

Computational recipes for electromagnetic inverse problems

Gary D. Egbert and Anna Kelbert

College of Oceanic and Atmospheric Sciences, Oregon State University, 104 COAS Admin Bldg., Corvallis, OR 97331-5503, USA.

E-mail: egbert@coas.oregonstate.edu

Accepted 2011 December 18. Received 2011 December 13; in original form 2011 August 25

SUMMARY

The Jacobian of the non-linear mapping from model parameters to observations is a key component in all gradient-based inversion methods, including variants on Gauss–Newton and non-linear conjugate gradients. Here, we develop a general mathematical framework for Jacobian computations arising in electromagnetic (EM) geophysical inverse problems. Our analysis, which is based on the discrete formulation of the forward problem, divides computations into components (data functionals, forward and adjoint solvers, model parameter mappings), and clarifies dependencies among these elements within realistic numerical inversion codes. To be concrete, we focus much of the specific discussion on 2-D and 3-D magnetotelluric (MT) inverse problems, but our analysis is applicable to a wide range of active and passive source EM methods. The general theory developed here provides the basis for development of a modular system of computer codes for inversion of EM geophysical data, which we summarize at the end of the paper.

Key words: Numerical solutions; Inverse theory; Magnetotelluric; Geomagnetic induction.

1 INTRODUCTION

Over the past decade or so, regularized inversion codes have been developed for a range of three-dimensional (3-D) frequency-domain electromagnetic (EM) induction problems, including magnetotellurics (MT; e.g. Newman & Alumbaugh 2000; Siripunvaraporn *et al.* 2004) global geomagnetic depth sounding (Kelbert *et al.* 2008), and controlled source methods including cross-well imaging (e.g. Alumbaugh & Newman 1997) and marine controlled source EM (CSEM; e.g. Commer & Newman 2008). Generally, these efforts have been based upon minimization of a penalty functional, a sum of data misfit and model norm terms. Several distinct algorithms have been applied to solve the minimization problem, including Gauss–Newton (GN) schemes (Mackie & Madden 1993; Sasaki 2001; Siripunvaraporn *et al.* 2004) and direct gradient-based minimization schemes such as non-linear conjugate gradients (NLG; e.g. Newman & Alumbaugh 2000) or quasi-Newton schemes (e.g. Newman & Boggs 2004; Avdeev & Avdeeva 2009). All of these various applications, and the different inversion algorithms that have been used, share many common elements. Here, we consider these commonalities explicitly, developing a general mathematical framework for frequency-domain EM inverse problems. Through this framework, we provide recipes for adapting previously developed inversion algorithms to new applications and for developing extensions to standard applications (e.g. new data types, model parametrizations and regularization approaches), and a basis for development of more efficient inversion algorithms.

Recently, Pankratov & Kuvshinov (2010) have given a general formulation for calculation of derivatives for 3-D frequency-domain EM problems. A general formalism for derivative calculation is

also central to our development, so in principal there is considerable overlap between their development and what is presented here. However, in contrast to Pankratov & Kuvshinov (2010), we adopt a purely discrete approach, assuming from the outset that the forward problem has been discretized for numerical solution, so that all spaces (EM fields, model parameters and data) are finite dimensional. The penalty functional to be minimized is explicitly taken to be a discrete quadratic form, and derivatives, adjoints, etc. are all derived for this discrete problem. Similarly, we explicitly consider the need to use discrete interpolation operators to simulate the measurement process, and to represent dependence of the discrete model operator on the unknown parameters.

There has been considerable discussion in the ocean data assimilation literature concerning the virtues of discrete versus continuous formulations of inverse problems (e.g. Bennett 2002). Certainly, there are some issues in inverse problems (e.g. regularity and well-posedness) that can only be understood and discussed rigorously through consideration of the problem in continuous form (e.g. Egbert & Bennett 1996). However, for development of actual practical inversion algorithms there are good reasons to focus on the discrete formulation. In particular, only through a direct treatment of the discrete problem can symmetry (with respect to appropriate inner products) of the numerical implementation of adjoints be guaranteed. Furthermore, in discrete form many derivations are trivial, and the steps actually required for computations are often more clearly and explicitly laid out.

Of course, details about Jacobian calculations and discussions of the solution of EM inverse problems in discrete form have been given in many previous publications, both for specific EM methods (e.g. see references earlier), and with some degree of generality

(McGillivray *et al.* 1994; Newman & Hoversten 2000). One motivation for presenting this material again here, with a more abstract formulation and using homogeneous mathematical notation, is to provide a foundation for a system of modular computer codes for frequency-domain EM inverse problems that we have recently developed. We sketch key aspects of this system at the end of this paper, and provide a more detailed description elsewhere. Development of this modular system motivates us to clearly define all of the fundamental objects and methods required for a generic EM inverse problem, and to analyse the dependencies among these objects. This framework for the modular system, which we believe is unique both in its abstraction and completeness, is one of the key results presented in this paper. Another key result which emerges from our analysis concerns the structure of the Jacobian calculations in multifrequency and multitransmitter inverse problems. The rather obvious factorization of the Jacobian into receiver and transmitter components (previously used to improve efficiency in cross-well EM inversion by Newman & Alumbaugh 1997) is a simple consequence of our analysis. A less obvious consequence, which has not to our knowledge been previously noted or made use of, is that computations of sensitivities for problems with multicomponent transfer functions (TFs; e.g. 3-D MT) can also be factored, reducing required computations by a factor of 4 relative to a more naive approach. Our abstract treatment of Jacobian calculations thus provides a basis for developing more efficient computational strategies for specific problems.

