GRAV3D Version 2.0

A Program Library for Forward Modelling and Inversion of Gravity Data over 3D Structures.

UBC-Geophysical Inversion Facility
Department of Earth and Ocean Sciences
University of British Columbia
Vancouver, British Columbia

January, 2001

Table of Contents

GRAV3D Manual Summary

The Program and its purpose

The program library GRAV3D is a suit of algorithms for inverting gravity data gathered over a three dimensional earth. Version 1.0 of this facility is described in detail in Li and Oldenburg (1998), and the current version (Version 2.0) adds greater flexibility and efficiency to the basic forward modelling and inversion algorithms. In particular it allows larger problems to be solved through the use of wavelet transforms, and it allows geophysical constraints, in the form of upper and lower bounds on the density of each cell, to be included.

The problem addressed by GRAV3D involves gravimetric data gathered anywhere at or above the surface of the Earth. These data are the vertical component of the gravity field caused by a three dimensional distribution of density contrast within the volume of ground directly beneath the survey area. This subsurface volume (with optional topography) is modelled as a set of rectangular cells each with constant density contrast. For forward calculations the anomalous density in each cell is known and the data that be measured over this known Earth model are calculated. The inverse problem involves estimating the density contrasts of all the cells based upon measurements gathered during a field survey.

The manual for GRAV3D V2.0 includes four sections:

1. theoretical background,
2. elements of the program library,
3. instructions for running the programs, and
4. two examples of synthetic inversions managed in several ways.

In the remainder of this summary we point out some highlights, and we emphasize the importance of understanding what kind of models the program can recover, and how the program operates.

Summary of the chapters on theoretical background

- The introduction provides a basic understanding of how gravity data relate to the Earth, and the goal of inverting such data.
- The solution to the forward problem is described, and an example of a forward calculation is provided.
- The method of solving the gravity inverse problem is outlined. In summary, the inversion is solved as an optimization problem with the simultaneous goals of *i)* minimizing an objective function on the model and *ii)* generating synthetic data that match observations to within a degree of misfit consistent with the statistics of those data.
- By minimizing the model objective function, distributions of subsurface density contrast are found that are both close to a reference and smooth in three dimensions. The degree to which either of these two goals dominates is controlled by defining length scales for smoothness. This is a crucial step and allows the user to incorporate *a priori* geophysical or geological

information into the inversion. Explicit prior information may also take the form of upper and lower bounds on the density contrast in any cell.

- The relative importance of the objective function the misfit term is controlled by a regularization parameter. This parameter is determined in one of three ways, and depends upon how much is known about errors in the measured data. Section 1.6 discusses this problem, and it is important to understand how the choice of options affects the outcome from the inversion program.
- Potential fields data have no inherent information about the distance between source and measurement, therefore the incorporation of a depth or distance weighting term in the formulation is critical. Section 1.4 describes this issue, and explains the options available for controlling how cells in the model enter into the solution regardless of their depth or distance from the measurements.
- The large size of useful 3D inversion problems is mitigated by the use of wavelet compression. Parameters controlling the implementation of this compression are available for advanced users, and section 1.5 provides some details on how wavelet compression is applied.

Summary of program elements, instructions for use, and examples

Files and file formats are similar to those of other UBC-GIF codes, although there are some aspects specific to GRAV3D V2.0. Highlights are noted below, but you must read Chapter 2 of the manual for details.

- The mesh file defines how the 3D subsurface volume of ground is discretized. Specify cell sizes that will image your target with adequate detail without resulting in a too many model cells. Then add padding cells outside the region of investigation. Recall that the program must invert a sensitivity matrix that has a size proportional to $N \times M$, where N is the number of data points and M is the total number of cells in the model. This sensitivity matrix should reside within the computer's memory for efficient execution. Problems with more than 10,000 to 20,000 model cells, and/or more than a few thousand data points would be considered large, and may be expected to require a considerable amount of computing time. Also, it is considered prudent to upward continue data that are gathered very close to the ground so that measurements appear as if collected at an elevation roughly equal to half a cell width. This is especially true in the presence of sever topography.
- Files that define topography, observation locations and observations (measured data) are self explanatory. The manual includes details on how to ensure consistency regarding elevations and topography.
- There are separate model files defining density contrasts for the initial model, reference model, upper and lower bounds on density contrast, and final output models, all with similar structures. These models are defined via the mesh file, and each cell has a constant density contrast. Also, you should be clear on how cells above topography are managed.

Forward modelling, sensitivity calculation and inversion programs are run from the command line with parameters specified in separate files. In the descriptions of these parameter files look for details regarding depth or distance weighting, wavelet compression, and how to specify model objective function parameters. There are also important notes on inversion "mode". Your choice of mode determines what method the program uses to find the regularization parameter. This choice is based upon what type of information you have on data errors, and error statistics.

Output files are also specified, and you should take particular note of the information provided in the log file. This file summarizes the progress of inversion, and careful examination of its information is a critical aspect of quality control. The log file contents depend upon which mode was use for inversion; see the final section of chapter 3 for details.

The manual concludes with two synthetic examples. The first illustrates the newly added facility for specifying upper and lower bounds on recovered cell density contrasts. The second example illustrates inversion of a larger problem.

Conclusions

Successful application of inversion results to geological problems demands understanding of the inversion process. While models will be as smooth and as close to a reference model as data misfit will allow, the actual results are controlled by careful selection of length scales, weighting functions, and constraining parameters. The size of the problem is reduced using wavelet compression and the management of misfit is controlled by the choice of mode and associated parameters. This degree of flexibility makes it imperative that the user has a good understanding of general inversion theory and the specifics of its implementation for GRAV3D.

1 General Background for GRAV3D

1.1 Introduction

This manual presents theoretical background, numerical examples, and explanation for implementing the program library GRAV3D. This suite of algorithms, developed at the UBC-Geophysical Inversion Facility, is needed to invert gravimetric responses over a 3 dimensional distribution of density contrast, or anomalous density (henceforth referred to as 'density' for simplicity). The manual is designed so that geophysicists who are familiar with the gravity experiment, but who are not necessarily versed in the details of inverse theory, can use the codes and invert their data. In the following, we describe the basics of the algorithm, but readers are referred to Li and Oldenburg (1997, 1998) for an in-depth discussion of various aspects of the algorithm. Note that an understanding of these components is necessary for the user to have a global view of the algorithms and to use the program library correctly.

A gravity experiment involves measuring the vertical components of the gravity field produced by anomalous (either excess or deficient) mass beneath the surface. A distribution of anomalous mass, characterized by density $\rho(x, y, z)$, produces its own gravity field, $\vec{F_s}$, which is superimposed on the ambient gravity field. By measuring the resultant field and removing the ambient field from the measurements through numerical processing, one obtains the field due to the anomalous mass.

The vertical component of the gravity field produced by the density $\rho(x, y, z)$ is given by

$$F_z(\vec{r}_0) = \gamma \int\limits_V \rho(\vec{r}) \frac{z - z_0}{|\vec{r} - \vec{r}_0|^3} dv \tag{1}$$

where $\vec{r}_0$ is the vector denoting the observation location and $\vec{r}$ is the source location. V represents the volume of the anomalous mass, and γ is the gravitational constant. Here we have adopted a Cartesian coordinate system having its origin on the earth's surface and the z-axis pointing vertically downward.

The data from a typical gravity survey are a set of field measurements acquired over a 2D grid. These data are first processed to yield an estimate of the anomalous field, which is due to the excess or deficient mass below the data area. The goal of the gravity inversion is to obtain, from the extracted anomaly data, quantitative information about the distribution of the anomalous density in the ground.

The GRAV3D library consists of three programs:

1. GZFOR3D: calculates the surface gravity data produced by a given density model.
2. GZSEN3D: calculates the sensitivity matrix for use in the inversion.
3. GZINV3D: inverts the anomalous gravity field to generate a density model.

In the following, we outline the basics of the forward and inverse procedures used by these programs.

1.2 Forward Modelling

Forward modelling of gravity data is a linear problem and can be carried out by performing the integration in eq.(1). We divide the region of interest into a set of 3D prismatic cells by using a 3D orthogonal mesh and assume a constant density within each cell. We discretize the density model in this manner since it is best suited for our inversion methodology. Given such a discretization, the gravity field at the *i'th* location can be written as,

$$\begin{aligned} d_i &\equiv F_z(\vec{r}_{0i}) \\ &= \sum_{j=1}^{M} \rho_j \left\{ \gamma \int_{\Delta V_j} \frac{z - z_0}{|\vec{r} - \vec{r}_{0i}|^3} dv \right\} \\ &\equiv \sum_{j=1}^{M} \rho_j G_{ij}. \end{aligned} \tag{2}$$

In eq.(2), ρ_j and ΔV_j are the density and volume of the *j'th* cell, d_i is introduced as a generic symbol for the *i'th* datum, and G_{ij}, defined by the expression in brackets, quantifies the contribution of the *j'th* cell to the *i'th* datum. The solution for the integral in eq.(2) can be found in Nagy (1966) and we have adopted the solution by Haaz (1953) here. Expressed in matrix notation, the gravity data consisting of N observations are given by

$$\mathbf{G}\vec{\rho} = \vec{d}, \tag{3}$$

where $\vec{d} = (d_1, \cdots, d_N)^T$ is the data vector and $\vec{\rho} = (\rho_1, \cdots, \rho_M)^T$ is the vector containing the density values of the M cells. The program GZFOR3D performs forward modelling calculation to generate the surface gravity data produced by a given density model. Program GZSEN3D calculates the complete matrix $\mathbf{G}$ to be used in the inversion, with optional wavelet compression applied.

To illustrate the forward modelling program, and to introduce the example used in the inversion section of this manual, we calculate the gravity anomaly on the surface of the earth produced by the density model shown in Fig.1. The model consists of a dipping dyke with anomalous density of 1.0 g/cm^3 situated in a uniform background. The model is represented by 4000 cells (with 20 in each horizontal direction and 10 in the vertical direction). The anomaly is calculated at a height of 0.5 m above the surface over a 21 by 21 grid with 50 m spacing in both directions. A contour presentation of the data is shown in Fig.2.

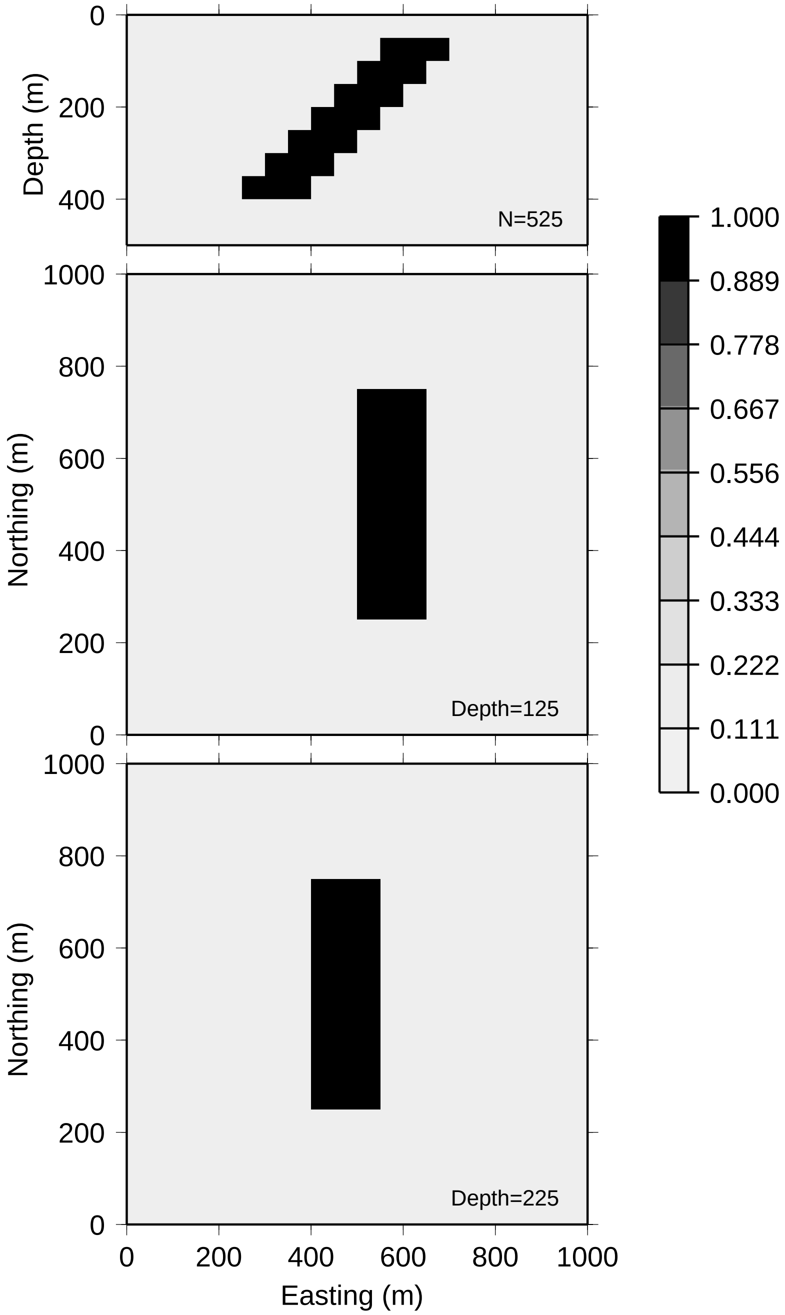


Figure 1. Synthetic model consisting of a dense dipping dyke buried in a uniform background. A cross-section at north 525 m is shown in the top panel and plan-sections at depths 125 m and 225 m respectively are shown below. The anomalous density of the dyke is 1 g/cm^3.

