DCIP2D

A Program Library for Forward Modelling and Inversion of DC/IP Data over 2D Structures

Version 5.0

Release date: July 2013

UBC - Geophysical Inversion Facility 1988 – 2016

Table of Contents

1 Package overview

1.1 Background

DCIP2D v5.0 is a program library for carrying out forward modelling and inversion of DC resistivity and induced polarization data in two dimensions. Version 1.0 of the code was first developed in 1992. That code, or a modified subsequent version, has been extensively used in mining and geotechnical communities and it has also been valuable as instructional software for learning about practical inverse theory. Improvements to the code however, have not kept pace with advances in optimization theory, scientific computing and computer architecture such as the prevalence of multi-core machines. To incorporate these advances, and renew DCIP2D v5.0 to a state-of-the-art inversion code, we are releasing Version 5.0, which has been modified and fully parallelized using OpenMP to be used on computers with hyper-threading capabilities. The following is a list of modifications implemented in DCIP2D:

- **Electrodes can be located anywhere in the mesh**: Previous versions of the code required electrodes to be on mesh nodes. While convenient for simple arrays and few data, this added complications for working in regions if high topography and with non-uniform surveys such as Schlumberger.

- **Arbitrary electrode arrays**: All linear surface array data, or combinations thereof, can be inverted. For instance Wenner data and pole-dipole data can now be inverted.

- **Borehole data can be inverted**: In the new version of DCIP2D we have added the capability to incorporate borehole data. These can be inverted individually or in conjunction with surface array data.

- **Parallelization using OpenMP**: When many transmitters are used the computations can be distributed over multiple processors. The parallelization is invoked whenever a forward modeling is carried out or when $\mathbf{J}$, the sensitivity matrix, is multiplied onto a vector which is required in the Conjugate Gradient (CG) solution.

- **Handling spatially large data sets**: Solution of the numerical optimization problem in the previous codes was carried out using a subspace technique. This worked well for typical exploration surveys. In some cases where the number of transmitters was very large, the subspace method had difficulties. In Version 5.0we overcome this by solving the Gauss-Newton equations using a Conjugate Gradient (CG) method. The CG formulation is the suggested method of solution but we have left the original subspace solution methodology in the code. This will allow users to reproduce results obtained with the older codes. Many of the new features of the codes, for example the Huber and Ekblom norms, bound constraints, active and inactive cells work only with the CG approach.

- **Huber norm for data misfit**: Outliers in the data will negatively impact the performance of the inversion algorithm. This can be ameliorated by working with robust norms for data misfit. The Huber norm, which can range from l_1 to l_2 has been implemented here.

- **Incorporation of geologic information via bound constraints**: A priori knowledge about the background conductivity or chargeability can be incorporated via bound constraints. That is, each cell is restricted to have a model value $m_j^{\text{low}} < m_j < m_j^{\text{high}}$, where the bounds m_j^{low} and m_j^{high} are prescribed by the user. The CG solution implements this through projected gradient techniques. This is faster than the interior point methodology used in the subspace method.

- **Incorporation of geologic information using the derivative terms in the model objective function**: The derivative terms in the model objective function can take the form:

  $$\psi_x = \int_\Gamma \left| \frac{\partial (m - m_o)}{\partial x} \right|^2 dv \tag{1}$$

  $$\psi_x = \int_\Gamma \left| \frac{\partial m}{\partial x} \right|^2 dv \tag{2}$$

  Similar equations can be written for the z derivatives. The difference is subtle but it has an important consequence in the inversion output. Consider a case where the a priori knowledge that a conductive overburden exists and a corresponding reference model has been generated. If the thickness of the overburden is known then the objective function with equation 1 should be used. The resultant model will have a boundary at the designated location. If the thickness is not known then equation 2 should be used. When this is done the reference model will appear only in the smallest model component ψ_s and the final model may show the discontinuity at a location different from that in the reference model. The capability of using the reference model in the derivative terms in these two ways is incorporated into Version 5.0. The choice of which objective function to use also arises with reference models generated from borehole information.

- **Incorporation of a priori information using active and inactive cells**: Considerable flexibility in constructing different types of solutions is afforded by being able to have certain cells “inactive.” This can reduce the size of the inverse problem and also be used to incorporate a priori knowledge about the model. Inactive cells are used in the forward modeling but they are held fixed in the inversion. Again, there are choices to be made regarding whether the values of the inactive cells should, or should not, affect the neighboring cells. Different models are obtained from the two options.

- **Mesh builder**: The forward modelling requires that an underlying mesh be built and evaluated. An updated GUI is provided to help accomplish this job. The validity of the mesh is evaluated by computing the apparent resistivity for a homogenous halfspace.

- **Sensitivity Matrix**: A cumulative sensitivity matrix is generated. This is insightful for survey design and also it can be used as proxy for depth of investigation.

- **Depth of investigation**: The forward problem is numerically solved on a large domain but the data are only sensitive to a portion of that volume. Structure appearing at depth and outside the lateral extent of the data array is controlled only by the model objective function. As such it is best removed for final presentation. The useful region of investigation can be estimated using the DOI process (Oldenburg and Li, 1999) or by using the cumulative sensitivity analysis.

- **Selecting wave numbers**: Working in 2D requires that Maxwell's equations are solved in the wave number domain and a cosine transform is applied to obtain data in space. In previous codes the number and value of wave numbers were hardwired into the code. Default values are still generated but for highly unusual data sets there may be a reason to explore the dependence of solution accuracy with wave number selection.

1.2 DCIP2D v5.0 capabilities and limitations

The package that can be licensed includes the following components:

1.2.1 Array types

All linear survey surface-array types, including non-standard or un-even arrays, as well as their combinations can be inverted. The GUI is mainly designed to handle the most commonly used array types and therefore it works well with dipole-dipole, pole-dipole, pole-pole and Wenner or RealSection arrays. Borehole data can be inverted but the user interface does not yet support borehole array types.

1.2.2 The Earth model

Inversion for a 2D model of the earth implies that data were gathered along a survey line at the surface. DCIP2D v5.0 considers the subsurface in terms of a mesh of rectangular cells. Numerical accuracy in increased by using smaller cells but this also increases the size of the problem. here are usually at least three cells between electrode positions and the discretized volume extends well beyond the data area.

1.3 DCIP2D program library content

1. **Executable programs** for performing forward modelling and inversion of 2D DC resistivity or induced polarization surveys. The DCIP2D library (Windows or Linux platforms) consists of the programs:

   - DCIPF2D: calculates DC resistivity and/or IP data based on a given model of the earth.
   - DCINV2D: inverts 2D DC resistivity data
   - IPINV2D: inverts 2D IP resistivity data

2. **Graphical user interfaces** are supplied for the Windows platforms *only*. Facilities include:

   - DCIP2D-GUI: a primary interface for setting up the inversion and monitoring progress.

- DCIP2D-DATA-VIEWER: a utility for viewing raw data, error distributions, and for comparing observed to predicted data directly or as difference maps.
- DCIP2D-MODEL-VIEWER: a utility for displaying resulting 2D models, and graphs of convergence behaviour.
- DCIP2D-MODEL-MAKER: a separate utility for building synthetic 2D models and then running forward modelling to produce a synthetic data set.

3. **Example data sets and excercises** can be provided upon request.

2 Background theory

2.1 Introduction

This section presents theoretical background, numerical examples, and explanation for implementing the program library DCIP2D v5.0. This suite of algorithms, developed at the UBC-Geophysical Inversion Facility, is needed to invert DC potentials and IP responses over a 2-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:

$$\phi_\sigma = \mathcal{F}_{dc}[\sigma], \tag{3}$$

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

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

and also by appropriate boundary conditions. In equation 4, 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 4 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 4 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:

$$\phi_\eta = \mathcal{F}_{dc}[\sigma(1 - \eta)], \tag{5}$$

or

$$\nabla \cdot (\sigma(1 - \eta)\nabla \phi_\sigma) = -\mathbf{I}\delta(r - r_s). \tag{6}$$

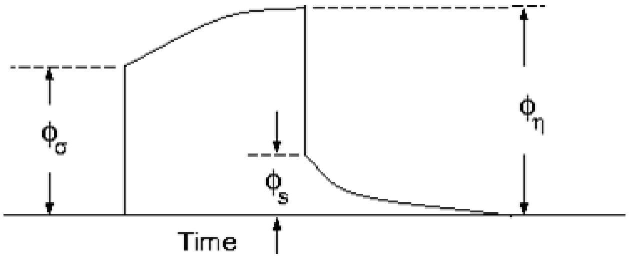


Figure 1: Definition of the three potentials associated with DC/IP experiments.

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

$$\eta_a = \frac{\phi_s}{\phi_\eta} = \frac{\phi_\eta - \phi_\sigma}{\phi_\eta}, \tag{7}$$

or

$$\eta_a = \frac{\mathcal{F}_{dc}[\sigma(1-\eta)] - \mathcal{F}_{dc}[\sigma]}{\mathcal{F}_{dc}[\sigma(1-\eta)]}. \tag{8}$$

Equation 8 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 $\eta(x,y,z)$.

The distribution of conductivity and chargeability in the earth can be extremely complicated. Assuredly earth structure is 3D, but for the DC/IP codes developed here we restrict ourselves to 2D structures and assume that the survey has been carried out along a traverse that is perpendicular to strike. The cross-section of the earth is divided into rectangular prisms each having a constant value of conductivity and chargeability.

2.2 Forward modelling

The forward modelling for the DC potentials and IP apparent chargeabilities and secondary potentials is accomplished using a finite difference technique to solve equation 4. The program which performs this calculation is DCIPF2D. 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

$$\mathbf{d}_{ip} = \mathbf{J}_{ip}\eta, \tag{9}$$

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

$$\mathbf{J}_{ip} = -\frac{\partial \ln \phi_\eta}{\partial \ln \sigma} = -\frac{1}{\sigma_\eta}\frac{\partial \phi_\eta}{\partial \ln \sigma} = -\frac{1}{\mathbf{d}_{dc}}\mathbf{J}_{dc}, \tag{10}$$

given DC data, $\mathbf{d}_{dc}$. Forward modeling using equation 10 is further explained in the section 2.5.

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. Next, we invert IP data to recover the chargeability $\eta(x, z)$. Because chargeabilities are usually small quantities ($\eta < 0.3$) it is possible to linearize equation 8 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.

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 reconstructing the chargeability section.

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 $\mathbf{d}^{obs} = (d_1^{obs}, d_2^{obs}, ..., d_n^{obs})^T$. 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.

Alternatively, if we fit the data too poorly then information about the conductivity that is coded in the data will not have been recovered. Our objective therefore is to neither under-fit nor over-fit the data. Rather, we want to find a model that reproduces the data only to within an amount that is justified by the estimated uncertainty in the data. To accomplish this we introduce a global misfit criterion:

$$\psi_d = \|\mathbf{W}_d(\mathbf{Gm} - \mathbf{d})\|^2 . \tag{11}$$

where $\mathbf{W}_d$ is a data weighting matrix. In this work, we shall assume that the noise contaminating the ith observation is an uncorrelated Gaussian random variable having zero mean and standard deviation ϵ_i. As such, an appropriate form for the $N \times N$ matrix is $\mathbf{W}_d = diag\,\{1/\epsilon_1, \ldots, 1/\epsilon_n\}$. With this choice, ψ_d is the random variable distributed as chi-squared with N degrees of freedom. Its expected value is approximately equal to N and accordingly, ψ_d^*, the target misfit for the inversion, should be approximately equal to this value.

It is common to use an l_2 norm measure of data fit as shown in equation 11. However, the Huber norm (Huber, 1964) has been incorporated to handle outliers in the data. The general form of the Huber norm is

$$\tau(y) = \begin{cases} y^2 & |y| \le c \\ 2c|y| - c^2 & |y| > c. \end{cases} \tag{12}$$

From equation 12, let $y = \mathbf{W}_d(\mathbf{Gm} - \mathbf{d})$ and the data misfit function then becomes

$$\Phi_d = \sum_{i=1}^{n} \begin{cases} \left[{\mathbf{W}_d}^i(\mathbf{G}_i\mathbf{m} - d_i)\right]^2 & |y_i| \le c \\ 2c|{\mathbf{W}_d}^i(\mathbf{G}_i\mathbf{m} - d_i)| - c^2 & |y_i| > c. \end{cases} \tag{13}$$

where c is a constant that separates the elements of vector y into those considered large and those that are considered small (Farquharson and Oldenburg, 1998).

Earth conductivity distributions are complex. To allow maximum flexibility to produce a model of arbitrary shape it is important that M, the number of cells representing the model, is large. In our inversions, M will almost always be greater than N, the number of data. The inverse problem therefore reduces to finding a set of M model parameters using only N data constraints under the condition that $M > N$. Clearly the solution is no unique and this non-uniqueness represents the principle obstacle for obtaining unambiguous information about earth structure from the observations.

Any inversion algorithm (if it works) will produce a model, which reproduces the data. But there are infinitely many possible models. So which one does the algorithm produce? It is not good practice to let the program make a random selection. Rather, a responsible approach is to direct the inversion algorithm to produce a model that is geologically reasonable and is constrained by additional information if such information is available. This can be implemented by formulating a "model objective function" which, when minimized, produces a model with desirable characteristics. The critical aspect of the inversion is therefore to form the model objective function which we characterize by ψ_m. To do this, the user must ask the question "what type of model is desired?" Should the model be smooth or should it be blocky? Is there a reference or background model that the constructed model should emulate? If there is a reference model, is it better known in some places than others so that the constructed model should be close to the reference model in certain locations but can depart from our preconceived ideas in other areas? Whatever the answer to these questions, a guiding philosophy should always be to find a model which (in some sense) is "as simple as possible." The non-uniqueness inherent in the inversion generally means that we can generate models which are arbitrarily complicated. We cannot however, make models that are arbitrarily simple. For example, a half space will generally not reproduce data acquired from a geophysical survey.

In the inversion algorithms in DCIP2D, our choice for the objective function ψ_m is guided by a desire to find a model which has minimum structure in the vertical and horizontal directions and at the same time is close to a reference model m_o. To accomplish this, we minimize a discretized approximation to

$$\begin{aligned} \psi_m(m, m_o) = \quad & \alpha_s \int\int w_s(x,z)(m - m_o)^2 dxdz + \\ & \int\int \left\{ \alpha_x w_x(x,z) \left(\frac{\partial(m-m_o)}{\partial x}\right)^2 + \alpha_z w_z(x,z) \left(\frac{\partial(m-m_o)}{\partial z}\right)^2 \right\} dxdz \end{aligned} \tag{14}$$

In equation 14, the functions w_s, w_x, w_z are specified by the user and the constant α_s controls the importance of closeness of the constructed model to the reference model m_o and α_x, α_z controls the smoothness of the model in the two directions. Varying the ratio α_x/α_z allows the construction of models that are smoother, thus more elongated, in either $x-$ or $z-$direction. The discrete form of

14 is the following:

$$\begin{aligned}\psi_m \quad &= (\mathbf{m}-\mathbf{m}_o)^T \left\{\alpha_s \mathbf{W}_s^T\mathbf{W}_s + \alpha_x \mathbf{W}_x^T\mathbf{W}_x + \alpha_z \mathbf{W}_z^T\mathbf{W}_z\right\}(\mathbf{m}-\mathbf{m}_o), \\ &\equiv (\mathbf{m}-\mathbf{m}_o)^T\mathbf{W}_m^T\mathbf{W}_m(\mathbf{m}-\mathbf{m}_o)^T, \qquad (15) \\ &= \|\mathbf{W}_m(\mathbf{m}-\mathbf{m}_o)\|^2 . \qquad (16)\end{aligned}$$

If w_s, w_x, and w_z are set equal to unity, then $\mathbf{W}_s$ is a diagonal matrix with elements $\sqrt{\Delta x \Delta z}$, where Δx is the length of the cell and Δz is its thickness, $\mathbf{W}_x$ has elements $\sqrt{\Delta z/dx}$ where dx is the distance between the centres of horizontally adjacent cells, and $\mathbf{W}_z$ has elements $\sqrt{\Delta x/dz}$ where dz is the distance between the centres of vertically adjacent cells.

For blockier models, we have incorporated the measure proposed by Ekblom (1973; 1987) that has been found to be useful. The generalized version is given as

$$\tau(y) = (y^2+\epsilon^2)^{\frac{\rho}{2}}, \qquad (17)$$

where ϵ is some positive number. The smaller ϵ becomes, the measure tends towards the l_ρ norm. Large values of ϵ tend the measure to behave like a scaled sum-of-squares. For the model objective function in equation 16, $y = \mathbf{W}_m(\mathbf{m}-\mathbf{m}_o)$ and the system of equations is solved with the projected gradients through a chi-factor regularization. The resulting model objective function is

$$\begin{aligned}\psi_m \quad &= \left[(\mathbf{m}-\mathbf{m}_o)^T\alpha_s\mathbf{W}_s^T\mathbf{W}_s(\mathbf{m}-\mathbf{m}_o)+\epsilon^2\right]^{\frac{\rho}{2}} + \left[(\mathbf{m}-\mathbf{m}_o)^T\alpha_x\mathbf{W}_x^T\mathbf{W}_x(\mathbf{m}-\mathbf{m}_o)+\epsilon^2\right]^{\frac{\rho}{2}} \\ &+ \left[(\mathbf{m}-\mathbf{m}_o)^T\alpha_z\mathbf{W}_z^T\mathbf{W}_z(\mathbf{m}-\mathbf{m}_o)+\epsilon^2\right]^{\frac{\rho}{2}} . \qquad (18)\end{aligned}$$

Details of the Eklom norm within the context of geophysical inversion can be found in Farquharson and Oldenburg (1998).

It should be noted that in equation 16, the reference model can be removed from the spatial (x and z) components. The effect is that the reference model places emphasis on the magnitude of the model, but its spatial variations do not influence the spatial derivatives. The model objective function becomes

$$\psi_m = (\mathbf{m}-\mathbf{m}_o)^T\left(\alpha_s\mathbf{W}_s^T\mathbf{W}_s\right)(\mathbf{m}-\mathbf{m}_o) + \mathbf{m}^T\left\{\alpha_x\mathbf{W}_x^T\mathbf{W}_x + \alpha_z\mathbf{W}_z^T\mathbf{W}_z\right\}\mathbf{m} \qquad (19)$$

and for the Ekblom norm

$$\begin{aligned}\psi_m \quad &= \left[(\mathbf{m}-\mathbf{m}_o)^T(\alpha_s\mathbf{W}_s^T\mathbf{W}_s)(\mathbf{m}-\mathbf{m}_o)+\epsilon^2\right]^{\frac{\rho}{2}} \\ &+ \left[\mathbf{m}^T(\alpha_x\mathbf{W}_x^T\mathbf{W}_x)\mathbf{m}+\epsilon^2\right]^{\frac{\rho}{2}} + \left[\mathbf{m}^T(\alpha_z\mathbf{W}_z^T\mathbf{W}_z)\mathbf{m}+\epsilon^2\right]^{\frac{\rho}{2}} . \qquad (20)\end{aligned}$$

This is a new feature in DCIP2D and gives the user greater flexibility.

The inverse problem is now properly formulated as an optimization problem:

$$\begin{aligned}&\text{minimize } \psi_m(\mathbf{m},\mathbf{m}_o) &&= \|\mathbf{W}_m(\mathbf{m}-\mathbf{m}_o)\|^2 \qquad (21) \\ &\text{subject to } \psi_d(\mathbf{d},\mathbf{d}^{obs}) &&= \left\|\mathbf{W}_d(\mathbf{d}-\mathbf{d}^{obs})\right\|^2 = \psi_d^*.\end{aligned}$$

In equation 21, $\mathbf{m}_o$ is a starting model and $\mathbf{W}_m$ is a general weighting matrix which is designed so that a model with specific characteristics is produced. The minimization of ψ_m yields a model that is close to $\mathbf{m}_o$ with the metric defined by $\mathbf{W}_m$ and so the characteristics of the recovered model are directly controlled by these two quantities. If the data errors are Gaussian and their standard deviations have been adequately estimated then the target misfit should be $\psi_d^* = N$. The data misfit function can take the form of the l_2 norm as shown above or the Huber norm from equation 13.

