

---

# **GRAV3D Documentation**

*Release 6.0*

**UBC-GIF**

November 09, 2016



|          |   |           |
|----------|---|-----------|
| <b>1</b> | <b>GRAV3D package overview</b>                      | <b>3</b>  |
| 1.1      | Description . . . . .                               | 3         |
| 1.2      | Program library content . . . . .                   | 4         |
| 1.3      | Licensing . . . . .                                 | 4         |
| 1.4      | Installing . . . . .                                | 4         |
| 1.5      | Highlights of changes from version 5.0 . . . . .    | 5         |
| <b>2</b> | <b>Background theory</b>                            | <b>7</b>  |
| 2.1      | Introduction . . . . .                              | 7         |
| 2.2      | Forward Modelling . . . . .                         | 7         |
| 2.3      | Inversion methodology . . . . .                     | 8         |
| 2.4      | Depth Weighting and Distance Weighting . . . . .    | 10        |
| 2.5      | Wavelet Compression of Sensitivity Matrix . . . . . | 11        |
| <b>3</b> | <b>Elements of the program GRAV3D</b>               | <b>13</b> |
| 3.1      | Introduction . . . . .                              | 13        |
| 3.2      | General files for GRAV3D programs . . . . .         | 13        |
| <b>4</b> | <b>Running the programs</b>                         | <b>21</b> |
| 4.1      | Introduction . . . . .                              | 21        |
| 4.2      | Input and output files . . . . .                    | 23        |
| <b>5</b> | <b>References</b>                                   | <b>33</b> |
|          | <b>Bibliography</b>                                 | <b>35</b> |



GRAV3D v6.0 is a program library for carrying out forward modelling and inversion of surface, borehole, and airborne gravity data in 3D. The contents of this manual are as follows:



---

## GRAV3D package overview

---

### 1.1 Description

GRAV3D is a program library for carrying out forward modelling and inversion of surface and airborne gravity data over 3D structures. The program library carries out the following functions:

1. Forward modelling of the vertical component of the gravity response to a 3D volume of density contrast.
2. The model is specified in the mesh of rectangular cells, each with a constant value of density contrast. Topography is included in the mesh. The vertical gravity 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.
  - 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 a priori 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 either 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.
3. 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
- **GZFOR3D**: performs forward modelling
- **GZSEN3D**: calculates the sensitivity matrix
- **GZPRE3D**: multiplies the sensitivity file by the model to get the predicted data
- **GZINV3D**: performs 3D gravity 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 GRAV3D 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 the . 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 5.0

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 the previous 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 be 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 `gzinv3d.log` file has been simplified and detailed information on the inversion can be found in the `gzinv3d.out` file.
7. 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.

### 1.5.1 Notes on computation speed

- For large problems, GZSEN3D 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 and . 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 5.0 is primarily that the projected gradient results in a significantly faster solution than the logarithmic barrier technique used in earlier versions.



---

## Background theory

---

### 2.1 Introduction

The GRAV3D suite of algorithms, developed at the UBC Geophysical Inversion Facility, is used to invert gravimetric responses over a three dimensional distribution of density contrast, or anomalous density. This 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 [LO98] 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 to its fullest extent.

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 anomalous density  $\rho(x, y, z)$ , produces its own gravity field,  $\mathbf{g}_s$ , 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

$$g_z(\mathbf{r}_o) = \gamma \int_V \rho(\mathbf{r}) \frac{z - z_o}{|\mathbf{r} - \mathbf{r}_o|^3} dv, \quad (2.1)$$

where  $\mathbf{r}_o = (x_o, y_o, z_o)$  is the vector denoting the observation location and  $\mathbf{r} = (x, y, z)$  is the source location. The volume of the anomalous mass is  $V$  and  $\gamma$  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. In the following, we outline the basics of the forward and inverse procedures used by the GRAV3D program library.