This paper is organized as follows: In Section 2, we summarize some common linearized EM inversion methods based on gradient-based minimization of a penalty functional, demonstrating at a coarse level the basic objects used for EM inversion methods. A key component in all methods is the Jacobian of the mapping from model parameters to data; we derive general expressions for this linear operator in Section 3. In Section 4, we consider more explicitly the discretization of the governing differential equations, and the dependence of the discrete equation coefficients on the model parameter. Here, we introduce specific examples of EM inverse problems (2-D and 3-D MT) which we will follow throughout the remaining development. These EM inverse problems are sufficiently different to motivate and illustrate much of the abstraction required of a flexible modular system. With these examples as motivation, we then show in Section 5 how operations with the Jacobian can be factored into reusable components, and we consider how these components depend on each other, and on details of the EM method (e.g. sources and receivers), model parametrization and numerical discretization. In Section 6, we consider more explicitly the form of the Jacobian when there are multiple frequencies and multiple, possibly coupled, source geometries. Some new results on possible computational efficiencies are given here. In Section 7, we provide a brief overview and illustration of the modular system of Fortran 95 computer codes that we have developed based on the framework for general frequency-domain inversion developed in the preceding sections.

2 LINEARIZED EM INVERSION: OVERVIEW

We consider regularized inversion through gradient-based minimization of a penalty functional of the form

$$\Phi(\mathbf{m}, \mathbf{d}) = (\mathbf{d} - \mathbf{f}(\mathbf{m}))^T \mathbf{C}_d^{-1} (\mathbf{d} - \mathbf{f}(\mathbf{m})) + \nu (\mathbf{m} - \mathbf{m}_0)^T \mathbf{C}_m^{-1} (\mathbf{m} - \mathbf{m}_0) \quad (1)$$

to recover, in a stable manner, \mathbf{m} , an M -dimensional Earth's conductivity model parameter vector, which provides an adequate fit to a data vector \mathbf{d} of dimension N_d . In (1), \mathbf{C}_d is the covariance of data errors, $\mathbf{f}(\mathbf{m})$ defines the forward mapping, \mathbf{m}_0 is a prior or first guess model parameter, ν is a trade-off parameter, and \mathbf{C}_m (or more properly $\nu^{-1} \mathbf{C}_m$) defines the model covariance or regularization term. In practice, \mathbf{C}_d is always taken to be diagonal, so by a simple rescaling of the data and forward mapping ($\mathbf{C}_d^{-1/2} \mathbf{d}$, $\mathbf{C}_d^{-1/2} \mathbf{f}$), we may eliminate \mathbf{C}_d^{-1} from the definition of Φ .

The prior model \mathbf{m}_0 and model covariance \mathbf{C}_m can also be eliminated from (1) by the affine linear transformation of the model parameter $\tilde{\mathbf{m}} = \mathbf{C}_m^{-1/2} (\mathbf{m} - \mathbf{m}_0)$, and forward mapping $\tilde{\mathbf{f}}(\tilde{\mathbf{m}}) = \mathbf{f}(\mathbf{C}_m^{1/2} \tilde{\mathbf{m}} + \mathbf{m}_0)$, reducing (1) to

$$\begin{aligned} \Phi(\tilde{\mathbf{m}}, \mathbf{d}) &= (\mathbf{d} - \tilde{\mathbf{f}}(\tilde{\mathbf{m}}))^T (\mathbf{d} - \tilde{\mathbf{f}}(\tilde{\mathbf{m}})) + \nu \tilde{\mathbf{m}}^T \tilde{\mathbf{m}} \\ &= \|\mathbf{d} - \tilde{\mathbf{f}}(\tilde{\mathbf{m}})\|^2 + \nu \|\tilde{\mathbf{m}}\|^2. \end{aligned} \quad (2)$$

After minimizing (2) over $\tilde{\mathbf{m}}$, the model parameter in the untransformed space can be recovered as $\mathbf{m} = \mathbf{C}_m^{1/2} \tilde{\mathbf{m}} + \mathbf{m}_0$. Note that this model space transformation is in fact quite practical if instead of following the usual practice of defining $\mathbf{C}_m^{-1} = \mathbf{D}^T \mathbf{D}$, where \mathbf{D} is a discrete representation of a gradient or higher order derivative operator, the regularization is formulated directly in terms of a smoothing operator (i.e. model covariance) \mathbf{C}_m . It is relatively easy to construct computationally efficient positive definite discrete symmetric smoothing operators for regularization (e.g. Egbert 1994; Siripunvaraporn & Egbert 2000; Chua 2001). Although the resulting covariance matrix \mathbf{C}_m will not generally be sparse and may not be practical to invert, all of the computations required for gradient evaluations and for minimization of the transformed penalty functions require only multiplication by the smoothing operator $\mathbf{C}_m^{1/2}$ (i.e. half of the smoothing of \mathbf{C}_m). It is also straightforward to define model covariances that can be inverted (i.e. so that multiplication by both \mathbf{C}_m and \mathbf{C}_m^{-1} are practical.) In the following, we focus on the simplified 'canonical' penalty functional (2), with tildes omitted.

We begin by summarizing some standard approaches to gradient-based minimization of (2) using a consistent notation. Siripunvaraporn & Egbert (2000), Rodi & Mackie (2001), Newman & Boggs (2004), Avdeev (2005) provide further details and discussion on these and related methods. A key component in all of these linearized search schemes is the $N_d \times M$ Jacobian, or sensitivity matrix, which we denote \mathbf{J} . This gives the derivative of \mathbf{f} with respect to the model parameters, with $J_{ij} = \partial f_i / \partial m_j$. Newman & Alumbaugh (1997); Spitzer (1998); Rodi & Mackie (2001) provide detailed expressions for \mathbf{J} for some specific EM inverse problems, and we will consider the general case extensively below (Section 3).

