DCIP3D

A Program Library for Forward Modelling and Inversion of DC/IP Data over 3D Structures

Version 5.0

Developed under the consortium research project:

INVERSION OF 3D DC RESISTIVITY AND
INDUCED POLARIZATION DATA

Release date: May 2014

UBC - Geophysical Inversion Facility 1988 – 2014

Table of Contents

1 Package overview

1.1 Description

DCIP3D v5.0 is a program library that performs forward modelling and inversion of DC resistivity and IP data over a 3D distribution of electrical conductivity and chargeability. Arbitrary transmitter and receiver electrode configurations are modelled along with 3D surface topography with the numerical processing distributed over multiple computers. The basic structure of the codes is identical to DCIP3D V1.0 provided to UBC-GIF sponsors in 1998 and to a subsequent release in 2006 of DCIP3D V2.1. However the improvements since the initial release are substantial. In the following we outline the major capabilities of DCIP3D v5.0 and also provide some background about changes that occurred since the v2.1 release. That is followed by a technical background, details about how to run the programs and examples.

Major items regarding DCIP3D v5.0

1. Forward modelling of DC and IP data is carried out over arbitrary 3D structures. The model is specified in the mesh of rectangular cells, each with a constant value of conductivity. Topography is included in the mesh. The electrode locations can be anywhere within the model volume, including below the topography to simulate borehole surveys. Electrodes do not need to be located on cell nodes. The DC equations are numerically solved using a finite volume method.

2. 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.

3. By minimizing the model objective function, distributions of subsurface conductivity/chargeabiilty are found that are both close to a reference model and smooth in three dimensions. The degree to which either of these two goals dominates is controlled by the user by incorporating prior geophysical or geological information into the inversion via reference models and additional weighting functions. Explicit prior information may also take the form of upper and lower bounds on the conductivity or chargeability in any cell.

4. 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.

5. The sensitivities are calculated and temporarily stored in memory rather than creating a large .mtx file.

6. Implementation of parallel computing architecture (OpenMP) allows the user to take full advantage of multi-core processors on a CPU.

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 (Canada's **N**ational **S**cience and **E**ngineering **R**esearch

Council) 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.

1.2 DCIP3D program library content

1. **Executable programs**. For performing 3D forward modelling and inversion of DC and IP surveys. The DCIP3D library (Windows or Linux platforms) consists of the programs:
   - `DCIPF3D`: performs forward modelling for DC and IP data
   - `DCINV3D`: inverts DC potentials to recover a 3D conductivity model using a Gauss-Newton method
   - `IPSEN3D`: calculates the sensitivity matrix for the 3D IP inversion
   - `IPINV3D`: inverts IP data to recover a 3D chargeability model
   - `MAKE_WDAT`: makes a weighting file for smoothing near surface zones of the model
2. **Graphical user interface**. The only available GUI-based utility for these codes is a viewer for model and mesh. It are only available on Windows platforms and can be freely downloaded through the UBC-GIF website:
   - `MESHTOOLS3D`: a utility for displaying resulting 3D models as volume renderings. Conductivity (linear or log scales) or chargeability volumes can be sliced in any direction, or isosurface renderings can be generated.
3. **Documentation** is provided for DCIP3D v5.0.
4. **Example** data sets are provided in the "Examples" directory within the compressed folder. One of those examples is shown at the end of this manual.

1.3 Licensing

There currently is no **educational version** of the program. Licensing for an unconstrained academic version is available if applicable; see the Licensing policy document.

NOTE: All academic licenses will be **time-limited to one year**. You can re-apply after that time.

Licensing for commercial use is managed by third party distributors. Details are in the Licensing policy document or on the webpage for DCIP3D.

1.4 Installing DCIP3D v5.0

There is no automatic installer currently available for the DCIP3D v5.0. 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 `C:\ubcgif\dcip3d\bin`

2. Add this directory as new path to your environment variables.

NOTE: do not store anything in the "bin" directory other than executable applications and Graphical User Interface applications (GUIs).

1.5 DCIP3D v5.0: Highlights of changes from version 2.1

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

Improvements since version 2.1:

1. A new projected gradient algorithm allows the user to implement bound constraints throughout the model.

2. Fully parallelized computational capability (for both sensitivity matrix calculations and inversion calculations).

The input file now requires an extra line for the bounds, which can be two values (upper and lower), or a file. Details of the structure of the input file and optional bounds files can be found within the manual.

1.5.1 Historical changes from v1.0 to v2.1

Previously, major changes were made to improve upon version 1.0. The major changes were

1. Electrodes can be located anywhere and are not restricted to mesh node locations. This greatly enhances the useability of the code and allows for more uniform meshes with fewer cells to be used in the modelling.

2. A linearized forward modelling of IP data is available. The output is the sensitivity multiplied by the chargeability. In this case, "chargeability" data can be the true dimensionless chargeability, or any of the other chargeability units that are commonly used, eg. pfe, mV/V, mrad etc.

3. `dcinv3d` now saves the model after each iteration. This allows additional flexibility when selecting an inverted model that is best suited for interpretation purposes, or possibly choosing a starting model for a subsequent inversion.

4. `ipinv3d` saves intermediate models for each Lagrange multiplier.

5. The details of the model objective function can be controlled either with length scales or alpha parameters.

6. An initial check is now carried out to determine if some data have an incorrect sign. Data signs are not changed but suspect data are identified in a new file (see end of section 2.2).

7. In `dcinv3d`, when the sensitivities are calculated, they are temporarily stored in memory rather than in the file dcinv3d.mtx. Hence the large .mtx file is not created. This results in a significant gain in performance.

8. `dcinv3d` and `ipsen3d` output a file called sensitivity.txt that contains the average absolute value of the sensitivity matrix for each cell. This is useful to determine which portions of the model domain are sensitive to the survey.

9. Each cell in a model can be set as "active" or "inactive" in the inversion process. In `dcinv3d`, inactive cells will be held at the value of the reference model. In the IP inversion, inactive cells will be set to zero.

10. An upgraded pre-conditioner is used for the CG (Conjugate Gradient) solver for the Gauss-Newton equations. This enhances the performance of the DC resistivity inversion and it has an even larger impact upon the IP inversion.

11. All floating-point arithmetic is now done in double precision. More accurate results are obtained.

12. The code has been reorganized. Large working arrays are only allocated and used when needed. This results in reduced memory requirements.

13. When calculating the sensitivity matrix $\mathbf{G}$ (in programs `dcinv3d`, `ipsen3d`, `dcipf3d` with the ip option), the number of times a forward system must be solved is equal to the number of transmitters plus the number of receivers. To speed up the process of calculating $\mathbf{G}$, if the same electrode location appears more than once in the data file, it is solution is stored in memory for future use.

1.6 Notes on computation speed

For large problems, DCIP3D v5.0 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 is substantial. It is strongly recommended to use multi-core processors for running the `dcinv3d` and `ipinv3d`. 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.

2 Background theory

2.1 Introduction

This section presents theoretical background, numerical examples, and explanation for implementing the program library DCIP3D v5.0. This suite of algorithms, developed at the UBC-Geophysical Inversion Facility, is needed to invert DC potentials and IP responses over a 3-D earth structure. The manual is designed so that a geophysicist who has an understanding of DC resistivity and Induced Polarization field experiments, but who is not necessarily versed in the details of inverse theory, can use the codes and invert his or her data.

A typical DC/IP experiment involves inputting a current $\boldsymbol{I}$ to the ground and measuring the potential away from the source. In a time-domain system the current has a duty cycle which alternates the direction of the current and has off-times between the current pulses at which the IP voltages are measured. A typical time-domain signature is shown in Figure 1. In this Figure, ϕ_σ is the potential that is measured in the absence of chargeability effects. This is the "instantaneous" value of the potential measured when the current is turned on. In mathematical terms this potential is related to the electrical conductivity, σ, by:

$$\phi_\sigma = \mathcal{F}_{dc}[\sigma], \tag{1}$$

where the forward mapping operator $\mathcal{F}_{dc}$ is defined by the equation

$$\nabla \cdot (\sigma \nabla \phi_\sigma) = -\mathbf{I}\delta(r - r_s), \tag{2}$$

and also by appropriate boundary conditions. In equation 2, σ is the electrical conductivity in Siemens/meter (S/m), ∇ is the gradient operator, $\mathbf{I}$ is the strength of the input current in Amperes, and r_s is the location of the current source. For typical earth structures, σ, while positive, can vary over many orders of magnitude. The potential ϕ_σ in equation 2 is the potential due to a single current. This is the value that would be measured in a pole-pole experiment. If potentials from pole-dipole or dipole-dipole surveys are to be generated then they can be obtained by using equation 2 and the principle of superposition.

When the earth material is chargeable the measured voltage will change with time and reach a limit value which is denoted by ϕ_η in Figure 1. There are a multitude of microscopic polarization phenomena, which collaborate so that this final value is achieved but all of these effects can be consolidated into a single macroscopic parameter called "chargeability". We denote chargeability by the symbol η. Chargeability is dimensionless, positive, and confined to the region [0,1).

To carry out forward modelling to compute ϕ_η we adopt the formulation of Siegel (1959), which says that the effect of a chargeable ground is modelled by using the dc resistivity forward mapping, $\mathcal{F}_{dc}$, but with the conductivity replaced by $\sigma = \sigma(1 - \eta)$. Thus:

$$\phi_\eta = \mathcal{F}_{dc}[\sigma(1 - \eta)], \tag{3}$$

or

$$\nabla \cdot (\sigma(1 - \eta) \nabla \phi_\sigma) = -\mathbf{I}\delta(r - r_s). \tag{4}$$

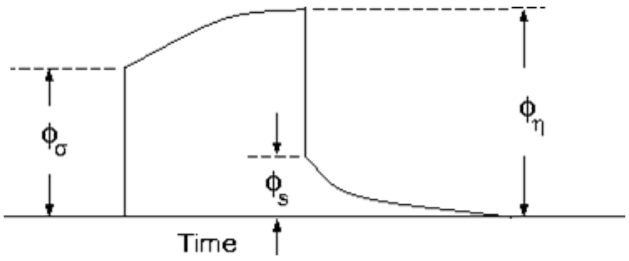


Figure 1: Definition of the three potentials associated with DC/IP experiments.

The IP datum, which we refer to as "apparent chargeability" is defined by

$$\eta_a = \frac{\phi_s}{\phi_\eta} = \frac{\phi_\eta - \phi_\sigma}{\phi_\eta}, \tag{5}$$

or

$$\eta_a = \frac{\mathcal{F}_{dc}[\sigma(1-\eta)] - \mathcal{F}_{dc}[\sigma]}{\mathcal{F}_{dc}[\sigma(1-\eta)]}. \tag{6}$$

Equation 6 shows that the apparent chargeability can be computed by carrying out two DC resistivity forward modelling routines with conductivities σ and $\sigma(1-\eta)$. Note that in this definition apparent chargeability is dimensionless and, in the case of data acquired over an earth having constant chargeability η_o, we have $\eta_a = \eta_o$.

The field data from a DC/IP survey are a set of N potentials (ideally ϕ_σ, but usually ϕ_η) and a set of N secondary potentials ϕ_s or a quantity that is related to ϕ_s. The goal of the user is to utilize these data to acquire quantitative information about the distribution of the two physical parameters of interest: conductivity $\sigma(x, y, z)$ and chargeability $\eta(x, y, z)$.

In this program library, the 3-D nature of the physical properties and surface topography are fully incorporated. The earth model is divided into cuboidal cells each having a constant value of conductivity and chargeability. The surface topography is approximated by a piecewise constant surface.

2.2 Forward modelling

The forward modelling for the DC potentials and IP apparent chargeabilities and secondary potentials is accomplished using a finite volume method (Dey and Morrison, 1979) technique to solve equation 2. The program that performs this calculation is `DCIPF3D`. In Version 5.0 we include the option to calculate IP data by multiplying the sensitivity matrix $\mathbf{J}$ by the chargeability provided by user. That is, we forward model with the linear equations that will be used for the inversion. The chargeability in this case can have arbitrary units. The forward modelled data are calculated as

$$\mathbf{d}_{ip} = \mathbf{J}_{ip}\eta, \tag{7}$$

where $\mathbf{d}_{ip}$ is the IP data and $\mathbf{J}_{ip}$ is the sensitivity matrix for the IP problem:

$$\mathbf{J}_{ip} = -\frac{\partial \ln \phi_\eta}{\partial \ln \sigma} = -\frac{1}{\sigma_\eta}\frac{\partial \phi_\eta}{\partial \ln \sigma} = -\frac{1}{\mathbf{d}_{dc}}\mathbf{J}_{dc}, \tag{8}$$

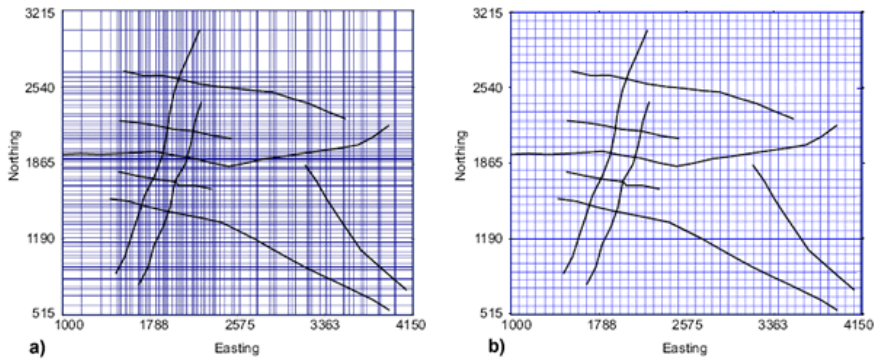


Figure 2: Mesh design with DCIP3D. Current and potential electrodes are located on the solid lines. (a) Version 1.0 required electrodes be placed on cell nodes. (b) Update versions allow for the electrodes to be placed anywhere.

given DC data, $\mathbf{d}_{dc}$. Forward modelling using equation 8 is further explained in the section 2.5.