### 2.2 Forward Modelling

Forward modelling of gravity data is a linear problem and can be carried out by performing the integration in equation [eq:gzfield]. We divide the region of interest into a set of 3D prismatic cells by using a 3D orthogonal mesh and assume a constant density contrast within each cell. We discretize the density contrast 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:

$$\begin{aligned}
 d_i &\equiv g_z(\mathbf{r}_{oi}) \\
 &= \sum_{j=1}^M \rho_j \left\{ \gamma \int_{\Delta V_j} \frac{z - z_o}{|\mathbf{r} - \mathbf{r}_{oi}|^3} dv \right\}, \\
 &\equiv \sum_{j=0}^M \rho_j G_{ij}.
 \end{aligned} \tag{2.2}$$

In equation (2.2),  $\rho_j$  and  $\Delta V_j$  are the anomalous density and volume of the  $j^{\text{th}}$  cell,  $d_i$  is introduced as a generic symbol for the  $i^{\text{th}}$  datum, and  $G_{ij}$ , defined by the expression in brackets, quantifies the contribution of the  $j^{\text{th}}$  cell to the  $i^{\text{th}}$  datum. The solution for the integral in equation (2.2) can be found in [Nag66] and we have adopted the solution by [Haa53] here.

## 2.3 Inversion methodology

Let the set of extracted anomaly data be  $\mathbf{d} = (d_1, d_2, \dots, d_N)^T$  and the density contrast of cells in the model be  $\rho = (\rho_1, \rho_2, \dots, \rho_M)^T$ . The two are related by the sensitivity matrix

$$\mathbf{d} = \mathbf{G}\rho. \tag{2.3}$$

The matrix has elements  $g_{ij}$  which quantify the contribution to the  $i^{\text{th}}$  datum due to a unit density in the  $j^{\text{th}}$  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.

For the inversion, the first question that arises concerns definition of the ‘‘model’’. We choose density contrast,  $\rho$ , as the model for since the anomalous field is directly proportional to the density contrast. The inverse problem is formulated as an optimization problem where a global objective function,  $\phi$ , is minimized subject to the constraints in equation (2.3). The global objective functions consists of two components: a model objective function,  $\phi_m$ , and a data misfit function,  $\phi_d$ , such that

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

where  $\beta$  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  $\phi_d$  and we will search for the value of  $\beta$  via an L-curve criterion [Han00] that produces the expected misfit. Otherwise, a user-defined value is used. Bound are imposed through the projected gradient method so that the recovered model lies between imposed lower ( $\rho^l$ ) and upper ( $\rho^u$ ) bounds.

We next discuss the construction of a model objective function which, when minimized, produces a model that is geophysically interpretable. The objective function gives the flexibility to incorporate as little or as much information as possible. At the very minimum, this function drives the solution towards a reference model  $\rho_o$  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

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

where the functions  $w_s, w_x, w_y$  and  $w_z$  are spatially dependent, while  $\alpha_s, \alpha_x, \alpha_y$  and  $\alpha_z$  are coefficients, which affect the relative importance of different components in the objective function. The reference model is given as  $\rho_o$  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 density contrast is not concentrated near the observation locations. The details of the depth weighting function will be discussed in the next section.

The objective function in equation (2.3) 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  $\rho_o$  in the derivative terms can have significant effect on the recovered model as shown through the synthetic example. 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 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 (2.3) is discretized onto the mesh defining the density contrast model using a finite difference approximation. This yields:

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

where  $\rho$  and  $\rho_o$  are  $M$ -length vectors representing the recovered and reference models, respectively. Similarly, there is an option to remove to the reference model from the spatial derivatives in equation (2.3) such that

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

In the previous two equations, the individual matrices  $\mathbf{W}_s, \mathbf{W}_x, \mathbf{W}_y$ , and  $\mathbf{W}_z$  are straight-forwardly 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.