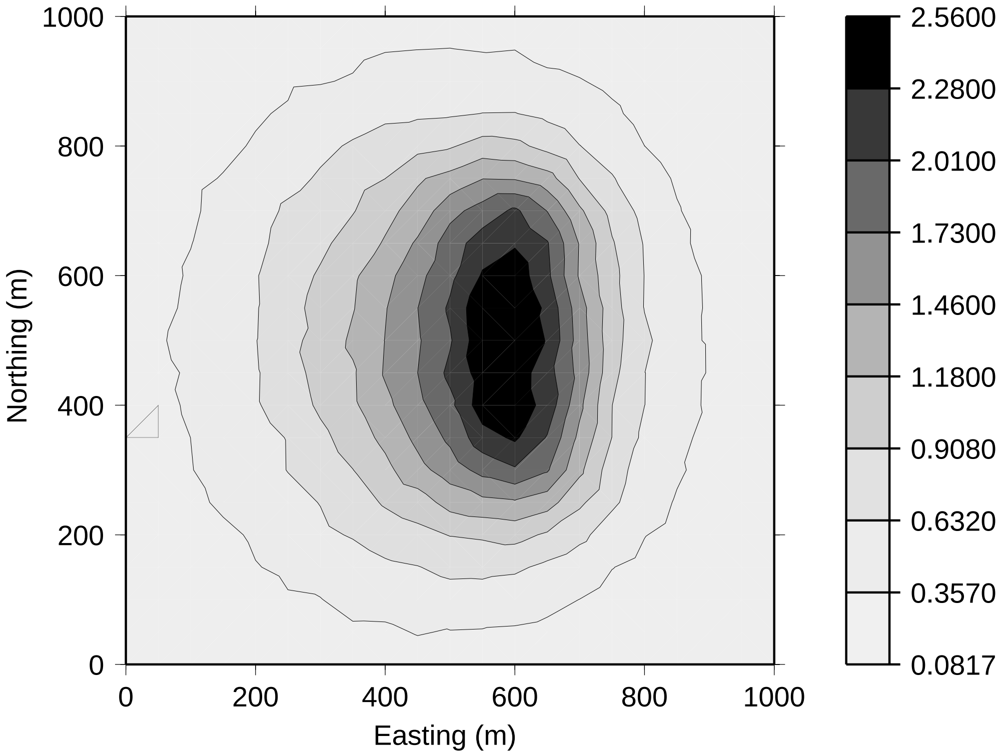


Figure 2. The gravity anomaly produced by the density model shown in Figure 1. The data are calculated over a 21 by 21 grid of 50 m spacings. The units are in mGal. Uncorrelated Gaussian noise has been added to the data. The standard deviation of the additive noise is 0.01 mGal plus 5% of the magnitude of the datum.

1.3 Inversion Methodology

The inverse problem is formulated as an optimization problem where an objective function of the density model is minimized subject to the constraints that the data be reproduced to within an error tolerance. The details of the objective function are problem-dependent and can vary according to the available a priori information, but generally the objective function should have the flexibility of constructing a model that is close to a reference model ρ_0 and producing a model that is smooth in three spatial directions. An objective function that accomplishes this is

$$\begin{aligned}\phi_m(\rho) = \alpha_s \int\limits_V w_s w^2(z)(\rho-\rho_0)^2 dv + \alpha_x \int\limits_V w_x \left(\frac{\partial w(z)(\rho-\rho_0)}{\partial x}\right)^2 dv \\ + \alpha_y \int\limits_V w_y \left(\frac{\partial w(z)(\rho-\rho_0)}{\partial y}\right)^2 dv + \alpha_z \int\limits_V w_z \left(\frac{\partial w(z)(\rho-\rho_0)}{\partial z}\right)^2 dv\end{aligned} \tag{4}$$

where the functions w_s, w_x, w_y, and w_z are spatially dependent, whereas α_s, α_x, α_y, and α_z are coefficients which affect the relative importance of the different components of the objective function. The greater the ratio α_x/α_s and so on, the smoother the recovered model is in that axis direction. A length scale of smoothness can be defined for each direction as $L_x = \sqrt{\alpha_x/\alpha_s}$, $L_y = \sqrt{\alpha_y/\alpha_s}$, $L_z = \sqrt{\alpha_z/\alpha_s}$; and specifying greater values for the length scales will produce smoother models.

The objective function in eq.(4) has the flexibility to allow many different models to be constructed. The reference model ρ_0 may be a general background model that is estimated from previous investigations or it could be the zero model. 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. The weighting functions w_x, w_y, and w_z can be designed to enhance or attenuate structures in various regions in the model domain. If geology suggests a rapid transition zone in the model, then a decreased weighting for flatness can be put there and the constructed model will exhibit higher gradients provided that this feature does not contradict the data.

The function $w(z)$ in eq. (4) is a depth weighting that depends upon the model discretization and observation location. The weighting function is used to counteract the decay of the kernel functions, G_{ij}, with depth. Because of this decay, an inversion which minimizes $||\rho - \rho_0||^2$ subject to fitting the data will generate a density distribution that is concentrated near the surface. We introduce a weighting of the form $w(z) = (z + z_0)^{-\beta/2}$ into the objective function to counteract this effect. The values of β and z_0 are discussed in the next section. They are important in defining the model objective function and, thus, the type of final model.

The next step in setting up the inversion is to define a measure of misfit. Here we use the 2–norm measure

$$\phi_d = \left|\left|W_d\left(d - d^{obs}\right)\right|\right|^2. \tag{5}$$

We assume that the noise contaminating the data is independent and Gaussian with zero mean. Specifying 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 random variable distributed with N degrees of freedom. Accordingly $E[\phi_d] = N$ provides a target misfit for the inversion.

The inverse problem is solved by finding a density $\rho(\vec{r})$ which minimizes ϕ_m and misfits the data according to the noise level. This is accomplished by minimizing $\phi(\rho) = \phi_d + \mu\phi_m$ where $\mu \in [0, \infty]$ is a regularization parameter that controls the relative importance of data misfit ϕ_d and model objective function ϕ_m. A value of μ is sought so that the data are neither fit too well nor too poorly. To perform a numerical solution, we first discretize the objective function in eq.(4) according to the mesh that defines the density model. This yields

$$\begin{aligned}\phi_m(\vec{\rho}) &= (\vec{\rho}-\vec{\rho}_0)^T\Big(W_s^TW_s + W_x^TW_x + W_y^TW_y + W_z^TW_z\Big)(\vec{\rho}-\vec{\rho}_0)\\ &\equiv (\vec{\rho}-\vec{\rho}_0)^T\Big(W_m^TW_m\Big)(\vec{\rho}-\vec{\rho}_0)\\ &= ||W_m(\vec{\rho}-\vec{\rho}_0)||^2\end{aligned} \tag{6}$$

where $\vec{\rho}$ and $\vec{\rho}_0$ are *M*-length vectors. The individual matrices W_s, W_x, W_y, W_z are straightforwardly calculated once the model mesh and the weighting functions $w(z)$ and $w_s,\ w_x,\ w_y,\ w_z$ are defined. The cumulative matrix $W_m^TW_m$ is then formed. The details of these quantities are specified by the user through the inputs to programs GZSEN3D and GZINV3D.

Since the density contrast is limited to a small range for any practical problems, and often there are well-defined bounds on the density contrast based on direct sampling or other geological information, we also impose constraints to restrict the solution to lie between a lower and upper bound. Thus the solution is obtained by the following constrained minimization problem,

$$\begin{aligned}&\text{minimize}: \quad \phi = \phi_d + \mu\phi_m,\\ &\text{subject to}: \quad \vec{\rho}_{\min} \le \vec{\rho} \le \vec{\rho}_{\max},\end{aligned} \tag{7}$$

where $\vec{\rho}_{min}$ and $\vec{\rho}_{max}$ are vectors containing the lower and upper bounds on the model values.

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 μ that produces the expected misfit. Otherwise, an estimated value of μ will be prescribed. The details of various aspects of choosing a regularization parameter will be discussed in a following section.

We use a primal logarithmic barrier method with the conjugate gradient technique as the central solver. In the logarithmic barrier method, the bound constraints are implemented as a logarithmic barrier term. The new objective function is given by,

$$\phi(\lambda) = \phi_d + \mu\phi_m - 2\lambda\sum_{j=1}^{M}\Big[\ln\big(\rho_j - \rho_j^{min}\big) + \ln\big(\rho_j^{max} - \rho_j\big)\Big], \tag{8}$$

where λ is the barrier parameter, and the regularization parameter μ is fixed during the minimization. As the name suggests, the logarithmic barrier term forms a barrier along the boundary of the feasible domain and prevents the minimization from crossing over to the infeasible region. The method solves a sequence of nonlinear minimizations with decreasing λ and, as λ approaches zero, the sequence of solutions approaches the solution of eq.(7).

The above methodology provides a basic framework for solving 3D gravity inversion with arbitrary observation locations. The basic components are the forward modelling, a model objective function that incorporates a "depth" weighting, a data misfit function, a regularization parameter that ultimately determines how well the data will be fit, and the logarithmic barrier method to obtain the solution with bound constraints. Without getting into the algorithmic details, we discuss three of these basic components in the next sections, namely, the depth weighting, efficient forward mapping, and the choice of the regularization parameter. An understanding of these components is necessary for the user to have a global view of the algorithm and to use the program library correctly.

1.4 Depth Weighting

It is well known that gravity data have no inherent depth resolution. As a result, structures concentrate near the surface in a simple smallest or flattest model regardless of the true depth of the causative bodies. In terms of model construction, this is a direct manifestation of the kernels' decay with depth: because of their rapidly diminishing amplitude at depth, the kernels of surface data are not sufficient to generate a function that possess significant structure at depth. In order to overcome this, the inversion needs to introduce a depth weighting to counteract this natural decay. Intuitively, such a weighting will approximately cancel the natural decay and give cells at different depths equal probability of entering the solution with a non-zero density.

For data acquired over a relatively flat surface, the sensitivity decays predominantly in the depth direction. Numerical experiments indicate that a function of the form $(z+z_0)^{-2}$ closely approximates the kernels' decay directly under the observation point provided that a reasonable value is chosen for z_0. Having the exponent to be 2 is consistent with the fact that the gravity field decays as inverse distance squared. The value of z_0 can be obtained by matching the function $(z+z_0)^{-2}$ with the field produced by a column of cells directly beneath the observation point. In accordance with the discretization used in the inversion, we use a depth weighting function of the form

$$w(\vec{r}_j) = \left[\frac{1}{\Delta z_j} \int\limits_{\Delta z_j} \frac{dz}{(z+z_0)^2} \right]^{1/2}, \qquad j = 1, \cdots, M \tag{9}$$

for the inversion and $\vec{r}_j$ is used to identify the *j*th cell and Δz_j is its thickness. This weighting function is first normalized so that the maximum value is unity. Numerical tests indicate that when this weighting is used, the density model constructed by minimizing a model objective function in eq.(4), subject to fitting the data, places the recovered anomaly at approximately the correct depth.

