Geophysical Journal International

Geophys. J. Int. (2018) **212,** 2073–2087
Advance Access publication 2017 December 4
GJI Marine geosciences and applied geophysics

doi: 10.1093/gji/ggx511

Electrical Resistivity Tomography using a finite element based BFGS algorithm with algebraic multigrid preconditioning

A.L. Codd and L. Gross

School of Earth and Environmental Sciences, The University of Queensland, St Lucia, QLD 4072*, Australia. E-mail: a.codd@uq.edu.au*

Accepted 2017 November 26. Received 2017 September 12; in original form 2017 April 17

SUMMARY

We present a new inversion method for Electrical Resistivity Tomography which, in contrast to established approaches, minimizes the cost function prior to finite element discretization for the unknown electric conductivity and electric potential. Minimization is performed with the Broyden–Fletcher–Goldfarb–Shanno method (BFGS) in an appropriate function space. BFGS is self-preconditioning and avoids construction of the dense Hessian which is the major obstacle to solving large 3-D problems using parallel computers. In addition to the forward problem predicting the measurement from the injected current, the so-called adjoint problem also needs to be solved. For this problem a virtual current is injected through the measurement electrodes and an adjoint electric potential is obtained. The magnitude of the injected virtual current is equal to the misfit at the measurement electrodes. This new approach has the advantage that the solution process of the optimization problem remains independent to the meshes used for discretization and allows for mesh adaptation during inversion. Computation time is reduced by using superposition of pole loads for the forward and adjoint problems. A smoothed aggregation algebraic multigrid (AMG) preconditioned conjugate gradient is applied to construct the potentials for a given electric conductivity estimate and for constructing a first level BFGS preconditioner. Through the additional reuse of AMG operators and coarse grid solvers inversion time for large 3-D problems can be reduced further. We apply our new inversion method to synthetic survey data created by the resistivity profile representing the characteristics of subsurface fluid injection. We further test it on data obtained from a 2-D surface electrode survey on Heron Island, a small tropical island off the east coast of central Queensland, Australia.

Key words: Electrical resistivity Tomography (ERT); Inverse theory; Numerical modelling; Numerical solutions.

1 INTRODUCTION

Electrical resistivity surveys have long been used to estimate electric conductivity (or resistivity) profiles of a subsurface region. In these experiments, electrodes are placed on the ground surface or in boreholes and current load is applied to one (pole load) or two electrodes (dipole load). Electric potentials are recorded at some or all of the non-loading electrodes. Inversion methods convert measured potential data from many different applied loads into subsurface maps of conductivity. Traditional resistivity methods use standard loading and measuring arrays (e.g. Schlumberger, Wenner or dipole–dipole (Telford 1990)) with only a limited number of load dipoles and data measurements used to estimate apparent resistivity values based on the geometry of loading and measuring electrodes. Modern resistivity imaging devices have large numbers of electrodes and data collection is automated with measurements recorded at all non-loading electrodes. Data can be collected for every possible dipole or pole loading configuration at all measuring electrodes.

For a known conductivity profile, electric potential in the subsurface is modelled using a partial differential equation (PDE; Daily *et al.* 2012); in the following referred to as the forward model. Data misfit is defined as a measure of the defects between potentials computed from this forward model with an assumed conductivity and measured potentials. The misfit can also be defined using apparent resistivity values (or their logarithmic value), which can be seen as potential value misfit data scaled with some geometric factors. Inversion methods minimize a measure of the defects to compute possible conductivity distributions. The information obtained may be used further to guess at underground behaviour including water table location or movement (LaBrecque *et al.* 2004), location of anomalies (Rücker *et al.* 2006a) or influx of salt water (Pidlisecky *et al.* 2016).

The computed conductivity distribution and the speed with which it is computed are impacted by a number of design choices. Initial decisions are made with the definition of the domain (2.5-D (Zhou *et al.* 2009) or 3-D (Rücker *et al.* 2006a)), boundary

condition (Dirichlet, Neumann or mixed (Dey & Morrison 1979b; Li & Spitzer 2002)) and conductivity characteristics (isotropic or anisotropic (Zhou *et al.* 2009)) used to model electric potential in the subsurface. For the algorithm outlined here, we are using a large 3-D domain, Dirichlet and Neumann boundary conditions and isotropic conductivity.

A cost function is defined that has a sum of data misfit contributions and an appropriately scaled regularization term to ensure a unique solution. Since we are not restricted to the traditional loading or measuring arrays, scaling the data misfit terms relative to each other is important (Loke *et al.* 2010, 2015). Regularization can take many different forms and depends on the additional assumption and expectation of the conductivity profile (Zhdanov 2002). We simply choose the H^1 norm of the log of the conductivity. The smoothness of the solution is then dependent on the scaling between the regularization and misfit contributions to the cost function.

The next step is to decide the order of discretization and minimization. The forward PDE can be discretized first so that misfit becomes a function of discrete values representing the electric conductivity at cells (Pidlisecky *et al.* 2007). Minimization, using an inexact Gauss–Newton scheme, requires computation of the gradient of the discrete cost function as well as an approximation to its Hessian, both a function of the large dense sensitivity matrix, the partial derivative of the defects with respect to changes in value of electric conductivity at each cell. To compute each Newton step, either the Hessian needs to be inverted or an iterative solver used.

In this paper we follow the approach used by Gross *et al.* (2015) for inversion of potential field data. We choose to minimize the cost function first, using a function space version of the Broyden–Fletcher–Goldfarb–Shanno (BFGS) method (Nocedal & Wright (2006)), prior to discretization. For our case, the unknown, conductivity, is a part of the differential operator not the right hand side of the PDE and the measurements taken are potential measurements not derivatives of potential measurements. An adjoint problem is defined for each forward problem with virtual currents applied at each measuring electrode proportional to the forward problem defects. We avoid computation of a sensitivity matrix by using the product of the forward and adjoint potential gradients in the calculation of the cost function gradient. After each minimization step is taken, the resulting equation is discretized using finite elements, see Zienkiewicz *et al.* (2013). There are two major advantages of this approach: the optimization process is independent to the discretization mesh, potentially allowing adaptive mesh refinement during inversion, and adding more misfit measurements to the arrays does not increase the number of PDEs to be solved. Consequently, using a comprehensive measurement array with a large number of measuring and loading dipoles does not increase the computation time from simply using a Wenner or Schlumberger array with minimal number of loading and measuring dipoles.

Solving the discretized equations, whether derived prior to minimization or after, dominates computation time. Both the finite difference method (Dey & Morrison 1979a; Pidlisecky *et al.* 2007) and finite element method (Li & Spitzer 2002; Zhou *et al.* 2009; Rucker *et al.* 2011) have been used. Existing finite element method parallel algorithms (or other discretization methods) that have already been optimized can be directly applied as solvers for the inversion problem. Our method is implemented using *esys–escript* (Schaa *et al.* 2016). To solve the matrix equations it uses the TRILINOS C++ library (Heroux *et al.* 2005) with *MueLu* (Prokopenko *et al.* 2014) algebraic multigrid (AMG) implementation. For each iteration of the property function, the time intensive part of AMG (defining coarse grids and their interpolation and restriction operators and setting up the coarsest grid direct solver) need only be done once for each conductivity profile, consequently, there are significant time savings over standard preconditioned conjugate gradient (PCG) as demonstrated later in the paper. While versions of multigrid (MG) have been used to solve the forward problem, including conductivity coarsening used by (Moucha & Bailey 2004) for a 2-D problem and (Pan & Tang 2014) used a cascading MG algorithm with no coarse grid correction), AMG is more versatile than both these approaches.

Singularity removal, where the potential is split into two components, a primary homogeneous conductivity part and a variable secondary potential part, is a feature of many inversion algorithms (Lowry *et al.* 1989; Dey & Morrison 1979a; Li & Spitzer 2002; Pidlisecky *et al.* 2007). It is essential that the homogeneous conductivity used in the primary potential computations, for a particular electrode, is the value at that electrode. Otherwise, the singularity is still a part of the secondary PDE. Conductivity at electrodes is unknown, varies between electrodes and will not remain constant throughout the inversion. Also, unless the earth is flat it will need to be solved numerically. For these reasons we have not included it in our approach but use mesh refinement near the electrodes instead.

The forward problem is discussed in Section 2.1. Cost function terms, data misfit and regularization, are discussed in Sections 2.2 and 2.3, respectively. Section 2.4 contains the Gateaux derivative of the cost function and details of the adjoint equations. Implementation of the iterative method BFGS is explained in Section 3. A brief description of AMG PCG is in Section 4. Results from a structured mesh synthetic test case are in Section 5.1. A reduction in the total number of forward and adjoint solves can be made by taking advantage of the superposition principle. Each dipole is the superposition of two poles (Rücker *et al.* 2006a). We only need to solve pole loads for all electrodes with dipole potentials computed by adding scaled pole potentials. This leads to significant time advantages especially if electrodes always act as either source or recorder for each current load. (Details of the simplifications can be found in Appendix A.) We take advantage of this feature of the PDE in Section 5.2 where we use our inversion method on data obtained from Heron Island using an unstructured grid with refinement around the electrodes. Section 6 contains some concluding remarks and future plans.

2 PROBLEM FORMULATION

Consider a bounded domain $\Omega \subset \mathbb{R}^3$, with boundary $\partial\Omega$ that includes air/ground interface, $\Gamma_s \subset \partial\Omega$. Our objective is to reconstruct the electric conductivity profile, σ, on domain Ω, given potential measurements recorded at a number of locations for different applied electric current density loads. To find σ, we minimize a cost function F, defined for all admissible conductivity profiles, that includes both a term for data misfit between measured and computed potentials F_d, as well as a regularization term F_R,

$$F = F_d + \mu F_R, \tag{1}$$

where μ is an appropriately chosen Tikhonov regularization factor. The regularization term is included to ensure a unique solution (Tikhonov & Arsenin 1977). Functions F_d and F_R are discussed in more detail later.

In order to maintain appropriate characteristics of the conductivity, like positivity, a property function m, a parametrization of

$\sigma = \sigma(m)$ is introduced. The minimum of cost function F in equation (1) is characterized by vanishing Gateaux derivative

$$\left\langle F'(m), \delta m \right\rangle = \left. \frac{\partial F(m + \xi \cdot \delta m)}{\partial \xi} \right|_{\xi=0} = 0, \tag{2}$$

for $\xi \in \mathbb{R}$ and all admissible increments δm of the unknown minimizing property function m.

In contrast to conventional Electrical Resistivity Tomography (ERT) inversion approaches, for instance (Rücker *et al.* 2006b), we do not choose to minimize for a discretized version of property function m. Following the approach by Gross *et al.* (2015) we leave the discretization to a later stage in the inversion approach. We minimize first, solving the resulting problem using the BFGS algorithm formulated for a continuous function space, not the usual discrete Euclidean space formulation as developed in Nocedal & Wright (2006). After discretization using the finite element method (FEM), we solve the resulting matrix equations with the AMG method (Briggs *et al.* 2000; Vaněk *et al.* 1996). In the following we present more details of this approach.

2.1 Forward problem

For a given conductivity, σ (or property function m), the forward problem models for electric potentials due to various electric loading experiments need to be solved in Ω. For each applied load represented by the current density source term $q^{(i)}$, total electric potential, $\phi^{(i)}$, is the solution of PDE

$$-\nabla \cdot (\sigma \nabla \phi^{(i)}) = q^{(i)}, \qquad \text{in } \Omega. \tag{3}$$

Symbols ∇ and $\nabla \cdot$ refer to the gradient and divergence operators respectively and index i is used to differentiate between different applied load cases. For a dipole with applied electric current $I^{(i)}$, positive electrode at $\mathbf{x}_{a^{(i)}}$ and negative electrode at $\mathbf{x}_{b^{(i)}}$, load is applied with the 3-D form of Dirac's delta function δ (Dey & Morrison 1979a),

$$q^{(i)} = I^{(i)} \left(\delta(\mathbf{x} - \mathbf{x}_{a^{(i)}}) - \delta(\mathbf{x} - \mathbf{x}_{b^{(i)}})\right). \tag{4}$$

To complete the problem we set boundary conditions on $\partial\Omega$. There is no flux across the air/ground interface,

$$\sigma \mathbf{n} \cdot \nabla \phi^{(i)} = 0, \qquad \text{on } \Gamma_s, \tag{5}$$

with outer normal field $\mathbf{n}$. There are several possibilities for the boundary conditions on the remaining boundaries, see Rücker *et al.* (2006a). In order to simplify the discussion we use Dirichlet boundary conditions of the form

$$\phi^{(i)} = 0, \qquad \text{on } \partial\Omega \backslash \Gamma_s. \tag{6}$$

This condition is justifiable if the domain is sufficiently large so that boundary conditions have no significant impact on the small region where σ is to be recovered.