Search for a minimizer of (2) using \mathbf{J} is iterative, as, for example, in the classical GN method. Let \mathbf{m}_n be the model parameter at the n th iteration, \mathbf{J} the sensitivity matrix evaluated at \mathbf{m}_n and $\mathbf{r} = \mathbf{d} - \mathbf{f}(\mathbf{m}_n)$ the data residual. Then, linearizing the penalty functional in the vicinity of \mathbf{m}_n for small perturbations $\delta \mathbf{m}$ leads to the $M \times M$ system of normal equations

$$(\mathbf{J}^T \mathbf{J} + \nu \mathbf{I}) \delta \mathbf{m} = \mathbf{J}^T \mathbf{r} - \nu \mathbf{m}_n, \quad (3)$$

which can be solved for $\delta \mathbf{m}$ to yield a new trial solution $\mathbf{m}_{n+1} = \mathbf{m}_n + \delta \mathbf{m}$. As discussed in Parker (1994), this basic linearized scheme generally requires some form of step length damping for stability (e.g. a Levenberg–Marquardt approach; Marquardt 1963; Rodi & Mackie 2001).

There are many variants to this basic algorithm. For example, in the Occam approach (Constable *et al.* 1987; Parker 1994), (3) is

rewritten as

$$(\mathbf{J}^T \mathbf{J} + \nu \mathbf{I}) \mathbf{m} = \mathbf{J}^T \hat{\mathbf{d}}, \quad (4)$$

where $\hat{\mathbf{d}} = \mathbf{d} - \mathbf{f}(\mathbf{m}_n) + \mathbf{Jm}_n$. Although \mathbf{m}_{n+1} is obtained directly as the solution to (4) the result is exactly equivalent to solving (3) for the change in the model at step $n + 1$, and adding the result to \mathbf{m}_n . A more substantive difference is that in the Occam scheme step length control is achieved by varying ν , computing a series of trial solutions, and choosing the regularization parameter so that data misfit is minimized. An advantage of this approach is that ν is determined as part of the search process, and at convergence one is assured that the solution attains at least a local minimum of the model norm $\|\mathbf{m}\| = (\mathbf{m}^T \mathbf{m})^{1/2}$, subject to the data fit attained (Parker 1994). The Occam scheme can also be implemented in the data space (Siripunvaraporn & Egbert 2000; Siripunvaraporn *et al.* 2005). The solution to (4) can be written as

$$\mathbf{m}_{n+1} = \mathbf{J}^T \mathbf{b}_n; \quad (\mathbf{J}^T + \nu \mathbf{I}) \mathbf{b}_n = \hat{\mathbf{d}}, \quad (5)$$

as can be verified by substituting (5) into (4) and simplifying. This approach requires solving an $N_d \times N_d$ system of equations in the data space instead of the $M \times M$ model space system of equations (4), and can thus be more efficient if the model is heavily overparametrized.

Computing the full Jacobian \mathbf{J} required for any direct GN algorithm is a very demanding computational task for multidimensional EM problems, since (as we shall see in Section 3) the equivalent of one forward solution is required for each row (or column) of \mathbf{J} . An alternative is to solve the normal eqs (4) or (5) with a memory efficient iterative Krylov-space solver such as conjugate gradients (CG). This requires computation of matrix-vector products such as $[\mathbf{J}^T \mathbf{J} + \nu \mathbf{I}] \mathbf{m}$, which can be accomplished without forming or storing \mathbf{J} at the cost of two forward solutions (e.g. Mackie & Madden 1993). Mackie & Madden (1993), Zhang *et al.* (1995), Newman & Alumbaugh (1997), Rodi & Mackie (2001) and others have used CG to solve (3), whereas Siripunvaraporn & Egbert (2007) have applied the same approach to the corresponding data space equations of (5).

The GN scheme requires solving normal equations derived from a quadratic approximation to (1). Alternatively, the penalty functional can be directly minimized using a gradient-based optimization algorithm such as NLCG (e.g. Rodi & Mackie 2001; Newman & Boggs 2004; Avdeev 2005; Kelbert *et al.* 2008). With this NLCG approach, one must evaluate the gradient of (1) with respect to variations in model parameters \mathbf{m}

$$\left. \frac{\partial \Phi}{\partial \mathbf{m}} \right|_{\mathbf{m}_n} = -2 \mathbf{J}^T \mathbf{r} + 2 \nu \mathbf{m}_n. \quad (6)$$

The gradient is then used to calculate a new ‘conjugate’ search direction in the model space. After minimizing the penalty functional along this direction using a line search which requires at most a few evaluations of the forward operator, the gradient is recomputed. NLCG again uses essentially the same basic computational steps as required for solving the linearized equations (3). In particular, the forward problem must be solved to evaluate $\mathbf{f}(\mathbf{m})$ and the residual \mathbf{r} must be multiplied by \mathbf{J}^T . Quasi-Newton schemes (e.g. Nocedal & Wright 1999; Newman & Boggs 2004; Haber 2005; Avdeev & Avdeeva 2009) provide an alternative approach to NLCG for direct minimization of (1), with similar advantages with regard to storage and computation of the Jacobian, and similar computational requirements.

All of these schemes for minimizing (1) can be abstractly expressed in terms of a small number of basic ‘objects’ (data and

model parameter vectors, \mathbf{d} and \mathbf{m}), and operators (the forward mapping $\mathbf{f}(\mathbf{m})$, multiplication by the corresponding Jacobian \mathbf{J} and its transpose \mathbf{J}^T (and, implicitly, the data and model covariances \mathbf{C}_m and \mathbf{C}_d). Given modular computer codes which implement these basic objects, any of the inversion algorithms outlined here, as well as many variants, are readily implemented. In the next sections, we analyse further the discrete forward problem, and provide a finer grained general formulation of the modules required to implement essentially any linearized inverse scheme for any EM problem. In particular, we provide ‘recipes’ for \mathbf{J} in terms of more basic objects associated with the model parametrization, the forward solver and the numerical simulation of the observation operators.