2.4 Inversion of DC data

The inversion of the apparent resistivity data is carried out using the program DCINV2D. The inversion of DC resistivity data formulated as the minimization in equation 21 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 21

$$\psi(\mathbf{m} + \delta\mathbf{m}) = \left\|\mathbf{W}_d\left(\mathcal{F}_{dc}[\mathbf{m}^{(n)}] + \mathbf{J}\delta\mathbf{m} - \mathbf{d}\right)\right\|^2 + \beta\left\|\mathbf{W}_m\left(\mathbf{m} + \delta\mathbf{m} - \mathbf{m}_o\right)\right\|^2 + H.O.T., \qquad (22)$$

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

$$\mathbf{J} = \frac{\partial d_i}{\partial m_j} = \frac{\partial \phi_i}{\ln \sigma_j}. \qquad (23)$$

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

$$\left(\mathbf{J}^T\mathbf{J} + \beta\mathbf{W}_m^T\mathbf{W}_m\right)\delta\mathbf{m} = -\mathbf{J}^T\left(\mathcal{F}_{dc}[\mathbf{m}^{(n)}] - \mathbf{d}\right) - \beta\mathbf{W}_m^T\mathbf{W}_m\left(\mathbf{m}^n - \mathbf{m}_o\right). \qquad (24)$$

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

$$\mathbf{m}^{(n+1)} = \mathbf{m}^{(n)} + \gamma\delta\mathbf{m}, \qquad (25)$$

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

2.5 Inversion of IP data

To invert IP data, we first linearize equation 7. Let η_j and σ_j denote the respective chargeability and electrical conductivity of the jth cell. Linearizing the potential ϕ_η about the conductivity model σ yields:

$$\phi_\eta = \phi(\sigma - \eta\sigma) = \phi(\sigma) - \sum_{j=1}^{M}\frac{\partial\phi}{\partial\sigma_j}\eta_j\sigma_j + H.O.T. \qquad (26)$$

The above equation is then substituted into equation 7:

$$d = \frac{\phi_\eta - \phi_\sigma}{\phi_\eta} = \frac{-\sum\limits_{j=1}^{M} \frac{\partial \phi}{\partial \sigma_j} \eta_j \sigma_j}{\phi(\sigma) - \sum\limits_{j=1}^{M} \frac{\partial \phi}{\partial \sigma_j} \eta_j \sigma_j}. \tag{27}$$

This can be approximately written as

$$d = -\sum_{j=1}^{M} \frac{\sigma_j}{\phi} \frac{\partial \phi}{\partial \sigma_j} \eta_j = -\sum_{j=1}^{M} \frac{\partial \ln \phi}{\partial \ln \sigma_j} \eta_j, \tag{28}$$

and therefore the ith datum is

$$d_i = \sum_{j=1}^{M} \mathbf{J}_{ij} \eta_j, \tag{29}$$

where

$$\mathbf{J}_{ij} = -\frac{\partial \ln \phi_i[\sigma]}{\partial \ln \sigma_j} \tag{30}$$

is the sensitivity matrix. Our inversion problem is formulated as

$$\begin{aligned} \text{minimize} \qquad & \psi_m = \|\mathbf{W}_m(\eta - \eta_o)\|^2 \\ \text{subject to} \quad & \psi_d = \left\|\mathbf{W}_d(\mathbf{J}\eta - \mathbf{d}^{obs})\right\|^2, \end{aligned} \tag{31}$$

where ψ_d^* is a target misfit. In reality the true conductivity σ is unknown and so we use the conductivity recovered from the inversion of the DC resistivity data to construct the sensitivity matrix elements in equation 30.

The functional in equation 31 can be minimized directly but we need to ensure that the recovered chargeability is positive. In the inversion of the DC potentials to recover the conductivity we ensured positivity by working with $\ln(\sigma)$ as the model in the inversion and applying the model norm to this quantity. This is justified, since conductivity varies over many orders of magnitude and it is the variation of conductivity that is diagnostic of earth structure. Intrinsic chargeability is confined to the region $[0, 1)$. Moreover, we are not generally interested in the variation of chargeability in the range between zero and some small number (e.g., 0.01). Working with logarithmic values however, puts undue emphasis on these small values. An efficient method by which to solve the linear inverse problem with positivity constraints is through a non-linear mapping of variables. More details of the IP inversion algorithm can be found in Oldenburg and Li (1994).

3 Elements of the program DCIP2D

3.1 Introduction

The `DCIP2D v5.0` program library consists of the programs:

1. `DCIPF2D`: performs forward modelling for DC and IP data
2. `DCINV2D`: inverts DC potentials to recover a conductivity model
3. `IPINV2D`: inverts IP data to recover a chargeability model

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

3.2 General files for DCIP2D v5.0 programs

There are six general files which are used in DCIP2D v5.0. These are:

1. observation: specifies the observed measurements and the associated electrode locations
2. electrodes: contains the electrode locations for forward modelling
3. mesh: contains the finite difference mesh for the 2D modelling and inversions
4. topography: contains the topographic data
5. model: structure to hold cell values for models: conductivity, chargeability or active
6. weighting: file that contains special weightings which alter the type of model produced in the inversions
7. active: a special type of model file specifying the active cells

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 `DCIPF2D`, and are input to `DCINV2D` and `IPINV2D`.

It is assumed that the survey is carried out either along a line in the $x-$direction perpendicular to geologic strike or along a borehole. In the presence of topography, it is assumed that all current and potential electrodes are located along the line and are placed on the topography at each x location. In other words, the programs in the DCIP2D v5.0 package will automatically place the electrodes on the surface according to the topography and the x location given in the observations file. Electrodes will be at $z = 0$ if topography is not given (e.g., set to null). The z coordinates should be provided in the observations file when the data are located down a borehole. Any surface/borehole combination of electrodes is allowed in DCIP2D. The observations file can have three different formats: the `general`, `surface`, or `simple` format. Only a single format is allowed in a data file. **The general format is the only format that will allow the use of borehole locations**. The type of format chosen to store the data does not make any difference to the DCIP2D v5.0 program library and is determined only by the user's preference. At the beginning of execution, the programs will determine the format and the output files will be written in the same format. Standard deviations are optional in each format. The calculation of the standard deviation within `DCIP2D` is given below in the detailed summaries of components. All standard deviations must either be given or not given to be calculated.

Calculation of default standard deviations The default standard deviation is calculated from the DC data using the following rules: $\text{stn}_i = 0.05 \times |\phi_i| + \phi_{max}$, where ϕ_{max} is the potential at the maximum electrode separation averaged over five (5) adjacent stations. The default standard deviation is calculated for IP data by: $\text{stn}_i = 0.05 \times |\eta_i| + \eta_{min}$, where η_{min} is calculated standard deviation of all apparent chargeabilities in the data set. We stress that the default errors are only an initial guess and they facilitate a preliminary inversion of the data. The user will want to alter these error estimates for the final inversion used for interpretation. The data with the default errors are written in dcinv2d.log using the given format. They can be copied to a file for finer adjustment of the error estimates or the user can supply his or her own errors directly.

General format

The DCIP2D 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. The format consists of a line with the current electrode location and number of potential electrode locations associated with it. Each location has x and z coordinates. An example of the general format file structure is as follows:

```
COMMON_CURRENT
!  Comment line
[IPTYPE= 1 | 2]
```

$$\begin{array}{cccccc}
A_1^x & A_1^z & B_1^x & B_1^z & n_1 & \\
M_1^x & M_1^z & N_1^x & N_1^z & val_1 & [stn_1] \\
\vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\
M_{n_1}^x & M_{n_1}^z & N_{n_1}^x & N_{n_1}^z & val_{n_1} & [stn_{n_1}] \\
A_2^x & A_2^z & B_2^x & B_2^z & n_2 & \\
M_1^x & M_1^z & N_1^x & N_1^z & val_1 & [stn_1] \\
\vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\
M_{n_2}^x & M_{n_2}^z & N_{n_2}^x & N_{n_2}^z & val_{n_2} & [stn_{n_2}] \\
A_3^x & A_3^z & B_3^x & B_3^z & n_3 & \\
\vdots & \vdots & \vdots & \vdots & \vdots & \vdots
\end{array}$$

`COMMON_CURRENT` This flag is given prior to `IPTYPE` to let the code know that it is a general format file

`!  Comment Line` Any comments can go here. This line is ignored by `dcinv2d` and must have a preceding “!”

`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 horizontal position along line of current electrode A

A_i^z ith elevation of current electrode A

B_i^x ith horizontal position along line of current electrode B

B_i^z ith elevation of current electrode B

M_j^x jth horizontal position along line of potential electrode M associated with the ith current pair

M_j^z jth elevation of potential electrode M associated with the ith current pair

N_j^x jth horizontal position along line of potential electrode N associated with the ith current pair

N_j^z jth elevation of potential electrode N associated with the ith current pair

val_j jth observed datum related to the jth potential pair and ith current 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.

$\texttt{stn}_j$ Optional jth standard deviation associated with the jth datum. This is a positive, absolute value (not a percentage). If the first datum does not have an error field (or the error field is commented out by placing a "!" before it), the default errors for the entire data set will be calculated from the data.

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

```
COMMON_CURRENT
!  dc data (general format)
221  -45  221  -45        6
 50  250  100   25   -2.31552E-01  1.16776E-02
100  250  150   50   -2.64516E-01  1.33258E-02
150  500  200   75    2.70551E-03  2.35276E-04
200   75  250  100    2.11746E-01  1.06873E-02
250  100  300  125    2.37240E-01  1.19620E-02
300  125  350  150    1.59822E-01  8.09110E-03
221  -45  600  -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 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 at an elevation of 0. 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. An example of the surface format file structure is as follows:

```
COMMON_CURRENT
!  Comment line
[IPTYPE= 1 | 2]
```

$$\begin{array}{cccc}
A_1^x & B_1^x & n_1 & \\
M_1^x & N_1^x & val_1 & [stn_1] \\
\vdots & \vdots & \vdots & \vdots \\
M_{n_1}^x & N_{n_1}^x & val_{n_1} & [stn_{n_1}] \\
A_2^x & B_2^x & n_2 & \\
M_1^x & N_1^x & val_1 & [stn_1] \\
\vdots & \vdots & \vdots & \vdots \\
M_{n_2}^x & N_{n_2}^x & val_{n_2} & [stn_{n_2}] \\
A_3^x & B_3^x & n_3 & \\
\vdots & \vdots & \vdots & \vdots
\end{array}$$

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

`COMMON_CURRENT` This flag is given prior comment line to let the code know that it is a surface format file

`!  Comment Line` Any comments can go here and should be preceded by “!”

`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

$\mathtt{A}_i^x$ ith horizontal position along line of current electrode A

$\mathtt{B}_i^x$ ith horizontal position along line of current electrode B

$\mathtt{M}_j^x$ jth horizontal position along line of potential electrode M associated with the ith current pair

$\mathtt{N}_j^x$ jth horizontal position along line of potential electrode N associated with the ith current pair

$\mathtt{val}_j$ jth observed datum related to the jth potential electrode pair and i^{t}h 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.

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

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

```
COMMON_CURRENT
!  IP surface example
IPTYPE=1
221  -45        4
 50   25   -2.31552E-01  1.16776E-02
100   50   -2.64516E-01  1.33258E-02
250  125    2.37240E-01  1.19620E-02
300  150    1.59822E-01  8.09110E-03
221  -55        2
100  150   -2.64516E-01  1.33258E-02
150  200    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.

Simple format

The simple format is the most straightforward, but also most restrictive of the three formats. The elevations are not given similar to the surface 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 at an elevation of 0. 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 pair location and potential electrode location pair. An example of the surface format file structure is as follows:

```
!  Comment line
[IPTYPE= 1 | 2]
```

$$\begin{array}{cccccc}
A_1^x & B_1^x & M_1^x & N_1^x & val_1 & [stn_1] \\
A_2^x & B_2^x & M_2^x & N_2^x & val_2 & [stn_2] \\
\vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\
A_n^x & B_n^x & M_n^x & N_n^x & val_n & [stn_n]
\end{array}$$

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

`!  Comment Line` Any comments can go here. This line is ignored by `dcinv2d`. General comments should be preceded by a "!"

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

$\mathtt{A}_i^x$ ith horizontal position along line of current electrode A

$\mathtt{B}_i^x$ ith horizontal position along line of current electrode B

$\mathtt{M}_j^x$ ith horizontal position along line of potential electrode M

$\mathtt{N}_j^x$ ith horizontal position along line of potential electrode N

$\mathtt{val}_i$ ith observed datum. 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.

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

Example of simple format The following is an example of the simple format. The data are the same as given in the surface format example; IP data in units of apparent chargeability:

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

3.2.2 Electrodes file

This file contains the electrode locations for DCIPF2D. The electrodes file follows the same formats at the Observations files, but without data and standard deviations. Thus, an electrode file can have three different formats: the general, surface, or simple format. Only a single format is allowed in a data file. **The general format is the only format that will allow the use of borehole locations**. The type of format chosen for forward modelling does not make any difference to the DCIPF2D and is determined only by the user's preference. At the beginning of execution, the programs will determine the format and the output files will be written in the same format.

General format

The forward modelling code 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 location. 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. Each location has x and z coordinates. An example of the general format file structure is as follows:

```
Comment line
COMMON_CURRENT
[IPTYPE= 1 | 2]
```

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

`Comment Line` Any comments can go here. This line is ignored by `dcinv2d`

`COMMON_CURRENT` This flag must go prior to `IPTYPE` in order to let the code know the file is of general format

`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 horizontal position along line of current electrode A

A_i^z ith elevation of current electrode A

B_i^x ith horizontal position along line of current electrode B

B_i^z ith elevation of current electrode B

M_j^x jth horizontal position along line of potential electrode M associated with the ith current pair

M_j^z jth elevation of potential electrode M associated with the ith current pair

N_j^x jth horizontal position along line of potential electrode N associated with the ith current pair

N_j^z jth elevation of potential electrode N associated with the ith current pair

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

```
!  dc data
COMMON_CURRENT
221  -45  221   -45   6
 50  250  100    25
100  250  150    50
150  500  200    75
200   75  250   100
250  100  300   125
300  125  350   150
221  -45  600   -55   2
100   25  150   500
150  500  200  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. 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 at an elevation of 0. 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. An example of the surface format file structure is as follows:

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

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

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

`Comment Line` Any comments can go here. This line is ignored by `dcinv2d`

`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

$\mathtt{A}_i^x$ ith horizontal position along line of current electrode A

$\mathtt{B}_i^x$ ith horizontal position along line of current electrode B

$\mathtt{M}_j^x$ jth horizontal position along line of potential electrode M associated with the ith current pair

$\mathtt{N}_j^x$ jth horizontal position along line of potential electrode N associated with the ith current pair

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

```
IPTYPE=1
221  -45  4
 50   25
100   50
250  125
300  150
221  -55  2
100  150
150  200
```

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.

Simple format

The simple format is the most straightforward, but also most restrictive of the three formats. The elevations are not given similar to the surface 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 locations are on top of the mesh at an elevation of 0. 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 pair location and potential electrode location pair. An example of the surface format file structure is as follows:

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

$$\begin{array}{cccc}
A_1^x & B_1^x & M_1^x & N_1^x \\
A_2^x & B_2^x & M_2^x & N_2^x \\
\vdots & \vdots & \vdots & \vdots \\
A_n^x & B_n^x & M_n^x & N_n^x
\end{array}$$

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

`Comment Line` Any comments can go here. This line is ignored by `dcinv2d`

`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

$\mathtt{A}_i^x$ ith horizontal position along line of current electrode A

$\mathtt{B}_i^x$ ith horizontal position along line of current electrode B

$\mathtt{M}_j^x$ ith horizontal position along line of potential electrode M

$\mathtt{N}_j^x$ ith horizontal position along line of potential electrode N

Example of simple format The following is an example of the simple format. The data are the same as given in the surface format example; IP data in units of apparent chargeability:

```
IPTYPE=1
221   -45    50    25
221   -45   100    50
221   -45   250   125
221   -45   300   150
221   -55   100   150
221   -55   150   200
```

3.2.3 Mesh file

This is the file used to define a finite difference mesh for 2D modelling and inversion. In the file, the x-z plane is divided into a rectangular grid. By convention, the z-axis is positive down. The mesh can be designed by considering it as consisting of a core portion representing the region of interest and a padding zone which ensures that the boundary conditions in the finite difference modelling are handled correctly. In the core portion, the horizontal mesh is mainly controlled by the experiment grid on which to collect the data, so the mesh partition in this region is usually uniform given that the data are collected at constant intervals and the cell width is an integer fraction of the station spacing (or dipole length of the array).

An example of the mesh file structure is as follows:

$$
\boxed{\begin{array}{lll}
\mathtt{N}_x & & \\
X_0 & X_1 & \mathtt{NX}_1 \\
 & X_2 & \mathtt{NX}_2 \\
 & \vdots & \vdots \\
 & X_{N_x} & \mathtt{NX}_{N_x} \\
\mathtt{N}_z & & \\
Z_0 & Z_1 & \mathtt{NZ}_1 \\
 & Z_2 & \mathtt{NZ}_2 \\
 & \vdots & \vdots \\
 & Z_{N_z} & \mathtt{NZ}_{N_z}
\end{array}}
$$

$\mathtt{N}_x$ Number of segments used to define the horizontal partitioning

$\mathtt{X}_i$ Boundary of the segment in $x-$coordinates. The first line has the edged of the mesh (west-most coordinate) and the next boundary. From there, each line is the boundary for the previous line.

$\mathtt{NX}_i$ Number of cells within the segment. The cell size is determined by (X$_i$ - X$_i - 1$) / NX$_i$

$\mathtt{N}_z$ Number of segments used to define the vertical partitioning.

$\mathtt{Z}_i$ Boundary of the segment in $z-$coordinates. These coordinates are in depth (z+ down). The first line has the top of the mesh (west-most coordinate) and the next boundary. From there, each line is the boundary for the previous line. NOTE: When topography is given, the top of the mesh (Z$_0$) is set to the `elev0` in the Topography file. Therefore, it is encouraged to set Z$_0 = 0$ in all cases for user readability of the thicknesses of cells in the first segment.

$\mathtt{NZ}_i$ Number of cells within the segment. The cell size is determined by (Z$_i$ - Z$_i - 1$) / NZ$_i$

Example of general format The following is an example of a 2D mesh:

```
9
-300   -180    1
       -130    1
       -110    1
       -100    1
        100   40
        110    1
        130    1
        180    1
        300    1
14
  0      10    5
         22    4
         42    5
         57    3
         63    1
         71    1
         81    1
         95    1
        115    1
        140    1
        170    1
        205    1
        245    1
        300    1
```

In the above example, the mesh is (48 × 27) covering from $x = [-100, 100]$ and $z = [0, 300]$. There are 9 segments in the file defining the horizontal partitioning which goes from $x = -300$ to $x = 300$. The region $x = [-100, 100]$ has been divided into 40 equal length cells. Four padding cells of increasing width extend the mesh out to $x = [-300, 300]$. In depth, there are 14 segments over the depth range $z = [0, 300]$. Note that the bottom depth is considerably larger than the depth of investigation of the survey. This procedure is carried out automatically in all-default mode. If default meshing is chosen, the horizontal cell size is set to 1/3 the average electrode spacing. The vertical discretization is set to accommodate 1:2 vertical to horizontal ratio and the padding is increased by 1.5 times. The example of the mesh is as follows:

Notes on the 2D mesh Generally, at least three cells are recommended between adjacent stations (i.e., per dipole length). In the presence of topography, one might use a finer partitioning in sections where the topographic relief changes rapidly. This helps to better approximate the topographic surface using the finite difference mesh. In the presence of surface topography, the top of the finite difference mesh corresponds to the highest point on the surface (see also the description of Topography). Since the current sources are all on the surface of the earth, the vertical mesh has thicknesses which generally increase with depth. The cell thickness should be small near the surface and increase slowly in the upper region that is within the array's depth of investigation. Finer partitioning can be used at a depth where there is rapid change in the conductivity for forward modelling or where one might expect anomalous structures in the inversion. The maximum depth for the mesh should be considerably larger than the depth of investigation provided by the survey. The thickness of the cells, especially near the surface, should be determined in reference to the horizontal partitioning so that the cells do not have an extremely large aspect ratio (width divided by thickness). Cells with extreme aspect ratio tend to degrade the quality of the forward solution and they can also cause undesirable abrupt changes in the inversion results. It is good practice to keep the aspect ratio of cells less than 5. Once the core mesh is designed, a set of padding cells is required to extend the mesh horizontally. Three to five cells whose width progressively increases by a factor of two to three provide sufficient extension so that the boundary conditions are adequately handled.

It is good practice to generate the mesh and perform a forward modelling using a uniform conductivity model, called the half-space test. The test for a properly designed mesh is performed by calculating the apparent conductivities from the uniform conductivity model. The mesh is considered adequate if the modelled responses do not deviate from the true conductivity by more than a few percent. Otherwise, the mesh should be modified and the test performed again. This test is valid when there is no topography. However, a mesh designed properly for a model with topography should pass the half-space test when the mesh is used alone without topography.

3.2.4 Topography file

This is the file used to define the surface topography along the traverse of the DC/IP experiment by specifying the elevations of selected points. It should be noted that the Mesh file is linked to the topography. The specification of the top elevation, elev0, sets the top of the mesh. In other words, elev0 in the topography is the top of the mesh and where $z = 0$. This point is assumed to be the highest point on the topographic surface. Locations above elev0 on the surface are set to elev0. The elevation is positive up similar to the Observations file and thus it can be given in relative values. The topography file must cover the core portion of the mesh where electrodes are placed. The coverage should ideally extend to both ends of the mesh, otherwise the remaining portion towards the ends will be assumed to have the same surface elevation in each direction as given at the end points within the file. The topographic surface is discretized onto the mesh using the elevations at the horizontal nodes that are obtained by linear interpolation from this file.

An example of the Topography file structure is as follows:

N	elev0
$\texttt{X}_1$	$\texttt{elev}_1$
$\texttt{X}_2$	$\texttt{elev}_2$
$\vdots$	$\vdots$
$\texttt{X}_N$	$\texttt{elev}_N$

N Number of locations defining the topographic profile.

elev0 The elevation of the top of the 2D mesh. See the introductory paragraph in this section for details.

$\mathtt{X}_i$ ith horizontal location.

$\mathtt{elev}_i$ ith elevation at X$_i$.

Example of topography The following is an example of topography:

```
  10   40
-250   10
-180   20
-130   30
-110   40
 -50   50
  50   50
 110   40
 130   30
 180   20
 250   10
```

In the above example, there are 10 locations that vary in elevation from 10 to 50 m. The two locations above 40 m (elev0) will be represented as 40 m of elevation within DCIP2D. The top of the mesh will also be placed at 40 m.

3.2.5 Model file

This is the file used to define any model given to DCIP2D. Each file contains the cell values within the model. The number of cells in the $x-$ and $z-$directions are specified in this file and should coorespond to the values given in the Mesh file.

An example of the model file structure is as follows:

$$\begin{array}{cccc}
\mathtt{N}_x\ \mathtt{N}_z & & & \\
\mathtt{V}_{1,1} & \mathtt{V}_{2,1} & \dots & \mathtt{V}_{N_x,1} \\
\mathtt{V}_{1,2} & \mathtt{V}_{2,2} & \dots & \mathtt{V}_{N_x,2} \\
\vdots & \vdots & \vdots & \vdots \\
\mathtt{V}_{1,N_z} & \mathtt{V}_{2,N_z} & \dots & \mathtt{V}_{N_x,N_z}
\end{array}$$

$\mathtt{N}_x$ Number of cells in the horizontal direction

$\mathtt{N}_z$ Number of cells in the vertical direction.

$\mathtt{V}_{i,k}$ Value of the model at the ith cell in the horizontal direction and jth cell in depth. When $j = 1$, the cell is at the top of the mesh and $j = N_z$ corresponds to the bottom of the mesh. In the horizontal direction $i = 1$ corresponds to the beginning of the line (e.g., West) and $i = N_x$ is at the end of the line (e.g., East).

$\mathtt{X}_i$ ith horizontal location

$\mathtt{elev}_i$ ith elevation

Example of a model The following is an example of a model:

```
3 10
.01   .23   .20   .46   0.2
.64   .32   .54
.19   1E-5
.01   .23   .20   .46   0.2
.64   .33   .21   .85   0.2
.11   .75   .25   .62   0.4
.25   .64   .23   .15   0.4
```

In the above example, there are 10 horizontal cells and 3 vertical cell associated with the mesh. The first row is broken into the first three lines of the file. The second row of the model is given on lines 2-4. The third row of the model is given on lines 5 and 6. DCIP2D would also read the same model with three rows of ten values.

Notes on the 2D model The model is stored in a row format with $V_{1,1}$ being the top left cell. As shown above, each row of the model, $V_{1,k} \ldots V_{N_x,k}$, can be broken down into more than a single line in the file, however, each model row must start on a new line.

In the presence of topography, the cells above the surface specified by topo.dat represent the air and, therefore, are not part of the model. For simplicity, however, the files model.con and model.chg must include those cells as defined by the Mesh file. These cells in the model file will be ignored when the model is input to a program. To distinguish these cells, the user can assign a special value to them, for instance, a very small value (say 10^{-10}) for conductivity and a negative value for the chargeability model. These cells in the output model files from the inversion programs have a conductivity value equal to 10^{-8} times the mean conductivity value immediately beneath the surface and a chargeability value of -10^{30}.

Active model file

The active cell model file is a specific type of model that has the same format, but only contains values of -1, 0, or 1. Default (null option) active cells models place all cells in the model as active, meaning they can be changed during the inversion and influence the model objective function and data misfit. Cells that are inactive (not allowed to change) and *do not* influence the model objective function should be set to 0. Cells that are inactive, but can influence the values of neighbouring cells through the model objective function should have a value of -1.

3.2.6 Weighting file

This is the file used to define the smallness and spatial weights to be incorporated into the inversion codes DCINV2D or IPINV2D. These weights are given in the model objective function (equation 14). As with the Model file, the weights are stored in a row format. Weights with the value of 1.0 do contribute extra information to model objective function. Deviating from 1.0 will increase the relative importance (or not) of the cell value or interface. One has the choice of a single file containing all the weights or a file for each weight.

An example of the Weighting file structure *containing all weights* is as follows:

$$\boxed{\begin{array}{llll}
\texttt{N}_x\ \ \texttt{N}_z & & & \\
\texttt{Ws}_{1,1} & \texttt{Ws}_{2,1} & \ldots & \texttt{Ws}_{N_x,1} \\
\vdots & \vdots & \vdots & \vdots \\
\texttt{Ws}_{1,N_z} & \texttt{Ws}_{2,N_z} & \ldots & \texttt{Ws}_{N_x,N_z} \\
\texttt{Wx}_{1,1} & \texttt{Wx}_{2,1} & \ldots & \texttt{Wx}_{N_x-1,1} \\
\vdots & \vdots & \vdots & \vdots \\
\texttt{Wx}_{1,N_z} & \texttt{Wx}_{2,N_z} & \ldots & \texttt{Wx}_{N_x-1,N_z} \\
\texttt{Wz}_{1,1} & \texttt{Wx}_{2,1} & \ldots & \texttt{Wx}_{N_x,1} \\
\vdots & \vdots & \vdots & \vdots \\
\texttt{Wz}_{1,N_z-1} & \texttt{Wz}_{2,N_z-1} & \ldots & \texttt{Wz}_{N_x,N_z-1}
\end{array}}$$

$\texttt{N}_x$ Number of cells in the horizontal direction. This number is the same as found in Model andMesh files.

$\texttt{N}_z$ Number of cells in the vertical direction. This number is the same as found in Model andMesh files.

$\texttt{Ws}_{i,k}$ Weight given to the smallest model component. Ws has size of $N_x \times N_z$. The values of i and k count consistent with the Model format.

$\texttt{Wx}_{i,k}$ Weight given across the faces of cells in the x-direction. Wx has size of $N_x - 1 \times N_z$. The values of i and k count consistent with the Model format.

$\texttt{Wz}_{i,k}$ Weight given across the faces of cells in the z-direction. Wz has size of $N_x \times N_z - 1$. The values of i and k count consistent with the Model format.

An example of a Weighting file structure *containing only one weight* is as follows:

$$\boxed{\begin{array}{llll}
\texttt{N}_x^W\ \ \texttt{N}_z^W & & & \\
\texttt{W}_{1,1} & \texttt{W}_{2,1} & \ldots & \texttt{W}_{N_x^W,1} \\
\vdots & \vdots & \vdots & \vdots \\
\texttt{W}_{1,N_z^W} & \texttt{W}_{2,N_z^W} & \ldots & \texttt{W}_{N_x^W,N_z^W}
\end{array}}$$

$\mathtt{N}_x^W$ Number of cells in the horizontal direction for that weight (N_x for W_s and W_z and $N_x - 1$ for W_x).

$\mathtt{N}_z^W$ Number of cells in the vertical direction for that weight (N_z for W_s and W_x and $N_z - 1$ for W_z).

$\mathtt{W}_{i,k}$ Weight value

Example of a weight file The following is an example of a weight file *with all of the weights*:

```
5 4
1    1     1     1     1
1    1     1     1     1
1    1     1     1     1
1   100   100    1
1   100   100    1
1   100   100    1
1   0.01  0.01  0.01  1
1   0.01  0.01  0.01  1
```

In the above example, there are 5 horizontal cells and 3 vertical cell associated with the mesh. Therefore, W_s is 5×3, W_x is 4×3, and W_z is 5×2. The is no additional emphasis placed on the smallest model component. Large emphasis on minimizing the derivatives has been place on two cell faces in the $x-$direction in all three vertical rows. Small values have been placed at two vertical faces to allow for large jumps. This weighting file would be analogous to driving the model toward a vertical dyke.

Note on weight design When the weighting is supplied by the user, care should be taken so that the final weighting matrix constructed by the program is positive definite. This property can be destroyed when the combination of the cell weighting coefficients and component coefficients (s, x, z, see the description of special weighting w.dat file) makes the diagonal elements too small. Under such circumstances, the program will usually print in the log file the indices of the rows which are not diagonally dominant. In the most severe situation, the program will output an error message indicating that the matrix may not be positive definite. When these messages appear, the user should stop the program execution and redesign the special weighting by increasing the value of s or decreasing the dynamic range of WS .

4 Running the programs

The software package DCIP2D uses five general codes:

- DCIPF2D: performs 2D forward modelling for DC and IP data
- DCINV2D: inverts DC potentials to recover a 2D conductivity model
- IPINV2D: inverts IP data to recover a 2D chargeability 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:

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

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 all of the keyword combinations allowed for that program.

Each control file contains a formatted list of arguments, parameters, and file names in a combination specific for the executable, which can be in any order. Values that are being set by the user are given to each program through a specific list of keywords (e.g., WEIGHT to specify the type of weighting). 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. When inputting a file, the word FILE should be following the keyword (e.g., MESH FILE mesh.msh). If a value is being used than the word VALUE will follow a keyword (e.g., COND VALUE 0.001).

4.2 DCIPF2D

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

```
dcipf2d dcipf2d.inp
```

where the input file, dcipf2d.inp, is described below. The options can be in any order.

4.2.1 Input files

Keywords for the input file dcipf2d.inp are:

```
FWD [DC | IP | IPL]                 !  Type of data to model
MESH FILE fileName                  !  Mesh
LOC [LOC_X | LOC_XZ] fileName       !  Locations file
TOPO [FILE fileName | DEFAULT]      !  Topography
COND [VALUE c | FILE fileName]      !  Conductivity value or model file
CHG [VALUE c | FILE fileName]       !  Chargeability value or model file
WAVE w_min w_max N                  !  Optional min/max of N wave values
```

FWD The choices after this keyword are:

1. DC for DC forward modelling. The chargeability model and wave, if given, is ignored for DC forward modelling only.
2. IP for IP forward modelling.
3. IPL for IP forward modelling using product of the sensitivity matrix and chargeability.

MESH FILE The 2D mesh file name is followed after these keywords. For example MESH FILE mesh.msh.

LOC The observation locations. The choices after this keyword are:

1. LOC_X when giving simple or surface locations formats.
2. LOC_XZ when using the general locations format.

TOPO The choices for the topography are:

1. FILE followed by the name of the topography file.
2. DEFAULT for flat topography at and elevation of 0.

COND The choices for the conductivity model are:

1. FILE followed by the name of the conductivity file .
2. VALUE followed by a number for the conductivity throughout the mesh.

CHG The choices for the chargeability model are:

1. FILE followed by the name of the chargeability file .
2. VALUE followed by a number for the chargeability throughout the mesh.

WAVE is followed by 3 constants: w_min w_max N. These are the wave numbers used in the cosine transform. There will be N wave values, log spaced from w_min to w_max in time. The default values (if WAVE is not given) is w_min = 2.5e-4, w_max = 1.0, and N= 13.

Example of dcipf2d.inp

Example of an input file for DCIPF2D to model DC data that are given in general format and a with a topography file:

```
FWD DC                 !  DC input type
MESH FILE myMesh.msh   !  mesh file
LOC LOC_XZ obs.loc     !  general formatted locations file
TOPO FILE myTopo.dat   !  topography file
COND FILE myModel.con  !  conductivity model file
```

4.2.2 Output files

The files created by DCIPF2D are:

obs_dc.dat The computed DC potential data.

obs_ip.dat The computed IP data if the option IP is chosen.

obs_ipL.dat The computed IP data if the option IPL is chosen.

4.3 DCINV2D

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

```
dcinv2d dcinv2d.inp
```

where the input file, dcinv2d.inp, is described below. The options can be in any order. The minimum keywords needed for an inversion are MESH and OBS.

4.3.1 Input Files

Keywords for the input file dcinv2d.inp are:

```
MESH [DEFAULT | FILE | NC_ASPR n a]         !  Specify the mesh
OBS [LOC_X | LOC_XZ] fileName               !  Observations file follows
NITER n                                     !  Maximum number of iterations
CHIFACT [c | DEFAULT]                       !  Chifact c or default
TOPO [FILE | DEFAULT]                       !  Topography
INIT_MOD [VALUE | FILE | DEFAULT]           !  Initial conductivity model
REF_MOD [VALUE | FILE | DEFAULT]            !  reference conductivity model
ALPHA [VALUE | LENGTH | DEFAULT]            !  Alphas or length scales
WEIGHT [FILE | FILES | DEFAULT]             !  Alphas or length scales
WAVE w_min w_max N                          !  min/max of N wave values
STORE_ALL_MODELS [TRUE | FALSE]             !  store all models or write to disk
INVMODE [CG | SVD]                          !  way to solve the system
CG_PARAM maxit tol                          !  parameters for CG system
HUBER c                                     !  constant for the Huber norm
EKBLOM rho_s rho_x rho_z eps_s eps_x eps_z  !  six constants for the Ekblom norm
ACTIVE_CELLS fileName                       !  specify file for active cells
USE_MREF [TRUE | FALSE]                     !  ref model throughout spatial terms
BOUNDS [VALUE | FILE_L | FILE_U | NONE]     !  specify bounds
```

MESH The choices after this keyword are:

1. DEFAULT the programs creates a mesh (output dcinv2d.msh) with 3 cells between electrodes and the aspect ratio of the top cells set to 3. **NOTE**: This option assumes that the data are collected by commonly used arrays and that the topographic relief is moderate. Thus, this option may not be optimal when the data are collected with unusual electrode geometry or when data are collected over severe surface topography. In such cases, the user should redesign the mesh so that it is better suited for the particular needs of the data set.
2. FILE filename file name of the mesh
3. NC_ASPR n a creates a mesh (output dcinv2d.msh) that has n cells between the electrodes and the aspect ratio of the top cells is set to a

OBS The observation locations. The choices after this keyword are:

1. LOC_X when giving simple or surface locations formats
2. LOC_XZ when using the general locations format.

NITER A value follows this keyword representing the number of maximum iterations for the inversion. **NOTE**: The program will terminate before the specified maximum number of iterations is reached if the expected data misfit is achieved and if the model norm has plateaued. However, if the program exits when the maximum iteration is reached, the file dcinv2d.out should be checked to see if the desired (based on the number of data and chi factor) has been reached and if the model norm is no longer changing. If either of these conditions has not been met then the program should be restarted. If the desired misfit level is not achieved, but the model norm has plateaued and the model is not changing between successive iterations, then the user may want to adjust the target misfit to a higher value. Also an investigation as to which data are most poorly fit can be informative. It may be that the assigned standard deviations to specific data are unrealistically small. The program restarts using the information in dcinv2d.out and dcinv2d.con.

CHIFACT The value at which the program reproduced the data. The choices after this keyword are:

1. `DEFAULT` where the program will start with 1e-3 initially and then when the misfit stop decreasing, the chi factor will be changed by 10%
2. `constant` the value to set the chi factor (1 is when the data misfit equals the number of data), or if a value is not there, but `CHIFACT` is given, the program will stop when the data misfit reaches the number of data

TOPO The choices after this keyword are:

1. `FILE` followed by the name of the topography file
2. `DEFAULT` for flat topography at an elevation of 0.

INIT_MOD The choices for the initial model are:

1. `FILE fileName` name of the initial conductivity file
2. `VALUE constant` the value for the initial conductivity throughout the mesh
3. `DEFAULT` for the initial model to be set to the reference model.

REF_MOD The choices for the reference model are:

1. `FILE fileName` name of the reference conductivity file
2. `VALUE constant` the value for the reference conductivity throughout the mesh
3. `DEFAULT` the reference model is equal to the best fitting half-space model.

WAVE is followed by 3 constants: `w_min w_max N`. These are the wave numbers used in the cosine transform. There will be `N` wave values, log spaced from `w_min` to `w_max` in time. The default values (if `WAVE` is not given) is `w_min = 2.5e-4`, `w_max = 1.0`, and `N= 13`.

ALPHA The choices after this keyword are:

1. `DEFAULT` where the program will set $\alpha_s = 0.001$*(90/max electrode separation)^2 and $\alpha_x = \alpha_z = 1$.
2. `VALUE a_s a_y a_z` the user gives the coefficients for the each model component for the model objective function from equation 14: α_s is the smallest model component, α_x is along line smoothness, and α_z is vertical smoothness.
3. `LENGTH L_x L_z` the user gives the length scales and the smallest model component is calculated accordingly. The conversion from α's to length scales can be done by:

$$L_x = \sqrt{\frac{\alpha_x}{\alpha_s}}; \;\; L_z = \sqrt{\frac{\alpha_z}{\alpha_s}} \tag{32}$$

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

WEIGHT The weighting for the model objective function allows for three options:

1. `DEFAULT` No weighting is supplied (all values of weights are 1)
2. `FILE fileName` The weighting is supplied as a weight file with all the weights in one file fileName
3. `FILES fileS fileX fileZ` The weighting is supplied as three separate weight files with the weight for the smallest model component in fileS, the $x-$component written in file fileX and the $z-$component written in fileZ.

STORE_ALL_MODELS There are two choices:

1. `TRUE` Write all models and predicted data to disk. Each iteration will have dcinv2d_xx.con and dcinv2d_xx.pre files where xx is the iteration (e.g., 01 for the first iteration)
2. `FALSE` Only the final model and predicted data file are written. These files are named dcinv2d.con and dcinv2d.pre for the conductivity and predicted data, respectively.

INVMODE This specifies the way the system is solved:

1. `SVD` Solve the system using a subspace method with basis vectors. This is the solution methodology of the original code and the default if not given.
2. `CG` Solve the system using a subspace method with conjugate gradients (CG). This allows additional constraints (i.e., Huber and Ekblom norms) to be incorporated into the code.

CG_PARAM is used when the inversion mode is CG. The keyword is followed by two constants: `maxit` specifying the maximum number of iterations (default is 10), and `tol` specifying the solution's accuracy (default is 0.01)

HUBER The Huber norm is used when evaluating the data misfit. A constant follows this keyword and this option is only available when using the CG inversion mode option. The default value is 1e100. The constant is `c` from equation 13.

EKBLOM Use the Ekblom norm. Six (6) values should follow this keyword: $\rho_s, \rho_x, \rho_z, \epsilon_s, \epsilon_x, \epsilon_z$ representing the constants found in equation 18.

ACTIVE_CELLS followed by the file name of the active cell file.

USE_MREF This option is used to decide if the reference model should be in the spatial terms of the model objective function (equation 14). There are two options: `TRUE` to include the reference model in the spatial terms or `FALSE` to have the reference model only in the smallest model component.

BOUNDS The bounds options are:

1. `NONE` Do not include bounds in the inversion
2. `VALUE lwr upr` Give a constant global lower bound of `lwr` and upper bound of `upr`.
3. `FILE_L fileName` The lower bound is given in a file filename and is in the Model format.
4. `FILE_U fileName` The upper bound is given in a file filename and is in the Model format.

Example of dcinv2d.inp

Below is an example of the input file dcinv2d.inp. The code will create a mesh with 4 cell between electrode locations and the aspect ratio of the size top cells set to 2. This means the reference and initial models will not be given in a file, but rather set to 0.001 S/m. The length scales will be 5 m in each direction and the Ekblom norm will have exponents of 1.0 in each direction to emphasize blockiness. It will start from scratch and stop after 50 iterations if the desired misfit (equal to 90% of the number of data) is not achieved. Conjugate gradients are used to solve the system of equations with a maximum number of CG iterations set at 800 and a relative accuracy of 1e-5. There are no bounds in this inversion.

