Effective multidimensional resistivity inversion using finite-element techniques

Christopher C. Pain, Jörg V. Herwanger, Michael H. Worthington and Cassiano R. E. de Oliveira

Computational Physics and Geophysics Group, Department of Earth Science and Engineering,
Imperial College of Science, Technology and Medicine, Prince Consort Rd, London SW7 2BP, *UK. E-mail: c.pain@ic.ac.uk*

Accepted 2002 June 4. Received 2002 June 4; in original form 2001 May 5

SUMMARY

This paper describes the development of a multidimensional resistivity inversion method that is validated using two- and three-dimensional synthetic pole–pole data. We use a finite-element basis to represent both the electric potentials of each source problem and the conductivities describing the model. A least-squares method is used to solve the inverse problem. Using a least-squares method rather than a lower-order method such as non-linear conjugate gradients, has the advantage that quadratic terms in the functional to be optimized are treated implicitly allowing for a near minimum to be found after a single iteration in problems where quadratic terms dominate. Both the source problem for a potential field and the least-squares problem are solved using (linear) pre-conditioned conjugate gradients. Coupled with the use of parallel domain decomposition solution methods, this provides the numerical tools necessary for efficient inversion of multidimensional problems.

Since the electrical inverse problem is ill-conditioned, special attention is given to the use of model-covariance matrices and data weighting to assist the inversion process to arrive at a physically plausible result. The model-covariance used allows for preferential model regularization in arbitrary directions and the application of spatially varying regularization. We demonstrate, using two previously published synthetic models, two methods of improving model resolution away from sources and receivers. The first method explores the possibilities of using depth-dependent and directionally varying smoothness constraints. The second method preferentially applies additional weights to data known to contain information concerning poorly resolved areas. In the given examples, both methods improve the inversion model and encourage the reconstruction algorithm to create model structure at depth.

Key words: constrained inversion, domain decomposition, electrical conductivity, finite-element methods, four-pole data, model-covariance matrices, pole–pole data.

1 INTRODUCTION

The goal of inversion of data from DC electrical experiments is to reconstruct the conductivity distribution of a region of the Earth. In geophysical applications electrical current is injected into the ground by electrodes located either on the Earth's surface or in boreholes. The resulting electrical field is a function of the conductivity distribution in the Earth and is monitored by measuring voltages between further electrodes. These voltages form the data that are then inverted to calculate a conductivity distribution.

For many years standard electrode configurations, such as Wenner and Schlumberger arrays were successfully used for profiling and sounding experiments. Processing and interpretation of the data consisted of plotting apparent resistivities for profiling experiments and 1-D depth-profiles derived from matching synthetic data from simple models (e.g. three layers of variable thicknesses and resistivities) with field data. It was realized quite early on that using simple assumptions (such as a 1-D Earth) could severely mislead interpretation (Allaud & Martin 1977). Although data from a 2-D or 3-D subsurface can be inverted using a 1-D model, the usefulness of such a model is certainly debatable. In some instances the model consists of spurious anomalies only. Another common model simplification is the neglect of surface topography. For example, surface topography can distort the electric field and result in an electrical anomaly. The exclusion of topography in the processing of data has a detrimental effect on model interpretation (Fox *et al.* 1980; Biswas & Bhattacharya 1998). A finite-element (FE)-based inversion method such as that presented here has the geometrical flexibility to model surface topography.

The introduction of digital computers in the interpretation of geoelectrical data, made it feasible to assimilate data into arbitrary geometry models (Dey & Morrison 1979a,b; Spitzer 1995; Zhao &

Yedlin 1996). It is now also possible to automate the inversion of subsurface models, see, for example, Pelton *et al.* (1978), Smith & Vozoff (1984) and Tripp *et al.* (1984) for 2-D inversion and Park & Van (1991) for an early example of 3-D inversion. These examples demonstrate a steady progression in the level of sophistication in the inversion algorithms along with an increase in the number of parameters used to describe the subsurface. Electrical inversion techniques have progressed from approximate inverse methods using a series of 1-D inversion steps (Li & Oldenburg 1994) to 3-D inversion methods. The latter have been developed using linearized and backprojection methods (Shima 1992), and non-linear optimization techniques such as non-linear conjugate gradient techniques (Ellis & Oldenburg 1994; Zhang *et al.* 1995). Other second-order techniques that have been applied to inversion, involve solving a system of linear equations for changes in material properties, using a Hessian or approximate Hessian (Loke & Barker 1996; LaBreque *et al.* 1996). The success of these methods is manifested by the popularity of the Marquardt–Levenberg algorithm (e.g. Bishop 1995).

Furthermore, approximations to the equations describing the physics of the forward and inverse problem of electrical prospecting are frequently employed to make the inverse problem more manageable on current computers (e.g. Beard *et al.* 1996). The approximations usually allow the conductivities to vary only by a 'small' amount around a background or starting model. This allows a low-contrast approximation to be used, thus speeding up the computation. However, it has also been shown that current computers are sufficiently fast to invert field scale data sets within a reasonable time while solving the full non-linear inversion problem using an accurate forward model (Ellis & Oldenburg 1994; Zhang *et al.* 1995).

Applications of resistivity inversion techniques are found in hydrogeological, hydrological and medical studies. For example, the method has been successfully used to monitor hydraulic experiments on the field scale (Daily *et al.* 1992), laboratory scale (Weller *et al.* 1996a) and on core samples (Binley *et al.* 1996). There is also an abundance of literature on electrical inversion in medical imaging. In medical imaging, the method is usually called electric impedance tomography (EIT) and a recent summary of the state-of-the-art in EIT is given in Cheney *et al.* (1999).

An inversion algorithm intended for geophysical data from a field experiment must be capable of describing all the features of the experiment to be modelled. It is only recently that errors in the inverted model caused by an oversimplification of the modelling have been studied systematically. For example, Sugimoto (1999) studied the effect of neglecting to model the fluid-filled borehole in crosshole tomography and showed that substantially skewed images can result if boreholes are not modelled appropriately.

The inversion method described by this paper uses a flexible and accurate inversion algorithm that is capable of modelling most features of realistic experiments. For example, by using finite elements to discretize the subsurface, the topography of the surveyed region can easily be incorporated into the model. The result is that the distorting effects of an undulated surface are modelled and cannot be a source of error in the inversion. Furthermore, sources and receivers are not restricted to being situated on nodes of the grid. This is essential in cases where electrodes cannot be placed on a regular grid, for example in crosshole applications, where boreholes almost always deviate from a straight line.

In the first section (Section 2) we briefly discuss a field study that provided the starting point for developing the presented inversion methodology and present a subsurface model resulting from the inversion of the field data. We use this model to explain some of the concepts of the inversion methodology put forward in this paper. Specifically, we show the power of using model covariance information for guiding inversion models in the desired direction by solving a 3-D problem for a 2-D subsurface. We explain why instead of using apparent resistivities (the 'standard' data in electrical prospecting) we use resistances and data-dependent weighting as input data for the inversion. It must be noted, however, that it is not the purpose of this paper to present a case study. We use this field data example to show that the proposed method is capable of dealing with life-size surveys and to illustrate some of the novel concepts involved in the proposed inversion method.

After motivating our approach, we state the equations controlling the physics of the experiment in Section 3. The inverse problem of deriving conductivities is posed as an optimization problem with the aid of a functional containing data-misfit and model covariance information (Section 4). The least-squares solution to the optimization problem is described in Section 5. The numerical inversion method is discussed in Section 6 and its parallel implementation is presented in Section 7. To model arbitrarily complicated structures, we need to discretize the subsurface with a fine grid/mesh. For life-sized problems, resulting finite-element meshes can contain up to a million node-points with the result that current computational resources are strained to their limit. In order to make the problem tractable, we use the pre-conditioned conjugate gradient method for the solution of the linear equations resulting from the discretization of the forward problem and harness the power of parallel computing with domain decomposition techniques. We apply the parallel inversion method to a synthetic example proposed in the literature and test the performance of the algorithm on different parallel architectures. The example can still be solved in a reasonable time on a single processor. However, the real advantage of the proposed method lies in its ability to solve massively large problems that cannot be solved on a single processor or in applications where speed is crucial.

In the final section we apply the inversion technique to three different synthetic test models. In testing an inversion method using synthetic data, the quality of the inversion model can be gauged against the known true subsurface model. In the first test we compare the newly developed second-order technique described in this paper with a non-linear conjugate gradient search and demonstrate the superiority of the proposed technique in cross-well geometries. The next two tests use synthetic models proposed in the literature (Zhang *et al.* 1995; Ellis & Oldenburg 1994). We show that the proposed method results in inversion models that are at least comparable in quality to published results. Additionally, we demonstrate the necessity of including model covariance information in the inverse problem. We present a series of tests in which we use both spatially and directionally varying smoothness constraints and test the influence of these constraints on the inversion model. We also demonstrate that using resistances as input data and using an appropriate weighting, results in improved inversion images. The examples illustrate how the techniques described in this paper can be used to help guide inversion of real data sets.

2 DESIGN CRITERIA FOR THE INVERSION METHOD

2.1 Reskajeage field experiment

In 1998 we acquired an electric crosshole data set in the pole–pole geometry at the Reskajeage hydrological testsite in Cornwall, UK.

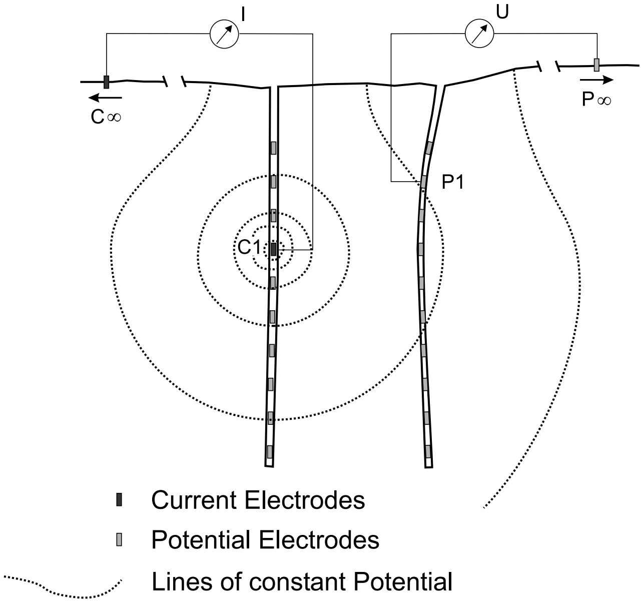

Figure 1. Sketch of electrical crosshole field experiment carried out at Reskajeage Quarry, Cornwall, SW England. The experiment uses a quasi-pole–pole geometry. The well-separation is approximately 25 m and the surveyed depth extends roughly from 20 to 110 m with an electrode spacing of 1 m. The remote electrodes are at 25 m from the respective borehole and need therefore be included in the modelling.

A schematic sketch of the acquisition geometry is given in Fig. 1. This sketch shows a current injection experiment, with the current being injected between one electrode in the left borehole and a remote electrode. The resulting lines of constant potential are depicted by stippled lines. The potential can be monitored at all remaining electrodes by measuring the voltage between the potential electrodes and a remote electrode.

A total of 175 borehole electrodes and two remote electrodes were used in the survey. The electrodes were deployed in two boreholes spaced at approximately 25 m at a 1 m interval over a total depth interval of approximately 80 m, resulting in 30 450 data points. Owing to reciprocity of source and receiver electrodes, all inversion information is contained in half the measurements. Including reciprocal measurements in a measurement scheme allows for the calculation of reciprocal errors, which can be used as one source of information concerning data quality and can be included in a data-covariance matrix.

Owing to access restrictions and safety regulations the remote electrodes were placed 22 m from each borehole. This distance hardly qualifies as infinity, therefore remote electrodes have been included in the forward and inverse modelling. The resulting acquisition geometry can be called a four-pole or quasi-pole–pole geometry.

Fig. 2 shows an inversion model calculated from the Reskajeage field data. The approximate positions of the two wells are indicated by grey lines. We use a 3-D finite-element mesh consisting of 42 840 nodes to parametrize the subsurface. Only a small portion of the mesh, extending 20 m around the boreholes, is shown. The modelling domain extends 150 m beyond the region of interest between the two wells. Application of Dirichlet boundary conditions at this distance is a good approximation to the physically correct boundary conditions of zero potential at infinity.

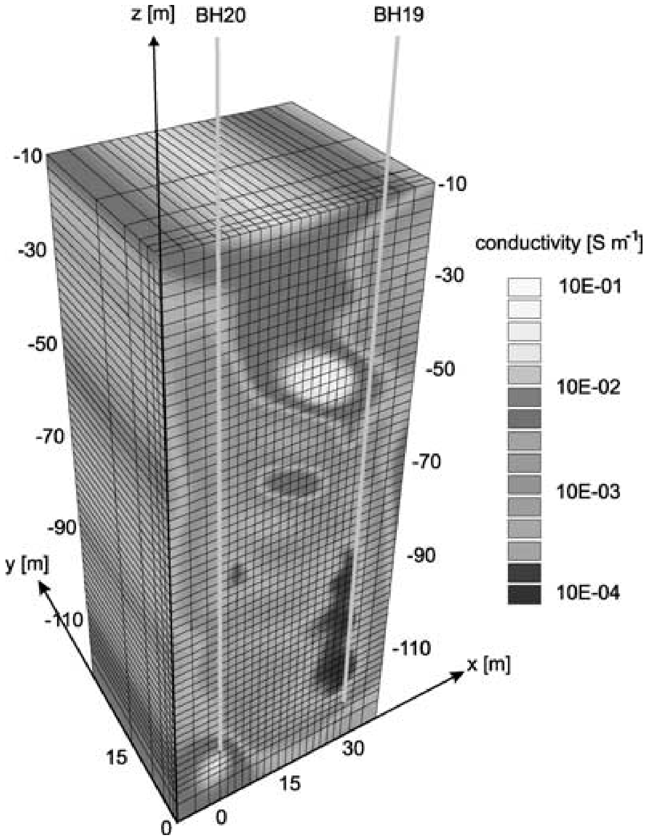

Figure 2. Inversion model of Reskajeage field-data. The part of the inversion model shown extends 10 m beyond the region of interest in *x*- and *z*-directions. The front of the inversion model is the plane of $y = 0$ and contains the two boreholes. Note the exponential mesh-coarsening outside the region of interest. Note further the smoothness of the inversion model in *y*-direction. This is achieved by applying large smoothness constraints in the *y*-direction.