In addition, more realistic geologic-looking models can often be obtained by introducing various levels of sharpness and compactness into the inverted results. This is accomplished by using different norms in the model objective function. Minimizing an  $l_p$ -norm of a model, as  $p$  reduces from two to zero, generates a result that goes from smooth to blocky to compact. Our generalized norms, which we refer to as  $l_p, l_q$ , allow various degrees of smoothness and compactness on different components of the model objective function:

$$\phi_m(\rho) = \|\mathbf{W}_s (\rho - \rho_o)\|^p + \|\mathbf{W}_i(\rho)\|^{q_i} \quad (i = x, y, z) \quad (2.4)$$

The above equation is solved through an iteratively re-weighted least-squares (IRLS) approach. The inversion is solved to the  $l_2$  measure and then the model objective function is changed. The  $p$  norm promotes sparseness through the model. The  $q_i$  norms promotes blockiness (or smoothness for  $q_i = 2$ ) in each principal direction. In equation (2.4), the depth/distance weighting is absorbed into the  $\mathbf{W}$  matrices.

The next step in setting up the inversion is to define a misfit measure. Here we use the  $l_2$ -norm measure

$$\phi_d = \|\mathbf{W}_d (\mathbf{G}\rho - \mathbf{d})\|^2. \quad (2.5)$$

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  $\sigma_i$  is the standard deviation of the  $i^{th}$  datum

makes  $\phi_d$  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 (2.3).

To solve the optimization problem when constraints are imposed we use the projected gradients method [CM87][Vog02]. 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 used the logarithmic barrier method [Wri97][NW99].

The weighting function is generated by the program that is in turn given as input to the sensitivity generation program . 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 is required to be used when borehole data are present.

## 2.4 Depth Weighting and Distance Weighting

It is a well-known fact that vertical gravity 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 density contrast 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 gravity data are not sufficient to generate a function, which 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 give cells at all locations equal probability to enter into the solution with a non-zero density contrast.

### 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  $(z + z_o)^{-2}$  closely approximates the kernel's decay directly under the observation point provided that a reasonable value is chosen for  $z_o$ . The value of 2 in the exponent is consistent with the fact that, to first order, a cuboidal cell acts like a dipole source whose magnetic field decays as inverse distance cubed. The value of  $z_o$  can be obtained by matching the function  $1/(z + z_o)^2$  with the field produced at an observation point by a column of cells. Thus we use a depth weighting function of the form

$$w(\mathbf{r}_j) = \left[ \frac{1}{\Delta z_j} \int_{\Delta z_{ij}} \frac{dz}{(z + z_o)^\alpha} \right]^{1/2}, \quad j = 1, \dots, M. \quad (2.6)$$

For the inversion of surface data, where  $\alpha = 2$ ,  $\mathbf{r}_j$  is used to identify the  $j^{th}$  cell, and  $\Delta z_j$  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 [eq:mof], 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:

$$w(\mathbf{r}_j) = \frac{1}{\sqrt{\Delta V_j}} \left\{ \sum_{i=1}^N \left[ \int_{\Delta V_j} \frac{dv}{(R_{ij} + R_o)^\alpha} \right]^2 \right\}^{1/4}, \quad j = 1, \dots, M, \quad (2.7)$$

where  $\alpha = 2$ ,  $V_j$  is the volume of  $j^{\text{th}}$  cell,  $R_{ij}$  is the distance between a point within the source volume and the  $i^{\text{th}}$  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. The 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 [LO03][LO10]. 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 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:

$$\tilde{\mathbf{G}} = \mathbf{G}\mathcal{W}^T, \quad (2.8)$$

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  $\tilde{\mathbf{G}}^S$ ,

$$\tilde{g}_{ij}^s = \begin{cases} \tilde{g}_{ij} & \text{if } |\tilde{g}_{ij}| \geq \delta_i \\ 0 & \text{if } |\tilde{g}_{ij}| < \delta_i \end{cases}, \quad i = 1, \dots, N, \quad (2.9)$$

where  $\delta_i$  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  $\delta_i$  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:

$$r_i(\delta_i) = \sqrt{\frac{\sum_{|\tilde{g}_{ij}| < \delta_i} \tilde{g}_{ij}^2}{\sum_j \tilde{g}_{ij}^2}}, \quad i = 1, \dots, N, \quad (2.10)$$

Here the numerator is the norm of the discarded coefficients and the denominator is the norm of all coefficients. The threshold level  $\delta_{i_o}$  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

$$\delta_i = \epsilon \max_j |\tilde{g}_{ij}|, \quad i = 1, \dots, N. \quad (2.11)$$

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  $\delta_i$ . 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 and ones that are re-inverting the data, the program also allows the direct input of the relative threshold level.