```
OBS LOC_XZ obs_dc.dat                  !  general formatted data
TOPO FILE topography.txt               !  topography file
MESH NC_ASPR 4 2                       !  DCINV2D created mesh
ALPHA LENGTH 5 5                       !  length scales of 5 m
CHIFACT 0.9                            !  data misfit equal to number of data
INIT_MOD DEFAULT                       !  initial model is ref model
REF_MOD VALUE 0.001                    !  ref model
EKBLOM 1.0 1.0 1.0 1e-5 1e-5 1e-5      !  Ekblom norm
NITER 50                               !  max iterations
INVMODE CG                             !  use CG solver
CG_PARAM 800 1e-5                      !  Solver specs
```

4.3.2 Output Files

DCINV2D will create the following files:

1. dcinv2d.log The log file containing the minimum information for each iteration, summary of the inversion, and standard deviations if assigned by DCINV2D.

2. dcinv2d.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

3. dcinv2d_iter.con Conductivity model for each iteration (iter defines the iteration step) if STORE_ALL_MODELS FALSE is used

4. dcinv2d_iter.pre Predicted data for each iteration (iter defines the iteration step) if STORE_ALL_MODELS FALSE is used

5. dcinv2d.pre Predicted data file that is updated after each iteration (will also be the final predicted data)

6. dcinv2d.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

4.4 IPINV2D

This program performs the 2D inversion of induced polarization data. Command line usage:

```
ipinv2d ipinv2d.inp
```

for the control file ipinv2d.inp described below. The options can be in any order. The minimum keywords needed for an inversion are `MESH`, `OBS`, and `COND`.

4.4.1 Input Files

Keywords for the input file ipinv2d.inp are:

```
MESH [DEFAULT | FILE | NC_ASPR n a]         !  Specify the mesh
OBS [LOC_X | LOC_XZ] fileName               !  Observations file follows
NITER n                                     !  Maximum number of iterations
CHIFACT [c | DEFAULT]                       !  Chifact c or default
TOPO [FILE | DEFAULT]                       !  Topography
INIT_MOD [VALUE | FILE | DEFAULT]           !  Initial chargeability model
REF_MOD [VALUE | FILE | DEFAULT]            !  Reference chargeability model
COND [VALUE | FILE ]                        !  Conductivity model
ALPHA [VALUE | LENGTH | DEFAULT]            !  Alphas or length scales
WEIGHT [FILE | FILES | DEFAULT]             !  Alphas or length scales
WAVE w_min w_max N                          !  min/max of N wave values
STORE_ALL_MODELS [TRUE | FALSE]             !  store all models or write to disk
INVMODE [CG | SVD]                          !  way to solve the system
CG_PARAM maxit tol                          !  parameters for CG system
HUBER c                                     !  constant for the Huber norm
EKBLOM rho_s rho_x rho_z eps_s eps_x eps_z  !  six constants for the Ekblom norm
ACTIVE_CELLS fileName                       !  specify file for active cells
USE_MREF [TRUE | FALSE]                     !  ref model throughout spatial terms
BOUNDS [VALUE | FILE_L | FILE_U | NONE]     !  specify bounds
```

MESH The choices after this keyword are:

1. `DEFAULT` the programs creates a mesh (output dcinv2d.msh) with 3 cells between electrodes and the aspect ratio of the top cells set to 3. **NOTE**: This option assumes that the data are collected by commonly used arrays and that the topographic relief is moderate. Thus, this option may not be optimal when the data are collected with unusual electrode geometry or when data are collected over severe surface topography. In such cases, the user should redesign the mesh so that it is better suited for the particular needs of the data set.
2. `FILE filename` file name of the mesh

3. NC_ASPR n a creates a mesh (output dcinv2d.msh) that has n cells between the electrodes and the aspect ratio of the top cells is set to a

OBS The observation locations. The choices after this keyword are:

1. LOC_X when giving simple or surface locations formats
2. LOC_XZ when using the general locations format.

NITER A value follows this keyword representing the number of maximum iterations for the inversion. **NOTE**: The program will terminate before the specified maximum number of iterations is reached if the expected data misfit is achieved and if the model norm has plateaued. However, if the program exits when the maximum iteration is reached, the file ipinv2d.out should be checked to see if the desired (based on the number of data and chi factor) has been reached and if the model norm is no longer changing. If either of these conditions has not been met then the program should be restarted. If the desired misfit level is not achieved, but the model norm has plateaued and the model is not changing between successive iterations, then the user may want to adjust the target misfit to a higher value. Also an investigation as to which data are most poorly fit can be informative. It may be that the assigned standard deviations to specific data are unrealistically small. The program restarts using the information in ipinv2d.out and ipinv2d.con.

CHIFACT The value at which the program reproduced the data. The choices after this keyword are:

1. DEFAULT where the program will start with 1e-3 initially and then when the misfit stop decreasing, the chi factor will be changed by 10%
2. constant the value to set the chi factor (1 is when the data misfit equals the number of data), or if a value is not there, but CHIFACT is given, the program will stop when the data misfit reaches the number of data

TOPO The choices after this keyword are:

1. FILE followed by the name of the topography file
2. DEFAULT for flat topography at an elevation of 0.

INIT_MOD The choices for the initial model are:

1. FILE fileName name of the initial chargeability file
2. VALUE constant the value for the initial chargeability throughout the mesh
3. DEFAULT for the initial model to be set to the reference model.

REF_MOD The choices for the reference model are:

1. FILE fileName name of the reference chargeability file
2. VALUE constant the value for the reference chargeability throughout the mesh
3. DEFAULT the reference model is set to zero.

COND The choices for the conductivity model (required) are:

1. `FILE fileName` name of the conductivity file
2. `VALUE constant` the value for the conductivity throughout the mesh. **NOTE**: The conductivity of a uniform half space for IP inversions should only be used for preliminary examination of the data. When there is little structure in the background conductivity, the inversion using this default mode can yield a reasonable chargeability model and it is justifiable to fit the data close to the expected misfit value. However, when the background conductivity deviates greatly from a uniform half space, reproducing the data to within the assumed errors will certainly result in over-fitting the data. If the half-space conductivity is assumed, then it is prudent to assign a value greater than 1.0 for chi factor when the background conductivity is structurally complex. The judgment can be made based upon the complexity of the apparent resistivity pseudo-section.

WAVE is followed by 3 constants: `w_min w_max N`. These are the wave numbers used in the cosine transform. There will be `N` wave values, log spaced from `w_min` to `w_max` in time. The default values (if `WAVE` is not given) is `w_min = 2.5e-4`, `w_max = 1.0`, and `N= 13`.

ALPHA The choices after this keyword are:

1. `DEFAULT` where the program will set α_s = 0.001*(90/max electrode separation)^2 and $\alpha_x = \alpha_z = 1$.
2. `VALUE a_s a_y a_z` the user gives the coefficients for the each model component for the model objective function from equation 14: α_s is the smallest model component, α_x is along line smoothness, and α_z is vertical smoothness.
3. `LENGTH L_x L_z` the user gives the length scales and the smallest model component is calculated accordingly. The conversion from α's to length scales can be done by:

$$L_x = \sqrt{\frac{\alpha_x}{\alpha_s}};\ L_z = \sqrt{\frac{\alpha_z}{\alpha_s}} \tag{33}$$

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

WEIGHT The weighting for the model objective function allows for three options:

1. `DEFAULT` No weighting is supplied (all values of weights are 1)
2. `FILE fileName` The weighting is supplied as a weight file with all the weights in one file fileName
3. `FILES fileS fileX fileZ` The weighting is supplied as three separate weight files with the weight for the smallest model component in fileS, the $x-$component written in file fileX and the $z-$component written in fileZ.

STORE_ALL_MODELS There are two choices:

1. `TRUE` Write all models and predicted data to disk. Each iteration will have dcinv2d_xx.con and dcinv2d_xx.pre files where xx is the iteration (e.g., 01 for the first iteration)
2. `FALSE` Only the final model and predicted data file are written. These files are named dcinv2d.con and dcinv2d.pre for the conductivity and predicted data, respectively.

INVMODE This specifies the way the system is solved:

1. `SVD` Solve the system using a subspace method with basis vectors. This is the solution methodology of the original code and the default if not given.
2. `CG` Solve the system using a subspace method with conjugate gradients (CG). This allows additional constraints (i.e., Huber and Ekblom norms) to be incorporated into the code.

CG_PARAM is used when the inversion mode is CG. The keyword is followed by two constants: `maxit` specifying the maximum number of iterations (default is 10), and `tol` specifying the solution's accuracy (default is 0.01)

HUBER The Huber norm is used when evaluating the data misfit. A constant follows this keyword and this option is only available when using the CG inversion mode option. The default value is 1e100.

EKBLOM Use the Ekblom norm. Six (6) values should follow this keyword: $\rho_s, \rho_x, \rho_z, \epsilon_s, \epsilon_x, \epsilon_z$ representing the constants found in equation 18.

ACTIVE_CELLS followed by the file name of the active cell file.

USE_MREF This option is used to decide if the reference model should be in the spatial terms of the model objective function (equation 14). There are two options: `TRUE` to include the reference model in the spatial terms or `FALSE` to have the reference model only in the smallest model component.

BOUNDS The bounds options are:

1. `NONE` Do not include bounds in the inversion
2. `VALUE lwr upr` Give a constant global lower bound of `lwr` and upper bound of `upr`.
3. `FILE_L fileName` The lower bound is given in a file filename and is in the Model format.
4. `FILE_U fileName` The upper bound is given in a file filename and is in the Model format.

Example of ipinv2d.inp

Below is an example of the input file i[inv2d.inp. The code reads mesh from the file mesh2d.msh with topography from topography.txt. The means the reference and initial models will be set to one another and equal zero. The conductivity model is given as the output from `DCINV2D`. The alpha values have been given for $\alpha_s = 0.001$ and $\alpha_x = \alpha_z = 1$. The model objective function will have an l_2 norm (which would also be the same as `EKBLOM 2 2 2 epsS epsX epsZ`). It will start from scratch and stop after 50 iterations if the desired misfit (equal to the number of data) is not achieved. Conjugate gradients are used to solve the system of equations and the bounds are given in two separate files.

