values of parameters specifying the inducing field anomaly type and observation locations are identical to those in file. The output of the forward modelling program has the same structure except that the column of standard deviations for the error is omitted. Lines starting with ! are comments. The structure of the magnetics observations file is:

```

2007
12300.00  9000.00  0.109411E+04
12300.00  9025.00  0.109545E+04
12300.00  9050.00  0.109805E+04
12300.00  9075.00  0.110147E+04
12300.00  9100.00  0.110555E+04
12300.00  9125.00  0.111011E+04
12300.00  9150.00      1114.9
12300.00  9175.00  0.111971E+04

```

```

! comment
incl decl geomag
ainc adec dir
ndat
E1  N1  ELEV1  [ainc1  adec1]  Mag1  Err1
E2  N2  ELEV2  [ainc2  adec2]  Mag2  Err2
⋮    ⋮    ⋮      ⋮          ⋮      ⋮
Endat Nndat ELEVndat [aincndat adecndat] Magndat Errndat

```

Parameter definitions:

- incl, decl: Inclination and declination of the inducing magnetic field. The declination is specified positive east with respect to the northing used in the mesh and locations files. The inclination is positive down.
- geomag: Strength of the inducing field in nanoTesla (nT).
- ainc, adec, idir: Inclination and declination of the anomaly projection. Set idir=0 for a multi-component dataset and idir=1 for a single component dataset where all the observations have the same inclination and declination of the anomaly projection.
- ndat: Number of observations. When single component data are specified the number of observations is equal to the number of data locations. When multi-component data are specified the number of observations will exceed the number of data locations. For example, if three-component data are specified at N locations, the number of observations is $3N$.
- E, N, ELEV: Easting, northing, and elevation of the observation measured in meters. Elevation should be above the topography for surface data, and below the topography for borehole data. The observation locations can be listed in any order.
- (ainc _{n} , adec _{n}): Inclination and declination of the anomaly projection for n th observation. This is used only when idir=0. The brackets $[\dots]$ indicate that these two field are optional depending upon the value of idir.
- Mag _{n} : Magnetic anomaly data, measured in nT.
- Err _{n} : Standard deviation of Mag _{n} . This represents the absolute error. It must be positive and non-zero.

NOTE: It should be noted that the data are **extracted anomalies** which are derived by removing the regional from the field measurements. It is crucial that the data be prepared as such. The total field anomaly is calculated when ainc=incl and adec=decl. An example is inputting the vertical field anomaly, B_z, calculated by setting ainc=90 and adec=0. The easting and northing components are respectively given by the inclination and declination pairs (0, 90) and (0, 90). The user can specify other (ainc, adec) pairs to calculate the other anomaly

components such as the Bx or By. Easting, northing, and elevation information should be in the same coordinate system as defined in the mesh.

Examples

The following is an example of total-field anomaly data.

```
! surface data
45.0    45.0    50000 ! incl, decl, geomag
45.0    45.0     1 ! ainc, adec, direction of anomaly (dir)
144    ! Number of data
0.00    0.00    1.00    0.174732E+02    0.598678E+01
0.00    50.00    1.00    0.265550E+02    0.613890E+01
0.00    100.00    1.00    0.311366E+02    0.629117E+01
:       :       :       :       :
1000.00  900.00    1.00   -0.109595E+01    0.530682E+01
1000.00  950.00    1.00   -0.902209E+01    0.523738E+01
1000.00 1000.00    1.00   -0.397501E+01    0.518496E+01
```

3.2.4 Model file

This file contains the cell property values (SI) of the model and is the most common of the model files. Inversion models (forward, initial, reference, recovered, and lower and upper bounds) are in this format. The following is the file structure of the model file

```
 $m_{1,1,1}$ 
 $m_{1,1,2}$ 
:
 $m_{1,1,NZ}$ 
 $m_{1,2,1}$ 
:
 $m_{1,j,k}$ 
:
 $m_{NN,NE,NZ}$ 
```

Each $m_{i,j,k}$ is the property in the $m_{i,j,k}$ model cell. $[i, j, k]=[1, 1, 1]$ is defined as the cell at the top, south-west corner of the model. The total number of lines in this file should equal $(NN \times NE \times NZ)$, where (NN) is the number of cells in the north direction, (NE) is the number of cells in the east direction, and (NZ) is the number of cells in the vertical direction. The model ordering is performed first in the z-direction (top-to-bottom), then in the easting, and finally in the northing.

NOTE: Only the cells completely below the (interpolated) topographic surface are kept within an inversion. The cells above or at the topographic surface are removed from the model, although these must still be included in the as if they are a part of the model. For input model files these cells can be assigned any value. The recovered model produced by inversion program also includes the cells that are excluded from the model, but these cells will have unrealistic values set to -100.

Active cells file

This file is optional. The active cells file contains information about the cells that will be incorporated into the inversion. It has exact same format as the , and thus must be the same size, with one exception. Values of this file are restricted to either -1, 0 or 1. By default, all cells below the earth's surface are active (1) and incorporated into the inversion. Inactive cells are set to the values of the reference model and influence the forward modelling. There are two kinds of inactive cells:

- inactive cells that *do not* influence the model objective (set to 0) and
- inactive cells that *do* influence the model objective function (set to -1).

3.2.5 Weights file

This file supplies the user-based weights that acts upon the model objective function. Each set of weights correspond to the functions (e.g., w_x) given in the *model objective function*. For ease, the weights in geographic coordinates are provided by the user. The following is the file structure is for the weights file:

$W.S_{1,1,1}$	\cdots	$W.S_{NN,NE,NZ}$
$W.E_{1,1,1}$	\cdots	$W.E_{NN,NE-1,NZ}$
$W.N_{1,1,1}$	\cdots	$W.N_{NN-1,NE,NZ}$
$W.Z_{1,1,1}$	\cdots	$W.Z_{NN,NE,NZ-1}$

Parameter definitions:

- $W.S_{i,j,k}$: Cell weights for the smallest model component in the *model objective function*.
- $W.E_{i,j,k}$: Cell weights for the interface perpendicular to the easting direction.
- $W.N_{i,j,k}$: Cell weights for the interface perpendicular to the northing direction.
- $W.S_{i,j,k}$: Cell weights for the interface perpendicular to the vertical direction.

Within each part, the values are ordered in the same way as in *model file*, however, they can be all on one line, or broken up over several lines. Since the weights for a derivative term are applied to the boundary between cells, the weights have one fewer value in that direction. For instance, the weights for the derivative in easting direction has $(NE - 1) \times NN \times NZ$ values, whereas the number of cells is $NE \times NN \times NZ$.

If the surface is supplied, the cell weights above the surface will be ignored. It is recommended that these weights be assigned a value of -1.0 to avoid confusion. If `null` is entered instead of the weights file, then all of the cell weights will be set equal (1.0).

Running the programs

The software package MAG3D uses five general codes:

- MAGFOR3D: performs forward modelling.
- PFWEIGHT: calculates the depth weighting function.
- MAGSEN3D: calculates sensitivity.
- MAGINV3D: performs 3D inversion of magnetic data
- MAGPRE3D: multiplies the sensitivity file by the model to get the predicted data.

This section discusses the use of these codes individually.

4.1 Introduction

All programs in the package can be executed under Windows or Linux environments. They can be run by typing the program name followed by a control file in the `command prompt` (Windows) or `terminal` (Linux). They can be executed directly on the command line or in a shell script or batch file. When a program is executed without any arguments, it will print the usage to screen.

4.1.1 Execution on a single computer

The command format and the control, or input, file format on a single machine are described below. Within the command prompt or terminal, any of the programs can be called using:

```
program arg1 [arg2 ... argi]
```

where:

- `program`: the name of the executable
- `argi`: a command line argument, which can be a name of corresponding required or optional file. Typing as the control file, serves as a help function and returns an example input file. Some executables do not require control

files and should be followed by multiple arguments instead. This will be discussed in more detail later in this section. Optional command line arguments are specified by brackets: []

Each control file contains a formatted list of arguments, parameters and filenames in a combination and sequence specific for the executable, which requires this control file. Different control file formats will be explained further in the document for each executable.

4.1.2 Execution on a local network or commodity cluster

The MAG3D program library's main programs have been parallelized with Message Pass Interface (MPI). This allows running these codes on more than one computer in parallel. MPI installation package can be downloaded from <http://www.mcs.anl.gov/research/projects/mpich2/>. The following are the requirements for running an MPI job on a local network or cluster:

- An identical version of MPI must be installed on all participating machines
- The user must create an identical network account with matching “username” and “password” on every machine
- Both the executable folder and the working directory should be “shared” and visible on every participating computer
- Before the MPI job is executed, the firewall should be turned off on every participating computer
- The path should be defined to the executable directory

The following is an example of a command line executing an MPI process:

```
C:\Program Files\MPICH2\bin\mpiexec.exe -machinefile machine.txt nproc  
-priority 0 maginv3d
```

An explanation of the arguments used in this command line are:

- Properly defined path to the `mpiexec`.
- The list of participating machines will be read from a “machine file.”
- Name of the machine file. This file lists the network names of the participating machines and number of processors to be allocated for the MPI job for each machine. The following is an example of a machine file:

<code>machine01</code>	<code>16</code>
<code>machine02</code>	<code>16</code>

In this simple example, there are two participating machines (named `machine01` and `machine02`) are required to allocate 16 processors for the MPI job.

- The total number of allocated processors. This number should be equal to the sum of all processors listed for all machines in the machine file.
- Sets the priority of the process. Integer grades from -1 (lowest) to 4 (highest) follow. Higher priority means that RAM and processing resources will be primarily allocated for this process, at expense of lower priority processes. Generally, a large job should be assigned a lower priority, as selective resource allocation may slow down other important processes on the computer, including those needed for stable functioning of the operating system.
- The name of the executable. In our case it is assumed that there is an existing path to the executable directory, otherwise proper path should be provided.

4.2 Input and output files

4.2.1 MAGFOR3D

This program performs forward modelling. Command line usage:

```
magfor3d mesh.msh obs.loc model.sus [topo.dat]
```

and will create the forward modelled data file `magfor3d.mag`.

Input files

All files are in ASCII text format - they can be read with any text editor. Input files can have any name the user specifies. Details for the format of each file can be found in *Elements of the program MAG3D*. The files associated with MAGFOR3D are:

- `mesh.msh`: The 3D *mesh*.
- `obs.loc`: The observation *locations*.
- `model.den`: The susceptibility *model* in SI.
- `topo.dat`: Surface *topography* (optional). If omitted, the surface will be treated as being flat and the top of the 3D mesh.

Output file

The created file is `magfor3d.mag`. The file format is that of the *data file* without the associated standard deviations. The forward modelled data are in **nT**.

4.2.2 PFWEIGHT

This program performs the depth weighting function calculation. This allows users the flexibility to either use this code or their in-house code for the weighting calculation. Command line usage:

```
pfweight weights.inp [nThreads]
```

For a sample input file type:

```
pfweight -inp
```

The argument specifying the number of CPU threads used in the OpenMP format is optional. If this argument is not given to the program, chooses to use all of the CPU threads on the machine. This argument allows the user to specify half, for example, of the threads so that the program does not take all available RAM. Note that this option is not available in the MPI-based code used for clusters.

Input files

Format of the control file:

The input parameters for the control file are:

- `type`: Use MAG to have `pfweight` read magnetic location files.
- `mesh.msh`: Name of 3D *mesh file*.

```
type           ! input type:  MAG | GRAV
mesh.msh       ! mesh file
obs.grv        ! observations file
topo.dat       ! topography file | null
iwt            ! weighting type
alpha, znot    ! weighting parameters | null
```

- `obs.grv` The *data file* that contains the observation locations and the observed magnetic anomaly with estimated standard deviation.
- `topo.dat`: Surface *topography file*. If `null` is entered, the surface will be treated as being flat on the top of the mesh.
- **iwt**: An integer (1 or 2) identifying the type of generalized depth weighting to use in the inversion.

=1 for depth weighting (not applicable to borehole data);

=2 for distance weighting.

- **alpha, znot**: Parameters defining the depth weighting function:

When `iwt=1`, `alpha` and `znot` are used as α and z_0 to define the *depth weighting*

When `iwt=2`, `alpha` and `znot` are used as α and (R_o) to define the *distance weighting*

NOTE 1: If `null` is entered on this line (line 6), then the program sets `alpha=2` and calculates the value of z_0 based upon the mesh and data location. This is true for `iwt=1` or `iwt=2`

NOTE 2: For most inversions, setting this input line to `null` is recommended. The option for inputting α and z_0 is provided for experienced users who would like to investigate the effect of the generalized depth weighting for special purposes. The value of α should normally be close to 2.0. Smaller values of give rise to weaker weighting as distance increases from the observation locations.

Example of input file

```
MAG           ! mag data
mesh.msh      ! mesh file
obs_noise.mag  ! data file
null          ! topography | null
2             ! iwt=1 depth, =2 distance
null          ! alpha, znot | null
```

Output files

The program outputs `x_weight.txt`. A file in the format, which contains weights for each cell, based on *depth weighting* (`x = "depth"`) or *distance weighting* (`x = "distance"`). A log file `pfweight.log` is also written.

4.2.3 MAGSEN3D

This program performs the sensitivity calculation. Command line usage:

```
magsen3d magsen3d.inp [nThreads]
```

For a sample input file type:

```
magsen3d -inp
```

The argument specifying the number of CPU threads used in the OpenMP format is optional. If this argument is not given to the program, chooses to use all of the CPU threads on the machine. This argument allows the user to specify half, for example, of the threads so that the program does not take all available RAM. Note that this option is not available in the MPI-based code used for clusters.

Input files

Format of the control file:

```
mesh      ! mesh file
obs       ! observations file
topo      ! topography file | null
x_weight.txt ! weights file
wvltx     ! wavelet type: daubx | symmx | NONE
itol, eps ! wavelet parameters | null
Diag      ! Diagnostic files (1= output | 0= none)
```

The input parameters for the control file are:

- **mesh:** Name of 3D *mesh file*.
- **obs:** The *data file* that contains the observation locations. Note for sensitivity calculations, standard deviations are not required, but this file may be the observations that will be used in the inversion (with uncertainties).
- **topo:** Surface *topography*. If `null` is entered, the surface will be treated as being flat on top of the mesh.
- **x_weight.txt:** A file (in the *model file format*) giving the sensitivity the weighting function. This can be user-specific or generated from *pfweight*. It does *not* have to specifically be named “distance_weight.txt” or “depth_weight.txt”.
- **wvltx:** A five-character string identifying the type of wavelet used to compress the sensitivity matrix. The types of wavelets available are Daubechies wavelet with 1 to 6 vanishing moments (`daub1`, `daub2` and so on) and Symmlets with 4 to 6 vanishing moments (`symm4`, `symm5`, `symm6`). Note that `daub1` is the Haar wavelet and `daub2` is the Daubechies-4 wavelet. The Daubechies-4 wavelet is suitable for most inversions, while the others are provided for user’s experimentation. If `NONE` is entered, the program does not use wavelet compression.
- **itol, eps:** An integer and real number that specify how the wavelet threshold level is to be determined. This line is

`itol=1`: program calculates the relative threshold and `eps` is the relative reconstruction error of the sensitivity. A reconstruction error of 0.05 (95%) is usually adequate.