---

## Elements of the program GRAV3D

---

### 3.1 Introduction

The program library consists of the programs:

1. **GZFOR3D**: performs forward modelling.
2. **PFWEIGHT**: calculates depth or distance weighting function.
3. **GZSEN3D**: calculates sensitivity for the inversion.
4. **GZINV3D**: performs 3D gravity inversion.
5. **GZPRE3D**: 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 GRAV3D programs

There are seven general files which are used in GRAV3D. 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*: Number of cells in the East direction.
- *NN*: Number of cells in the North direction
- *NZ*: Number of cells in the vertical direction

|              |              |       |                 |
|--------------|--------------|-------|-----------------|
| NE           | NN           | NZ    |                 |
| $E_o$        | $N_o$        | $Z_o$ |                 |
| $\Delta E_1$ | $\Delta E_2$ | ...   | $\Delta E_{NE}$ |
| $\Delta N_1$ | $\Delta N_2$ | ...   | $\Delta N_{NN}$ |
| $\Delta Z_1$ | $\Delta Z_2$ | ...   | $\Delta Z_{NZ}$ |

- $E_o, N_o, Z_o$ : 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.
- $\Delta E_n$ :  $n^{th}$  cell width in the easting direction (ordered W to E).
- $\Delta N_n$ :  $n^{th}$  cell width in the northing direction (ordered S to N).
- $\Delta Z_n$ :  $n^{th}$  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.

|         |      |     |         |    |     |     |
|---------|------|-----|---------|----|-----|-----|
| 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     |    |     |     |

### 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:

|                |                |                   |
|----------------|----------------|-------------------|
| !              | comment        |                   |
| npt            |                |                   |
| E <sub>1</sub> | N <sub>1</sub> | ELEV <sub>1</sub> |
| E <sub>2</sub> | N <sub>2</sub> | ELEV <sub>2</sub> |
| ⋮              | ⋮              | ⋮                 |
| E <sub>n</sub> | N <sub>n</sub> | ELEV <sub>n</sub> |

Parameter definitions:

- npt: Number of points defining the topographic surface.
- E<sub>*i*</sub>: Easting of the *i*<sup>th</sup> point on the surface.
- N<sub>*i*</sub>: Northing of the *i*<sup>th</sup> point on the surface.
- ELEV<sub>*i*</sub>: Elevation (metres) of the *i*<sup>th</sup> 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 Observations file

This file is used to specify the observed gravity anomalies with estimated standard deviation. The output of the forward modelling program GZFOR3D has the same structure except that the column of standard deviations for the error is omitted. Lines starting with ! are comments. The following is the GIF-formatted file structure of a gravity observations file:

Parameter definitions:

- ndat: Number of observations.
- 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.

```

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
ndat
E1   N1   ELEV1  Grav1  Err1
E2   N2   ELEV2  Grav2  Err2
⋮     ⋮     ⋮       ⋮       ⋮
Endat Nndat ELEVndat Gravndat Errndat

```

- Grav<sub>*i*</sub>: Anomalous gravity of *i*th datum measured in mGal.
- Err<sub>*i*</sub>: Standard deviation of Grav<sub>*n*</sub>. 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. Furthermore, the inversion program assumes that the anomalies are produced by a density contrast distribution in g/cm<sup>3</sup> with mesh cells in meters. Therefore, it is crucial that the data be prepared in mGal.

### Predicted data file

The predicted data file is the exact same format as above, but omitting the uncertainty column. The forward modelling and inversion code will output the predicted data in this format.