```
OBS LOC_XZ obs_ip.dat       !  general formatted data
TOPO FILE topography.txt    !  topography file
MESH FILE mesh2d.msh        !  mesh
COND FILE dcinv2d.con       !  conductivity model
ALPHA VALUE 0.001 1 1       !  length scales of 5 m
CHIFACT 1.0                 !  data misfit equal to number of data
INIT_MOD DEFAULT            !  initial model is ref model
REF_MOD DEFAULT             !  ref model
NITER 50                    !  max iterations
INVMODE CG                  !  use CG solver
BOUNDS FILE_L lower.bnd     !  lower bounds
BOUNDS FILE_U upper.bnd     !  upper bounds
```

4.4.2 Output Files

IPINV2D will create the following files:

1. ipinv2d.log The log file containing the minimum information for each iteration, summary of the inversion, and standard deviations if assigned by DCINV2D.

2. ipinv2d.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

3. ipinv2d_iter.chg Chargeability model for each iteration (iter defines the iteration step) if STORE_ALL_MODELS FALSE is used

4. ipinv2d_iter.pre Predicted data for each iteration (iter defines the iteration step) if STORE_ALL_MODELS FALSE is used

5. ipinv2d.pre Predicted data file that is updated after each iteration (will also be the final predicted data)

6. ipinv2d.chg Chargeability 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

5 Workflow

This section describes the standard workflow for running the DCIP2D inversion. It is provided in a form of a checklist, where elements are linked to other sections of this manual with a more detailed explanation of the concepts. Figure 2 shows the workflow guiding the user through the steps of correctly performed inversion. Inverting data is not a one-step process and each inversion should be carried out with an understanding of each step of the process.

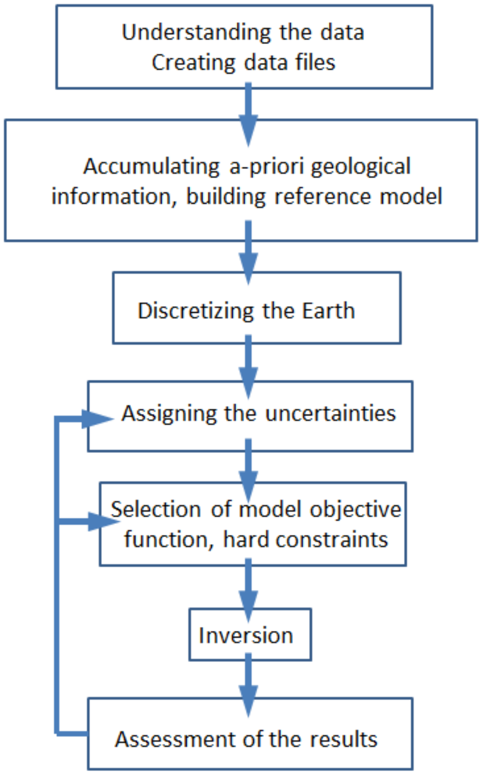

Understanding the data
Creating data files
Accumulating a-priori geological information, building reference model
Discretizing the Earth
Assigning the uncertainties
Selection of model objective function, hard constraints
Inversion
Assessment of the results

Figure 2: Inversion workflow chart.

5.1 Understanding the data

Before inverting the data, please ensure that you have good general understanding of the resistivity/IP data. The following is the checklist of the questions the user should have answers for, prior to preparation of the data for the inversion:

- What is the array configuration?
- What are the electrode locations and spacing?
- What are the units of the data?
- Are the data normalized to unit current?

- What is the total length of the survey line?
- Are estimates of the noise on the data available?

The inversion code assumes that the data are measured voltages due to a unit current in the transmitter. The formats are described in Observations file section. For display purposes however, the `DCIP2D_data_viewer` GUI provided with DCIP2D v5.0 will plot data as apparent resistivities.

5.2 Accumulating prior information; building a reference model

Prior information includes field geology, drill data, borehole measurements and also an "educated guess" about the physical property values and anticipated structures for a particular survey. The goal is to include as much of this information as possible into the final inversion and version 5.0has much flexibility to do this. To be useful, all prior information needs to be connected directly with the physical property in question. Some examples of information that can be included are:

- A listing of the different rock units and the expected range of physical property values.
- A characteristic geologic model for the deposit (example: typical porphyry geometry).
- Physical property reference model. The reference model can be defined by a single value or it can be more complicated. Prior knowledge of geology and/or anomaly structure, and associated physical property values, should be incorporated into the reference model.
- In addition to the reference model it is important to have a sense of its validity. If portions of the reference model arose from point measurements (surface or borehole) then high confidence might exist close to the measurement locations but uncertainty likely increases away from there. If the reference model has a contact zone, then knowledge about the uncertainty in its location is also sought. Thus, in the end, the goal is to have a reference model and also as much knowledge as possible about the data support that generated that model, what is known and what interpolation or hypothesis.

The above information can be input into the inversion in various ways. The combination of reference models, choice of objective functions to make structures more blocky or smooth, localized weightings for each term in the model objective function, bound constraints on the physical properties, and the use of active and inactive cells allows the user much flexibility to obtain a solution that is compatible with the data and his knowledge about the deposit. This will be addressed further for the workflow item concerned with selecting the model objective function and constraints.

5.3 Discretizing the Earth

For the 2D models of the Earth it is assumed that there is no variation of the physical property perpendicular to the line of DCIP data. The earth is represented by a cross-section and this 2D earth is discretized using rectangular cells. Maxwell's equations are solved using a finite volume approach and the accuracy of the solution depends upon the cell size and the total volume. Smaller cells are needed around current or potential electrode sites where the fields or sensitivities change rapidly. The data define a primary region of investigation but the earth model must extend sufficiently far beyond that so the assumed boundary conditions are satisfied (see Figure 9b for details). It is essential to verify the modeling mesh via forward modeling prior to inverting the data. A half-space conductivity is forward modeled and the predicted data from this forward modeling can be viewed. The apparent resistivities should not deviate by more than several percent from the half-space values used in the forward simulation.

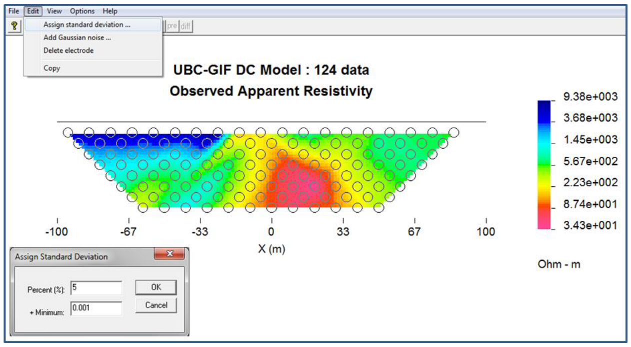
Topography must be approximated using rectangular cells. When a default mesh is used with UBC-GIF codes, the program builds a discretized 2D Earth. The topography is thus blocky.

5.4 Assigning uncertainties

For the inversion, each datum must be assigned an uncertainty. In practice this is challenging because the uncertainty may arise from additive noise, a misplaced electrode, 3D effects that are not modeled by a 2D geometry, numerical modeling errors etc. It is assumed that the error associated with d_i, the ith datum, is Gaussian with zero mean and standard deviation ϵ. Estimating values of ϵ is aided by having some knowledge about the acquisition of data in the field, the equipment used, and the likely sources of noise. A general recipe for assigning uncertainty is:

$$\epsilon_i = p * |d_i| + f \tag{34}$$

where p is a fractional percent and f is a floor in the same units of d. The percentage is important when there is large dynamic range in the data. For example, it helps capture errors that arise from electrode mis-allocations. Percentages of 5-7% are usually reasonable numbers to apply. The floor is a base measurement and reflects the minimum value of the signal that can be measured with the instrument. For example, a percentage (say 20%) of the average value of the voltages measured with at the largest electrode spacing might suffice as a floor value. It is extremely important not to set the floor to a value that is unrealistically small. If this is done then the inversion algorithm will concentrate its efforts on fitting data that have very small amplitude. An example of using a floor based upon data from a largest electrode separation and a percentage is shown in Figure 3. In Figure 3, the floor value of 0.001 has been assigned followed by 5% noise in the data. These values can be revised upon the assessment of the inversion results. Errors can also be adjusted



Percent (%): 5

+ Minimum: 0.001

Figure 3: Assigning standard deviations using the DCIP2D-DATA-VIEWER GUI.

for individual data points if you suspect any datum is particularly noisy. For example, it is not uncommon for all data values recorded at one electrode location to have additional noise, due for example to a poor electrical contact, a nearby metallic fence, or other reasons. In standard form, the data misfit Ψ_d is calculated according to equation 11. This is appropriate when the data errors are independent and Gaussian with zero mean and standard deviation ϵ. In reality, however, some data might have very large errors. These are outliers and if an incorrect uncertainty is supplied, the weighted difference will be very large and this datum will contribute disproportionately compared to other data. This arises because of the squaring operation in equation 11. In order to handle situations where there are outliers, a more robust norm such as a Huber norm can be implemented (See Background Theory for details) can be implemented. The Huber norm is calculated according to equation 12. It has a user-specified coefficient `c` and acts like a hybrid between l_1 and l_2 norms. Essentially normalized misfits with a value less than `c` are evaluated using l_2 and those above are evaluated using l_1.

5.5 Selection of model objective function

The model objective function is specified in equation 16 and is a critical component of the inversion procedure. It is an important conduit for incorporating geologic information. It controls the type of model that will be generated and also geologic detail. The inversion algorithm will find a model that minimizes this function subject to the constraint that the chosen model can generate predicted data that satisfy the misfit criteria. The model objective function can be subdivided into two components: the smallness ($\mathbf{W}_s$ components) and smoothness ($\mathbf{W}_x$ and $\mathbf{W}_z$ components). The components work hand-in-hand and the model objective function will

1. try to find a model that is as close as possible to a reference model defined either as a half space (by default a half space with a resistivity equal to a weighted average of measured apparent resistivities), or as some other, more complicated model defined by the user (if there is enough prior knowledge), and
2. be as smooth as possible in the X and Z directions. The reference model can be left in or omitted in the derivative terms.

The significance of each component is controlled using the alpha coefficients α_s, α_x, and α_z,. Therefore the user can request a model that emphasizes either component 1 (smallness) or component 2 (smoothness). The weighting functions $\mathbf{W}_s$, $\mathbf{W}_x$ and $\mathbf{W}_z$ can be incorporated to enforce more detailed information about the structure. Default values of these coefficients are determined by the program based upon the length scales of the survey and mesh.

Figure 4 illustrates the effect of changing smallness and smoothness parameters on the inversion results. For the program DCIP2D V5.0, the default specifications for these ”alpha” parameters have been found to work well as a first attempt, but experimentation and adjustment of the parameters defining the desired model type may be needed upon assessing the inversion results.

In addition to the smallness and smoothness coefficients, the new version of `DCIP2D` offers additional degrees of freedom to edit the model objective function. It is now possible to define the reference model in arbitrary form, as specified in equation 17. The parameter ρ which takes values

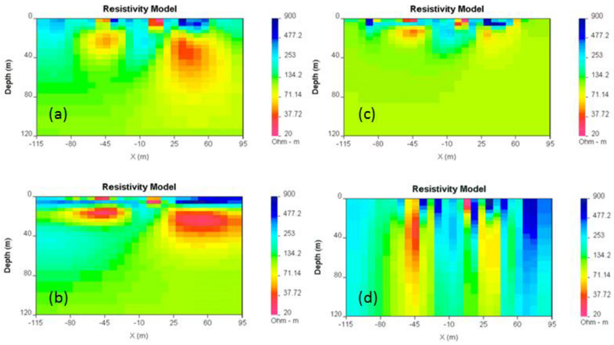


Figure 4: Effects of different values of $\alpha_s, \alpha_x, \alpha_z$: (a) [0.01,1,1]; (b) [0.01,1,0.01]; (c) [1,0.01,0.01]; and (d): [0.01,0.01,1]. All models produce the same data.

$1 \leq \rho \leq 2$ controls the character of the model. If $\rho = 1$ then the recovered model will tend to be blocky while if $\rho = 2$ we obtain our usual l_2 smooth model. Again, the reference model can be included, or not, in these terms. Examples of using this new objective function are shown in the Examples section of the manual and additional detail about the numerical implementation is provided in Background Theory section. Bounds constraints can be imposed on the model using the projected gradient method (Calamai and Moré, 1987). Each cell can be provided with an upper and lower bound (m^l and m^u), such that $m_i^L \leq m_i \leq m_i^u$.

5.6 Evaluation of the results

The following steps should be taken on order to properly assess the results of an inversion:

1. Check the log file. This file contains all the information about the input parameters and the inversion progress. Here are some key concepts of checking the log file:
   - Did the inversion end with convergence?
   - Have all the correct files been incorporated and inversion parameters properly set?
   - Was the target misfit achieved?
   - How many iterations were performed?
2. Predicted data should be compared to the observations using the `DCIP2D-DATA-VIEWER` GUI. The observed data and the predicted data should look nearly identical. To see variations between them, click the diff button in the data viewing window. This changes the second pseudo-section to a misfit map, which shows the differences between the two data sets (Figure 5). The normalized misfit map (normalized by the assigned standard deviation) should look random, with maximum values of some small percentage of the measured data (based upon noise specifications).
3. The resulting model should be checked using the `DCIP2D-MODEL-VIEWER` GUI (provided with DCIP2D v5.0) Select Padding cells in the Options menu in this GUI to specify how many padding cells to drop from the display. You can also adjust the minimum / maximum values

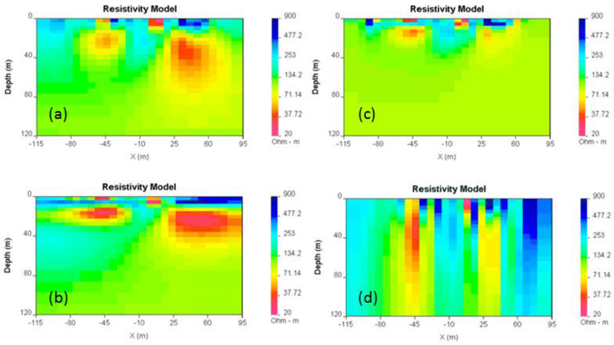
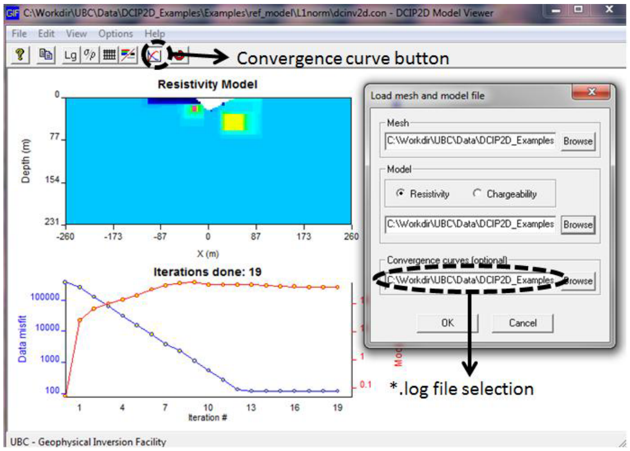


Figure 5: (a) Comparison of predicted data with the observed data. (b) Viewing the difference between the predicted data and observed data, normalized by standard deviation.


C:\Workdir\UBC\Data\DCIP2D_Examples\Examples\ref_model\L1norm\dcinv2d.con - DCIP2D Model Viewer

Load mesh and model file

Mesh: C:\Workdir\UBC\Data\DCIP2D_Examples

Model: ● Resistivity ○ Chargeability

C:\Workdir\UBC\Data\DCIP2D_Examples

Convergence curves (optional): C:\Workdir\UBC\Data\DCIP2D_Examples

Figure 6: Viewing the model and the inversion progress.

for the colour scale - necessary for comparing various models (see GUI usage manual for details). The progress of the inversion (or the convergence curve) during its iterations should also be checked (Figure 6). In the model viewing window, the algorithm's progress can be displayed graphically by selecting the "Curves" toolbar button in the View menu. The resulting graph shows how the values of misfit and model norm varied at each iteration. (Model norm is the value of the model objective function - this is what we are trying to minimize. The algorithm is programmed to add structure gradually in order to find a model that explains the data - i.e. it works on reducing the misfit value (blue curve) until the target misfit is reached. Then it must try to minimize the model norm without changing misfit. Thus, you should see a slight drop in the model norm value (red curve) until no more adjustments can be made to improve the situation.

4. Is the model geologically reasonable? It is important to decide whether the resulting model is geologically reasonable. This final consideration is more subjective. A simple example is shown here, in Figure 7 in which data produced by calculating data over the true 2D model (Figure 7a) are inverted twice to produce two inversion results which are both inadequate. Figure 7b shows the model, which is underfit (a model recovered when the target misfit was

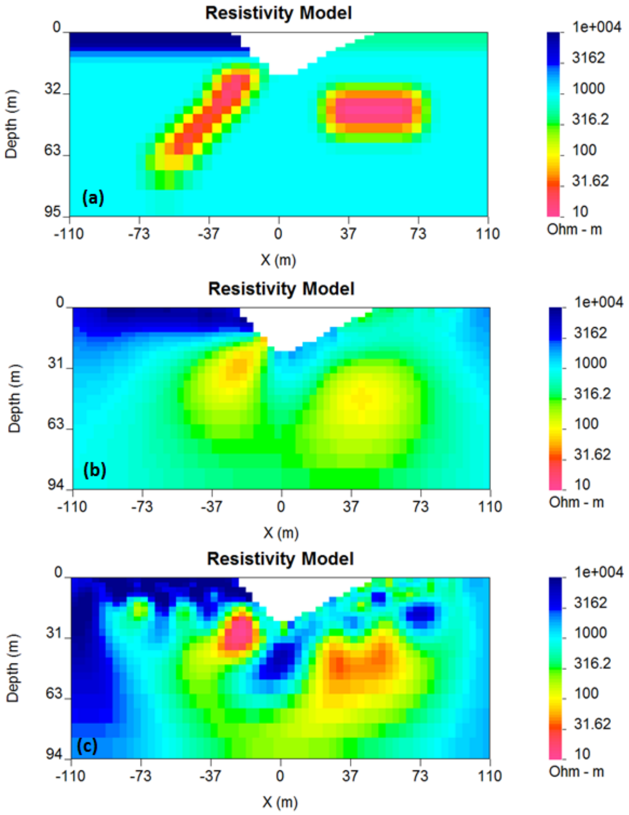


Figure 7: Viewing the model and the inversion progress.

too large). The program has stopped looking for details when predictions look only somewhat like observations. The image in Figure 7c show an overfit model (a model recovered when the program has tried too hard to find details that explain every nuance in the observation and resulted in adding structure, which does not exist). In both cases the `CHIFACT` should be reviewed. In the underfit case it should be made smaller, in overfit case, bigger..

Another important concept to keep in mind during the verification of the inversion results is the depth of investigation concept. Some of the structure observed in the final model is strongly controlled by the data but other structure is controlled by the details of the regularization functional. By performing two inversions with different reference models or by computing the sensitivities it is possible to obtain some insight regarding which areas of the model are not controlled by the data. These should be removed (or blurred out) from the image before final presentation. An example of this concept is shown in Figure 8. The models in (a) and (b) were respectively recovered from inversions using reference models of a 1000 ohm-m and a 106 ohm-m (given by a default inversion). The DOI analysis was carried out and a threshold value of 0.5 was used to omit parts of the model domain on the model recovered from a 1000 ohm-m background. See figure 14 (c). It is important to note that the model GUI will also perform this analysis given the sensitivity.txt file or multiple output model files.

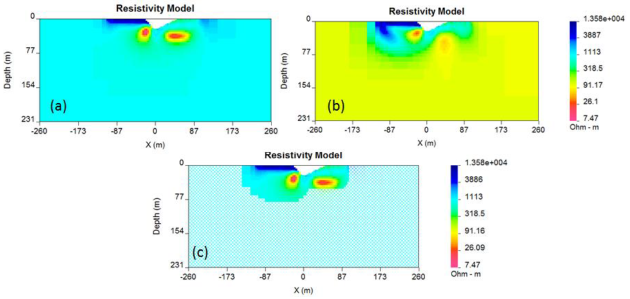


Figure 8: (a) The model recovered using a 1000 ohm-m background. (b) The model recovered from using a background of 106 ohm-m. (c) The model in (a) truncated to the depth of investigation using a cutoff value 0.5.

6 Examples

6.1 Forward modelling

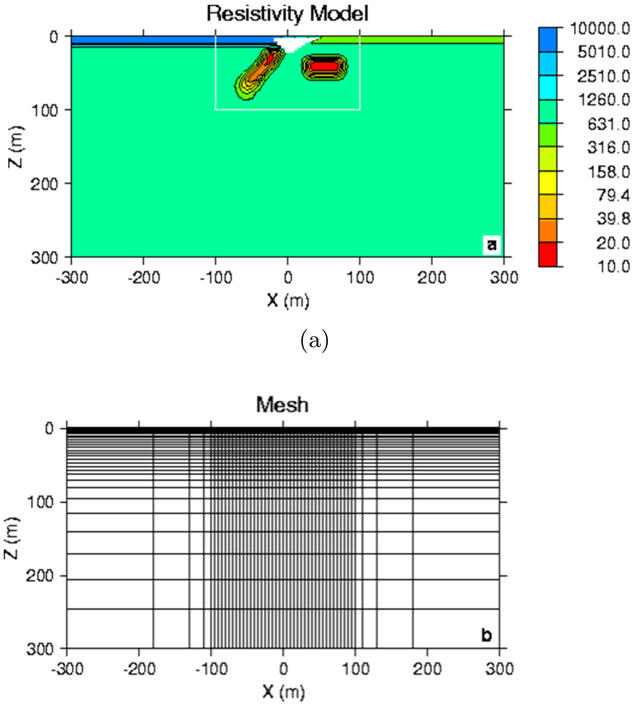
To illustrate the DC resistivity forward modelling algorithm, we generate synthetic data that would be acquired over the 2D conductivity structure shown in Figure 9a. The model region consists of a 10 meter thick overburden having conductivity of 0.1 mS/m on the left and 2 mS/m on the right. A V-shaped valley is cut out to simulate the surface topography. Two conductors are buried in the underlying background of 1 mS/m. On the left, a dipping conductor having a dip of $135^/circ$ and conductivity of 100 mS/m is buried at a depth of 20 m to the top. On the right, a 50 meter long, 20 meter thick conductive block of 100 mS/m is buried at a depth of 25 meters. Rather than keep the model as discrete blocks we have attempted to make it more geologically realistic by applying a smoothing filter to the model. The model is divided into 48 cells in the $x-$direction and 27 cells in the $z-$direction including a total of 1296 cells. The finite difference mesh is shown in Figure 9b. In the survey, surface electrodes are located every ten meters in the interval $x = (-100, 100)$ meters. We compute the potential differences from a pole-dipole array with the potential electrodes on the right. Our designation for this is *PDR* (Potential Dipole Right). There are 19 possible current electrode locations and we record the data for each electrode to a maximum $n-$spacing of 8. The observed data set consists of 124 potential difference values. It is our intention to use these data as input to an inversion. In order to make them more realistic we contaminate each datum by adding Gaussian noise having a standard deviation equal to 5% of the true potential. The apparent resistivity pseudo-section is shown in Figure 10a. That figure can be compared with the conductivity model within the region of interest in Figure 10b. There is some manifestation of the horizontal conductive block, but much of the conducting “pants leg” anomaly seen is really the result of the near surface variation of conductivity and topography.

The forward modelling for the IP data is also performed with the program DCIPF2D. To calculate IP data, the program performs two DC forward modelling routines and the IP data are generated by the operations indicated in equation 8. For a synthetic example, we choose the chargeability model in Figure 11a. It consists of a chargeable layer at the surface with $\eta = 0.05$ and two chargeable blocks of $\eta = 0.15$ at depth. The 124 IP data collected in the survey are also contaminated with Gaussian noise having a standard deviation equal to 5% of the accurate value of the apparent chargeabilites, plus an additional floor of 0.001. The apparent chargeabilities are plotted as percentages in pseudo-section form in Figure 11b. There is little manifestation of the chargeable block, which is the target for this inversion, but the user is faced with the difficulty of deciding how much of the high chargeability feature sloping downward to the left is the result of a pant leg from the termination of the surface chargeable block.

6.2 DC Inversion of the forward model

We use several synthetic examples to illustrate various aspects of DCIP2D v5.0. The emphasis of the test examples is to show the newly added features of the inversion programs. A synthetic conductivity model is shown in Figure 10a and resultant synthetic data I shown in Figure 10b.

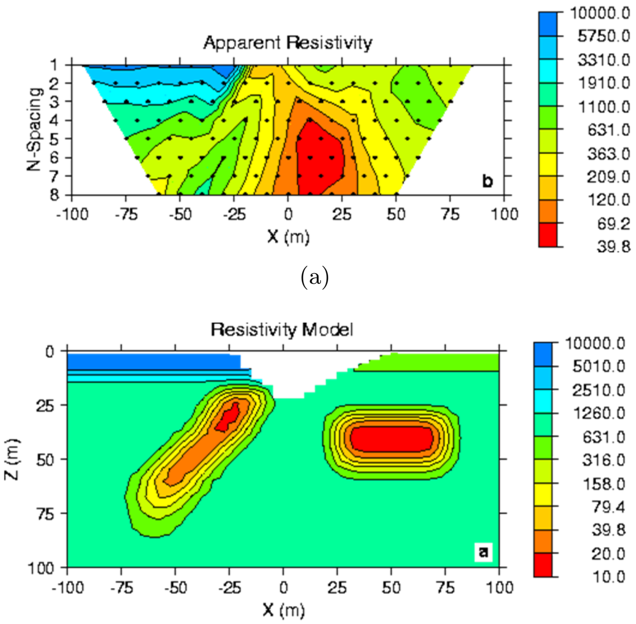
The synthetic model consists of two conductors buried in a uniform half space, which is overlain



(a)

(b)

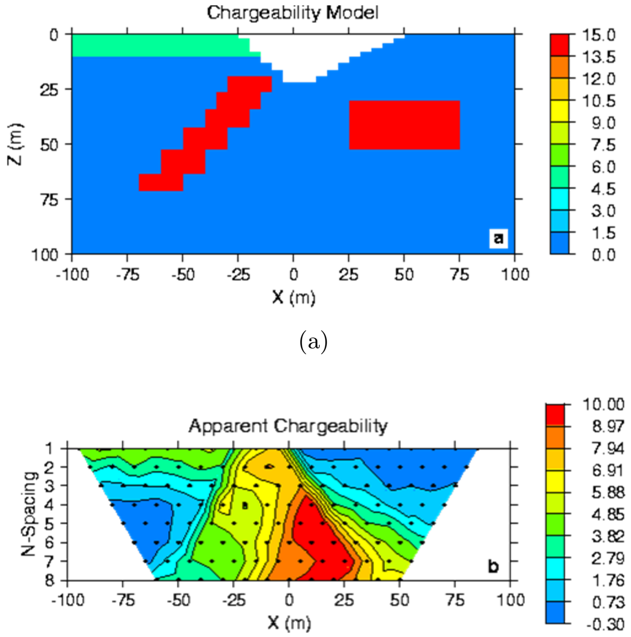
Figure 9: (a) The synthetic model consists of two conductors buried in a uniform halfspace overlain by an overburden of variable conductivity. A V-shaped valley simulates the surface topography. The region of interest is outlined by the white lines, but padding cells are added so that the correct boundary conditions can be applied during the forward modelling. (b) The finite-difference mesh used in the modelling.



(a)

(b)

Figure 10: (a) The conductivity model, which is the same as the region outlined in Figure 9a. (b) The apparent resistivity pseudo-section measured using a pole-dipole array with $a = 10$ m and $n = 1, 8$. The data have been contaminated by Gaussian noise.



(a)

(b)

Figure 11: (a) The chargeability model associated with the conductivity (Figure 10b). (b) The apparent chargeability pseudo-section measured using a pole-dipole array with $a = 10$ m and $n = 1, 8$. The data have been contaminated by Gaussian noise.

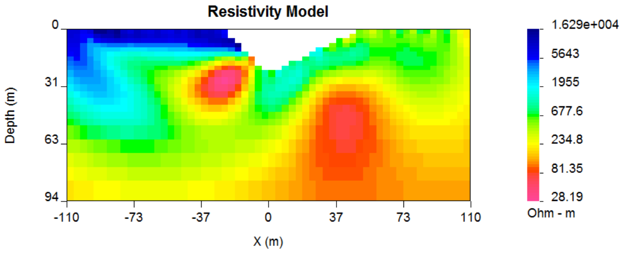


Figure 12: Inversion of synthetic DC data using all-default mode.

by an overburden of variable conductivity. A V-shaped valley is cut out to simulate the surface topography. The background has a conductivity of 1 mS/m and the overburden has a conductivity of 0.1 mS/m on the left and 2 mS/m on the right. The buried conductor on the left has a dip of 135o and a conductivity of 100 mS/m, and it is buried at a depth of 20 m to the top. The conductor on the right is a horizontal and conductive block of 100 mS/m buried at a depth of 25 m. The forward modelling uses a mesh of 48 cells in the x-direction and 27 cells in the z-direction so there are 1296 cells. In the survey, surface electrodes are located every 10 m in the interval $x = (-100, 100)$ m. We have simulated pole-dipole data with a=10 m and n=1, 8. The data have been contaminated with independent Gaussian noise whose standard deviation is equal to 5% of each accurate datum.

We have carried out nine DC inversions of the above DC data set. The examples were designed to illustrate the performance of the inversion program when different combinations of input parameters are used.

6.2.1 DC Inversion: All default

First, the synthetic data were inverted using the all-default option:

```
OBS LOC_X obs_dc.dat    !  DC data
TOPO FILE topo.dat      !  Topography
MESH FILE dcinv2d.msh   !  Mesh
```

In this file, the first line indicates that the data file obs_dc.dat is of surface format. The second line contains the reference to topography file topo.dat. The third line is in reference to the mesh file]dcinv2d.msh. If there is no topography it is not necessary to include either the mesh, or the topography file in all-default mode. DCIP2D v5.0 will construct a mesh in automatic mode and consider topography to be zero. In our case we have user-defined topography and mesh file is provided. The result of the all-default inversion is shown in Figure 12. The best fitting half space was approximately 120 ohm-m.

6.2.2 DC Inversion: CG solution using a constant reference model

In the next example, definition of some parameters has been set to user-defined and changed. In real life this can be done if there is a higher level of certainty regarding some starting parameters

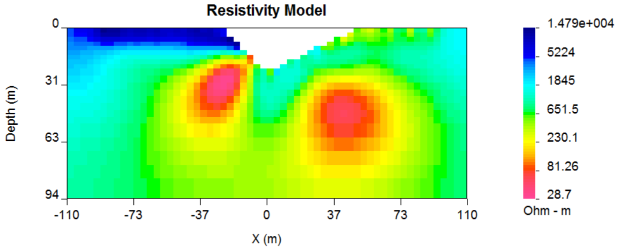


Figure 13: Inversion of synthetic DC data using user-defined smallness parameter (α_ss) 1000 Ohm m half space as both: reference and a starting model.

(prior information). The control file for the next example is provided below:

```
OBS LOC_X obs_dc.dat          !  DC data
TOPO FILE topo.dat            !  Topography
MESH FILE dcinv2d.msh         !  Mesh
ALPHA VALUE 1.e-3 1.0 1.0     !  Alphas
INVMODE CG                    !  Use CG
REF_MOD VALUE 1.e-3           !  Reference model
INIT_MOD VALUE 1.e-3          !  Initial model
```

In this example, the fourth line indicates that the smallness coefficient (α_s) is now user-defined (set to 0.001). The fifth line means that the system solver has been switched from default (Singular Value Decomposition or SVD solver) to the Conjugate Gradient solver (CG). The reference model has been set to 0.001 S/m (or 1000 Ohm m). The initial model has been set to the same as reference model. The results of applying these control file parameters are shown in the inversion model in Figure 13.

The basic features of the models in Figure 12 and Figure 13 are similar. Both conductors have been located and so has the resistive overburden on the left. (Compare with the synthetic model in Figure 10b). Nevertheless, there are differences. Figure 13, which uses a more resistive reference model, supports the interpretation of closure for the body on the right. There is also an odd structure emanating from the left most part of the resistive overburden observed in Figure 12 that is not observed in Figure 13. The primary differences between the two models can be explained through the use of a DOI (Depth of Investigation) plot. For the present we use Figure 13 as a reference and Figure 12 as an additional model.

When the inversion volume is cut with respect to the DOI, then differences in the images are no longer so apparent. For the remainder of the example section we shall use the reference the model described in Figure 13 (1000 Ohm m half space) as the default model.

Depth of Investigation (DOI)

Models produced by inversion of DC resistivity data tend to approach the background conductivity of the reference model. At those depths the recovered model is no longer being influenced by

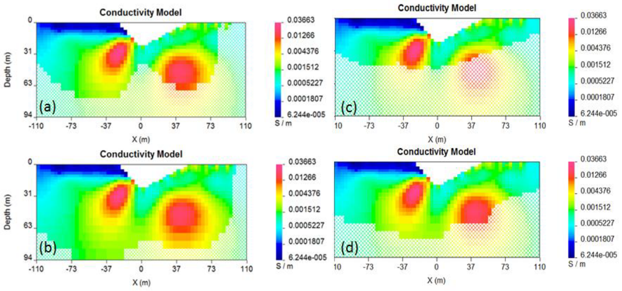


Figure 14: Assessing the depth of investigation (DOI): (a) based on recovered model (cut-off=0.1), (b) based on recovered model (cut-off = 0.4), (c) based on sensitivity (cut-off = 0.5), and (d) based on sensitivity (cut-off = 0.6).

the data. We can use this result to help estimate our depth of investigation. If there are at two reasonable models obtained using different reference models, the two models can be compared to identify which regions of the model are most significantly affected by the measurements. The results of doing this are explained next.

Using DCIP2D v5.0, the method is applied within the `DCIP2D-MODEL-VIEWER` GUI, using Depth of investigation option in the Options menu. There must be a second model that was recovered using the same mesh as the one being observed. Any two different inversions results can be used. Here we use 1000 Ohm-m halfspace as our best model and we want blank out those sections of the model that are not well controlled by the data. A second inversion using a background of 106 Ohm-m (the default value from the code) and used that to compute the DOI. In Figure 14(a-b) shows the model with cutoffs of 0.1 and 0.4.

Another option to assess the depth of investigation is through the analysis of the sensitivities. In DCIP2D v5.0 there is a capability to visualize the sensitivities using the `DCIP2D-MODEL-VIEWER` GUI (Figure 14c and Figure 14d). Generally, the lower sensitivities correspond to less reliable model parameters (deeper-seated cells); higher sensitivities correspond to those model cells, which have most effect on the data (usually closer to surface). A good way to assess the DOI is by plotting the model on the full mesh extent (including the padding cells, Figure 19). In this figure we use the DOI evaluated from 1000 and 106 Ohm-m half spaces (that is, the same as Figure 14a and Figure 14b). As the DOI threshold decreases we limit the region of the model to that which is most controlled by the data. See (Figure 15a-c). The final choice of cutoff is selected by the user.

6.2.3 DC Inversion: Non-uniform reference model

The next example is very similar to the previous inversion, with an exception that a different reference model is introduced (Figure 16). As opposed to the previous example, where the reference model was set to a 1000 Ohm m half space, the new model includes an elongated conductive (10 Ohm m) rectangular block. The elongated block has the same value as the conductivity anomaly but the boundaries do not coincide. Moreover the block in the true model has smoothed boundaries. In summary, the supplied reference model has captured some aspects of the true conductivity but it is not an exact reflection of what is there. This example has been contrived to illustrate what

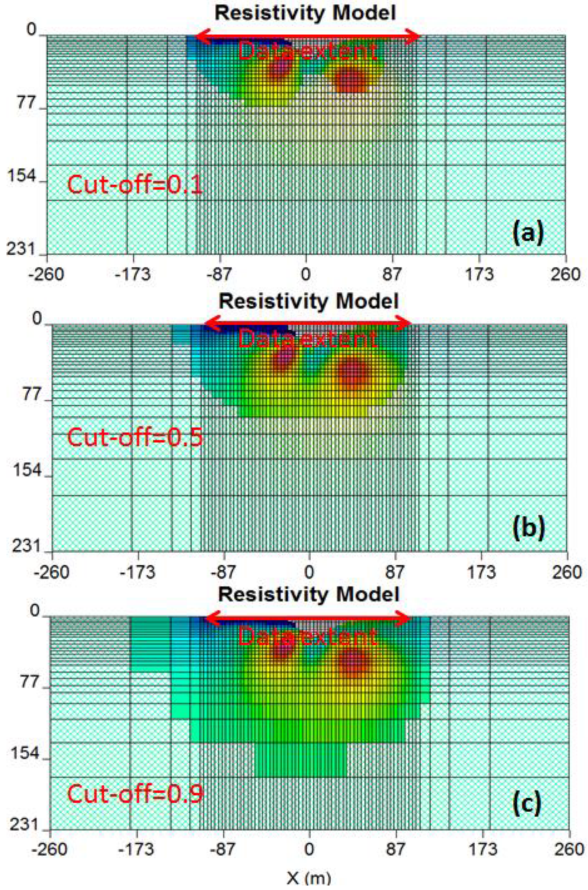


Figure 15: Assessing the depth of investigation (DOI): (a) based on recovered model (cut-off=0.1), (b) based on recovered model (cut-off = 0.4), (c) based on sensitivity (cut-off = 0.5), and (d) based on sensitivity (cut-off = 0.6).

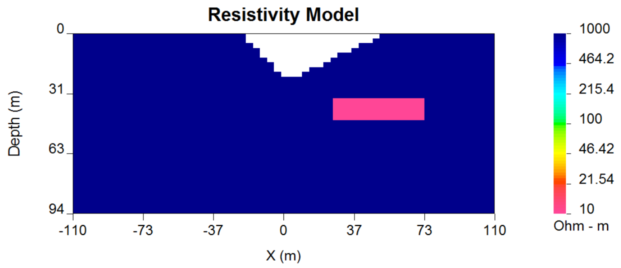


Figure 16: Reference model applied for the synthetic example illustration.

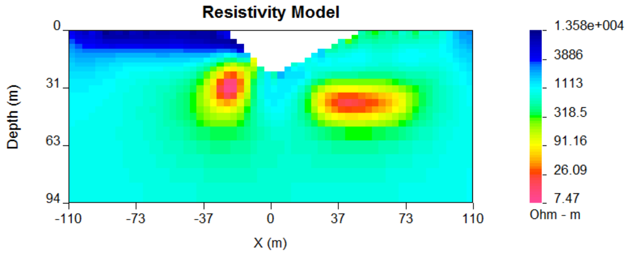


Figure 17: Reference model applied for the synthetic example illustration.

happens with the options of including, or omitting, the reference model in derivative terms in the objective function according to equations 16 and 19.

In the first example (control file provided below) the reference model was used in only the smallest model component.

```
OBS LOC_X obs_dc.dat         !  DC data
TOPO FILE topo.dat           !  Topography
MESH FILE dcinv2d.msh        !  Mesh
ALPHA VALUE 1.e-2 1.0 1.0    !  Alphas
INVMODE CG                   !  Use CG
USE_MREF FALSE               !  Ref out of spatial terms
REF_MOD FILE new_ref.con     !  Reference model
INIT_MOD VALUE 1.e-3         !  Initial model
NITER 40                     !  Max iterations
```

In this control file line 7 now indicates that the reference model should be read from a file, rather than assigned a constant value; line 6 indicates that the reference model should be defined in non-derivative terms and line 9 is indicating that the maximum number of iterations for this inversion should not exceed 40. The results of this inversion can be seen in Figure /reffig:synWithRef.

This is a superior model compared to that in Figure 13. The magnitude of the conductive anomaly is much better recovered, although at 7.6 Ohm-m it is slightly less resistive than the true value of 10 Ohm-m. It has a well-defined elongated shape with steep gradational boundaries that are good representations of the true model. If we are more confident in the locations of the boundaries of the block in the reference model, then this can be incorporated into the inversion.

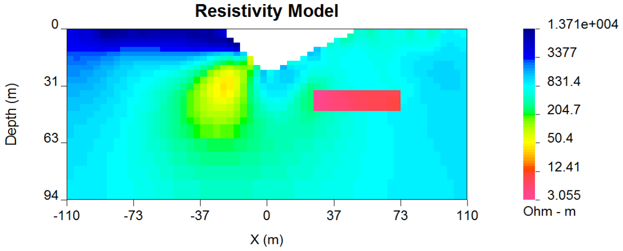


Figure 18: Reference model applied for the synthetic example illustration.

We next carry out an inversion in which the reference model is included in the derivative terms. Below is the control file used for this inversion.

```
OBS LOC_X obs_dc.dat          !  DC data
TOPO FILE topo.dat            !  Topography
MESH FILE dcinv2d.msh         !  Mesh
ALPHA VALUE 1.e-2 1.0 1.0     !  Alphas
INVMODE CG                    !  Use CG
REF_MOD FILE new_ref.con      !  Reference model
INIT_MOD VALUE 1.e-3          !  Initial model
NITER 40                      !  Max iterations
```

The line (USE_MREF_FALSE) from the previous example has been eliminated, switching the inversion into the default mode (reference model is defined in the derivative terms in default mode). This line also could have been changed to USE_MREF_TRUE).

The result is shown in Figure 18 and it produces a model that has boundaries at the same location as the reference block and there is even more over-shoot of the conductivity. For this example however, putting in the reference model into the derivative terms is stronger information than is justified. In most cases, the previous solution, where the reference model was left out of the derivative terms is preferable.

This is not always the case. Consider a situation where the goal is to find a body beneath an overburden layer. The model and the reference model are shown in Figure 19. It might be supposed that information about the overburden thickness and its resistivity have been obtained through drilling. Two inversions are carried out. In the first (Figure 20a) the reference model is omitted from the derivative term and the overburden boundary is characterized by a smooth transition. In the second case (Figure 20b) the reference model is included in the derivative terms and the result is a cleaner delineation of the overburden and better definition of the sought body.

6.2.4 DC Inversion: Incorporating inactive cells constraint

In the next example it is illustrated how drilling data can be incorporated in the inversion using fixed cells constraint. In this example, the reference model has been set to the same elongated conductive block model as shown in Figure 19. The difference is that in this case additional

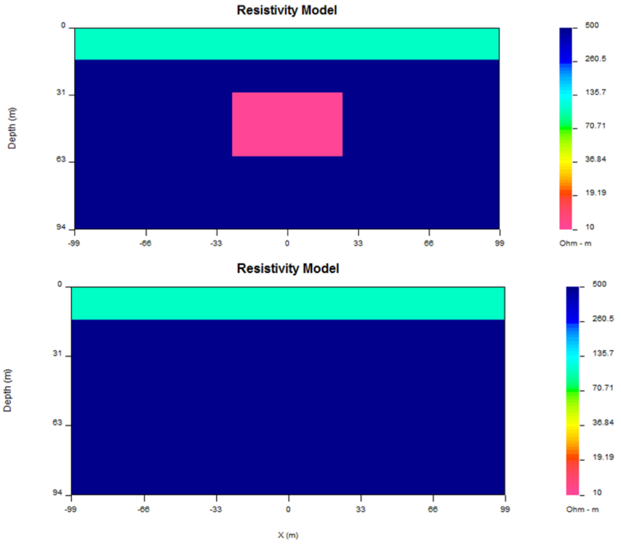


Figure 19: A conductive block underneath the overburden: (a) the true model and (b) the reference model.

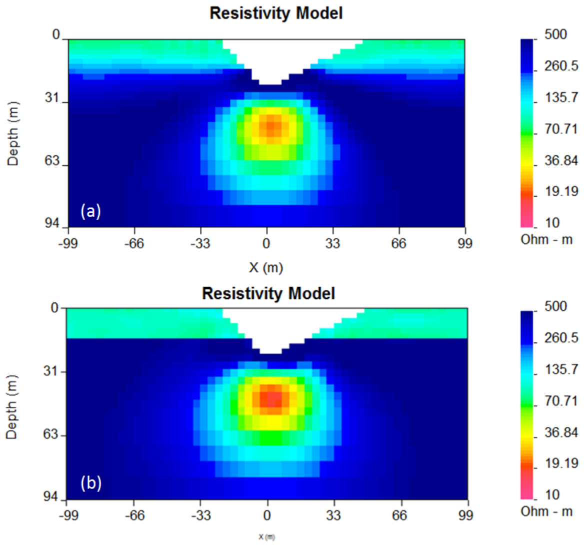


Figure 20: Inversion results when (a) the reference model is not included in the derivative terms and when (b) the reference model is defined in derivative terms.

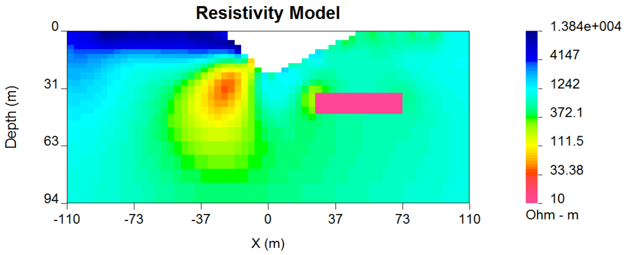


Figure 21: Recovered model when the reference model cells are inactive and they do not influence the neighbouring cells.

information has been incorporated by fixing some reference model cell values. The values are taken from the reference model file (ref_new.con) but their values are fixed using active cells file (`ACTIVE_CELLS active.txt`), defined in line 6 of the control file provided below.

```
OBS LOC_X obs_dc.dat          !  DC data
TOPO FILE topo.dat            !  Topography
MESH FILE dcinv2d.msh         !  Mesh
ALPHA VALUE 1.e-3 1.0 1.0     !  Alphas
INVMODE CG                    !  Use CG
REF_MOD FILE new_ref.con      !  Reference model
INIT_MOD VALUE 1.e-3          !  Initial model
ACTIVE_CELLS active.txt       !  Active cells
```

The active file format was previously discussed within the model subsection in the Elements section of the manual, however another example is provided below:

```
1  1  1  1  1  1  1  1  1  1  1  1  1  1
1  1  1  1  1  1  1  1  1  1  1  1  1  1
1  1  1  1  0  0  0  0  1  1  1  1  1  1
1  1  1  1  0  0  0  0  1  1  1  1  1  1
1  1  1  1  0  0  0  0  1  1  1  1  1  1
1  1  1  1  1  1  1  1  1  1  1  1  1  1
1  1  1  1  1  1  1  1  1  1  1  1  1  1
```

The format of this file is consistent with the model file, and the values equal to 1 define the model cells marked as active, while values equal to 0 define the model cells marked as inactive (without the capability affect the neighbouring cells). The case when inactive cells do not influence their neighbours is shown in Figure 21.

If it is desired to have the inactive cells influence the values of neighboring cells, then their values are set to -1 as in the file below. The resultant inversion model is shown in Figure 22. The region of high conductivity has been extended away from the reference model and the anomaly smoothly transitions to the background.

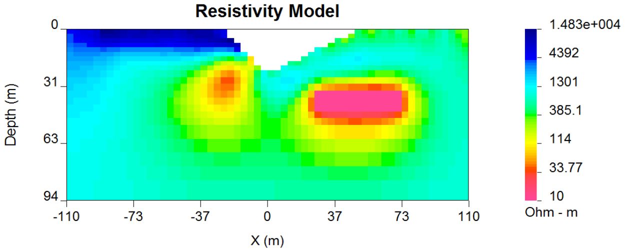


Figure 22: Recovered model when cells are inactive, but their values influence those of the neighbouring cells.

```
1  1  1  1   1   1   1   1  1  1  1  1  1  1
1  1  1  1   1   1   1   1  1  1  1  1  1  1
1  1  1  1  -1  -1  -1  -1  1  1  1  1  1  1
1  1  1  1  -1  -1  -1  -1  1  1  1  1  1  1
1  1  1  1  -1  -1  -1  -1  1  1  1  1  1  1
1  1  1  1   1   1   1   1  1  1  1  1  1  1
1  1  1  1   1   1   1   1  1  1  1  1  1  1
```

6.2.5 DC inversion: Using weighting functions

The next example illustrates the situation when prior information is incorporated using the weighting function file. The synthetic model for this example is the same as illustrated in Figure 19. Instead of reference model, a weighting file was used. The control file used for this inversion is shown below. The reference to the weighting file is provided in line 11 (WEIGHT W.DAT).

```
OBS LOC_X obs_dc.dat           !  DC data
TOPO FILE topo.dat             !  Topography
MESH FILE dcinv2d.msh          !  Mesh
ALPHA VALUE 1.e-3 1.0 1.0      !  Alphas
INVMODE CG                     !  Use CG
REF_MOD FILE 2e-3              !  Reference model
INIT_MOD VALUE 2e-3            !  Initial model
USE_MREF FALSE                 !  Ref out of spatial terms
WEIGHT w.dat                   !  Weighting file
CHIFACT 1                      !  Chi factor of 1
NITER 50                       !  Max iterations
```

The recovered model is illustrated in Figure 23 and is very similar to the model shown in Figure 20b. The alternative of using a weighting file instead of the reference model facilitated the technical implementation of the prior constraints and brings an additional degree of freedom in being able to adjust the level of certainty in the a priori information by editing the weighting coefficients. In our case, the weighting coefficients were edited for the $\mathbf{W}_z$ matrix, where the sixth interface

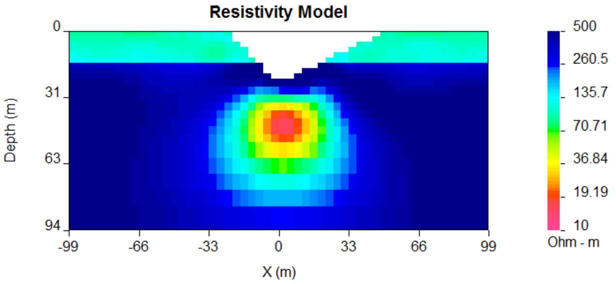


Figure 23: Recovered model from the inversion using weighting file

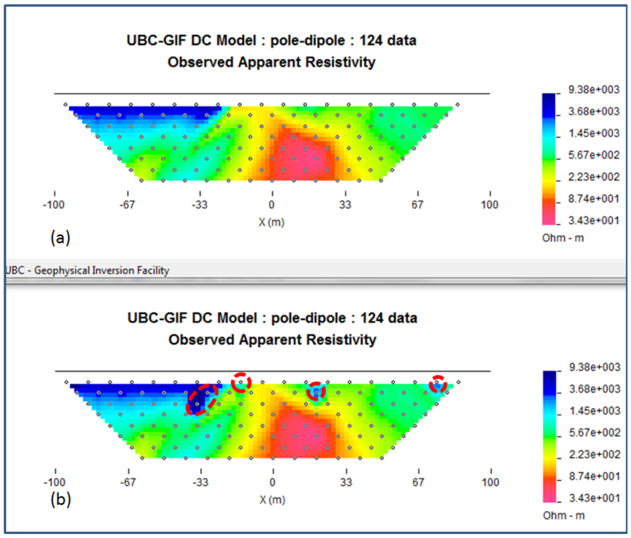


Figure 24: The (a) true data and (b) data contaminated with noise that will be inverted.

(corresponding to the bottom of the overburden) was set to 0.1 (as opposed to default weights of 1.0).

6.2.6 DC Inversion: Using the Huber norm for data misfit

The next example illustrates the effects that large data errors can have on the inversion and how these can be ameliorated with the Huber norm. The data are the same as used in previous examples except that 5 data have been severely perturbed. The inversions are carried out with the same standard deviation estimates, as used previously, a 1000 ohm-m background, and a data file contaminated with bad apparent resistivity values. Figure 24 shows the contamination introduced to the apparent resistivity file used for the inversions.

The contaminated data were inverted using a standard l_2 norm for the data misfit. The control file for this inversion is provided below:

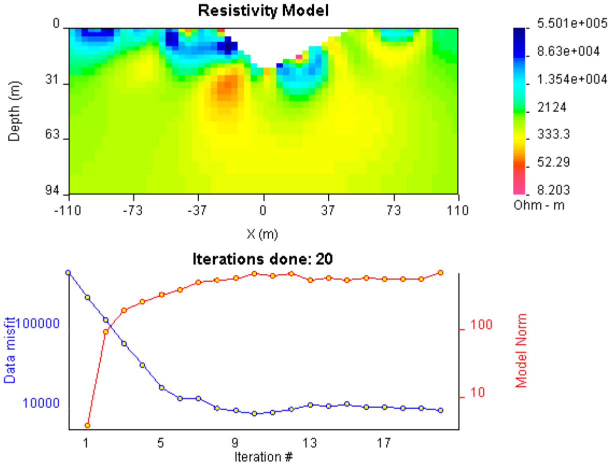


Figure 25: Recovered model (top) and conversion curves (bottom) from the inversion of the contaminated data. The data misfit utilized an l_2 norm.

```
OBS LOC_X obs_dc.dat         !   DC data
TOPO FILE topo.dat           !   Topography
MESH FILE dcinv2d.msh        !   Mesh
ALPHA VALUE 1.e-3 1.0 1.0    !   Alphas
INVMODE CG                   !   Use CG
REF_MOD FILE 1e-3            !   Reference model
INIT_MOD VALUE 1e-3          !   Initial model
USE_MREF TRUE                !   Ref everywhere
WEIGHT w.dat                 !   Weighting file
CG_PARAM 20 1.e-2            !   CG max iter and tol
NITER 20                     !   Max iterations
```

The results of the inversion are shown in Figure 25. The inversion ran for 20 iterations and the target misfit was not achieved and there were many artifacts. The reason is that the great effort was being made to fit the five erroneous data.

In Figure 26 we show the observed data and the normalized misfit. Three of the five outliers are distinct and they contribute a value of 2067.05 to the final misfit of 9303. By recognizing them as outliers, they might be winnowed from further analysis but two erroneous data have been over fit by the modeling and as a result produced incorrect structure. This has led to other, higher quality data, having large misfits. This is characteristic of non-robust norms.

In order to combat the effect that outliers in the data file may have on fitting the data using the l_2 measure, Huber norm was imposed on the data fit. The example of the control file with Huber norm is shown below:

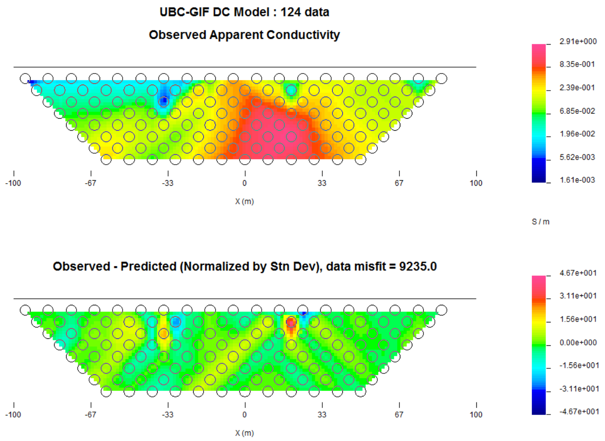


Figure 26: Observed data (top) and the normalized difference (bottom) from the inversion using an l_2 misfit measure.