For data sets acquired over rugged terrain, the simple depth decay does not describe the dominant variation in sensitivity. Therefore a weighting function that varies in three dimensions is needed. We generalize the depth weighting to form a distance weighting:

$$w(\vec{r}_j) = \frac{1}{\sqrt{\Delta V_j}} \left\{ \sum_{i=1}^{N} \left[\int\limits_{\Delta V_j} \frac{dv}{(R_{ij}+R_0)^2} \right]^2 \right\}^{1/4}, \qquad j = 1, \cdots, M \tag{10}$$

ΔV_j is the volume of *j*th cell, R_{ij} is the distance between a point in ΔV_j and the *i*th observation, and R_0 is a small constant used to ensure that the integral is well defined (chosen to be a quarter of the smallest cell dimension). Similarly, this weighting function is normalized to have a maximum value of unity. For inversion of data sets acquired in areas with high topographic relief, it is advantageous to use this more general form of weighting function.

The weighting function is directly incorporated in the sensitivity file generated by program GZSEN3D. This program allows the user to specify whether to use the depth weighting or the distance weighting depending on the terrain of the observed data.

1.5 Wavelet Compression of Sensitivity Matrix

The two major obstacles to the solution of large scale gravity inversion problems 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. The GRAV3D 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 Li and Oldenburg (1997). In the following, we give a brief description of the method necessary for the use of the GRAV3D library.

Each row of the sensitivity matrix in a 3D gravity 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 the coefficients with small magnitude 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,

$$\tilde{\mathbf{G}} = \mathbf{G}\mathcal{W}^T. \qquad (11)$$

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

$$\tilde{g}^s_{ij} = \begin{cases} \tilde{g}_{ij}, & |\tilde{g}_{ij}| \ge \delta_i \\ \\ 0, & |\tilde{g}_{ij}| < \delta_i \end{cases}, \qquad i = 1, \cdots N \qquad (12)$$

where δ_i is the threshold level, and $\tilde{g}_{ij}$ and $\tilde{g}^s_{ij}$ 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:

$$r_i(\delta_i) = \sqrt{\frac{\sum\limits_{|\tilde{g_{ij}}|<\delta_i} \tilde{g_{ij}}^2}{\sum\limits_{j} \tilde{g_{ij}}^2}}, \qquad i = 1, \cdots, N. \qquad (13)$$

Here the numerator is the norm of the discarded coefficients. For each row we choose δ_i such that $r_i(\delta_i) = r^*$, where r^* is the prescribed reconstruction accuracy. However, this is a costly process. Instead, we choose a representative row, i_0, and calculate the threshold level δ_{i_0}. This threshold is then used to define a relative threshold $\epsilon = \delta_{i_0} / \max_j |\tilde{g}_{i_0 j}|$. The absolute threshold level for each row is obtained by

$$\delta_i = \epsilon \max_j \left(|\tilde{g_{ij}}| \right), \qquad i = 1, \cdots, N. \tag{14}$$

The program that implements this compression procedure is GZSEN3D. The user is asked to specify the relative error r^* and the program will determine the relative threshold level ϵ. Usually a value of a few percent is appropriate for r^*. For experienced users, the program also allows the direct input of the relative threshold level ϵ.

1.6 Choice of Regularization Parameter

The choice of the regularization parameter μ ultimately depends upon the magnitude of the error associated with the data. The inversion of noisier data requires heavier regularization, thus a greater value of μ is required. In this section, we discuss the various implementations for the choice of μ in the GRAV3D library.

If the standard deviation associated with each datum is known, then the data misfit defined by eq.(5) has a known expected value ϕ_d^*, which is equal to the number of data when the errors are assumed to be independent Gaussian noise with zero mean. The value of μ should be such that the expected misfit is achieved. This entails a line search based on the misfit curve as a function of μ. Because of the positivity constraint, our problem is nonlinear. Thus for each μ a nonlinear solution using a logarithmic barrier method must be obtained. This is computationally demanding and we therefore have developed the following strategy to reduce the cost.

It is observed that, when plotted on a log-log scale, the misfit curves for 3D inversion with and without bound constraints often parallel each other in the vicinity of the expected misfit. The curve with positivity must lie above the curve without positivity. Therefore, we can first perform a line search without positivity to find a μ_0 that gives rise to ϕ_d^*. This search also generates the slope, s_0, of misfit curve at μ_0. This process is very efficient and the required CPU time is much smaller compared to the time required for the solution with positivity. We next assume that s_0 can be used to approximate the slope of the misfit curve when the positivity is imposed. A rigorous line search incorporating positivity starts with an initial guess of $\mu = 0.5\mu_0$. This usually yields a misfit that is very close to the target value. If the misfit is not sufficiently close to ϕ_d^*, however, a new guess for μ is obtained which makes use of the approximate slope s_0. The inversion with updated μ can be solved efficiently if the logarithmic barrier algorithm is started from an initial model close to the final solution. That model is obtained by perturbing the solution corresponding to the previous μ away from the zero bound. The line search using this strategy is often successful in reaching the target ϕ_d^* after testing two to four values of μ. This strategy is implemented in GZINV3D as the first method for choosing the tradeoff parameter.

In practical applications the estimate of data error is often not available. Then the degree of regularization, hence the value of μ, needs to be determined based on other criteria. A commonly used method in linear inverse problems is the generalized cross-validation (GCV) technique. The use of GCV in inverse problems with inequality constraints such as positivity is far more involved and numerically expensive to implement. However, applying GCV on the 3D gravity inversion without bounds still produces a reasonable estimate of the data error when the data are dominated by anomalies of the same sign. That error can serve as a starting point for further adjustment by the user based on his or her judgement. Since no other information is assumed, we have chosen to use the value of μ obtained in this manner directly in the final inversion, which has bound constraints imposed. In this case, only one logarithmic barrier solution is needed. Numerical tests have indicated that this simplistic use of GCV is in fact surprisingly effective unless the data have a large negative bias or are distributed sparsely. GZINV3D has implemented this approach as the third method for choosing the tradeoff parameter.

Figure 3 illustrates the structure of the program GZINV3D. It has three options for determining the tradeoff parameters. The controlling parameter is *mode*. When $mode = 1$, the line search based on known target value of data misfit is used. Two stages, as discussed above, are

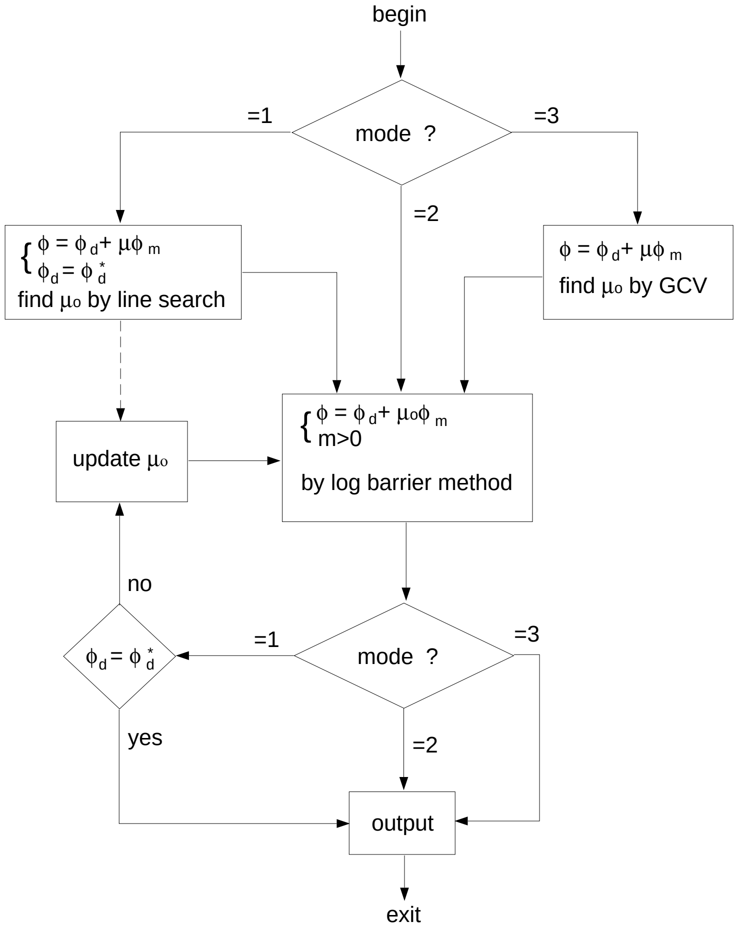


Figure 3. Flow chart illustrating the solution of the 3D gravity inversion by GZINV3D using different strategies for choosing the tradeoff parameter.

used, and several solutions for different values of μ must be tested to obtain one that produces the target misfit. When $mode = 2$, the user specifies a tradeoff parameter and a single solution is produced. When $mode = 3$, the program first performs GCV analysis on the inversion without positivity and then uses the resultant value of μ in the final inversion.

1.7 Example Inversion

We now invert the surface gravity data in the example given in Section 1.2 using program GZINV3D. The model region is again discretized into 20 by 20 by 10 cells of 50 m on a side. We invert the 441 noise-contaminated data illustrated in Fig.2 to recover the density contrasts within the 4000 cells.

For the model objective function, we choose $L_x = L_y = L_z = 100.0$ and unit 3D weighting functions for all components and the default parameters for $w(z)$. A zero reference model is used. We have also imposed a lower bound of 0.0 and an upper bound of 2.0.

The final model is shown in Fig. 4. The top panel is a cross-section through the centre of the model and the other two panels are horizontal slices at different depths. The tabular shape of the anomaly and its dipping structure are very clear and the depth extent is reasonably recovered. The dip angle inferred from the recovered model is close to the true value. The amplitude of the recovered model is slightly less than the true value. Over all, however, this recovered model compares favorably with the true model shown in Fig.1.

1.8 References

Haaz, I. B., 1953, Relationship between the potential of the attraction of the mass contained in a finite rectangular prism and its first and second derivatives, Geofizikai Kozlemenyek, II, No 7.

Li, Y. and Oldenburg, D. W., 1997, Fast inversion of large scale magnetic data using wavelets, 67th Ann. Internat. Mtg., Soc. Expl. Geophys., Expanded Abstracts, 490–493.

Li, Y. and Oldenburg, D. W., 1998, 3D inversion of gravity data, Geophysics, **63, 109–119**

Nagy, D. 1966, The gravitational attraction of a right rectangular prism, Geophysics, **31,** 361–371.

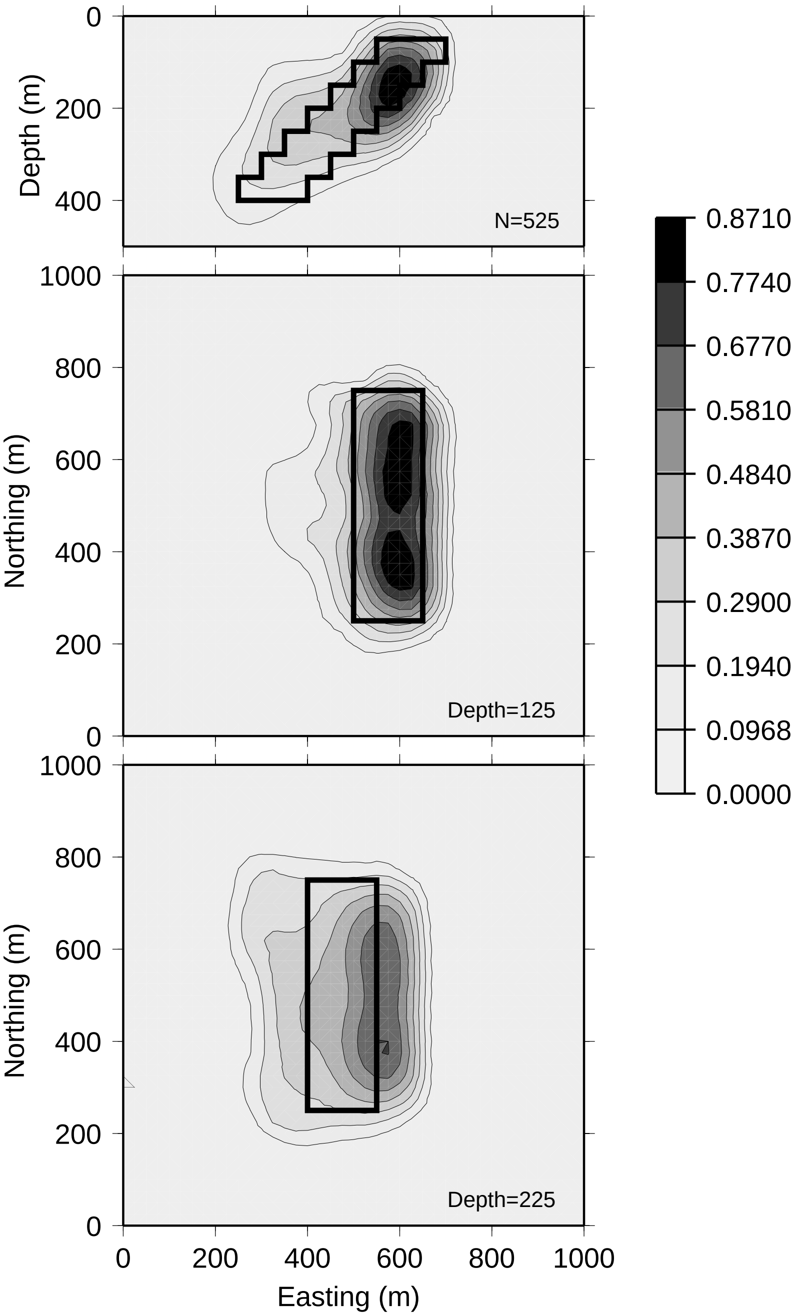


Figure 4. The recovered density model displayed in one cross-section and two horizontal slices. The location of the true model is indicated by the bold outlines.