`itol=2`: the user defines the threshold level and `eps` is the threshold to be used. If `null` is entered on this line, a default relative reconstruction error of 0.05 (e.g. 5%) is used and the relative threshold level is calculated (i.e., `itol=1`, `eps=0.05`).

NOTE The detailed explanation of threshold level and reconstruction error can be found in the *wavelet section* of this manual.

- **Diag:** Option to output (`Diag=1`) or not output (`Diag=0`) diagnostic files. These files are: (1) the predicted data for a model of $\rho = 0.1$ with the wavelet compressed sensitivity, (2) the predicted data for a model of $\rho = 0.1$ with the full sensitivity, (3) the averaged sensitivity in each cell based on the wavelet compression. An extra line in the log file is also written giving the user the achieved reconstruction error (e.g. `eps` when `itol=1` from above).

Example of input file

```
mesh.msh      ! mesh file
obs.mag       ! observations file
topo.dat      ! topography file | null
distance_weight.txt ! weights file
daub2        ! wavelet type: daubx | symmx | NONE
null         ! wavelet parameters | null
0            ! Diagnostic files (1= output | 0= none)
```

Output files

The program `magsen3d` outputs three files (with diagnostic files being optional output). They are:

1. `magsen3d.log`: A log file summarizing the run of the program.
2. `maginv3d.mtx`: The sensitivity matrix file to be used in the inversion. This file contains the sensitivity matrix, generalized depth weighting function, mesh, and discretized surface topography. It is produced by the program and its name is not adjustable. It is very large and may be deleted once the work is completed.
3. `sensitivity.txt`: This file is a *model file* that contains the average sensitivity for each cell. This file can be used for depth of investigation analysis or for use in designing special model objective function weighting.
4. Diagnostic files as described above to examine the wavelet compression properties, if chosen (`Diag=1`).

4.2.4 MAGINV3D

This program actually performs the 3D inversion of magnetic data. Command line usage is:

```
maginv3d maginv3d.inp [nThreads]
```

For a sample input file type:

```
maginv3d -inp
```

The argument specifying the number of CPU threads used in the OpenMP format is optional. If this argument is not given to the program, chooses to use all of the CPU threads on the machine. This argument

allows the user to specify half, for example, of the threads so that the program does not take all available RAM. Note that this option is not available in the MPI-based code used for clusters.

Input files

Input files can be any file name. If there are spaces in the path or file name, you *MUST* use quotes around the entire path (including the filename). Files that may be used by the inversion are:

1. `obs.mag`: Mandatory *observations file*.
2. `maginv3d.mtx`: Mandatory sensitivity matrix from *MAGSEN3D*
3. `initial.sus`: Optional initial *model file*. This can be substituted by a value within the input file (see below).
4. `ref.sus`: Optional reference *model file*. This can be substituted by a value within the input file (see below).
5. `active.txt`: Optional ref: *active model file* <activeFile>
6. `upperBound.sus`: Optional upper bounds *model file*. A value can be used to set a global bound (see below).
7. `lowerBound.sus`: Optional lower bounds *model file*. A value can be used to set a global bound (see below).
8. `weights.wt`: Optional *weighting file*.
9. `maginv3d.inp`: The control file containing the options. Does not need to be specifically called “maginv3d.inp”.

Format of the control file has been changed since previous version. Any numeric entries beyond the trade-off parameter, and tolerance should be preceded by VALUE. The input files has been modified as follows:

```
mode          ! 1 (Find beta), 2 (Fixed beta)
par, tolc     ! beta parameter and tolerance
obs           ! Observed data
matrixFile    ! Sensitivity matrix
init          ! file | VALUE [SI]
ref           ! file | VALUE [SI]
act           ! file | null
lowerBounds   ! VALUE [bl] | file
upperBounds   ! VALUE [bu] | file
 $\alpha_s, \alpha_x, \alpha_y, \alpha_z$  ! alphas | Le Ln Lz | null
SMOOTH_MOD_DIF ! SMOOTH_MOD_DIF | SMOOTH_MOD
weight        ! Weighting file | null
```

The parameters within the control file are:

- mode: An integer specifying one of two choices for determining the trade-off parameter.
 1. mode=1: the program chooses the trade off parameter by carrying out a line search so that the target value of data misfit is achieved (e.g., $\phi_d^* = N$).
 2. mode=2: the user inputs the trade off parameter.

- `par, tol` Two real numbers that are dependent upon the value of `mode`.
 1. `mode=1`: the target misfit value is given by the product of `par` and the number of data N , i.e., `par=1` is equivalent to $\phi_d^* = N$ and `par=0.5` is equivalent to $\phi_d^* = N/2$. The second parameter, `tol`, is the misfit tolerance in fractional percentage. The target misfit is considered to be achieved when the relative difference between the true and target misfits is less than `tol`. Normally, `par=1` is ideal if the true standard deviation of error is assigned to each datum. When `tol=0`, the program assumes a default value of `tol=0.02` since this number must be positive.
 2. `mode=2`: `par` is the user-input value of trade off parameter. In this case, `tol` is not used by the program.

NOTE: When both `par` and `tol` are used. When only `par` is used. When `mode=3`, neither nor `tol` are used. However, the third line should always have two values.

- `obs`: Input data file. The file must specify the standard deviations of the error. By definition these values are greater than zero.
- `matrixFile`: The binary file of sensitivities created by *MAGSEN3D*.
- `initial`: The initial susceptibility model (SI) can be defined as a value for uniform models (e.g. `VALUE 0.001`), or by a filename. The initial model must be within the upper and lower bounds.
- `ref`: The reference susceptibility model (SI) can be defined as a value for uniform models (e.g. `VALUE 0`), or by a filename (for non-uniform reference models).
- `active`: The *active model file* defining which cells in the model are allowed to be solved.
- `lowerBound`: The lower bounds model (SI) can be defined as a value for uniform models (e.g., `VALUE -1`) or by a filename.
- `upperBound`: The upper bounds model (SI) can be defined as a value for uniform models (e.g., `VALUE 1`) or by a filename.
- $\alpha_s, \alpha_x, \alpha_y, \alpha_z$: Coefficients for the each model component. α_s is the smallest model component. Coefficient for the derivative in the easting direction. α_y is the coefficient for the derivative in the northing direction. The coefficient α_z is for the derivative in the vertical direction.

If `null` is entered on this line, then the above four parameters take the following default values: $\alpha_s = 0.0001, \alpha_x = \alpha_y = \alpha_z = 1$. All alphas must be positive and they cannot be all equal to zero at the same time.

NOTE: The four coefficients in line 9 of the control file may be substituted for three corresponding *length scales* L_x, L_y and L_z and are in units of metres. To understand the meaning of the length scales, consider the ratios $\alpha_x/\alpha_s, \alpha_y/\alpha_s$ and α_z/α_s . They generally define smoothness of the recovered model in each direction. Larger ratios result in smoother models, smaller ratios result in blockier models. The conversion from α 's to length scales can be done by:

$$L_x = \sqrt{\frac{\alpha_x}{\alpha_s}}; L_y = \sqrt{\frac{\alpha_y}{\alpha_s}}; L_z = \sqrt{\frac{\alpha_z}{\alpha_s}}$$

When user-defined, it is preferable to have length scales exceed the corresponding cell dimensions. Typically having length scales of four cell widths are a good starting point.

- `SMOOTH_MOD`: This option was not available in previous versions of the code and can be used to define the reference model in and out of the derivative terms. The options are: `SMOOTH_MOD_DIF`

(reference model is defined in the derivative terms) and SMOOTH_MOD (reference model is defined in only the smallest term). See equation details on the *model objective function* section for details.

- `weights`: Name of the *weights file* containing weighting matrices. If `null` is entered, default values of unity are used (no extra weighting).

Example of control file

Below is an example of a control file with comments:

```
1          ! 1 (Find beta), 2 (Fixed beta)
1.0, 0.05  ! beta parameter and tolerance
obs_noisy.mag ! Observed data
maginv3d.mtx ! Sensitivity matrix
VALUE 0.001 ! file | VALUE [SI]
VALUE 0     ! file | VALUE [SI]
null       ! file | null
VALUE 0     ! VALUE [bl] | file
VALUE 1     ! VALUE [bu] | file
200 200 200 ! alphas | Le Ln Lz | null
SMOOTH_MOD  ! SMOOTH_MOD_DIF | SMOOTH_MOD
null       ! Weighting file | null
```

Output files

Five general output files are created by the inversion. They are:

1. `maginv3d.log`: The log file containing the minimum information for each iteration and summary of the inversion.
2. `maginv3d.out`: The “developers” log file containing the details of each iteration including the model objective function values for each component, number of conjugate gradient iterations, etc.
3. `maginv3d_xxx.sus`: Susceptibility (SI) *model files* output after each “xxx” iteration (i.e., `maginv3d_012.sus`)
4. `maginv3d_xxx.pre`: *Predicted data files* (without uncertainties) output after each “xxx” iteration.

4.2.5 MAGPRE3D

This utility multiplies a model by the stored sensitivity matrix in to produce predicted data. This program is included so that users who are not familiar with the wavelet transform and the structure of could utilize the available sensitivity matrix to carry out modelling exercises. The command line usage is:

```
magpre3d maginv3d.mtx obs.loc model.sus
```

Input files

1. `maginv3d.mtx`: The sensitivity matrix file create from *MAGSEN3D*.
2. `obs.loc`: The magnetic *location file*.
3. `model.sus`: The susceptibility *model file* in SI.

Output file

The output file is a *predicted data file* (omitting uncertainty column) and is named `magpre3d.mag`. This program can be used to reproduce output predicted files from *MAGINV3D*.

Note: A new example has been developed for demonstrating functionality specific to v6.0. The example can be completed using v5.0, however some functionality may not exist in v5.0 and the format of certain input files may differ slightly.

Mag3d v6.0

Here, the program libraries for Mag3d v6.0 will be used to:

- define a magnetic susceptibility model on a tensor mesh
- predict total magnetic intensity data for a synthetic susceptibility model
- generate sensitivity weights for the inverse problem
- compute and store the sensitivity matrix
- invert total magnetic intensity data to recover a susceptibility model

Zip folders containing all necessary files can be downloaded here:

- [Files for mag3d v6 example](#)

The full example is parsed into 5 sections:

5.1 Create Model

Here the code **blk3cell.exe** and the input file **blk3cell.inp** are used to create a susceptibility model on the tensor mesh provided (**mesh.txt**). Files relevant to this part of the example are in the sub-folder *model*. Before running this example, you may want to do the following:

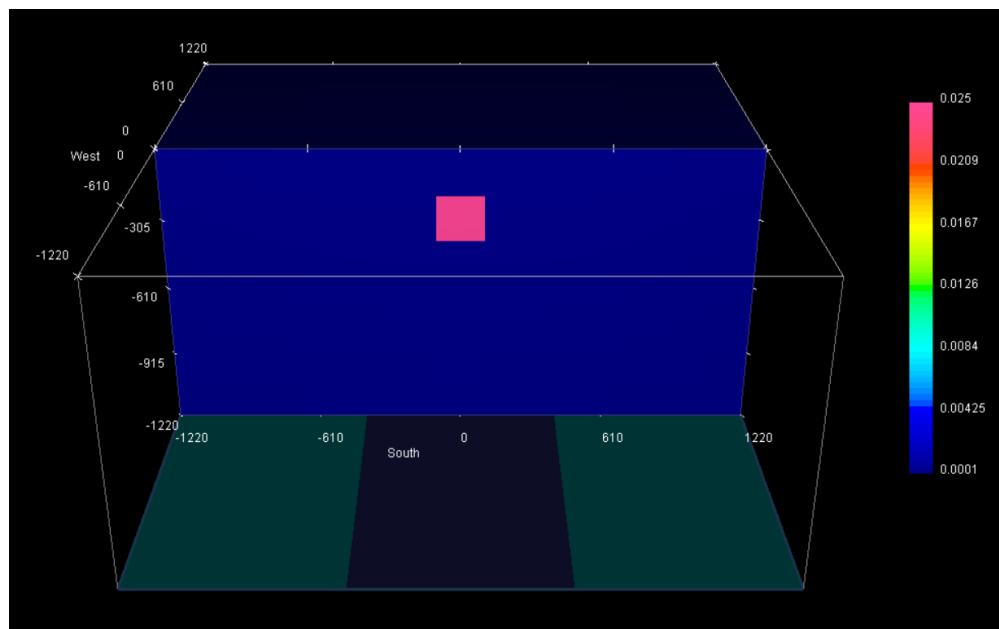
- [Download and open the zip folder containing the entire mag3d example](#) (if not done already)
- Learn how to [run blk3cell](#)
- Learn the format of the [input file](#)

Here is the input file used to generate the synthetic model.

The resulting tensor model shows a more susceptible block ($\chi = 0.025$ SI) within minimally susceptible background ($\chi_b = 0.0001$ SI). The region has a constant topography of 0 m.

```
blk3cell - Notepad
File Edit Format View Help
1e-4                                ! background value
1                                  ! number of blocks
-100 100 -100 100 -200 -400 0.025 ! e1 e2 n1 n2 z1 z2 value

Ln 3, Col 48    100%    Windows (CRLF)    UTF-8
```



Note: This example has been developed to demonstrate functionality specific to v6.0. The example can be completed using v5.0, however some functionality may not exist in v5.0 and the format of certain input files may differ slightly.