3 DATA SENSITIVITIES

The EM forward operator $\mathbf{f}(\mathbf{m})$ generally involves two steps: (1) Maxwell’s equations, with conductivity defined by the parameter \mathbf{m} are solved numerically with appropriate boundary conditions and sources; (2) the resulting solution is used to compute predicted data—for example, an electric or magnetic field component, TF or apparent resistivity—at a set of site locations. For the first step, we write the numerical discretization of the frequency-domain EM partial differential equation (PDE) generically as

$$\mathbf{S}_m \mathbf{e} = \mathbf{b}, \quad (7)$$

where the vector \mathbf{b} gives appropriate boundary and forcing terms for the particular EM problem, \mathbf{e} is the N_e -dimensional vector representing the discretized electric and/or magnetic fields (or perhaps potential functions), and \mathbf{S}_m is an $N_e \times N_e$ coefficient matrix which depends on the M -dimensional model parameter \mathbf{m} . We take \mathbf{e} to represent both interior and boundary components of the discrete solution vector, so that any boundary conditions required for the problem are included in \mathbf{b} . The second step then takes the form

$$f_j(\mathbf{m}) = \psi_j(\mathbf{e}(\mathbf{m}), \mathbf{m}), \quad (8)$$

where ψ_j is some generally non-linear, but usually simple, function of the components of \mathbf{e} (and possibly \mathbf{m}).

With this general setup we have, by the chain rule,

$$J_{jk} = \frac{\partial f_j}{\partial m_k} = \sum_l \frac{\partial \psi_j}{\partial e_l} \frac{\partial e_l}{\partial m_k} + \frac{\partial \psi_j}{\partial m_k}. \quad (9)$$

Letting \mathbf{F} , \mathbf{L} , \mathbf{Q} be the partial derivative matrices

$$F_{jk} = \left. \frac{\partial e_l}{\partial m_k} \right|_{\mathbf{m}_0}, \quad L_{jl} = \left. \frac{\partial \psi_j}{\partial e_l} \right|_{\mathbf{e}_0, \mathbf{m}_0}, \quad Q_{jk} = \left. \frac{\partial \psi_j}{\partial m_k} \right|_{\mathbf{e}_0, \mathbf{m}_0}, \quad (10)$$

where \mathbf{e}_0 is the solution to (7) for model parameter \mathbf{m}_0 , the Jacobian at \mathbf{m}_0 can be written in matrix notation as

$$\mathbf{J} = \mathbf{L}\mathbf{F} + \mathbf{Q}. \quad (11)$$

The j th row of \mathbf{L} represents the linearized data functional, which is applied to the perturbation in the EM solution to compute the perturbation in d_j . These row vectors are generally very sparse, supported only on a few nodes surrounding the corresponding data site. When the observation functionals are independent of the model parameters (as they often are) $\mathbf{Q} \equiv \mathbf{0}$. When \mathbf{Q} is required it is also typically sparse, but this depends on the specific nature of the model parametrization. Although, as we show below, derivation of expressions for \mathbf{L} and \mathbf{Q} can be quite involved for realistic EM data functionals, calculation of \mathbf{F} presents the only real computational burden.

To derive a general expression for \mathbf{F} , differentiate (7) at \mathbf{m}_0 with respect to the model parameters \mathbf{m} . We assume that \mathbf{b} is constant, independent of \mathbf{m} , although, as discussed in Appendix A (see also Newman & Boggs 2004) some subtle issues related to this point may arise with specific solution approaches. Then, letting \mathbf{e}_0 be the solution of (7) at \mathbf{m}_0 , and noting that the EM solution \mathbf{e} varies as \mathbf{m} is varied, we obtain

$$\mathbf{S}_{\mathbf{m}_0} \left[\frac{\partial \mathbf{e}}{\partial \mathbf{m}} \right]_{\mathbf{m}=\mathbf{m}_0} = - \frac{\partial}{\partial \mathbf{m}} (\mathbf{S}_{\mathbf{m}} \mathbf{e}_0) \Big|_{\mathbf{m}=\mathbf{m}_0}, \quad (12)$$

or

$$\mathbf{S}_{\mathbf{m}_0} \mathbf{F} = \mathbf{P}. \quad (13)$$

The $N_e \times M$ matrix \mathbf{P} depends on details of both the numerical model implementation and the conductivity parametrization (as discussed later), but is in general inexpensive to calculate, once the solution \mathbf{e}_0 has been computed. Putting together (11) and (13), we obtain an expression for the numerical Jacobian (or sensitivity matrix) \mathbf{J} for general EM problems

$$\mathbf{J} = \mathbf{L} \mathbf{S}_{\mathbf{m}_0}^{-1} \mathbf{P} + \mathbf{Q}. \quad (14)$$

Computing all of \mathbf{J} would appear to require solving the induction equation M times (i.e. applying the inverse operator $\mathbf{S}_{\mathbf{m}_0}^{-1}$ to each of the columns of \mathbf{P} .) However, simply taking the transpose of (14) we obtain

$$\mathbf{J}^T = \mathbf{P}^T [\mathbf{S}_{\mathbf{m}_0}^T]^{-1} \mathbf{L}^T + \mathbf{Q}^T, \quad (15)$$

so the sensitivity matrix can in fact be obtained by solving the transposed discrete EM system N_d times (once for each column of \mathbf{L}^T), the usual ‘reciprocity’ trick for efficient calculation of sensitivities (e.g. Rodi 1976; de Lugao *et al.* 1997). It should also be emphasized that for many of the inversion algorithms described in Section 2, \mathbf{J} is not explicitly calculated. Instead, a series of multiplications of model space vectors by \mathbf{J} and/or data space vectors by \mathbf{J}^T are required. These, in turn, require implementation of the component operators \mathbf{P} , \mathbf{L} , \mathbf{Q} and the solver $\mathbf{S}_{\mathbf{m}_0}^{-1}$, together with the adjoints (or transposes) of these operators.