2 Elements of the GRAV3D (Ver2.0) Program Library

2.1 Introduction

The GRAV3D library consists of three major programs:

1. **gzfor3d:** performs forward modelling
2. **gzsen3d:** calculates sensitivity and the depth weighting function.
3. **gzinv3d:** performs 3D gravity inversion

Each of the above programs requires input files, as well as 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.

2.2 General Files for GRAV3D Programs

There are six general files which are used in GRAV3D. These are:

1. **mesh:** 3D mesh defining the discretization of the 3D model region
2. **topo.dat:** file specifying the surface topography
3. **obs.loc:** file specifying the observation locations
4. **obs.grv:** file specifying the observation locations and the observed gravity anomalies with estimated standard deviation
5. **model.den:** density model file
6. **bounds.den:** lower and upper bounds
7. **w.dat:** contains the 3D weighting functions

FILE: mesh

This file contains the 3D mesh which defines the model region. **mesh** has the following structure:

$\mathtt{NE\ NN\ NV}$

$E_0\ N_0\ V_0$

$\Delta E_1\ \Delta E_2\ \ldots\ \Delta E_{NE}$

$\Delta N_1\ \Delta N_2\ \ldots\ \Delta N_{NN}$

$\Delta V_1\ \Delta V_2\ \ldots\ \Delta V_{NV}$

NE Number of cells in the East direction.

NN Number of cells in the North direction.

NV Number of cells in the vertical direction.

E_0, N_0, V_0 Coordinates, in meters, of the southwest top corner, specified in (Easting, Northing, Elevation). The elevation can be relative, but it needs to be consistent with the elevation used to specify the observation location in **obs.loc** or **obs.grv** and in **topo.dat** (see the relevant files for description).

ΔE_n Cell widths in the easting direction (from W to E).

ΔN_n Cell widths in the northing direction (from S to N).

ΔV_n Cell depths (top to bottom).

The mesh should be designed in accordance with the area of interest, the desired resolution, and the total memory available in the computer that will run the inversion. 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 spacing of the data should be comparable to the size of the cells. See also the description of the “obs.grv” data file. There is no restriction on the relative position of data location and nodal points in horizontal directions. The cell width in this area is usually uniform. The maximum depth of the mesh used for inversion should be large enough so that the no density contrast below that depth would produce a noticeable anomaly with the length scale covered by the data area. A rule of thumb is that the maximum depth should be at least half of the longest side of the data area. Based upon the user’s knowledge of the survey area, one may adjust the maximum depth as necessary. The cell thickness in vertical direction usually increases slightly with depth. In the shallow region, the ratio of thickness to width of about 0.5 is good, especially when surface topography is present. At depth, a cell thickness close to the cell width is advisable. Once this core mesh is designed, it can be extended laterally by padding with a few cells, possibly of variable width. This padding is necessary when the extracted anomalies are close to the boundary of the core mesh or if there are influences from anomalies outside the area which cannot be easily removed. Problems with more than 10,000 to 20,000 model cells, and/or more than a few thousand data points would be considered large, and can be expected to require a considerable amount of computing memory and time.

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 and one 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 V_0) is just above the highest elevation point, and then to strip off cells that are above the topographic surface. This is the approach taken in GRAV3D. The number of cells to be stripped off in each column is determined by the user supplied topography file **topo.dat** (see explanation under **FILE: topo.dat**). Only the remaining cells will be used in the forward modelling or included in the inversion as model parameters.

Example of the mesh file: The following is a 10×10×5 mesh where each cell is 50m by 50m by 50m.

```
10 10 5

0 0 0

50.0 50.0 50.0 50.0 50.0 50.0 50.0 50.0 50.0 50.0

50.0 50.0 50.0 50.0 50.0 50.0 50.0 50.0 50.0 50.0

50.0 50.0 50.0 50.0 50.0
```

FILE: topo.dat

This optional file is used to define the surface topography of the 3D model by specifying the elevation at different locations. **topo.dat** has the following structure:

```
!  comment
!
npt
```
E_1 N_1 elev_1

E_2 N_2 elev_2

:

E_{npt} N_{npt} elev_{npt}

! comments top lines beginning with `!` are comments.

npt number of points.

E_i,N_i,$elev_i$ Easting, Northing and elevation of the i^{th} point. The elevation in this file and V_o in the **mesh** file must be specified relative to a common reference.

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. GRAV3D assumes a set of scattered points for generality and uses 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.

If **topo.dat** is not supplied, the surface will be treated as being flat.

Example of topo.dat file:

```
!! topographic data
4
   0.0    0.0   50.0
   0.0 1000.0   50.0
1000.0    0.0  -50.0
1000.0 1000.0  -50.0
```

NOTE: Although the cells above the topographic surface are removed from the model, they must still be included in the model file, **model.den**, 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 gzinv3d also includes the cells that are excluded from the model, but these cells will have a value of -100.0 to identify them.

FILE: obs.loc

This file is used to specify the observation locations. It is used by the forward modelling code only. The following is the file structure of **obs.loc**:

```
!  comments ...
!
ndat
$E_1$          $N_1$          $Elev_1$
$E_2$          $N_2$          $Elev_2$
:
$E_{ndat}$     $N_{ndat}$     $Elev_{ndat}$
```

! comments top lines beginning with `!` are comments.

ndat number of observations.

E_n, N_n, $Elev_n$ Easting, Northing and elevation of the observation, measured in meters. Elevation values of the observation locations should be above the surface topography since we assume all data are acquired above the surface. The observation locations can be listed in any order.

Easting, Northing and elevation information should be in the same coordinate system as defined in the mesh.

Example of obs.loc file: We provide an example file below. This file specifies 441 observation locations.

Example:

```
! Test data
!
441                    !! # of data
     0.00      0.00   1.0
     0.00     50.00   1.0
     0.00    100.00   1.0
:
  1000.00    900.00   1.0
  1000.00    950.00   1.0
  1000.00   1000.00   1.0
```

FILE: obs.grv

This file is used to specify the observed gravity anomalies, their locations and their estimated standard deviations. The values of parameters specifying observation locations are identical to those in **obs.loc**. The output of the forward modelling program **gzfor3d** has the same structure except that the column of standard deviations for the error is omitted. The following is the file structure of **obs.grv**:

```
!  comments ...
!
ndat
E_1       N_1       Elev_1     Grav_1      Err_1
E_2       N_2       Elev_2     Grav_2      Err_2
:
E_ndat    N_ndat    Elev_ndat  Grav_ndat   Err_ndat
```

! comments top lines beginning with `!` are comments.

ndat number of observations.

E_n, N_n, $Elev_n$ Easting, Northing and elevation of the observation, measured in meters. Elevation values of the data locations should be above topography. Observation elevations and V_0 in the mesh file must be specified relative to a common reference. The observations can be listed in any order.

$Grav_n$ gravity anomaly data, measured in mGal.

Err_n standard deviation of $Grav_n$. This quantifies the absolute error, and the standard deviation CANNOT be zero or negative.

It should be noted that the data $Grav_n$ are extracted anomalies after standard data reduction and regional removal are applied. A suitable data set is one in which there are just a few data values for each mesh cell at the Earth's surface. Also, it is prudent to upward continue data so they appear as if they were gathered at an elevation roughly equal to half a cell width above the Earth's surface.

Example of obs.grv file:

```
! Examples file
!
      441                         !! # of data
      0.00      0.00      2.00  0.978189E-01  0.293238E-02
      0.00     50.00      2.00  0.109359E+00  0.320770E-02
      0.00    100.00      2.00  0.122059E+00  0.351328E-02
      ...
      0.00    500.00      2.00  0.236872E+00  0.562092E-02
      0.00    550.00      2.00  0.224701E+00  0.556866E-02
```

FILE: model.den

This file contains the cell values of the density model. The density contrast must have values in g/cm^3. The following is the file structure of **model.den**:

$\texttt{den}_{\texttt{1,1,1}}$

$\texttt{den}_{\texttt{1,1,2}}$

:

$\texttt{den}_{\texttt{1,1,NV}}$

$\texttt{den}_{\texttt{1,2,1}}$

:

$\texttt{den}_{\texttt{i,j,k}}$

:

$\texttt{den}_{\texttt{NN,NE,NV}}$

den$_{i,j,k}$ density contrast at location *[i j k]*.

`[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×NE×NV, where NN is the number of cells in the North direction, NE is the number of cells in the East direction, and NV is the number of cells in the vertical direction. The lines must be ordered so that `k` changes the quickest (from 1 to NV), followed by `j` (from 1 to NE), then followed by `i` (from 1 to NN). If the surface topography (**topo.dat**) file is supplied, the values above the surface will be ignored. These values should be assigned an large negative value (e.g. –100.0) to avoid confusion with the other model elements.

FILE: bounds.den

This file contains the cell values of the lower and upper bounds on the sought density model. It is only required optionally by **gzinv3d**. The bounds have the same dimension as the density contrast. The following is the file structure of **bounds.den**:

$\mathtt{lb}_{1,1,1}$	$\mathtt{ub}_{1,1,1}$
$\mathtt{lb}_{1,1,2}$	$\mathtt{ub}_{1,1,2}$
:	
$\mathtt{lb}_{1,1,NV}$	$\mathtt{ub}_{1,1,NV}$
$\mathtt{lb}_{1,2,1}$	$\mathtt{ub}_{1,1,1}$
:	
$\mathtt{lb}_{i,j,k}$	$\mathtt{ub}_{j,j,k}$
:	
$\mathtt{lb}_{NN,NE,NV}$	$\mathtt{ub}_{NN,NE,NV}$

$lb_{i,j,k}$ lower bound on cell *[i j k]*.
$ub_{i,j,k}$ upper bound on cell *[i j k]*.

The ordering of the cells is the same as that for model cells: `[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×NE×NV, where NN is the number of cells in the North direction, NE is the number of cells in the East direction, and NV is the number of cells in the vertical direction. The lines must be ordered so that `k` changes the quickest (from 1 to NV), followed by `j` (from 1 to NE), then followed by `i` (from 1 to NN). If the surface topography (**topo.dat**) file is supplied, the bounds for cells above the surface will be ignored. These values should be assigned a large negative value (e.g. –100.0) to avoid confusion.

FILE: w.dat

User supplied weighting function. The following is the file structure for **w.dat**:

$\texttt{W.S}_{\texttt{1,1,1}}$ ... $\texttt{W.S}_{\texttt{NN,NE,NV}}$

$\texttt{W.E}_{\texttt{1,1,1}}$ ... $\texttt{W.E}_{\texttt{NN,NE-1,NV}}$

$\texttt{W.N}_{\texttt{1,1,1}}$ ... $\texttt{W.N}_{\texttt{NN-1,NE,NV}}$

$\texttt{W.Z}_{\texttt{1,1,1}}$ ... $\texttt{W.Z}_{\texttt{NN,NE,NV-1}}$

$W.S_{i,j,k}$ cell weights for the smallest model.
$W.E_{i,j,k}$ 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.
$W.Z_{i,j,k}$ cell weights for the interface perpendicular to the vertical direction.

Within each part, the values are ordered in the same way as in **model.den**, 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)*NN*NV` values, whereas the number of cells is `NE*NN*NV`.

If the surface topography (**topo.dat**) file is supplied, the cell weights above the surface will be ignored. These weights should be assigned a value of -1.0 to avoid confusion. If `null` is entered instead of the file **w.dat**, then all of the cell weights would equal 1.0.