2.2 Advantages of finite-element techniques

Note the exponential coarsening of the mesh moving away from the region of interest. This allows small elements in the region of interest to be used and large elements in the region where conductivities and the potential field are varying slowly. The total number of nodes in the finite-element mesh can thus be reduced substantially compared with a regular grid-spacing as would be employed by finite-difference techniques.

Furthermore, finite-element techniques are ideally suited to dealing with irregular model structure and irregularly shaped interfaces at which boundary conditions need to be applied. For example, the Earth’s surface usually exhibits topography. Using a finite-element description of the subsurface, the effect of an undulating surface are more easily implemented than using other techniques such as finite differences.

2.3 Anisotropic and inhomogeneous smoothness constraints

Since the two wells containing sources and receivers are nearly coplanar (i.e. in the x-z plane), a 2-D subsurface model geometry can be assumed. We achieve this by applying very large smoothness constraints in the *y*-direction and only moderate smoothness constraints in the *x*- and *z*-directions. The effectiveness of this approach is clearly demonstrated by the homogeneity of the inversion model in the *y*-direction (Fig. 2). Note that the solution of the forward problem, that is the electrical potential in the domain, is a true 3-D problem justifying the use of a 3-D mesh. The current approach for this problem is to Fourier-transform the governing equations in the out-of-plane direction and solve the Fourier-transformed equations at discrete wavenumbers (Dey & Morrison 1979a). This approach is computationally more efficient. The main benefit of our approach is the added flexibility of allowing for 3-D structure. For example, the effect of out-of-plane anomalies can be readily assessed.

Anisotropic smoothness constraints can also be used to incorporate *a priori* knowledge into an inversion model. For example, in layered sedimentary environments, the structure of the Earth is smooth within one layer and rough across layer boundaries. If the general orientation of the layering is known, this knowledge can be readily incorporated into anisotropic smoothness constraints.

2.4 Choice of input data

Observed voltages from the Reskajeage data set range from −1 to 3 V for constant injection currents of 100 mA and negative transfer resistances (observed voltage divided by injection current) result. Observed negative voltage or zero-voltage readings can result from the geometrical arrangement of electrodes (Fig. 3). The example shows the potential field of a crosshole experiment for the pole–pole geometry (Fig. 3a) and for a four-pole geometry (Fig. 3b). The Earth's surface is assumed to be far removed and its effect has not been modelled. For pole–pole geometries the observed voltage is always positive (Fig. 3a). However, for the example shown in the four-pole geometry (Fig. 3b) the observed voltage is zero and electrode locations resulting in negative observed voltages are easily identified.

Note that for the geometry shown in Fig. 3(b) the geometry factor *G* (see, for example, Telford *et al.* 1990) also becomes zero. Calculating the apparent resistivity ρ_a,

$$\rho_a = 4\pi \frac{1}{G}\frac{V}{I}, \tag{1}$$

then involves a division of zero (for the voltage) by zero (for the geometry factor) and the calculation of apparent resistivities becomes unstable.

In working with the Reskajeage field data, we experienced the problem of dividing by very small numbers for the geometry factor. Geometry factors for the quasi-pole–pole geometry of the Reskajeage experiment and the geometry factor of the experiment shown in Fig. 3(b) are affected in the same manner by the numerical instability of division of two very small numbers (Herwanger 2001).

Using resistances as input data rather than the commonly used apparent resistivities, the problem of numerical instability in the calculation of apparent resistivities is avoided. On the other hand, observed resistances can vary over orders of magnitude, an undesired characteristic in an inversion problem. In order to avoid the difficulties arising from the unstable calculation of apparent resistivities, but retaining the advantage of scaled data, we introduce a data-magnitude-dependent weight factor. In true pole–pole geometries, this weighting factor can be chosen according to the inverse $1/G$ of the geometry factor *G* and in effect apparent resistivities result as input data. In four-pole geometries and quasi-pole–pole geometries the weighting factor can be chosen in such a way that small data-values receive relatively more weight than large data-values, while introducing an upper limit for the weight.

2.5 Solution techniques

The solution of both the forward and the inverse problem using a finite-element method involves the solution of a large linear system of equations for the number-of-model-parameter unknowns. To achieve this, either direct or iterative solvers can be employed. In the following, the advantages and disadvantages of either method are stated and our choice of iterative solvers is motivated.

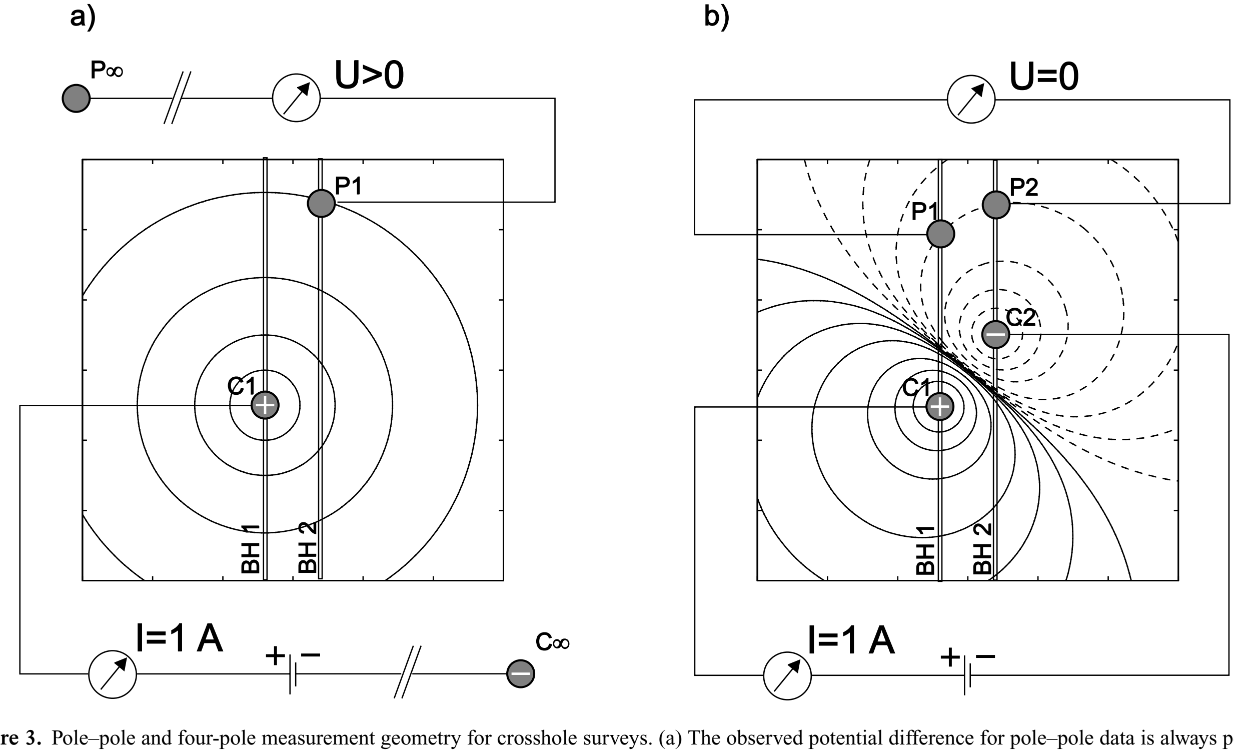

Figure 3. Pole–pole and four-pole measurement geometry for crosshole surveys. (a) The observed potential difference for pole–pole data is always positive. (b) The observed potential difference for the shown four-pole source–receiver geometry is equal to zero. Calculating apparent resistivities would involve a division by zero (for the shown geometry) and the calculation of apparent resistivities becomes unstable.

The advantage of direct solvers is the speed by which multiple source problems can be solved once an inverse matrix has been found. Solving source problems reduces to a matrix–vector multiplication and is thus very fast. Forming an inverse matrix explicitly, necessitates storing all elements of the inverse matrix in memory. Since the inverse matrix loses the sparsity of the matrix describing the forward problem, memory requirements become prohibitive for large-scale 3-D problems.

Iterative solvers are memory efficient. However, their computation time scales linearly with the number of source problems solved. This disadvantage can be somewhat remedied, by using a previous solution of a similar source problem as the starting vector for the iterative procedure.

The scale of problems that can result from modern 3-D surveys using automated acquisition equipment, necessitate grid-sizes of up to a million nodes. For these massive scale models, direct solvers become impractical and we feel that iterative solvers present the only viable way of solving inverse problems on the scale of up to a million nodes.

3 GOVERNING EQUATIONS

The forward problems, describing the electrical potential ψ_s that results from injecting a current $I_s\delta(\Omega - \Omega_s)$ into a region Ω at a location Ω_s with a conductivity distribution σ is given by

$$\nabla \cdot \sigma(\Omega)\nabla\psi_s = I_s\delta(\Omega - \Omega_s). \tag{2}$$

The above equation describes a series of S source problems, each denoted by a subscript $s \in \{1, 2, \dots, S\}$. The source in eq. (2) is a Dirac delta function, centred at position Ω_s of domain Ω. ψ_s is the potential owing to the source number s and $\sigma = \sigma(\Omega)$ is the spatially dependent conductivity ($\sigma = 1/\rho$, where ρ is the resistivity) and I_s is the magnitude of the source strength. In geo-electrical experiments current is usually injected between two electrodes. This experiment can be simulated by the superposition of two solutions of the above equation with two point-sources with positive and negative injection currents, respectively.

3.1 Boundary conditions

Boundary conditions need to be imposed on the entire surface Γ of the solution domain. For geo-electrical problems there is a free-surface boundary condition at the Earth's surface, that is no current is allowed to flow perpendicular across the boundary. The remaining boundaries of the modelling domain result from the necessity of limiting the size of the modelling domain and they are placed at a somewhat arbitrary location. The boundaries should be sufficiently far away from sources and receivers that the effect of these boundaries models the effect of the physical boundary condition of zero-potential at infinity sufficiently well.

The boundary conditions at the free surface Γ_{surf} for each source problem s are Neumann boundary conditions and are described mathematically by the value of the normal derivative of the potential field:

$$\left.\frac{\partial\phi_s}{\partial n}\right|_{\Gamma_{\text{surf}}} = 0, \quad \forall s \in \{1, 2, \dots, S\}, \tag{3}$$

and the Dirichlet boundary conditions at the remainder of the boundary $(\Gamma - \Gamma_{\text{surf}})$ of the domain are given by

$$\phi_s|_{(\Gamma - \Gamma_{\text{surf}})} = 0, \quad \forall s \in \{1, 2, \dots, S\}. \tag{4}$$

The latter Dirichlet boundary conditions in eq. (2) are exact at infinity and thus valid if the domain extends far enough away from the sources for the zero-potential approximation to be good. The approach adopted here is to use large finite elements near the boundaries of the domain to achieve a large modelling domain with a reasonable number of elements. More accurate Robin boundary conditions would allow the domain size to be made smaller (Zhang *et al.* 1995; Weller *et al.* 1996b). However, the gain in speed is not likely to be very big, since we need only a small number of extra elements to increase the modelling domain to an extent that Dirichlet boundary conditions can safely be applied.

3.2 Finite-element approximation

In order to solve eq. (2) numerically using finite elements the source term is subtracted from both sides and the equation is multiplied by finite-element weight functions Q_i and subsequently integrated over the solution domain (Zienkiewicz & Morgan 1983). Additionally, a finite-element approximation ϕ_s replaces the continuous potential distribution ψ_s and a finite-element approximation C_s is used to approximate the source function. The system of equations becomes

$$\int_\Omega Q_i(\nabla \cdot \sigma\nabla\phi_s - C_s)\, d\Omega = 0. \tag{5}$$

The approximation ϕ_s for the potential field ψ_s for a source problem s is described using the finite-element basis functions Q_j (Zienkiewicz & Taylor 1991) via

$$\psi_s \approx \phi_s = \sum_{j=1}^{N} Q_j\phi_{s\,j}, \quad \forall s \in \{1, 2, \dots, S\} \tag{6}$$

and the conductivity $\sigma = 1/\rho$ is similarly discretized by

$$\sigma = \sum_{j=1}^{N} Q_j\sigma_j. \tag{7}$$

In the presented applications we use eight-node hexahedral elements with tri-linear basis functions $Q_j = Q_j(\Omega)$, which are centred on the nodes. The number of basis functions is N, the total number of nodes in the FE mesh.

We use the Galerkin method (Zienkiewicz & Taylor 1991), that is multiplying eq. (2) by a weighting function Q_i that are the same as the basis functions and carrying out the integration over the whole domain. Using the above approximations for ψ_s and σ, the following matrix equation results:

$$\mathbf{A}\Phi_s = \mathbf{b}_s, \quad \forall s \in \{1, 2, \dots, S\}. \tag{8}$$

The matrix and right-hand-side vector elements are $A_{ij} = -\int_\Omega \nabla Q_i \cdot \sigma\nabla Q_j\, d\Omega$ and $b_{s\,i} = \int_\Omega Q_i C_s\, d\Omega$, respectively. The integrals for calculating the elements of the matrix $\mathbf{A}$ and the source vector $\mathbf{b}_s$ are evaluated numerically using Gauss quadrature. Since the finite-element basis functions are local, that is they are only non-zero in a small part of the domain, the discretized set of equations becomes sparse.

Note that a total of S source problems must be solved. Each source problem results in a solution vector for the potentials $\Phi_s = (\phi_{s\,1}\phi_{s\,2}\dots\phi_{s\,N})^{\mathrm{T}}$. By solving the system of eq. (8) a solution for the potential at all points of the FE mesh is obtained. In an experiment, the potential will be observed at discrete points at surface electrodes or borehole electrodes in the domain Ω. The potential at a receiver location is calculated using a Gaussian function to interpolate the potential from the adjacent nodes to the location of the receiver.

3.3 Treatment of sources

If the source-position Ω_s is at a node, then C_s in its simplest form is equal to the finite-element basis function centred on the node multiplied by a scalar $I_s = \int C_s \, d\Omega$ such that the source is scaled to the correct strength. If the source is inside an element, it must be distributed to the nodes associated with that element and possibly nodes associated with neighbouring elements. We have chosen to achieve this with a Gaussian function such that the finite-element approximation

$$C_s = \sum_{j=1}^{N} Q_j C_{s\,j} \tag{9}$$

is defined by the nodal values $C_{s\,i}$ of C_s:

$$C_{s\,i} = \frac{1}{n_s} \exp\left[\frac{-(\mathbf{x}_i - \Omega_s)^2}{l_s^2}\right]. \tag{10}$$

In this equation, l_s is the width of the Gaussian employed and thus controls the contribution to non-nearest node cells, Ω_s and $\mathbf{x}_i$ are the positions of the source s and of node i, respectively, and the normalization coefficient n_s is chosen such that $\int_\Omega C_s \, d\Omega = I_s$. Usually, we choose the width of the Gaussian l_s to be the size of the elements in the vicinity of source s, unless the source is situated on a node. In that case, we choose l_s to be small relative to the mesh spacing. This places the source on to this node. If a small value of l_s is used this will effectively place the source on to the nearest node. The use of l_s provides control over spreading of the sources to the nearest nodes; that is particularly important when the sources are not positioned on the nodes. This is the situation for the problem involving real data solved in Section 2.