The EM equations are self-adjoint (except for time reversal) with respect to the usual L^2 inner product (i.e. reciprocity holds). For now leaving aside complications regarding boundary conditions (these are discussed in Appendix B), this implies that on a uniform grid operator $\mathbf{S}_{\mathbf{m}}$ is symmetric. For more general grids, the fact that the EM operator is self-adjoint implies

$$\mathbf{S}_{\mathbf{m}}^T = \mathbf{V} \mathbf{S}_{\mathbf{m}} \mathbf{V}^{-1}, \quad (16)$$

where \mathbf{V} is a diagonal matrix of integration volume elements for the natural discrete representation of the L_2 integral inner product on the model domain (see Appendix B). Eq. (16) implies $\mathbf{S}_{\mathbf{m}}^T \mathbf{V} = \mathbf{V} \mathbf{S}_{\mathbf{m}}$ is a symmetric (though not Hermitian) matrix. It is easier to compute solutions to this symmetrized problem, so solutions to the forward problem are generally computed as $\mathbf{e} = (\mathbf{V} \mathbf{S}_{\mathbf{m}})^{-1} \mathbf{V} \mathbf{b}$ (e.g. see Uyeshima & Schultz 2000). The solution for the adjoint problem can also be written in terms of the symmetrized inverse operator as $\mathbf{e} = (\mathbf{S}_{\mathbf{m}}^T)^{-1} \mathbf{b} = \mathbf{V} (\mathbf{V} \mathbf{S}_{\mathbf{m}})^{-1} \mathbf{b}$; the principal difference from the forward case is thus the order in which multiplication by the diagonal matrix \mathbf{V} and the symmetrized solver are called. In general, the adjoint solver $(\mathbf{S}_{\mathbf{m}}^T)^{-1}$ for EM problems is trivial to implement, once a suitably general forward solver is available.

Before proceeding, two general points require discussion. First, we note that many of the computations in frequency-domain EM problems are most efficiently implemented (and described) using

complex arithmetic, but the model conductivity parameter \mathbf{m} is real. Data might be complex (e.g. in MT a complex impedance, formed as the ratio of electric and magnetic fields) or real (e.g. an apparent resistivity or phase, derived from the MT impedance). As already implicit in our formulation of the penalty functional (1), we formally assume that all data are real, that is, real and imaginary parts of a complex observation are separate elements of the real data vector \mathbf{d} , and that the basic operators \mathbf{f} and \mathbf{J} have been recast as real mappings from model parameter to data vector. However, we will frequently use complex notation and we will often be somewhat casual in moving between real and complex variables. Thus, for example, the frequency domain forward problem (7) will generally be formulated and solved in terms of complex variables. \mathbf{P} will then be a mapping from the real parameter space to the complex space of forcings for the forward problem, whereas \mathbf{L} will be a mapping from a complex, back to a real space. Both \mathbf{P} and \mathbf{L} can be most conveniently represented by complex matrices, with the convention that for \mathbf{L} only the real part of the matrix-vector product is retained. We discuss this more explicitly in Appendix C.

Secondly, in most cases EM data are obtained for a large number of distinct sources, that is, different frequencies and/or different current source geometries. For example, in the case of MT, there are data for two source polarizations and a wide range of frequencies, whereas for controlled source problems there may be a multiplicity of transmitter geometries or locations. Each of these distinct sources, which we refer to in general as ‘transmitters’, requires solving a separate forward problem. In most, but not all, cases these forward problems are decoupled, so the data vector and forward modelling operator can be decomposed into $t = 1, \dots, N_T$ blocks, one for each transmitter

$$\mathbf{d} = \begin{pmatrix} \mathbf{d}_1 \\ \vdots \\ \mathbf{d}_{N_T} \end{pmatrix}, \quad \mathbf{f} = \begin{pmatrix} \mathbf{f}_1 \\ \vdots \\ \mathbf{f}_{N_T} \end{pmatrix}. \quad (17)$$

Here \mathbf{d}_t gives the data associated with a group of ‘receivers’, consisting of possibly multiple components, at multiple locations, all making observations of fields generated by transmitter t . Thus, with multiple (decoupled) transmitters the Jacobian can be partitioned into N_T blocks in the obvious way, and each block can be represented as in (14), so that the full sensitivity matrix can be expressed as

$$\mathbf{J} = \begin{pmatrix} \mathbf{J}_1 \\ \vdots \\ \mathbf{J}_{N_T} \end{pmatrix} = \begin{pmatrix} \mathbf{L}_1 \mathbf{S}_{1,\mathbf{m}}^{-1} \mathbf{P}_1 + \mathbf{Q}_1 \\ \vdots \\ \mathbf{L}_{N_T} \mathbf{S}_{N_T,\mathbf{m}}^{-1} \mathbf{P}_{N_T} + \mathbf{Q}_{N_T} \end{pmatrix}. \quad (18)$$

The matrices \mathbf{P}_t and \mathbf{Q}_t generally depend on the solution for transmitter t . If the transmitter t only specifies the source geometry, the differential operator for the PDE $\mathbf{S}_{t,\mathbf{m}}$ will be independent of the transmitter; however, in general the transmitter will also define the forward problem to solve. An obvious example is the MT case, where the operator depends on frequency. Only \mathbf{L}_t and \mathbf{Q}_t depend on the configuration of receivers; these also in general depend on the forward solution, and thus on transmitter t .