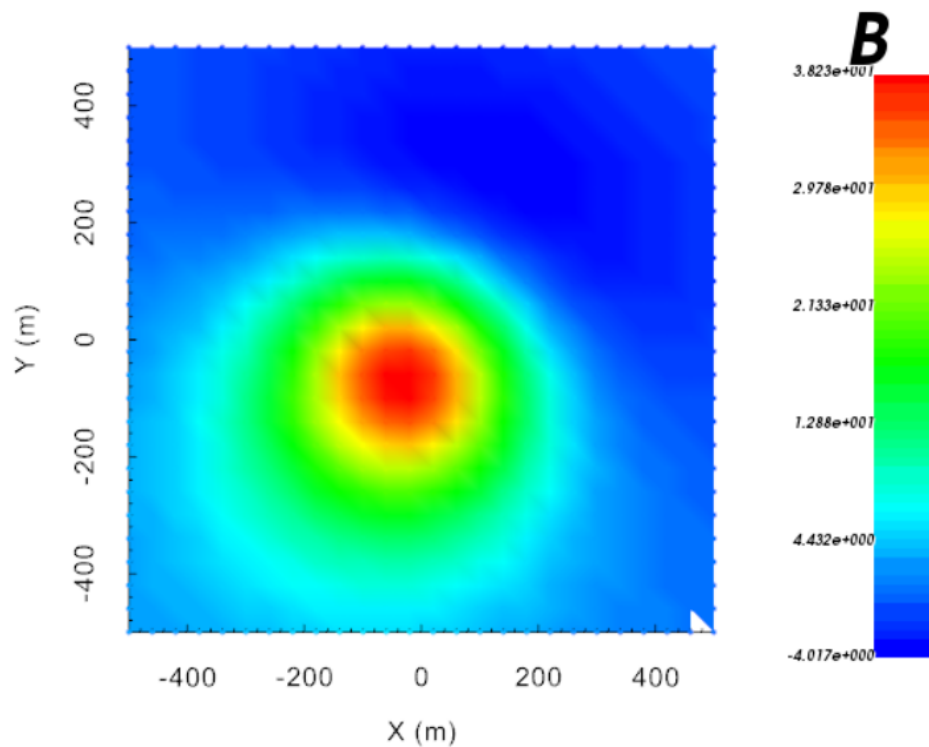
5.2 Forward Modeling

Here the code **magfor3d_v60.exe** is used to forward model the magnetic anomaly for the mesh and susceptibility model provided. We consider an airborne survey with a uniform station spacing of 40 m and a flight height of 30 m. The inclination, declination and background field intensity are 60 degrees, 25 degrees and 50,000 nT, respectively.

Files relevant to this part of the example are in the sub-folder *fwd*. We used the same model that was created in the *create model* section. Before running this example, you may want to do the following:

- Download and open the zip folder containing the entire mag3d example (if not done already)
- Learn how to run *magfor3d_v60*
- There is no input file

The total magnetic intensity anomaly (nT) is shown below.



5.3 Depth/Distance Weighting

Here the code **pfweight.exe** and the input file **pfweight.inp** is used to generate distance weighting for the inversion. Files relevant to this part of the example are in the sub-folder *pfweights*. Before running this example, you may want to do the following:

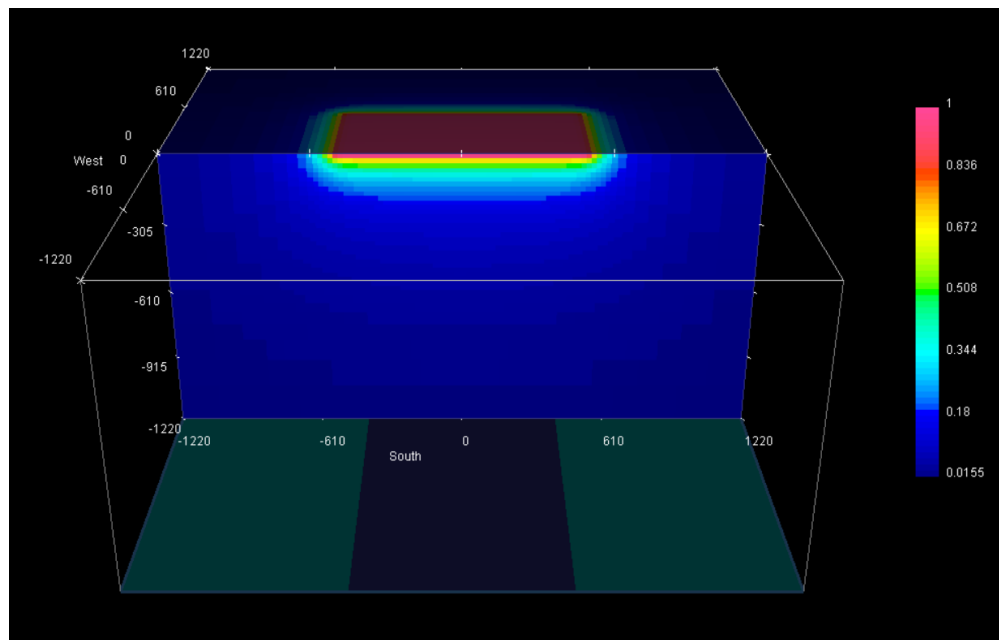
- Download and open the zip folder containing the entire mag3d example (if not done already)
- Learn how to *run pfwweight and learn the format of the input files*

Here is the input file for **pfweight.exe**

```
pfweight - Notepad
File Edit Format View Help
MAG          ! MAG | GRAV | GG
mesh.txt     ! tensor mesh
dobs.mag     ! observations/survey file
null        ! topography file | null
2           ! 1=depth weighting | 2=distance weighting
null        ! beta, z0 | null
```

Ln 3, Col 41 100% Windows (CRLF) UTF-8

The resulting distance weights are plotted on the mesh below.



Note: This example has been developed to demonstrate functionality specific to v6.0. The example can be completed using v5.0, however some functionality may not exist in v5.0 and the format of certain input files may differ slightly.

5.4 Computing Sensitivities

Here the code **magsen3d_v60.exe** is used to compute and output the sensitivities for use in the inversion. Here, we generate we use the input file **sensitivity_L2.inp** to generate sensitivities for a standard smooth inversion. We then

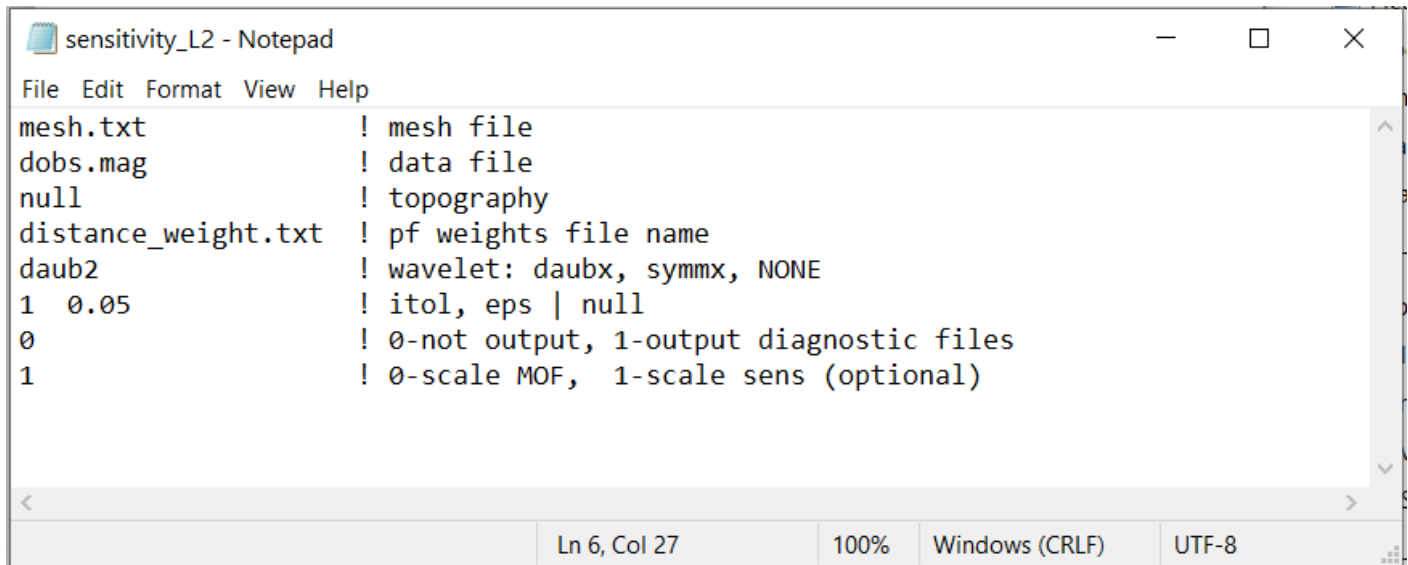
use the input file **sensitivity_sparse.inp** to generate sensitivities that can be used for sparse inversion.

Before running this example, you may want to do the following:

- Download and open the zip folder containing the entire mag3d example (if not done already)
- Learn how to *compute sensitivities and learn the format of the input files*

5.4.1 Smooth Inversion

For standard L2 inversion, you can generate the sensitivities using the input file **sensitivity_L2.inp** and supporting files in the sub-folder *sensitivity_L2*. This input file is shown below.



```

sensitivity_L2 - Notepad
File Edit Format View Help
mesh.txt          ! mesh file
dobs.mag          ! data file
null              ! topography
distance_weight.txt ! pf weights file name
daub2             ! wavelet: daubx, symmx, NONE
1 0.05            ! itol, eps | null
0                 ! 0-not output, 1-output diagnostic files
1                 ! 0-scale MOF, 1-scale sens (optional)
  
```

5.4.2 Smooth Inversion

The last line of the input file must be set to 0 to compute sensitivities for sparse inversion. In this case, the sensitivities were generated using the input file **sensitivity_sparse.inp** and supporting files in the sub-folder *sensitivity_sparse*. This input file is shown below.

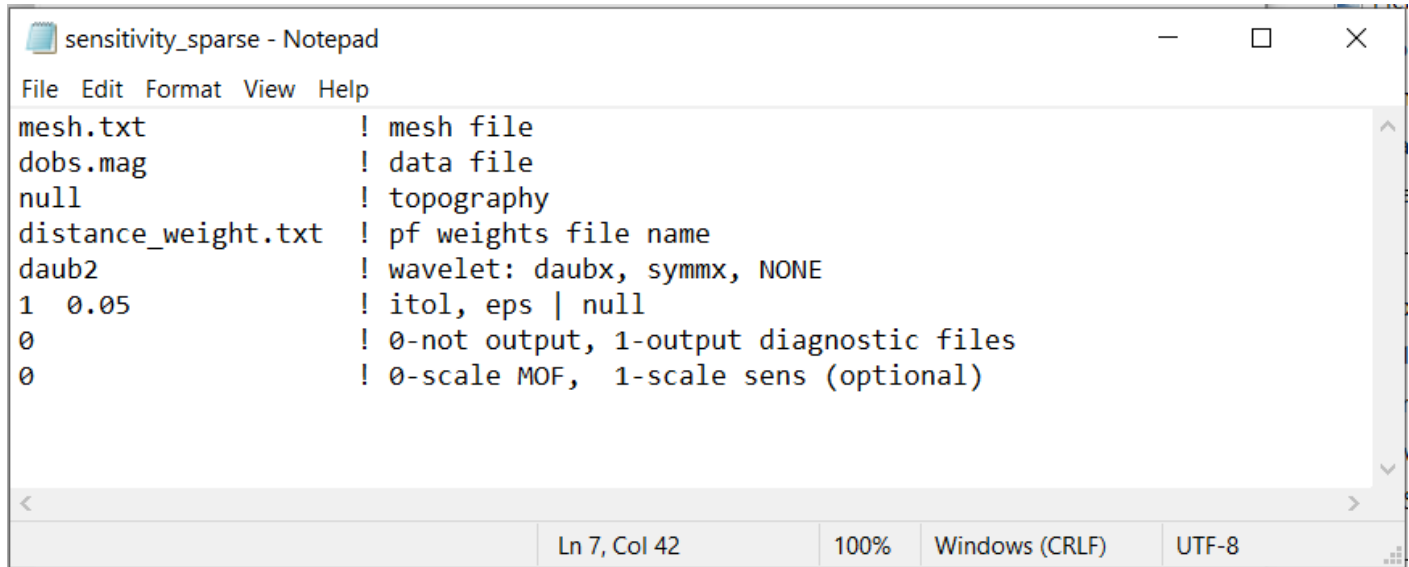
Note: This example has been developed to demonstrate functionality specific to v6.0. The example can be completed using v5.0, however some functionality may not exist in v5.0 and the format of certain input files may differ slightly.

5.5 Inversion

Here the code **maginv3d_v60.exe** invert synthetic magnetic data to recover a susceptibility model. We perform both a smooth and a sparse inversion. We use the same data that were computed in *forward modeling example*. Gaussian noise were added to these data with a standard deviation of 0.5 nT. Uncertainties of 0.5 nT were assigned to all data

Before running this example, you may want to do the following:

- Download and open the zip folder containing the entire mag3d example (if not done already)
- Learn how to *run maginv3d_v60 and learn the format of the input files*



```
sensitivity_sparse - Notepad
File Edit Format View Help
mesh.txt      ! mesh file
dobs.mag      ! data file
null          ! topography
distance_weight.txt ! pf weights file name
daub2         ! wavelet: daubx, symmx, NONE
1 0.05        ! itol, eps | null
0             ! 0-not output, 1-output diagnostic files
0             ! 0-scale MOF, 1-scale sens (optional)

Ln 7, Col 42    100%    Windows (CRLF)    UTF-8
```

Important: Since the sensitivities output by **magsen3d_v60.exe** produce a large file, we have not provided them in the zip file. You need to complete the *compute sensitivities* example to run the inversion.

5.5.1 Smooth Inversion

For standard L2 inversion, we use the input file **inv_L2.inp** and the supporting files in the sub-folder *inv_L2*. The input file is shown below.

The algorithm reaches an optimum model after 5 iterations. Least-squares gravity inversion will generally place structures at the appropriate locations but will underestimate the amplitude of density contrast.

5.5.2 Smooth Inversion

For sparse inversion, we use the input file **inv_sparse.inp** and the supporting files in the sub-folder *inv_sparse*. The input file is shown below.

The inversion was set to recover a model that is more compact. By forcing the model to be compact, we recover a structure who density contrast of much closer to the true value of 0.1 g/cc.