The weak form of (3) (Brenner & Scott 1994), with boundary conditions (5) and (6) and dipole source given by (4) is

$$\begin{aligned} \int_\Omega \sigma \nabla \psi \cdot \nabla \phi^{(i)} \mathrm{d}x &= I^{(i)} \int_\Omega \left(\delta(\mathbf{x} - \mathbf{x}_{a^{(i)}}) - \delta(\mathbf{x} - \mathbf{x}_{b^{(i)}})\right) \psi \, \mathrm{d}x \\ &= I^{(i)} \left(\psi(\mathbf{x}_{a^{(i)}}) - \psi(\mathbf{x}_{b^{(i)}})\right), \end{aligned} \tag{7}$$

for all admissible potential functions ψ. The solution space for $\phi^{(i)}$ consists of all sufficiently smooth functions, ψ, for which equation (7) is well defined and boundary conditions (6) are satisfied. Note that in (7) the first gradient operator on the left above only acts on ψ as there are no brackets.

2.2 Data misfit

To quantify the defect of predicted potentials $\phi^{(i)}$ at recorder locations $\{\mathbf{x}_{r^{(i)}}\}$ for a given electric conductivity σ, we introduce a misfit function using the Euclidean norm

$$F_d = \frac{1}{2} \sum_i \sum_r w_r^{(i)} \left(\phi^{(i)}(\mathbf{x}_r^{(i)}) - V_r^{(i)}\right)^2, \tag{8}$$

where $V_r^{(i)}$ is the measured potential at recorder $\mathbf{x}_r^{(i)}$ from source configuration $q^{(i)}$. Summation is taken over all measurement points $\mathbf{x}_r^{(i)}$ for each current density load $q^{(i)}$. Coefficients $w_r^{(i)} \geq 0$ are used to weight measurements according to confidence or using some geometric factor. The value of misfit F_d is a function of the electric conductivity σ (and therefore property function m) via (3) for potentials $\phi^{(i)}$.

In field experiments it is more likely that dipole potential measurements are taken with data misfit defined

$$F_{dc} = \frac{1}{2} \sum_i \sum_{r,c} w_{rc}^{(i)} \left(\phi^{(i)}\left(\mathbf{x}_r^{(i)}\right) - \phi^{(i)}\left(\mathbf{x}_c^{(i)}\right) - V_r^{(i)} + V_c^{(i)}\right)^2, \tag{9}$$

where weighting factor $w_{rc}^{(i)}$ is determined by both the load current source term $q^{(i)}$ and position of nodes $\mathbf{x}_r$ and $\mathbf{x}_c$. Summation is taken over all load cases (i) and all receiver pairs ($\mathbf{x}_r$, $\mathbf{x}_c$). Classical electrode configurations such as Wenner and Schlumberger can be defined using appropriate values for the weighting factors. The following derivations are presented for the pole configuration (8) but are easily generalized for dipole recorder configuration (9).

With modern multichannel loggers, say one with 64 electrodes that can act as either source or recorder, there are $\frac{64\times63}{2} = 2016$ possible loading dipoles and for each loading dipole there are $\frac{62\times61}{2} = 1891$ possible dipole recorder combinations (not all independent (Noel & Xu 1991)). For field data using all possible dipole loads and dipole recorders could possibly reduce the impact of data noise. In our method we need to solve PDE (3) (and the corresponding adjoint PDE as introduced later) for each loading dipole. All 4032 PDEs need to be solved for each iteration of the property function. However, forward problem (3) (and its corresponding adjoint problem) are second order linear PDEs and the superposition principle can be used to reduce computation time. Instead of solving 4032 PDEs, with superposition only 64 need be solved.

2.3 Property function and regularization

Data misfit functions (8) and (9) do not have unique minimums. In order to pick the 'simplest' answer a regularization term F_R is added to the cost function F (see Tikhonov & Arsenin 1977; Li & Oldenburg 1996). For simplicity we use the H^1 norm with a Tikhonov factor μ, however the method presented can easily be extended to other regularization approaches. Regularization could also reduce the impact on computed property functions of measurement errors in real data.

To ensure positive conductivity we introduce the property function m, with $\sigma = \sigma(m)$ such that $\sigma(m) > 0$ in Ω. Here we choose

$$\sigma = \sigma_0 e^m$$

with an estimated σ_0, for instance as the outcome of previous inversion in time-lapse ERT. We assume that throughout the inversion σ remains constant on the boundary of the domain except surface Γ_s. This is enforced by keeping $m = 0$ on $\partial\Omega\backslash\Gamma_s$. Again we assume that domain Ω is chosen sufficiently large so that errors in σ_0 on $\partial\Omega\backslash\Gamma_s$ have very little impact on the computed value of the

potential near the electrodes where we need to recover the distribution of σ.

We use scaled H^1 regularization of the property function in the form

$$\mu F_R = \frac{1}{2}\left(\mu_0 \|m\|^2_{L^2(\Omega)} + \mu_1 \|\nabla m\|^2_{L^2(\Omega)}\right), \tag{10}$$

where μ_0 and μ_1 are non-negative trade-off factors which need to be chosen appropriately, and $\|.\|_{L^2(\Omega)}$ is the L^2 norm over domain Ω,

$$\|a\|_{L^2(\Omega)} = \int_\Omega |a|^2 \mathrm{d}x.$$

Factor μ_0 is usually chosen to be small or zero (we will choose zero in our example) and factor μ_1 needs to be chosen small enough so that regularization does not dominate the cost function but also large enough to provide sufficient smoothing to the property function. It is pointed out here that it is a common approach in geophysics to apply smoothing to the already discrete version of m across a chosen grid or mesh, for example, Rücker *et al.* (2006b), but in this paper the continuous, integral form (10) is used to implement the minimization method.

2.4 Cost function and Gateaux derivative

The cost function to be minimized is

$$F = \frac{1}{2}\sum_i \sum_r w_{ir}\left(\phi^{(i)}(\mathbf{x}_r) - V_{ir}\right)^2 + \frac{1}{2}\left(\mu_0 \|m\|^2_{L^2(\Omega)} + \mu_1 \|\nabla m\|^2_{L^2(\Omega)}\right). \tag{11}$$

To minimize this cost function, we need to find function m for which the Gateaux derivative with respect to any incremental change in δm is zero as stated in equation (2). We express this in the form

$$\left\langle F'(m), \delta m\right\rangle = 0.$$

We will establish the representation for the derivative in the weak form

$$\left\langle F'(m), \delta m\right\rangle = \int_\Omega \left(\mathbf{X}\cdot\nabla\delta m + Y\,\delta m\right)\,\mathrm{d}x, \tag{12}$$

with suitable functions $\mathbf{X}$ and Y which depend on m but are independent of δm.

The scaled H^1 inner product corresponding to norm (10) is defined as

$$(p, v)_1 = \mu_0 \int_\Omega p\, v\,\mathrm{d}x + \mu_1 \int_\Omega \nabla p\cdot\nabla v\,\mathrm{d}x, \tag{13}$$

for all admissible property functions p, v. By admissible property functions we mean those functions for which norm (10) is well defined and that satisfy boundary conditions $m = 0$ on $\partial\Omega\backslash\Gamma_s$. For the regularization term in cost function (11), the Gateaux derivative is just the scaled H^1 inner product

$$\left\langle F'_R(m), \delta m\right\rangle = (m, \delta m)_1$$

which can easily be linked with representation (12).

Computation of the data misfit component of the derivative is more complicated. To simplify notation, we define the weighted defect for pole load i for electrode located at $\mathbf{x}_r$ to be

$$D_r^{(i)} = w_r^{(i)}\left(\phi^{(i)}\left(\mathbf{x}_r^{(i)}\right) - V_r^{(i)}\right). \tag{14}$$

Using this definition, the data misfit contribution to the cost function gradient is

$$\left\langle F'_d(m), \delta m\right\rangle = \sum_i \sum_r D_r^{(i)}\delta\phi^{(i)}\left(\mathbf{x}_r^{(i)}\right), \tag{15}$$

where $\delta\phi^{(i)}$ is the change in potential $\phi^{(i)}$ corresponding to the change in property function δm. While F_d is easily computed (it is just the Euclidean norm of the defects), it is more difficult to obtain estimates for $\left\langle F'_d(m), \delta m\right\rangle$.

Our approach is to establish a relationship between a change in property function δm and a corresponding change in potential $\delta\phi^{(i)}$. We first consider the incremented potential forward PDE. Misfit is a function of electric conductivity $\sigma(m)$ via PDE (3) for potentials $\phi^{(i)}$, so change in property function δm corresponds to change in potential and PDE (3) becomes (with no change to the right hand side)

$$\nabla\cdot\left(\sigma(m + \xi\delta m)\nabla(\phi^{(i)} + \xi\delta\phi^{(i)})\right) = q^{(i)}, \tag{16}$$

with ξ from (2). Clearly, $\phi^{(i)} + \xi\delta\phi^{(i)}$ will be in the same solution space as $\phi^{(i)}$, satisfying the same requirements and boundary conditions. Using a first order approximation $\sigma(m + \xi\delta m) \approx \sigma(m) + \xi\sigma'\delta m$, where σ' denotes the derivative of the conductivity σ with respect to property function m, (16) becomes

$$\nabla\cdot\left((\sigma + \xi\sigma'\delta m)\nabla(\phi^{(i)} + \xi\delta\phi^{(i)})\right) = q^{(i)}. \tag{17}$$

Expanding (17)

$$\nabla\cdot(\sigma\nabla\phi^{(i)}) + \xi\nabla\cdot(\sigma\nabla\delta\phi^{(i)}) + \xi\nabla\cdot\sigma'\delta m\nabla\phi^{(i)}) + \xi^2\nabla\cdot(\sigma'\delta m\nabla\delta\phi^{(i)}) = q^{(i)},$$

then simplifying, by cancelling the zero order terms and ignoring second-order terms in ξ as $\xi \to 0$, reduces it to

$$\nabla\cdot(\sigma\nabla\delta\phi^{(i)}) = -\nabla\cdot(\sigma'\delta m\nabla\phi^{(i)}).$$

In weak form this is given as

$$\int_\Omega \sigma\nabla\psi\cdot\nabla\delta\phi^{(i)}\mathrm{d}x = -\int_\Omega \sigma'\delta m\nabla\psi\cdot\nabla\phi^{(i)}\mathrm{d}x, \tag{18}$$

for any admissible potential ψ in the potential solution space.

For each forward PDE (3), we solve for an adjoint potential, $\phi_*^{(i)}$, induced by a load applied at each measuring electrode equal to the weighted defect $D_r^{(i)}$ at that electrode, with conductivity σ the same as the forward PDE,

$$-\nabla\cdot(\sigma\nabla\phi_*{}^{(i)}) = \sum_r D_r{}^{(i)}\delta(\mathbf{x} - \mathbf{x}_r), \qquad \text{in } \Omega, \tag{19}$$

where r is summed over all recorders. Just as superposition can be used to reduce the number of forward problems, superposition can also be used to reduce the number of adjoint PDE's that need to be solved. For pole or dipole measurements, the number of adjoint equations to be solved would be the minimum of the number of load cases and the number of recorder positions. (If the loading electrodes and recorder electrodes are the same and the boundary conditions for the forward and adjoint PDE's are the same then no new PDEs need be solved.) For a fixed number of electrodes, adding more measurements only increases the complexity of the right hand side of the adjoint PDE (19). As a consequence, minimizing the use of recording locations, driving the use of Schlumberger and Wenner configurations, is not a requirement, allowing the use of comprehensive arrays.