The forward (and inversion) code uses a nodal-based finite volume technique in which the current is input on a node. This is an important change from the original version of DCIP3D and is illustrated in Figure 2. When inverting field data, the usual procedure is to generate a mesh whose nodes coincide with the location of the current electrodes. The difficulty with accomplishing this task is illustrated in Figure 2a. The left panel is an attempt to design a mesh that permits each electrode to be on a node. The number of cells required to accomplish this is large and the aspect ratios are undesirable. High aspect ratios of cells reduces the numerical accuracy and also reduces the speed at which the forward modelling equations can be solved. This problem is greatly exacerbated when field surveys are carried out in regions of considerable topography. It would be preferable to have a uniform gridding in which the cell size is dictated by the resolving power of the data rather than by small details regarding exact placement of electrodes. A desired grid is shown in Figure 2b.

To handle a current electrode that is at an arbitrary position (x_s, y_s, z_s) in the cell we made a modification to distribute any current amongst the 8 nodes of the cell. This approach is shown in Figure 3, where a current I is distributed onto nodes P1 through P8. Effectively, we write

$$I\delta(r_s) \approx \sum_{i=1}^{8} I w_i(r_i, r_s)\delta(r_i), \tag{9}$$

where $r_s = (x_s, y_s, z_s)$ is the position of the current electrode, $r_i = (x_i, y_i, z_i)$ is the position of the ith node, and w_i is the linear-interpolation weighting for the ith node

$$\sum_{i=1}^{8} w_i = 1, \tag{10}$$

so that the total current distributed among the 8 nodes is equal to I. With the linear interpolation we note that if the source electrode is on one of the faces of the cell, then only 4 nodes will be activated. If the source electrode is along an edge then the current is distributed between two

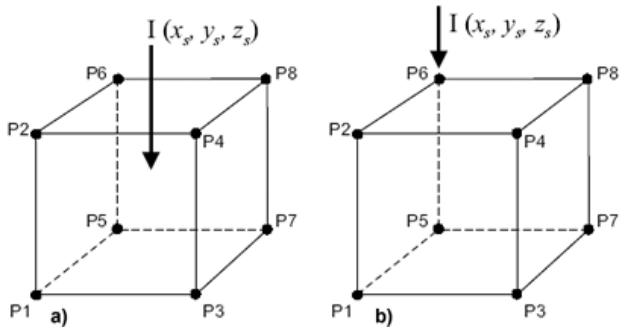


Figure 3: Current electrode can be placed at an arbitrary position (x_s, y_s, z_s) within a cell, or on a node. The currents are distributed to each node of the cell through linear interpolation.

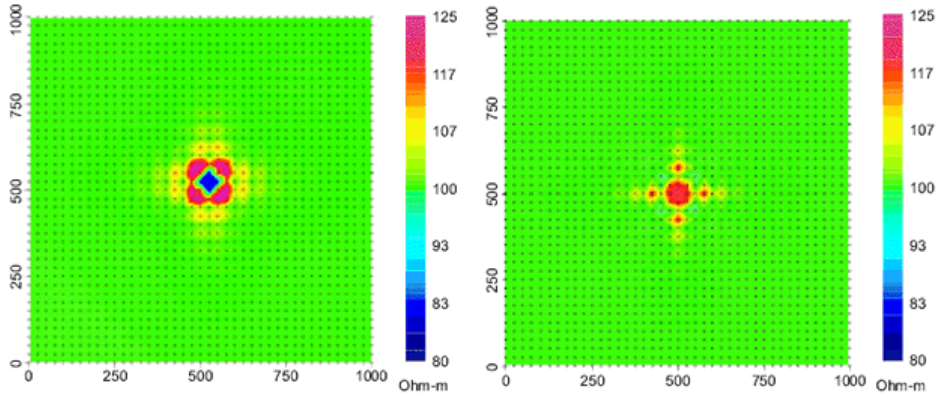


Figure 4: Forward modelled apparent resistivities of a halfspace showing differences between placing the current electrode at (a) cell centers versus (b) cell nodal points.

nodes, and if the source electrode is at a corner of the prism, then only one node is activated. If potential electrodes are not on the nodes then fields are linearly interpolated.

As an example of the importance of the interpolation, a halfspace is modelled in which the cell size is 50 m and a current is injected at the surface in the center of the cell. Potentials are obtained on a 25-m grid. Apparent resistivities should equal 100 Ohm-m, which is the true halfspace value. The results are shown in Figure 4a. Errors up to 25% are observed at locations that are 25 m (1/2 cell) from each of the four corners where the distributed current is input. At distances 50 m (one cell width) the error has dropped to about 7%. These are expected results and are in accordance with testing using the first version of `DCIP3D`. In Figure 4b the apparent resistivities for a current on a nodal location are plotted. At a distance of 50 m from the current, the error is less than 5% if with on a nodal discretization. The errors increase somewhat between distances of one and two cells. We conclude that for numerical accuracies of about 5% or less, the observation should be at least one cell width away from the location of a current electrode.

2.3 General inversion methodology

The computing programs outlined in this manual solve two inverse problems. In the first we invert the DC potentials ϕ_σ to recover the electrical conductivity $\sigma(x, z)$. This is a non-linear inverse problem that requires linearization of the data equations and subsequent iteration steps. After DC data have been inverted, the conductivity model is used to invert the IP data to recover the chargeability $\eta(x, z)$. Because chargeabilities are usually small quantities ($\eta < 0.3$) it is possible to linearize equation 6 and derive a linear system of equations to be solved. Irrespective of which data set is being inverted however, we basically use the same methodology to carry out the inversions. We outline this methodology here.

To outline our methodology it is convenient to introduce a single notation for the "data" and for the "model". We let $\mathbf{d} = (d_1, d_2, \ldots, d_n)^T$ denote the data so that d_i is the ith potential in a DC resistivity data set or the ith apparent chargeability in an IP survey. Let the physical property of interest be denoted by the symbol m. The quantity m_j can denote the conductivity or chargeability for the jth cell. For the inversion we choose $m_j = \ln(\sigma_j)$, when inverting for conductivities and $m_j = \eta_j$ when inverting the chargeability data.

The goal of the inversion is to recover a model vector $\mathbf{m} = (m_1, m_2, \ldots, m_m)^T$, which acceptably reproduces the n observations of $\mathbf{d}$. Importantly, the data are noise contaminated, therefore we don't want to fit them precisely. A perfect fit in our case would be indicative, that incorrect earth model is recovered, as some features observed in the constructed model would assuredly be artifacts of the noise. Therefore, the inverse problem is formulated as an optimization problem where a global objective function, ψ, is minimized. The global objective functions consists of two components: a model objective function, ψ_m, and a data misfit function, ψ_d, such that

$$\begin{aligned} \min \psi &= \psi_d + \beta \psi_m \\ \text{s. t. } & m^l \leq m \leq m^u, \end{aligned} \tag{11}$$

where β is a trade off parameter that controls the relative importance of the model smoothness through the model objective function and data misfit function. When the standard deviations of data errors are known, the acceptable misfit is given by the expected value ψ_d and we will search for the value of β via an L-curve criterion (Hansen, 2000) that produces the expected misfit at each linearized step (see section 2.4 or 2.4). Otherwise, a user-defined value is used. Bound are imposed through the projected gradient method so that the recovered model lies between imposed lower (m^l) and upper (m^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 m_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

$$\begin{aligned}\phi_m(m) &= \alpha_s \int_V \{w_s(\mathbf{r})[m(\mathbf{r}) - m_o]\}^2 \, dv + \alpha_x \int_V w_x(\mathbf{r}) \left\{ \frac{\partial [m(\mathbf{r}) - m_o]}{\partial x} \right\}^2 dv \\ &+ \alpha_y \int_V w_y(\mathbf{r}) \left\{ \frac{\partial [m(\mathbf{r}) - m_o]}{\partial y} \right\}^2 dv + \alpha_z \int_V w_z(\mathbf{r}) \left\{ \frac{\partial [m(\mathbf{r}) - m_o]}{\partial z} \right\}^2 dv, \end{aligned} \tag{12}$$

where the functions w_s, w_x, w_y and w_z are user-given generalized weighting functions, while α_s, α_x, α_y and α_z are coefficients, which affect the relative importance of different components in the objective function. The reference model is denoted as m_o. The purpose of the genearalized weighting functions are to place emphasis throughout the model to utilize prior information.

The objective function in equation 12 has the flexibility to incorporate many types of prior knowledge into the inversion. The reference model may be a general background model (e.g., background conductivity) that is estimated from previous investigations or it will be a zero model (in terms of chargeability). 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 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 12 is discretized onto the mesh defining the conductivity or chargeability model using a finite difference approximation. This yields:

$$\begin{aligned}\psi_m(\boldsymbol{m}) = (\boldsymbol{m} - \boldsymbol{m}_o)^T (\alpha_s \mathbf{W}_s^T \mathbf{W}_s + \alpha_x \mathbf{W}_x^T \mathbf{W}_x + \alpha_y \mathbf{W}_y^T \mathbf{W}_y + \alpha_z \mathbf{W}_z^T \mathbf{W}_z)(\boldsymbol{m} - \boldsymbol{m}_o), \\ \equiv (\boldsymbol{m} - \boldsymbol{m}_o)^T \mathbf{W}_m^T \mathbf{W}_m (\boldsymbol{m} - \boldsymbol{m}_o), \\ = \left\| \mathbf{W}_m (\boldsymbol{m} - \boldsymbol{m}_o) \right\|^2, \end{aligned} \tag{13}$$

where $\mathbf{m}$ and $\mathbf{m}_o$ are M-length vectors representing the recovered and reference models, respectively. Similarly, there is an option to remove to the reference model from the spatial derivatives in equation 13 such that

$$\begin{aligned}\psi_m(\boldsymbol{m}) = (\boldsymbol{m} - \boldsymbol{m}_o)^T (\alpha_s \mathbf{W}_s^T \mathbf{W}_s)(\boldsymbol{m} - \boldsymbol{m}_o) + \boldsymbol{m}^T (\alpha_x \mathbf{W}_x^T \mathbf{W}_x + \alpha_y \mathbf{W}_y^T \mathbf{W}_y + \alpha_z \mathbf{W}_z^T \mathbf{W}_z) \boldsymbol{m}, \\ \equiv (\boldsymbol{m} - \boldsymbol{m}_o)^T \mathbf{W}_s^T \mathbf{W}_s (\boldsymbol{m} - \boldsymbol{m}_o) + \boldsymbol{m}^T \mathbf{W}_m^T \mathbf{W}_m \boldsymbol{m}, \\ = \left\| \mathbf{W}_s (\boldsymbol{m} - \boldsymbol{m}_o) + \mathbf{W}_m \boldsymbol{m} \right\|^2. \end{aligned} \tag{14}$$

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.

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

$$\psi_d = \|\mathbf{W}_d(\mathcal{F}[\mathbf{m}] - \mathbf{d})\|^2 . \tag{15}$$

For the work here, we assume that the contaminating noise on the data is independent and Gaussian with zero mean. Specifying $\mathbf{W}_d$ to be a diagonal matrix whose i^{th} element is $1/\sigma_i$, where σ_i is the standard deviation of the i^{th} datum makes ψ_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 11.

To solve the optimization problem when constraints are imposed we use the projected gradients method (Calamai and Moré, 1987; Vogel, 2002). 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.

2.4 Inversion of DC data

The inversion of the apparent resistivity data is carried out using the program DCINV3D. The inversion of DC resistivity data formulated as the minimization in equation 11 is nonlinear since the data do not depend linearly upon the conductivity model. We tackle this problem using a Gauss-Newton approach in which the objective function is linearized about a current model, $m(n)$, and a model perturbation is solved for and used to update the current model. Substituting $m(n+1) = m(n) + m$ into the objective function in equation 11

$$\psi(\mathbf{m} + \delta\mathbf{m}) = \left\|\mathbf{W}_d \left(\mathcal{F}_{dc}[\mathbf{m}^{(n)}] + \mathbf{J}\delta\mathbf{m} - \mathbf{d}\right)\right\|^2 + \beta \left\|\mathbf{W}_m \left(\mathbf{m} + \delta\mathbf{m} - \mathbf{m}_o\right)\right\|^2 + H.O.T., \tag{16}$$

where $\mathbf{J}$ is the sensitivity matrix and the element J_{ij} quantifies the influence of the model change in jth cell on the ith datum such that

$$\mathbf{J} = \frac{\partial d_i}{\partial m_j} = \frac{\partial \phi_i}{\ln \sigma_j}. \tag{17}$$

Neglecting the higher order terms and setting to zero the derivative with respect to δm yields

$$\left(\mathbf{J}^T\mathbf{J} + \beta\mathbf{W}_m^T\mathbf{W}_m\right)\delta\mathbf{m} = -\mathbf{J}^T\left(\mathcal{F}_{dc}[\mathbf{m}^{(n)}] - \mathbf{d}\right) - \beta\mathbf{W}_m^T\mathbf{W}_m\left(\mathbf{m}^n - \mathbf{m}_o\right). \tag{18}$$

Here we assume that the matrix $\mathbf{W}_d$ has been absorbed into the sensitivity matrix and data vectors. This is the basic equation that is solved to obtain the model perturbation. The new model is then generated by

$$\mathbf{m}^{(n+1)} = \mathbf{m}^{(n)} + \gamma\delta\mathbf{m}, \tag{19}$$

where $\gamma \in (0, 1]$ limits the step size and is chosen to ensure that the total objective function is reduced.

The major computational effort in this approach includes the calculation of the sensitivity matrix, solution of the basic linearized equation 18, and the choice of regularization parameter β. The sensitivity is computed using the standard adjoint equation approach. The equation 18 is solved using a pre-conditioned conjugate gradient technique, in which the sensitivity matrix $\mathbf{J}$ is applied to vectors by sparse multiplications in the wavelet domain after it is compressed using fast wavelet transform.