4 THE FUNCTIONAL

The functional to be optimized to obtain a solution to the inverse problem is

$$F = F_\mathrm{d} + F_\mathrm{r}, \tag{11}$$

in which F_d is the data misfit between observed data and predicted data and F_r is a regularization term, which can be used to encourage certain types of models. In the following we first discuss the form of the data-misfit functional in the pole–pole geometry and in a general four-pole geometry, allowing for any acquisition geometry. We then discuss the error-functional used for imposing structural constraints and for steplength damping. We designed the functional for applying structural constraints in such a way that we can impose anisotropic and inhomogeneous structural constraints. In Section 8 we show how this new form of error-functional can be used to steer the inversion process in a desired direction. This new error-functional is also the key to forcing a 2-D inversion model in Fig. 2.

4.1 Pole–pole data

The data misfit F_d for pole–pole data is given by the difference between the observed and calculated potentials with

$$F_\mathrm{d} = \frac{1}{2} \sum_{s=1}^{S} \sum_{r=1}^{R} \sum_{i=1}^{N} w_{s\,i}^{r} \left(\phi_{s\,i} - d_{s\,i}^{r}\right)^2. \tag{12}$$

This expression is the weighted sum of the squared differences between observed data (e.g. from a field experiment) and data predicted from a given conductivity distribution for all S sources and R receivers. The datum d_s^r is the potential observed at detector r (position Ω_s^r) for source problem s and $\phi_{s\,i}$ is the potential at node i for source problem s. The total number of sources is S and the total number of receivers is R. The additional loop over all N nodes is caused by the way the receivers are distributed via a Gaussian function.

The weighting factor

$$w_{s\,i}^{r} = \frac{\tilde{w}_s^r}{n_s^r} \exp\left[-\frac{\left(\mathbf{x}_i - \Omega_s^r\right)^2}{l_\mathrm{r}^2}\right] \tag{13}$$

consists of three contributions. In the first two contributions, $\tilde{w}_s^r$ is used to weight the datum associated with each source–receiver pair according to (1) the 'confidence' in this datum and (2) for preferential weighting of this datum. Typically, the confidence contribution to this weight will be one over the estimated standard deviation of this datum. When using $\tilde{w}_s^r$ for preferential weighting, typically large weights are given to small data-values. Since we are using resistances as input data (see Section 2) this can be used to affect a similar result as the use of apparent resistivities achieves. The success of this approach is demonstrated in Section 8.3. The final contribution in the weight factor $w_{s\,i}^{r}$ arises from the way we treat receivers, by distributing them to the nearest nodes via a Gaussian in exactly the same way as sources are distributed (see eqs 9 and 10). For the ith node at position $\mathbf{x}_i$, the rth detector at position Ω_s^r and the sth source problem this contribution is

$$\frac{1}{n_s^r} \exp\left[-\frac{\left(\mathbf{x}_i - \Omega_s^r\right)^2}{l_\mathrm{r}^2}\right]. \tag{14}$$

In this equation, l_r is the length over which interpolation is performed and the normalizing coefficient n_s^r is chosen such that

$$\frac{1}{n_s^r} \int_\Omega \sum_i Q_i \exp\left[-\frac{\left(\mathbf{x}_i - \Omega_s^r\right)^2}{l_s^2}\right] d\Omega = 1. \tag{15}$$

In matrix form, the functional eq. (12) is

$$F_\mathrm{d} = \tfrac{1}{2}(\hat{\Phi} - \mathbf{d})^\mathrm{T} \mathbf{W} (\hat{\Phi} - \mathbf{d}) \tag{16}$$

in which

$$\hat{\Phi}^\mathrm{T} = \left(\Phi_1^\mathrm{T} \Phi_2^\mathrm{T} \cdots \Phi_S^\mathrm{T}, \cdots, \Phi_1^\mathrm{T} \Phi_2^\mathrm{T} \cdots \Phi_S^\mathrm{T}\right), \tag{17}$$

and

$$\mathbf{d}^\mathrm{T} = \left(\mathbf{d}_1^{1\,\mathrm{T}} \mathbf{d}_2^{1\,\mathrm{T}} \cdots \mathbf{d}_S^{1\,\mathrm{T}}, \cdots, \mathbf{d}_1^{R\,\mathrm{T}} \mathbf{d}_2^{R\,\mathrm{T}} \cdots \mathbf{d}_S^{R\,\mathrm{T}}\right). \tag{18}$$

The superscript 'T' is used to indicate the transpose of a vector. Each of the data vectors of predicted data Φ_s and observed data $\mathbf{d}_s^r$ is of the length of the number of nodes. This is caused by the use of a Gaussian to interpolate sources and receivers on to the nodes of the FE mesh. The data covariance matrix $\mathbf{W}^{-1}$ in our work is diagonal and contains the weights $\tilde{w}_s^r$ associated with each datum. The vector $\hat{\Phi} - \mathbf{d}$ represents the residuals (data misfit) at each node for a given source problem and for a given receiver or receiver pair.

4.2 Four-pole data

Pole–pole data cannot be obtained in a physical experiment since the current needs to be injected between two electrodes and potentials can only be monitored by measuring a potential difference between two points. However, one of the current and potential electrodes might be placed at a large distance, assumed to be at infinity for the pole–pole realization, from the electrodes in the domain of interest. In this case the above calculation for the error-functional F_d is a good approximation to reality. If both source and receiver electrodes are placed near the zone of interest, one can speak of four-pole measurements. For experiments in the four-pole configuration, a different functional from the pole–pole eq. (12) needs to be evaluated:

$$F_\mathrm{d} = \frac{1}{2}\sum_{s=1}^{S}\sum_{r=1}^{R}\tilde{w}_s^r\sum_{i=1}^{N}\left[\left(g1_{si}^r - g2_{si}^r\right)\phi_{si} - \mathcal{T}_s^r d_s^r\right]^2. \tag{19}$$

Now $\tilde{w}_s^r$ is the weight associated with the source injection pair s and receiver pair r, $g1_{si}^r$ and $g2_{si}^r$ being Gaussian distributions at node i associated with the potential difference between points 1 and 2, respectively, $\mathcal{T}_s^r$ is the integral under these Gaussians and d_s^r is independent of i and contains the potential difference for the source experiment s and receiver pair r.

4.3 Regularization

Most electrical inverse problems are ill-posed and a standard method to achieve a unique result is to introduce a regularization term F_r into the error-functional to be minimized, constraining the solution to certain classes of models (Constable *et al.* 1987). The usual approach in geo-electrical inversion is to use a smoothness constraint, where the smoothness is locally defined by the square of the gradient or the square of the curvature of the model. We have designed a regularization term that is an extension to this isotropic smoothing. For example, in a layered environment the subsurface is smooth in the layering plane, but rough perpendicular to the layering. To enable us to be able to use this kind of model, we introduce anisotropic smoothness constraints. Additionally, the smoothness constraints are also allowed to vary spatially. If a good starting model can be obtained, it can be advisable to include a penalty for deviation from this starting model into the error-functional.

4.3.1 Anisotropic smoothness constraints

The regularization functional we designed to achieve anisotropic smoothing is given by

$$F_\mathrm{r} = \frac{1}{2}\lambda\int_\Omega \nabla^\mathrm{T} m\boldsymbol{k}\nabla m\, d\Omega. \tag{20}$$

In this equation, λ is a Lagrange multiplier allowing the desired level of regularization to be applied and m are the material properties, in this case $m = \ln(\sigma)$. The gradient ∇m measures the smoothness and the local structure of the model is measured by $\nabla^\mathrm{T} m\boldsymbol{k}\nabla m$. Penalizing structure as a function of direction is achieved by using $\boldsymbol{k}$, a symmetric positive-definite 3×3 matrix. By this definition, $\boldsymbol{k}$ defines a scalar product, making the integrand positive in the whole domain and thus F_r is positive. In the case that $\boldsymbol{k}$ equals a unit-matrix, this form of smoothness constraints reduces to the traditional gradient smoothing. In the general form $\boldsymbol{k}$ can be formed with the use of a rotation matrix $\mathbf{R}$ and a diagonal matrix Λ by $\boldsymbol{k} = \mathbf{R}^\mathrm{T}\Lambda\mathbf{R}$, with the diagonal matrix Λ containing the eigenvalues of $\boldsymbol{k}$. The directions of maximum and minimum smoothing coincide with the directions of the eigenvectors associated with the largest and smallest eigenvalue of the matrix $\boldsymbol{k}$. If inhomogeneous smoothing is desired, the elements of $\boldsymbol{k}$ need to be a function of location Ω in the domain. In one of the synthetic examples (see Fig. 8), we investigate the effect of using inhomogeneous and/or anisotropic regularization.

In matrix notation, using the FE description $m = \sum_j Q_j m_j$, eq. (20) becomes

$$F_\mathrm{r} = \tfrac{1}{2}\lambda\mathbf{m}^\mathrm{T}\mathbf{K}\mathbf{m}. \tag{21}$$

The elements K_{ij} of the constraint matrix $\mathbf{K}$ are given by $K_{ij} = \int_\Omega \nabla Q_i\boldsymbol{k}\nabla Q_j\, d\Omega$, and $\mathbf{K}^{-1}$ is the model covariance matrix. The vector $\mathbf{m}^\mathrm{T} = (m_1, m_2, \dots, m_N) = [\ln(\sigma_1), \ln(\sigma_2), \dots, \ln(\sigma_N)]$ contains the discretized material properties, where σ_j denotes the conductivity centred on node j, while $m_j = \ln(\sigma_i)$ is called the material property in this paper.

Note the use of the logarithm of conductivity m in the definition of the roughness penalty and not the conductivity σ. In the next section, we describe a second-order inversion scheme to invert for these material properties. If we regularize m, the resulting contributions to the error-functional are quadratic. In this case the functional is easily optimized with standard inversion methods (e.g. Bishop 1995). In addition, σ can vary over many orders of magnitude, which would make the direct regularization of σ vary over orders of magnitude. This was considered undesirable—regularization of m does not have this problem.

4.3.2 Penalizing deviation from a starting model

The penalty for an updated model m_new to deviate from a starting model m_old is given by

$$F_\mathrm{r} = \frac{1}{2}\nu\int_\Omega (m_\mathrm{new} - m_\mathrm{old})^2\, d\Omega. \tag{22}$$

Using a finite-element approximation for both the old model $\mathbf{m}_\mathrm{old}$ and an updated model $\mathbf{m}_\mathrm{new}$ this equation becomes:

$$F_\mathrm{r} = \tfrac{1}{2}\nu(\mathbf{m}_\mathrm{new} - \mathbf{m}_\mathrm{old})^\mathrm{T}\mathbf{M}(\mathbf{m}_\mathrm{new} - \mathbf{m}_\mathrm{old}). \tag{23}$$

The mass matrix $\mathbf{M}$ is given by $\mathbf{M}_{ij} = \int_\Omega Q_i Q_j\, d\Omega$ in eq. (27) and $\mathbf{M}$ can be approximated by the row sum lumped mass matrix $\mathbf{M}_\mathbf{L}$. $\mathbf{M}_\mathbf{L}$ is a diagonal matrix defined by its diagonals $\mathbf{M}_{\mathbf{L}ii} = \int_\Omega Q_i\, d\Omega$.

5 LEAST-SQUARES INVERSION

5.1 Inversion method

Linearizing about a starting model and using the abbreviation for the data misfit $\boldsymbol{\epsilon} = \hat{\boldsymbol{\Phi}} - \mathbf{d}$, the data misfit for an updated model can be approximated by

$$\boldsymbol{\epsilon}(m_\mathrm{new}) \approx \boldsymbol{\epsilon}(m_\mathrm{old}) + \mathbf{J}(\mathbf{m}_\mathrm{new} - \mathbf{m}_\mathrm{old}). \tag{24}$$

Here, $\mathbf{m}_\mathrm{old}$ represents the previously available best (in terms of corresponding data misfit) model and $\mathbf{m}_\mathrm{new}$ is an updated model. The notation $\boldsymbol{\epsilon}(m_\mathrm{new})$ and $\boldsymbol{\epsilon}(m_\mathrm{old})$ is used to indicate that the residual-vector $\boldsymbol{\epsilon}$ is a function of the updated model and the old model, respectively. The Jacobian matrix $\mathbf{J}$ is given by

$$\mathbf{J} = \frac{\partial\boldsymbol{\epsilon}(m_\mathrm{old})}{\partial\mathbf{m}_\mathrm{old}}. \tag{25}$$

The elements of the Jacobian consist of the changes in the data misfit $\boldsymbol{\epsilon}(m_\mathrm{old})$ for the source–receiver pair s and r, caused by a perturbation of the model parameters $m_{\mathrm{old}i}, \forall i \in \{1, 2, \dots, N\}$.

Using the approximation for the residuals given in eq. (24), the error-functional F in eq. (11) becomes

$$F = \tfrac{1}{2}[\epsilon(m_{\text{old}}) + \mathbf{J}(\mathbf{m}_{\text{new}} - \mathbf{m}_{\text{old}})]\mathbf{W}[\epsilon(m_{\text{old}}) + \mathbf{J}(\mathbf{m}_{\text{new}} - \mathbf{m}_{\text{old}})] + \tfrac{1}{2}\lambda\mathbf{m}^{\mathrm{T}}_{\text{new}}\mathbf{K}\mathbf{m}_{\text{new}} + \tfrac{1}{2}\nu(\mathbf{m}_{\text{new}} - \mathbf{m}_{\text{old}})\mathbf{M}_{\mathbf{L}}(\mathbf{m}_{\text{new}} - \mathbf{m}_{\text{old}}). \quad (26)$$

Here we have used eq. (16) for the finite-element approximation of the data contribution F_{d} to the error-functional F, and eqs (21) and (23) as a measure for the regularization penalty F_{r}, quantifying the structure penalty and the penalty for deviation from a starting model, respectively.