There is an important complication to the simple form of (18), most clearly illustrated by the case of 3-D MT. Here, evaluation of the forward operator for an impedance tensor element requires solutions for the pair of transmitters associated with two uniform source polarizations. Thus, for 3-D MT, the rows of the Jacobian corresponding to a single frequency are formed from components for two transmitters, corresponding to N–S and E–W polarized uniform sources, coupled through the linearized measurement operators

\mathbf{L} and \mathbf{Q} ,

$$\mathbf{J} = \sum_{t=1}^2 [\mathbf{L}_t \mathbf{S}_m^{-1} \mathbf{P}_t + \mathbf{Q}_t]. \quad (19)$$

The same complication would arise for any plane wave source TF (e.g. vertical, or intersite magnetic). Other examples of multiple polarization EM inverse problems can be imagined, for example, allowing for a combination of horizontal spatial gradients and uniform sources (e.g. Egbert 2002; Schmücker 2004; Semenov & Shuman 2009) would require allowing for five coupled sources. Pankratov & Kuvshinov (2010) discuss the general multiple polarization problem from a theoretical perspective, although to our knowledge, no actual applications of the theory to inversion of real data sets have yet been reported, beyond the standard two polarization uniform source case.

We return to the coupled multiple polarization case in Section 5.2, where we discuss measurement operators in more detail. Then, in Section 6, we consider the general multiple transmitter case further, and show more explicitly how the source and receiver configuration can result in special structure for the Jacobian, which can be exploited to improve computational efficiency. For the next few sections, we focus on the simpler case of a single transmitter, as we develop the basic building blocks for more complex and realistic problems.

4 DISCRETIZATION OF THE FORWARD PROBLEM

To derive more explicit expressions for the operators \mathbf{L} , \mathbf{P} and \mathbf{Q} , and hence for the full Jacobian, more specific assumptions about the numerical implementation of the forward problem (7) are required. To motivate the general development, we consider as examples two specific cases in detail: inversion of 2-D and 3-D MT data. We discuss most explicitly a finite difference (FD) modelling approach, though most of the results obtained are more broadly applicable.

Numerical schemes for solving Maxwell's equations are often most elegantly formulated in terms of a pair of vector fields defined on conjugate grids. For example, the space of primary fields which we denote as \mathcal{S}_p may represent the electric fields, whereas the space of dual fields, denoted \mathcal{S}_D , represents the magnetic fields. Even when the coupled first-order system (i.e. Maxwell's equations) is reduced to a second-order equation involving only the primary field, it is worthwhile to explicitly consider the dual field also. Most obviously, in many applications both electric and magnetic field components are required to evaluate the data functionals. Furthermore, depending on the model formulation, the dependence of the discrete PDE operator coefficients on the model parameter can generally be represented most explicitly through a mapping $\pi(\mathbf{m})$ from the model parameter space \mathcal{M} —sometimes to \mathcal{S}_p , but in other cases to \mathcal{S}_D , and a general treatment should allow for both cases. Boundary conditions are of course a critical part of the formulation of the forward problem. These are included implicitly in our generic formula of the forward problem (7). In the following, we omit technical details concerning implementation of boundary conditions, leaving discussion of these issues to Appendix B.

As a first illustration, we consider FD modelling of the 3-D quasi-static Maxwell's equations appropriate for MT. In the frequency domain (assuming a time dependence of $e^{i\omega t}$), the magnetic fields can be eliminated, resulting in a second-order elliptic system of PDEs in terms of the electric fields alone,

$$\nabla \times \nabla \times \mathbf{E} + i\omega\mu\sigma\mathbf{E} = 0, \quad (20)$$

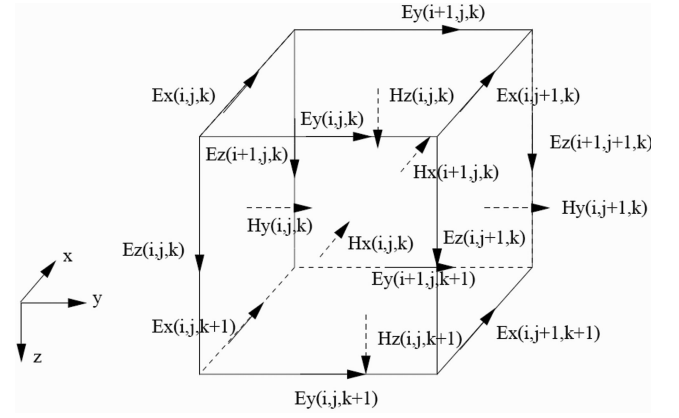


Figure 1. Staggered finite difference grid for the 3-D MT forward problem. Electric field components defined on cell edges are the primary EM field component, which the PDE is formulated in terms of. The magnetic field components can be defined naturally on the cell faces; these are the secondary EM field in this numerical formulation.

where ω is the angular frequency, μ is magnetic permeability and σ is electrical conductivity, with the tangential components of \mathbf{E} specified on all boundaries. To solve (20) numerically in 3-D, we consider an FD approximation on a staggered grid of dimension $N_x \times N_y \times N_z$, as illustrated in Fig. 1 (e.g. Yee 1966; Smith 1996; Siripunvaraporn *et al.* 2002). In the staggered grid formulation, the discretized electric field vector components are defined on cell edges (Fig. 1). In our terminology, the primary field space \mathcal{S}_p is the space of such finite-dimensional cell edge vector fields. A typical element will be denoted by \mathbf{e} . As illustrated in Fig. 1, the magnetic fields, which in continuous form satisfy $\mathbf{H} = (-i\omega\mu)^{-1} \nabla \times \mathbf{E}$, are naturally defined on the discrete grid of cell faces. The dual-field space \mathcal{S}_D is thus the space of discrete vector fields defined on faces. A typical element of this space will be denoted by \mathbf{h} .