3 Executing the GRAV3D Programs

3.1 Introduction

All programs in the package can be run by typing the program name followed by command line arguments. With such a format, they can be executed directly on the command line or in a shell script. When a program is executed without any arguments, it will print a simple message describing the usage. The command format is described below.

Command format:

```
PROGRAM   arg_1 arg_2 arg_3 ...
```

PROGRAM name of the executable program. If the program is not in the current directory, its path must be included also.

arg_n a command line argument which is the name of a file. It is usually one of those described in the preceding section or a control file containing input parameters.

3.2 gzfor3d

This program performs forward modelling.

Command line usage:

```
gzfor3d   mesh obs.loc model.den [topo.dat]
```

Input files:

mesh 3D mesh

obs.loc observation locations.

model.den density model.

topo.dat surface topography (optional). If omitted, the surface will be treated as being flat.

Output file:

gzfor3d.grv computed gravity anomaly data. Since the data in this file are accurate, the column of the standard deviations for the error is not included.

3.3 gzsen3d

This program performs the sensitivity and depth weighting function calculations.

Command line usage:

`gzsen3d   gzsen3d.inp`

For a sample input file, type: **`gzsen3d –inp`**

Format of the control file **gzsen3d.inp**:

```
mesh
obs.grv
topo.dat
iwt
beta   znot
wvlet
itol   eps
```

Input files:

mesh 3D mesh

obs.grv data file. Contains the observation locations and the observed gravity anomalies with estimated standard deviation.

topo.dat surface topography (optional). If `null` is entered, the surface will be treated as being flat.

Control parameters:

iwt an integer identifying the type of generalized depth weighting to use in the inversion.
`=1` for depth weighting (for data located on a flat surface),
`=2` for distance weighting (useful in the rugged terrian).

beta, *znot* parameters defining the depth weighting function.
When *iwt=1*, *beta* and *znot* are used as β and z_0 to define the depth weighting.
When *iwt=2*, *beta* and *znot* are used as β and R_0 to define the distance weighting.
If `null` is entered on this line (line 5), then the program sets *beta*=2 and calculates the value of *znot* based upon the mesh and data location. This is true for *iwt=1* or *2*.

For most inversions, however, setting this input line to “`null`” is recommended . The option for inputting β and *znot* is provided for experienced users who would like to investigate the effect of the generalized depth weighting for special purposes. The value of *beta* should normally be close to 2.0. (Note the difference from the value of 3.0 in MAG3D.) Smaller values of β give rise to weaker weighting.

wvlet 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 should be used for most inversions, while the others are provided for users' experimentation. If `null` is entered, no compression is performed and the program generates a dense matrix in its original form.

itol, eps an integer and a real number that specify how the wavelet threshold level is to be determined.
itol=1: program calculates the relative threshold and *eps* is the relative reconstruction error of the sensitivity. A reconstruction error of 0.05 is usually adequate.
itol=2: the user defines the threshold level and *eps* is the relative threshold to be used.
If `null` is entered on this line, a default relative reconstruction error of 0.05 is used and the relative threshold level is calculated (i.e., *itol*=1, *eps*=0.05).

The detailed explanation of threshold level and reconstruction error can be found in the Background (section 1.5) of MAG3D Ver3.0 Manual.

Output file:

gzinv3d.mtx sensitivity matrix file to be used in the inversion. This file contains the sensitivity matrix, generalized depth weighting function, mesh, and discretized surface topography.

Example of gzsen3d.inp control file:

```
mesh            ! mesh file
obs.nois        ! data file
null            ! topography
2               ! iwt=1 depth, =2 distance
null            ! beta, znot | null
daub2           ! wavelet type
1  0.05         ! itol, eps | null
```

3.4 gzinv3d

This program performs 3D gravity inversion.

Command line usage:

```
gzinv3d   gzinv3d.inp
```

For a sample input file, type: **`gzinv3d –inp`**

Format of the control file **gzinv3d.inp**

```
irest
mode
par     tolc
obs.grv
gzinv3d.mtx
ini.den
ref.den
bounds.den
L_E L_N L_Z
w.dat
idisk
```

Control parameters:

irest restarting flag:

=0: start inversion from scratch.

=1: restart inversion after it is interrupted. Restart requires two files written out by **gzinv3d** before the interruption: *gzinv3d.aux* and *gzinv3d.kap* (see below).

mode an integer specifying one of three choices for determining the regularization parameter (see the flowchart in Figure 3).

mode=1: the program chooses the regularization parameter by carrying out a line search so that the target value of data misfit is achieved.

mode=2: the user inputs the regularization parameter.

mode=3: the program calculates the regularization parameter by applying the GCV analysis to the inversion without positivity.

par, tolc two real numbers that are used differently. Their use depends upon the value of *mode*.

mode=1: the target misfit value is given by the product of *par* and the number of data *N*, i.e., $\phi_d^* = N * par$. The second parameter, *tolc*, is the misfit tolerance. The target misfit is considered to be achieved when the relative difference between the true and target misfits is less than *tolc*. Normally, the value of *par* should be 1.0 if the correct standard deviation of error is assigned to each datum. When 0.0 is entered for *tolc*, the program assumes a default value of *tolc=0.02*.

mode=2: par is the user-input value of regularization parameter. In this case, *tolc* is not used by the program.

mode=3: none of the two input values are used by the program. However, this line of input still needs to be there.

NOTE: When *mode*=1 both *par* and *tolc* are used. When *mode*=2 only *par* is used. When *mode*=3, neither *par* nor *tolc* are used. However, the third line should always have two values.

Le, Ln, Lz length scales in easting, northing, and depth directions respectively. These parameters determine the weighting coefficients $(\alpha_s,\ \alpha_x,\ \alpha_y,\ \alpha_z)$ in the model objective function. The recommended value of the length scale is two to five cell widths in the corresponding direction.

If the input sets *Le=Ln=Lz=0.0*, the inversion will recover a smallest model.

If `NULL` or `null` is entered, the length scale will be equal to two times the maximum cell width at the centre of the mesh. For example, if the cells are 50 m × 50 m × 25m at the centre of the mesh, then the default values are *Le*=*Ln*=*Lz*=100m.

idisk parameter which determines how the sensitivity matrix will be accessed.

=0: sensitivity matrix will be stored in memory. If there is not enough memory, *idisk* will be set to `1` automatically.

=1: sensitivity matrix will be accessed from disk when needed.

Input files:

obs.grv input data file. The file must specify the standard deviations of the error. By definition, these are greater than zero.

gzinv3d.mtx sensitivity matrix and depth weighting function file (calculated by **gzsen3d**).

ini.den initial model stored in the same way as **model.den**. If `null` is entered, a default value determined from the bounds is used. For a constant initial model, enter a value.

ref.den reference model stored in the same way as **model.den**. If `null` is entered, the default value of 0.0 is used. For a constant reference model, enter a value.

bounds.den lower and upper bounds on the recovered density value stored in the formats **bounds.den**. If `null` is entered on this line, the default value of –2.0 and 2.0 are used for all cells. For constant bounds, enter two values for the lower and upper bounds in that order. For example, to construct a model that is bounded between [0, 1.0], this line should be entered *0 1.0*.

w.dat weighting function (optional). If `null` is entered, the program assumes uniform weight of 1.0 .

Output files:

gzinv3d.log the log file containing more detailed information for each iteration and summary of the inversion. See the following **Notes** for a brief explanation.

gzinv3d.den recovered density model.

gzinv3d.pre the predicted data.

gzinv3d.aux auxiliary file listing the data misfit, model norm, and regularization parameter at different stages of the inversion. This is used only for the purpose of restarting the inversion.

gzinv3d.rho temporary file containing the density model produced at different stages of the inversion. This is used only for the purpose of restarting the inversion.

Example of gzinv3d.inp control file: The inversion is started from scratch with a zero reference model. The inversion will try to converge to a target misfit equal to the number of data. The sensitivity matrix will be stored in memory.

```
0                  !! irest
1                  !! mode
1.0  0             !! par, tolc
obs.nois           !! obsf
gzinv3d.mtx        !! mtx file
null               !! initial model
0.0                !! reference model
0.0  1.0           !! lower and upper bounds
100, 100, 100  .   !! alphaS, alphaE, alphaN, alphaZ
null               !! 3D weighting
0                  !! idisk
```

NOTE: The log file *gzinv3d.log* contains more detailed information about the convergence of the inversion. Depending on how the inversion is set up by the users, the content of the log file is slightly different. In general, there are two stages. In the first stage, the program estimates an approximate regularization parameter. In the second stage, the program performs the inversion with bound constraints using a logarithmic barrier method of minimization, which consists of outer iterations with the barrier parameter and inner iterations of conjugate gradient solution of an linear system within each barrier iteration. The log file information is organized according to these two levels iterations.

We can refer to the flow chart Figure 3 in the Section-I to understand the output in the log file. Below we briefly describe the content of the log file according to the parameter *mode* (see Section 1.6) chosen for the inversion.

Mode 1: In this mode, a user-defined target misfit is supplied by specifying *chifact*. The first stage of inversion performs a line search without the bound constraints to estimate an approximate value of the regularization parameter and the slope of the misfit curve. The log file identifies this segment and lists summaries of each linear solution such as the number of conjugate gradient (CG) iterations, data misfit, and model norm. The second stage of inversion carries out a number of logarithmic barrier solutions. Each solution corresponding to a single regularization parameter is obtained by a sequence of barrier iterations, and nested inside each barrier iteration is a set of CG iterations. This portion of the log file begins with the value of regularization parameter, initial values of data misfit and model norm. It is then organized in segments corresponding to barrier iterations and lists the number of CG iterations, data misfit, model norm, barrier parameter, and the value of the barrier function at the conclusion of that iteration. As barrier iterations progress, the data misfit, barrier parameter, and barrier function should decrease monotonically. The model objective function may increase or decrease depending on the nature of the initial model of the logarithmic barrier minimization. However, both data misfit and model objective function should plateau at the end.

One or more solutions may be obtained to complete the line search and to achieve the target misfit. Each solution will have a distinct portion in the log file. The trial values of the regularization are dynamically predicted during the inversion and they are not necessarily in any order. Upon completion of line search and the inversion, the log file will list a summary of the data misfit and model norm corresponding to different values of the regularization parameter sorted in increasing order.

Mode 2: In this mode, the user specifies a value of regularization parameter and the program performs a single logarithmic barrier minimization to obtained the solution. The log file mainly consists of the information for one logarithmic solution as described in *Mode 1*.

Mode 3:. In this mode, the program first performs a GCV estimate of data noise to obtain an approximate value of the regularization parameter, and it then carries out a single logarithmic barrier minimization to obtain the inverse solution. During the first stage, a number of trial values of the regularization parameter is tested, and two linear systems are solved for each value. Thus the first part of the log file is organized according to the value of regularization parameter. For each value, the numbers of CG iterations, data misfit, model norm, and GCV value are listed. At the end of GCV search, the log file lists the data misfit, model norm, GCV value according to the regularization parameter sorted in increasing order. The regularization parameter corresponding to the lowest GCV value is used to obtain the final solution in the second stage. The second part of the log file corresponds to a single logarithmic barrier minimization and it is identical to that when *mode 2* is chosen.

4 Examples

4.1 Introduction

We present two synthetic examples to illustrate the GRAV3D Library. The first model consists of a dipping slab of positive density contrast buried in a uniform background. This is a relatively small model and we use it to illustrate both the basic operation of the library as well as the newly added variable bound constraints in Version 2.0. The second model consists of several blocks of different configurations buried in a uniform background. This model has a relatively large number of observations and the number of cells in the inversion is large. It is used to illustrate the utility of wavelet compression in speeding up the inversion of large data sets. For both examples, we calculate the synthetic data from the true model and then add uncorrelated Gaussian noise to simulate noisy observations.

4.2 Example 1

Figure “*Dipping Dyke: density model*” displays one cross-section and two plan-sections of the true model. This model consists of a dipping dyke in a uniform background. The dyke has a westerly dip of 45° and width of 200 by 300 m in the easting and northing directions. It extends from 50 m to 400 m in depth. The density contrast is 1.0 g/cm^3.

The observations are simulated above the surface at an elevation of 1 m, and over a 21 by 21 grid with a grid interval of 50 m in both directions. The standard deviation of noise added to the data is 2% plus 0.01 mGal. The noisy data are displayed in Figure “*Dipping dyke: Observed data*”. It shows a peak that around (600, 500) m and it decays steeply on the east side but slower on the west side due to the dipping structure of the dyke. The small-scale variations reflect the additive noise in the data.