Mag3d v5.0

A legacy example using the Mag3d v5.0 package can be found here:

5.6 Examples

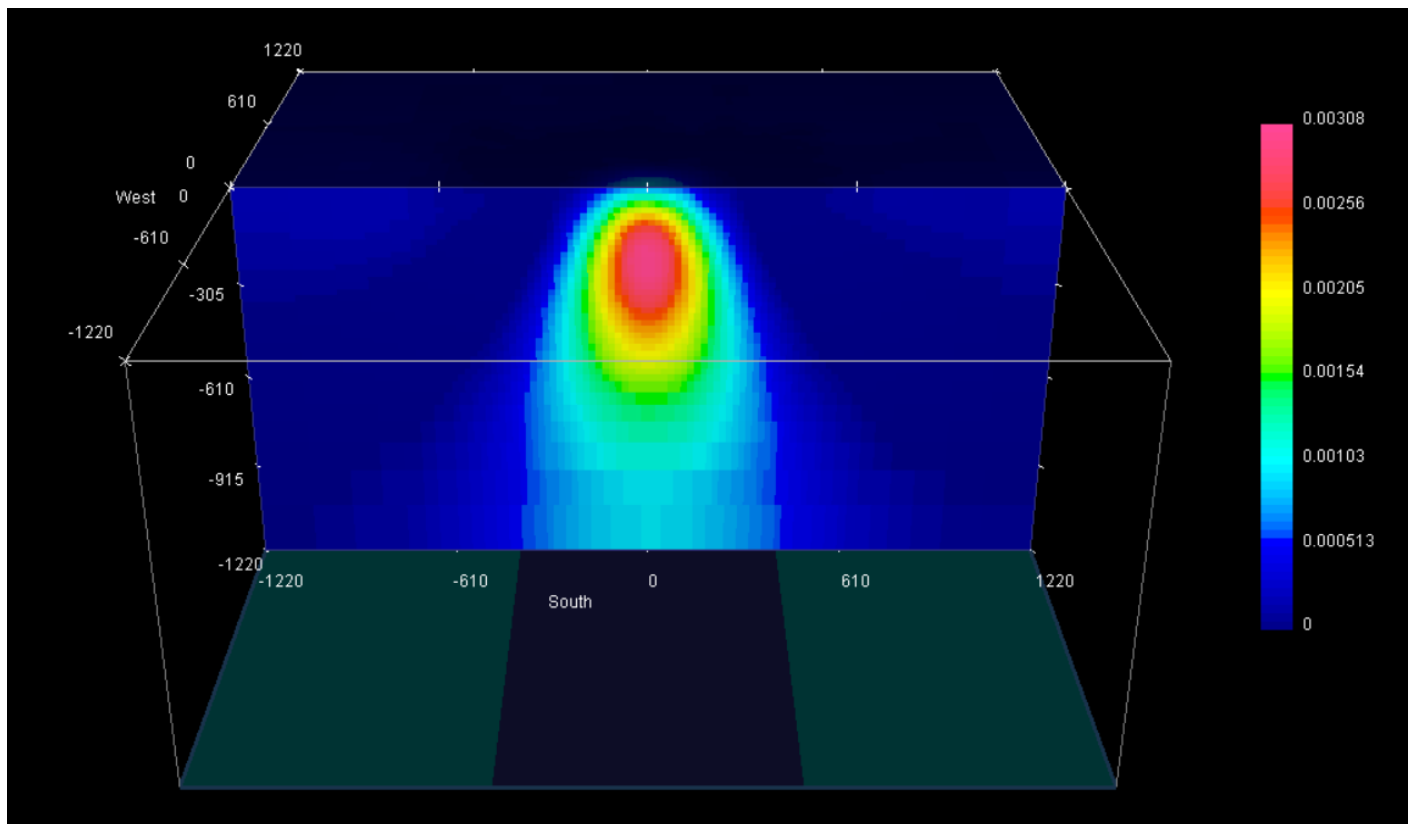
In this section, we present synthetic examples to illustrate forward modelling and inversion capabilities of MAG3D v5.0. Important functionalities of MAG3D v5.0 are parallelization, the ability to handle borehole data, an increased freedom in the design of the model objective function, a faster procedure for incorporating constraints (using projected gradients), and the ability to check the accuracy of the wavelet compression for large scale problems. The synthetic examples described below are constructed to show these features. They are based on a synthetic model, which consists


```

inv_L2 - Notepad
File Edit Format View Help
1                               ! mode: 1 | 2
1.0 0.02                       ! par, tolc
dobs.mag                       ! observations file
../sensitivity_L2/maginv3d.mtx ! mtx file
VALUE 1e-4                     ! initial model
VALUE 1e-4                     ! reference model
null                           ! active cells model | null
VALUE 0.                       ! lower bound
VALUE 10.                     ! upper bound
1e-4 1. 1. 1.                 ! alphaS alphaE alphaN alphaZ | null
SMOOTH_MOD                    ! SMOOTH_MOD | SMOOTH_MOD_DIF
null                          ! depth or distance weights | null
null                          ! ps px py pz | null
null                          ! Lq scaling [0,1], zeroS, zeroXYZ | null
null                          ! MOF derivatives | null

Ln 9, Col 48    100%    Windows (CRLF)    UTF-8

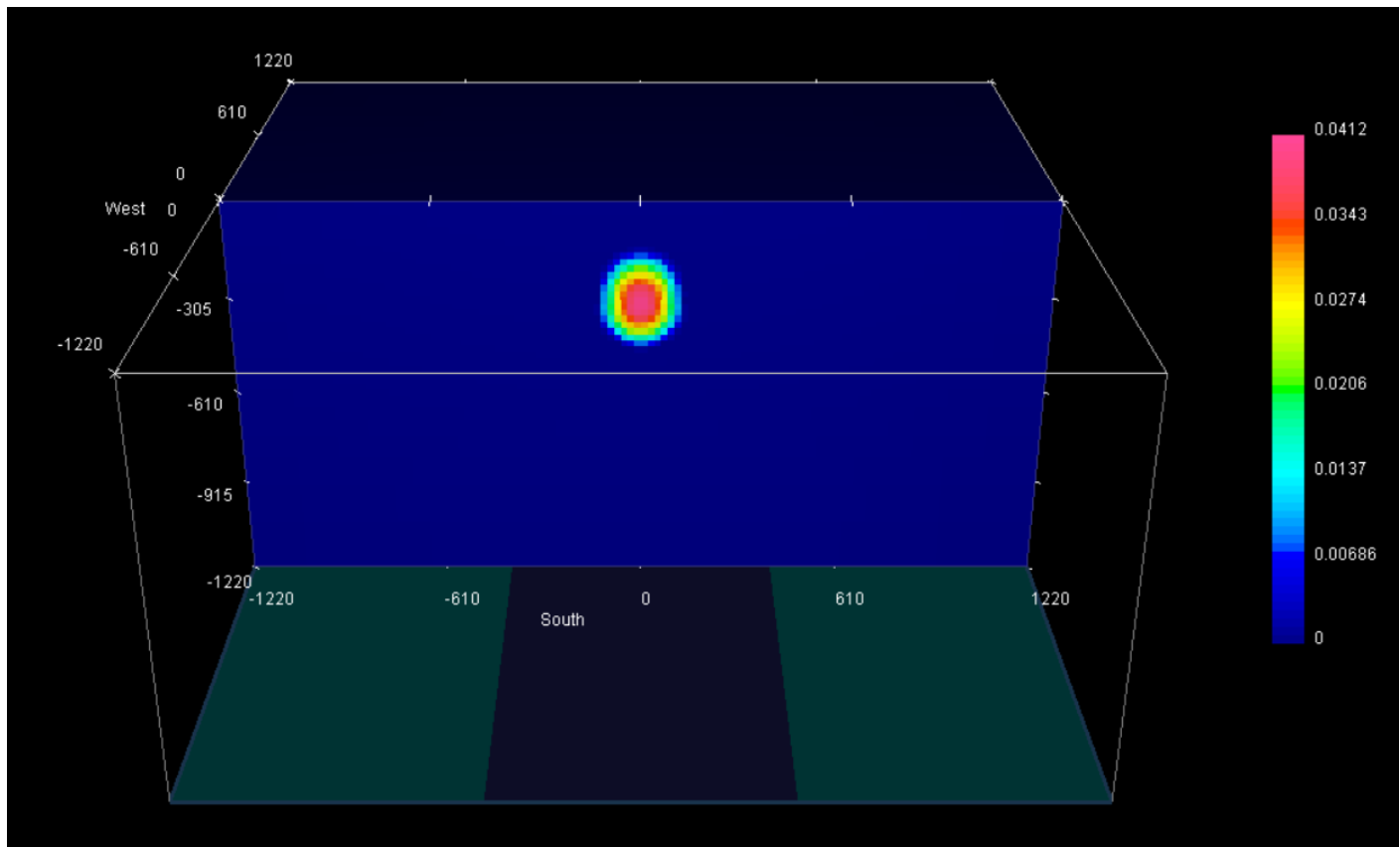
```



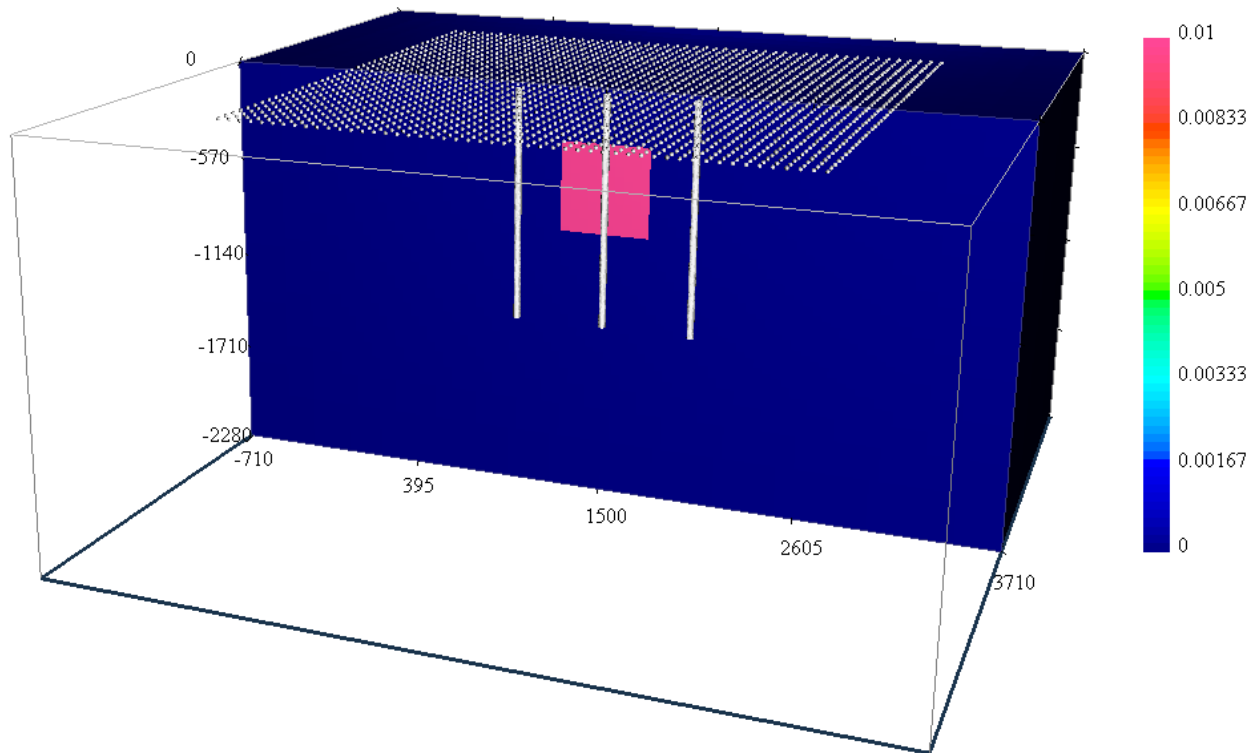
```

inv_sparse - Notepad
File Edit Format View Help
1 ! mode: 1 | 2
1.0 0.05 ! par, tolc
dobs.mag ! observations file
../sensitivity_sparse/maginv3d.mtx ! mtx file
VALUE 1e-4 ! initial model
VALUE 1e-4 ! reference model
null ! active cells model | null
VALUE 0. ! lower bound
VALUE 10. ! upper bound
1e-4 1. 1. 1. ! alphas alphaE alphaN alphaZ | null
SMOOTH_MOD ! SMOOTH_MOD | SMOOTH_MOD_DIF
null ! additional weights | null
VALUE 0 2 2 2 ! ps px py pz | null
0.5 1e-3 1e-3 ! Lq scaling [0,1], zeros, zeroXYZ | null
null ! MOF derivatives | null
Ln 12, Col 60 100% Windows (CRLF) UTF-8

```



of a 500 meter susceptible cube in a half space, placed 300 m below the flat surface (Fig. ??). Surface and multi-component borehole data are simulated. The examples can be downloaded directly from [here \(14 MB\)](#)



5.6.1 Forward modelling

The surface data set consists of 2,091 evenly gridded data 60 meters Easting by 75 meters Northing (see Fig. ??). The total-field anomaly was calculated at an inclination of 65° and declination of 25° with a field strength of 50,000 nT.

In order run in this example, the following was put into the command line:

```
magfor3d mesh.txt loc_surf.txt model.sus
```

We next simulated the borehole data. The borehole data are calculated in three vertical holes located 500 meters apart east-west (Fig. ??) with a depth interval of 20 meters for a total of 594 data. Each borehole accommodates 51 three-component measurements covering depth interval from 15 to 1,335 meters below surface. The command used to compute the simulated borehole data was:

```
magfor3d mesh.txt loc_bh.txt model.sus
```

The simulated data have been contaminated with Gaussian noise with a standard deviation of 3 nT plus 2% of the data. The noisy surface data are shown in Fig. 5.1. Figure Fig. 5.2 shows the accurate and noisy borehole data.

5.6.2 Inversion

We illustrate the inversion of the surface data and then a joint inversion of surface and borehole data.