### Locations file

The locations file is the exact same format as above, but omitting the gravity data and uncertainty columns. The forward modelling code will read in locations even when the gravity anomaly (and uncertainties) are given.

### Example

#### 3.2.4 Model file

This file contains the cell property values (g/cc) 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

```

! surface data
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

```

```

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

```

Each  $(m_{i,j,k})$  is the property in the  $(i,j,k)^{\text{th}}$  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 GRAV3D uses five general codes:

- GZFOR3D: performs forward modelling.
- PFWEIGHT: calculates the depth weighting function.
- GZSEN3D: calculates sensitivity.
- GZINV3D: performs 3D gravity inversion.
- GZPRE3D: 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 GRAV3D 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 gzinv3d
```

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:

|           |    |
|-----------|----|
| machine01 | 16 |
| machine02 | 16 |

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 GZFOR3D

This program performs forward modelling. Command line usage: `gzfor3d mesh.msh obs.loc model.den [topo.dat]` and will create the forward modelled data file `gzfor3d.grv`. **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 GRAV3D*. The files associated with GZFOR3D are:

- `mesh.msh`: The 3D *mesh*.
- `obs.loc`: The observation *locations*.
- `model.den`: The density contrast *model* in *g/cc*.
- `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 `gzfor3d.grv`. The file format is that of the *data file* without the associated standard deviations. The forward modelled data are in **mGal**.

### 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 GRAV to have `pfweight` read gravity files.
- `mesh.msh`: Name of 3D *mesh file*.
- `obs.grv`: The *data file* that contains the observation locations and the observed vertical gravity 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.

```
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
```

- **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  $\alpha$  and  $z_0$  to define the *depth weighting*

When `iwt=2`, `alpha` and `znot` are used as  $\alpha$  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  $\alpha$  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  $\alpha$  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

```
GRAV          ! gravity data
mesh.msh      ! mesh file
obs_noise.grv ! 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* ( $x = \text{“depth”}$ ) or *distance weighting* ( $x = \text{“distance”}$ ). A log file `pfweight.log` is also written.

### 4.2.3 GZSEN3D

This program performs the sensitivity calculation. Command line usage:

```
gzsen3d gzsen3d.inp [nThreads]
```

For a sample input file type:

```
gzsen3d -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)
Scale        ! Scale sensitivity (1= L2 norm | 0= Lp/L1 norm)
```

The input parameters for the control file are:

- `mesh.msh`: Name of 3D *mesh file*.
- `obs.grv`: 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.dat`: Surface *topography*. If `null` is entered, the surface will be treated as being flat on top of the mesh.
- `distance_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”.
- `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).
- **Scale**: Option to scale the sensitivities (`Scale=1`) or the model objective function (`Scale=0`) by the depth weighting function. This value **must be 0** to scale the sensitivities through the model objective function when the **compact or blocky model norms** are chosen. If scaling the sensitivity is chosen, the inversion code will output an error at the beginning. This value is optional and will be set to `Scale=0` if the line is not given.

**NOTE**: Empirical testing has shown that the results are consistent with either option. Setting `scale=0` will allow the user to run both a smooth and blocky inversion without having to re-compute the sensitivities.

#### Example of input file

```
mesh.msh      ! mesh file
obs.grv       ! 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)
0             ! Scale sensitivity (1= L2 norm | 0= Lp/L1 norm)
```

#### Output files

The program `gzsen3d` outputs three files. They are:

1. `gzinv3d.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.
2. `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.
3. Diagnostic files to examine the wavelet compression properties, if chosen (`Diag=1`).