Ex 1.1 Since the simulated data are entirely positive, it is reasonable to invert for a model consisting of entirely positive density contrast. We accomplish this by imposing a lower bound of 0.0 for positivity and a large upper bound of 4.0, which has no effect on the inversion. The model objective function has the length scales set to 100 m in all three directions, and the simple depth weighting with default parameters is used. The predicted data from this inversion is shown in Figure "*Dipping Dyke: Predicted Gravity Data*", which is a smooth representation of the noisy data. The recovered model is shown in one cross-section and two plan-sections in Figure "Dipping Dyke: Recovered Model-I". The model is characterized by a broad density high at the location corresponding to the true dipping dyke and there is clear indication of a dipping structure. The recovered density has a minimum of 0 as constrained by the lower bound of 0. The maximum value is slightly great than 1.0 g/cmˆ3, which is in fact very close to the true maximum density value in this case. Overall, we have a reasonably good inversion that delineate the essential structure of the true model.

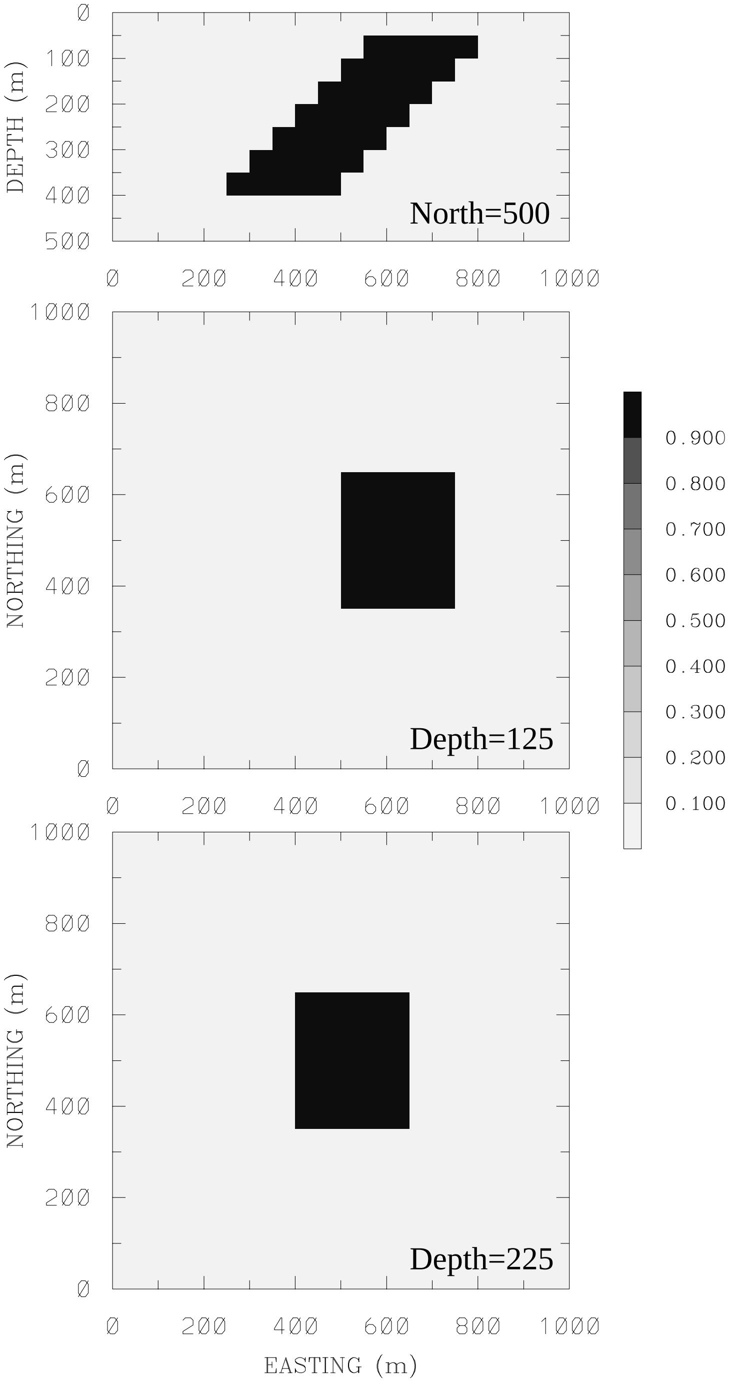
Ex 1.2 Next, we invert the same data (Figure "*Dipping Dyke: Observed Gravity Data*") with a slightly tighter upper bound to illustrate the use of simple upper bound. This is useful when a reliable estimate of maximum density contrast is available. Imposition of such a bound can often improve the solution. For this inversion, we have set the lower and upper bounds to be 0.0 and 0.8. The maximum value of density contrast in the true model is 1.0. The result is shown in Figure "*Dipping dyke: Recovered model-II*", displayed again in one cross-section and two plan-sections. This model is not very different from the one recovered in the preceding inversion, but the anomalous density appears to be slightly wider. This is to be expected since we now have a smaller density contrast and the required anomalous body should have a large dimension to reproduce the same observed anomalies.

Ex 1.3 One of the new features in the GRAV3D Version 2.0 is the ability to impose variable bounds on the density contrast to be recovered in the inversion. This provides users with one more tool for inputting geological information to improve the inversion. For instance, we may expect a lower density contrast in region of an orebody than that in another region. Similarly, one region of the subsurface may have a negative contrast while the rest has a positive contrast. In special cases, imposing a lower and upper bounds that are very close to each other will effective fix the recovered density contrast to a known value during the inversion.

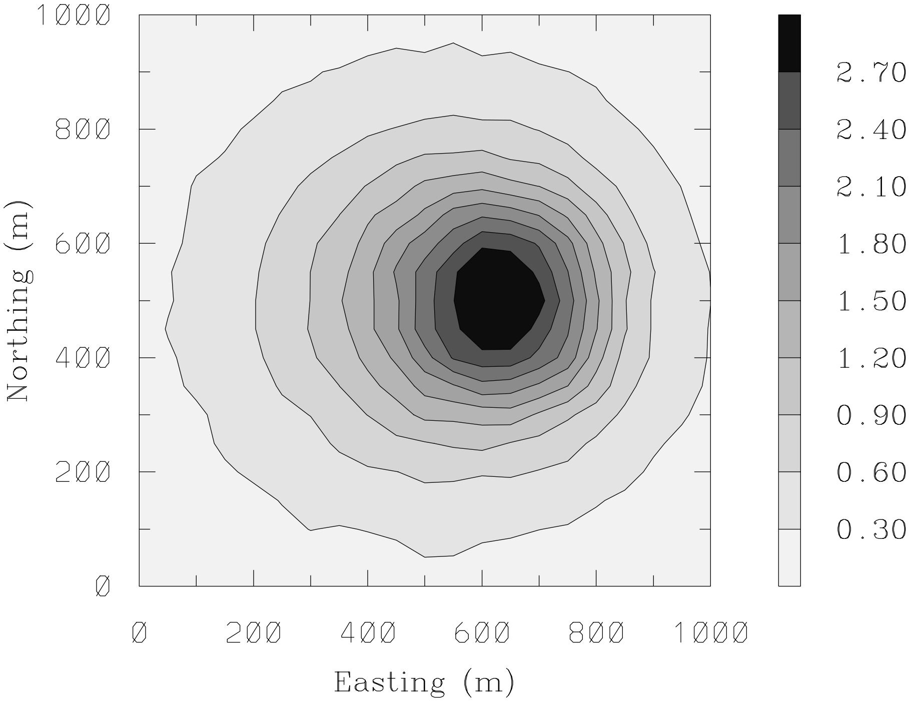
In this example, we illustrate both the variable bounds as well as the use of tight bounds to fix the model values. We invert the data in "*Dipping Dyke: Observed Gravity Data*" by imposing a lower bound of zero constant throughout the model, and a variable upper bounds shown in Figure "*Dipping Dyke: Variable Upper Bounds*". We assume that the top surface of the dipping dyke is known, but we do not know the lateral extent of the dyke nor its depth extent. Therefore, we impose an upper bound of 0.01 above the dyke and an upper bound of 1.0 below that surface. This effectively constrains the density contrast above in the upper region of the entire model to being very close to zero while allowing the density contrast to vary between 0 and be as high as 1.0 g/cm^3 as required by the data.

The recovered model is shown in Figu*re "Dipping Dyke: Recovered Model-III*". The top surface of the dipping dyke is well imaged as expected because of the imposed bound constraints. However, constraints on the top surface of the dyke has greatly helped image the bottom surface of the dyke. Furthermore, the lateral extent of the dyke is well imaged although we have not constrained it at all. This demonstrates the advantages of imposing specific geologic information through variable bounds that help better define the targets.