In weak form, adjoint PDE (19) is given as

$$\int_\Omega \sigma\nabla\phi_*^{(i)}\cdot\nabla\psi\,\mathrm{d}x = \sum_r D_r^{(i)}\,\psi(\mathbf{x}_{r^{(i)}}), \tag{20}$$

for any admissible potential ψ in the potential solution space. By setting $\psi = \delta\phi^{(i)}$, we obtain

$$\int_\Omega \sigma \nabla\phi_*^{(i)} \cdot \nabla\delta\phi^{(i)}\,\mathrm{d}x = \sum_r D_r^{(i)}\delta\phi^{(i)}(\mathbf{x}_{r^{(i)}}). \tag{21}$$

The summation of (21) over all load cases i is the value of the data misfit derivative (15), so

$$\left\langle F_d'(m), \delta m\right\rangle = \sum_i \int_\Omega \sigma \nabla\phi_*^{(i)} \cdot \nabla\delta\phi^{(i)}. \tag{22}$$

Returning to (18) and choosing $\psi = \phi_*^{(i)}$, we obtain

$$\int_\Omega \sigma \nabla\phi_*^{(i)} \cdot \nabla\delta\phi^{(i)}\mathrm{d}x = -\int_\Omega \sigma'\delta m \nabla\phi_*^{(i)} \cdot \nabla\phi^{(i)}\mathrm{d}x. \tag{23}$$

Combining (22) with the summation over the dipoles i in (23), we obtain

$$\left\langle F_d'(m), \delta m\right\rangle = -\int_\Omega \sigma' \sum_i \left(\nabla\phi_*^{(i)} \cdot \nabla\phi^{(i)}\right)\delta m\,\mathrm{d}x,$$

for all property function increments δm.

With this result and the gradient representation (12) for the regularization component, we can now establish coefficient functions for gradient F' of the cost function as

$$\mathbf{X} = \mu_1 \nabla m$$
$$Y = \mu_0 m - \left(\sigma' \sum_i \nabla\phi_*^{(i)} \cdot \nabla\phi^{(i)}\right). \tag{24}$$

It is emphasized that calculation of the gradient requires the solution of adjoint PDE (20) for each input source where the forward PDE and the adjoint PDE are solved independently.

For dipole measurements using the dipole defect function (9) the process is the same except with adjoint dipole applied loads,

$$-\nabla\cdot(\sigma\nabla\phi_*^{(i)}) = \sum_{r,c} D_{rc}^{(i)}(\delta(\mathbf{x}-\mathbf{x}_{r^{(i)}}) - \delta(\mathbf{x}-\mathbf{x}_{c^{(i)}})), \tag{25}$$

where

$$D_{rc}^{(i)} = w_{rc}^{(i)}\left(\phi^{(i)}(\mathbf{x}_{r^{(i)}}) - \phi^{(i)}(\mathbf{x}_{c^{(i)}}) - (V_{ir} - V_{ic})\right),$$

and $w_{rc}^{(i)}$ is the weighting for measuring dipole for load case (i), with electrodes at $\mathbf{x}_r$ and $\mathbf{x}_c$.

3 SOLUTION METHOD

We need to find property function m, such that

$$\left\langle F'(m), \delta m\right\rangle = \int_\Omega (\mathbf{X}\cdot\nabla\delta m + Y\,\delta m)\,\mathrm{d}x = 0,$$

for all admissible property increments δm. We use the BFGS method to iteratively find an acceptable approximation for m. This method is generally formulated for solving nonlinear, unconstrained optimization problems in the Euclidean space $\mathbb{R}^n$, see Nocedal & Wright (2006). For our implementation, we apply it directly to the property function m and gradient $F'(m)$ where the usual dot product is replaced by an integral. In the $k+1$ BFGS iteration step, property function m_k, is updated by first finding search direction p_k, then step size α_k. The update is

$$m_{k+1} = m_k + \alpha_k p_k. \tag{26}$$

This is a quasi-Newton method so $-p_k$ is an approximation to the product of the inverse Hessian of F at m_k and the gradient at m_k. To orthogonalize search directions the dual product (12) is used instead of standard Euclidean dot product which will be discussed later in more details.

Typically, the iteration is started from σ_0, that is, $m_0 = 0$. Each weak form of the forward PDE (7) is solved and weighted defects (14) are computed for each measuring electrode. An update to m is found in two parts—first search direction p_k (section 3.1) then step size α_k (Section 3.2). The calculation of the search direction requires the solutions of the adjoint PDEs (20) and an approximation to the Hessian which is modified within BFGS using the two step algorithm discussed below. Step size α_k is found via a line search method and satisfies the strong Wolfe conditions (Nocedal & Wright 2006). The property function is updated using (26). This process is repeated until convergence is detected. Convergence criteria for iteration termination is

$$\|m_k - m_{k-1}\|_\infty \le m_{\mathrm{tol}}\|m_k\|_\infty,$$

where m_{tol} is the relative tolerance and $\|.\|_\infty$ is the L-infinity norm (the maximum absolute value over the domain).

Algorithm 1 Two loop recursion in the BFGS method

$\langle T, \circ\rangle \leftarrow \langle F_k', \circ\rangle$
▷Orthogonalization:
for $j = k-1, k-2, \ldots k-a$ **do**
  $\gamma_j = \rho_j \langle T\ ,\ s_j\rangle$
  $\langle T, \circ\rangle \leftarrow \langle T, \circ\rangle - \gamma_j \langle G_j, \circ\rangle$
end for
▷Approximate Inverse of Hessian:
solve for p: $(p, \delta m)_1 = \langle T, \delta m\rangle$ for all δm
▷Update Inverse of Hessian:
for $j = k-a, k-a+1, \ldots k-1$ **do**
  $p \leftarrow p + s_j\left(\gamma_j - \rho_j\langle G_j, p\rangle\right)$
end for
return $p_k = p$

3.1 Search direction

Each new search direction p_k is constructed with a sequence of inner products and additions using the current gradient, stored gradient differences and stored property function differences as well as an initial approximation for the Hessian operator. Note that the BFGS algorithm does not require an approximation of the inverse Hessian at m_k and the gradient at m_k separately. Instead, an approximation to the product of the inverse Hessian and the gradient at m_k is used. The two loop algorithm from Nocedal & Wright (2006) returns step direction p_k. It is a outlined in Algorithm 1.

To simplify the notation, instead of $F'(m_k)$ we use F_k' with corresponding terms $\mathbf{X}_k = \mathbf{X}(m_k)$ and $Y_k = Y(m_k)$. We define notation

$$\left\langle F_k', \circ\right\rangle = \int_\Omega (\mathbf{X}_k\cdot\nabla\circ + Y_k\circ)\,\mathrm{d}x,$$

to indicate that this integral is not evaluated but its components $\mathbf{X}_k$ and Y_k are simply stored. Similarly, we store gradient differences as

$$\langle G_k, \circ\rangle = \left\langle F_{k+1}' - F_k', \circ\right\rangle = \int_\Omega (\mathbf{X}_{k+1} - \mathbf{X}_k)\cdot\nabla\circ + (Y_{k+1} - Y_k)\circ\,\mathrm{d}x.$$

The limited-memory BFGS (L-BFGS) version keeps gradient differences and search directions for only a fixed number of previous iterates, a. Property function differences are also stored,

$$s_j = m_{j+1} - m_j = \alpha_j p_j,$$

for all $j \in [k-a, k-1]$. The inner product of the gradient difference and the step is defined

$$\langle G_j, s_j \rangle = \int_\Omega \left(\mathbf{X}_j - \mathbf{X}_{j-1}\right) \cdot \nabla s_j + (Y_j - Y_{j-1}) s_j \, \mathrm{d}x.$$

Temporary variable T is initialized with F_k' and then orthogonalized against previous stored gradient differences. We denote the combinations of $\mathbf{X}_j$ and Y_j used in the computation of T as $\hat{\mathbf{X}}$ and $\hat{Y}$,

$$\langle T, \circ \rangle = \int_\Omega (\hat{\mathbf{X}} \cdot \nabla \circ + \hat{Y} \circ) \, \mathrm{d}x.$$

Stored T is updated at every iterate in the first part of the two loop recursion algorithm. We define and store

$$\rho_j = \frac{1}{\langle G_j, s_j \rangle},$$

for all $j \in [k-a, k-1]$. In the first part of the two loop recursion we compute and temporarily store terms γ_j. For these terms we evaluate T at s_j,

$$\langle T, s_j \rangle = \int_\Omega \hat{\mathbf{X}} \cdot \nabla s_j + \hat{Y} s_j \, \mathrm{d}x.$$

The BFGS algorithm requires an initial approximation to the Hessian which is updated in the second loop of Algorithm 1. In our case, as in many practical cases, it is difficult or expensive to construct the Hessian operator exactly but the BFGS method is able to deal with an initial approximation to the Hessian operator which then plays the role of a preconditioner. Here we use the Hessian operator of the regularization term only, the scaled H^1 inner product (13). Search direction p is found by solving

$$(p, \delta m)_1 = \langle T, \delta m \rangle \tag{27}$$

for every admissible property function increment δm with T from the first loop of Algorithm 1. It is important to note that it is not necessary to use the same high tolerance in the computation for the initial Hessian guess as was used to solve the forward and adjoint PDE's making a significant saving in computation time.

3.2 Step size

Once the search direction is known, the step size is computed using a line search method. Full details for this method can be found in Nocedal & Wright (2006). Optimal step length is given by

$$\alpha_k = \arg\min_\alpha \; F(m_k + \alpha \cdot p_k).$$

Solving this minimization problem is computationally expensive as every step size guess requires a new solution of forward PDEs (7) and cost function evaluation (11). Instead, we find an α_k that satisfies the strong Wolfe conditions (Nocedal & Wright 2006). The first Wolfe condition or Armijo condition for α ensures that the next iterate has sufficient decrease

$$F(m_k + \alpha p_k) \leq F_k + c_1 \alpha \left\langle F_k', p_k \right\rangle, \tag{28}$$

with $c_1 \in (0, 1)$. (We use $c_1 = 10^{-4}$). The second condition for α is a curvature condition and ensures that the step size is not too small

$$|\left\langle F'(m_k + \alpha p_k), p_k \right\rangle| \leq c_2 |\left\langle F_k', p_k \right\rangle|, \tag{29}$$

with $c_2 \in (c_1, 1)$. (We use $c_2 = 0.9$.)

The search strategy has two parts. The first part searches for a bracketing interval (α_a, α_b), that contains a step length that satisfies the strong Wolfe conditions (28) and (29). Let $\hat{\alpha}_i$ denote the ith guess for a bound in step length. For the first BFGS iterate we use $(\hat{\alpha}_0, \hat{\alpha}_1) = (0, 1)$ and for the kth BFGS iterate we use $(\hat{\alpha}_0, \hat{\alpha}_1) = (0, \alpha_{k-1})$ The value of each successive $\hat{\alpha}_i$ is increased until one of the following three conditions is met:

(i) $\hat{\alpha}_i$ violates the first Wolfe condition (28),
(ii) $F(m_k + \hat{\alpha}_i p_k) \geq F(m_k + \hat{\alpha}_{i-1} p_k)$ or
(iii) $\langle F'(m_k + \hat{\alpha}_i p_k), p_k \rangle \geq 0$.

If $\hat{\alpha}_i$ satisfies either of the first two conditions, then bracketing interval $(\alpha_a, \alpha_b) = (\hat{\alpha}_{i-1}, \hat{\alpha}_i)$ is used for the second part of the step size algorithm. If the third condition is satisfied then the roles of $\hat{\alpha}_{i-1}$, and $\hat{\alpha}_i$ are reversed and the interval $(\alpha_a, \alpha_b) = (\hat{\alpha}_i, \hat{\alpha}_{i-1})$ is used instead.

Once an appropriate interval is found, the second part of the line search algorithm is called. It uses a bisection method on interval (α_a, α_b) to find a step length that satisfies the strong Wolfe conditions (28) and (29). At each step, the interval is reduced such that it still brackets an appropriate step length (using the same three conditions above). Once a step length is found that satisfies both strong Wolfe conditions then it is returned as the step length α_k and the line search terminates.

4 FINITE ELEMENTS AND ALGEBRAIC MULTIGRID PRECONDITIONED CONJUGATE GRADIENT

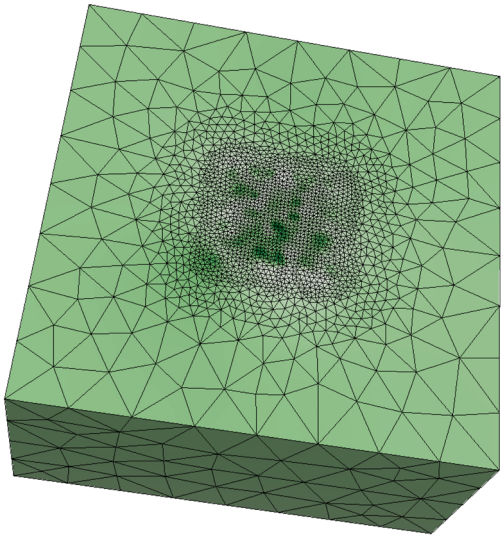
We discretize the PDEs using conforming first-order tetrahedral finite elements on structured and unstructured grids, see Fig. 1. The finite element method is well established so we do not present it here (see for instance Zienkiewicz *et al.* (2013) for details). As shown in Lamichhane & Gross (2017) well-posedness of the overall problem requires the same finite element mesh for the forward PDEs (7) and adjoint PDEs (20) while the mesh for the property function m can be chosen more or less independently. To simplify the discussion in this paper we use the same finite element mesh for all three PDEs involved. Potentials $\phi^{(i)}$, adjoint potentials $\phi_*^{(i)}$ and the property function m, are represented through the values at the vertices of the tetrahedrons. The cost function gradient and all the functions derived from this during the BFGS iteration are stored by their values at the quadrature points within each tetrahedron of the mesh. In our implementation, we also use this mesh to represent the conductivity (through the property function) although the resolution for the property function may then be much higher than necessary (Lamichhane & Gross 2017). Lower resolution meshes for the property function could be used and would improve efficiency of the inversion but would require a careful design of the inter-mesh interpolation, in particular when building the right-hand side of the preconditioner step (27) in the middle of the two loop recursion Algorithm 1.