2.5 Inversion of IP data

The inversion of IP data begins by linearizing and expanding equation 3 such that

$$\phi_\eta = \phi(\sigma - \eta\sigma) = \phi(\sigma) - \sum_{j=1}^{M} \frac{\partial \phi}{\partial \sigma_j} \eta_j \sigma_j + H.O.T. \tag{20}$$

The above equation is then subsituted into equation 5:

$$\begin{aligned} \phi_s &= \phi_\eta + \phi_\sigma \\ &= -\sum_{j=1}^{M} \frac{\partial \phi}{\partial \sigma_j} \eta_j \sigma_j + H.O.T. \end{aligned} \tag{21}$$

This can be approximately written as

$$\phi_s = -\sum_j \sigma_j \frac{\partial \phi}{\partial \sigma_j} \eta_j = -\sum_j \frac{\partial \phi}{\partial \ln \sigma_j} \eta_j. \tag{22}$$

When apparent chargeability is used as the IP data, substituting the above equation into equation 6 yields,

$$\eta_a = -\sum_j \frac{\sigma_j}{\phi_i} \frac{\partial \phi_i}{\partial \sigma_j} \eta_j = -\sum_j \frac{\partial \ln \phi}{\partial \ln \sigma_j} \eta_j. \tag{23}$$

Thus a datum (either secondary potential or apparent chargeability) is expressed as

$$d_i = \sum_{j=1}^{M} J_{ij} \eta_j, \tag{24}$$

where

$$J_{ij} = \begin{cases} -\frac{\partial \phi_i[\sigma]}{\partial \ln \sigma_j}, & d = \phi_s \\ -\frac{\partial \ln \phi[\sigma]}{\partial \ln \sigma_j}, & d = \eta_a \end{cases} \tag{25}$$

The general problem takes the form of $\mathbf{d} = \mathbf{Jm}$ and can be solved as described in section 2.3. Bound constraints (e.g., positivity) for IP are imposed through projected gradients (Calamai and Moré, 1987; Vogel, 2002). The sensitivity matrix is dense and thus wavelets are used for compression.

2.6 Wavelet Compression of Sensitivity Matrix

When storing the sensitivity matrix during the linearized step of the DC problem, the two major obstacles to the solution of the Gauss-Newton 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. These are also points of concern for the general inversion of IP data. The DCIP3D v5.0 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 (2003, 2010). In the following, we give a brief description of the method necessary for the use of the DCIP3D v5.0 library.

Each row of the sensitivity matrix in a 3D DC resistivity or IP 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.

We first denote $\mathcal{W}$ as the symbolic matrix-representation of the 3D wavelet transform. Then applying the transform to each row of $\mathbf{J}$ and forming a new matrix consisting of rows of transformed sensitivity is equivalent to the following operation:

$$\widetilde{\mathbf{J}} = \mathbf{J}\mathcal{W}^T, \tag{26}$$

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

$$\widetilde{J}^s_{ij} = \begin{cases} \widetilde{J}_{ij} & \text{if } \left|\widetilde{J}_{ij}\right| \geq \delta_i \\ 0 & \text{if } \left|\widetilde{J}_{ij}\right| < \delta_i \end{cases}, \quad i = 1, \ldots, N, \tag{27}$$

where δ_i is the threshold level, and $\widetilde{J}_{ij}$ and $\widetilde{J}^s_{ij}$ are the elements of $\widetilde{\mathbf{J}}$ and $\widetilde{\mathbf{J}}^S$, 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:

$$r_i(\delta_i) = \sqrt{\frac{\sum\limits_{|\widetilde{J}_{ij}|<\delta_i} \widetilde{J}^2_{ij}}{\sum\limits_j \widetilde{J}^2_{ij}}}, \quad i = 1, \ldots, N, \tag{28}$$

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

$$\delta_i = \epsilon \max_j \left| \widetilde{J}_{ij} \right|, \quad i = 1, \ldots, N. \tag{29}$$

The program that implements this compression procedure is DCINV3D. The user is asked to specify the relative error r^* and the program will determine the relative threshold level δ_i. Usually a value of a few percent is appropriate for r^*. When both surface and borehole data are present, two different relative threshold levels are calculated by choosing a representative row for surface data and another for borehole data. For experienced users, the program also allows the direct input of the relative threshold level.

3 Elements of the program DCIP3D

3.1 Introduction

The DCIP3D v5.0 program library consists of the programs:

1. DCIPF3D: performs forward modelling for DC and IP data
2. DCINV3D: inverts DC potentials to recover a 3D conductivity model using a Gauss-Newton method
3. IPSEN3D: calculates the sensitivity matrix for the 3D IP inversion
4. IPINV3D: inverts IP data to recover a 3D chargeability model
5. MAKE_WDAT: makes file a discrete topography file and/or weighting file for smoothing near surface zones of the model

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

3.2 General files for DCIP3D v5.0 programs

There are eight general files which are used in DCIP3D v5.0. These are:

1. observation: specifies the electrode locations and observed measurements
2. location: contains the electrode locations for forward modelling
3. mesh: contains the finite volume mesh for 3D modelling and inversion
4. topography: contains the topographic data
5. model: model file structure to hold cell values for the conductivity or chargeability model
6. active: Contains information about active/inactive cells (same format as a model)
7. bounds: Contains information about lower and upper bounds (differs slightly from model)
8. weighting: file that contains special weightings which alter the type of model produced in the inversions.

3.2.1 Observations file

This file contains the observed measurements and the associated electrode locations. Both potential data and apparent chargeability data are stored in the same format. This will be the format of all the data files that are output from DCIPF3D, and are input to DCINV3D and IPINV3D. For convenience, there are two slightly different file formats: surface and general. The two data formats cannot be mixed in one file. Both formats use what is called the "standard format" that is found in the two-dimensional version of the code (DCIP2D).

General format

The DCIP3D v5.0 library can handle arbitrary electrode configurations, and a mixture of different configurations can be present in the data file. This is accomplished by specifying the locations of four electrodes for each datum. Whenever the two current electrodes, or two potential electrodes, are given the identical location, that particular pair is considered to be a single pole with the negative electrode being at infinity. It should be noted that if this format is used and topography is desired, then **the topography format used must be discrete** (See the "topography" section). The format consists of a line with the current electrode location and number of potential electrode locations associated with it. An example of the general format file structure is as follows:

Comment lines
[IPTYPE= 1 | 2]

A_1^x	A_1^y	A_1^z	B_1^x	B_1^y	B_1^z	n_1	
M_1^x	M_1^y	M_1^z	N_1^x	N_1^y	N_1^z	val_1	stn_1
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮
$M_{n_1}^x$	$M_{n_1}^y$	$M_{n_1}^z$	$N_{n_1}^x$	$N_{n_1}^y$	$N_{n_1}^z$	val_{n_1}	stn_{n_1}
A_2^x	A_2^y	A_2^z	B_2^x	B_2^y	B_2^z	n_2	
M_1^x	M_1^y	M_1^z	N_1^x	N_1^y	N_1^z	val_1	stn_1
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮
$M_{n_2}^x$	$M_{n_2}^y$	$M_{n_2}^z$	$N_{n_2}^x$	$N_{n_2}^y$	$N_{n_2}^z$	val_{n_2}	stn_{n_2}
A_3^x	A_3^y	A_3^z	B_3^x	B_3^y	B_3^z	n_3	
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮

The following are detailed summaries of components of the standard-format observations file:

`Comment lines` Any comments can go here and should be preceded by an "!"

`IPTYPE` Only used for IP inversion and not required if only using DC inversion. NOTE: If omitted from IP inversion, the program will choose `IPTYPE=1`.
`=1`, Type of IP data is apparent chargeability
`=2`, Type of IP data is secondary potentials

A_i^x i^{th} easting position (x) of current electrode A

$\texttt{A}_i^y$ ith northing position (y) of current electrode A

$\texttt{A}_i^z$ ith vertical position (z) of current electrode A

$\texttt{B}_i^x$ ith easting position (x) current electrode B

$\texttt{B}_i^y$ ith northing position (y) of current electrode B

$\texttt{B}_i^z$ ith vertical position (z) of current electrode B

$\texttt{M}_j^x$ jth easting position (x) of potential electrode M associated with the ith current pair

$\texttt{M}_j^y$ jth northing position (y) of potential electrode M associated with the ith current pair

$\texttt{M}_j^z$ jth vertical position (z) of potential electrode M associated with the ith current pair

$\texttt{N}_j^x$ jth easting position (x) of potential electrode N associated with the ith current pair

$\texttt{N}_j^y$ jth northing position (y) of potential electrode N associated with the ith current pair

$\texttt{N}_j^z$ jth vertical position (z) of potential electrode N associated with the ith current pair

$\texttt{val}_j$ jth observed datum related to the jth potential electrode pair and ith current electrode pair. The potential measurements must be measured value in Volts, or a dimensionless real number (not percentage) for apparent chargeability (*potential is always normalized to unit current amplitude*). There are four types of IP data generally in use; two gathered in the time domain and two gathered in the frequency domain. For small chargeabilities, as is nearly always the case for earth materials, all data types can be used as input for inversion, and resulting models will have chargeabilities in the same units.

$\texttt{stn}_j$ jth standard deviation associated with the jth datum. This is a positive, absolute value (not a percentage) in units of the data.

Example of general format The following is an example of DC data (e.g., no IPTYPE):

```
!  dc data
221   500   -45    221    500   -45        6
 50   500   250    100    500    25   -2.31552E-01   1.16776E-02
100   500   250    150    500    50   -2.64516E-01   1.33258E-02
150   500   500    200    500    75    2.70551E-03   2.35276E-04
200   500    75    250    500   100    2.11746E-01   1.06873E-02
250   500   100    300    500   125    2.37240E-01   1.19620E-02
300   500   125    350    500   150    1.59822E-01   8.09110E-03
221   500   -45   1500.0  600   -55        2
100   500    25    150    500   500   -2.64516E-01   1.33258E-02
150   500   500    200    500  75.0    2.70551E-03   2.35276E-04
```

In the above example, there are two current electrode locations, the first with six potential electrodes and the second with two potential electrode data. The line "IPTYPE=2" would be added if this file were IP data of second potentials.

Surface format

The surface format is similar to the general format with difference that the elevation data is not given. Instead, the program places the electrodes on top of the discretized topographic surface. Accordingly, this format **cannot be used with borehole data** and if no topography is given, assumes the data are on top of the mesh. Whenever the two current electrodes, or two potential electrodes, are given the identical location, that particular pair is considered to be a single pole with the negative electrode being at infinity. The format consists of a line with the current electrode location and number of potential electrode locations associated with it. This format can be used with any topography type. An example of the surface format file structure is as follows:

```
Comment lines
[IPTYPE= 1 | 2]
A1x   A1y   B1x   B1y   n1
M1x   M1y   N1x   N1y   val1      stn1
 ⋮     ⋮     ⋮     ⋮     ⋮         ⋮
Mn1x  Mn1y  Nn1x  Nn1y  valn1     stnn1
A2x   A2y   B2x   B2y   n2
M1x   M1y   N1x   N1y   val1      stn1
 ⋮     ⋮     ⋮     ⋮     ⋮         ⋮
Mn2x  Mn2y  Nn2x  Nn2y  valn2     stnn2
A3x   A3y   B3x   B3y   n3
 ⋮     ⋮     ⋮     ⋮     ⋮         ⋮
```

The following are detailed summaries of components of the surface-format observations file:

`Comment Lines` Any comments can go here and should be preceded by an "!"

`IPTYPE` Only used for IP inversion and not required if only using DC inversion. NOTE: If omitted from IP inversion, the program will choose `IPTYPE=1`.
`=1`, Type of IP data is apparent chargeability
`=2`, Type of IP data is secondary potentials

A_i^x ith easting position (x) of current electrode A

A_i^y ith northing position (y) of current electrode A

B_i^x ith position (x) of current electrode B

B_i^y ith northing position (y) of current electrode B

M_j^x jth easting position (x) of potential electrode M associated with the ith current pair

M_j^y jth northing position (y) of potential electrode M associated with the ith current pair

N_j^x jth position (x) of potential electrode N associated with the ith current pair

N_j^y j^{th} northing position (y) of potential electrode N associated with the i^{th} current pair

val_j j^{th} observed datum related to the j^{th} potential electrode pair and i^th current electrode pair. The potential measurements must be measured value in Volts, or a dimensionless real number (not percentage) for apparent chargeability (*potential is always normalized to unit current amplitude*). There are four types of IP data generally in use; two gathered in the time domain and two gathered in the frequency domain. For small chargeabilities, as is nearly always the case for earth materials, all data types can be used as input for inversion, and resulting models will have chargeabilities in the same units.

stn_j j^{th} standard deviation associated with the j^{th} datum. This is a positive, absolute value (not a percentage) in units of the data.

Example of surface format The following is an example of IP data in units of apparent chargeability:

```
IPTYPE=1
221   -45    221    -45         4
 50   250    100     25   -2.31552E-01   1.16776E-02
100   250    150     50   -2.64516E-01   1.33258E-02
250   100    300    125    2.37240E-01   1.19620E-02
300   125    350    150    1.59822E-01   8.09110E-03
221   -45   1500.0  -55         2
100    25    150    500   -2.64516E-01   1.33258E-02
150   500    200   75.0    2.70551E-03   2.35276E-04
```

In the above example, there are two current electrode locations, the first with four potential electrodes and the second with two potential electrode data. The line "IPTYPE=1" would be absent if this file were DC data.

3.2.2 Locations file

This file contains the associated current and potential electrode locations. It is the in the same format as the observations file, without the data values and thier standard deviations. For convenience, there are two slightly different file formats: surface and general. The two data formats cannot be mixed in one file.