```
OBS LOC_X obs_dc.dat         !  DC data
TOPO FILE topo.dat           !  Topography
MESH FILE dcinv2d.msh        !  Mesh
ALPHA VALUE 1.e-3 1.0 1.0    !  Alphas
INVMODE CG                   !  Use CG
REF_MOD FILE 1e-3            !  Reference model
INIT_MOD VALUE 1e-3          !  Initial model
USE_MREF FALSE               !  Ref out of spatial terms
HUBER 0.1                    !  Huber constant
NITER 40                     !  Max iterations
```

Line 9 in this control file has been set to so that all normalized data misfits with value greater than 0.1 will be evaluated with the l_1 measure. The results are shown in Figure 27 and they appear much better, than in previous case. Nevertheless, they can still be improved by recognizing the existence of the highly erroneous data and winnowing them from the inversion. incorrect structure. This has led to other, higher quality data, having large misfits. This is characteristic of non-robust norms. Although the recovery is far from perfect, the main conductor bodies are now shown with satisfactory detail, comparing to the l_2 normalization.

6.3 IP Inversion of the forward model

The inversion of IP data is almost identical to the inversion of DC resistivity data. The primary difference is that IP is a linear problem and the forward modeling matrix is the sensitivity matrix from the DC resistivity inversion. The IP inversion code has the same functionality as the DC resistivity code and the control lines are identical. One essential difference is that positivity is strictly enforced in the IP inversion. IP data can be negative but the intrinsic chargeability is always positive. There is no need to repeat all of the inversions done for the DC. Rather, we will invert only a few examples to illustrate the algorithm.

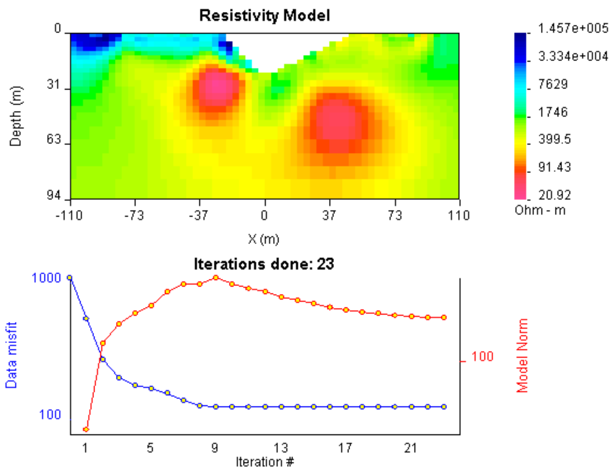


Figure 27: (top) The recovered model from inversion of contaminated data using Huber norm for the data misfit and (b) the convergence curves.

The inversion of IP data is almost identical to the inversion of DC resistivity data. The primary difference is that IP is a linear problem and the forward modeling matrix is the sensitivity matrix from the DC resistivity inversion. The IP inversion code has the same functionality as the DC resistivity code and the control lines are identical. One essential difference is that positivity is strictly enforced in the IP inversion. IP data can be negative but the intrinsic chargeability is always positive. There is no need to repeat all of the inversions done for the DC. Rather, we will invert only a few examples to illustrate the algorithm.

The examples were designed to replicate the capabilities of DCIP2D v5.0, shown using the DC examples. The conductivity models used for IP inversions were mainly those, acquired from the corresponding DC inversions.

6.3.1 IP inversion: Zero-chargeability reference model

The first example was carried out using zero-chargeability reference half space and the conductivity model acquired from inverting the dc resistivity with 1000 Ohm-m half space reference. The control file for this inversion is shown below:

```
OBS LOC_X obs_ip.dat          !  IP data
TOPO FILE topo.dat            !  Topography
MESH FILE dcinv2d.msh         !  Mesh
ALPHA VALUE 1.e-3 1.0 1.0     !  Alphas
INVMODE CG                    !  Use CG
REF_MOD FILE 1e-3             !  Reference model
INIT_MOD VALUE 1e-3           !  Initial model
COND FILE dcinv2d.con         !  Recovered conductivity
```

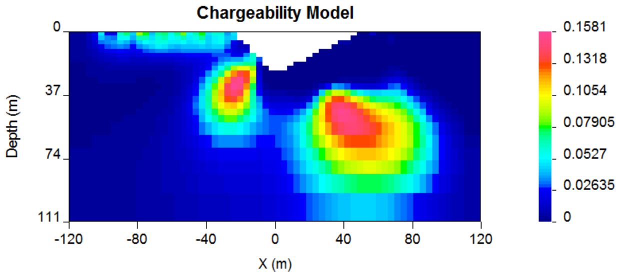


Figure 28: Recovered chargeability model for a zero chargeability reference model and 1000 Ohm-m conductivity model.

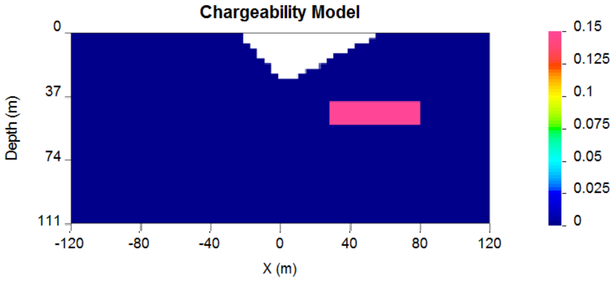


Figure 29: The reference model applied to the synthetic example for illustration.

On the last line of this control file, there is the reference to the conductivity file, an essential input parameter for an IP inversion. This file has to come from a corresponding DC inversion, carried out prior to the IP inversion. The results of this inversion are shown in Figure 28.

6.3.2 IP inversion: Non-uniform reference model

In the next example, similarly to the DC inversions, we have introduced a chargeable block into the reference model (Figure 29).

Further, the new reference model was introduced in the inversion and omitted from the derivative terms. The control file for the inversion is virtually identical as in case with analogous inversion of the DC data and is provided below. The resulting inversion is shown in Figure 30.

```
OBS LOC_X obs_ip.dat         !  IP data
TOPO FILE topo.dat           !  Topography
MESH FILE dcinv2d.msh        !  Mesh
ALPHA VALUE 1.e-2 1.0 1.0    !  Alphas
INVMODE CG                   !  Use CG
REF_MOD FILE ref_new.chg     !  Reference model
INIT_MOD VALUE 1e-5          !  Initial model
USE_MREF FALSE               !  Ref mod only in smallness
COND FILE dcinv2d.con        !  Recovered conductivity
```

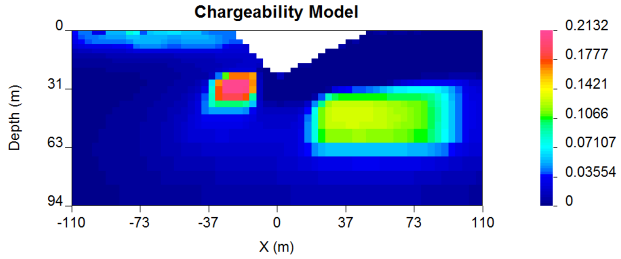


Figure 30: Recovered model from IP inversion using the non-uniform reference model in the smallness term.

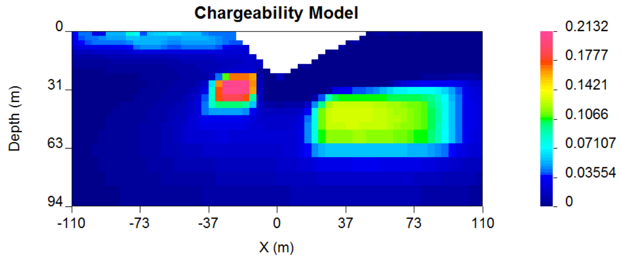


Figure 31: Recovered model from IP inversion Using l_1 measure (Ekblom norm) of model norm to recover a blocky model.

6.3.3 IP inversion: Using Ekblom measure to recover a blocky model

In this next example, the geological information is incorporated in the model objective function using the l_1 norm measure rather than the default l_2 norm. This allows recovery of a blocky model. The control file for this example is provided below, and the resultant inversion model is shown in Figure 31.

```
OBS LOC_X obs_ip.dat                      !  IP data
TOPO FILE topo.dat                        !  Topography
MESH FILE dcinv2d.msh                     !  Mesh
ALPHA VALUE 1.e-3 1.0 1.0                 !  Alphas
INVMODE CG                                !  Use CG
REF_MOD FILE 1e-5                         !  Reference model
INIT_MOD VALUE 1e-5                       !  Initial model
EKBLOM 1.  1.  1.  1.E-3 1.E-3 1.E-3      !  Ekblom variables
COND FILE dcinv2d.con                     !  Recovered conductivity
```

The resultant models are blocky and the central block has better defined boundaries than the deep block on the right. This arises because the right hand block is located close to the edge of the depth of investigation for the survey. To illustrate this we superpose the depth of investigation inferred by using the sensitivity function with a cutoff of 0.5. This is shown in Figure 32 to illustrate the depth of investigation (DOI) the model has been plotted on a larger scale.

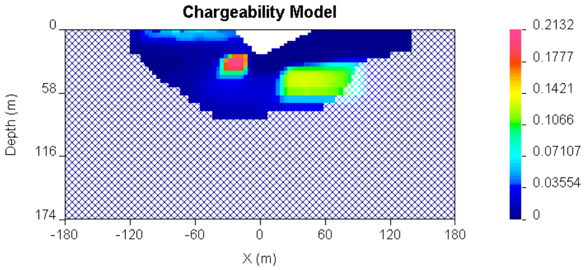


Figure 32: The depth of investigation (DOI) for the IP inversion with an l_1 model norm.

6.3.4 IP inversion: Reference model with inactive cells

This next example illustrates an inversion with a reference model with fixed cells (inactive). In this example, the inactive cells are representing a scenario when our constraints are acquired by incorporating borehole information. Out synthetic borehole is located on the profile at $x = 60$ (Figure 33a). This reference model is now different and involves only the knowledge we have from the borehole data (Figure 33b). The inversion was carried out in the mode, when the inactive cells may influence their neighbours and resulted in the chargeability distribution shown in Figure 33c. In this mode the inversion extends the chargeability of the fixed cells away from the reference block. The case is very similar to the analogous example shown in the DC inversion. The control file used for this inversion is provided below:

```
OBS LOC_X obs_ip.dat        !  IP data
TOPO FILE topo.dat          !  Topography
MESH FILE dcinv2d.msh       !  Mesh
ALPHA LENGTH 100 100        !  Length scales (m)
INVMODE CG                  !  Use CG
REF_MOD FILE new_ref.chg    !  Reference model
ACTIVE_CELLS active.txt     !  Active cell model
INIT_MOD VALUE 1e-5         !  Initial model
COND FILE dcinv2d.con       !  Recovered conductivity
```

The active file is shown below, the structure has been edited so that two cells (one in each direction) around the synthetic borehole are set inactive and with the capability to influence the neighbours (i.e., -1)

```
1  1  1  1  1  1  -1  -1  1  1  1  1  1  1
1  1  1  1  1  1  -1  -1  1  1  1  1  1  1
1  1  1  1  1  1  -1  -1  1  1  1  1  1  1
1  1  1  1  1  1  -1  -1  1  1  1  1  1  1
1  1  1  1  1  1  -1  -1  1  1  1  1  1  1
1  1  1  1  1  1  -1  -1  1  1  1  1  1  1
1  1  1  1  1  1  -1  -1  1  1  1  1  1  1
```

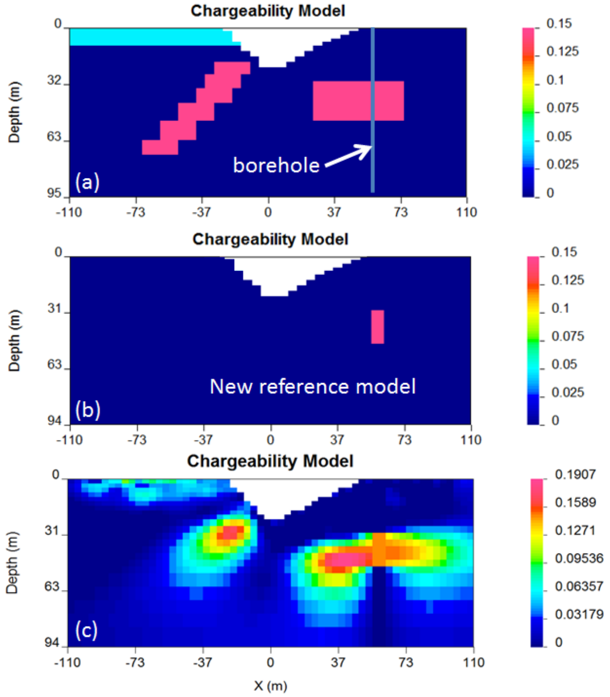


Figure 33: (a) The true chargeability model with the borehole location. (b) The new reference model created from the borehole information. (c) Recovered model with the borehole locations set to inactive with influence (-1) on neighbouring cells.

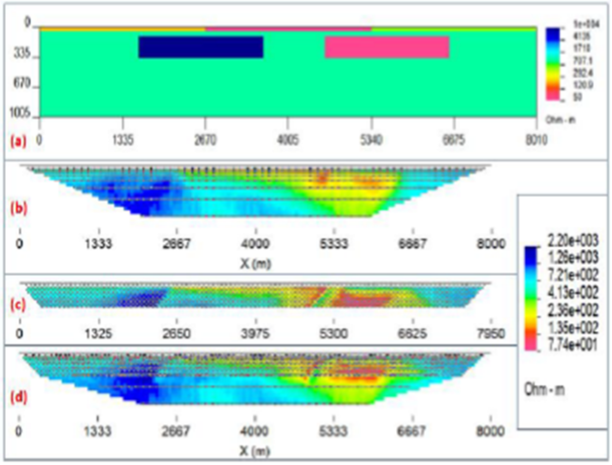


Figure 34: (a) The true model create for a large-scale synthetic data set by combining Wenner and Pole-dipole configurations. (b) The synthetic data from the Wenner array and (c) pole-dipole array are combined to get the (d) synthetic data for the entire data set.

6.4 Large data set example

In the next example a synthetic data set is introduced, where a Wenner array is combined with a pole-dipole array and covers an 8-km long profile. The synthetic model is a 1000 Ohm- m half space covered by a 50-m thick overburden of variable electrical resistivity (200 Ohm-m section on the left, followed by 50 Ohm-m section in the middle, followed by 500 Ohm-m section on the right. The background resistive media is hosting two rectangular bodies at 150-m depth each. The prism on the left side is resistive (10,000 Ohm-m resistivity) and the prism on the right side is conductive (50 Ohm-m) (Figure 34a).

For the Wenner array the following configuration was used: number of stations = 400; minimum a-spacing = 80 m; maximum a-spacing 1367 m (spreading coefficient: 1.5 to accommodate up to 8 spreads per station). The spreading coefficient in this case is the multiplier used to calculate the increased spread distance between the potential electrodes for each station, given the minimum separation a) The total number of data for Wenner array (considering number of stations and all possible separations) was 2610 (Figure 34b).

The pole-dipole synthetic survey used a=75 m and n=1,10. The current pole was fixed on the right hand side of the array. This resulted in a total number of pole-dipole data of 1005 (Figure 34c). The combined Wenner and pole-dipole data set contains 3615 data (Figure 34d).

This synthetic model was discretized with a mesh, composed of 17918 cells (including padding), with the smallest cells reaching 30 m horizontally and 15 m vertically for the core region (depth to 1 km).

This synthetic data set was contaminated with 5% Gaussian noise and inverted using l_1 measure for model objective function in order to accommodate a more blocky inversion result. The inversion control file is provided below:

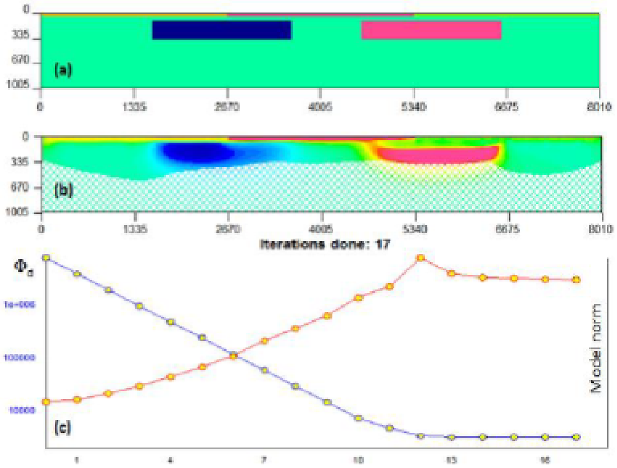


Figure 35: (a) The true model create for a large-scale synthetic data set by combining Wenner and Pole-dipole configurations. (b) The recovered model from inversion of the large synthetic data set with the Ekblom norm showing the DOI based on sensitivity analysis (threshold = 0.4). (c) The convergence curves show how the inversion performed.

```
OBS LOC_X obs_dc.dat                 !  DC data
MESH FILE mesh2d.msh                 !  Mesh
NITER 40                             !  Max iterations
INVMODE CG                           !  Use CG
REF_MOD FILE 1e-3                    !  Reference model
INIT_MOD VALUE 1e-3                  !  Initial model
CHIFACT 1                            !  data misfit to number of data
EKBLOM 1.   1.   1.   1e-3 1e-3 1e-3 !  Ekblom variables
BOUNDS 0.00001 0.02                  !  Global conductivity bounds
```

The inversion converged in 17 iterations (Figure 35a) and was able to reconstruct all of the features shallower than 500-m of depth. This is consistent with the depth of investigation for this survey, based on the sensitivity (Figure 35b).

The observed data were compared with the predicted data. The misfit is shown in Figure 36. The predicted data error does not exceed 3.9 standard deviations and overall data misfit is 3597.6.

Finally, the parallelization of DCIP2D v5.0 with OpenMP was analyzed on this example. It was inverted twice using 1 and 12 threads (6 cores with hyper-threading capability) with identical results. Running this example on one thread took 1:15:50.68 of CPU time, while running it on 6 cores (12 threads) resulted in convergence in 0:25:16.86 of CPU time, which is almost a threefold increase in productivity since the last release of DCIP2D.

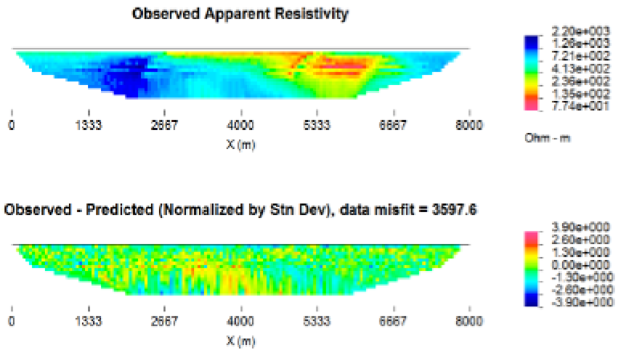


Figure 36: (a) Observed apparent resistivity (mixed Wenner/Pole-dipole data set) and the (b) data misfit, which is normalized by the standard deviation.

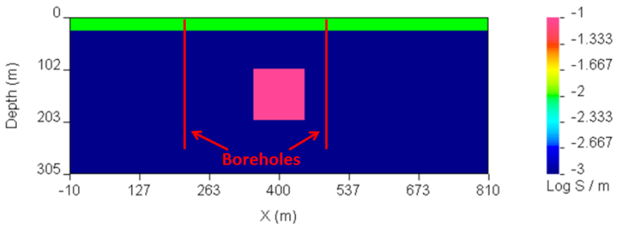


Figure 37: The synthetic model used for the three borehole examples.

6.5 Borehole examples

The next example is based on synthetic borehole DC data, where the simulated data are

1. Surface data simulated using dipole-dipole configuration

2. Borehole data simulated using larger dipole with one transmitter electrode placed in the borehole

3. Combination of surface and borehole data simulation

The synthetic data were generated using a 2D model (Figure 37) that contains a 25-m thick overburden of 100 Ohm-m over a 1000 Ohm-m half space. A 1 Ohm-m prism is located buried so that its top is 100 m below the surface. The prism is 100 m × 100 m. The model was discretized onto a 6869 cell mesh (104 horizontal by 66 vertical cells. Smallest cell equals 10 m by 5 m). The locations of two boreholes are also shown in the figure.

The first synthetic data set is surface-only dipole-dipole array. Electrodes are located every 25 m within the region (0 m, 800 m) and each dipole-dipole survey used n=1,15. A total of 30 transmitter locations were used. The total number of data is 345. The data were contaminated with 5% Gaussian noise and inverted with a chi factor of 1 (see control input file configuration provided below). The inversion has converged in 14 iterations. The results of the inversion of surface data are presented in Figure 38.

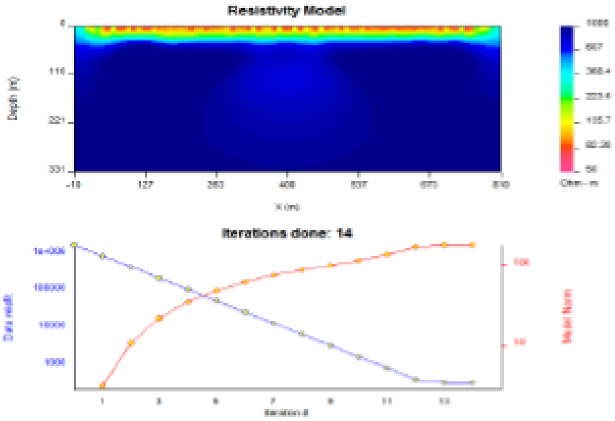


Figure 38: (a) The recovered model from the inversion of synthetic surface data set and (b) the associated convergence curves.

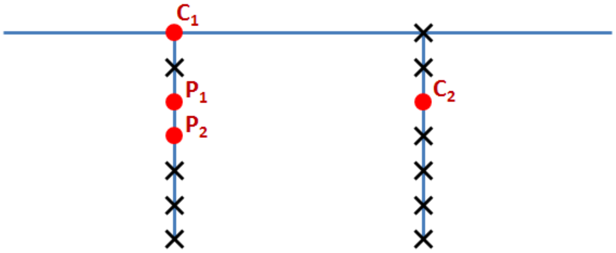


Figure 39: A schematic of the survey configuration for the synthetic borehole example. Red dots mark the electrode positions.