Differentiating eq. (26) with respect to $\mathbf{m}_{\text{new}}$ and equating to zero ($\partial F/\partial\mathbf{m}$ is zero at an extremum) yields

$$\left(\mathbf{J}^{\mathrm{T}}\mathbf{W}\mathbf{J} + \lambda\mathbf{K} + \nu\mathbf{M}_{\mathbf{L}}\right)\Delta\mathbf{m} = \frac{\partial F}{\partial\mathbf{m}_{\text{old}}} = -\mathbf{J}^{\mathrm{T}}\mathbf{W}^{\mathrm{T}}\epsilon(m_{\text{old}}) - \lambda\mathbf{K}\mathbf{m}_{\text{old}}. \quad (27)$$

This least-squares system for model updates $\Delta\mathbf{m} = \mathbf{m}_{\text{new}} - \mathbf{m}_{\text{old}}$ is usually referred to as Marquardt–Levenberg method. Once model updates have been calculated, the old model is updated. If the inverse problem is non-linear, one can re-linearize about the updated model and calculate a further model update.

Eq. (27) can be abbreviated to:

$$\mathbf{B}\Delta\mathbf{m} = \mathbf{g} \quad (28)$$

with

$$\mathbf{B} = \mathbf{J}^{\mathrm{T}}\mathbf{W}\mathbf{J} + \lambda\mathbf{K} + \nu\mathbf{M}_{\mathbf{L}} \quad (29)$$

and

$$\mathbf{g} = -\mathbf{J}^{\mathrm{T}}\mathbf{W}^{\mathrm{T}}\epsilon(m_{\text{old}}) - \lambda\mathbf{K}\mathbf{m}_{\text{old}}. \quad (30)$$

The term $\nu\mathbf{M}_{\mathbf{L}}$ in eq. (27) controls the magnitude of the model update $\Delta\mathbf{m}$. If ν is chosen to be large, $\Delta\mathbf{m}$ stays small. Therefore, this part of the equation can be called a 'creep' term. In addition, this term will increase the diagonal dominance of the matrix $\mathbf{B}$ in eq. (28) and therefore the conditioning of the matrix $\mathbf{B}$ above.

5.2 Calculations involving the Jacobian

This section discusses calculations involving the Jacobian $\mathbf{J}$ and the employed approximate Hessian $\mathbf{J}^{\mathrm{T}}\mathbf{W}\mathbf{J}$. The method described is computationally efficient as neither the Jacobian $\mathbf{J}$ nor the matrix $\mathbf{J}^{\mathrm{T}}\mathbf{W}\mathbf{J}$ are explicitly formed and stored and only matrix–vector multiplication with these matrices are required.

The Jacobian $\mathbf{J}$ in eq. (27) is given by

$$\mathbf{J} = \mathcal{I}^{\mathrm{T}}\begin{pmatrix}\mathcal{J}_1\\ \mathcal{J}_2\\ \vdots\\ \mathcal{J}_S\end{pmatrix}, \quad (31)$$

where $\mathcal{I}^{\mathrm{T}} = (I_1 I_2 \ldots I_S)$ and the I_s for $s \in [1, 2, \ldots, S]$ are identity matrices of order $N \times N$. Each of the $\mathcal{J}_s$ contains the change in the potential at a node caused by a perturbation of the material property at another node. These sensitivities of the potentials with respect to conductivity perturbations are calculated for every source forward problem s. The multiplication by the matrix $\mathcal{I}^{\mathrm{T}}$ affects a summation of the individual Jacobians $\mathcal{J}_s$ of every source experiment into the Jacobian $\mathbf{J}$ for the entire data set.

The Jacobian $\mathcal{J}_s$ of the sth-source experiment is calculated from

$$\mathcal{J}_s = \mathbf{A}^{-1}\mathbf{L}_s \quad (32)$$

in which the matrix $\mathbf{A}$ describing the forward problem is defined in eq. (8), the matrix $\mathbf{L}_s$ is given by $\mathbf{L}_s = (\mathbf{l}_{s1}\mathbf{l}_{s2}\ldots\mathbf{l}_{sN})$ and the nth column vector $\mathbf{l}_{sn}$ is given by

$$\mathbf{l}_{sn} = \frac{\partial\sigma_n}{\partial m_n}\frac{\partial\mathbf{A}}{\partial\sigma_n}\Phi_s \quad (33)$$

and

$$\left(\frac{\partial\mathbf{A}}{\partial\sigma_n}\right)_{ij} = -\int_{\Omega}\nabla Q_i Q_n \nabla Q_j \, d\Omega. \quad (34)$$

For the partial derivatives of the conductivity σ with respect to the material properties m we have used $\partial\sigma_n/\partial m_n = \exp(m_n)$, which is exact at the nodes. Since the nodal values of σ_n only manifest themselves in $\mathbf{A}$ in a small region in the local vicinity of node n, the matrix $\partial\mathbf{A}/\partial\sigma_n$ has very few non-zeros entries, i.e. of the order of 10s per row when linear finite elements are used. Thus $\mathbf{l}_{sn}$ also has only a small number of non-zeros that makes manipulations involving $\mathbf{l}_{sn}$ computationally efficient. Matrix–vector multiplication involving $\mathbf{L}_s$ or its transpose $\mathbf{L}_s^{\mathrm{T}}$ is demonstrated using the two vectors

$$\mathbf{q} = (q_1 q_2 \ldots q_N)^{\mathrm{T}} \quad (35)$$

and

$$\mathbf{v} = (v_1 v_2 \ldots v_N)^{\mathrm{T}}, \quad (36)$$

each of length N. The matrix–vector multiplications take on the form

$$\mathbf{L}_s\mathbf{q} = q_1\mathbf{l}_{s1} + q_2\mathbf{l}_{s2} + \cdots + q_N\mathbf{l}_{sN} \quad (37)$$

and

$$\mathbf{L}_s^{\mathrm{T}}\mathbf{v} = \begin{pmatrix}\mathbf{l}_{s1}^{\mathrm{T}}\mathbf{v}\\ \mathbf{l}_{s2}^{\mathrm{T}}\mathbf{v}\\ \vdots\\ \mathbf{l}_{sN}^{\mathrm{T}}\mathbf{v}\end{pmatrix}, \quad (38)$$

for multiplication by $\mathbf{L}_s$ and its transpose, respectively.

Multiplications involving $\mathbf{L}_s$ and $\mathbf{L}_s^{\mathrm{T}}$ are required to form $\mathbf{J}^{\mathrm{T}}\mathbf{W}\mathbf{J}\mathbf{q}$. This matrix–vector multiplication for an arbitrary vector $\mathbf{q}$ of length N, becomes

$$\mathbf{J}^{\mathrm{T}}\mathbf{W}\mathbf{J}\mathbf{q} = \sum_{s=1}^{S}\mathbf{L}_s^{\mathrm{T}}\mathbf{A}^{-1}\mathcal{W}_s\mathbf{A}^{-1}\mathbf{L}_s\mathbf{q}, \quad (39)$$

where $\mathcal{W}_s = \sum_{r=1}^{R}\mathbf{W}'_{s\text{r}}$ and the diagonal matrix $\mathbf{W}'_{s\text{r}}$ has diagonal elements $w^r_{\text{d}i}$ (see eqs 12 and 13). Examining eq. (39) reveals that in order to obtain the vector $\mathbf{J}^{\mathrm{T}}\mathbf{W}\mathbf{J}\mathbf{q}$ the solution of two matrix equations involving the matrix $\mathbf{A}$ are required for each source problem. In contrast, if the approximate eq. (26) was not employed and the Hessian was instead used, this would require four matrix equations involving the matrix $\mathbf{A}$ to be solved for each source problem. We are unclear whether using the exact Hessian, instead of this approximation to it, would result in a more efficient algorithm.

6 NUMERICAL SOLUTION OF THE INVERSE PROBLEM

The non-quadratic functional F defined by eq. (11) is minimized by solving a series of least-squares (Marquardt–Levenberg, see eq. 28) problems. After calculating a model update Δm in each iteration, the previous model is updated and the error-functional is evaluated again with the updated model. In the presented algorithm both the magnitude of the step length damping and the regularization can be adjusted between the solution of each least-squares problem.

6.1 Solution method

We use an iterative pre-conditioned conjugate gradient (PCG) solution procedure (see Golub & Van Loan 1996) to solve both the forward problem of calculating potentials for a source problem $\mathbf{A}\Phi_s = \mathbf{b}_s$ (see eq. 8) and the least-squares inverse problem $\mathbf{B}\Delta\mathbf{m} = \mathbf{g}$ (see eq. 28). The following equation exemplifies the procedure using the least-squares inverse problem matrix equation as an example:

for initialization:

$$\begin{aligned} &\mathbf{r}^0 = \mathbf{g} - \mathbf{B}\Delta\mathbf{m} \\ &\mathcal{B}\mathbf{z}^0 = \mathbf{r}^0 \\ &\mathbf{p}^0 = \mathbf{z}^0 \end{aligned} \tag{40}$$

and for each iteration:

$$\begin{aligned} &\alpha^k = \frac{\mathbf{z}^k \cdot \mathbf{r}^k}{\mathbf{p}^k \cdot \mathbf{B}\mathbf{p}^k} \\ &\mathbf{r}^{k+1} = \mathbf{r}^k - \alpha^k \mathbf{B}\mathbf{p}^k \\ &\Delta\mathbf{m}^{k+1} = \Delta\mathbf{m}^k + \alpha^k \mathbf{p}^k \\ &\mathcal{B}\mathbf{z}^{k+1} = \mathbf{r}^{k+1} \\ &\beta^{k+1} = \frac{\mathbf{z}^{k+1} \cdot \mathbf{r}^{k+1}}{\mathbf{z}^k \cdot \mathbf{r}^k} \\ &\mathbf{p}^{k+1} = \mathbf{z}^{k+1} + \beta^{k+1}\mathbf{p}^k. \end{aligned} \tag{41}$$

The vectors $\mathbf{p}^k$, $\mathbf{z}^k$ and $\mathbf{r}^k$ are working vectors used only in the PCG algorithm above. The factors α and β are weighting and acceleration factors, respectively, and the superscript k denotes the iteration number. The pre-conditioner $\mathcal{B}$ is an approximation to the matrix $\mathbf{B}$. We use

$$\mathcal{B} = \lambda\mathbf{K} + \nu\mathbf{M}_L, \tag{42}$$

as a pre-conditioner. Note that the pre-conditioner consists of the matrices containing structural constraints and steplength damping. The matrix $\mathcal{B}$ is non-singular and symmetric positive definite for a positive non-zero ν and thus can be solved efficiently using PCG. Typically, 20 iterations are required to solve a matrix equation involving matrix $\mathbf{B}$ during inversion. If α^n in the PCG algorithm above is greater than 0.01 on the final PCG iteration then we assume that the PCG is 'converging well'. An α close to 1 usually means the pre-conditioner is a close approximation to $\mathbf{B}$. Convergence to a local minimum is assumed when the maximum percentage difference in the conductivity updates from two consecutive iterations is less than a certain tolerance.

6.1.1 Choice of the steplength damping coefficient

The steplength damping coefficient ν controls the convergence of the PCG solution method in two ways. First, ν controls the conditioning (or diagonal dominance) of the matrix $\mathbf{B}$. Without this conditioning, $\mathbf{B}$ can be singular. Secondly, the pre-conditioner $\mathcal{B}$ contains the diagonal matrix $\nu\mathbf{M}_L$, which is also contained in $\mathbf{B}$. Increasing ν thus makes $\mathcal{B}$ a better approximation to $\mathbf{B}$.

An initial value for the steplength damping parameter is supplied and the level of steplength damping ν is adjusted according to the following heuristics: (1) increase ν by a factor of 10, if either the conjugate gradient method above is not converging well or the updated model is worse than the previous model, measured by the functional F in eq. (11); (2) if using a larger steplength damping coefficient does not decrease the value of the functional F, the previous steplength damping coefficient is used; (3) if the conjugate gradient solution of $\mathbf{B}\Delta\mathbf{m} = \mathbf{g}$ is converging well and the value of the new functional is less than the value of the old functional, then ν is reduced by a factor of 10.

6.1.2 Choice of structure penalty level

The structure penalty level λ influences the relative importance of the structure penalty functional to the error-functional. Using smoothness constraints as a structural penalty, large values of λ affect a smooth model and for small values of λ rough models are allowed.

The strategy for choosing the penalty level λ we adopt is to initially use a large penalty level and gradually, at each subsequent Marquardt–Levenberg step, relax the penalty level. This forces large-scale structure to emerge at early iteration stages. For later iteration stages structurally more complex models emerge. The magnitude λ of the regularization is relaxed according to the expression

$$\lambda_l = \gamma\lambda_{l-1}, \tag{43}$$

where l is the iteration level. The relaxation factor γ needs to be chosen to be less than or equal to one.

The inversion process is stopped when the observed data are predicted to a pre-determined level or the model updates become negligibly small. At that point we have found a model that is as smooth as possible and as rough as necessary to fit the data to a sufficient level.

6.2 Practical computational implementation

6.2.1 Speed

To decrease the quantity of CPU usage, the potentials Φ_s for all source problems s are stored. Storing the potentials has the advantage that for each iteration involving a matrix–vector multiplication involving the approximate Hessian $\mathbf{J}^{\mathrm{T}}\mathbf{W}\mathbf{J}$, the potentials do not need to be recalculated. Additionally, storing all potentials speeds up subsequent solutions to a forward problem since a good starting vector for the PCG algorithm is provided. However, if we have a very large number of source problems then it may not be practical to store all the potentials from the source problems ($\Phi_s, \forall s$). In this case we store as many of the potentials (Φ_s) as will fit in main memory and recalculate the other Φ_s as they are needed.

Storing Φ_s also benefits inversion using conjugate gradients with line searches, another optimization technique for minimizing the functional F (see Bishop 1995). We use non-linear conjugate gradients with line searches for comparison with the presented algorithm in the applications section. As a line search is performed, the material properties change little as a minimum is approached. Thus using the previously obtained Φ_s as an initial guess for the iterative solution for Φ_s substantially reduces the number of iterations needed for the PCG solution of the forward problem.

6.2.2 Memory

Assuming that each vector is of length equal to the number of nodes N, the memory requirements of the method are: three vectors for the coordinates ($3N$ real numbers), $12N$ real numbers for the two nested PCG solution methods, $3N$ real numbers for work space, $27N$ real numbers for the regularization matrix $\mathbf{K}$ (see eq. 21; the stencil size for the hexahedral elements used here is 27 points for approximating structure) and $27N$ for the matrix $\mathbf{A}$ (eq. 8) used to solve for potentials. The resulting memory requirements are thus $69N$ real numbers. However, if the potentials are also stored for each source problem then one needs to also store $S \times N$ real numbers. The potentials can be either stored or recalculated as and when they are required in the inversion algorithm, the former is used in the applications presented here. Note that using direct solvers, the memory requirements for storing the (full) inverse matrix would be $N \times N$ and are thus very much larger for large meshes.