#### 4.2.4 GZINV3D

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

```
gzinv3d gzinv3d.inp [nThreads]
```

For a sample input file type:

```
gzinv3d -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.grv`: Mandatory *observations file*.
2. `gzinv3d.mtx`: Mandatory sensitivity matrix from *GZSEN3D*
3. `initial.den`: Optional initial *model file*. This can be substituted by a value within the input file (see below).
4. `ref.den`: 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.den`: Optional upper bounds *model file*. A value can be used to set a global bound (see below).
7. `lowerBound.den`: Optional lower bounds *model file*. A value can be used to set a global bound (see below).
8. `weights.wt`: Optional *weighting file*.
9. `gzinv3d.inp`: The control file containing the options. Does not need to be specifically called “gzinv3d.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:

The parameters within the control file are:

- `mode`: An integer specifying one of three 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.
  3. `mode=3`: the program calculates the trade off parameter by applying the GCV analysis to the inversion without positivity.
- `par, tolC` 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, `tolC`, 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 `tolC`. Normally, `par=1` is ideal if the true standard

```

mode                ! 1 (Find beta), 2 (Fixed beta)
par, tolc           ! beta parameter and tolerance
obs                 ! Observed data
matrixFile          ! Sensitivity matrix
init                ! file | VALUE [g/cc]
ref                 ! file | VALUE [g/cc]
act                 ! file | null
lowerBounds         ! file | VALUE [bl] | file
upperBounds         ! file | 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.wt           ! Weighting file | null
VALUE P Qx Qy Qz    ! Lp/Lq exponents | null
scale, eps, epsGrad ! Lq scaling (0-1), zeroS, zeroXYZ | null
null                ! Place-holder

```

deviation of error is assigned to each datum. When `tolc=0`, the program assumes a default value of `tolc=0.02` since this number must be positive.

2. `mode=2`: `par` is the user-input value of trade off parameter. In this case, `tolc` is not used by the program.
3. `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 both `par` and `tolc` are used. When only `par` is used. When `mode=3`, neither nor `tolc` are used. However, the third line should always have two values.

- `obs.grv`: Input data file. The file must specify the standard deviations of the error. By definition these values are greater than zero.
- `gzinv3d.mtx`: The binary file of sensitivities created by *GZSEN3D*.
- `initial.den`: The initial density contrast model 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.den`: The reference density contrast model can be defined as a value for uniform models (e.g. `VALUE 0`), or by a filename (for non-uniform reference models).
- `active.txt`: The *active model file* defining which cells in the model are allowed to be solved.
- `lowerBound.den`: The lower bounds model can be defined as a value for uniform models (e.g., `VALUE -1`) or by a filename.
- `upperBound.den`: The upper bounds model 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.  $\alpha_s$  is the smallest model component. Coefficient for the derivative in the easting direction.  $\alpha_y$  is the coefficient for the derivative in the northing direction. The coefficient  $\alpha_z$  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  $\alpha_z/\alpha_s$ . 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  $\alpha$ 's to length scales can be done by:

$$L_x = \sqrt{\frac{\alpha_x}{\alpha_s}}; L_y = \sqrt{\frac{\alpha_y}{\alpha_s}}; L_z = \sqrt{\frac{\alpha_z}{\alpha_s}} \quad (4.1)$$

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 the *model object function* section for details.
- `weights.wt`: Name of the *weights file* containing weighting matrices. If `null` is entered, default values of unity are used (no extra weighting).
- `VALUE P Qx Qy Qz`: The  $L_p/L_q$  exponents for the *model objective function*. The  $P$  is for the smallest model component and the  $Q_s$  are for the spatial components. This line is optional and the  $L_2$  norm will be assumed for inputs of `null` or if the file ends on the previous line.
- `scale, eps, epsGrad`: The scaling between  $L_p$  and  $L_q$  components in range  $[0, 1]$ . `eps` is an effective zero for the model values. `epsGrad` is an effective zero value for the change in model values spatially (i.e., derivatives). The program will calculate these zeros based on a single standard deviation of the  $L_2$  model if `null` is given with no extra scaling between  $L_p$  and  $L_q$  (`scale = 0.5`). **Note:** This line is optional and is only required if the  $L_p/L_q$  constants are given or MOF derivatives below.
- `mof.wt`: This input is currently disabled because of the upgrade to the model objective function. Use `null` or end the file prematurely.

### Example of control file

Below is an example of a control file with comments for an  $L_2$  inversion.

Below is an example of an input file for an  $L_p/L_q$  file. The program will try to find a sparse model ( $L_p=0$ ) with smooth sides ( $L_qx=L_qy=L_qz=2$ ) and fit the data misfit to within 5% of the desired misfit.

### Output files

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

1. `gzinv3d.log`: The log file containing the minimum information for each iteration and summary of the inversion.
2. `gzinv3d.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. `gzinv3d_xxx.den`: Density contrast *model files* output after each “xxx” iteration (i.e., `gzinv3d_012.den`)