After FEM discretization, each PDE reduces to a matrix equation of type

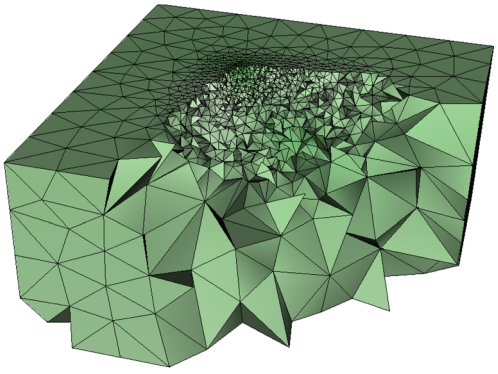
$$\mathbf{A}^h \mathbf{u}^h = \mathbf{f}^h, \tag{30}$$

where $\mathbf{A}^h$ is the operator matrix, $\mathbf{u}^h$ is the vector of unknowns at the element vertices, $\mathbf{f}^h$ represents the right hand side. The upper index h indicates the mesh size. For the PDEs to be solved the operator matrix is sparse, symmetric and positive definite and therefore the system of equations (30) can be solved iteratively using the PCG method (Golub & Van Loan 1996). To reduce computation time we use smoothed aggregation AMG (Vaněk *et al.* 1996) as a preconditioner for conjugate gradient. We give a brief outline of the method.

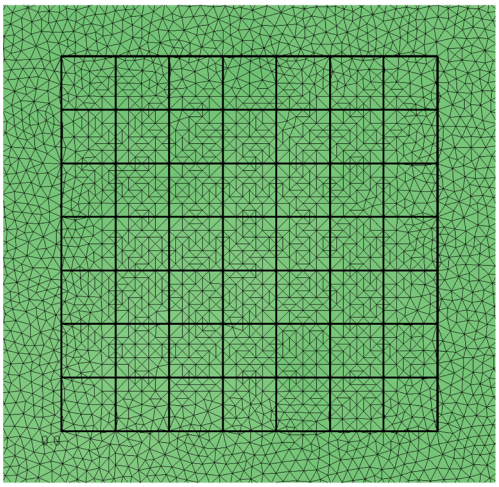
Iterative methods initially converge rapidly as the oscillatory error is reduced and stall when only smooth error is left. At this stage further iterations are ineffective at removing error. Geometric Multigrid (GMG) methods were developed to reduce computation



(a) Finite element mesh for entire domain 80m × 80m × 40m.



(b) Cut through the finite element mesh to show finer mesh around electrodes.



(c) Close up view of finite element mesh around electrodes at the air/ground interface, 16m × 16m.

Figure 1. Example 5.2: Finite Element Mesh for Rectangular Surface 8 × 8 electrode survey on Heron Island with a close up of the mesh at the ground surface and electrodes. It has 40 209 nodes and 248 521 elements.

Algorithm 2 Two grid AMG solver

Grid	**Action**
h	**Iterate** on $\mathbf{A}^h\mathbf{u}^h = \mathbf{f}^h$ to get $\mathbf{v}^h$
h	**Compute** fine grid residual $\mathbf{r}^h = \mathbf{f}^h - \mathbf{A}^h\mathbf{v}^h$
$h \rightarrow H$	**Restrict** fine grid residual $\mathbf{r}^h$ to coarse grid,$\mathbf{r}^H$ $\mathbf{r}^H = \mathbf{I}_h^H\mathbf{r}^h$
H	**Iterate** on $\mathbf{A}^H\mathbf{e}^H = \mathbf{r}^H$ to get $\mathbf{e}^H$ (or use direct solver) $\mathbf{A}^H = \mathbf{I}_h^H\mathbf{A}^h\mathbf{I}_H^h$
$H \rightarrow h$	**Interpolate** coarse error $\mathbf{e}^H$ to fine grid $\mathbf{e}^h$ $\mathbf{e}^h = \mathbf{I}_H^h\mathbf{e}^H$
h	**Correct** fine grid solution with error $\mathbf{v}^h = \mathbf{v}^h + \mathbf{e}^h$
h	**Iterate** on $\mathbf{A}^h\mathbf{u}^h = \mathbf{f}^h$ to get $\mathbf{v}^h$

time in iterative matrix solvers for structured grids, with the object to bound the number of iteration steps even for an increasing number of unknowns (Briggs *et al.* 2000). GMG uses a sequence of structured grids with each successive coarse grid usually doubling grid size in each direction. The following description is for two level GMG, with fine grid h and coarse grid H. For multiple levels, this algorithm is just applied recursively. We iterate on the fine grid solving eq. (30) to get an approximate fine grid solution $\mathbf{v}^h$. Fine grid residual $\mathbf{r}^h$ is defined as

$$\mathbf{r}^h = \mathbf{f}^h - \mathbf{A}^h\mathbf{v}^h. \tag{31}$$

Iteration error on the fine grid, $\mathbf{e}^h$, can be written

$$\mathbf{e}^h = \mathbf{u}^h - \mathbf{v}^h. \tag{32}$$

Combining (30), (31) and (32) leads to the residual equation on the fine grid

$$\mathbf{A}^h\mathbf{e}^h = \mathbf{r}^h.$$

After iteration on the fine grid, the error is smooth on this grid and it can be represented well on the coarse grid, where it will appear more oscillatory. It is important to remember that even if the residual is small, it does not mean that the error $\mathbf{e}_h$ is small, only that the error is smooth in the sense of $\mathbf{A}^h\mathbf{e}^h \approx 0$. A restriction operator, $\mathbf{I}_h^H$, is defined to restrict fine grid residual to the coarse grid,

$$\mathbf{r}^H = \mathbf{I}_h^H\mathbf{r}^h,$$

where $\mathbf{r}^H$ is a representation of the fine grid residual on coarse grid H. Restriction could be injection where the value at each coarse grid point is simply the value at the corresponding fine grid point. More generally the value at a coarse grid point is a combination of values from the corresponding fine grid point and its immediately adjacent neighbours. As well as restriction operators to transfer solutions from level h to level H, there are also interpolation operators, $\mathbf{I}_H^h$, that transfers solution from level H to level h. The simplest interpolation operator is linear interpolation, fine grid values at nodes that appear in the coarse grid just take that value and all other fine grid values are linearly interpolated from neighbouring points. The full weighting restriction operator is the transpose of linear interpolation. Using restriction and interpolation operators, the coarse grid matrix is defined

$$\mathbf{A}^H = \mathbf{I}_h^H\mathbf{A}^h\mathbf{I}_H^h.$$

The next step of the GMG algorithm is to iterate on the coarse grid residual equation

$$\mathbf{A}^H\mathbf{e}^H = \mathbf{r}^H. \tag{33}$$

with zero initial guess, where $\mathbf{e}^H$ is the coarse grid error. If the coarse grid is sufficiently small this equation is solved using a direct solver (based on LU factorization). Alternatively, the coarsening process can be applied again after the equation has been solved iteratively to remove oscillatory components. Once eq. (33) has been solved with sufficient accuracy, $\mathbf{e}^H$ can be easily interpolated to the fine grid,

$$\mathbf{e}^h = \mathbf{I}_H^h \mathbf{e}^H,$$

where it is used to correct the approximate solution

$$\mathbf{v}^h \leftarrow \mathbf{v}^h + \mathbf{e}^h.$$

Further iteration on the fine grid (30) smooths the error before returning back to the coarse level. The method is outlined in Algorithm 2 for two level case with more details in Briggs *et al.* (2000). It is important to note that moving errors or residuals between grids only occurs after the error has been smoothed on that grid to minimize loss of information.

AMG was developed for use in unstructured grids but also to overcome some of the limitations of GMG in the presence of spatially variable conductivity and of anisotropy. Coarse 'grids' (they are not grids as such, but this terminology aids understanding), including the restriction and interpolation operators, are determined from the entries in matrix $\mathbf{A}^h$ alone, without reference to any mesh information. There are a number of different AMG algorithms, see Stuben (2001). We use smoothed aggregation, see Vaněk *et al.* (1996), Tuminaro (2000).

The first step of this algorithm is to form the coarse grid by using strong connections in the matrix $\mathbf{A}^h$ to select disjoint aggregates. Two unknowns i and j are called strongly connected if $\mathbf{A}^h$ has a large nonzero entry in the ij position relative to the diagonal entry. A graph of the strong connections is created with each vertex representing an unknown and each strong connection represented by an edge. The idea is to form a new coarse graph with a maximum independent set of vertices, each coarse vertex representing an aggregate of fine grid vertices that defines the set of unknowns for the coarse grid. Any two coarse grid vertices do not share an edge in the fine grid, but each coarse grid vertex shares an edge with a vertex outside the maximum independent set. Such a set can easily be constructed using a greedy search. Once the coarse level is known the next step is to determine the interpolation, $\mathbf{I}_H^h$, and restriction, $\mathbf{I}_h^H = (\mathbf{I}_H^h)^t$, operators. A piecewise constant initial guess for the interpolation operator is chosen with 1 in position ij only if node i is strongly connected to node j in the fine grid, otherwise the entry is 0. The initial interpolation operator is then smoothed with a few steps of a Jacobi method, see Vaněk *et al.* (1996)) for more details. Once the interpolation and restriction operators are known and grids chosen, algorithm 2 is followed.

The most time critical part of an AMG algorithm is determining coarse grids and corresponding operators. Computation of the direct solver used for the coarsest grid solve can also be computationally intensive. As the operator matrix for the forward and adjoint PDEs are identical, we are able to implement an AMG reuse strategy that constructs the AMG operators for the current electric conductivity once only at the beginning of each BFGS iteration step. All subsequent forward and adjoint PDEs for this property function reuse the AMG data, significantly reducing computational costs. The Hessian computation is considered separately and is also solved using AMG, however it does not need to be solved to the same tolerance as the forward or adjoint PDEs. We have implemented the BFGS inversion method in *python* using FEM solver environment *esys–escript*

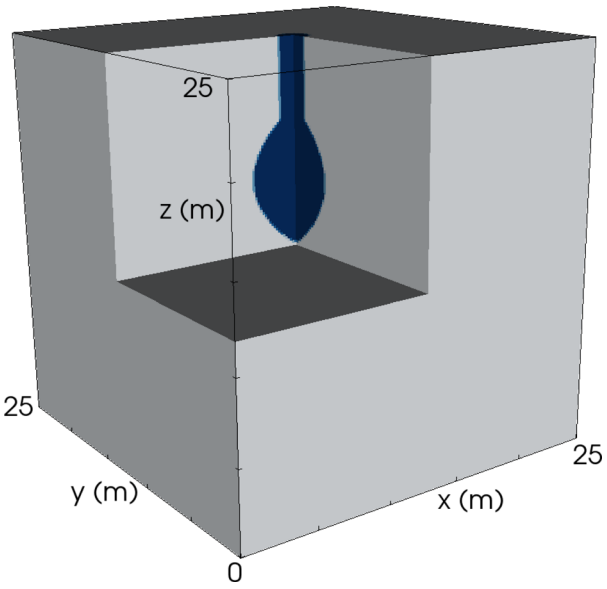


Figure 2. Example 5.1: Input conductivity for synthetic test case. The grey area has conductivity 0.01Ωm and the dark blue region has conductivity 0.2 Ωm

(Schaa *et al.* 2016). To solve the matrix equations it uses the TRILINOS C++ library (Heroux *et al.* 2005) with *MueLu* (Prokopenko *et al.* 2014) AMG implementation. In the test on parallel computers presented in the next section we use two levels of coarsening. A better parallel repartitioning of the coarse grids would increase the number of levels we could use and potentially reduce computation time further.