The integer requirements of the algorithm are negligibly small compared with the amounts of memory used for the real numbers. In parallel inversion/modelling the nodes and associated variables local to subdomains (see the next section) are stored on each processor, spreading the memory load across the processors.

7 PARALLELIZATION

Parallelization of the presented inversion algorithm is accomplished by subdividing the spatial domain into $\mathcal{P}$ subdomains and attaching a processor to each of the subdomains. Each processor is then responsible for forming and solving the equations internal to the subdomain it is attached to. For efficient parallel assembly of the equations, all the elements surrounding the internal nodes of a subdomain are stored on the processor holding that subdomain. In this approach no communication is necessary to assemble the equations, at the slight expense of storing the elements that contain the halo nodes on more than one processor. Halo nodes are the nodes that lie at the perimeter of the subdomains in the vicinity of the subdomain interfaces. Thus the algorithm experiences a near-optimal speed up of the assembly of the equations, forming the equations as and when they are needed. We illustrate how this procedure works using a source problem for the potential Φ_s with N equations and N unknowns. The parallelization of the inverse problem works in exactly the same way. Recall the discretized form of Laplace equation (see eq. 8):

$$\mathbf{A}\Phi_s = \mathbf{b}_s. \tag{44}$$

This system of equations can be written in a partitioned form as:

$$\begin{pmatrix} \mathbf{A}_{11} & \cdots & \mathbf{A}_{1\mathcal{P}} \\ \vdots & \ddots & \vdots \\ \mathbf{A}_{\mathcal{P}1} & \cdots & \mathbf{A}_{\mathcal{P}\mathcal{P}} \end{pmatrix} \begin{pmatrix} \Phi_{s1} \\ \vdots \\ \Phi_{s\mathcal{P}} \end{pmatrix} = \begin{pmatrix} \mathbf{b}_{s1} \\ \vdots \\ \mathbf{b}_{s\mathcal{P}} \end{pmatrix} \tag{45}$$

and can be solved iteratively via a PCG procedure similar to that above for the inverse problem. Since the $\mathbf{A}_{ii}$ are principal submatrices of a symmetric positive-definite matrix, they are also symmetric positive definite and so

$$\mathcal{A} = \begin{pmatrix} \mathbf{A}_{11} & 0 & \cdots & 0 \\ 0 & \mathbf{A}_{22} & \cdots & 0 \\ \vdots & & \ddots & \vdots \\ 0 & \cdots & 0 & \mathbf{A}_{\mathcal{P}\mathcal{P}} \end{pmatrix} \tag{46}$$

is therefore a viable pre-conditioner. A single domain iteration thus involves the solution of a system of equations of the form

$$\mathcal{A}\hat{\mathbf{z}} = \hat{\mathbf{r}} \tag{47}$$

at the pre-conditioning stage. Eq. (47) contains $\mathcal{P}$ independent submatrices, and equations containing $\mathcal{A}$ can be solved independently on separate processors. Once the pre-conditioning step in eq. (47) has been performed, the values of $\hat{\mathbf{z}}$ at the halo nodes of a subdomain are passed from the processor holding that subdomain to processors holding the neighbouring subdomains. Apart from performing the global vector dot products involving vectors spread (fragmented) across processors (subdomains) and determining the termination within the PCG algorithms, this is the only time communication is needed between the processors.

The solution of the pre-conditioner $\mathcal{A}\hat{\mathbf{z}} = \hat{\mathbf{r}}$ is approximated with m-step parametrized (Adams 1983) forward backward Gauss–Seidel (FBGS) or symmetric successive over relaxation (SSOR) pre-conditioning (Golub & Van Loan 1996). These form an alternative to the more commonly used incomplete Choleski factorization method. For example, two-step FBGS pre-conditioning involves two FBGS iterations starting from a guess of zero and SSOR is similar to FBGS but with a relaxation parameter. For parallel problems typically parametrized five-step FBGS pre-conditioning (Adams 1983) is used within each subdomain to pre-condition the PCG method. This method has the advantage that the quantity of work done in each subdomain can be predicted (in contrast to solving the subdomain problems iteratively allowing them to converge) and therefore the load on each processor can be balanced, for example by making the number of nodes on each processor approximately equal.

A precursor to parallelizing the FEM method is that the domain spanned by the finite-element mesh must be partitioned into subdomains. Ideally, this should be accomplished in a manner that simultaneously balances the load associated with each subdomain and minimizes the communication between the subdomains. Partitioning unstructured meshes, the situation here, is achieved using the recall mechanism of a mean field neural network (Pain *et al.* 1999). To exemplify the domain decomposition method, the five-prism problem (see Section 8.3), discretized with 19 025 nodes has been decomposed into 16 subdomains. Each subdomain contains between 1128 and 1274 nodes and there are 30 677 edges of the finite-element stencil between the subdomains, which reflects the communication cost between the subdomains. The domain decomposition is shown in Fig. 4.

7.1 Performance test of the parallel implementation

Table 1 shows the results of a performance test of the parallel performance of the presented algorithm for one Marquardt–Levenberg iteration on a shared memory computer comprising of four Alpha ev6 833 MHz processors and on a distributed cluster of 16 Alpha ev6 667 MHz processors. We test the performance on both machines as a function of the number of processors and the solver parameters.

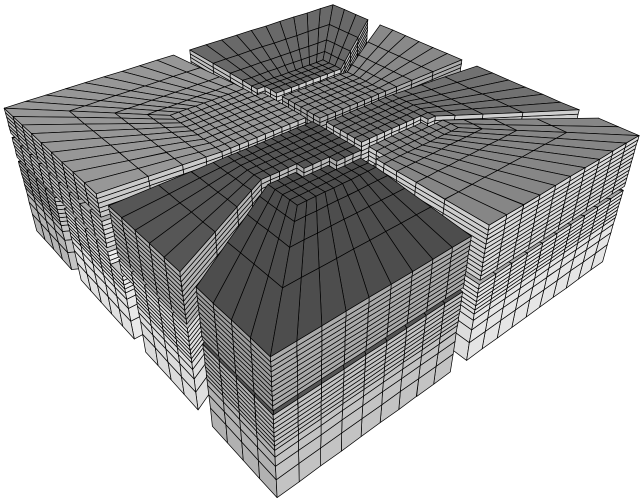

Figure 4. Decomposition of FE mesh for parallel computation. The mesh shown above is the mesh used in the inversion of the five-prism problem.

Table 1. Table showing the parallel performance of the inversion algorithm for a single Marquardt–Levenberg iteration and for the 3-D single block inversion problem. The performance (CPU and wall clock time in seconds) of the algorithm as functions of computer architecture (shared and distributed memory) and number of processors. We also explore the effect of solver options (number of SSOR and FBGS iterations, SSOR with relaxation, FBGS with parametrization) on the time used for one iteration.

	Computer	No of procs	SSOR its	Relax	Parametrization	CPU of proc. 1	Wall clock time
(1)	Shared	1	1	1.0	No	4199	4372
(2)	Shared	2	1	1.0	No	2332	2553
(3)	Shared	4	1	1.0	No	1472	1937
(4)	Shared	1	5	1.0	Yes	5257	5406
(5)	Shared	4	5	1.0	Yes	1396	1753
(6)	Shared	4	5	1.5	No	1266	1608
(7)	Shared	4	1	1.5	No	1572	2096
(8)	Distributed	1	1	1.0	No	4853	4955
(9)	Distributed	4	1	1.0	No	1290	1857
(10)	Distributed	16	1	1.0	No	358	1276
(11)	Distributed	1	1	1.5	No	4855	4956
(12)	Distributed	1	5	1.5	No	4586	4681
(13)	Distributed	16	5	1.0	Yes	450	1270
(14)	Distributed	16	5	1.5	No	431	1283

The different architectures of the two parallel machines, necessitate different solver options. The communication speed of the shared memory computer is superior to that of a distributed memory cluster, which only allows communication between distributed processors via a 100 Mb s^{-1} switch. Thus it is advisable to minimize communication between processor on a distributed memory computer. We achieve this by adjusting solver options (i.e. the number of SSOR iterations and the relaxation parameter), to vary the amount of work before processors need to communicate.

7.1.1 *Speedup with number of processors*

Table 1 lists the wall clock time and the CPU time for processor 1 of the employed $\mathcal{P}$ processors used for one Marquardt–Levenberg iteration. The wall clock time includes the CPU time used for numerical calculations, the time it takes to read and write disk files and the time necessary for communication between processors. Assuming perfect load balancing and neglecting the time used for disk access, the difference between the wall clock time and the listed CPU time equals the time used for communication. Thus, the difference between the wall clock time and the CPU time for processor one shows how the performance of the algorithm is affected owing to communication costs as the fragmentation of the problem increases with the number of subdomains/processors.

We note, that the CPU time of processor 1 is approximately that of the other processors since there is good load balancing with a maximum of 10 per cent difference in the number of nodes on each processor and the cost of each SSOR iteration and matrix–vector multiplication is approximately proportional to the number of nodes. The matrix manipulations dominate the CPU time of the inversion.

Both parallel computers exhibit a nearly linear speed up in terms of CPU time used (tests 1–3 for the shared memory computer and tests 8–10 for the distributed memory computer in Table 1). For example, using the CPU times measured on the distributed memory parallel computer (using one SSOR relaxation step with a relaxation parameter of $w = 1$), the CPU time is reduced by a factor of 3.8 by using four processors instead of one. A further CPU time reduction by a factor of 3.6 is observed when employing 16 processors instead of four.

On the shared memory computer, this speed-up in terms of CPU time used translates into a marked reduction of total computation time, measured by the wall clock-time. For example in tests 3, using four processors on the shared memory computer, the wall clock time equals 1937 s, out of which 1472 s are spent for numerical calculations on the processors, i.e. a total of 76 per cent of the wall clock time is taken up by the CPU. On the distributed parallel computer only 34 per cent of the wall clock time is taken up by CPU (see test 10) and the reduction in total computation time resulting from using 16 processors instead of one is only a factor of 4. This behaviour reflects the large amount of time spent for communication via the slow 100 Mb s^{-1} switch. However, for inversion problems using larger meshes, the communication overhead is smaller than in this test case. In this example, the ratio of the number of halo nodes, involved in communication between processors, to the number of nodes in each subdomain is high. For larger meshes, the ratio between the number of halo nodes and nodes in each subdomain is smaller and relatively more time is spent on each CPU before communicating the results. For example, we have seen 85 per cent CPU usage for a 500 000 node problem. This indicates that distributed memory parallel computers perform well for large-scale problems.

7.1.2 *Choice of solver options*

In this section we discuss the possibility of saving computation time by adjusting solver options. In our computational framework we have two options available that can influence computation time.

First, the number of iterations at the pre-conditioning stage on each processor (using SSOR or FBGS solvers for the solution of the equation involving the pre-conditioner $\mathcal{A}$) can be increased. In so doing, the amount of work done on each processor before the processors communicate is increased and communication costs are reduced. In Table 1 CPU time usage and wall clock time are listed for a series of tests using either one or five SSOR iterations (Adams 1983). Using five SSOR iterations, the time spent on communication between processors (i.e. the difference between wall clock time and CPU time) is reduced for both parallel computers (tests 3, 5, 6 and 7 on the shared memory computer and tests 10, 13 and 14 on the distributed memory computer). The saving in wall clock time on the shared memory computer is of the order of 15 per cent (a reduction in time from 1937 to 1608 s in tests 3 and 6). On the shared memory computer the savings in communication time are offset by the increase in CPU time, and the resulting wall clock time is almost independent of the number of SSOR iterations.

The second solver option to fine-tune the performance of the solver involves adjusting the relaxation parameter w used with SSOR iteration ($w = 1$ or 1.5, see Golub & Van Loan 1996). In addition, FBGS iterations are used with parametrization of the resulting series expansion (see Adams 1983). Timing results from these tests are again listed in Table 1. The changes in both CPU time and wall clock time caused by changes in relaxation w or parametrization (using the FBGS method) are small compared with the total time used and no clear pattern in computation time savings can be discerned.

The achieved speed-up on the shared memory computer as a result of increasing the number of SSOR iterations, the independence of run-time on number of SSOR relaxation on the distributed memory computer and the marginal run-time differences caused by changes in the relaxation parameter w make it difficult to draw general conclusions and we recommend that similar tests be performed for every problem and a given parallel machine. However, if there is a reluctance to perform such tests then five SSOR iteration and $w = 1$ would not be too far from optimal and could be used.

7.1.3 Discussion of the future potential of the parallel method

We have presented results for meshes and problems comparable to those solved in the literature using other methods. However, the method presented here can solve larger problems (in terms of numbers of grid points) than any other method presented so far in the literature on DC geo-electrical inversion, with inversion involving millions of nodes becoming feasible. In addition, the use of unstructured meshes provides the flexibility to use large domains without a large CPU penalization, since large elements can be used in domain regions, where both the material properties and the potential field vary slowly. For example, we use this quality of unstructured finite elements for imposing Dirichlet boundary conditions at a large distance. For the future, we envision remeshing according to sensitivity, placing small elements only in regions where the resolution is high and needed.

We note that this method may not be as fast as other highly optimized methods for regular meshes. However, it is memory efficient and uses efficient scalable iterative solvers. The parallel solver has a reduced communication overhead, in terms of percentage of run time, for larger problems.

We further note that the domain decomposition methods may not be the most efficient method of parallelizing a multisource problem. For these problems, it would be particularly efficient to send each source problem to a separate processor that can then be solved independently. However, for the large parallel problems of the future each of these source problems may need to be parallelized using a domain decomposition solution algorithm such as that described above. Furthermore, for a fast communicating parallel computer parallelization by the domain decomposition method is nearly as efficient as parallelization by subdivision into source problems.

8 APPLICATIONS

In this section we present three synthetic data inversion tests. In a first test (Section 8.1) we use the presented algorithm and compare the resulting inversion model with a model derived using a non-linear conjugate gradient search. This example is used to demonstrate the advantage of treating regularization implicitly. In the following two tests, we use synthetic models from the literature. In the inversion of the test case of one single conductive prism at depth (Section 8.2) we explore the influence of various forms of model regularization. The tests clearly show the benefits of the newly developed regularization functional. In the last example, the test model contains five prism-shaped anomalies and we explore the influence of weighting data points in the error-functional according to their amplitude. We show that the use of resistances and appropriate data weighting has the same performance characteristics as using apparent resistivities as input data. We have shown in Section 2 that the calculation of apparent resistivities can become unstable for unfortunate but entirely reasonable electrode configurations. Using resistances does not suffer from this disadvantage. We conclude that we have found a valuable alternative to using apparent resistivities.