```
OBS LOC_XZ dc_surface.dat   !  General-formatted DC data
MESH FILE mesh2d.msh        !  Mesh
INVMODE CG                  !  Use CG
REF_MOD FILE 1e-3           !  Reference model
```

Next the same conductivity model was used to simulate a synthetic borehole data set for two boreholes located at $x = 200$ and $x = 500$. Electrodes were spaced every 25 m down each borehole to a maximum depth of 250 m. The transmitter electrodes are located in the opposite boreholes in all possible configurations. The schematic diagram illustrating this survey configuration is shown in Figure 39. This resulted in a total of 121 transmitters. Each transmitter configuration is a common-current source for up to 20 receiver dipoles in both boreholes. The total number of data is 2200. The simulated data set was contaminated by 5% Gaussian noise and inverted using same inversion parameters as the surface data set. The results are shown in Figure 40.

Further, the data sets were combined to accommodate 151 transmitter locations and 2545 data. The combined data set was inverted with the same parameters as each individual data set with the results shown in Figure 41. As it is evident from these three inversions, the surface geometry alone has strong limitations in depth resolution, while the borehole configuration has limitations in near-surface recovery and it is the combination of the two surveys, which allows better recovery

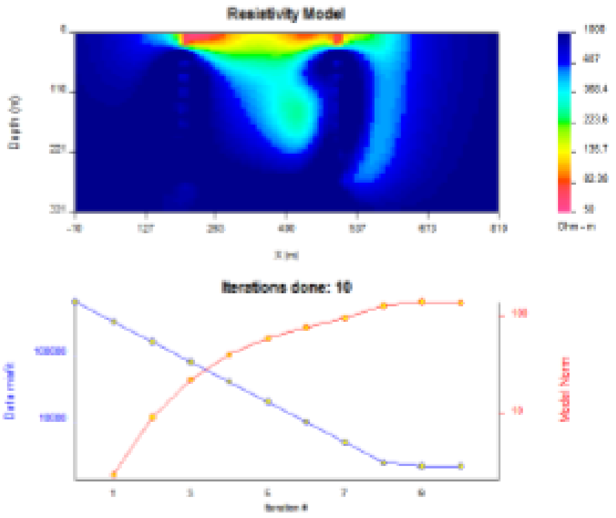


Figure 40: (a) The recovered model from the inversion of synthetic down-hole data set and (b) the associated convergence curves.

of the conductivity.

The inversion was then constrained using a weighting matrix with small weights for the vertical interface at the depth of 25 m, ensuring a sharper contrast across this boundary and large weights assigned to all horizontal interfaces up to the depth of 25 m, ensuring a smooth transition of electrical properties in this direction. This weighting matrix is designed, assuming there is prior knowledge about the overburden and defines the latter as a horizontal (laterally smooth) structure with abrupt transition in the vertical direction. The control file used for this inversion is shown below. The result of the inversion is shown in Figure 42.

```
OBS LOC_XZ obs_dc_n.dat    !  General-formatted DC data
MESH FILE mesh2d.msh       !  Mesh
INVMODE CG                 !  Use CG
REF_MOD FILE 1e-3          !  Reference model
WEIGHT FILE weights.txt    !  Weight file
```

It is evident from Figure 42, that not only did the weighting function assure clean resolution of overburden, but it also helped to remove some noise from the background, if compared with Figure 41.

The next step was to simulate a scenario, when the down-hole conductivity data is available. This was done using the inactive cells constraint. Figure 43a shows the new reference model with fixed cells along $x = 200$ and $x = 500$ to the depth of 250 m. The data were inverted using inactive cells constraint with no ability to affect the neighbouring cells (Figure 43b), with ability to interfere with the neighbours (Figure 43c) and in combination with the weighting matrix (Figure 43d).

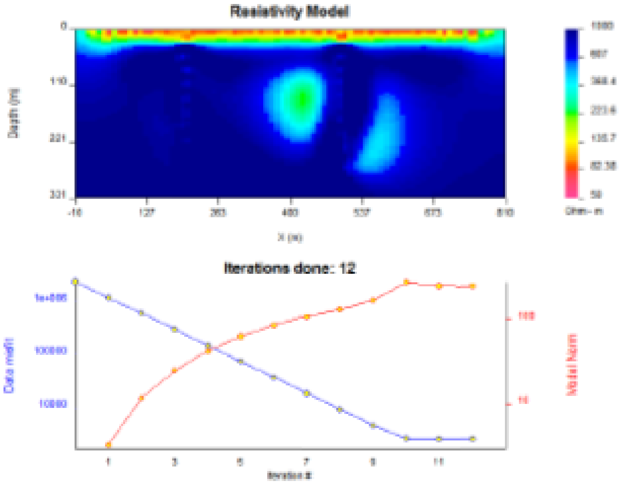


Figure 41: (a) The recovered model from the inversion of synthetic down-hole and surface data sets and (b) the associated convergence curves.

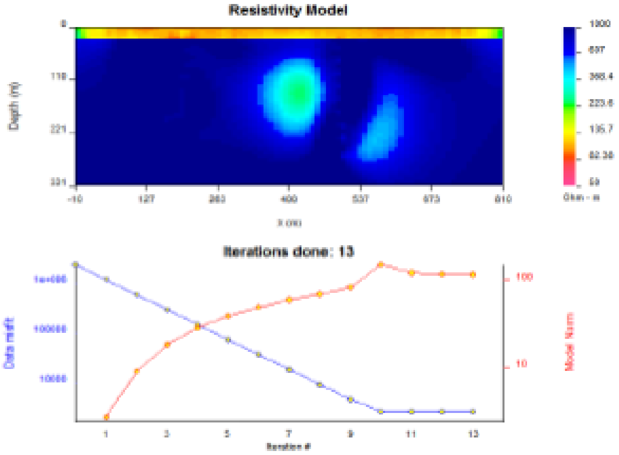


Figure 42: (a) The recovered model from the inversion of synthetic down-hole and surface data sets with a weighting constraint and (b) the associated convergence curves.

```
OBS LOC_XZ obs_dc_n.dat   !  General-formatted DC data
MESH FILE mesh2d.msh      !  Mesh
INVMODE CG                !  Use CG
REF_MOD FILE 1e-3         !  Reference model
ACTIVE_CELLS active.txt   !  Active cell file
```

Finally, the area of inactive cells was extended, simulating a scenario, when a-priori information suggests that the anomalous conductivity lies between the two boreholes. The final control file used for inverting data under these constraints is presented below:

```
OBS LOC_XZ obs_dc_n.dat   !  General-formatted DC data
MESH FILE mesh2d.msh      !  Mesh
INVMODE CG                !  Use CG
REF_MOD FILE 1e-3         !  Reference model
WEIGHT FILE weights.txt   !  Weight file
ACTIVE_CELLS active.txt   !  Active cell file
```

The results of the final inversion are presented in Figure 44.

6.6 Field example

In the next example the field data set had been collected over the Cluny prospect in Mt. Isa area, Queensland, Australia by MIM Exploration Pty, Ltd (Rutley et al., 2001). The data were collected using pole-dipole and dipole-pole array configuration. The length of the profiles reaches 2000 m in East-West direction. A total of 10 profiles of DC and IP data were collected with electrode spacing of 50 m. There is topographic disturbance along the profile ranging from 429 to 505 meters in absolute elevation above the mean sea level. The DC and IP data, collected over profile 13700 are shown in (Figure 45). The data sets were examined for bad data values (outliers) and based on this criterion 45 data had to be eliminated from the IP inversion.

Resistivity inversion was carried out using the error assignment of 0.0001 base +5% of the data and constant uniform reference and starting model (100 Ohm m). A default meshing option was used, resulting in 16.8-m wide and 8.33-m high cells in the core region. The control file used for the inversion is provided below with user-defined smallness coefficient and maximum number of iterations set to 50.

```
OBS LOC_X 2D_13700.dat   !  DC data
MESH DEFAULT             !  Created mesh
TOPO FILE topo.txt       !  Topography
INVMODE SVD              !  Use CG
REF_MOD FILE 1e-2        !  Reference model
INIT_MOD VALUE 1e-2      !  Initial model
CHIFACT 1                !  Data misfit to number of data
STORE_ALL_MODELS TRUE    !  Write out all models
ALPHA VALUE 1E-4 1 1     !  Alpha values
NITER 50                 !  Max iterations
```

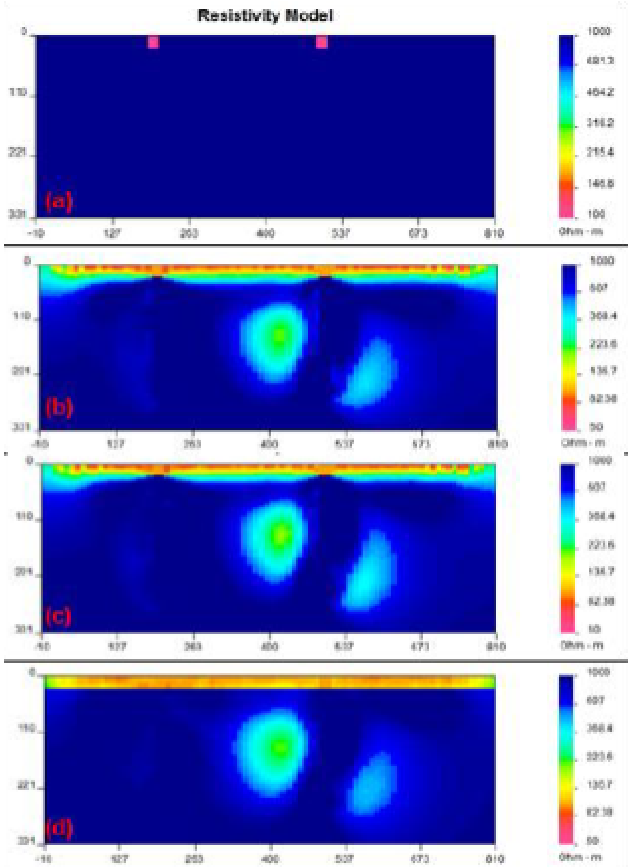


Figure 43: (a) The new reference model, accommodating the active cells. The inversion was then carried out such that the inactive cells both (b) influenced and (c) did not influence the neighbouring cells. Lastly, both the active cells and weighting file was combined to recover the model shown in (d).

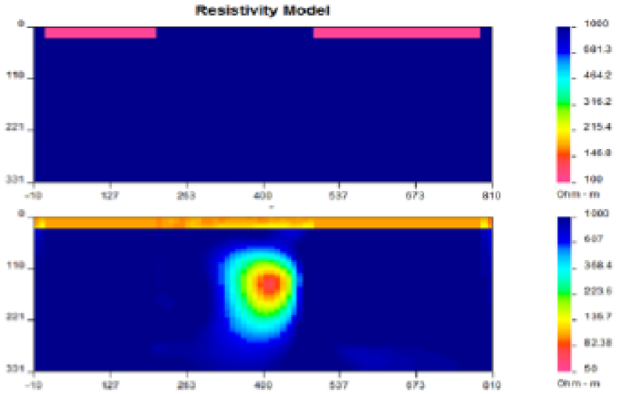


Figure 44: The reference model with an extended region of inactive cells is shown in the top panel. The recovered model from the subsequent inversion using both weighting and inactive cell constraints is presented in the bottom panel.

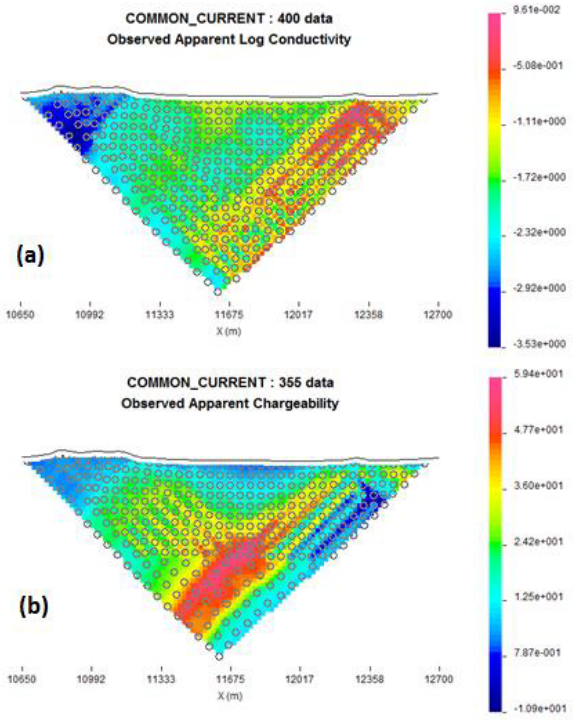


Figure 45: The observed (a) DC and (b) IP data measured along profile 13700 over the Cluny prospect, Australia.

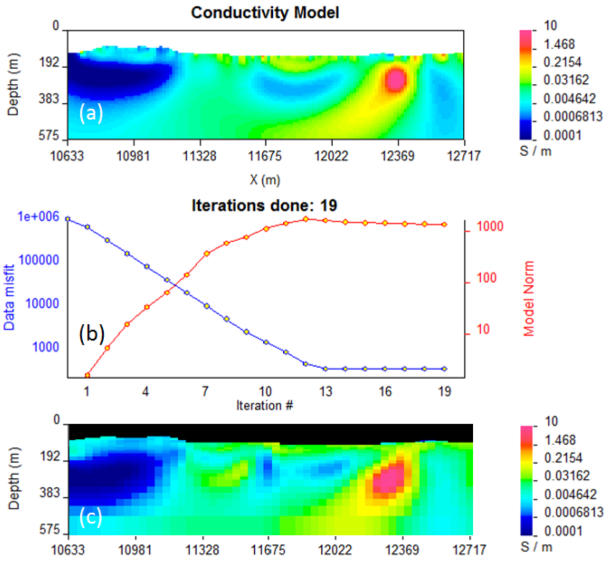


Figure 46: (a) The recovered model from inverting DC data along profile 13700, Cluny prospect, Queensland, Australia and the associated (b) convergence curves. For reference, (c) the DCIP3D results acquired in 2001 over the same profile.

The results of the inversion (Figure 46a) are based on convergence to assigned misfit in 21 iterations (Figure 46b). They were compared with the earlier results acquired by DCIP3D code (Rutley et al., 2001) and are shown in Figure 46c. The large conductor on the right hand side is a black shale unit. The main geologic structure runs north-south and is essentially perpendicular to the survey line. It is expected therefore that the 2D inversion should produce geologically reasonable results. This is substantiated by comparing this cross-section with a similar cross-section extracted from the 3D inversion (Figures 46a and 46c).

The conductivity models recovered from the DC inversion were further used to carry out the IP inversion. The IP inversion was carried out using the same mesh as for the DC inversion and was further compared to the 3D inversion carried out previously (Rutley et al., 2001). Below is the control file used for the IP inversion:

```
OBS LOC_X L13700_COMPLETE.dat   !  IP data
MESH FILE dcinv2d.msh           !  Created mesh from DC inversion
TOPO FILE topo.txt              !  Topography
INVMODE SVD                     !  Use CG
REF_MOD FILE 1e-7               !  Reference model
INIT_MOD VALUE 1e-7             !  Initial model
CHIFACT 1                       !  Data misfit to number of data
STORE_ALL_MODELS TRUE           !  Write out all models
ALPHA VALUE 1E-4 1 1            !  Alpha values
COND dcinv2d.con                !  Conductivity model
NITER 50                        !  Max iterations
```

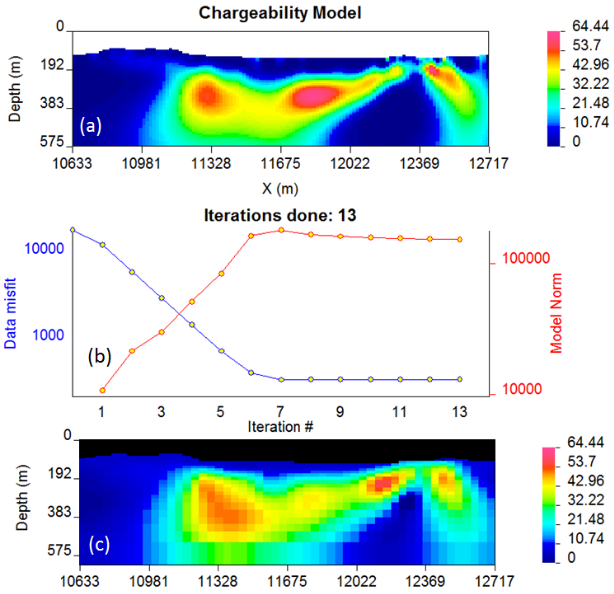


Figure 47: (a) The recovered model from inverting IP data along profile 13700, Cluny prospect, Queensland, Australia and the associated (b) convergence curves. For reference, (c) the DCIP3D results acquired in 2001 over the same profile.

The results of the inversion were once again compared to the corresponding 3D IP inversion (Rutley et al., 2001) and are shown in Figure reffig:realIPres.

The predicted data from the inversions has been verified against the measured data and plotted in Figure 48 and Figure 49. Both inversions (DC and IP) have successfully converged and the misfit does not exceed 5 standard deviations, which is one of the criterions of successful inversions. Another criterion is the verification of the 2D results against the 3D results, which show very comparable results.

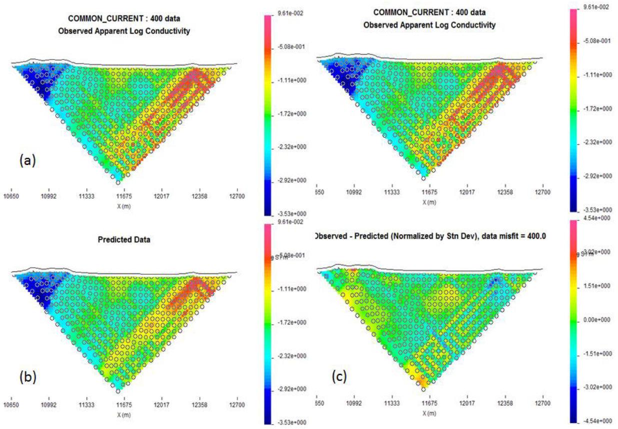


Figure 48: (a) The observed DC data along profile 13700 and (b) the predicted data for comparison. The data misfit normalized by standard deviation is presented in (c).

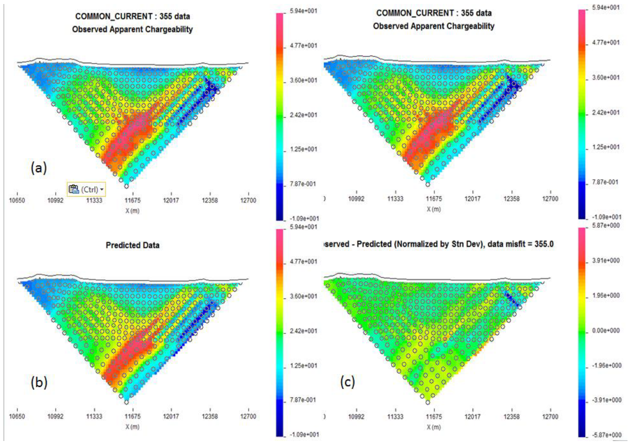


Figure 49: (a) The observed IP data along profile 13700 and (b) the predicted data for comparison. The data misfit normalized by standard deviation is presented in (c).

7 References

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

Ekblom, H., 1973, Calculation of linear best l_p-approximation: BIT, **13**, 292–300.

———, 1987, The l_1-estimate as limiting case of an l_p- or Huber-estimate, *in* Statistical Data Anlysis Based on the L_1-Norm and Related Methods: Elsevier, 109–116.

Farquharson, C. G., and D. W. Oldenburg, 1998, Non-linear inversion using general measures of data misfit and model structure: Geophysical Journal International, **134**, 213–227.

Huber, P. J., 1964, Robust estimation of a location parameter: The Annals of Mathematical Statistics, **35**, 73–101.

Oldenburg, D. W., and Y. Li, 1994, Inversion of induced polarization data: Geophysics, **59**, 1327–1341.

———, 1999, Estimating depth of investigation in dc resistivity and ip surveys: Geophysics, **64**, 403–614.

Rutley, A., D. W. Oldenburg, and R. Shekhtman, 2001, 2-d and 3-d IP/Resistivity inversion or the interpretation of Isa-style targets: Presented at the Expanded abstract, ASEG 15th Geophysical Conference and Exhibition, Brisbane, Australia.

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