5 TEST CASES

We tested our algorithm using synthetic data created with a known conductivity distribution to mimic CO_2 injection. For this inversion problem we looked at the contribution AMG made to reduction in computation time as well as optimal BFGS parameters and Tikhonov regularization factor. There were two distinct sets of electrodes, the loading electrodes and the measuring electrodes. For this test case, loading electrodes never acted as measuring electrodes.

The second inversion test case used data obtained from Heron Island off the east coast of Australia. For this test case, all recorded data (576 dipoles) were used for the inversion and the 64 electrodes acted as either a loading electrode or a receiver electrode for each load case. Superposition was used to compute potentials for both the forward and adjoint problems. Using superposition for this case significantly reduced computation time.

Tests were carried out on an Intel (R) Core(TM)i7-4770 CPU@3.4 GHz desktop computer running Debian 8.5 Linux, using Open MP with 8 threads, 1 MPI process and 32GiB of memory and also on Savanna, an Intel Xeon CPU E5-2660 computer running Linux, using 4 nodes, each with 126 GiB of memory and 2 MPI processors and 10 open MP threads per process.

5.1 Synthetic case

Our first test case was a cube of edge length 25m similar to a test case from Pidlisecky *et al.* (2007). The air/ground interface was the top surface of the cube. Input conductivity, σ, is shown in Fig. 2. Sources were set up as a 5×5 grid at a depth of 4 m with 2 m spacing.

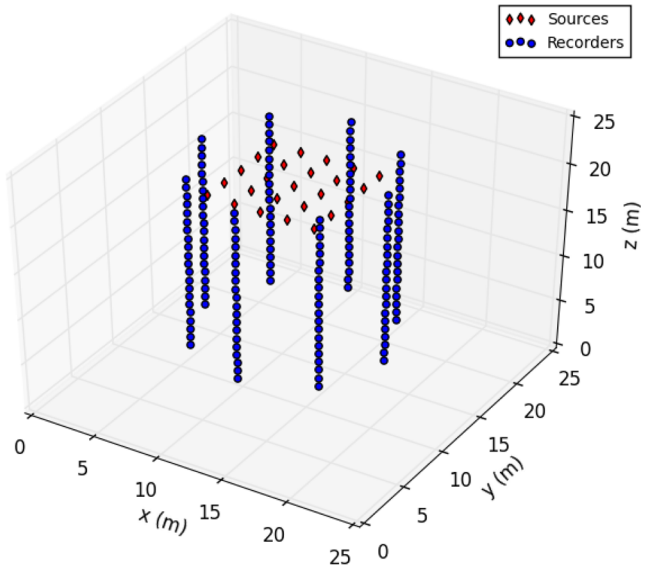


Figure 3. Example 5.1: Source and receiver placement for synthetic test case.

Table 1. Example 5.1: Total computation time for 40 forward solves of forward problem (7) with default *esys–escript* PCG solver, TRILINOS PCG (both with Jacobi preconditioning) and TRILINOS AMG PCG on desktop.

Total nodes	Grid	*esys–escript* PCG (s)	TRILINOS PCG (s)	TRILINOS AMG (s)
140 608	51^3	40	41	7
1 000 000	99^3	696	616	46
3 511 808	151^3	3719	3372	165
8 000 000	199^3			366
10 648 000	219^3			500

Recorders were located on the surface of a cylinder with radius 7m and vertical central axis in the centre of the horizontal x–y plane. There were 8 recorders at each horizontal level, (equally spaced around the circle) and there were 21 vertical levels of recorders from 2m depth to 20m depth, a total of 168 measuring electrodes, shown in Fig. 3. Data were created using dipole loads and pole measurements over structured meshes.

Inversion computation time is dominated by solving the forward and adjoint PDEs. Using *esys–escript* default tolerances, we compared computation times for solving forward problem (7) 40 times to mimic one step of BFGS so that solver choices could be optimized, see Table 1. Computation time differences between the PCG *esys–escript* solvers and TRILINOS PCG were minimal. There was significant reduction in computation time from Jacobi PCG to AMG PCG for solving forward problem (7). For the 51^3 grid, TRILINOS AMG was about 6 times faster than TRILINOS PCG and for the 151^3 grid this had improved even more to about 20 times faster, see Table 1. AMG setup time is only needed for the first forward PDE, see Table 2. Subsequent forward PDE solutions reuse the interpolation and restriction operators as well as the coarse grid direct solver. For this example, compute time for AMG grows linearly with grid size for both the initial solve and subsequent solves. Times are shown only for the smaller grids for Jacobi PCG as computation time for the 99^3 grid was larger than the time for the AMG PCG on the 219^3 grid.

Table 2. Example 5.1: Average time per forward PDE solve with default *esys–escript* PCG solver, TRILINOS PCG and AMG PCG on desktop. For AMG PCG time was recorded for the first solve which included the generation of the AMG operators and coarse grid solver and then the average of the subsequent 39 solves.

Total nodes	Grid size	*esys–escript* PCG (s)	TRILINOS PCG (s)	AMG first (s)	AMG rest (s)
14 068	51^3	1.0	1.0	0.6	0.17
1 000 000	99^3	17.4	15.4	3.9	1.1
3 511 808	151^3	93.0	84.3	13.9	3.9
8 000 000	199^3			33.7	8.5
15 813 251	219^3			47.5	12.5

Table 3. Example 5.1: Computation times for solving pole load forward PDEs (7), for the first solve, which included the generation of the AMG operators and coarse grid solver, the average of the subsequent 24 solves, total time for all 25 pole loads and construction time for 300 dipole potentials using superposition with TRILINOS AMG PCG on desktop.

Total nodes	Grid size	PDE first (s)	PDE rest (s)	PDE all 25 (s)	Construct 300 dipoles (s)
140 608	51^3	0.6	0.17	4.6	0.38
1 000 000	99^3	3.9	1.1	32.0	1.44
3 511 808	151^3	13.9	3.9	108.4	4.50
8 000 000	199^3	33.7	9.1	259.5	9.98
15 813 251	251^3	115.6	18.1	550.1	21.06

Instead of solving dipole load forward problems (7), pole load forward PDEs can be solved and dipole potentials computed using superposition. To assess the potential reduction in computation time using superposition, we solved 25 independent pole PDEs and used superposition to compute all possible dipole potentials. Computing all 300 possible dipole potentials by superposition takes less time than solving the first pole equation or two subsequent pole equations (see Table 3) for each grid size. This will be especially relevant for examples where dipole measuring is used.

There are a number of factors that could impact the computed solution and computation time for the inversion, including stopping criteria for the BFGS iterations m_{tol}, PDE solver tolerance pde_{tol}, Tikhonov regularization factors μ_0, μ_1, number of dipoles and choice of which dipoles were used for applied loads and for measurements if dipole measurements were being used. We chose $m_{\text{tol}} = 10^{-4}$ for LBFGS stopping criteria. There is no increase in AMG setup time for increasing PDE solver tolerance because computing coarse grids and their operators as well as computing the direct coarse grid solver are independent of solver tolerance. So increasing PDE solver tolerance simply increases computation time uniformly for all solves. We set $pde_{\text{tol}} = 10^{-8}$ for all the tests. For the synthetic case, there were 25 loading electrodes with no measurement errors. To ensure spatial symmetry of the solution, loading dipoles used in the inversion were every adjacent pair in both the x and y directions, 40 dipoles in total (from a possible 300). We assumed pole data measurements on the array of 168 recorder electrodes. Tikhonov factor μ_0 was set to 0.0. The PDE tolerance for the Hessian approximation used in the inversion was set at 10^{-2}. Increasing accuracy of the initial Hessian guess added to computation time without reducing the number of BFGS iterations. There are 40 adjoint PDEs, so using superposition for computing solutions to the adjoint PDEs would not reduce computation time. If all 300 load dipoles were used in the inversion then there would

Table 4. Example 5.1: Variation in number of stored BFGS iterates used in search direction computations for the 51^3 grid with $pde_{\text{tol}} = 10^{-8}$ and $\mu_1 = 10^{-4}$ with TRILINOS AMG PCG on desktop.

Recursion length	BFGS steps	PDE calls	Gradient calls	Inner Products	Norm calls	Time (s)
3	56	122	62	504	114	1480
6	51	112	57	741	104	1364
12	39	90	44	926	80	1108
18	38	87	43	1181	78	1115
24	38	84	42	1390	78	1079
27	37	80	41	1411	76	1067
30	31	71	35	1087	64	926

be an advantage to using superposition for both the forward and adjoint PDEs.

Variation was made to the recursion length a used in search direction computations in the BFGS algorithm, see Table 4. As one expects, increasing the number of stored BFGS iterates, increased the number of inner products used in search direction computations. However, each computed search direction, while potentially taking longer to compute due to the extra costs in the orthogonalization and inverse Hessian update, was improved and there was a significant reduction in the number of BFGS iterations needed. The number of PDE calls was the same as the number of cost function calls and these, along with gradient and norm calls, were reduced. In this test we used a 51^3 grid with $pde_{\text{tol}} = 10^{-8}$ and $\mu_1 = 10^{-4}$. In the following tests recursion length $a = 30$ is used.

The biggest impact on computation time and computed inversion result came from the choice of μ_1 (we set $\mu_0 = 0$). Increasing μ_1 makes the regularization term more dominant in the cost function value. As expected, the solution is smoother with much smaller variation in computed conductivity values and takes fewer BFGS steps with increasing size of μ_1, see Fig.4 and Table 5 for the 51^3 grid. As μ_1 decreases in value, the maximum value of the conductivity in the centre of the plume increases. A similar result was seen for the 99^3 grid, see Table 6. For this test case, there appeared to be little variation in the number of BFGS iterations with change in grid size compare Tables 5 and 6. There was no notable change if the input potential data was from a finer grid rather than the inversion grid as also reported by (Gross *et al.* 2015).

The implementation was also tested on Savanna, an Intel Xeon CPU E5-2660 running Linux, using 4 nodes each with 2 MPI processors and 126 GiB of memory and 10 open MP threads per process. As for the desktop computer, the forward PDE (7) was solved 40 times to mimic one step of BFGS and to assess computation time with grid size. As can be clearly seen in Table 7 and the corresponding graphs in Fig. 5 the total time for the PDE solves is growing almost linearly with number of finite element nodes. Inversion was tested for both the 51^3 and 99^3 grids see tables 8 and 9. The results are slightly different to the desktop results because different parallelization of the code occurs. As expected the computed conductivity σ was the same for both the desktop and Savanna. The result for the 99^3 grid on Savanna was smoother than the 51^3 grid as expected for similar values of μ_1 see Fig. 6.

5.2 Heron Island data using unstructured grid

Heron Island is small tropical island located at the southern end of the Great Barrier Reef, about 80km off the east coast of

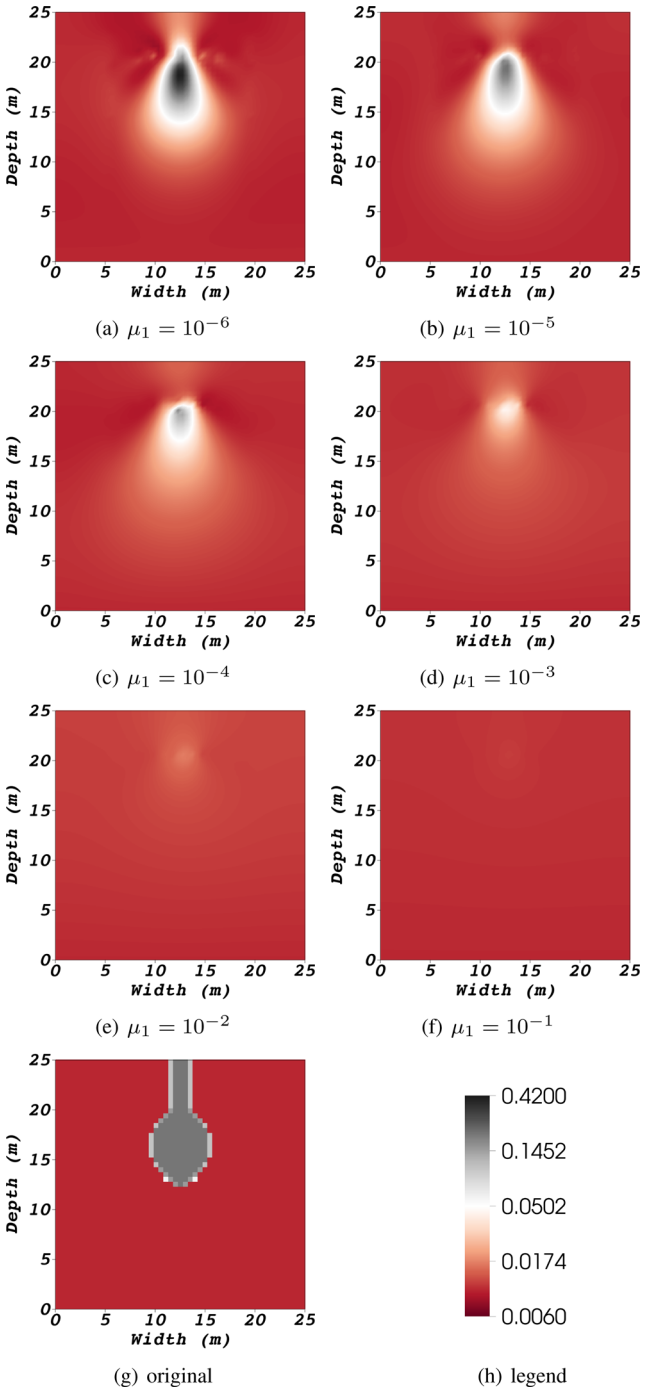


Figure 4. Example 5.1: Varying Tikhonov regularization factor μ_1 for the 51^3 grid with $m_{\text{tol}} = 10^{-4}$ and $pde_{\text{tol}} = 10^{-8}$ on desktop.