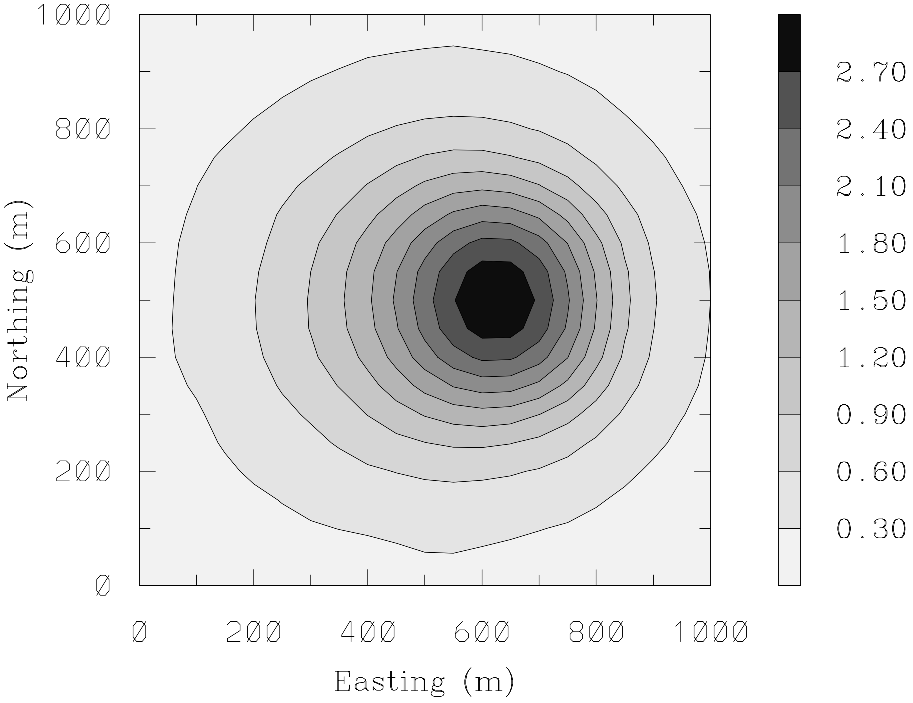
Dipping Dyke: Density Model



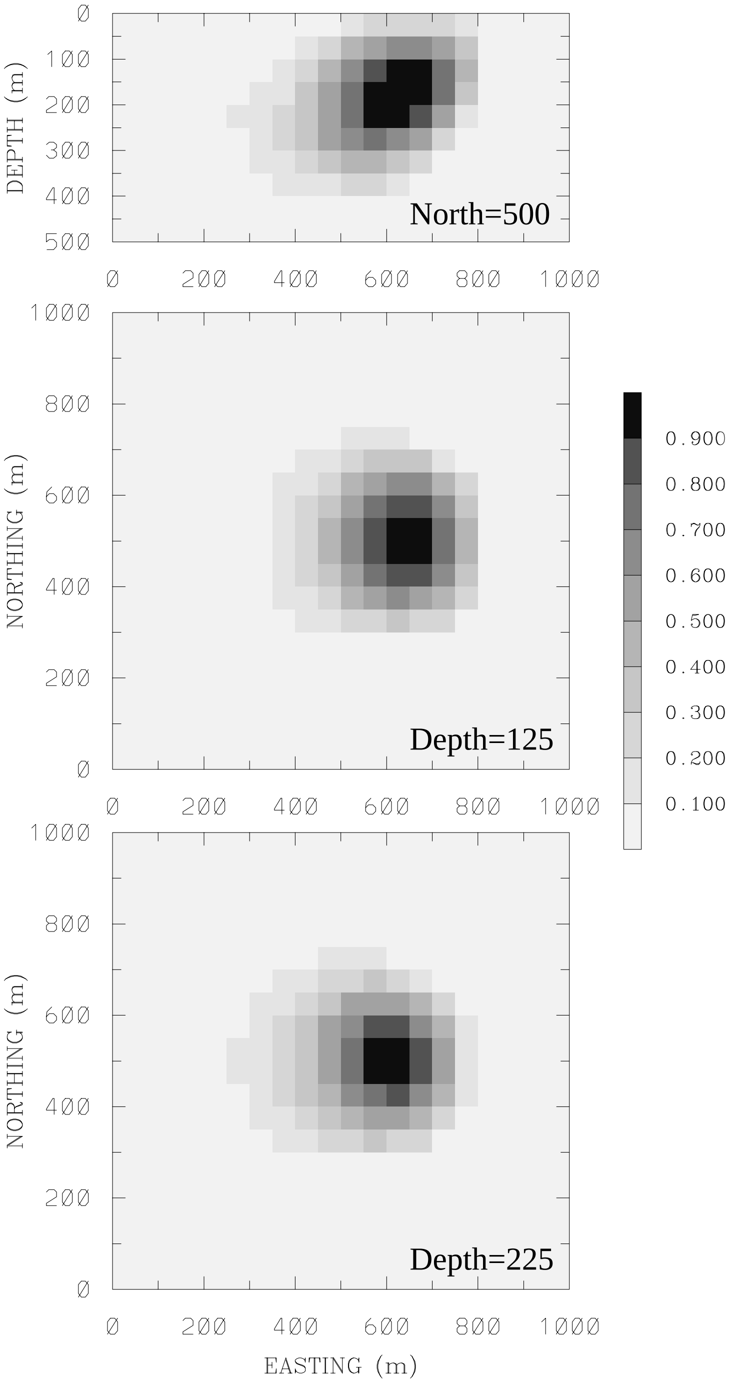
Dipping Dyke: Observed Gravity Data



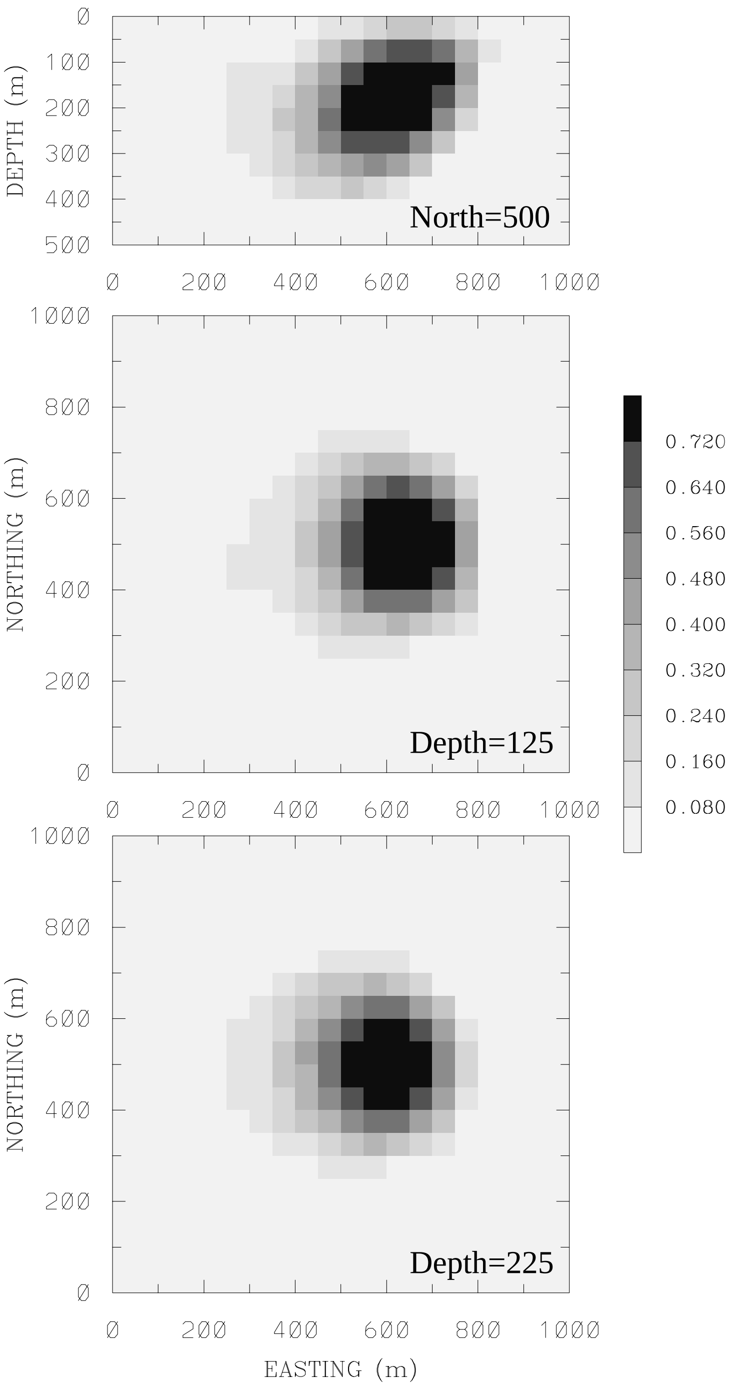
Dipping Dyke: Predicted Gravity Data (Positivity)



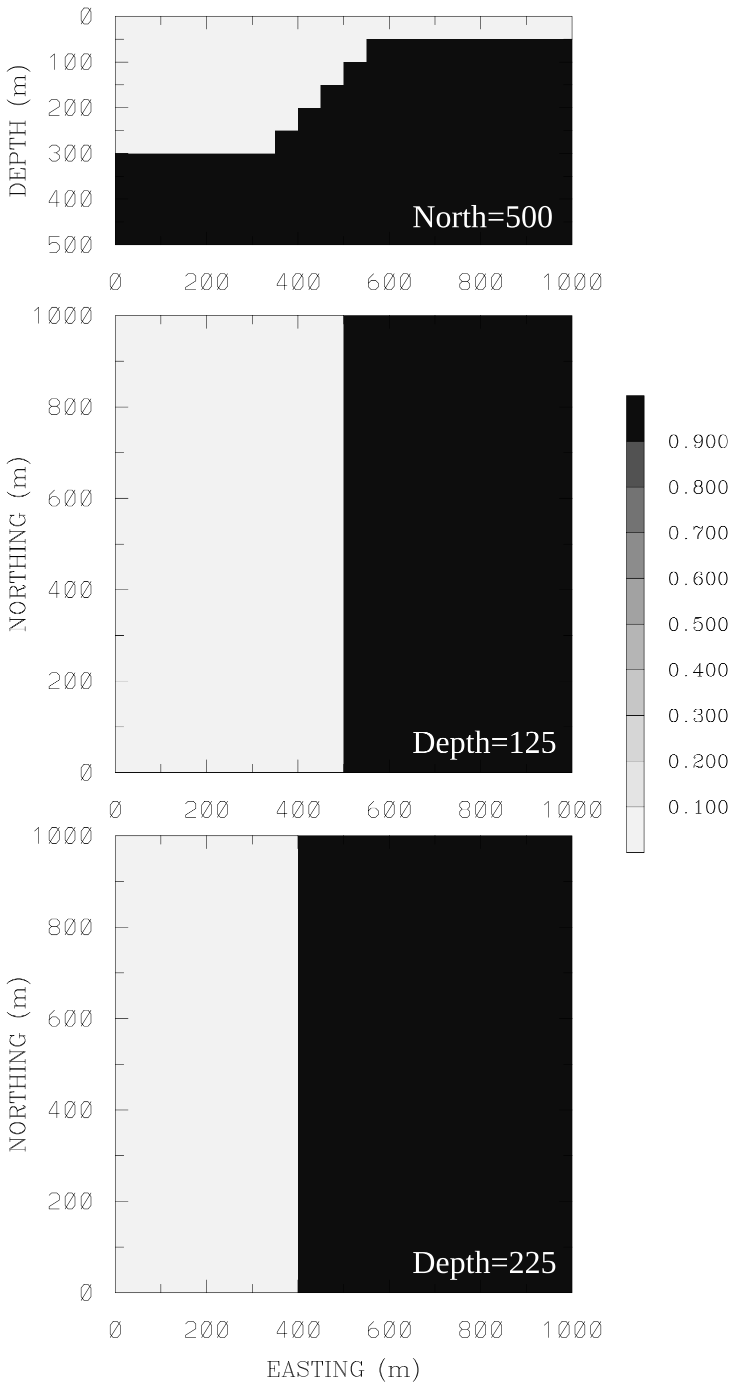
Dipping Dyke: Recovered Model-I (Positivity)



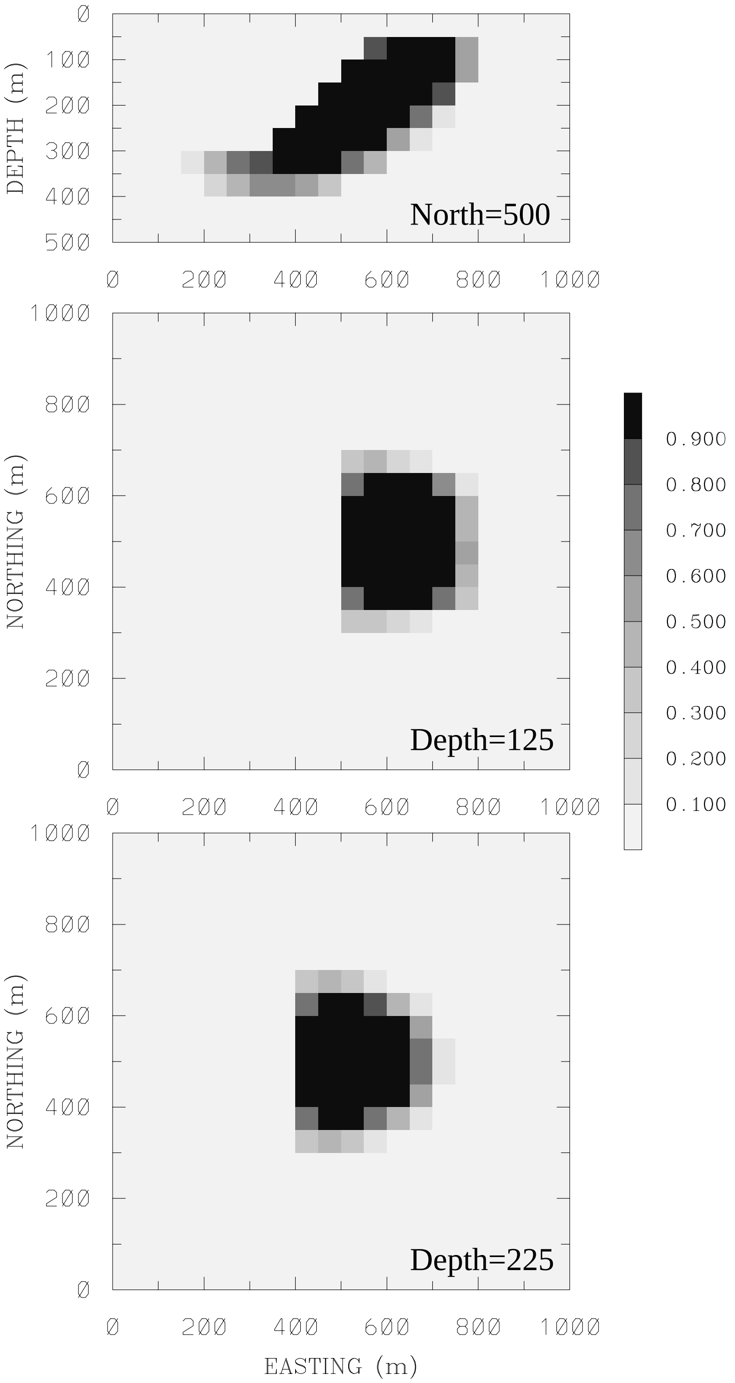
Dipping Dyke: Recovered Model-II
(Lower and Upper Bounds: [0 0.8])



Dipping Dyke: Variable Upper Bounds



Dipping Dyke: Recovered Model-III (Variable Bounds)



4.3 Example 2

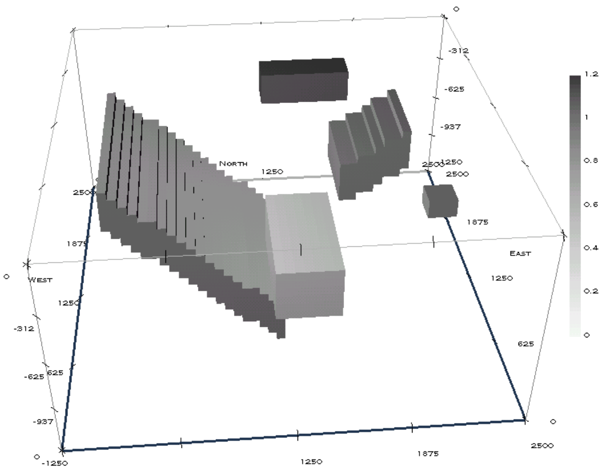
Figure "*Large Model: Perspective View*" displays a volume rendered image of the large test model. It consists of five blocks of different density contrasts in a uniform background. There is one large dipping dyke to the left that extends to a large depth. Four smaller blocks of various shapes are located at shallower depths to the right. Three cross-sections of the model shown in Figure "*Large Model: Cross-sections*" indicate the vertical locations the five blocks. The model occupies a volume of 2.5 km by 2.5 km by 1.25 km.