General location format

The DCIP3D v5.0 library can handle arbitrary electrode configurations, and a mixture of different configurations can be present in the data file. This is accomplished by specifying the locations of four electrodes for each datum. Whenever the two current electrodes, or two potential electrodes, are given the identical location, that particular pair is considered to be a single pole with the

negative electrode being at infinity. It should be noted that if this format is used and topography is desired, then **the topography format used must be discrete** (See the "topography" section). The format consists of a line with the current electrode location and number of potential electrode locations associated with it. An example of the general format file structure is as follows:

```
Comment lines
[IPTYPE= 1 | 2]
A1x  A1y  A1z  B1x  B1y  B1z  n1
M1x  M1y  M1z  N1x  N1y  N1z
 ⋮    ⋮    ⋮    ⋮    ⋮    ⋮
Mn1x Mn1y Mn1z Nn1x Nn1y Nn1z
A2x  A2y  A2z  B2x  B2y  B2z  n2
M1x  M1y  M1z  N1x  N1y  N1z
 ⋮    ⋮    ⋮    ⋮    ⋮    ⋮
Mn2x Mn2y Mn2z Nn2x Nn2y Nn2z
A3x  A3y  A3z  B3x  B3y  B3z  n3
 ⋮    ⋮    ⋮    ⋮    ⋮    ⋮   ⋮
```

The following are detailed summaries of components of the standard-format observations file:

`Comment lines` Any comments can go here and should be preceded by an "!"

`IPTYPE` Only used for IP inversion and not required if only using DC inversion. NOTE: If omitted from IP inversion, the program will choose `IPTYPE=1`.
`=1`, Type of IP data is apparent chargeability
`=2`, Type of IP data is secondary potentials

$\texttt{A}_i^x$ ith easting position (x) of current electrode A

$\texttt{A}_i^y$ ith northing position (y) of current electrode A

$\texttt{A}_i^z$ ith vertical position (z) of current electrode A

$\texttt{B}_i^x$ ith easting position (x) current electrode B

$\texttt{B}_i^y$ ith northing position (y) of current electrode B

$\texttt{B}_i^z$ ith vertical position (z) of current electrode B

$\texttt{M}_j^x$ jth easting position (x) of potential electrode M associated with the ith current pair

$\texttt{M}_j^y$ jth northing position (y) of potential electrode M associated with the ith current pair

$\texttt{M}_j^z$ jth vertical position (z) of potential electrode M associated with the ith current pair

$\texttt{N}_j^x$ jth easting position (x) of potential electrode N associated with the ith current pair

$\texttt{N}_j^y$ jth northing position (y) of potential electrode N associated with the ith current pair

$\texttt{N}_j^z$ jth vertical position (z) of potential electrode N associated with the ith current pair

Example of general format

The following is an example of IP data of apparent chargeability:

```
!  my IP data
IPTYPE=1
221   500   -45    221    500   -45   6
 50   500   250    100    500    25
100   500   250    150    500    50
150   500   500    200    500    75
200   500    75    250    500   100
250   500   100    300    500   125
300   500   125    350    500   150
221   500   -45   1500.0  600   -55   2
100   500    25    150    500   500
150   500   500    200    500  75.0
```

In the above example, there are two current electrode locations, the first with six potential electrodes and the second with two potential electrode data.

Surface location format

The surface format is similar to the general format with difference that the elevation data is not given. Instead, the program places the electrodes on top of the discretized topographic surface. Accordingly, this format **cannot be used with borehole data** and if no topography is given, assumes the data are on top of the mesh. Whenever the two current electrodes, or two potential electrodes, are given the identical location, that particular pair is considered to be a single pole with the negative electrode being at infinity. The format consists of a line with the current electrode location and number of potential electrode locations associated with it. This format can be used with any topography type. An example of the surface format file structure is as follows:

```
Comment lines
[IPTYPE= 1 | 2]
```

$$
\begin{array}{cccccc}
A_1^x & A_1^y & B_1^x & B_1^y & n_1 \\
M_1^x & M_1^y & N_1^x & N_1^y & \\
\vdots & \vdots & \vdots & \vdots & \\
M_{n_1}^x & M_{n_1}^y & N_{n_1}^x & N_{n_1}^y & \\
A_2^x & A_2^y & B_2^x & B_2^y & n_2 \\
M_1^x & M_1^y & N_1^x & N_1^y & \\
\vdots & \vdots & \vdots & \vdots & \\
M_{n_2}^x & M_{n_2}^y & N_{n_2}^x & N_{n_2}^y & \\
A_3^x & A_3^y & B_3^x & B_3^y & n_3 \\
\vdots & \vdots & \vdots & \vdots & \vdots
\end{array}
$$

The following are detailed summaries of components of the surface-format observations file:

`Comment lines` Any comments can go here and should be preceded by an "!"

`IPTYPE` Only used for IP inversion and not required if only using DC inversion. NOTE: If omitted from IP inversion, the program will choose `IPTYPE=1`.
`=1`, Type of IP data is apparent chargeability
`=2`, Type of IP data is secondary potentials

A_i^x ith easting position (x) of current electrode A

A_i^y ith northing position (y) of current electrode A

B_i^x ith position (x) of current electrode B

B_i^y ith northing position (y) of current electrode B

M_j^x jth easting position (x) of potential electrode M associated with the ith current pair

M_j^y jth northing position (y) of potential electrode M associated with the ith current pair

N_j^x jth position (x) of potential electrode N associated with the ith current pair

N_j^y jth northing position (y) of potential electrode N associated with the ith current pair

Example of surface format

The following is an example of DC data:

```
!  my DC data
221   500    -45     -45   5
 50   250    100      25
100   500    250     150
150   500    200      75
200   500     75     250
300   125    350     500
221   -45   1500.0   -55   2
100   500     25     500
150   500    200    75.0
```

In the above DC data example, there are two current electrode locations, the first with five potential electrodes and the second with two potential electrode data.

3.2.3 Mesh file

This file contains the 3D mesh, for example mesh.msh, which defines the model region. Mesh file has the following structure:

NE	NN	NZ	
E_o	N_o	Z_o	
ΔE_1	ΔE_2	$\cdots$	ΔE_{NE}
ΔN_1	ΔN_2	$\cdots$	ΔN_{NN}
ΔZ_1	ΔV_2	$\cdots$	ΔZ_{NZ}

NE Number of cells in the East direction.

NN Number of cells in the North direction

NZ Number of cells in the vertical direction

E_o N_o Z_o Coordinates, in meters, of the southwest top corner, specified in (Easting, Northing, Elevation). The elevation can be relative to a reference elevation other than the sea level, but it needs to be consistent with the elevation used to specify the locations, observations, and topography files (see the relevant file descriptions).

ΔE_n n^{th} cell width in the easting direction (ordered W to E).

ΔN_n n^{th} cell width in the northing direction (ordered S to N).

ΔZ_n n^{th} cell thickness (ordered top to bottom).

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

The vertical position of the mesh is specified in elevation. This is to accommodate the inversion of a data set acquired over a topographic surface. When there is strong topographic relief, which the user wishes to incorporate it into the inversion, special care should be taken to design the mesh. A conceptually simple approach is first to design a rectangular mesh whose top (specified by Z_o) is just below the highest elevation point, and then to strip off cells that are above the topographic surface. This is the approach taken in DCIP3D v5.0. 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 of mesh file

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
```

3.2.4 Topography file

This optional file is used to define the surface topography of the 3D model. If no topography is given, the electrodes are placed on the top of the mesh (for surface observations format) or assumed borehole data (for general observations format) unless the an active cell model is given that incorporates topography (essentially creates inactive cells in air with a coordinated reference model to incorporate the values of air cells). There are two different formats that are used for topography. The first format, the general topography format, has the code interpolate the topography and discretize it on the mesh. **The general format only works with the Surface observation format**. The second is a pre-defined discrezation and thus called the discrete topography format. **The discrete format works with both types of observations.**

General topography format

The general topography file has the following structure:

```
Comment lines
npt
E_1    N_1    ELEV_1
E_2    N_2    ELEV_2
 ⋮      ⋮       ⋮
E_n    N_n    ELEV_n
```

Parameter definitions:

`Comment lines` Any comments can go here and should be preceded by an "!"

`npt` Number of points defining the topographic surface.

`E`$_i$ Easting of the i^{th} point on the surface.

`N`$_i$ Northing of the i^{th} point on the surface.

`ELEV`$_i$ Elevation of the i^{th} point on the profile.

The lines in this file can be in any order as long as the total number is equal to `npt`. The topographic data need not be supplied on a regular grid. DCIP3D v5.0 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.

Example of general topography file

The following is an example of a general topography file:

```
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
```

NOTE: Only the cells completely below the topography surface are kept. The cells above the topographic surface are removed from the model, although these must still be included in the model file 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 as an identifier (e.g. `-100`).

Discrete topography format

The discrete topography format contains integer values of the number of cells from the top of the mesh to the discretized topographic surface. This format is compatible with both types of data formats.

NOTE: If you have a topography file *and* a general observations, use `MAKE_WDAT` first to create a discrete topography format by using the mesh and the general topography formatted file. Then use the discrete topography formatted file for the remainder of the inversions of the data set.

The discrete topography file has the following format

NE	NN	
i_1	j_1	$k[i_1, j_1]$
i_2	j_1	$k[i_2, j_1]$
$\vdots$	$\vdots$	$\vdots$
i_{NE}	j_{NN}	$k[i_{NE}, j_{NN}]$

where

NN Number of cells in the north (NN from the mesh)

NE Number of cells in the north (NE from the mesh)

i_n nth cell in the easting direction, starting with the west-most cells= 1

j_m mth cell in the northing direction, starting with the south-most cells = 1

$k[i_n, j_m]$ Number of cells from top of the mesh to the topography

The lines in this file can be in any order as long as the total number is equal to $NE \times NN$. There are no comments allowed in this file. The value of $k[i, j]$ indicates the number of cells above the earth's surface. If the topography reached the top row of cells in the vertical direction, then $k_{i,j} = 0$. If all values of k are zero, the topography would be at the very top of the mesh (i.e., the situation where no topography is given).

Example of discrete topography file

The following is an example of a discrete topography file:

30	40	
1	1	5
1	2	5
2	2	6
$\vdots$	$\vdots$	$\vdots$
30	40	7

In this example the topography at $(1, 2)$ is *higher* than at $(2, 2)$ by one cell thickness. The mesh consists of 30 cells in the easting, 40 cells in the northing, and at least 7 cells in the vertical direction. If the topography is above the mesh, then $k = 0$.

3.2.5 Model file

This file contains the cell values of the model in conductivity or chargeability. The model file for DC data must have units of S/m. The chargeability model will follow the units associated with the choice of data. The forward, initial, and reference models must be in this format. Likewise,

any recovered model files will be in this format. Values above the given topography (i.e., air) will be ignored when *inputting* models into DCIP3D, but will automatically be set to 10^{-7} S/m for DC and -1 for IP models being *output* from DCIP3D. The outputted conductivity models will have a file extension of `.con` and the chargeability modes will be output with the extension of `.chg`. The following is the file structure of the model file for conductivity (the chargeability follows the same format)

$$
\begin{array}{|l|}
\hline
\sigma_{1,1,1} \\
\sigma_{1,1,2} \\
\vdots \\
\sigma_{1,1,NZ} \\
\sigma_{1,2,1} \\
\vdots \\
\sigma_{1,j,k} \\
\vdots \\
\sigma_{NN,NE,NZ} \\
\hline
\end{array}
$$

Each $\sigma_{i,j,k}$ is conductivity at the $[i,j,k]^{\text{th}}$ model cell.

$\sigma_{i,j,k}$ conductivity in cell $[i,j,k]$. The conductivity is always in S/m and is always positive. The chargeability may be zero, but not negative.

$[i,j,k] = [1,1,1]$ 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.

3.2.6 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 model file, and thus must be the same size, with one exception. Values of this file are restricted to either 0 or 1. By default, all cells below the earth's surface are active (1) and incorporated into the inversion. Inactive cells (set to 0) are set to the values of the reference model and influence the forward modelling; they *do not* influence the model objective function. Cells that are defined as active and are solved for within the inversion and are set to 1. The following is an example of an active cells file:

```
0
0   !  inactive cell
⋮
0
1   !  active cell
⋮
1
```

3.2.7 Bounds file

This file contains the cell values of the lower and upper bounds on the sought model. It is an optional for the inversion program. The bounds have the same dimension as the model. The following is the file structure of a bounds file:

$$
\begin{array}{cc}
\mathtt{b}^l_{1,1,1} & \mathtt{b}^u_{1,1,1} \\
\mathtt{b}^l_{1,1,2} & \mathtt{b}^u_{1,1,2} \\
\vdots & \vdots \\
\mathtt{b}^l_{1,1,NZ} & \mathtt{b}^u_{1,1,NZ} \\
\mathtt{b}^l_{1,2,1} & \mathtt{b}^u_{1,2,1} \\
\vdots & \vdots \\
\mathtt{b}^l_{i,j,k} & \mathtt{b}^u_{i,j,k} \\
\vdots & \vdots \\
\mathtt{b}^l_{NN,NE,NZ} & \mathtt{b}^u_{NN,NE,NZ}
\end{array}
$$

Parameter definitions:

$\mathtt{b}^l_{i,j,k}$ Is the lower bound on cell $[i, j, k]$.

$\mathtt{b}^u_{i,j,k}$ Is the 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 $NN \times NE \times NZ$, 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 lines must be ordered so that k changes the quickest (from 1 to `NZ`), followed by j (from 1 to `NE`), then followed by i (from 1 to `NN`). If the surface topography file is supplied, the bounds for cells above the surface will be ignored. It is recommended that these values be assigned a negative value (e.g. `-1.0`) to avoid confusion.

3.2.8 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 equation 12. For ease, the weights in geographic coordinates are provided by the user. The following is the file structure is for the weights file:

$$
\begin{array}{ccc}
\texttt{W.S}_{1,1,1} & \cdots & \texttt{W.S}_{NN,NE,1} \\
\vdots & \vdots & \vdots \\
\texttt{W.S}_{1,1,NZ} & \cdots & \texttt{W.S}_{NN,NE,NZ} \\
\texttt{W.E}_{1,1,1} & \cdots & \texttt{W.E}_{NN,NE-1,1} \\
\vdots & \vdots & \vdots \\
\texttt{W.E}_{1,1,NZ} & \cdots & \texttt{W.E}_{NN,NE-1,NZ} \\
\texttt{W.N}_{1,1,1} & \cdots & \texttt{W.N}_{NN-1,NE,1} \\
\vdots & \vdots & \vdots \\
\texttt{W.N}_{1,1,NZ} & \cdots & \texttt{W.N}_{NN-1,NE,NZ} \\
\texttt{W.Z}_{1,1,1} & \cdots & \texttt{W.Z}_{NN,NE,1} \\
\vdots & \vdots & \vdots \\
\texttt{W.Z}_{1,1,NZ-1} & \cdots & \texttt{W.Z}_{NN,NE,NZ-1}
\end{array}
$$

Parameter definitions:

$\texttt{W.S}_{i,j,k}$ Cell weights for the smallest model component (equation 12).

$\texttt{W.E}_{i,j,k}$ Cell weights for the interface perpendicular to the easting direction.

$\texttt{W.N}_{i,j,k}$ Cell weights for the interface perpendicular to the northing direction.

$\texttt{W.Z}_{i,j,k}$ Cell weights for the interface perpendicular to the vertical direction.

Within each part, the values are ordered in the same way as in model 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 topography file 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).

4 Running the programs

The software package DCIP3D uses five general codes:

- DCIPF3D: performs forward modelling for DC and IP data
- DCINV3D: inverts DC potentials to recover a 3D conductivity model using a Gauss-Newton method
- IPSEN3D: calculates the sensitivity matrix for the 3D IP inversion
- IPINV3D: inverts IP data to recover a 3D chargeability model
- MAKE_WDAT: makes a weighting file for smoothing near surface zones of the model

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:

$$\texttt{program arg}_1 \ [\texttt{arg}_2 \cdots \texttt{arg}_i]$$

where:

`program` is the name of the executable

$\texttt{arg}_i$ is a command line argument, which can be a name of corresponding required or optional file. Optional command line arguments are specified by brackets: `[]`. **NOTE:** Typing `-inp` as the control file, serves as a help function and returns an example input file.

Each control file contains a formatted list of arguments, parameters and file names 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. All files are in ASCII text format - they can be read with any text editor. Input and control files can have any name the user specifies. Details for the format of each file can be found in Section 3. The files associated with DCIPF3D are:

4.2 DCIPF3D

This program performs forward modelling of DC and IP data. Command line usage:

```
dcipf3d fwd.inp [nThread]
```

where nThread is an optional integer argument to specify the number of OpenMP threads to use for the parallelization. If this value is missing, DCIPF3D will use the maximum number of threads based on the processor. The input file, fwd.inp is described below.

4.2.1 Input files

Format of the control file fwd.inp:

```
type        !  input type:  dc | ip | ipL
mesh        !  mesh file
obs         !  observations file
model.con   !  conductivity model file
model.chg   !  chargeability model file
topo        !  topography file
pot         !  output potentials:  0 | 1
tol         !  solver tolerance
vec         !  Max # of vectors to store
```

type The choices are: dc for DC forward modelling, ip for IP forward modelling, or ipL for IP forward modelling using product of the sensitivity matrix and chargeability. When DC is chosen, the chargeability model is ignored, but the line still must be there.

mesh The 3D mesh.

obs The observation locations.

model.con A conductivity model or a single value denoted by using the flag VALUE (e.g., VALUE 0.001).

model.chg A chargeability model or a single value denoted by using the flag VALUE (e.g., VALUE 0.001).

topo Topography file or null for no topography (see section 3 for caveats with location and topography file types).

pot Output of potentials:
When pot=0, No output is performed.
When pot=1, The potentials of every cell are written to the file potentials_x.txt where x is the current electrode pair number. This file can be viewed in meshTools3D.

tol This value indicates how well the forward modelled system is solved (1e-5 is a standard input)

vec Specifies how solution vectors are to be stored in the computer's memory. Use `-1` to store all vectors in memory.

Example of fwd.inp

Example of an input file for DCIPF3D to model DC data that are given in general format and a discrete topography file:

```
dc          !  DC input type
mesh.msh    !  mesh file
obs.loc     !  observations file
model.con   !  conductivity model file
model.con   !  Not used but must fill the line
topo.idx    !  discrete topo file
0           !  do not output potentials
1e-5        !  solver tolerance
-1          !  Store all vectors in memory
```

4.2.2 Output files

The files created by DCIPFD are:

dc3d.dat The computed DC potential data.

ip3d.dat The computed IP data if the option ip is chosen.

ip3d_lin.dat The computed IP data if the option ipL is chosen.

obs.loc The surface electrode positions in three dimensions. Produced only if the surface locations format is used.

topo.idx The discrete topography file generated when a general topography and surface locations are used.

4.3 DCINV3D

This program performs the inversion of DC resistivity data. Command line usage:

```
dcinv3d dcinv.inp [nThread]
```

where nThread is an optional integer argument to specify the number of OpenMP threads to use for the parallelization. If this value is missing, DCINV3D will use the maximum number of threads based on the processor. The input file, dcinv.inp is described below.

4.3.1 Input Files

The format for the main input file has been altered to allow for bounds (the bounds line has been added). The new format for dcinv.inp is:

```
maxit, irest   !  max iteration | irest
mode, par      !  mode par
data           !  observations file
mesh           !  mesh file
topo           !  topography file | null
ini            !  initial model file | VALUE m | null
ref            !  reference model file | VALUE m | null
act            !  active cell file | null
bounds         !  bounds:  BOUNDS_NONE | BOUNDS_CONST bl bu | BOUNDS_FILE file
lengths        !  α_s, α_x, α_y, α_z | Le Ln Lz | null
wvltx          !  wavelet type | null | none
itol,eps       !  wavelet tolerance and epsilon
weight         !  3D weighting | null
idisk          !  idisk:  0 | 1
tol            !  solver tolerance
vec            !  Max # of vectors to store
```

maxit,irest Two integers containing the maximum number of Gauss-newton iterations to be performed (maxit) and how to start the inversion. There are two choices for irest:

1. irest=0 Begins the inversion from scratch
2. irest=1 Restarts the inversion from a previous iteration. *The files dcinv3d.con, dcinv3d.out, and dcinv3d.log must be present for this option.*

mode,par An integer specifying one of the two choices for determining the trade-off parameter (a real value):

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., $\psi_d^* = N$). the target misfit value is given by the product of par and the number of data, N. Normally, the value of par should be 1.0 if the correct standard deviation of error is assigned to each datum.
2. mode=2: the user inputs the trade off parameter as defined by par.
3. mode=3: the program calculates the trade-off parameter according to the L-curve criterion and par is ignored

data The DC observation locations (with standard deviations).

mesh The 3D mesh.

topo Topography file or `null` for no topography (see section 3 for caveats with location and topography file types).

ini A conductivity model file, a single value denoted by using the flag `VALUE` (e.g., VALUE 0.001), or `null` to set the initial model to the reference model value

ref A chargeability model file, a single value denoted by using the flag `VALUE` (e.g., VALUE 0.001), or `null` to have `DCINV3D` compute the best fitting half space to use the reference model (and initial model if null is chosen as its option)

act The active cells model file or null to use all model cells (below the topography) as active

bounds The bounds flag can be one of three values:

1. bounds=BOUNDS_NONE: Do not use bounds
2. bounds=BOUNDS_CONST: Two values to set bounds throughout the entire model: bl and bu for the lower and upper bounds
3. bounds=BOUNDS_FILE: The bounds file

lengths Coefficients for the each model component for the model objective function from equation 12. α's is the smallest model component. Coefficient for the derivative in the easting direction. α_y is the coefficient for the derivative in the northing direction. The coefficient α_z is for the derivative in the vertical direction.

If null is entered on this line, then the above four parameters take the following default values: $\alpha_s = 0.0001, \alpha_x = \alpha_y = \alpha_z = 1.0$. None of the alpha's can be negative and they cannot be all equal to zero at the same time.

NOTE: The four coefficients α_s, α_x, α_y and α_z can be substituted for three corresponding length scales L_x, L_y and L_z. To understand the meaning of the length scales, consider the ratios α_x/α_s, α_y/α_s and α_z/α_s. They generally define smoothness of the recovered model in each direction. Larger ratios result in smoother models, smaller ratios result in blockier models. The conversion from α's to length scales can be done by:

$$L_x = \sqrt{\frac{\alpha_x}{\alpha_s}};\ L_y = \sqrt{\frac{\alpha_y}{\alpha_s}};\ L_z = \sqrt{\frac{\alpha_z}{\alpha_s}} \tag{30}$$

where length scales are defined in meters. When user-defined, it is preferable to have length scales exceed the corresponding cell dimensions.

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 (and is used for the null option, while the others are provided for users' experimentation. If none is entered, the program does not use wavelet compression.

itol,eps An integer and a real number that specify how the wavelet threshold level is to be determined.

1. 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.

2. 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 (e.g. 5%) 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 Section 2.6 of this manual.

weight Name of the file containing weighting matrix. If null is entered, the default value of one is used for no extra weighting.

idisk Integer flag of zero or one to write the sensitivities to disk

1. idisk=0: Store the entire sensitivity matrix in memory. This option will be desired in almost all cases.
2. idisk=1: Access the sensitivity matrix from memory when needed

tol This value indicates how well the forward modelled system is solved (`1e-5` is a standard input)

vec Specifies how solution vectors are to be stored in the computer's memory. Use `-1` to store all vectors in memory.

Example of dcinv.inp

Below is an example of the input file dcinv.inp. It will start from scratch and stop after 40 iterations if the desired misfit is not achieved. The desired misfit is the number of data and the program will compute the trade-off parameter. The reference and initial models are the best fitting half space. There are bounds throughout the model with the lowest bound of 1e-8 S/m and the upper bound of 0.1 S/m.

```
40 0                    !  max iteration | irest
1 1.0                   !  mode par
dobs.dc                 !  observations file
mesh.msh                !  mesh file
topo.idx                !  topography file | null
null                    !  initial model file | VALUE m | null
null                    !  reference model file | VALUE m | null
null                    !  active cell file | null
BOUNDS_CONST 1e-8 0.1   !  bounds:  BOUNDS_NONE | BOUNDS_CONST bl bu | BOUNDS_FILE file
100 100 100             !  Le, Ln, Lz| null
daub2                   !  wavelet type | null | none
1 0.02                  !  wavelet tolerance and epsilon
null                    !  3D weighting | null
0                       !  idisk:  0 | 1
1e-5                    !  solver tolerance
-1                      !  Max # of vectors to store
```

4.3.2 Output Files

DCINV3D will create the following files:

1. dcinv3d.log The log file containing the minimum information for each iteration and summary of the inversion
2. dcinv3d.out The "developers" log file containing the values of the model objective function value(ψ_m), trade-off parameter (β), and data misfit (ψ_d) at each iteration including
3. dcinv3d_iter.con Conductivity model for each iteration (iter defines the iteration step)
4. dcinv3d_iter.pre Predicted data for each iteration (iter defines the iteration step)
5. dcinv3d.pre Predicted data file that is updated after each iteration (will also be the "final" predicted data)
6. dcinv3d.con Conductivity model that matches the predicted data file and is updated after each iteration (will also be the "final" recovered model)
7. sensitivity.txt Model file of average sensitivity values for the mesh
8. check_sign.txt This file will prompt the user to check the sign of specific observed potentials after brief data checks. It may or may not be created.

4.4 IPSEN3D

This program calculates the sensitivity matrix for the inversion of IP data. Command line usage:

```
ipsen3d ipsen.inp [nThread]
```

where nThread is an optional integer argument to specify the number of OpenMP threads to use for the parallelization. If this value is missing, IPSEN3D will use the maximum number of threads based on the processor. The input file, ipsen.inp is described below.

4.4.1 Input Files

The format for the IP sensitivity controle file, ipsen.inp, is:

```
data       !  observations file
mesh       !  mesh file
con        !  conductivity model file | VALUE m
topo       !  topography file | null
act        !  active cell file | null
wvltx      !  wavelet type | null | none
itol,eps   !  wavelet tolerance and epsilon
tol        !  solver tolerance
vec        !  Max # of vectors to store
```

data The IP observation locations (with standard deviations).

mesh The 3D mesh.

con A conductivity model file, a single value denoted by using the flag `VALUE` (e.g., VALUE 0.001). The name of this file will most likely be dcinv3d.con - the inverted conductivities from the DC resistivity data.

topo Topography file or `null` for no topography (see section 3 for caveats with location and topography file types).

act The active cells model file or null to use all model cells (below the topography) as active

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 (and is used for the null option, while the others are provided for users' experimentation. If none is entered, the program does not use wavelet compression.

itol,eps An integer and a real number that specify how the wavelet threshold level is to be determined.

1. 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.
2. 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 (e.g. 5%) 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 Section 2.6 of this manual.

tol This value indicates how well the forward modelled system is solved (`1e-5` is a standard input)

vec Specifies how solution vectors are to be stored in the computer's memory. Use `-1` to store all vectors in memory.

Example of ipsen.inp

Below is an example of the control file for the IP sensitivity matrix calculation. The active cell model is active.mod, and the conductivity model is the recovered model from DCINV3D. No vectors are written out to file and the wavelet has a relative error of 2%.