8.1 2-D inversion—a comparison of methods

In this section we compare the performance of the method presented here with that of non-linear conjugate gradients with line searches (Bishop 1995). Using both methods we invert synthetically generated data from a model containing an L-shaped region with a conductivity of 2 S m^{-1}. The remainder of the domain has a conductivity of 1 S m^{-1}. The functional F_{d} given by eq. (12) is used to gauge data miss-fit. As far as possible the two inversions are performed using the same parameters. For both methods a constant regularization coefficient of $\lambda = 0.01$ is applied. The starting model for both inversions is the background conductivity of 1 S m^{-1}.

The extent of the 2-D domain is $x \in [-100, 150]$ and $y \in [-100, 150]$ with an infinite extent in the z-direction. The coordinates and dimensions of the model are in metres (m). The extent of the modelling domain is large enough to make the imposition of zero Dirichlet boundary conditions for the potentials valid. Each element in the inner region defined by $x \in [0, 50]$ and $y \in [0, 50]$ has dimensions of 1 × 1 m^2. A graphical representation of the FE mesh employed is given in Fig. 5. On the left-hand side of the figure, the whole domain is shown and the exponential coarsening can be seen. On the right-hand side, a close-up view of the inner, regular part of the mesh is shown. The L-shaped anomaly is seen in the true model shown in Fig. 6. The coordinates of the corner points of the L-shaped anomaly are $(x, y) =$: (15, 25), (18, 25), (18, 35), (35, 35), (35, 38) and (15, 38).

We use 18 sources and 18 receivers in a pole–pole configuration, roughly simulating a cross-hole geometry. Synthetic data are generated from this model for each source and each receiver (except the one where the source and receiver coincide), resulting in a total of 18 × 17 simulated measurements. Nine electrodes are at a constant x-coordinate of $x = 5$ m, with y-coordinates ranging from 5 to 45 m in increments of 5 m and nine electrodes are at a constant x-coordinate of $x = 45$ m, with y-coordinates ranging from 5 to 45 m in increments of 5 m.

Figs 6(b) and (c) compare the fully converged inversion results for both methods against the true model given in Fig. 6(a). The number of elements and nodes for this problem are 4500 and 4601, respectively, and as with all inversions presented here the same mesh is used to generate the synthetic data and to perform the inversions.

The non-linear conjugate gradient method adjusts the material properties mainly around the sources and receivers, since it is here that the sensitivities (given by the elements of the Jacobian in eq. 25) are largest and the non-linear conjugate method is guided mainly by the gradient. For example, on the first iteration the gradient contains no regularization contribution since the initial model is

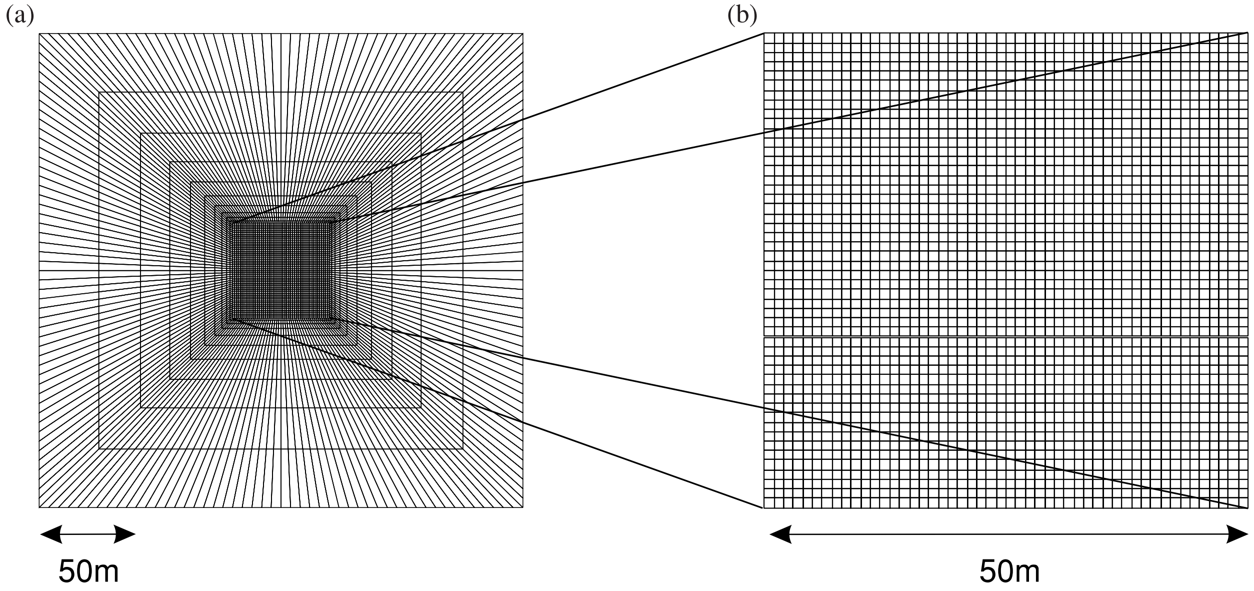

Figure 5. (a) Mesh used in 2-D inversion. Note the exponential coarsening towards the boundaries of the modelling domain. (b) Detailed view of the region of the modelling domain containing structure.

homogeneous. Thus the model after the first non-linear conjugate gradient iteration is not influenced by regularization and the 'damage' caused by a step in a structurally unconstrained direction remains throughout the inversion.

In contrast, the least-squares method described in this paper treats the regularization terms implicitly. For example, if the regularization were chosen to be large, then the resulting model after each iteration would be a uniform field. It is this implicitness that allows this method to modify the material properties in the L-shaped region on the first iteration. Both methods converged at 30 iterations to an approximately equally small data miss-fit F defined by eq. (12).

At this point we have found two models that predict the data equally well and for which the employed inversion algorithm has converged to a local minimum. More iterations do not change the inversion model significantly. This is a demonstration of the non-uniqueness of the solution to the inverse problem. The non-uniqueness of the inversion model is certainly caused by insufficient data: we invert 306 data points (of which only 153 are independent) for 4601 model parameters. This non-uniqueness is entirely expected. However, the presented example shows that inversion methods that treat regularization implicitly, are better suited for finding the class of model requested by the structural constraints.

8.2 Inversion of a 3-D single prism at depth

In this section we describe the inversion for a single conductive prism at depth using synthetically generated data. We use this example to show the effectiveness of using model covariance information to encourage the presented inversion algorithm to preferentially create structure in certain parts of the model.

The model geometry and a summary of the FE mesh employed are given in Table 2. A uniform mesh is used in the region containing the anomaly and the electrodes and an exponentially coarsening mesh is used away from this region. The mesh is shown in Fig. 7(a) and the synthetic 3-D subsurface model with source and receiver locations indicated is shown in Fig. 7(b). A rectangular array of 5×5 electrodes is deployed on the surface of the domain and an additional five electrodes are deployed in a borehole located in the corner of the surface array. We calculate synthetic data in the pole–pole configuration. For a given source problem, potentials at all receivers are observed with the exception of the receiver situated at the source. A vertical slice along the plane $x = 125$ m through the true subsurface model, is depicted in Fig. 8(a) together with the colour bar used for plotting the true model and inversion models. This model was advocated in Zhang *et al.* (1995) to assess their inversion algorithm.

By applying prior knowledge about the smoothness in different parts of the model or in different direction of the model, the inversion process can be guided towards a desired model. We use the functional F_{r} presented in eqs (20) and (21) to impose the desired structural constraints. In Fig. 8 the inversion results using a series of different tensors $\boldsymbol{k}$, resulting in different model covariance matrices $\mathbf{K}^{-1}$, are used and the inversion models are compared.

8.2.1 Unconstrained inversion

In a first test, we invert without applying the smoothness constraints. This corresponds to a covariance matrix filled with all zeros. The resulting inversion model after ten iterations is shown in Fig. 8(b) together with a plot of the error-functional versus number of iterations. Structure in the inversion model is concentrated near the surface where the sensitivities of the error-functional to changes in conductivity are largest. The final inversion model does not resemble the true model, even though the data-misfit is markedly reduced. The error-functional has reached a plateau—more inversion iterations do not reduce the error-functional further. However, the minimum found by this inversion is not a global minimum; the smoothness constrained inversions produce models that predict the observed data with a smaller misfit than the unconstrained inversion.

The unconstrained inversion does not find a good model for the same reasons that the non-linear conjugate gradient method produces a poor quality model. This is because the inversion is largely guided by the gradient $\partial F/\partial \ln(\sigma)$, which is largest near sources

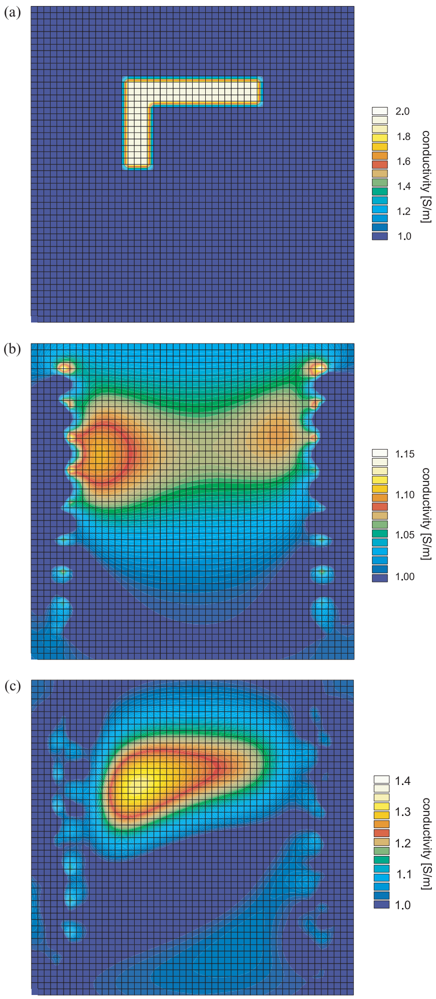

Figure 6. Comparison of different inversion techniques for 2-D inversion. (a) True model. (b) Inversion using non-linear conjugate gradients with line-search. (c) Inversion using a Marquardt–Levenberg-type algorithm.

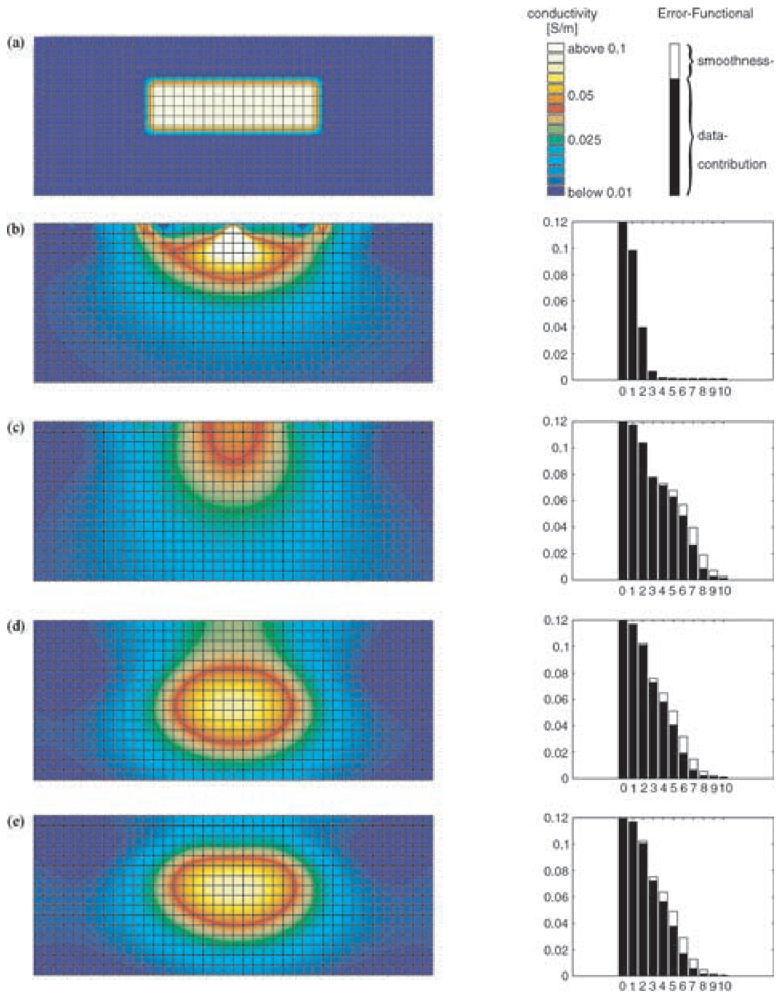

Figure 8. Comparison of inversion results for a one-block model using different smoothness constrains. (a) Inner region of true model with a colour scale used in (a)–(e). (b)–(e) show the inversion result using different strategies of imposing smoothness constrains. The evolution of the error-functional with number of iterations is shown to the right of the corresponding model. See text for detail and Table 3 for a listing of error functional values.

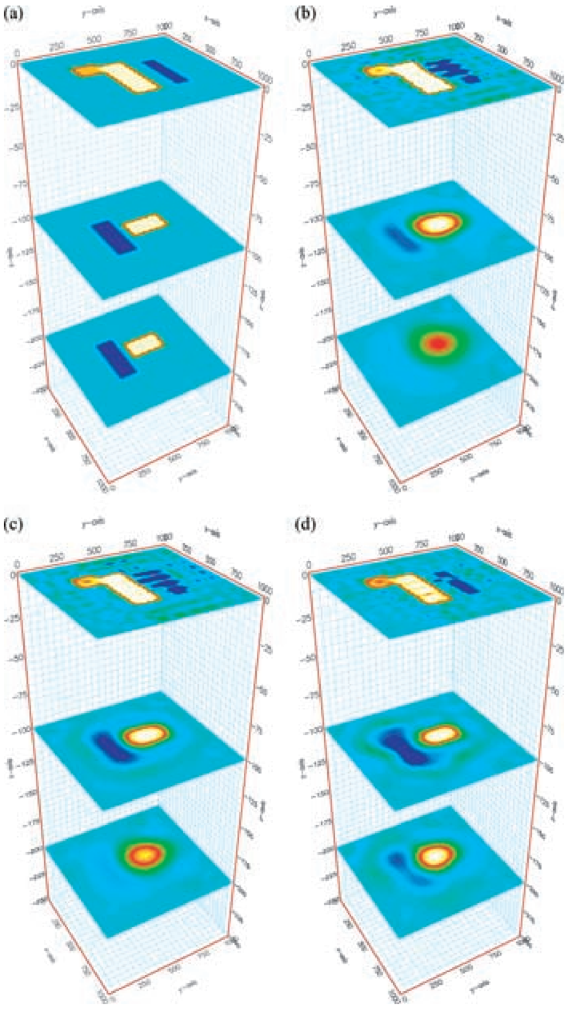

Figure 10. Comparison of inversion results for five-prism model using different data weighting terms in error-functional. (a) True model used to generate synthetic data. (b) Inversion model obtained with uniform weights in the data-covariance matrix. (c) Inversion model derived by choosing weights in the data-covariance matrix according to data-amplitude. This weighting scheme places large weights on small data-values, giving a larger importance to data points from long-offset source–receiver pairs containing information about the deep structure of the model. (d) Inversion model derived using both data-dependent weights in the data-covariance matrix and depth-dependent smoothness constraints in the model covariance matrix.