Table 5. Example 5.1: Varying Tikhonov regularization factor μ_1 for the 51^3 grid with $m_{\text{tol}} = 10^{-4}$ and $pde_{\text{tol}} = 10^{-8}$.

μ_1	BFGS steps	PDE calls	Gradient calls	Inner Products	Norm calls	$\mu_1 F_R$ value	F_d value
10^{-1}	4	6	6	34	10	0.0088	0.0834
10^{-2}	6	10	8	62	14	0.0128	0.0528
10^{-3}	13	24	16	224	28	0.0149	0.0173
10^{-4}	31	71	35	1087	64	0.0058	0.0017
10^{-5}	91	196	98	3940	184	0.0010	0.0002
10^{-6}	273	601	297	12633	548	0.0002	3 e-05

Table 6. Example 5.1: Varying Tikhonov regularization factor μ_1 for the 99^3 grid with $m_{\text{tol}} = 10^{-4}$ and $pde_{\text{tol}} = 10^{-8}$ on desktop.

μ_1	BFGS steps	PDE calls	Gradient calls	Inner Products	Norm calls	$\mu_1 F_R$ values	F_d values
10^{-1}	4	5	5	23	8	0.0029	0.0450
10^{-2}	6	9	7	47	12	0.0052	0.0331
10^{-3}	15	24	16	254	30	0.0081	0.0144
10^{-4}	30	62	35	963	60	0.0046	0.0018
10^{-5}	92	201	105	3947	184	0.0009	0.0002

Table 7. Example 5.1: Total computation time, first solve computation time and average computation time for subsequent 39 solves for solving forward PDE (7) using TRILINOS AMG on Savanna using 4 nodes, 2 MPI processors per node, 10 open MP threads per process and 126 GiB of memory.

Total nodes	Grid size	Total time (s)	First time (s)	Average time rest (s)
14 068	51^3	1.7	0.1	0.04
100 000	99^3	6.3	0.6	0.1
3 511 808	151^3	22.1	2.7	0.5
8 000 000	199^3	65.8	8.8	1.5
16 003 008	251^3	148.0	24.2	3.2
27 000 000	299^3	290.7	65.9	5.8
43 614 208	351^3	392.1	127.6	6.8
64 000 000	399^3	720.5	255.3	11.9
92 345 408	451^3	1269.5	589.5	17.4

Central Queensland, Australia (Jell & Webb 2012). It was formed by coral sand on top of a platform reef and is about 300 m × 800 m with elevation less than 4m. A small 3-D ERT survey was run in the centre of the island in a Pisonia forest with the objective to identify the density and depth of the Pisonia tree roots as they play a key role in the fresh water balance of the island. The area was undermined by abundant nesting burrows of the wedge-tailed shearwater.

Data was obtained using a FlashRES-UNIVERSAL System from ZZ Resistivity Imaging Pty Ltd with 64 electrodes. These were

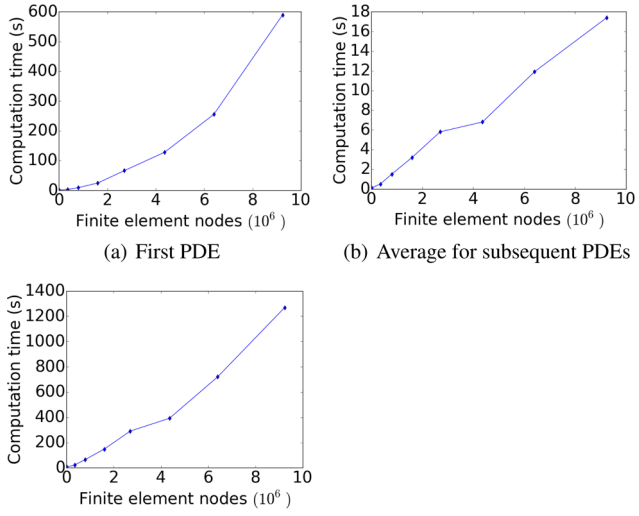


Figure 5. Example 5.1: First, average subsequent and total computation times for AMG solves varying grid size with $m_{\text{tol}} = 10^{-4}$ and $pde_{\text{tol}} = 10^{-8}$ using TRILINOS AMG on Savanna using 4 nodes, 2 MPI processors per node, 10 open MP threads per process and 126 GiB of memory.

Table 8. Example 5.1: Varying μ_1, with $m_{\text{tol}} = 10^{-4}$ and $pde_{\text{tol}} = 10^{-8}$ on the 55^3 grid using TRILINOS AMG on Savanna using 4 nodes, 2 MPI processors per node, 10 open MP threads per process and 126 GiB of memory per node.

μ_1	BFGS steps	PDE calls	Gradient calls	Inner products	Norm calls	F_R value	F_d value
10^{-2}	6	10	8	62	14	0.0128	0.0528
10^{-3}	15	24	18	288	32	0.0149	0.0173
10^{-4}	36	80	40	1402	74	0.0058	0.0017
10^{-5}	91	199	98	3940	184	0.0010	0.0002

Table 9. Example 5.1: Varying μ_1, with $m_{\text{tol}} = 10^{-4}$ and $pde_{\text{tol}} = 10^{-8}$ on the 99^3 grid using TRILINOS AMG on Savanna using 4 nodes, 2 MPI processors per node, 10 open MP threads per process and 126 GiB of memory per node.

μ_1	BFGS steps	PDE calls	Gradient calls	Inner Products	Norm calls	F_R value	F_d value
10^{-2}	5	9	7	47	12	0.0052	0.0331
10^{-3}	15	24	17	287	32	0.0081	0.0144
10^{-4}	31	68	36	1088	64	0.0046	0.0018
10^{-5}	91	203	103	3945	184	0.0009	0.0002
10^{-6}	240	537	265	11425	482	0.0001	0.00003

placed on the ground surface in an 8 × 8 grid with 2m spacing. Potentials were recorded for 576 applied dipoles (less than the 2016 available). These measured potentials must be used in dipole form because one electrode must be used as a reference voltage. Instead of solving for each of the 576 dipole the forward problems (7) and the corresponding 576 adjoint problems (25) with multiple load dipoles, we solved a unit load at each of the electrodes (64 forward problems for each property function) and then used superposition

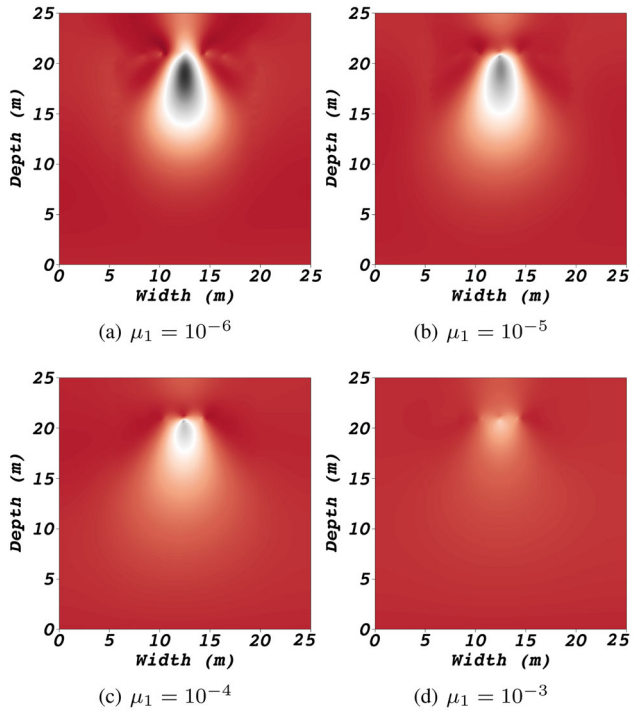


Figure 6. Example 5.1: Varying Tikhonov regularization factor μ_1 for the 99^3 grid with $m_{\text{tol}} = 10^{-4}$ and $pde_{\text{tol}} = 10^{-8}$ on Savanna. The original input and legend are the same as for the 51^3 grid in Figs 4(g) and (h), respectively.

Table 10. Example 5.2: Varying μ_1 with $m_{\text{tol}} = 10^{-3}$ and $\sigma_0 = 0.006$. Initial data misfit is $F_d = 0.372$.

μ_1	BFGS	PDE calls	grad calls	inner prods	norm calls	F_R value	F_d value
10^{-1}	2	68	3	13	6	0.049	0.244
10^{-2}	4	73	16	44	10	0.034	0.181
10^{-3}	10	92	21	151	22	0.030	0.108
10^{-4}	19	160	50	468	40	0.037	0.074

Table 11. Example 5.2: Varying μ_1 with $m_{\text{tol}} = 10^{-3}$ and $\sigma_0 = 0.005$. Initial data misfit is $F_d = 0.245$.

μ_1	BFGS	PDE calls	Grad calls	Enner prods	Norm calls	F_R value	F_d value
10^{-1}	1	40	2	6	4	0.008	0.230
10^{-2}	3	47	4	22	8	0.034	0.169
10^{-3}	10	113	38	168	22	0.034	0.104
10^{-4}	20	162	53	513	42	0.026	0.044

to obtain both the forward and adjoint potentials for each loading dipole, see Appendix A for details.

The mesh was created using *gmsh* (Geuzaine & Remacle 2009) over the computational domain $[-40\text{m}, 40\text{m}]^2 \times [-40\text{m}, 0]$ with FEM nodes located at electrodes. It had 40 209 vertices and 248 521 elements. Attractor nodes in *gmsh* were used to refine the mesh near the electrodes, see Fig. 1. Every possible dipole measurement was used in the inversion.

For $\sigma_0 = 0.006$ the initial misfit error was 0.372. This reduced to 0.245 for $\sigma_0 = 0.005$, see Tables 10 and 11. The smaller the regularization factor μ_1 the smaller the final data misfit. The conductivity maps of the ground surface are shown in Fig. 7. It is clear that increasing μ_1 results in a smoother conductivity, but the general shape of the result remains similar. The high resistivity regions are attributed to the partially collapsed nesting burrows of the wedge-tailed shearwater bird. The variation of conductivity with respect to depth for $\mu_1 = 10^{-4}$ is shown in the sequence of depth results in Fig. 8.