```
ipdata.dat    !  observations file
mesh.msh      !  mesh file
dcinv3d.con   !  conductivity model file | VALUE m
topo          !  topography file | null
active.mod    !  active cell file | null
daub2         !  wavelet type | null | none
1, 0.02       !  wavelet tolerance and epsilon
1e-5          !  solver tolerance
-1            !  Max # of vectors to store
```

4.4.2 Output Files

The program IPSEN3D outputs three files. They are:

1. ipinv3d.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 it's name is not adjustable for output (although it can be re-named). It is very large and may be deleted once the work is completed.

2. sensitivity.txt Output after running ipsen3d. It 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. Discrete topography file if the observations are surface format and the topography is general format.

4.5 IPINV3D

This program performs the inversion of induced polarization data. Command line usage:

```
ipinv3d ipinv.inp
```

for the control file ipinv.inp described below.

4.5.1 Input Files

The bounds for version 5.0 have NOT been added for IP data inversion. Positivity is enforced through the log-barrier method. The format for the main IP inversion input file is:

```
irest       !  Restart variable
mode, par   !  mode par
data        !  observations file
mtx         !  Sensitivity .mtx file
ini         !  initial model file | null
ref         !  reference model file | null
lengths     !  α_s, α_x, α_y, α_z | Le, Ln, Lz | null
weight      !  3D weighting | null
idisk       !  idisk:  0 | 1
```

irest An integer specifying how to start the inversion. There are two choices:

1. irest=0 Begins the inversion from scratch
2. irest=1 Restarts the inversion from a previous iteration. *The files ipinv3d.aux and ipinv3d.eta must be present for this option.*

mode,par An integer specifying one of the two choices for determining the trade-off parameter (a real value):

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., $\psi_d^* = N$). the target misfit value is given by the product of par and the number of data, N. Normally, the value of par should be 1.0 if the correct standard deviation of error is assigned to each datum.
2. mode=2: the user inputs the trade off parameter as defined by par.
3. mode=3: the program calculates the trade-off parameter using generalized cross validation (GCV) and par is ignored

data The DC observation locations (with standard deviations).

mtx A matrix file (.mtx) file from `IPSEN3D`.

ini A chargeability model file or `null` to set the initial model to 0.05 if IPTYPE=1 or 0.01 if IPTYPE=2. **The initial model must be non-zero.**

ref A chargeability model file or `null` to set the reference model to 0.

lengths Coefficients for the each model component for the model objective function from equation 12. α's is the smallest model component. Coefficient for the derivative in the easting direction. α_y is the coefficient for the derivative in the northing direction. The coefficient α_z is for the derivative in the vertical direction.

If null is entered on this line, then the above four parameters take the following default values: $\alpha_s = 0.0001, \alpha_x = \alpha_y = \alpha_z = 1.0$. None of the alpha's can be negative and they cannot be all equal to zero at the same time.

NOTE: The four coefficients α_s, α_x, α_y and α_z can be substituted for three corresponding length scales L_x, L_y and L_z. To understand the meaning of the length scales, consider the ratios α_x/α_s, α_y/α_s and α_z/α_s. They generally define smoothness of the recovered model

in each direction. Larger ratios result in smoother models, smaller ratios result in blockier models. The conversion from α's to length scales can be done by:

$$L_x = \sqrt{\frac{\alpha_x}{\alpha_s}};\ L_y = \sqrt{\frac{\alpha_y}{\alpha_s}};\ L_z = \sqrt{\frac{\alpha_z}{\alpha_s}} \tag{31}$$

where length scales are defined in meters. When user-defined, it is preferable to have length scales exceed the corresponding cell dimensions.

weight Name of the file containing weighting matrix. If null is entered, the default value of one is used for no extra weighting.

idisk Integer flag of zero or one to write the sensitivities to disk

1. idisk=0: Store the entire sensitivity matrix in memory. This option will be desired in almost all cases.
2. idisk=1: Access the sensitivity matrix from memory when needed

Example of ipinv.inp

This example of an IP inversion input file starts the inversion from scratch and performs GCV to find the trade-off parameter. The sensitivity matrix file was renamed to diffTol.mtx so the user new that they had used a different tolerance (and so they could switch to the other matrix file without re-running `IPSEN3D`). The initial model is set to null and depends upon the IP data type. The reference model was zero. Length scales were given to drive the recovered chargeabilities to more layered geometry. Additional weighting was applied through the file w.dat, supplied by the user.

```
0              !  Restart variable
3 1            !  mode par
ipNoisy.dat    !  observations file
diffTol.mtx    !  Sensitivity .mtx file
null           !  initial model
null           !  reference model file
100 100 10     !  αs, αx, αy, αz | Le Ln Lz | null
w.dat          !  3D weighting | null
0              !  idisk
```

4.5.2 Output Files

Seven types of output files are created by the program `IPINV3D`. They are:

1. ipinv3d.log The log file containing the minimum information for each iteration and summary of the inversion.

2. ipinv3d.aux An auxiliary file to allow the program to restart (Required for restart).
3. ipinv3d.eta Values of η so that the program can restart (Required for restart).
4. ipinv3d_iter.sus Chargeability files output after each iteration (iter defines the iteration step).
5. ipinv3d_iter.pre Predicted data files output after each iteration (iter defines the iteration step).
6. ipinv3d.pre Predicted data file that is updated after each iteration (will also be the "final" predicted data)
7. ipinv3d.chg Chargeability model that matches the predicted data file and is updated after each iteration (will also be the "final" recovered model)

4.6 MAKE_WDAT

4.6.1 Input Files

This is a utility used to make a w.dat file that has smoothing in the $x-$ and $y-$directions for the first few layers that underlie the topography surface. It suppresses the tendency of the algorithm to make highly variable structure in these top layers of the model.

NOTE: We stress that the above weighting should be applied with care. A strong horizontal smoothing can often eliminate horizontal changes in the conductivity, even if the earth model truly had these. The inclusion, and details, of the surface weighting therefore involves subjective decisions by the user. Without weighting, the model at the surface can be quite rough (due to electrode effects). On the other hand, local geology is also sometimes quite rough and hence the true earth model should exhibit a large heterogeneity.

The code is called by the command

```
make_wdat mkw.inp
```

for the control file mkw.inp described below.

```
mesh   !  mesh file
topo   !  Topography file | null
n      !  Number of layers to weight below topo
vals   !  n Values of weights
name   !  Discrete topo output name | null
```

mesh The 3D mesh file used in the DC or IP inversion

topo A topography file that is in the general format. Use null to apply to the top layers of the mesh.

n An integer specifying the number of layers to implement the weighting. The air layers will not be weighted and the remaining portion of the model will have a weight of 1.

vals Values of weights for the n layers specified on the previous line.

name Name of output file that writes the topography in the discrete format.

Example of mkw.inp

Here is an example of the MAKE_WDAT input file that places extra weights the first six layers beneath the topography. The remaining weights for the model are 1 and the air cells do not get weighted. Additionally, the program returns topo.idx that can be used with the general observations format.

```
mesh.msh          !  mesh file
topo.dat          !  Topography file
6                 !  Number of layers to weight below topo
64 32 16 8 4 2    !  n Values of weights
topo.idx          !  Topo output file name
```

4.6.2 Output Files

The program MAKE_WDAT outputs two files:

1. w.dat A 3D weighting file to use in the inversion
2. name The topography file in discrete format named from the last line in the input file. This file is not generated if the name line in the input file was null.

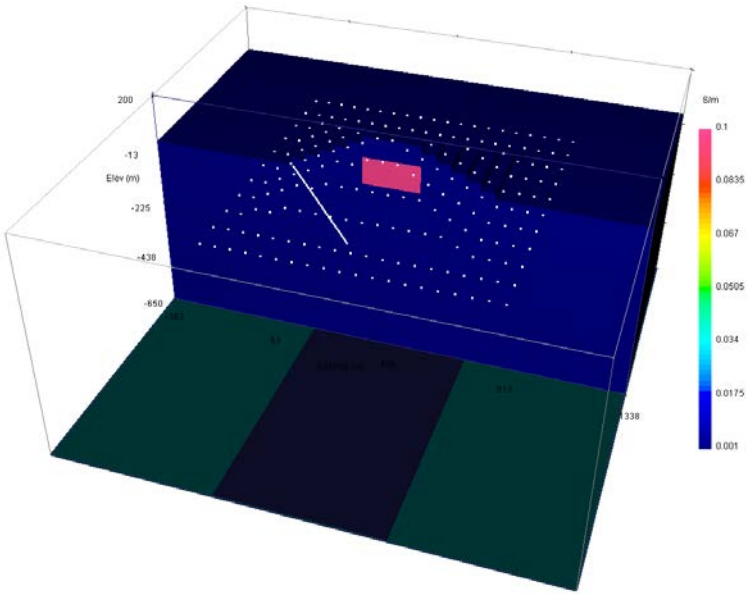


Figure 5: Pyramid topographic model of a used for DC forward modelling and the inversion examples. The block is 10 Ω-m (0.1 S/m) and the background is 1000 Ω-m (0.001 S/m). The colour scale is in S/m. The electrode locations are shown in white.

5 Examples

5.1 Introduction

In this section, we present forward modelling and inversion examples for both DC and IP data types. The examples are freely available for download 3 MB and can also be found on the DCIP3D website (without the `exe` files). The example is synthetic and is a 10 Ω-m (0.1 S/m) block in a 1000 Ω-m (0.001 S/m) half-space with a pyramid-shaped topography (Figure 5). For added complexity, surface data on a 25-m grid is simulated along with borehole data so that the general data format is required. The intrinsic chargeability of the block is 0.15 (Figure 6). The electrode locations are denoted by the white dots in both Figures. After forward modelling, 5% Gaussian noise is added to each data set to create the "observed" data sets. The uncertainties assigned are 5% of each datum with a 0.0001 floor (for both DC and IP). The last section shows the differences between the old code and version 5.0 with respect to the nodal-based finite difference scheme (shown in Figure 3). The mesh used for the example is $26 \times 26 \times 23$ and the file is given by

```
26 26 23
-362.5 -362.5 200.0
200 100 50 20*50.0 50 100 200
200 100 50 20*50.0 50 100 200
20*25.0 50 100 200
```

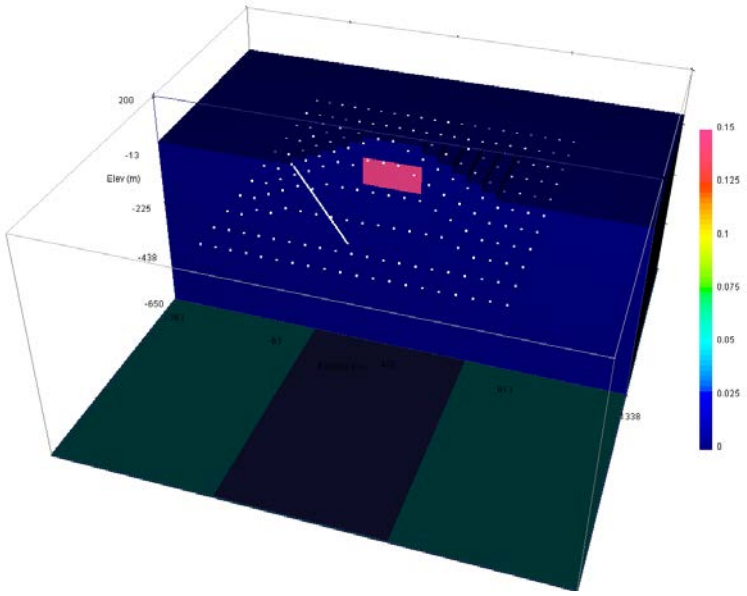


Figure 6: Pyramid topographic model of a used for IP forward modelling and the inversion examples. The intrinsic chargeability of the block is 0.15 and the background is not chargeable. The colour scale is in fractional percent chargeability. The electrode locations are shown in white.

5.2 Forward modelling

5.2.1 DC data

As an example of DC forward modelling, we consider the pyramid model (Figure 5). Using DCIPF3D, the conductive prism buried beneath a topographic high is modelled for the DC data. A pole-dipole survey with 50-m dipoles with $n = 1.5, 6.5$ was simulated. In order to run the forward model in this case, we must first convert our topo.dat, which is a general topo file to the discrete format by using the MAKE_WDAT utility. The command for this is

```
make_wdat changeTopo.inp
```

where changeTopo.inp is located in a folder labelled util was generated as

```
..\mesh.msh     !  mesh file
topo.dat        !  general topo file
4               !  nLevels
64 16 4 2       !  weights
discTopo.dat    !  discrete topo file | null
```

Since we will use the weighting file in an inversion example, we will go ahead and calculate weights for the top four layers of the model. If we were not interested in the weights, we could simply put $n = 1$ and use a value of 1.0. Then discard the file w.dat that has been created. The

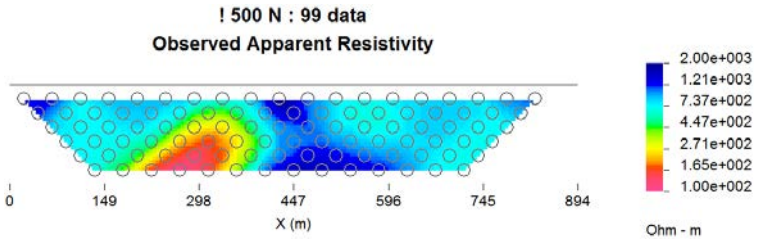


Figure 7: Forward modelled apparent resistivity for a line data across the center of the pyramid at 500 N for surface current locations only.

important step in this case is that the code has generated the discrete topography file discTopo.dat that will be used throughout the example.

Now, that we have our discretized topography, we can run the DC forward problem in the folder fwd with the command

```
dcipf3d dcFwd.inp
```

and the dcFwd.inp is

```
dc
..\mesh.msh            !  mesh file
pyramid.loc            !  observations file
trueModel.con          !  conductivity model file
ignored
..\util\discTopo.dat   !  topography file
0                      !  output potentials:  0 | 1
1e-5                   !  solver tolerance
-1                     !  Max # of vectors to store
```

where we have left the discrete topography in its util folder. Note that line 5 is ignored (the chargeability model), but something must be there. The file dc3d.dat is created (and will be overwritten if a file of this name already exists). No log file is created for the forward modelling, but the process took less than 2 sec on a PC with an 3.2Ghz Intel i7 processor with 12 threads used for OpenMP. The apparent resistivity for a line data across the center of the pyramid is shown in Figure 7. Only surface current electrodes were used to create the pseudo-section.

5.2.2 IP data

As an example of IP forward modelling, we again consider the pyramid model (Figure 5). Using DCIPF3D, the chargeable prism buried beneath a topographic high is modelled for the IP data. The same pole-dipole survey as the DC data with 50-m dipoles with $n = 1.5, 6.5$ was simulated. The locations file uses IPTYPE=1 so that we forward model apparent chargeabilities. If the general topo file topo.dat has not been converted, then we must first do that because of the general locations

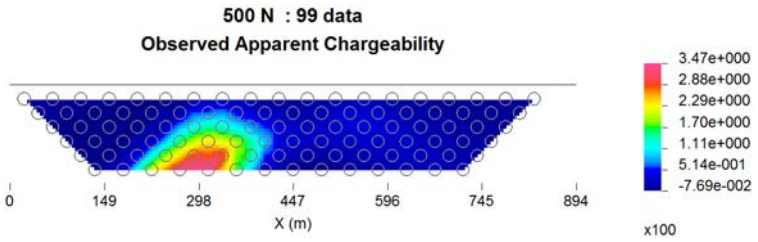


Figure 8: Forward modelled apparent chargeability for a surface line data across the center of the pyramid.

file. See the previous section (section 5.2.1) for details. The topography conversion only needs to be done once for a locations or observations file and mesh. If either of those elements change, then a new discrete topography file should be generated. Now, that we have our discretized topography, we can run the IP forward problem in the folder fwd with the command