Figure "*Large Model: Observed Gravity Data*" shows the gravity data simulated on the surface over a 51 by 51 grid of 50-m grid spacing. A total of 2601 data is generated. Noisy observations were simulated by adding Gaussian random noise. The data plot shows four of the five anomalies and the effect of the added noise.

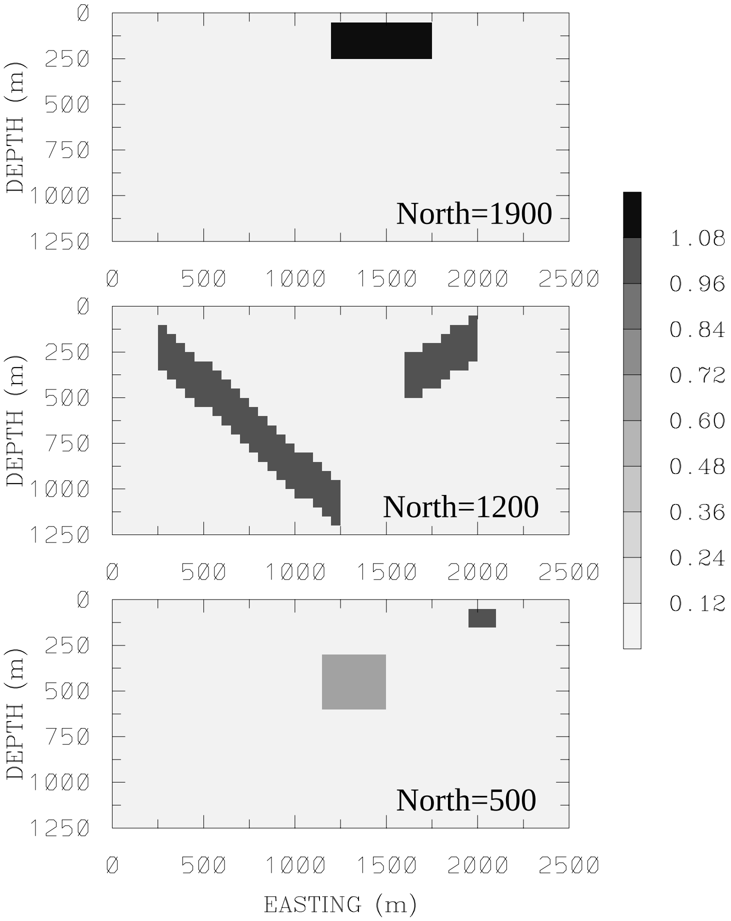
To invert this data set, we discretize the model region using 50-m cubes. This results in 62500 cells and the corresponding sensitivity matrix requires over 600 Mb to store. We use this example to illustrate the wavelet compression of the GRAV3D version 2.0. Using Daubechies-4 wavelet and a reconstruction accuracy of 5%, a compression ratio of 30 was achieved. The resulting matrix is stored in 42 Mb. With the compressed sensitivity matrix, the inversion was carried out readily without much demand on computer memory or CPU time.

The predicted data from the inversion are shown in Figure "*Large Model: Predicted Gravity Data*", which is a smooth version of the observation. The recovered density contrast model is shown in Figure "*Large Model: Recovered Density: Perspective View*". The cutoff value is 0.17 g/cm^3. All five anomalous blocks are imaged. The Recovered model is shown in three cross-sections in Figure "*Large Model: Recovered Density: Cross-sections*". The amplitudes of the recovered anomalous blocks are lower than the true value. The depth extent of the large dipping block is also smaller. This is expected since the area of the data is limited. Over all, this is a good representation of the true model, and the inversion utilizing the wavelet compression has allowed the inversion to be carried out with very moderate demand on computational resources.

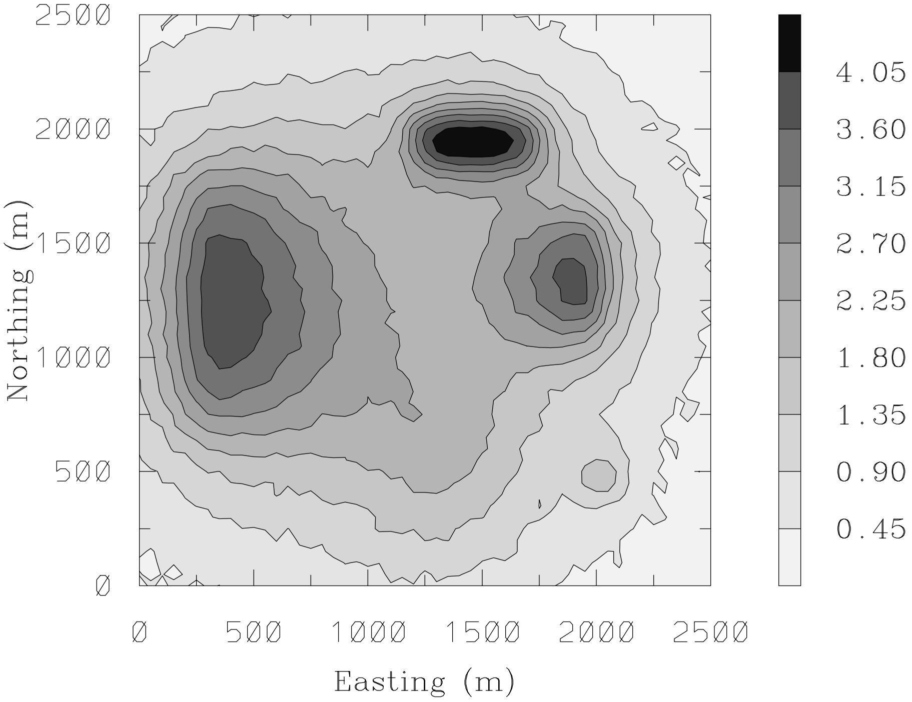
Large Model: Pespective View



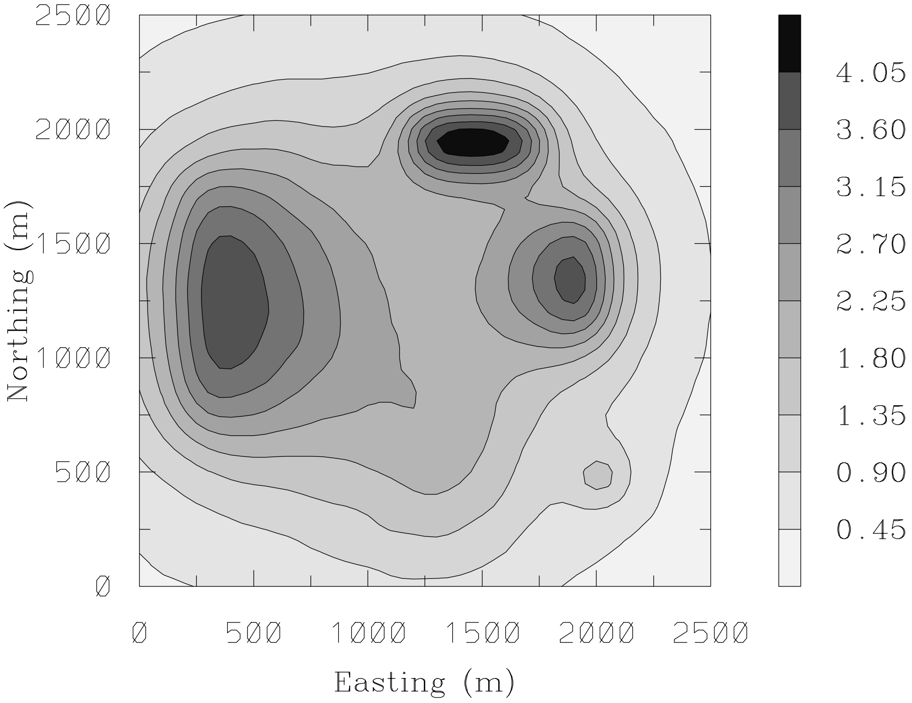
Large Model: Cross-sections



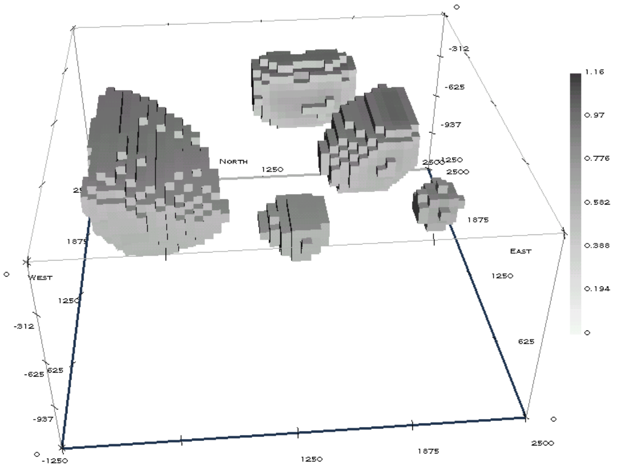
Large Model: Observed Gravity Data



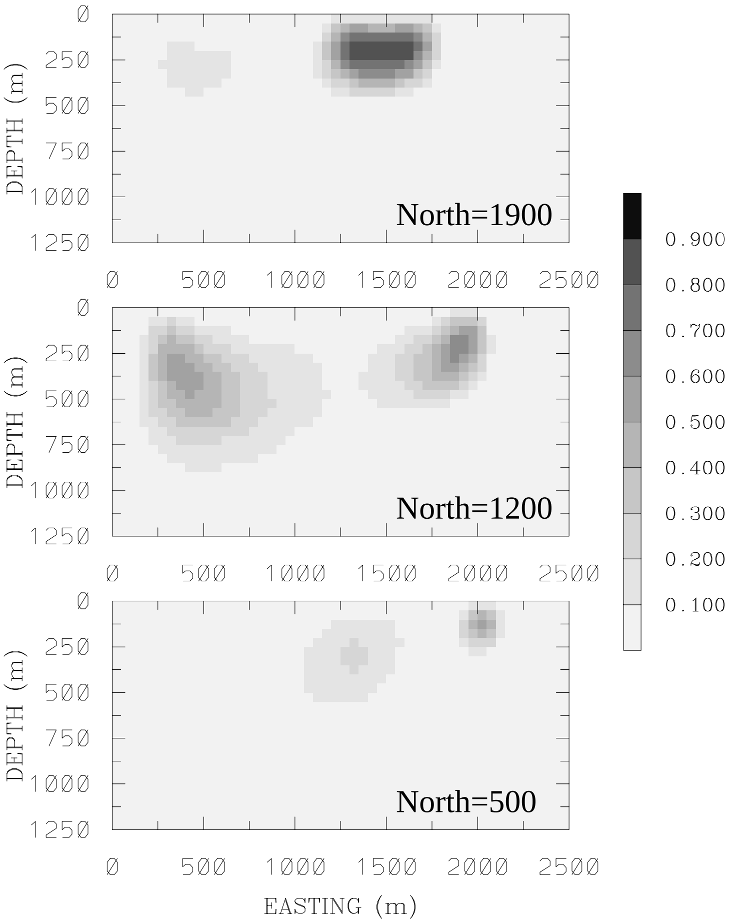
Large Model: Predicted Gravity Data



Large Model: Recovered Density (Cutoff at 0.17g/cm^3)



Large Model: Recovered Density (Cross-sections)


DEPTH (m)
Ø
25Ø
5ØØ
75Ø
1ØØØ
125Ø
North=1900
North=1200
North=500
Ø
5ØØ
1ØØØ
15ØØ
2ØØØ
25ØØ
EASTING (m)