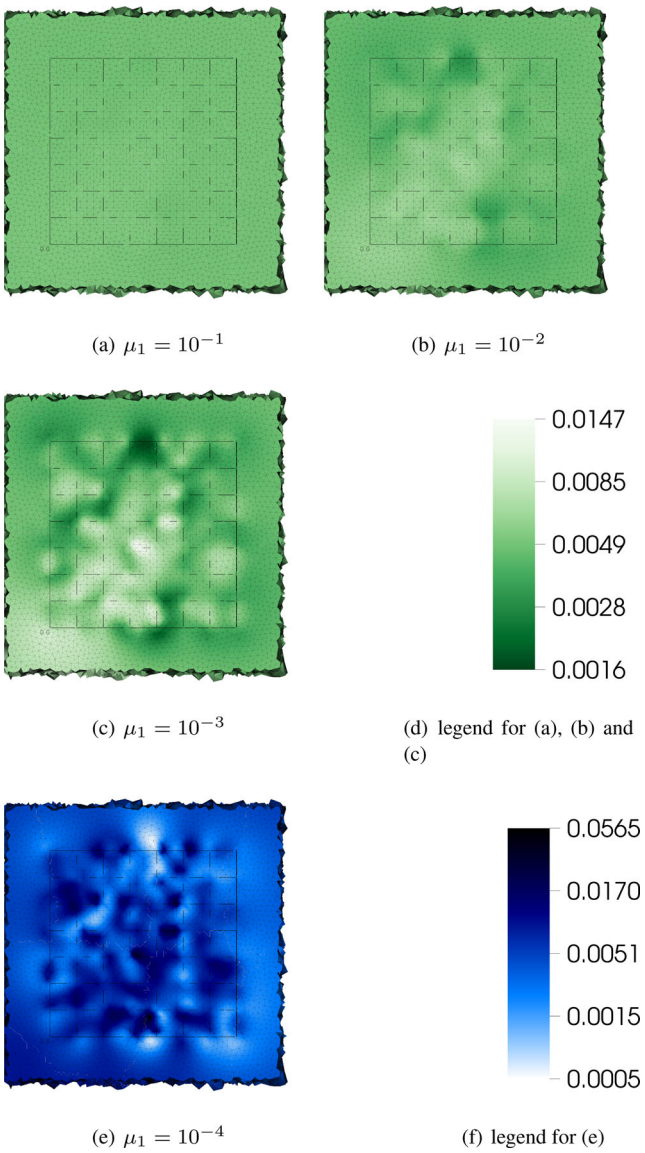


Figure 7. Example 5.2: Heron Island result for conductivity, varying μ_1, with $m_{\text{tol}} = 10^{-4}$ and $pde_{\text{tol}} = 10^{-8}$. A different scale was used for $\mu_1 = 10^{-4}$.

6 CONCLUSION

In the paper we have discussed the application of the quasi-Newton scheme BFGS to solve the ERT inversion problem. The iteration scheme is applied in a function space rather than in Euclidean space. A key advantage of the approach is that it avoids the assemblage of a dense Hessian matrix at the possible cost of additional PDE solves for the adjoint problem. Convergence of the BFGS scheme is accelerated using an AMG-based preconditioner applied to the Hessian of the smoothing term. Taking advantage of the superposition principal we have reduced the number of forward and adjoint PDEs to be solved to the number of loading electrodes when electrodes act as both sources and receivers. As a consequence adding extra measurements per loading array does not increase the number of PDEs to be solved and we can easily treat comprehensive 3-D recording arrays.

The precondition for BFGS could potentially be improved by incorporating the Hessian of the misfit term or an approximation thereof. This could be constructed using the all-at-once approach, see (Haber & Ascher 2001). However, computational costs would be significantly increased as additional solutions of the forward and adjoint problems are required in its construction and it is not clear if this really improves overall computing time. As already highlighted the BFGS iteration is independent to the underlying spatial discretization which allows for a change of the mesh during the iteration process as long as values of integrals are preserved. In particular, this allows a coarse mesh to be used initially, adapted or globally refined as integration progresses without the need to restart the iteration. We will investigate this approach in our future work.

ACKNOWLEDGEMENTS

This work has been supported by the AuScope National Collaborative Research Infrastructure Strategy of the Australian Commonwealth. The authors would also like to thank Adrien Guyot and Harald Hofmann from The University of Queensland for their support to collect the ERT data on Heron Island.

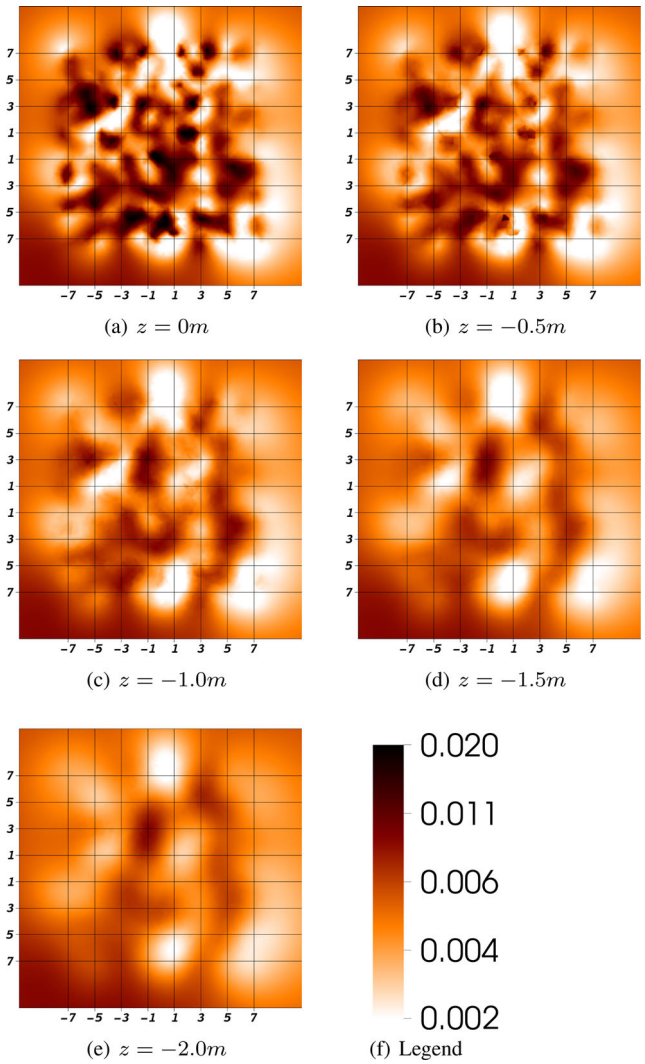


Figure 8. Example 5.1: Conductivity for varying depth with Tikhonov regularization factor $\mu_1 = 10^{-4}$, $m_{\text{tol}} = 10^{-3}$ and $pde_{\text{tol}} = 10^{-8}$.

REFERENCES

Brenner, S.C. & Scott, L.R., 1994. *The Mathematical Theory of Finite Element Methods,* Texts in applied mathematics, Springer-Verlag.

Briggs, W.L., Henson, V.E. & McCormick, S.F., 2000. *A Multigrid Tutorial,* Society for Industrial and Applied Mathematics.

Daily, W., Ramirez, A., Binley, A. & LaBrecque, D., 2012. Electrical Resistance Tomography—theory and practice, in *Near-Surface Geophysics,* pp. 525–550, ed. Butler, D.K., Society of Exploration Geophysicists.

Dey, A. & Morrison, H.F., 1979a. Resistivity modeling for arbitrarily shaped three-dimensional shaped structures, *Geophysics,* **44**(4), 753–780.

Dey, A. & Morrison, H.F., 1979b. Resistivity modelling for arbitrarily shaped two-dimensional structures, *Geophys. Prospect.,* **27**(1), 106–136.

Geuzaine, C. & Remacle, J.-F., 2009. Gmsh: a three-dimensional finite element mesh generator with built-in pre- and post-processing facilities, *Int. J. Numer. Methods Eng.,* **79**(11), 1309–1331.

Gilbarg, D. & Trudinger, N.S., 2001. *Elliptic Partial Differential Equations of Second Order,* Springer-Verlag.

Golub, G.H. & Van Loan, C.F., 1996. *Matrix Computations,* Johns Hopkins studies in the mathematical sciences, The Johns Hopkins University Press.

Gross, L., Altinay, C. & Shaw, S., 2015. Inversion of potential field data using the finite element method on parallel computers, *Comput. Geosci.,* **84,** 61–71.

Haber, E. & Ascher, U.M., 2001. Preconditioned all-at-once methods for large, sparse parameter estimation problems, *Inverse Probl.,* **17,** 1847–1864.

Heroux, M.A. *et al.*, 2005. An overview of the Trilinos project, *ACM Trans. Math. Softw.,* **31**(3), 397–423.

Jell, J.S. & Webb, G.E., 2012. Geology of Heron Island and Adjacent Reefs, Great Barrier Reef, Australia, *EPISODES,* **35**(1, SI), 110–119.

LaBrecque, D.J., Sharpe, R., Wood, T. & Heath, G., 2004. Small-scale electrical resistivity tomography of wet fractured rocks, *Ground Water,* **42**(1), 111–118.

Lamichhane, B.P. & Gross, L., 2017. Inversion of geophysical potential field data using the finite element method, *Inverse Probl.,* **33**(12), doi:10.1088/1361-6420/aa8cb0.

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

Li, Y. & Spitzer, K., 2002. Three-dimensional DC resistivity forward modelling using finite elements in comparison with finite-difference solutions, *Geophys. J. Int.,* **151**(3), 924.

Loke, M., Wilkinson, P. & Chambers, J., 2010. Fast computation of optimized electrode arrays for 2D resistivity surveys, *Comput. Geosci.,* **36**(11), 1414–1426.

Loke, M., Wilkinson, P., Chambers, J., Uhlemann, S. & Sorensen, J., 2015. Optimized arrays for 2-D resistivity survey lines with a large number of electrodes, *J. Appl. Geophys.,* **112,** 136–146.

Lowry, T., Allen, M. & Shrive, P., 1989. Singularity Removal: A refinement of resistivity modeling techniques, *Geophysics,* **54**(6), 766–774.

Moucha, R. & Bailey, R.C., 2004. An accurate and robust multigrid algorithm for 2d forward resistivity modelling, *Geophys. Prospect.,* **52**(3), 197–212.

Nocedal, J. & Wright, S.J., 2006. *Numerical Optimization,* Springer series in operations research and financial engineering, 2nd edn, Springer.

Noel, M. & Xu, B., 1991. Archaeological investigation by electrical resistivity tomography: a preliminary study, *Geophys. J. Int.,* **107**(1), 95.

Pan, K. & Tang, J., 2014. 2.5-D and 3-D DC resistivity modelling using an extrapolation cascadic multigrid method, *Geophys. J. Int.,* **197**(3), 1459–1470.

Pidlisecky, A., Haber, E. & Knight, R., 2007. RESINVM3D: A 3D resistivity inversion package, *Geophysics,* **72**(2), H1–H10.

Pidlisecky, A., Moran, T., Hansen, B. & Knight, R., 2016. Electrical Resistivity Imaging of Seawater Intrusion into the Monterey Bay Aquifer System, *Ground Water,* **54**(2), 255–261.

Prokopenko, A., Hu, J.J., Wiesner, T.A., Siefert, C.M. & Tuminaro, R.S., 2014. *MueLu User's Guide 1.0, Tech. Rep. SAND2014-18874*, Sandia National Labs.

Rücker, C., Günther, T. & Spitzer, K., 2006a. Three-dimensional modelling and inversion of dc resistivity data incorporating topography – I. Modelling, *Geophys. J. Int.,* **166,** 495–505.

Rucker, D.F., Fink, J.B. & Loke, M.H., 2011. Environmental monitoring of leaks using time-lapsed long electrode electrical resistivity, *J. Appl. Geophys.,* **74**(4), 242–254.

Rücker, T., Gunther, C. & Spitzer, K., 2006b. Three-dimensional modelling and inversion of dc resistivity data incorporating topography – II. Inversion, *Geophys. J. Int.,* **166,** 506–517.

Schaa, R., Gross, L. & Du Plessis, J., 2016. PDE-based geophysical modelling using finite elements: examples from 3D resistivity and 2D magnetotellurics, *J. Geophys. Eng.,* **13,** S59–S73.

Stuben, K., 2001. A review of algebraic multigrid, *J. Comput. Appl. Math.,* **128**(1–2), 281–309.

Telford, W.M., Geldart, L.P. & Sheriff, R.E., 1990. *Applied Geophysics,* Cambridge Univ. Press, pp. 770.

Tikhonov, A.N. & Arsenin, V.Y., 1977. *Methods for Solving Ill-posed Problems,* Wiley.

Tuminaro, R.S., 2000. Parallel smoothed aggregation multigrid: aggregation strategies on massively parallel machines, in *Proceedings of the 2000 ACM/IEEE Conference on Supercomputing,* SC '00, IEEE Computer Society, Washington, DC, USA.

Vaněk, P., Mandel, J. & Brezina, M., 1996. Algebraic multigrid by smoothed aggregation for second and fourth order elliptic problems, *Computing,* **56**(3), 179–196.

Zhdanov, M.S., 2002. *Geophysical Inverse Theory and Regularization Problems,* vol. 36 of ods in Geochemistry and Geophysics, Elsevier.

Zhou, B., Greenhalgh, M. & Greenhalgh, S.A., 2009. 2.5-D/3-D resistivity modelling in anisotropic media using Gaussian quadrature grids, *Geophys. J. Int.,* **176**(1), 63–80.