```
dcipf3d ipFwd.inp
```

where the ipFwd.inp is

```
ip
..\mesh.msh            !  mesh file
pyramid.loc            !  observations file
trueModel.con          !  conductivity model file
trueModel.chg          !  Chargeability model file
..\util\discTopo.dat   !  topography file
0                      !  output potentials:  0 | 1
1e-5                   !  solver tolerance
-1                     !  Max # of vectors to store
```

where we have left the discrete topography in its util folder. The file ip3d.dat is created (and will be over-written if a file of this name already exists). No log file is created for the forward modelling, but the process took less than 4 sec on a PC with an 3.2Ghz Intel i7 processor with 12 threads used for OpenMP. The apparent resistivity for a line data across the center of the pyramid at 500 N is shown in Figure 8 using only the surface electrodes.

5.3 Inversion

5.3.1 DC data

The data created in section 5.2.1 are contaminated with 5% Gaussian noise. The uncertainties are estimated at 5% and a floor of 0.0001 Volts. In this case, we know the true data misfit that should be achieved is near the number of data, therefore we select mode=1 to have DCINV3D perform a line search to choose the appropriate trade-off parameter. The parameter will be 1.0 to signify

that the data misfit is n number of data. The minimum file requirements for DCINV3D is just the data file. In this case, we will choose not to use bounds. However, we will use the w.dat that was created when discretizing the topography. We will cap the inversion at 40 iterations. We will use the default reference model (best fitting half-space) and initial model (same as the reference model). The inversion starts from scratch so irest=0 is given.

The new incorporation of bounds is useful in real-life situations where there is down-hole measurements that often have additional conductivity measurements from well logs. Soft constraints, such as the reference model and weighting file would attempt to enforce the values the geophysicist has chosen from the well log information (which would need to be discretized onto the mesh). The user could also use hard constraints on the inversion by incorporating bounds or by using a reference model in combination of an active cells file where the cells located along the borehole are inactive (i.e., set to 0 in the active file).

The inversion of the DC data is run by using the command

```
dcinv3d dcinv.inp
```

where the input file, dcinv.inp looks like the following:

```
40 0                          !  maxit irest
1 1.0                         !  mode chifact
dc3d_noisy.dat                !  data
..\..\mesh.msh                !  mesh
..\..\util\discTopo.dat       !  topography
NULL                          !  initial model
NULL                          !  reference model
null                          !  active cell file
BOUNDS_NONE                   !  bounds
100.0, 100.0, 100.0           !  Le, Ln, Lz | null
daub2                         !  wavelet | null | none
1 0.02                        !  itol, eps | null
w.dat                         !  3D weighting | null
0                             !  idisk 0|1
```

The inversion required 16 iterations to converge to the target misfit. A truncated copy of the log file that summarizes the inversion is shown in Figure 9. It gives the specifics at each iteration such as the number of threads used for OpenMP, the achieved data misfit, the model objective function value, the trade-off paramter, and the accuracy of the wavelet transform. The inversion itself only took a few minutes. The final data misfit was 1,521 for the 1,584 data. Figure 10a shows the average sensitivity for each cell from the output sensitivity.txt. The recovered model is shown in 10b.

For reference, the inversion is performed without the weighting file w.dat so that the next to last line on the input file is null. The inversion converged in 12 iterations to the desired data misfit (again ≈ 1500). The recovered model is shown in Figure 11 on the same scale as the inversion result

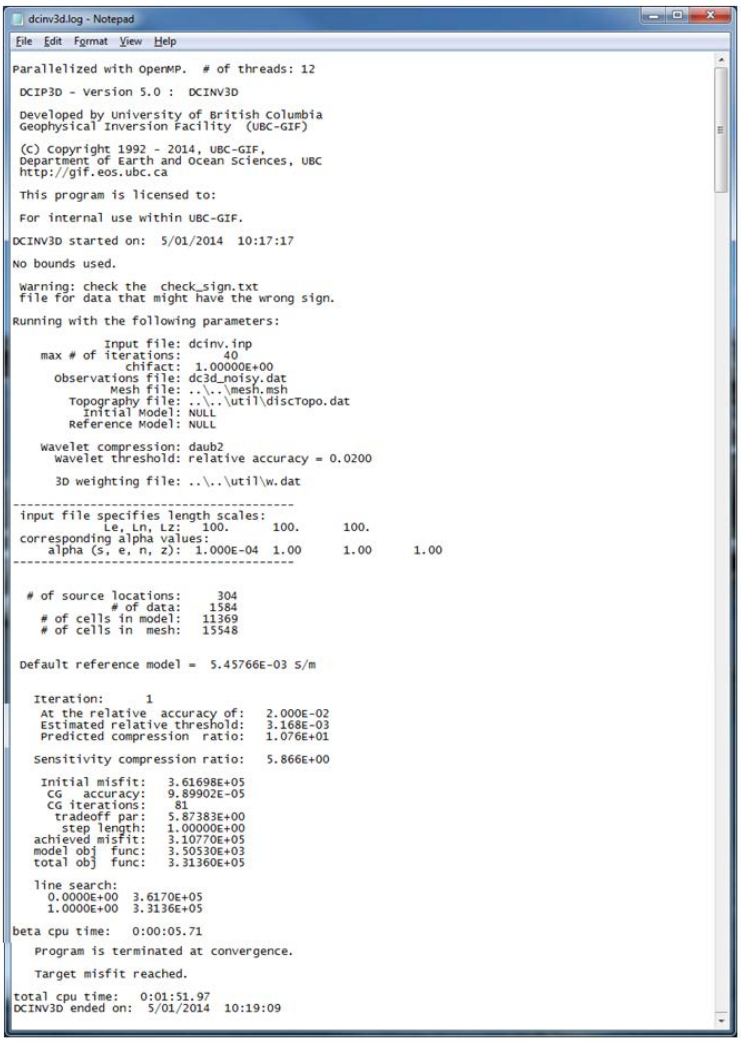
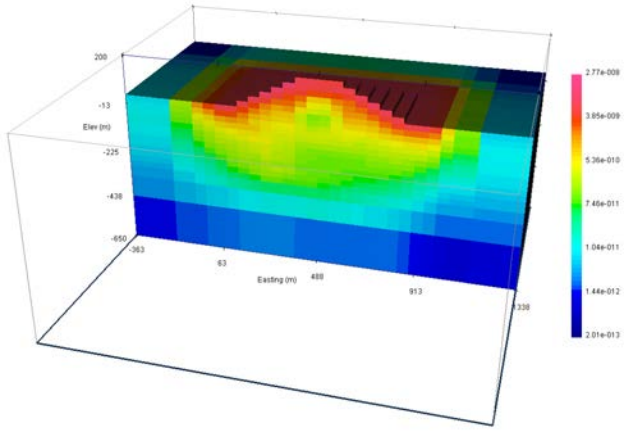
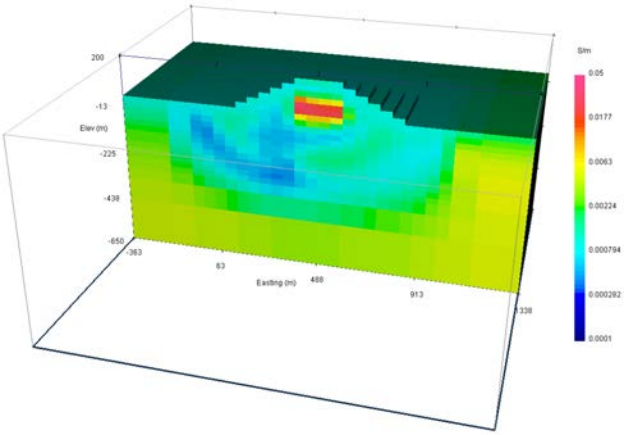


Figure 9: Truncated log file for DCINV3D showing the inputs of the inversion and specifics at each iteration. The bottom tells the user that the code stopped because it converged to the desired misfit.



(a)



(b)

Figure 10: (a) The average sensitivities in the DC inversion written to the file sensitivity.txt. (b) The inversion result (dcinv3d.con) in S/m after 16 iterations.

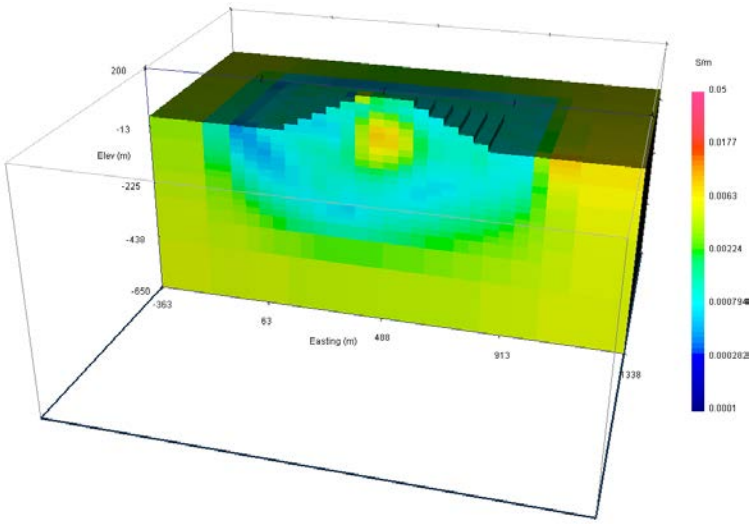


Figure 11: The inversion result (dcinv3d.con) in S/m after 12 iterations without the surface weighting file w.dat. The colour scale is consistent with Figure 10b.

with weighting. It has placed more conductive material at the surface so the main block is not as conductive.

5.3.2 IP data

The data created in section 5.2.2 are contaminated with 5% Gaussian noise. The uncertainties are estimated at 5% and a floor of 0.0001. We first must create the sensitivity matrix by running IPSEN3D. This is done via the command line:

```
ipsen3d ipsen.inp
```

and the sensitivity input file, ipsen.inp is

```
ip_noisy.dat                !IP data
..\..\mesh.msh              !  Mesh
dcinv3d.con                 !  Conductivity model
..\..\util\discTopo.dat     !  Discrete topo
null                        !  active cell file
daub2                       !  wavelet type
1 0.02                      !  wavelet param
1e-5                        !  solver tol
-1                          !  store vectors in memory
```

and where in this case the conductivity model dcinv3d.con is the result of the inversion of DC data using the surface weighting (it has been copied to the IP inversion folder). The result of running

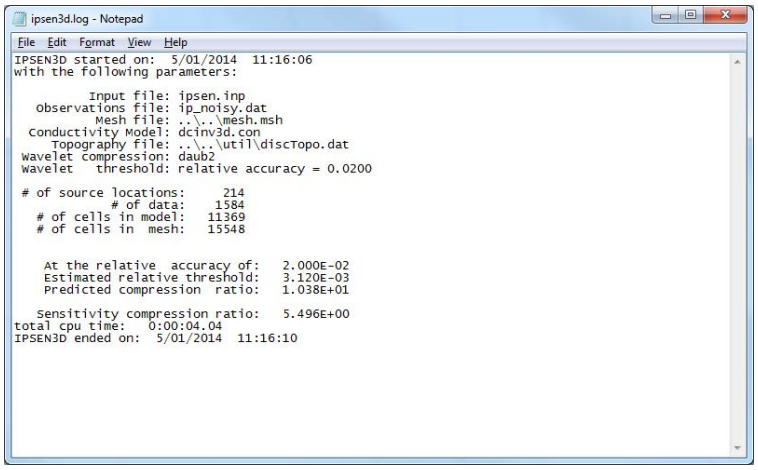


Figure 12: The log file for IPSEN3D showing the inputs used for the creation of the sensitivity matrix file.

the code is the creation of its log file, ipsen3d.log (Figure 12), the sensitivity matrix required for the inversion, ipinv3d.mtx, and the average sensitivities for the mesh, sensitivity.txt. NOTE: the latter will be written over if the IP inversion is performed in the same folder as the DC inversion. The average sensitivities are presented in Figure 14a.

As with the DC, we know the true data misfit that should be achieved is near the number of data, therefore we select mode=1 to have IPINV3D perform a line search to choose the appropriate trade-off parameter. We use the same length scales as the DC and will use the weight file w.dat as well. The initial model is set to null as well as the reference model. The program internally sets the initial model to 0.01 and the reference model to 0. The property idisk is set to 0 because the computer can handle storing the sensitivities is memory (this will be true in 99% of cases). The inversion is run by using the command:

```
ipinv3d ipinv.inp
```

where the input file ipinv.inp is

```
0                    !  irest
1 1.0                !  mode, par
ip_noisy.dat         !IP data
ipinv3d.mtx          !  Sensitivity matrix file
null                 !  initial model
null                 !  reference model
100 100 100          !  Length scales
..\..\util\w.dat     !  Weighting file
0                    !  Access matrix file from memory
```

The inversion finished in 2 iterations with a data misfit of 1579. The first iteration required 13 log barrier iterations and the second required 8. A truncated version of the log file produced

is shown in Figure 13, which gives the input file information, log barrier parameters, and CPU time. The recovered model is presented in Figure 14b and recovers the block with a maximum chargeability of 0.11 (the true chargeability was 0.15).

As with the DC, the inversion is again run without the weighting file so that the next to last line on the input file is `null`. The inversion converged in 2 iterations to the desired data misfit (again ≈ 1500). The recovered model is shown in Figure 15 on the same scale as the inversion result with weighting. Although not as stark in contrast as the DC, the model without weighting has a smaller chargeability at the location of the block and places slightly chargeable material at the surface.

5.4 Version comparison

In this section, we compare the differences between the original version and this current version of DCIP3D that allows arbitrary electrode locations. For consistency, we use pyramid model from the "Forward modelling" and "Inversion" examples section in this document. The noise and survey design added will be consistent with the previous example, however, we will only examine the surface DC data. The important changes for the current version are concerned with optimizing numerical computations. The major items include: no longer having to store the sensitivity matrix dcinv3d.mtx, and optimizing the way in which reciprocity is used in developing the sensitivity matrix. Generally, if the same electrode location is used more than once in a data file, its solution is stored for future use. These changes have no impact upon the final solution obtained by the inversion algorithm. Discussion of these, and some additional minor changes, are incorporated in the revised manual. In what follows, we focus upon the major implications of having the electrodes off of the nodal positions.

5.4.1 Forward modelling

Figure 5 shows a volume-rendered image that will be forward modelled. A pole-dipole survey with 50-m dipoles was simulated and the apparent resistivity for a line data across the center of the pyramid is captured. Figure 16a shows the result with current electrodes in the centers of cells. It is to be compared with 16b where the currents are at nodes. The two pseudosections are very similar. Difference occur primarily because the experiments are not identical. The pole-dipole data in Figure 16a correspond to $n = 1.5, 6.5$ while in Figure 16b the data are $n = 1, 6$. It can be seen that the two pseudosections are slightly shifted vertically with respect to each other, which should be the case.

5.4.2 Inversion

The data are inverted with DCIP3D v5.0 and a plot of the resistivities of the surface cells, and a cross-section of the recovered model, are shown in Figure 17. One characteristic of the inversion, irrespective of whether currents are on or off the nodes is that the model exhibits rough structure at the surface (or more generally, at current and potential electrode locations). The reason for this lies in the sensitivity values. The sensitivities for the DC resistivity (or IP problem) are highest at

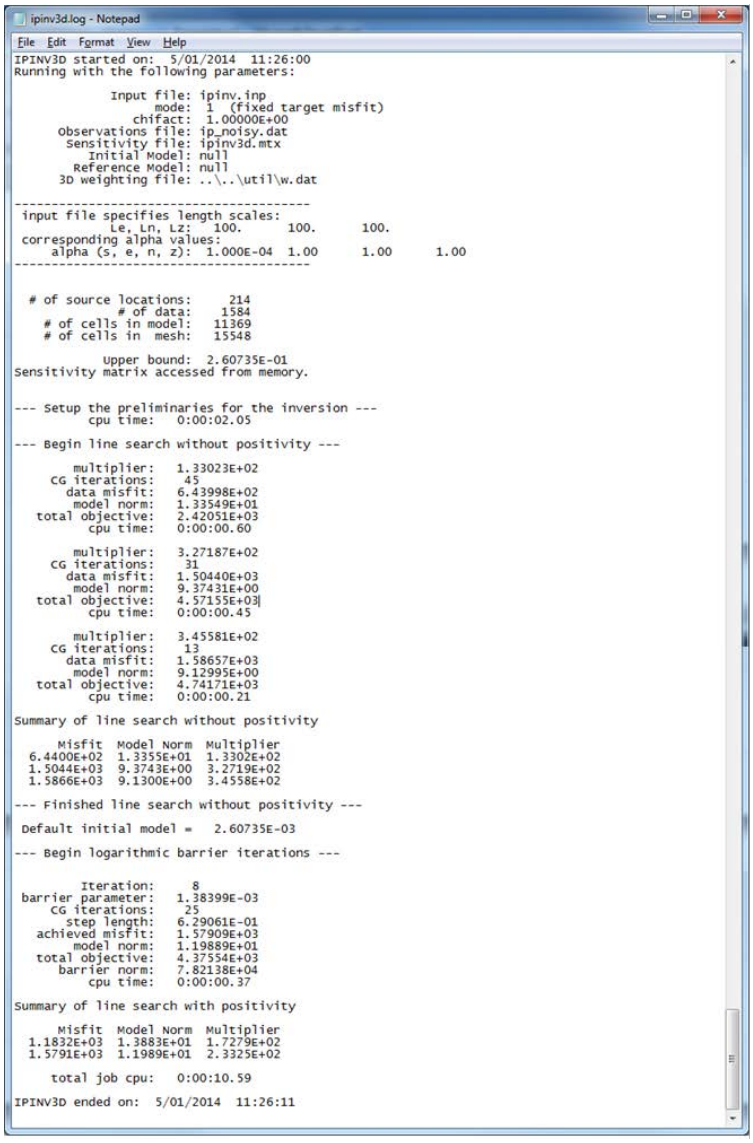
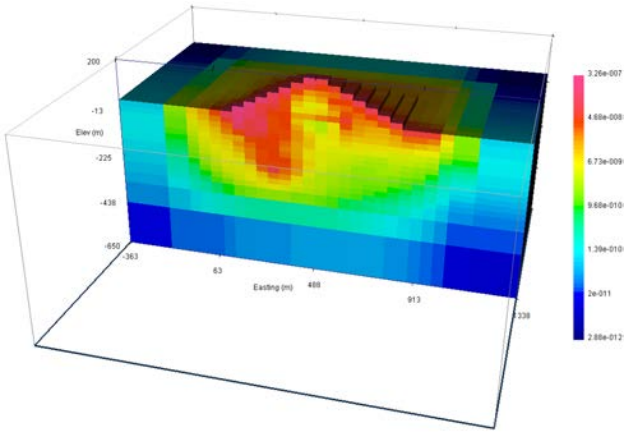
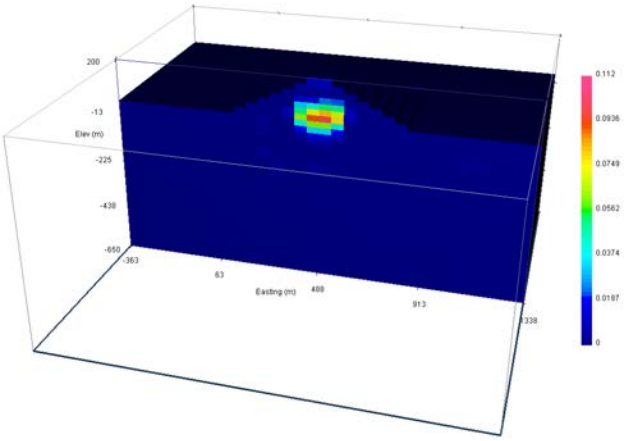


Figure 13: The truncated log file for IPINV3D showing the inputs of the inversion and specifics at each iteration. The bottom tells the user that the code stopped because it converged to the desired misfit.



(a)



(b)

Figure 14: (a) The average sensitivities in the IP inversion written to the file sensitivity.txt. (b) The inversion result (ipinv3d.chg) in fractional percent. The inversion required 2 log-barrier iterations.

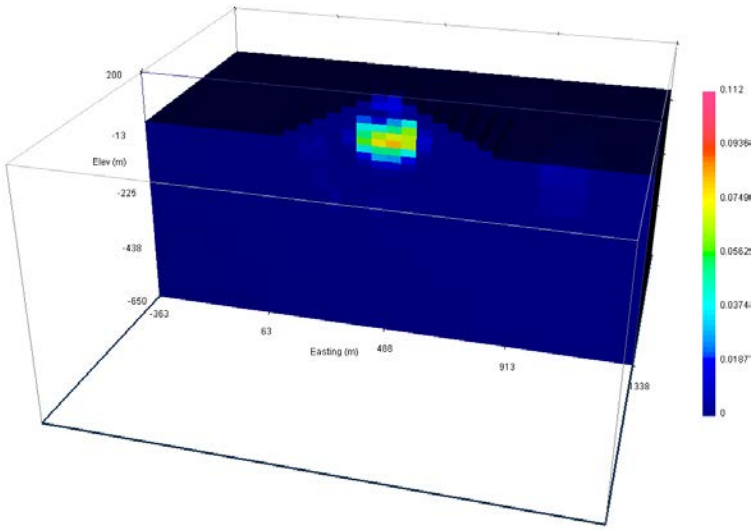


Figure 15: The inversion result (ipinv3d.chg) after 2 iterations without the surface weighting file w.dat. The colour scale is consistent with Figure 14b.

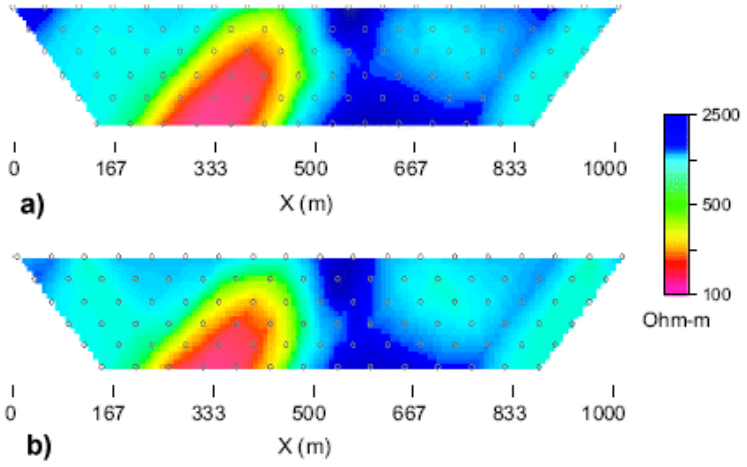


Figure 16: The forward modelled pseudosection with electrodes on (a) nodes and (b) cell centres.

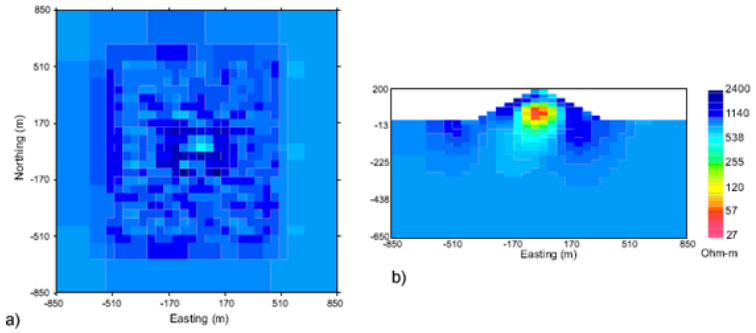


Figure 17: Inversion of the pyramid model (a) on the surface and (b) slice through the middle of the anomaly. Both current and potential electrodes are placed at cell centres.

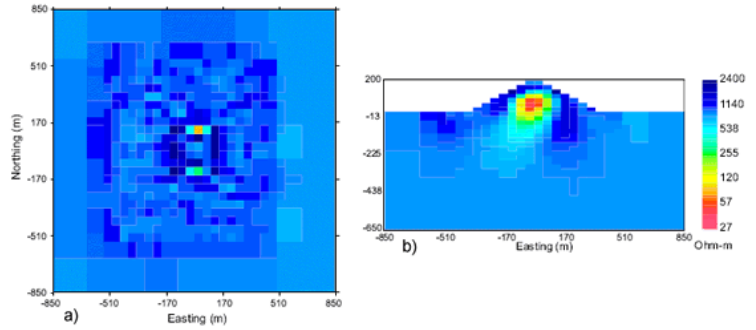


Figure 18: Inversion of the pyramid model (a) on the surface and (b) slice through the middle of the anomaly. Both current and potential electrodes are placed at cell nodes.

the location of an electrode (either a current or potential electrode). Changing the conductivity at a location closest to the electrode makes the greatest change in the measured datum and hence that is the preferential location to add conductivity. This effect is well understood and a number of GIF software programs have incorporated ways to ameliorate this "electrode" artifact, including the weighting file created from `MAKE_WDAT`. For reference, Figure 18 is the inversion from the previous versions of `DCIP3D` that use the nodal discretization.

In Figure 19a, a we plot the sensitivity due to a current at a nodal location and for a potential field value at a distance of 300 m from the current. This is a rough function which exhibits sign changes at the location of both electrodes. In the inverse problem the conductivity perturbation is made up of a linear combination of these sensitivities. This is one of the fundamental reasons why the first layer of surface cells often has considerable "chatter." We distribute the true current amongst neighboring electrodes and hence there is a superposition of the sensitivities shown in Figure 19a. In the case that the true current is in the center of the cell the sensitivity due to the modelled distributed currents, is that shown in Figure 19b.

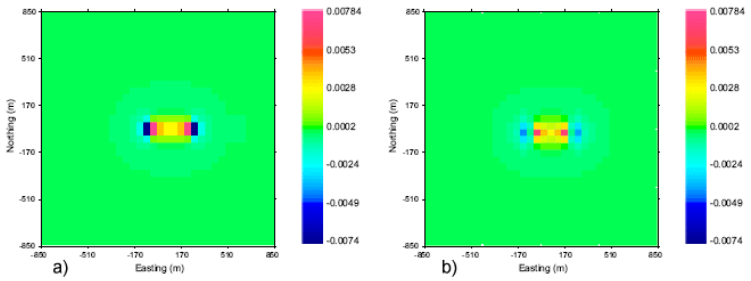


Figure 19: Sensitivities calculated from the pyramid example by placing the current electrodes at (a) the cell nodes and at (b) the cell centres.

6 References

Calamai, P. H., and J. J. Moré, 1987, Projected gradient methods for linearly constrained problems: Mathematical programming, **39**, 96–116.

Dey, A., and H. F. Morrison, 1979, Resistivity modeling for arbitrarily shaped three-dimensional structures: Geophysics, **44**, 753–780.

Hansen, P. C., 2000, The L-curve and its use in the numerical treatment of inverse problems: WIT Press.

Li, Y., and D. W. Oldenburg, 2003, Fast inversion of large-scale magnetic data using wavelet transforms and a logarithmic barrier method: Geophysics Journal International, **152**, 251–265.

———, 2010, Rapid construction of equivalent sources using wavelets: Geophysics, **75**, no. 3, L51–L59.

Siegel, H. O., 1959, Mathematical formulation and type curves for induced polarization: Geophysics, **24**, 547–565.

Vogel, C. R., 2002, Computational Methods for Inverse Problems (Frontiers in Applied Mathematics): Society for Industrial Mathematics.