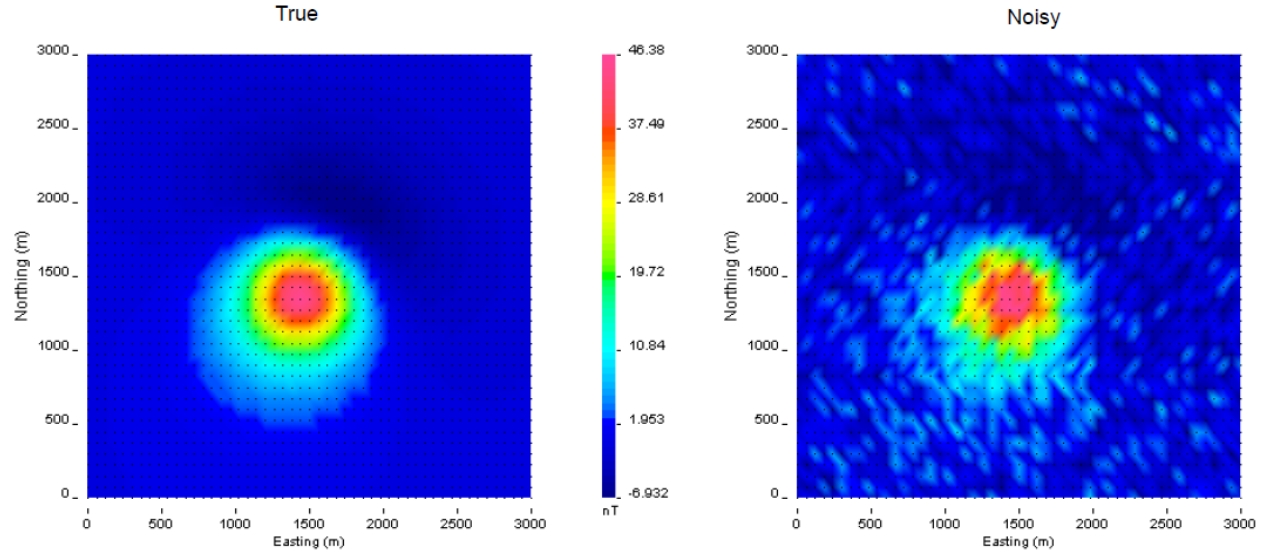


Fig. 5.1: The true data generated by a 500 m^3 block with a magnetic susceptibility of 0.01 SI (left). Gaussian noise with a floor of 3 nT and 2% of the data was added to create the observed data for the inversion (right).

Default parameters

First, the distance weighting function is calculated with *PFWEIGHT*. Distance weighting was chosen for consistency between the inversion of surface data and joint inversion of the surface and borehole data (borehole data require distance weighting). The input for the weighting function code is shown below. An example of the created log file is presented in Fig. 5.3.

Next, the sensitivity matrix is calculated with *MAGSEN3D*. The default wavelet compression parameters are used (daub2 with 5% reconstruction error ($\text{eps}=0.05$)). We also choose *not* to output diagnostic parameters, although these are discussed in the *Wavelet diagnostic output* section. With this scenario chosen, outputs and (these files will be over-written if the program has been run in the same folder more than once without re-naming them). The former file gives the user information about the sensitivity calculation such as compression ratio with the wavelet transform. The latter file will be required during the inversion process. The sensitivity will only have to be re-run if the mesh or data locations are changed. Changing the standard deviations does not require a new *.mtx* file. The log file from the sensitivity calculation is shown in Fig. 5.4. The input file given to is shown below.

Once the matrix file is created, the inversion can be run by with a general input file. The control file example is provided below. The bounds are set to so positivity is enforced. Therefore, the initial model is set slightly above zero so that the entire model is included in the first iteration. The initial model is also usually set to the reference model so the reference model is set slightly above zero.

The inversion converges in four iterations. Fig. 5.5 shows the convergence curve for the data misfit versus the iteration number. The desired misfit is approximately 2100 and is achieved within the tolerance given ($\pm 2\%$). The predicted data from the recovered model is shown on the right of Fig. 5.6 with the observed data on the right for comparison.

An example log file created by within this example set (borehole and surface data) is shown in Fig. 5.7. The file gives the input parameters and general information for every iteration such as the data misfit and iteration CPU time. A developers log (maginv3d.out) is also written (Figure [fig:developlog]). This file contains detailed information for every iteration including the beta parameter, data misfit, model norm and its components, total objective function, number of conjugate gradient iterations, and the number of truncated cells. The latter is the amount of cells that are at or beyond the bounds and are not included in the minimization with the projected gradient. In this case, it would be cells at or below zero that were set to zero.

A slice of the recovered model through the centre of the anomalous body is presented in Fig. 5.9 (top). The recovered

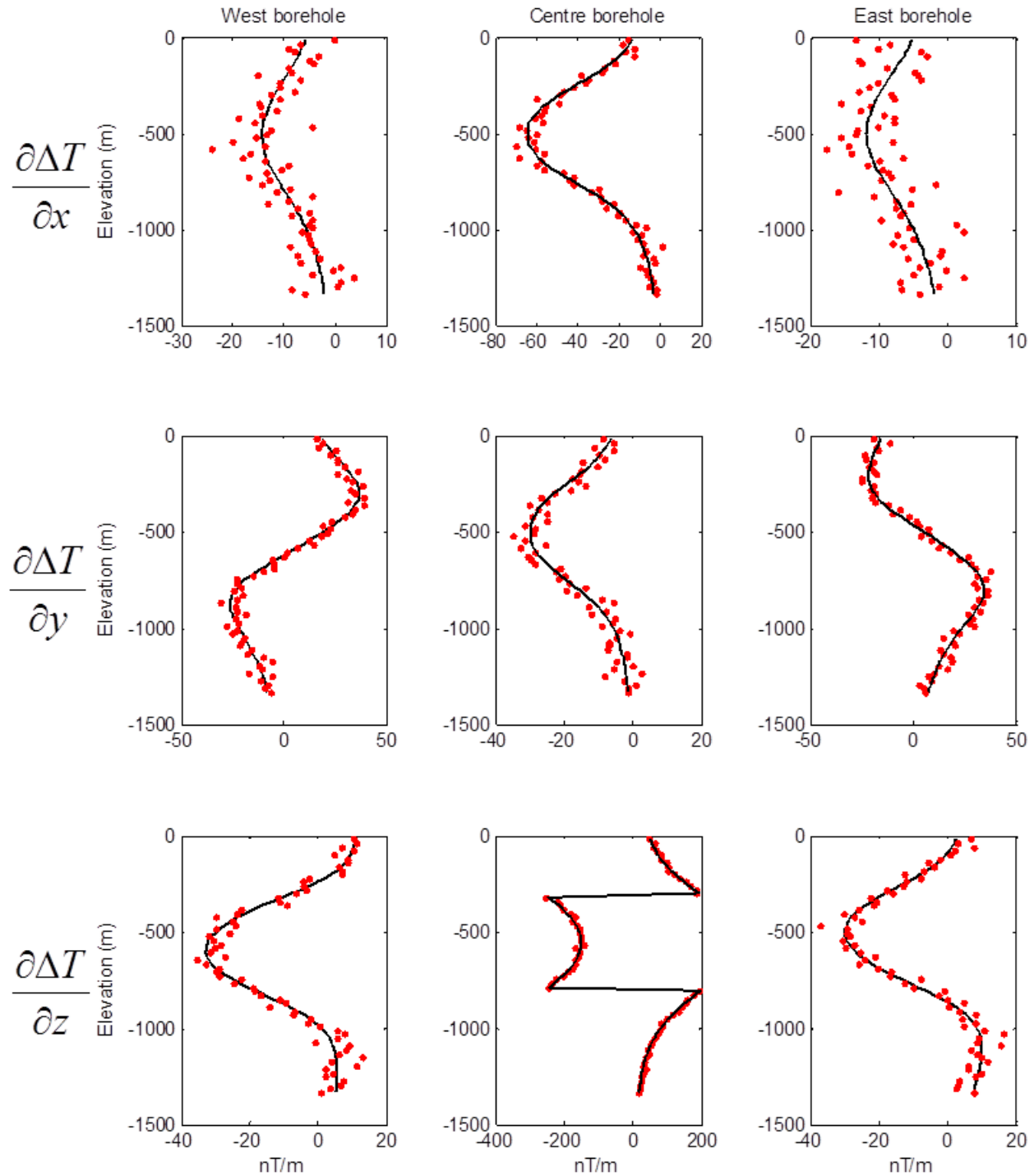


Fig. 5.2: Synthetic multi-component borehole data used in the inversions. Each column represents a single borehole and each row is the same component. The black lines are true data and red dots are the noisy data.

MAG	!	MAG or GRAV
..\mesh.txt	!	mesh file
obs_surf.mag	!	observations file
null	!	topography file
2	!	weighting type
null	!	alpha, znot

anomaly has smaller amplitude and is smoothed relative to the true model. The two boreholes that do not intersect the anomaly are then added. The data are inverted with the same parameters as previously given for the surface-only example and achieves the appropriate data misfit. The recovered model is shown in Fig. 5.9 (middle). The anomalous body is tighter and a bit more constrained with the addition of subsurface data. Finally, data from the third borehole that intersects the anomaly are added to the observed data. An interesting observation from the recovered model (Fig. 5.9 ; bottom) is the lack of magnetic susceptibility where the borehole is physically located.

There are two different types of constraints that can be used in order to recover an anomalous body near the borehole that physically intersects it. Those types are soft or hard constraints. Soft constraints are applied through the model objective function and hard constraints are provided through active and inactive cells, and bounds. The following two sections apply each one of these types of constraints, respectively, in order to alleviate this problem.

Use of soft constraints

One type of constraints that can be used to connect the body for a more realistic interpretation is soft constraints. We examine both the use of distance weighting and the reference model through the *model objective function*.

Distance weighting

Distance weighting is utilized to avoid placing susceptible cells near the observation locations where the mesh has a higher sensitivity and can bias the final solution. We therefore manually change the R_o in the *distance weighting* in the input file from the default value of 1/4 of a cell to 100 - much larger than what is needed. Since the values are then normalized, this will allow susceptible material near the borehole locations. The α value should be 3.0 due to the field decay to a cubic power. The sensitivity and inversion input files stay the same. The weighting input file for this example is

The inversion is run with all three boreholes and surface data. A slice of the recovered model is shown in Figure [fig:all_ro]. The recovered model has a single anomaly as desired. The anomaly is near the true susceptibility (0.01 SI) and has a block-like shape. A by-product of using this weighting is that the algorithm is able to place susceptibility not only near the borehole locations, but also near surface observations. To improve upon the results, we examine the use of the reference model with this weighting in order to centralize the anomalous susceptibility.

Reference model

As previously discussed in the *inversion methodology*, the reference model can either be incorporated into the spatial derivatives or only the smallest model component of the model objective function. We use the *MAGSEN3D* input file from distance weighting and examine the differences in the recovered model with the addition of the reference model.

The centre borehole intersects the anomaly so we assume that we know the true model at the location of those subsurface observations. The reference model is then designed so that everywhere else it promotes a zero model. A cross section of the reference model is shown in Fig. 5.11 (top). Only the cells that the borehole intersects the anomaly are given as susceptibilities over zero.

```

pfweight.log - Notepad
File Edit Format View Help

Parallelized with openMP. # of threads: 12
MAG3D - Version 5.0 : PFWEIGHT

Developed by University of British Columbia
Geophysical Inversion Facility (UBC-GIF)

(C) Copyright 1992 - 2013, UBC-GIF,
Department of Earth and Ocean Sciences, UBC
http://gif.eos.ubc.ca/

This program is licensed to:

For internal use within UBC-GIF.

Reading input file: pfweight.inp

-----

MAG          ! data type (MAG | GRAV)
..\mesh.txt  ! mesh file
obs_surf.mag ! mag file
null        ! topography
2           ! distance weighting
null        ! beta,znot

-----

PFWEIGHT started on: 4/29/2013 10:30:43
with the following parameters:

    Input file: pfweight.inp
    Observations file: obs_surf.mag
    Mesh file: ..\mesh.txt
    Topography file: null
    distance weighting
    exp, R0: null

    # of cells: 72 x 72 x 66
    total # of cells: 342144
    # of cells below topo: 342144

    # of surface data: 2091
    # of borehole data: 0

    default distance weighting parameters:
    pwr = 3.00
    rnot = 6.25

    TOTAL cpu time: 0:00:19.45

PFWEIGHT ended on: 4/29/2013 10:31:03

```

```
..\mesh.txt      ! mesh file
obs_surf.mag     ! observations file
null            ! topography file
distance_weight.txt ! weighting file
null            ! wavelet type
null            ! wavelet parameters
0               ! do NOT output diagnostics
```

The input file for the inversion with the reference model throughout model objective function is shown below. The initial model is the same as the reference model and the choice `SMOOTH_MOD_DIF` is invoked in order to place the reference model in the spatial derivatives.

The recovered model is found in [Fig. 5.11](#) (middle). There is a single anomaly with the maximum amplitude where the non-zero portion of the reference model influenced the solution. The surrounding part of the body goes to zero to try to minimize the difference spatially leaving a strip where the non-zero part of the reference model is located. In this light, the affects of penalizing the derivatives with the reference model included become apparent.

Next, the input file for the inversion is changed so that the option `SMOOTH_MOD` is used in order to place the reference model only in the smallest component of the model objective function. A cross section of the recovered model with this option is presented in [Fig. 5.11](#) (bottom). This time the recovered anomaly is much more homogeneous and is closer to the true model throughout the body. The solution is similar to just the distance weighting, though it recovers higher susceptibilities and does not decrease in amplitude as much within the anomalous volume.

Use of hard constraints

The last section discussed the flexibility of the model objective function to influence the result of *MAGINV3D*. This section examines the use of hard constraints that strictly enforce a range of values rather than promote the values mathematically. We first incorporate bound constraints and then set key cells to be inactive within the inversion.

Bounds cells

To be able to appropriately bound the model, we examine the susceptibility given by the borehole information. The bound model file is two columns and requires a lower and upper bound, respectively. For the lower bound, we set the model to zero everywhere but the intersection of the anomaly with the centre borehole. The true model is observed here, so we set the bounds in this region to 0.0099 - just below the 0.01 of the anomalous body ([Fig. 5.12](#) ; top). The upper bounds are 1 (i.e. positivity only) everywhere but in the locations of the zero susceptibility found in the boreholes. This model can be found in [Fig. 5.12](#) (bottom). These two models create the bounds file. We use the same reference model from the soft constraints section. The reference model is only incorporated in the smallest model component of the model objective function. The input file for the inversion with bounds is

and a cross section of the recovered model is found in [Fig. 5.12](#) (bottom). The bounds force the model to the correct 0.01 SI values where the centre borehole intersects the anomalous body, to zero where the boreholes do not intersect any susceptibility, and allows the rest of the model to change as necessary. This results in large values in the centre of the anomaly with smoothly decaying amplitudes towards the outsides of the body. The shape is still correctly recovered.