Zienkiewicz, O.C., Taylor, R.L. & Zhu, J.Z., 2013. *The Finite Element Method: Its Basis and Fundamentals,* 7th edn, Elsevier.

APPENDIX A: SUPERPOSITION

The superposition principle (Gilbarg & Trudinger 2001) can be applied to second order PDEs. A significant reduction in computation time can be achieved for solving both the forward (3) and adjoint (20) or (25) PDEs by computing potentials for pole loads and using superposition to obtain dipole potentials. To compute the minimum of cost function (11) we need to compute its gradient (12). It is the computation of the product term in (24), one of the terms in (12), that can be simplified using superposition.

Consider the case for n electrodes used both as loading and measuring electrodes. Let $\mathbf{x}_p$ denote the location of electrode p for $p \in [1, n]$. For a given conductivity, σ, define unit pole PDEs for each electrode

$$-\nabla \cdot \sigma \nabla \phi_p = \delta(\mathbf{x} - \mathbf{x}_p), \quad \forall p \in [1, n]. \tag{A1}$$

There are four different standard arrays: pole–pole, dipole–pole, pole–dipole and dipole–dipole. To improve clarity in the following discussion, we have changed the notation for subscripts. The colon in the subscript is used to separate loading and measuring electrodes. This means that subscripts $p:r$ indicate pole load at x_p and measurement at x_r, subscripts $pq:r$ indicate dipole load with positive node at x_p and negative node at x_q and measurement at x_r, subscripts $p:rs$ indicate pole load at x_p and measurement dipole with positive node at x_r and negative node at x_s and finally, subscripts $pq:rs$ indicate dipole load with positive node at x_p and negative node at x_r and dipole measurement with positive node at x_r and negative node at x_s. These subscripts are used for defects d, weighting factors w and measurements V.

A1 Pole–pole

In this case the defect at electrode $\mathbf{x}_r$ for load at $\mathbf{x}_p$ is defined

$$d_{p:r} = \phi_p(\mathbf{x}_r) - V_{p:r}$$

where $\phi_p(\mathbf{x}_r)$ is the computed potential at $\mathbf{x}_r$ for load at $\mathbf{x}_p$ and $V_{p:r}$ is the corresponding measured potential. The pole–pole adjoint equation is

$$-\nabla \cdot \sigma \nabla \phi^*_p = \sum_{\substack{r=1 \\ r \neq p}}^{n} w_{p:r} d_{p:r} \delta(\mathbf{x} - \mathbf{x}_r),$$

where $w_{p:r}$ is the weighting for the defect measurement $d_{p:r}$. Clearly, weighting factors are symmetric, $w_{p:r} = w_{r,p}$. It follows from eq. (A1) that the adjoint potential is

$$\phi^*_p = \sum_{\substack{r=1 \\ r \neq p}}^{n} w_{p:r} d_{p:r} \phi_r.$$

Consequently, the product term in (24) for all possible pole measurements is

$$\sum_p \nabla \phi^*_p \cdot \nabla \phi_p = \sum_{p=1}^{n} \sum_{\substack{r=1 \\ r \neq p}}^{n} w_{p:r} d_{p:r} \nabla \phi_r \cdot \nabla \phi_p,$$

which simplifies to

$$\sum_p \nabla \phi^*_p \cdot \nabla \phi_p = \sum_{p=1}^{n-1} \sum_{r=p+1}^{n} \alpha_{pr} \nabla \phi_r \cdot \nabla \phi_p, \tag{A2}$$

where

$$\alpha_{pr} = w_{p:r}(d_{p:r} + d_{r:p}) \tag{A3}$$

Using (A2) simplifies the computation of the product term in (24) for cost function gradient (12).

A2 Dipole–pole

For dipole loads, with positive pole at $\mathbf{x}_p$ and negative pole at $\mathbf{x}_q$, the potential equation is

$$-\nabla \cdot \sigma \nabla \phi_{pq} = \delta(\mathbf{x} - \mathbf{x}_p) - \delta(\mathbf{x} - \mathbf{x}_q) \tag{A4}$$

and we immediately obtain

$$\phi_{pq} = (\phi_p - \phi_q), \tag{A5}$$

with $p \in [1, n-1]$, $q \in [p+1, n]$. There are $\frac{n(n-1)}{2}$ possible loading dipoles and the dipole-pole defect for dipole (A4) at $\mathbf{x}_r$ is defined

$$d_{pq:r} = \phi_{pq}(\mathbf{x}_r) - V_{pq:r}.$$

The corresponding adjoint PDE for pole measurements taken at all non-loading electrodes is

$$-\nabla \cdot \sigma \nabla \phi^*_{pq} = \sum_{\substack{r=1 \\ r \neq p,q}}^{n} w_{pq:r} d_{pq:r} \delta(\mathbf{x} - \mathbf{x}_r),$$

where $w_{pq:r}$ is the weighting for defect measurement $d_{pq:r}$. Hence, the adjoint potential is

$$\phi^*_{pq} = \sum_{\substack{r=1 \\ r \neq p,q}}^{n} w_{pq:r} d_{pq:r} \phi_r. \tag{A6}$$

The product term in (25) for dipole loading and pole defects using (A5) and (A6) can be written

$$\sum_{p=1}^{n-1} \sum_{q=p}^{n} \nabla \phi^*_{pq} \cdot \nabla \phi_{pq} = \sum_{p=1}^{n-1} \sum_{q=p+1}^{n} \sum_{\substack{r=1 \\ r \neq p,q}}^{n} (w_{pq:r} d_{pq:r}) \nabla \phi_r \cdot (\nabla \phi_p - \nabla \phi_q). \tag{A7}$$

To simplify the expression for the product term, it is easy to see that the dipole defect is antisymmetric in p and q

$$d_{qp:r} = -d_{pq:r},$$

and weighting factors are symmetric in p and q,

$$w_{qp:r} = w_{pq:r}.$$

So (A7) simplifies to

$$\sum_{p=1}^{n-1} \sum_{q=p}^{n} \nabla \phi^*_{pq} \cdot \nabla \phi_{pq} = \sum_{p=1}^{n-1} \sum_{r=p+1}^{n} \alpha_{pr} \nabla \phi_r \cdot \nabla \phi_p \tag{A8}$$

where

$$\alpha_{pr} = \sum_{\substack{q=1 \\ q \neq p,r}}^{n} (w_{pq:r} d_{pq:r} + w_{rq:p} d_{rq:p}).$$

Using (A8) simplifies the computation of the product term in (24) for cost function gradient (12).

A3 Pole–dipole

In this case the dipole defect at electrodes $\mathbf{x}_r$ and $\mathbf{x}_s$ for load at $\mathbf{x}_p$ is defined

$$d_{p:rs} = \phi_p(\mathbf{x}_r) - \phi_p(\mathbf{x}_s) - (V_{p:r} - V_{p:s}) = d_{p:r} - d_{p:s}$$

where $\phi_p(\mathbf{x}_r)$ is the computed potential at $\mathbf{x}_r$ for load at $\mathbf{x}_p$, $\phi_p(\mathbf{x}_s)$ is the computed potential at $\mathbf{x}_s$ for load at $\mathbf{x}_p$ and $V_{p:r}$ and $V_{p:s}$ are the corresponding measured potentials. The pole-dipole adjoint equation is

$$-\nabla \cdot \sigma \nabla \phi_p^* = \sum_{\substack{r=1 \\ r\neq p}}^{n-1} \sum_{\substack{s=r+1 \\ s\neq p}}^{n} w_{p:rs} d_{p:rs} (\delta(\mathbf{x} - \mathbf{x}_r) - \delta(\mathbf{x} - \mathbf{x}_s)),$$

where $w_{p:rs}$ is the weighting factor for load at $\mathbf{x}_p$ and dipole measurements at $\mathbf{x}_r$ and $\mathbf{x}_s$. Consequently, the adjoint potential is given by

$$\phi_p^* = \sum_{\substack{r=1 \\ r\neq p}}^{n-1} \sum_{\substack{s=r+1 \\ s\neq p}}^{n} w_{p:rs} d_{p:rs} (\phi_r - \phi_s).$$

It follows that the product term in (24) for all possible pole measurements is

$$\sum_p \nabla \phi_p^* \cdot \nabla \phi_p = \sum_{p=1}^{n} \sum_{\substack{r=1 \\ r\neq p}}^{n-1} \sum_{\substack{s=r+1 \\ s\neq p}}^{n} w_{p:rs} d_{p:rs} (\nabla \phi_r - \nabla \phi_s) \cdot \nabla \phi_p, \quad \text{(A9)}$$

It follows that (A9) simplifies to

$$\sum_p \nabla \phi_p^* \cdot \nabla \phi_p = \sum_{p=1}^{n-1} \sum_{r=p+1}^{n} \alpha_{pr} \nabla \phi_r \cdot \nabla \phi_p. \quad \text{(A10)}$$

where

$$\alpha_{pr} = \sum_{\substack{s=1 \\ s\neq p,r}}^{n} \left(w_{p:rs} (d_{p:r} - d_{p:s}) + w_{r:ps} (d_{r:p} - d_{r:s}) \right)$$

Using (A10) simplifies the computation of the product term in (24) for cost function gradient (12).

A4 Dipole–dipole

The final standard loading and measuring case is dipole–dipole as used for the Heron Island survey. The forward PDE is as for the dipole-pole case (A4) and the adjoint potential equation is

$$-\nabla \cdot \sigma \nabla \phi_{pq}^* = \sum_{\substack{r=1 \\ r\neq p}}^{n-1} \sum_{\substack{s=r+1 \\ s\neq p}}^{n} w_{pq:rs} d_{pq:rs} (\delta(\mathbf{x} - \mathbf{x}_r) - \delta(\mathbf{x} - \mathbf{x}_s)),$$

with weighting $w_{pq:rs}$ and defect $d_{pq:rs}$ is the dipole defect measured at electrodes $\mathbf{x}_r$ and $\mathbf{x}_s$ for dipole load at $\mathbf{x}_p$ and $\mathbf{x}_q$. Clearly,

$$w_{pq:rs} d_{pq:rs} = w_{pq:r} d_{pq:r} - w_{pq:s} d_{pq:s}$$

where $w_{pq:r} d_{pq:r}$ and $w_{pq:s} d_{pq:s}$ are the measured weighted defects at $\mathbf{x}_r$ and $\mathbf{x}_s$ respectively. It follows that the adjoint potential is

$$\phi_{pq}^* = \sum_{\substack{r=1 \\ r\neq p}}^{n-1} \sum_{\substack{s=r+1 \\ s\neq p}}^{n} w_{pq:rs} d_{pq:rs} (\phi_r - \phi_s).$$

The gradient product term in (24) is

$$\nabla \phi_{pq}^* \cdot \nabla \phi_{pq} = \sum_{\substack{r=1 \\ r\neq p,q}}^{n-1} \sum_{\substack{s=r+1 \\ s\neq p,q}}^{n} w_{pq:rs} d_{pq:rs} (\nabla \phi_r - \nabla \phi_s) \cdot (\nabla \phi_p - \nabla \phi_q).$$

Clearly, $d_{pq:r}$ is antisymmetric in p and q,

$$d_{pq:r} = -d_{qp:r},$$

and weights have symmetry

$$w_{pq:rs} = w_{qp:rs} = w_{pq:sr} = w_{qp:sr} = w_{rs:pq}.$$

The product term can be reduced to the form

$$\sum_{p=1}^{n-1} \sum_{q=p}^{n} \nabla \phi_{pq}^* \cdot \nabla \phi_{pq} = \sum_{p=1}^{n-1} \sum_{r=p+1}^{n} \alpha_{pr} \nabla \phi_r \cdot \nabla \phi_p, \quad \text{(A11)}$$

where

$$\alpha_{pr} = \sum_{\substack{q=1 \\ q\neq p,r}}^{n} \sum_{\substack{s=1 \\ s\neq p,r,q}}^{n} \left(w_{pq:rs} d_{pq:rs} + w_{qp:rs} d_{qp:rs} \right)$$

or in terms of defect measured at the node, this is

$$\alpha_{pr} = \sum_{\substack{q=1 \\ q\neq p,r}}^{n} \sum_{\substack{s=1 \\ s\neq p,r,q}}^{n} w_{pq:rs} (d_{pq:r} - d_{pq:s}) + w_{rs:qp} (d_{rs:q} - d_{rs:p}).$$

Using (A11) simplifies the computation of the product term in (24) for cost function gradient (12).