In the staggered grid FD discretization used for (20), the discrete magnetic and electric fields are related via

$$\mathbf{h} = (-i\omega\mu)^{-1} \mathbf{C} \mathbf{e}, \quad (21)$$

where $\mathbf{C} : \mathcal{S}_p \mapsto \mathcal{S}_D$ is the discrete approximation of the curl of cell edge vectors, and (20) can be expressed in its discrete form as

$$[\mathbf{C}^\dagger \mathbf{C} + \text{diag}(i\omega\mu\sigma(\mathbf{m}))]\mathbf{e} = 0. \quad (22)$$

Here, $\text{diag}(\mathbf{v})$ denotes a diagonal matrix with the components of the vector \mathbf{v} on the diagonal, and $\mathbf{C}^\dagger : \mathcal{S}_D \mapsto \mathcal{S}_p$ is the discrete curl mapping interior cell face vectors to interior cell edges. As the notation indicates this operator is the adjoint of \mathbf{C} , relative to appropriate (i.e. volume weighted) inner products on the spaces \mathcal{S}_D and \mathcal{S}_p . Although \mathbf{e} is the full solution vector (including boundary components), (22) provides equations only for the interior nodes. Additional equations are required to constrain \mathbf{e} on the boundary, and to complete specification of the discrete forward operator \mathbf{S}_m . These details, and further discussion of \mathbf{C} and its adjoint, are provided in Appendix B. The key point here is that the dependence of the operator coefficients on the model parameter (which we take to be an element of some finite-dimensional space \mathcal{M}) is made explicit through the mapping $\sigma : \mathcal{M} \mapsto \mathcal{S}_p$ in (22).

The 3-D EM induction forward problem can also be formulated in terms of magnetic fields

$$\nabla \times \rho \nabla \times \mathbf{H} + i\omega\mu\mathbf{H} = 0, \quad (23)$$

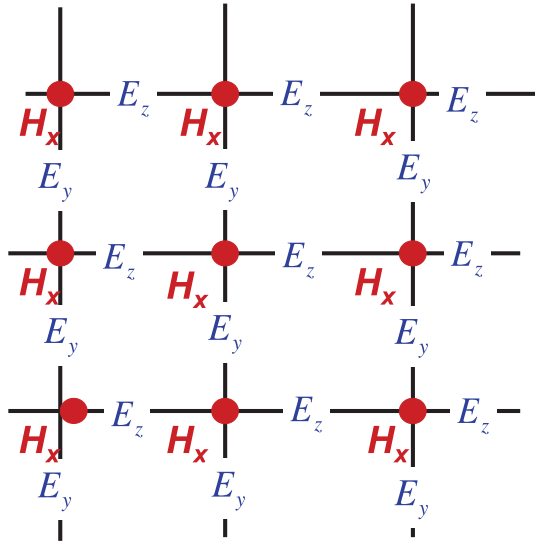


Figure 2. Finite difference grid for the 2-D MT TM mode forward problem. The scalar H_x magnetic field, defined on 2-D cell corners is the primary field. The secondary field components are E_y and E_z , defined on vertical and horizontal cell edges, respectively.

where ρ is electrical resistivity, now with the tangential component of the magnetic fields specified on boundaries. With this formulation (e.g. Mackie *et al.* 1994; Uyeshima & Schultz 2000), \mathbf{H} would be the primary field, and the electric field $\mathbf{E} = \rho \nabla \times \mathbf{H}$ would be the dual field. Using an analogous staggered grid FD discretization, with magnetic field components defined on cell edges, and electric field components defined on cell faces, the discrete induction equation now takes the form

$$[\mathbf{C}^\dagger \text{diag}(\rho(\mathbf{m}))\mathbf{C} + \text{diag}(i\omega\mu)]\mathbf{e} = 0. \quad (24)$$

In this case, the dependence of the coefficients on model parameter \mathbf{m} is made explicit through the mapping to the dual-field space $\rho : \mathcal{M} \mapsto \mathcal{S}_D$. Note that for both the electric and magnetic field formulations \mathbf{e} represents the primary field, and \mathbf{h} the dual field. Thus, in (24), \mathbf{e} represents the discrete magnetic field \mathbf{H} , and \mathbf{h} would represent the discrete electric field.

It is also instructive to consider the 2-D MT inverse problem. Now there are effectively two distinct modelling problems: for transverse electric (TE) and transverse magnetic (TM) modes, with electric and magnetic fields, respectively, parallel to the geologic strike. The TE mode case is essentially identical to the 3-D electric field formulation of (20)–(22). The TM mode case, which is solved in terms of the magnetic field instead of the electric field, is more instructive with regard to generalization. In the TM mode, the magnetic field parallels the geological strike (x) and (23) can be reduced to a scalar PDE in the y - z plane

$$\partial_y \rho \partial_y H_x + \partial_z \rho \partial_z H_x + i\omega\mu H_x = 0, \quad (25)$$

with H_x specified on boundaries.

As for the 3-D problems, for the discrete 2-D problem we can define finite-dimensional spaces of primary (\mathcal{S}_P) and dual (\mathcal{S}_D) EM fields. Now the primary field is H_x , defined on the nodes (corners) of the 2-D grid, and the dual fields are the electric field components E_y and E_z defined on the vertical and horizontal cell edges (Fig. 2). A natural centred FD approximation of (25) can be written in terms of a discrete 2-D gradient operator $\mathbf{G} : \mathcal{S}_P \mapsto \mathcal{S}_D$ and a 2-D divergence

operator $\mathbf{D} : \mathcal{S}_D \mapsto \mathcal{S}_P$. Using $\mathbf{e} \in \mathcal{S}_P$ to denote the primary discrete EM field solution (H_x), we have a more explicit form of (7) for this discrete TM mode implementation,

$$[\mathbf{D} \text{diag}(\rho(\mathbf{m}))\mathbf{G} + i\omega\mu\mathbf{I}]\mathbf{e} = 0, \quad (26)$$

with additional equations again required to specify boundary conditions.