```
1           ! mode
1.0 0.0     ! par, tolc
obs_noise.grv ! observation file
gzinv3d.mtx ! sensitivity matrix file
VALUE 0.0   ! initial model
VALUE 0.0   ! reference model
NULL        ! active cells file
VALUE -2.0  ! lower bound
upperBounds.den ! upper bound file
null        !  $\alpha_s, \alpha_x, \alpha_y, \alpha_z$ 
SMOOTH_MOD_DIF ! reference model in the derivative terms
null        ! weighting matrix
```

```
1           ! Find beta
1, 0.05     ! phid and tolerance
obs.grv     ! Observed data
gzinv3d.mtx ! Sensitivity matrix
VALUE 0.001 ! initial model
VALUE 0     ! reference model
null        ! all cells active
VALUE -2    ! lower bound
VALUE 2     ! upper bound
125 125 125 ! length scales
SMOOTH_MOD  ! no ref in XYZ derivatives
null        ! no weighting
VALUE 0 2 2 2 ! Lp/Lq exponents
null        ! let GZINV3D scale
null        ! place holder
```

4. `gzinv3d_XXX.pre`: *Predicted data files* (without uncertainties) output after each “XXX” iteration.

### 4.2.5 GZPRE3D

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:

```
gzpre3d gzinv3d.mtx obs.loc model.den
```

#### Input files

1. `gzinv3d.mtx`: The sensitivity matrix file create from *GZSEN3D*.
2. `obs.loc`: The gravity *location file*.
3. `model.den`: The density contrast *model file*.

#### Output file

The output file is a *predicted data file* (omitting uncertainty column) and is named `gzpre3d.grv`. This program can be used to reproduce output predicted files from *GZINV3D*.



---

**References**

---



- [CM87] P H Calamai and J J Moré. Projected gradient methods for linearly constrained problems. *Mathematical programming*, 39:96–116, 1987.
- [Han00] P C Hansen. *The L-curve and its use in the numerical treatment of inverse problems*. WIT Press, Southampton, 2000.
- [Haa53] I B Haáz. Relations between the potential of the attraction of the mass contained in a finite rectangular prism and its first and second derivatives. *Geofizikai Közlemenyek*, 2(7):57–66, 1953. in Hungarian.
- [LO98] Y Li and D W Oldenburg. 3-D inversion of gravity data. *Geophysics*, 63:109–119, 1998.
- [LO03] Y Li and D W Oldenburg. Fast inversion of large-scale magnetic data using wavelet transforms and a logarithmic barrier method. *Geophysics Journal International*, 152:251–265, 2003.
- [LO10] Y Li and D W Oldenburg. Rapid construction of equivalent sources using wavelets. *Geophysics*, 75:L51–L59, 2010.
- [Nag66] D Nagy. The gravitational attraction of a right rectangular prism. *Geophysics*, 31:361–371, 1966.
- [NW99] J Nocedal and S J Wright. *Numerical optimization*. Springer Science, New York, NY, 1999.
- [Vog02] Curtis R Vogel. *Computational Methods for Inverse Problems (Frontiers in Applied Mathematics)*. Society for Industrial Mathematics, 2002. URL: <http://www.amazon.ca/exec/obidos/redirect?tag=citeulike09-20&path=ASIN/0898715504>.
- [Wri97] S J Wright. *Primal-Dual Interior-Point Methods*. SIAM, Philadelphia, PA, 1997.