Table 2. Model description of a one-prism inversion. This model is suggested in Zhang *et al.* (1995). See Fig. 7 for a graphical representation of the FE mesh and the model.

Modelling domain (m)	$x \in [-150, 400]$, $y \in [-150, 400]$, $z \in [-300, 0]$
Inner region (m)	$x \in [0, 250]$, $y \in [0, 250]$, $z \in [-100, 0]$
Dimension of elements	$6.25 \times 6.25 \times 6.25\ \mathrm{m}^3$
Total number of elements	56 320
Total number of nodes	60 743
Background conductivity	$\sigma = 0.01\frac{S}{m}(=100\,\Omega\ \mathrm{m})$
Prism anomaly	$x \in [75, 175]$, $y \in [75, 175]$, $z \in [-56.25, -31.25]$ $\sigma = 0.1\frac{S}{m}(=10\Omega\ \mathrm{m})$
Electrodes	
5×5 surface electrodes	50 m apart in x- and y-directions starting at corner point (25, 25, 0)
Five borehole electrodes	$x = 225$ m, ∀ electrodes $y = 225$ m, ∀ electrodes $z = [-87.5, -68.75, -50.0, -31.25, -12.5]$

and receivers. Therefore, the largest adjustments to the conductivity distribution are made near to the surface. This initially decreases the data-misfit, but prohibits updates of the model at depth in later iterations and the search algorithm is guided into a local minimum (see Table 3 for a listing of data misfit versus the number of least-squares iterations).

8.2.2 Inversion using homogeneous, isotropic constraints

The results shown in Fig. 8(c) are obtained with a homogeneous isotropic regularization based on the squared gradient, as described in Section 4.3. The strength of the regularization is $\lambda = 1.0$ (see eq. 21), which is reduced by a factor of $\gamma = \frac{1}{3}$ from one least-squares inversion iteration to the next (see eq. 43). The inversion procedure starts off with large regularization, resulting in materials that are relatively uniform. As λ is reduced, more structure can emerge in the material properties. The resulting model (see Fig. 8c), is much closer to the true model than the inversion model produced without regularization. However, the inversion model in Fig. 8(c) still fails to reproduce deep conductivity information. This behaviour is caused by the large values of the sensitivities near the surface, where most of the sources and receivers are located, and the rapid decay of sensitivity with increasing depth. This affects large adjustments to model parameters near the surface, where sensitivities are large. At depth, where the sensitivities are small, only minor model changes are made. Again, the inversion process is trapped in a local minimum (see Table 3).

8.2.3 Inversion using depth-dependent, isotropic constraints

One way to assist an inversion scheme in adjusting model parameters in regions of the model space with small sensitivities is to increase sensitivities by some physically motivated empirical rule (e.g. Zhang *et al.* 1995; Li & Oldenburg 1996). Another way to achieve the same goal is to encourage structure to be created by including model covariance information in certain parts of the model space. This is the approach adopted here. It is realized by using spatially varying smoothness constraints. In the presented case, large smoothness constraints are imposed near sources and receivers that are mostly situated at the surface, discouraging structure to be created in areas where sensitivities are large. For increasing depths, the smoothness constraints are relaxed, thus allowing structure to be created.

We use an exponentially decaying function with depth with a fixed lower limit to improve depth-dependent smoothness constraints. The

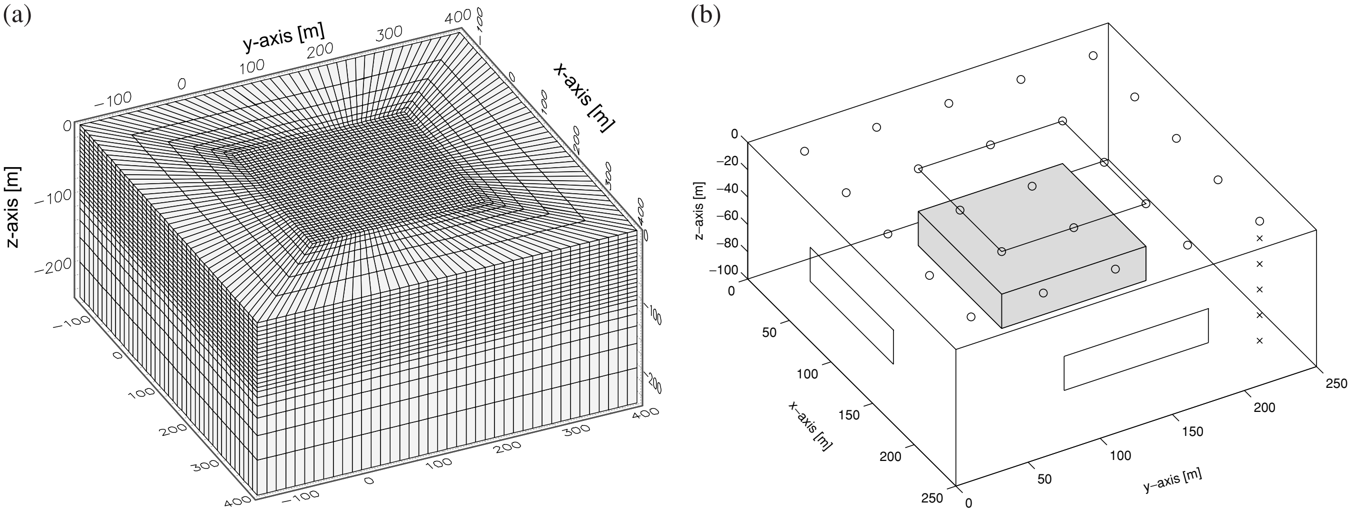

Figure 7. FE mesh and model for inversion of a one-block model. (a) Finite-element mesh. Note the fine mesh in the inner region and the coarsening of the mesh towards the boundary. (b) Model used for inversion tests. Surface electrodes are denoted by circles and borehole electrodes are denoted by crosses. The outlines of the prism-shaped anomaly are projected on to the coordinate planes to facilitate reading of the prism size. This model is proposed in Zhang *et al.* (1995).

Table 3. Error-functional of a one-prism inversion with number of iterations for various smoothing strategies. For each strategy, the value of the error-functional before (F_d) and after (F) imposing the roughness penalty is tabulated. The different strategies used are: (1) no smoothness constraints, (2) homogeneous, isotropic smoothness constraints, (3) depth-dependent, isotropic smoothness constraints and (4) depth-dependent, directionally varying smoothness constraints.

Iteration	Unconstrained inversion		Homogeneous, isotropic constraints		Depth dependent, isotropic constraints		Depth dependent, anisotropic constraints	
	F_d	F	F_d	F	F_d	F	F_d	F
0	0.119	0.119	0.119	0.119	0.119	0.119	0.119	0.119
1	0.982E−01	0.982E−01	0.117	0.117	0.116	0.117	0.116	0.117
2	0.398E−01	0.398E−01	0.103	0.103	0.101	0.102	0.100	0.102
3	0.655E−02	0.655E−02	0.775E−01	0.779E−01	0.729E−02	0.758E−01	0.721E−01	0.754E−01
4	0.188E−02	0.188E−02	0.713E−01	0.730E−01	0.580E−01	0.647E−01	0.564E−01	0.636E−01
5	0.157E−02	0.157E−02	0.630E−01	0.679E−01	0.406E−01	0.512E−01	0.379E−01	0.491E−01
6	0.120E−02	0.120E−02	0.484E−01	0.571E−01	0.191E−01	0.317E−01	0.170E−01	0.291E−01
7	0.112E−02	0.112E−02	0.264E−01	0.396E−01	0.660E−02	0.146E−01	0.565E−02	0.130E−01
8	0.112E−02	0.112E−02	0.869E−02	0.192E−01	0.212E−02	0.550E−02	0.182E−02	0.474E−02
9	0.106E−02	0.106E−02	0.267E−02	0.725E−02	0.117E−02	0.215E−02	0.100E−02	0.182E−02
10	0.106E−02	0.106E−02	0.119E−02	0.269E−02	0.599E−03	0.893E−03	0.499E−03	0.736E−03

isotropic diffusion tensor (see eq. 20) as a function of depth 'z' is defined by its diagonal entries:

$$k_{jj}(z) = (1 - 0.02)\exp(-0.1|z|) + 0.02, \tag{48}$$

and all off-diagonal elements of the tensor are set to zero. The regularization magnitude defined by k_{jj} decays exponentially with depth to a lower limit of 0.02. In using this function for the diagonal tensor elements, structure in the shallow part of the model is penalized more strongly than structure in the lower part of the model. The roughness penalty level λ is again relaxed after each iteration as described previously.

This choice of roughness penalty successfully discourages conductivities from becoming non-uniform near sources and receivers and forces the model conductivities to adjust at depth. The resulting and much improved inversion result is shown in Fig. 8(d). The misfit F_d (see eq. 12) between the data predicted by this model and the observed 'synthetic' data has improved by a factor of 2 compared with the misfit calculated using the unconstrained inversion (see Table 3). Furthermore, the subsurface model derived from inversion with depth-dependent smoothing is a lot closer to the true model than the inversion model using no regularization or homogeneous smoothing. This demonstrates the power and advantages of including model-covariance information into the inversion procedure.

8.2.4 Inversion using depth-dependent, anisotropic constraints

Assuming a horizontally layered subsurface, the model is smooth in horizontal directions and rough in the vertical direction. In other words, we can assume anisotropic smoothness constraints. In practice this is realized by making use of the diffusion tensor $\boldsymbol{k}$ in eq. (20). In this example, we desire a smooth model in x- and y-directions but allow a rough model in the z-direction. Besides anisotropic smoothness constraints, the same depth-dependent smoothness constraints as in the previous example are imposed. The diagonal elements of the diffusion tensor are defined by

$$k_{jj}(z) = \hat{k}_{jj}[(1 - 0.02)\exp(-0.1|z|) + 0.02], \tag{49}$$

with $\hat{k}_{11} = \hat{k}_{22} = 1$ and $\hat{k}_{33} = 0.001$, where the 1, 2 and 3 indices refer to the x-, y- and z-coordinates, respectively. Note that the smoothness constraints in the horizontal x- and y-directions are a factor of 1000 stronger than the constraints in the vertical z-direction. The off-diagonal entries of the diffusion tensor are zero.

The resulting inversion model is shown in Fig. 8(e). The shape of the conductive prism is recovered remarkably well. Note especially the flat top surface with a large conductivity contrast of the recovered conductive structure at exactly the same depth as the top of the prism in the true model. This large change of conductivity in the z-direction is a direct consequence of the anisotropic smoothness constraints, allowing large contrasts in z-direction.

This example clearly demonstrates the need for using appropriate model-covariance information. Without any additional guidance in the form of smoothness constraints, the inversion procedure produces a skewed subsurface model that hardly resembles the true subsurface distribution of conductivities. Additionally, this test demonstrates again how the minimization procedure can be trapped in a local minimum when no model covariance information is used. The use of standard smoothness constraints, that is smoothness constraints that are the same in all regions of the model and that are not directionally varying, results in a model with a slightly improved error-functional (see Table 3), and a slightly improved subsurface model. However, only the use of spatially varying smoothness constraints, 'boosting' sensitivities at depth, enabled a good reconstruction of the true subsurface.

8.3 Five-prism inversion

In this section a further inversion test using synthetic data is presented. We aim to show the influence of the weights $\tilde{w}_s^r$ applied to the individual contributions to the error-functional (see eqs 12 and 13). The observed voltages (and resistances) in geophysical electrical experiments vary over orders of magnitude. It stands to reason that large data values will form a larger contribution to the error-functional than small data values. However, the small data values possibly contain important information. For example in pole–pole surveys, large potentials are encountered when sources and receivers are closely spaced and therefore the model is especially sensitive to the small region between and around the source and receiver. Small potentials at receivers are observed for large source–receiver spacings. These potentials (data) are sensitive to changes in conductivity over a much larger part of the region under investigation. Therefore, it can be desirable to give greater 'importance' to these data. This

Table 4. Model description of five-prism inversion. This model is suggested by Li & Oldenburg (1994). See Fig. 9 for a graphical representation of the FE mesh and the model.

Modelling domain (m)	$x \in [-600, 1600]$, $y \in [-600, 1600]$, $z \in [-800, 0]$
Inner region (m)	$x \in [0, 1000]$, $y \in [0, 1000]$, $z \in [-500, 0]$
Dimension of elements	$50 \times 50 \times 50$ m^3
Total number of elements	17 280
Total number of nodes	19 025
Background conductivity	$\sigma = 0.001\frac{S}{m}(=1000\,\Omega\text{m})$
Prism 1	$x \in [250, 350]$, $y \in [250, 350]$, $z \in [-25, 0]$ $\sigma = 0.005\frac{S}{m}(=200\,\Omega\text{m})$
Prism 2	$x \in [300, 700]$, $y \in [400, 500]$, $z \in [-25, 0]$ $\sigma = 0.01\frac{S}{m}:(=100\,\Omega\text{m})$
Prism 3	$x \in [350, 700]$, $y \in [650, 700]$, $z \in [-25, 0]$ $\sigma = 0.0005\frac{S}{m}(=2000\,\Omega\text{m})$
Prism 4	$x \in [300, 650]$, $y \in [300, 400]$, $z \in [-250, -50]$ $\sigma = 0.0005\frac{S}{m}(=2000\,\Omega\text{m})$
Prism 5	$x \in [400, 500]$, $y \in [500, 700]$, $z \in [-275, -75]$ $\sigma = 0.01\frac{S}{m}(=100\,\Omega\text{m})$
Electrodes	
9 × 9 surface electrodes (m)	100 m apart in x- and y-directions, starting at corner point (100, 100, 0)
4 × 10 borehole electrodes (m)	$(100, 100, z_i)$, $(100, 900, z_i)$, $(900, 100, z_i)$, $(900, 900, z_i)$ with $z_i = [-500, -450, -400, \ldots, -50]$

is achieved by choosing the weight $\tilde{w}_s^r$ proportional to the inverse of the observed data. This has a similar effect as using apparent resistivities as data.