Active/inactive cells

An added functionality of *MAGINV3D* is the ability to set cells to a prescribed value and not to incorporate them directly into the inversion. For this example, the model cells in the boreholes are set to inactive. This means they will


```

magsen3d.log - Notepad
File Edit Format View Help
|
Parallelized with openMP. # of threads: 12
MAG3D - Version 5.0 : MAGSEN3D
Developed by University of British Columbia
Geophysical Inversion Facility (UBC-GIF)
(C) Copyright 1992 - 2013, UBC-GIF,
Department of Earth and Ocean Sciences, UBC
http://gif.eos.ubc.ca/
This program is licensed to:
  For internal use within UBC-GIF.
Reading input file: magsen3d.inp
-----
..\mesh.txt           ! mesh file
obs_surf.mag          ! data file
null                  ! topo.dat
distance_weight.txt    ! weights
null                  ! wavelet (Daub2)
null                  ! wavelet accuracy
0                     ! diagnostics
-----
MAGSEN3D started on:  4/29/2013  10:36:18
with the following parameters:
      Input file: magsen3d.inp
      Observations file: obs_surf.mag
      Mesh file: ..\mesh.txt
      Topography file: null
      wavelet compression: daub2
      wavelet threshold: default
      # of cells: 72 x 72 x 66
      total # of cells: 342144
      # of cells below topo: 342144
      # of surface data: 2091
      # of borehole data: 0
      weighting was read from file: distance_weight.txt
      init cpu time: 0:00:00.31
      Sensitivity of surface data:
      At the relative accuracy of: 5.0000E-02
      Estimated relative threshold: 4.6158E-03
      Predicted compression ratio: 1.4498E+02
      Number of non-zeros: 5967581
      Compression ratio: 1.1222E+02

```

```

1          ! mode (1, 2, or 3)
1.0 0.02   ! beta, tolerance
obs_surf.mag ! data file
maginv3d.mtx ! mtx file
VALUE 1.e-2 ! initial model
VALUE 0.0   ! reference model
null       ! active cells
VALUE 0.0   ! lower bounds
VALUE 0.5   ! upper bounds
null       ! Length scales
SMOOTH_MOD ! Reference model only in the smallest component
null       ! 3D weighting

```

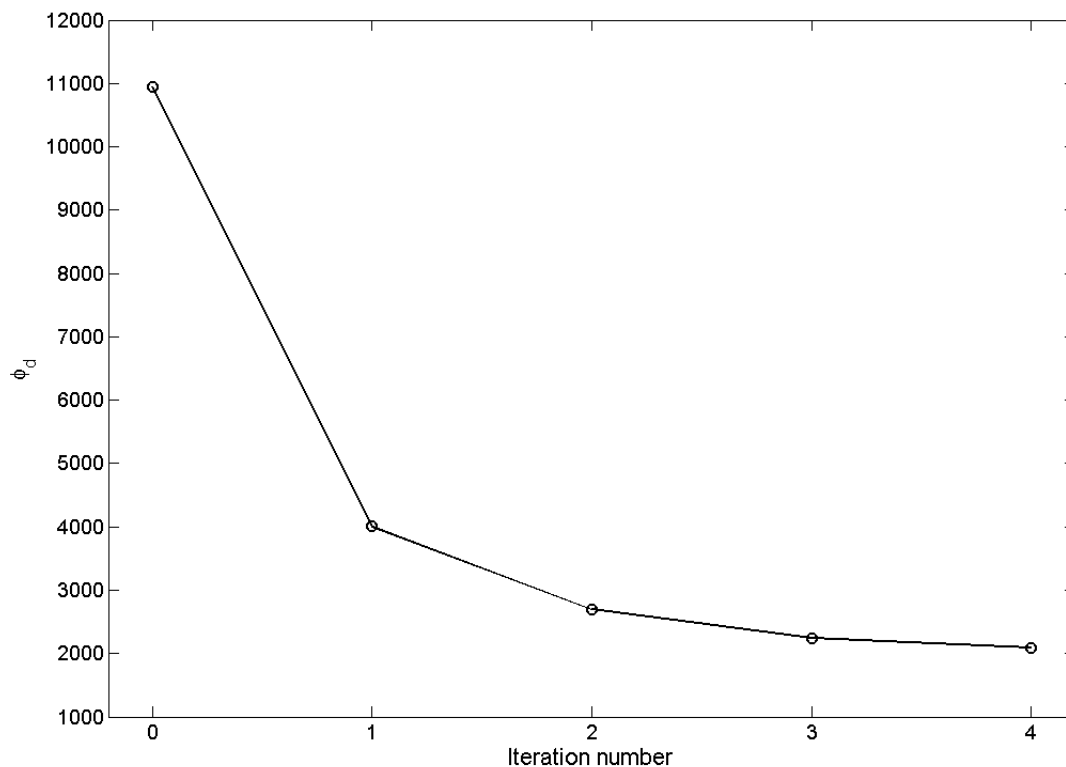


Fig. 5.5: The convergence curve for for the inversion of surface data. The 0th iteration is the initial misfit. The target misfit is approximately 2,100 where the inversion stops.

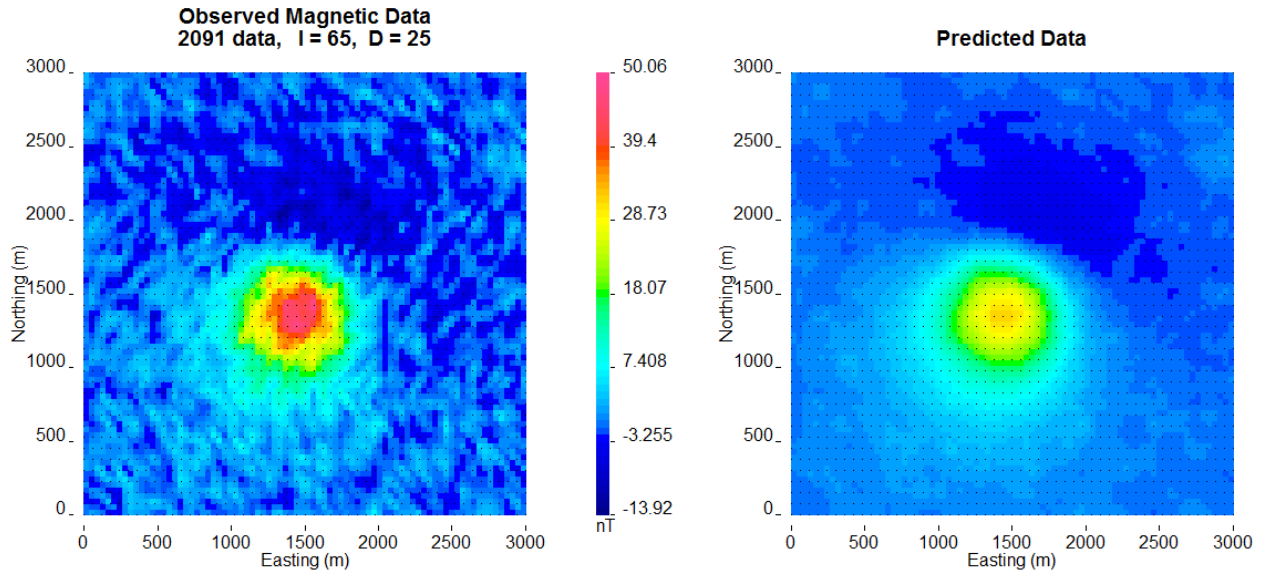


Fig. 5.6: The observed surface data (left) and the surface data created from the recovered model (right). The data are on the same colour scale.

stay the value given in the initial model and will not be part of the model objective function (they will contribute to the produced data of the solution). For this example, we set the active cells with values of 1 near the boreholes where the inversion will solve for susceptibility. The cells intersecting the boreholes where susceptibility is known is set to -1 in order to influence the model objective function, yet set the cell values. The cells outside the region of interest and that we know have no anomalous susceptibility are set to 0 (also inactive) and are not included within the inversion. Fig. 5.13 (top) is a cross section of the active cell model. The reference model determines the cell values within the inactive region so the file `reference.sus` is used. An initial model using the reference model is also set. The inversion input file for this example is

The recovered model is shown in Fig. 5.13 (bottom). The centre of the anomaly has the expected value of 0.01 (it was not part of the inversion) and the surrounding susceptibility expands to the region of the true anomalous body continuously due to keeping the reference model in the smallest model component of the model objective function. Active cells can improve the inversion when prior information is available.

5.6.3 Wavelet diagnostic tests

In this section, we discuss two approaches to try to understand the influence of the wavelet compression on the recovered model. The diagnostic test output from is first examined. Then, we show how to perform similar experiments through the combination of `ref:magfor3d` and `MAGPRE3D`.

Running magsen3d diagnostic test tool

In order to run the diagnostic test via `MAGSEN3D`, a 1 is given on the bottom line of the input file. The weighting code is run prior to the sensitivity and the sensitivity matrix output can be used (as if the test was not run) in the inversion. It should be noted that the testing can require up to twice the CPU time compared to running the sensitivity matrix calculation alone. Once the diagnostic testing begins, the user may decide to stop the code. In that case, the testing files are not output yet the matrix file has been written and the inversion process can proceed. An example input file for the sensitivity calculation with testing is

The standard outputs of running are the sensitivity matrix file (`maginv3d.mtx`), the average sensitivity for each cell (`sensitivity.txt`), and the log file (`magsen3d.log`). The average sensitivity for each cell is a model

```

maginv3d.log - Notepad
File Edit Format View Help
Parallelized with openMP. # of threads: 12

MAG3D - Version 5.0 : MAGINV3D

Developed by University of British Columbia
Geophysical Inversion Facility (UBC-GIF)

(C) Copyright 1992 - 2013, UBC-GIF,
Department of Earth and Ocean Sciences, UBC
http://gif.eos.ubc.ca/

This program is licensed to:

For internal use within UBC-GIF.

Reading input file: maginv3d.inp

-----
1          ! mode (1, 2, or 3)
1.0 0.2    ! amu, tolC
obs_surf.mag ! data file
maginv3d.mtx ! mtx file
VALUE 1.e-2 ! initial model
VALUE 0.0   ! reference model
null        ! active cells
VALUE 0.0   ! lower bounds
VALUE 1.0   ! upper bounds
null        ! length scales
SMOOTH_MOD  ! Ref model only is ws
null        ! 3D weighting

-----

MAGINV3D started on: 4/29/2013 10:40:11

Running with the following parameters:

      Input file: maginv3d.inp
      mode: 1 (fixed target misfit)
      chifact: 1.000
      observations file: obs_surf.mag
      Sensitivity file: maginv3d.mtx
      Initial Model: VALUE 1.e-2
      Reference Model: VALUE 0.0
      mref: [NOT] in spatial derivatives
      Active cell model: null
      3D weighting file: null

      # of data: 2091

      # of cells: 72 x 72 x 66
      total # of cells: 342144
      # of cells below topo: 342144

reference model was set to: 0.0000E+00
initial model was set to: 1.0000E-02
lower bounds was set to: 0.0000E+00
upper bounds was set to: 1.0000E+00

```

```

! MAGINV3D started on: 4/29/2013 10:40:11
! 1 1.0000 !! mode, chifact
! 3.3030E+00 !! initial norm S
! 1.7967E-01 !! initial norm E
! 2.0506E-01 !! initial norm N
! 8.0926E+00 !! initial norm Z
! 1.1780E+01 !! initial norm total
! 4.4996E+06 !! initial misfit
! 2.0910E+03 !! target data misfit
! 1.000E-04 1.000 1.000 !! alpha values

! iter    beta    data misfit    norm S    norm E    norm N    norm Z    model norm    total objective    CG iterations    truncated cells
! 1 9.127192E+06 3.946859E+03 1.767681E-04 1.284167E-05 1.628220E-05 8.500599E-06 2.143926E-04 5.903661E+03 33 227886
! 2 4.563596E+06 2.705649E+03 3.074865E-04 2.973957E-05 3.698086E-05 1.373707E-05 3.879440E-04 4.476069E+03 30 233033
! 3 3.350528E+06 2.245358E+03 3.721979E-04 4.383810E-05 5.386696E-05 2.192646E-05 4.918295E-04 3.893246E+03 29 225643
! 4 2.977555E+06 2.096862E+03 3.990242E-04 5.008203E-05 6.099598E-05 2.475964E-05 5.348618E-04 3.689442E+03 26 213282

! MAGINV3D ended on: 4/29/2013 10:41:03

```

Fig. 5.8: The developer's log created by *MAGINV3D* (*maginv3d.out*). The top portion shows the start time and date and the details of the inversion at each iteration: beta, data misfit, model norm in each direction, total objective function, CG iterations, and the number of truncated cells within the projected gradient routine. The ending data and time is also written to file to be able to match with the simplified log file.

file and can be viewed in *meshTools3D*. The average sensitivity is calculated from the *full, non-compressed* sensitivity. Running the diagnostic test performs this calculation on the *compressed* sensitivity and outputs the file *sensitivity_compressed.txt*. The true compression error is also given in the log file with this setting to be able to compare to the given compression error tolerance (e.g. 0.05).

Examining how the two average sensitivity models differ can give insight on how well the wavelet compression has performed. The general shape should be the same, but large jumps in cell size can create large differences, which will be observed with the two outputs. Fig. 5.14 (top) shows a cross-section of the uncompressed sensitivity average for the block example given in this manual. The same cross-section for the compressed average sensitivity for a 5% reconstruction error is presented in Fig. 5.14 (middle). In general, the compression shows good accuracy. The difference between the two models is given in Fig. 5.14 (bottom) for reference. All of the pictures are shown in log scale.

The data from the compressed and uncompressed sensitivity given a constant model of 0.01 is also written, aptly named *data_compressed.txt* and *data_uncompressed.txt* respectively. This also gives insight to the differences in column-based integration of the compressed and uncompressed sensitivity matrix. However, the model output is much more intuitive.

Recovered model-based diagnostic test

Users of the package often are curious how the wavelet transform is affecting the predicted data. Although the diagnostic test does this calculation on a constant model of 0.1, this test is actually easy to perform once the inversion code has a solution. The *maginv3d_xxx.pre* is the predicted data for the compressed sensitivity and can also be calculated with the code given a file and a recovered model. To obtain the predicted data for an uncompressed sensitivity matrix, run *MAGFOR3D* on the recovered model, *maginv3d_xxx.sus*. The difference between the generated data sets will show how the wavelet compression is affecting the final data. Large discrepancies in the data may suggest the use of a smaller reconstruction error given on the *eps* line of the sensitivity input file. An example is shown using the surface-only data set. The two data sets for the uncompressed, compressed sensitivity matrix, and their difference is respectively shown in Fig. 5.15. The maximum difference between the two data sets less than is 1 nT.

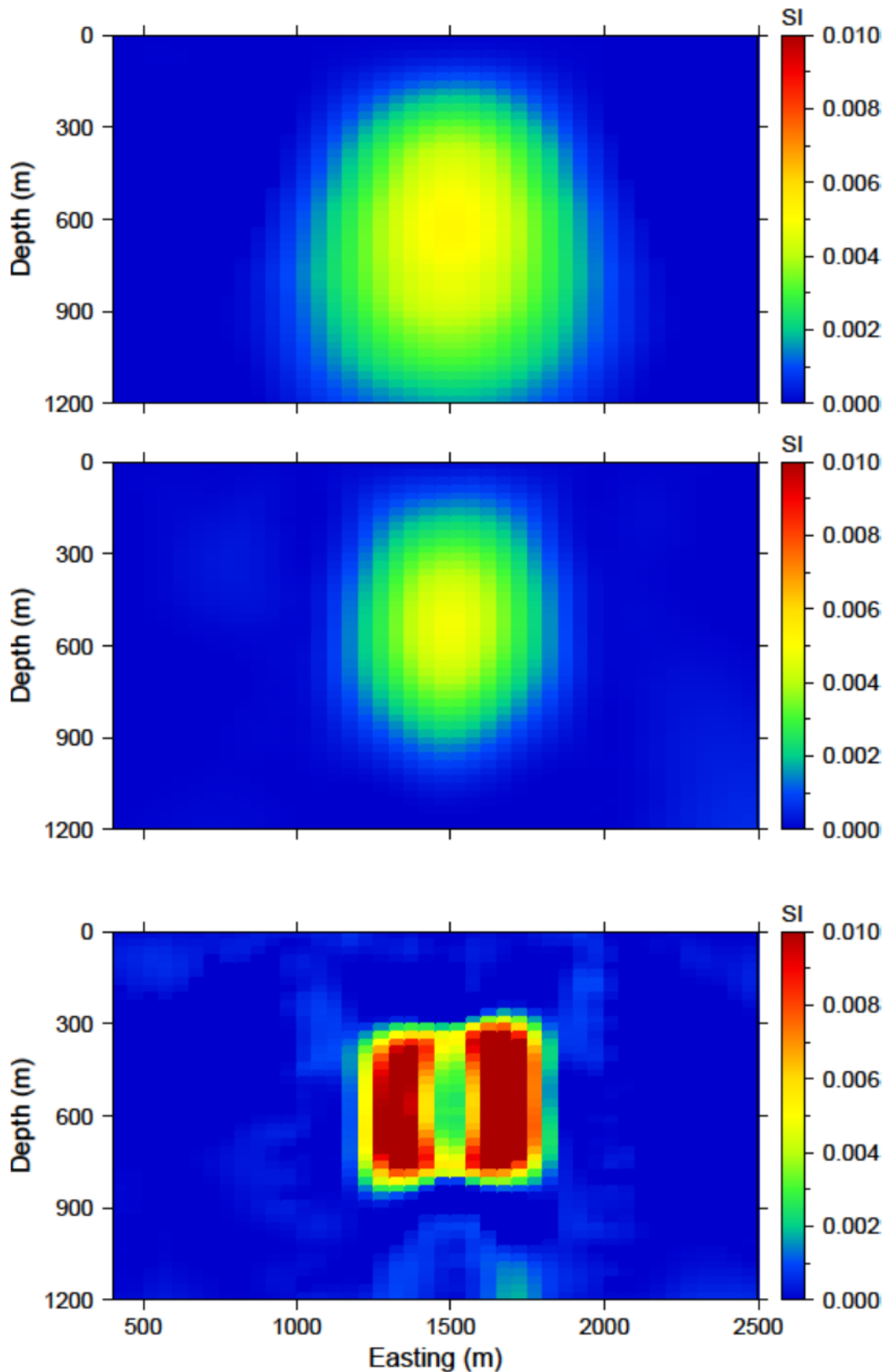


Fig. 5.9: Cross sections of the recovered models from (top) surface data only, (middle) surface data and data from the east and west boreholes, and (bottom) surface data and data from all three boreholes. The addition of borehole data aids by increasing the amplitude of the recovered anomaly and its compactness. The middle of the anomaly has higher susceptibility when the borehole that intersects the anomaly is used. Therefore, the recovered model has even higher susceptibility enabling the solution to reproduce the data.

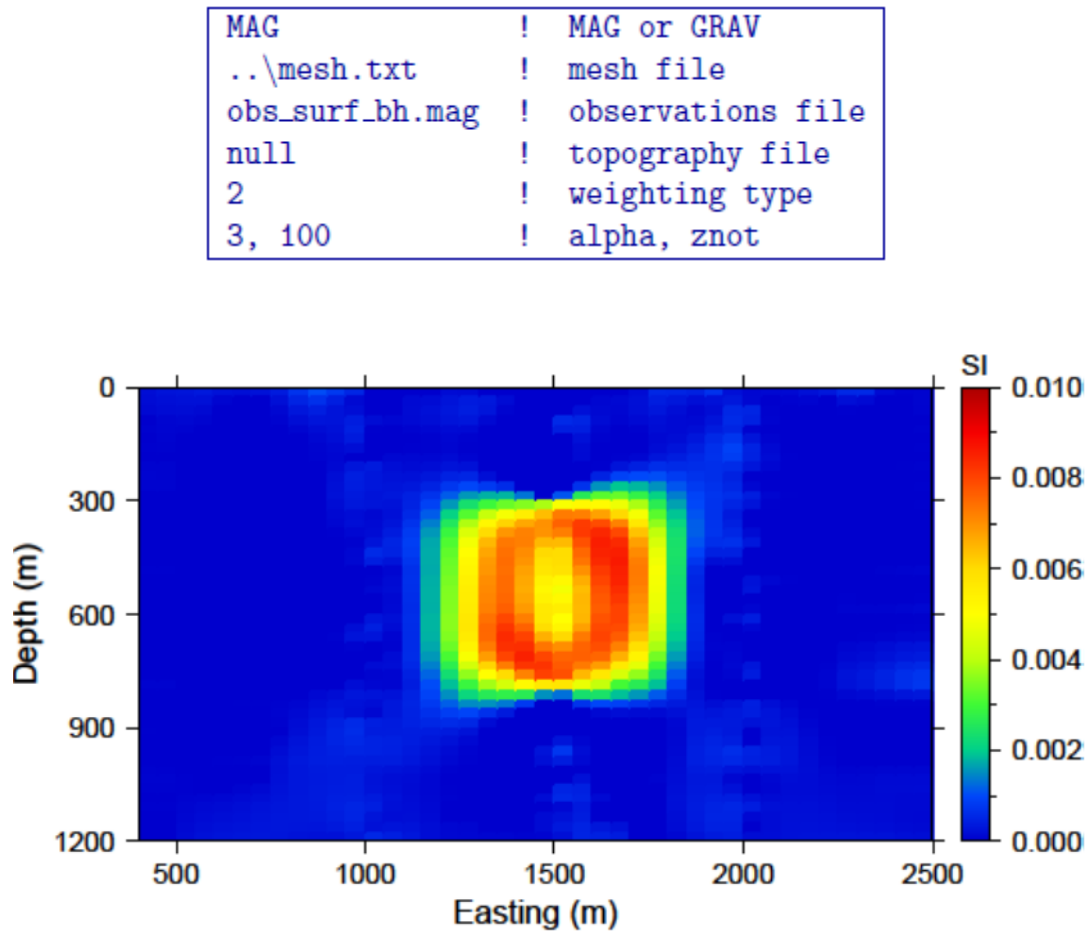


Fig. 5.10: The recovered model after increasing the distance weighting function during the sensitivity calculation. The R_o was increased to 100. The anomaly is near the true susceptibility but lacks continuity throughout the anomalous volume.

1	!	mode (1, 2, or 3)
1.0 0.02	!	beta, tolerance
..\obs_surf_bh.mag	!	data file
maginv3d.mtx	!	mtx file
..\reference.sus	!	initial model
..\reference.sus	!	reference model
null	!	active cells
VALUE 0.0	!	lower bounds
VALUE 0.5	!	upper bounds
null	!	length scales
SMOOTH_MOD_DIF	!	reference model in all components
null	!	3D weighting

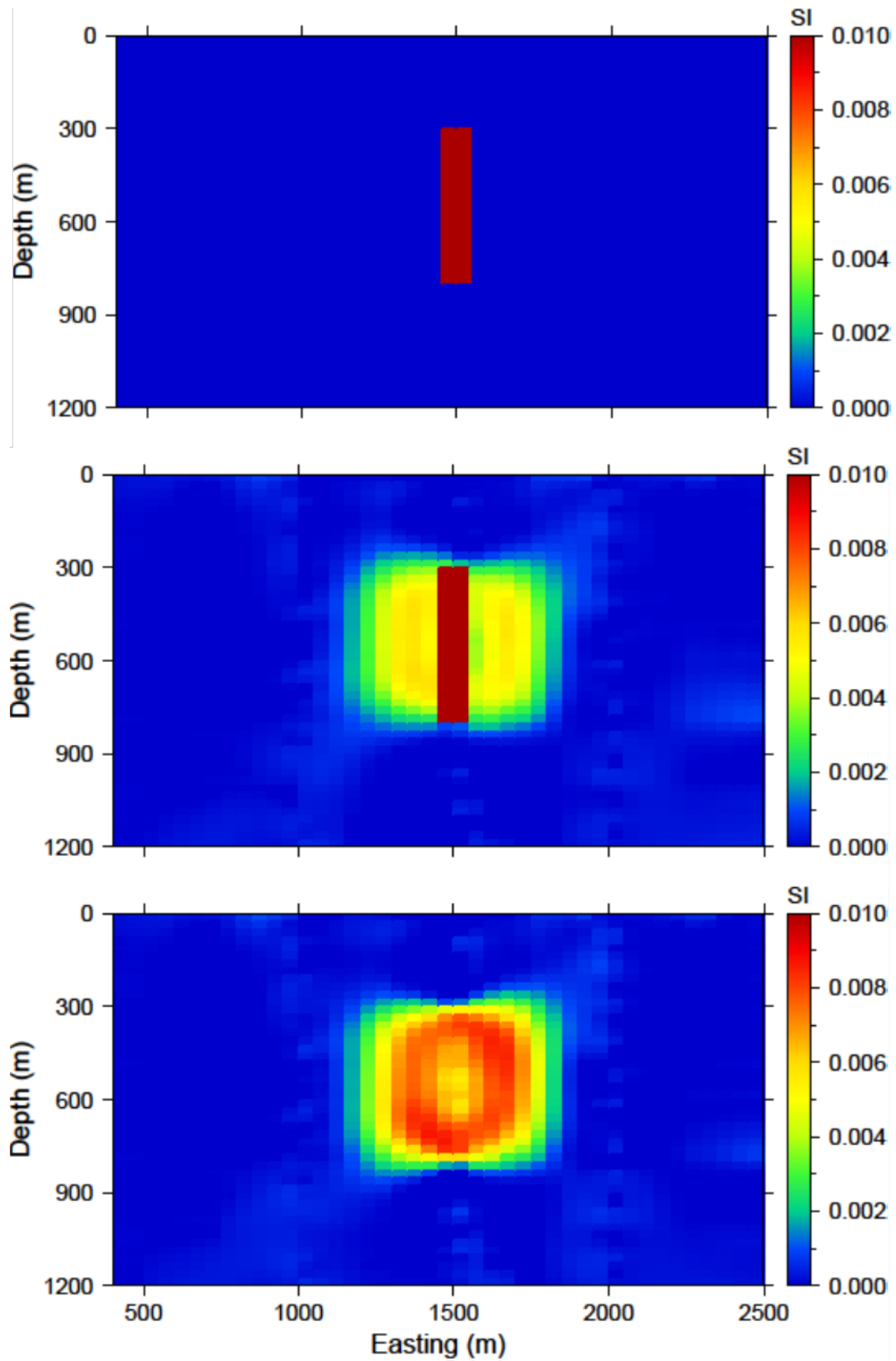


Fig. 5.11: (top) The reference model used from prior information given in the boreholes. The reference model can be utilized (middle) throughout all derivatives of the model objective function or (bottom) just in the smallest model component.

1	!	mode (1, 2, or 3)
1.0 0.02	!	beta, tolerance
obs_surf_bh.mag	!	data file
maginv3d.mtx	!	mtx file
..\reference.sus	!	initial model
..\reference.sus	!	reference model
null	!	active cells
bounds_lower.sus	!	lower bound
bounds_upper.sus	!	upper bound
null	!	Length scales
SMOOTH_MOD	!	Reference model in Ws
null	!	3D weighting

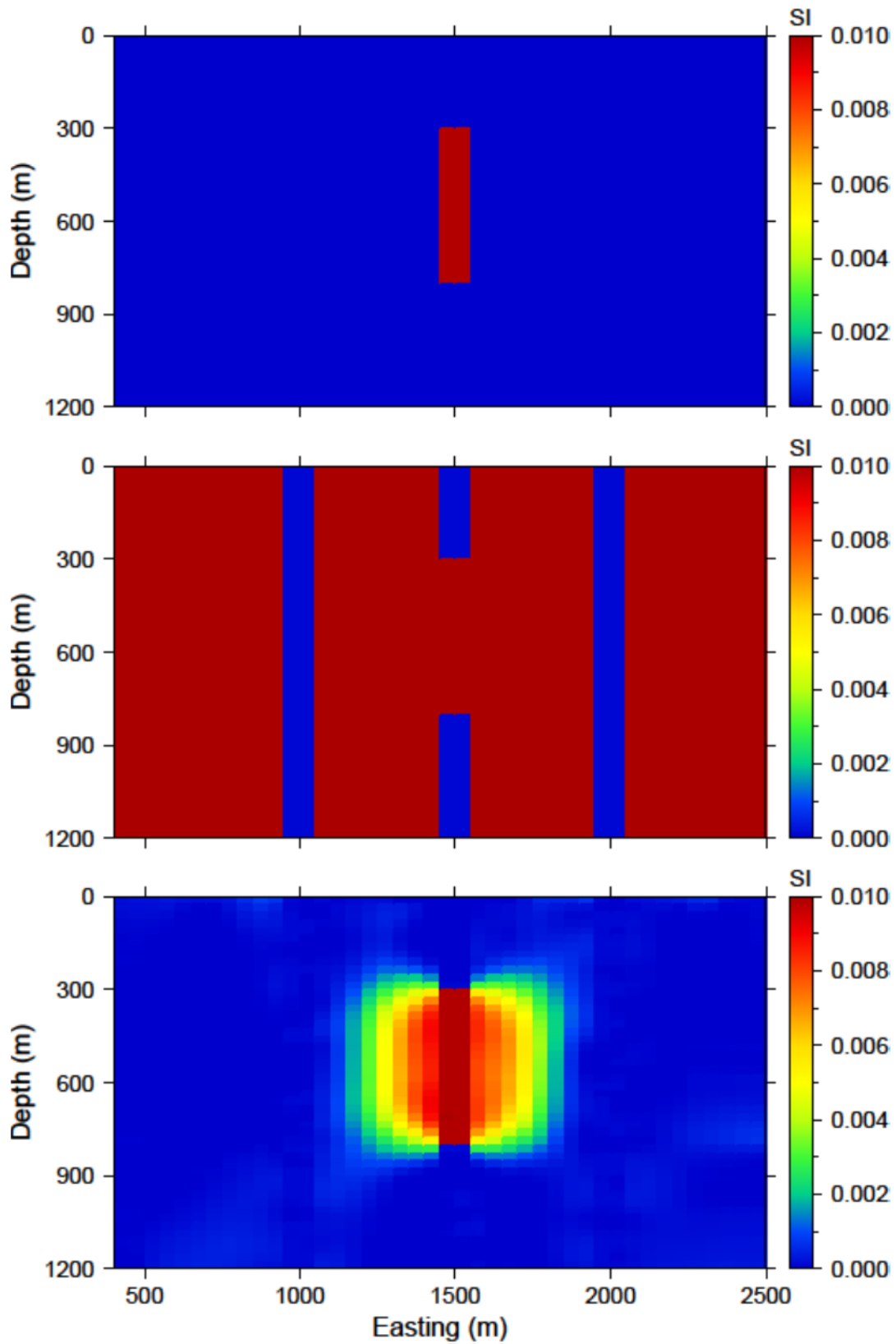


Fig. 5.12: (top) The lower bounds are zero everywhere but the intersecting section of the centre borehole. (middle) The upper bounds are 0.0001 where no susceptibility was found in the boreholes, 0.0101 in the centre borehole where the anomaly is, and 1 everywhere else in the model effectively enforcing only positivity. (c) The recovered model with bounds and an initial model.