To investigate the influence of the weights $\tilde{w}_s^r$, we use the five-prism model proposed by Li & Oldenburg (1994) and Ellis & Oldenburg (1994). Three near-surface prisms with resistivities of 100, 200 and 2000 Ωm and two prisms at depth with resistivities of 100 and 2000 Ωm are embedded in a homogeneous background with a resistivity of 1000 Ωm. For further details of the model and the survey geometry see Table 4. The FE mesh used for the generation of synthetic pole–pole data and for inversion is shown in Fig. 9(a) and the geometry of the experiment is shown in Fig. 9(b).

In Fig. 10 we compare the true subsurface model with three inversion results. In one of the inversions, all weights are chosen to be $\tilde{w}_s^r = 1$ (Fig. 10b). Note that at the surface in Fig. 10(b) the data has been overfitted resulting in a rough inversion model. The other two inversions are obtained using the weights $\tilde{w}_s^r = 1/(d_s^r)^2$, where d_s^r is the observed data (see eq. 12).

The inversion result using data weighting according to amplitude (Fig. 10c) is closer to the true model (Fig. 10a) than the inversion result calculated using uniform data weights (Fig. 10b). In the lowest horizontal slice at a depth of −200 m, both the conductive prism and the resistive prism are markedly better resolved by the inversion employing the data weighting. Adjusting the weights in the error-functional also adjust the sensitivities, (or the Jacobian). Giving large weights to residuals from data points with a small amplitude will generally emphasize residuals from source–receiver pairs with a long spacing. Sensitivities for these data points show, that they

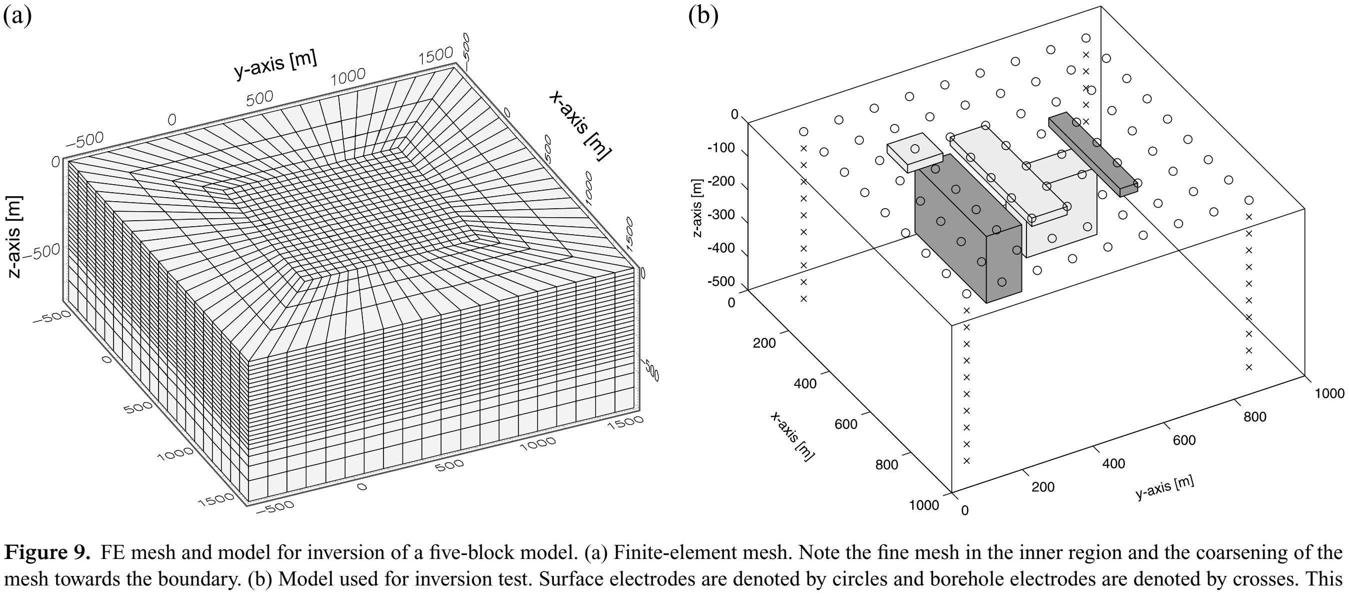

Figure 9. FE mesh and model for inversion of a five-block model. (a) Finite-element mesh. Note the fine mesh in the inner region and the coarsening of the mesh towards the boundary. (b) Model used for inversion test. Surface electrodes are denoted by circles and borehole electrodes are denoted by crosses. This model is proposed by Li & Oldenburg (1994).

are influenced noticeably by conductivities in the bottom part of the model. Thus emphasizing the contributions from these data results in better resolution at depth.

We note, that in the case of pole–pole data, the proposed use of data weights $\tilde{w}_s^r = 1/(d_s^r)^2$ is similar to using apparent resistivities as input data. For homogeneous subsurface models, the two approaches are identical. However, our approach can work robustly without adverse geometrical effects for four-pole problems. We have shown in Section 2 that geometry factors and observed voltages can become very small for four-pole geometries, causing problems in the calculation of apparent resistivities since a division by a very small number is involved. In the case of the proposed data weighting scheme, the weights can be chosen in such a way that data weights for small data are large, while ensuring that the weights cannot grow out of bounds. Here we have satisfied our aim of demonstrating the viability of our approach of using resistance as input data and using data-dependent weighting to account for large differences in the magnitude of input data.

In the previous section, we demonstrated the advantage of using depth-dependent regularization. We again apply depth-dependent smoothness constraints, using a diffusion tensor (see eq. 20) with diagonal elements given by

$$k_{jj}(z) = (1 - 0.02)\exp\left(\frac{-|z|}{25}\right) + 0.02, \quad \forall i \in \{1, 2, 3\}, \tag{50}$$

and zero entries for the off-diagonal entries. This tensor affects isotropic, depth-dependent smoothness constraints. The resulting inversion model shows a close similarity with the true model. The inversion model shows a marked increase in resolution at depth (see Fig. 10d). We have also used anisotropic regularization as used for the previous problem and the inversion result is marginally better than the result presented here, particularly at depth. This seems to suggest that anisotropic constraints may be optimal for inversion with mainly surface electrodes.

9 CONCLUSIONS

We have presented a multidimensional algorithm for solving the forward and inverse problem of geo-electrical experiments. The proposed method has been used to invert two problems proposed in the published literature (see Li & Oldenburg 1994; Ellis & Oldenburg 1994; Zhang *et al.* 1995). The inversion images produced in this work are of similar quality to those given in the literature. Using these examples we have explored in depth: (1) the influence of smoothing operators (by use of the model-covariance matrix) and (2) data weights (by use of the data-covariance matrix) on the inversion results. Both methods can be used to improve model resolution away from sources and receivers. Special attention has been given to the construction of the operator measuring structure (see eq. 20). By allowing directionally varying smoothness constraints at every location in the modelling domain (anisotropic smoothness constraints) and also allowing spatially varying smoothness constraints (inhomogeneous smoothness constraints) a great degree of control can be exerted on the solution of the (ill-posed) inverse problem.

Furthermore, we have demonstrated the importance of treating the regularization terms implicitly by comparing the performance of the advocated least-squares method (Marquardt–Levenberg-type approach) solved with pre-conditioned conjugate gradients, with a non-linear conjugate gradient method. The latter method performed poorly because of the explicit treatment of regularization. In fact, relaxing the magnitude of the regularization, from large to small values during the inversion, is the key to the success of the proposed method.

We believe that 2-D inversion and associated approximations will increasingly be found to be lacking the power to describe real geophysical imaging problems—providing the motivation for this work. The necessary efficiency for these computationally intensive problems has been achieved with appropriate pre-conditioning and adaptive control of the diagonal dominance of the least-squares problem and the implementation of these techniques on parallel computers. The meshing flexibility of the finite-element method and associated representations of electrical conductivity and potential will become increasingly important, for example to capture surface topography and for computational economy in order to place the mesh resolution in regions where it is required.

ACKNOWLEDGMENTS

The authors would like to thank Andrew Binley of Lancaster University, UK for advice provided at a particularly difficult stage in this work, Bo Holm Jacobson of Aarhus University, Denmark for constructive comments on this work, Gerard Gormon for advice on the parallel computations, Adrian Umpleby for providing computational support and the reviewers for greatly improving the content of this paper.

REFERENCES

Adams, L., 1983. An *m*-step pre-conditioned conjugate gradient method for parallel computations, in *IEEE Parallel Computation Conference Proceedings*, pp. 36–43, IEEE, Bellaire, MI.

Allaud, L. & Martin, M., 1977. *Schlumberger, the History of a Technique,* Wiley, New York.

Beard, L.P., Hohmann, G.W. & Tripp, A., 1996. Fast resistivity/IP inversion using a low-contrast approximation, *Geophysics,* **61,** 169–179.

Binley, A., Shaw, B. & Henry-Poulter, S., 1996. Flow pathways in porous media: electrical resistance tomography and dye staining image verification, *Meas. Sci. Technol.,* **7,** 384–390.

Bishop, C.M., 1995. *Neural Networks for Pattern Recognition,* Oxford University Press, Oxford.

Biswas, D. & Bhattacharya, B.B., 1998. 2-D electrical modeling over undulated topography, *Geophysics,* **63,** 898–907.

Cheney, M., Isaacson, D. & Newell, J.C., 1999. Electrical impedance tomography, *SIAM Rev.,* **41,** 85–101.

Constable, S.C., Parker, R.L. & Constable, C.G., 1987. Occam's inversion: a practical algorithm for generating smooth models from electromagnetic data, *Geophysics,* **52,** 289–300.

Daily, W., Ramirez, A., LaBreque, D. & Nitao, J., 1992. Electrical resistivity tomography of vadose water movement, *Water Resources Res.,* **28,** 1429–1442.

Dey, A. & Morrison, H.F., 1979a. Resistivity modeling for arbitrarily shaped two-dimensional structures, *Geophys. Prospect.,* **27,** 106–136.

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

Ellis, R.G. & Oldenburg, D.W., 1994. The pole–pole 3-D DC-resistivity inverse problem: a conjugate-gradient approach, *Geophys. J. Int.,* **119,** 187–194.

Fox, R.C., Hohmann, G.W., J, K.T. & Rijo, L., 1980. Topographic effects in resistivity and induced-polarization surveys, *Geophysics,* **45,** 75–93.

Golub, G. & Van Loan, C., 1996. *Matrix Computations,* 3rd edn, Johns Hopkins University Press, London.

Herwanger, J., 2001. Seismic and electric crosshole tomography for fracture detection and characterization, *PhD thesis,* Imperial College of Science, Technology and Medicine, London.

LaBreque, D.J., Miletto, M., Daily, W., Ramirez, A. & Owen, E., 1996.

The effects of noise on Occam's inversion of resistivity tomography data, *Geophysics,* **61,** 538–548.

Li, Y. & Oldenburg, D.W., 1994. Inversion of 3-d DC resistivity data using approximate inverse mapping, *Geophys. J. Int.,* **116,** 527–537.

Li, Y. & Oldenburg, D.W., 1996. 3-d inversion of magnetic data, *Geophysics,* **61,** 394–408.

Loke, M. & Barker, R., 1996. Practical techniques for 3-D resistivity surveys and data inversion, *Geophys. Prospect.,* **44,** 499–523.

Pain, C.C., de Oliveira, C.R.E. & Goddard, A.J.H., 1999. A neural network graph partitioning procedure for grid-based domain decomposition, *Int. J. Numer. Methods Eng.,* **44,** 593–613.

Park, S.K. & Van, G.P., 1991. Inversion of pole–pole data for 3-D resistivity structure beneath arrays of electrodes, *Geophysics,* **56,** 951–960.

Pelton, W.H., Rijo, L. & Swift, C.M., 1978. Inversion of two-dimensional resistivity and induced-polarization data, *Geophysics,* **43,** 788–803.

Shima, H., 1992. 2-D and 3-D resistivity image reconstruction using crosshole data, *Geophysics,* **57,** 1270–1281.

Smith, N. & Vozoff, K., 1984. Two-dimensional DC resistivity inversion for dipole–dipole data, *IEEE Trans. Geosci. Remote Sensing,* **GE-22,** 21–28.

Spitzer, K., 1995. A 3-D finite-difference algorithm for DC resistivity modelling using conjugate gradient methods, *Geophys. J. Int.,* **123,** 903–914.

Sugimoto, Y., 1999. Shallow high-resolution 2-D and 3-D electrical crosshole imaging, *Leading Edge,* **12,** 1425–1428.

Telford, W., Geldart, L. & Sheriff, R., 1990. *Applied Geophysics,* 2nd edn, Cambridge University Press, Cambridge.

Tripp, A.C., Hohmann, G.E. & Swift, C., 1984. Two-dimensional resistivity inversion, *Geophysics,* **49,** 1708–1717.

Weller, A., Gruhne, M., Seichter, M. & Börner, F.D., 1996a. Monitoring hydraulic experiments by complex conductivity tomography, *Eur. J. Environ. Eng. Geophys.,* **1,** 209–228.

Weller, A., Seichter, M. & Kampke, A., 1996b. Induced-polarization modelling using complex electrical conductivities, *Geophys. J. Int.* **127,** 387–398.

Zhang, J., Mackie, R.L. & Madden, T.R., 1995. 3-D resistivity forward modelling and inversion using conjugate gradients, *Geophysics,* **60,** 1313–1325.

Zhao, S. & Yedlin, M.J., 1996. Multidomain Chebyshev spectral method for 3-D DC resistivity modeling, *Geophysics,* **61,** 1616–1623.

Zienkiewicz, O. & Taylor, R., 1991. *Finite Element Method: Solid and Fluid Mechanics Dynamics and Non-Linearity,* 4th edn, Vol. 2, McGraw-Hill, New York.

Zienkiewicz, O.C. & Morgan, K., 1983. *Finite Elements and Approximation,* Wiley, New York.