1	!	mode (1, 2, or 3)
1.0 0.02	!	beta, tolerance
..\obs_surf_bh.mag	!	data file
maginv3d.mtx	!	mtx file
..\reference.sus	!	initial model
..\reference.sus	!	reference model
active.txt	!	active cells
VALUE 0.0	!	lower bound
VALUE 0.15	!	upper bound
null	!	length scales
SMOOTH_MOD	!	reference model in Ws
null	!	3D weighting

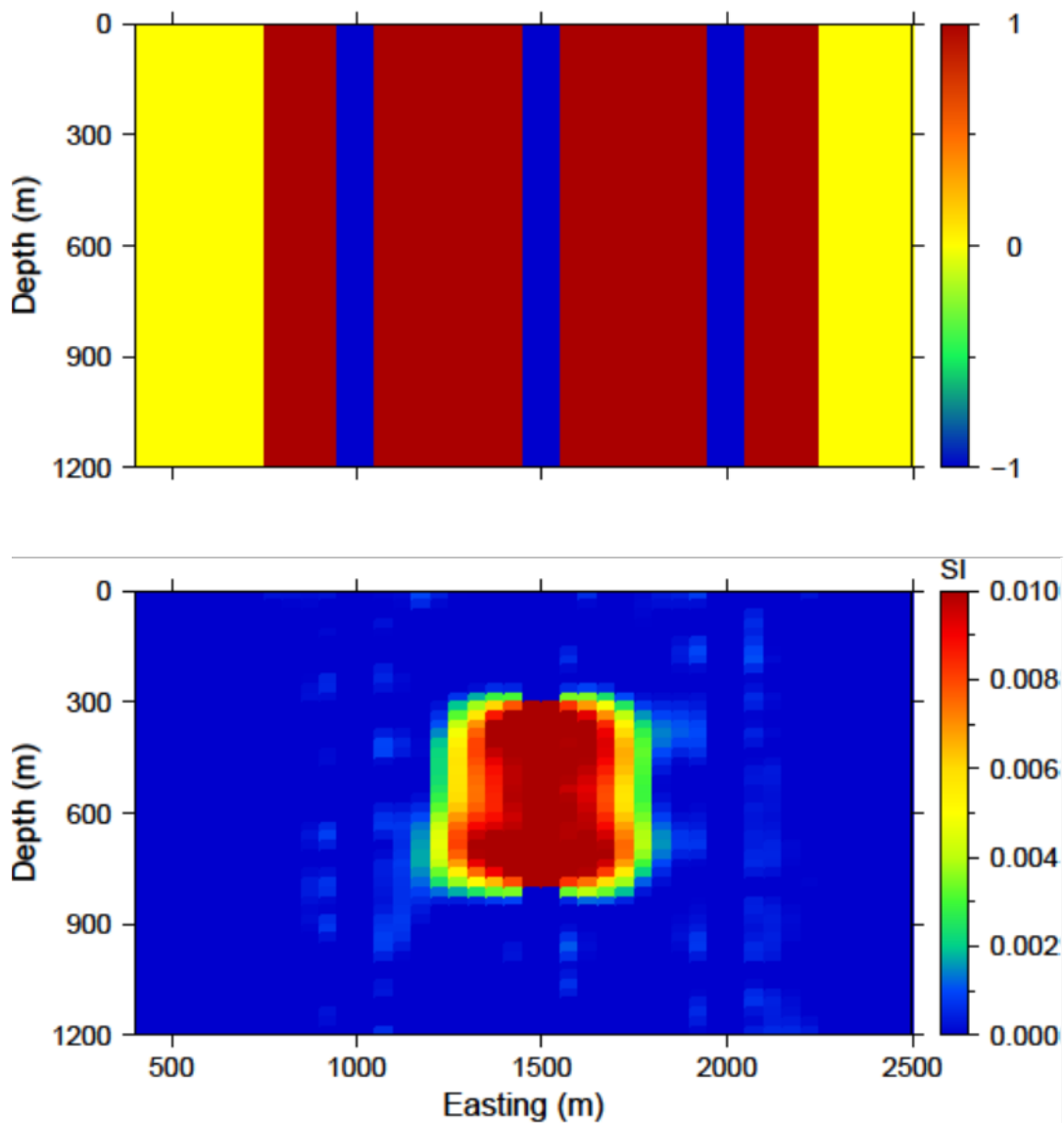


Fig. 5.13: (top) The inactive (-1 and 0) and active (1) cells that are incorporated into the inversion. The reference model sets the values of the inactive cells. The inactive cells set to -1 influence the model objective function. (bottom) A cross section of the recovered model given the inactive cells with the true susceptibility values.

```

..\mesh.txt           ! mesh file
..\obs_surf_bh.mag    ! observations file
null                  ! topography file
distance.weight.txt   ! weighting file
null                  ! wavelet type
null                  ! wavelet parameters
1                     ! output diagnostics

```

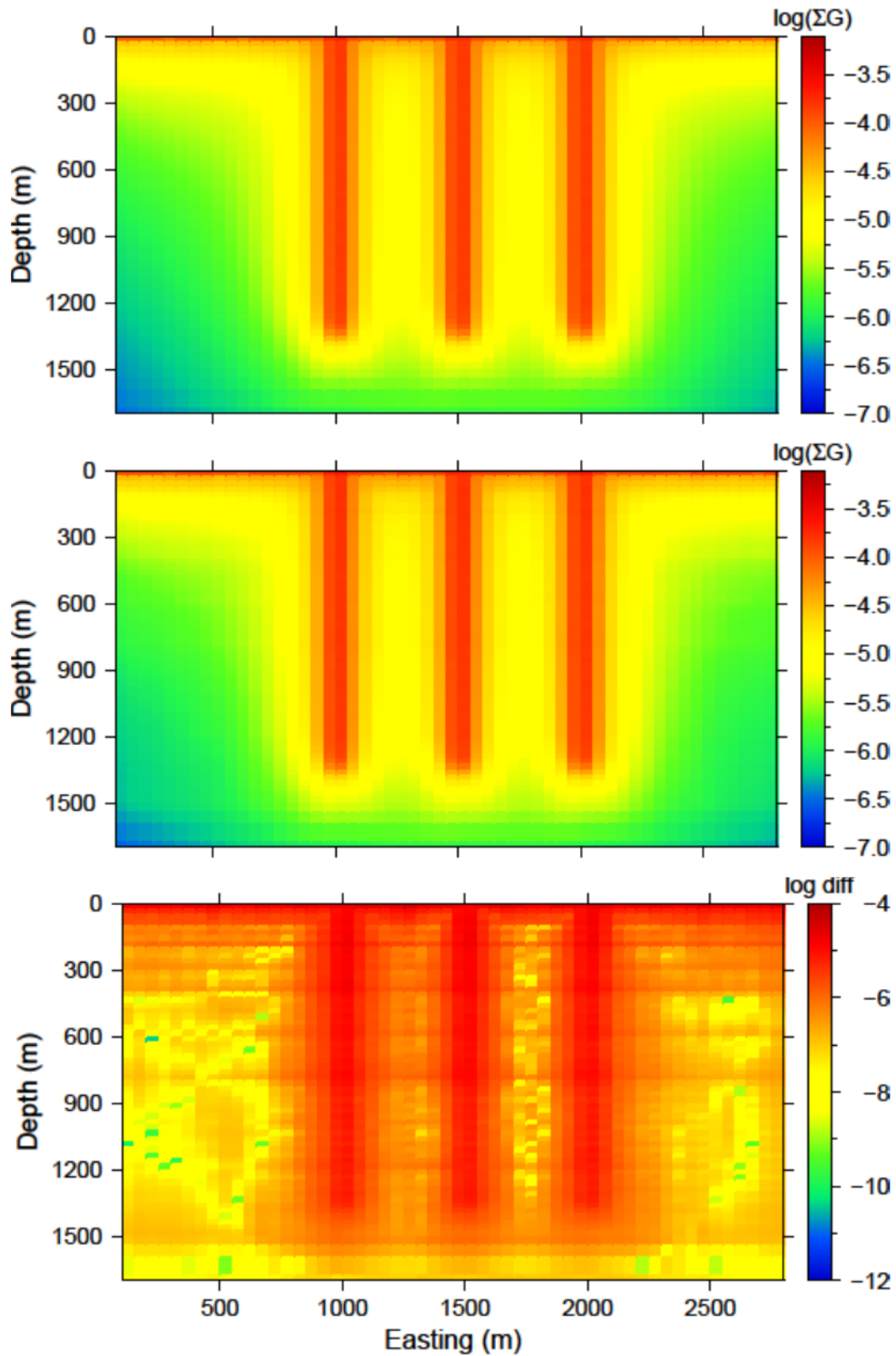


Fig. 5.14: (top) The log of average sensitivity for each cell prior to compression. (middle) the log of average sensitivity for each cell after compression with a 95% reconstruction accuracy. (bottom) The difference between (top) and (middle) on a log scale.

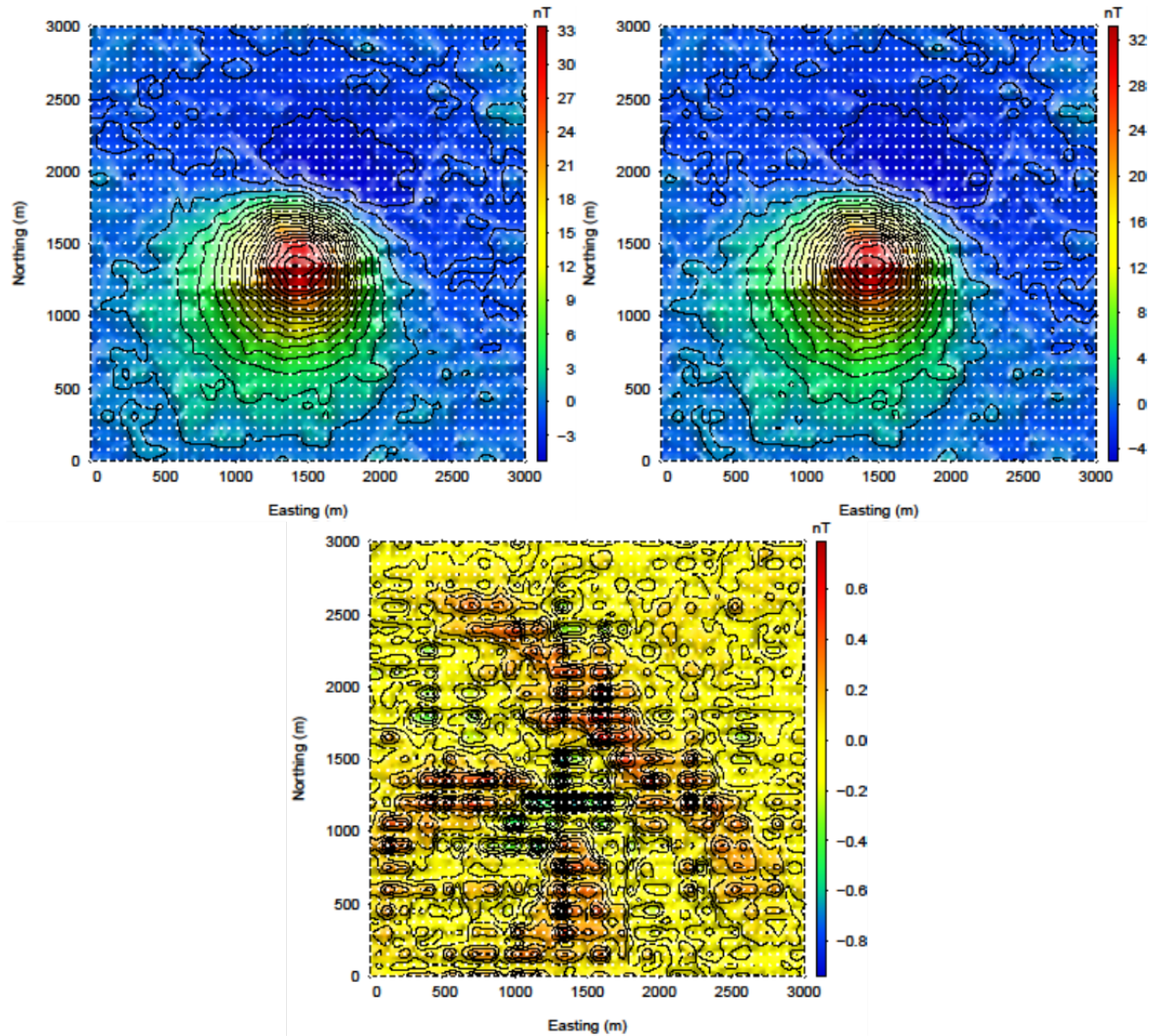


Fig. 5.15: (top-left) The predicted data from the uncompressed sensitivity matrix by using `magfor3d` on the recovered model. (top-right) The predicted data from the compressed sensitivity matrix given by `maginv3d` or calculated by `magpre3d` given the recovered model. (bottom) The difference of the two data sets is less than 1 nT. Data locations are denoted by the white dots.

CHAPTER 6

References