In most other FD or finite volume modelling approaches, for example, with Maxwell's equations cast in terms of vector potentials, similar (although potentially more complicated) sets of conjugate spaces can be defined, the differential operator can be decomposed into discrete approximations to first-order linear differential operators which map between conjugate grids, and the dependence of discrete operator coefficients on an abstract model parameter space can be described explicitly through a mapping $\pi : \mathcal{M} \mapsto \mathcal{S}_{P,D}$. Finite-element approaches to EM modelling will result in similar structures. For example, the space of linear edge elements (or more properly, the degrees of freedom associated with these elements; Nédélec 1980) can be taken as the primary space, representing the discrete electric field. The natural dual space is then the space of face elements, representing the discrete magnetic field (e.g. Rodrigue & White 2001). The natural model parameter mapping then defines conductivity associated with each edge degree of freedom.

5 COMPONENTS OF THE JACOBIAN

5.1 Matrix \mathbf{P}

We can give an explicit expression for the operator \mathbf{P} , assuming the forward operator \mathbf{S}_m can be written in the general form

$$\mathbf{S}_m \mathbf{e} \equiv \mathbf{S}_0 \mathbf{e} + \mathbf{U}(\pi(\mathbf{m}) \circ \mathbf{V} \mathbf{e}), \quad (27)$$

where \mathbf{S}_0 , \mathbf{U} and \mathbf{V} are some linear operators that do not depend on the model parameter vector \mathbf{m} , $\pi(\mathbf{m})$ is a (possibly non-linear) operator that maps the model parameter space \mathcal{M} to the primary or dual grid, and (\circ) denotes the component-wise multiplication of the two vectors in $\mathcal{S}_{P,D}$ (also known as the Hadamard product). Note that on an FD grid, \mathbf{S}_m (and hence \mathbf{S}_0 and \mathbf{V}) act on a full solution vector that includes both the interior and boundary edges (see Appendix B). All of the examples outlined earlier are special cases of (27), as we will discuss.

Assuming (27) and recalling the definition of \mathbf{P} from (12) and (13), we find

$$\mathbf{P} = -\frac{\partial}{\partial \mathbf{m}} (\mathbf{S}_m \mathbf{e}_0) \Big|_{\mathbf{m}_0} = -\mathbf{U} \left(\frac{\partial \pi}{\partial \mathbf{m}} \Big|_{\mathbf{m}_0} \circ \mathbf{V} \mathbf{e}_0 \right), \quad (28)$$

$$= -\mathbf{U} \left(\mathbf{V} \mathbf{e}_0 \circ \frac{\partial \pi}{\partial \mathbf{m}} \Big|_{\mathbf{m}_0} \right), \quad (29)$$

$$= -\mathbf{U} \text{diag}(\mathbf{V} \mathbf{e}_0) \frac{\partial \pi}{\partial \mathbf{m}} \Big|_{\mathbf{m}_0}. \quad (30)$$

Writing $\Pi_{\mathbf{m}_0}$ for the Jacobian of the (in general, non-linear) model parameter mapping $\pi(\mathbf{m})$ evaluated at the background model parameter \mathbf{m}_0 , we have

$$\mathbf{P} = -\mathbf{U} \text{diag}(\mathbf{V} \mathbf{e}_0) \Pi_{\mathbf{m}_0}, \quad (31)$$

$$\mathbf{P}^T = -\Pi_{\mathbf{m}_0}^T \text{diag}(\mathbf{V} \mathbf{e}_0) \mathbf{U}^T. \quad (32)$$

Note that only the operator $\Pi_{\mathbf{m}}$ depends on the details of the model parametrization—the other terms depend only on the numerical discretization of the governing equations. Eqs (31) and (32) provide broadly applicable recipes for implementation of the operators \mathbf{P} and \mathbf{P}^T , as illustrated in the following examples. If the dependence of the forward operator on the model parameter cannot be cast as a special case of (27), similar formal steps could almost certainly be used to derive appropriate expressions for these operators.

5.1.1 Example: 2-D MT

For the 2-D TM problem (26), the PDE coefficients depend on the model parameters through $\rho : \mathcal{M} \mapsto \mathcal{S}_D$, that is, the resistivity $\rho(\mathbf{m})$ defined on the dual grid, cell edges. To be specific, we consider the simplest model parametrizations, with conductivity or log conductivity for each cell in the numerical grid an independent parameter. From physical considerations, it is most reasonable to compute the required edge resistivities from cell conductivities by first transforming to resistivity, and then computing the area weighted average of resistivities of the two cells on either side of the edge. Representing the averaging operator from 2-D cells to cell sides as \mathbf{W}_{TM} , and letting $(\mathbf{m})^{-1}$ denote the component-wise inverse of the model parameter vector, we then have

$$\rho(\mathbf{m}) = \mathbf{W}_{\text{TM}}(\mathbf{m})^{-1}, \quad (33)$$

$$\rho(\mathbf{m}) = \mathbf{W}_{\text{TM}} \exp(-\mathbf{m}), \quad (34)$$

for linear and log conductivity, respectively. The model operator of (26) can be cast in the general form of (27) with the identifications $\mathbf{S}_0 \equiv -i\omega\mu\mathbf{I}$, $\mathbf{U} \equiv \mathbf{D}$, $\mathbf{V} \equiv \mathbf{G}$ and $\pi(\mathbf{m}) \equiv \rho(\mathbf{m})$, where \mathbf{D} and \mathbf{G} are the discrete 2-D divergence and gradient operators defined in Section 4. Thus, we obtain the expressions for \mathbf{P} and \mathbf{P}^T in the 2-D TM mode case.

$$\mathbf{P} = -\mathbf{D} \text{diag}(\mathbf{G}\mathbf{e}_0) \Pi_{\mathbf{m}_0}, \quad (35)$$

$$\mathbf{P}^T = -\Pi_{\mathbf{m}_0}^T \text{diag}(\mathbf{G}\mathbf{e}_0) \mathbf{D}^T, \quad (36)$$

where $\Pi_{\mathbf{m}_0} = -\mathbf{W}_{\text{TM}}[\text{diag}(\mathbf{m}_0)]^{-2}$ for the parametrization in terms of linear conductivity, or $\Pi_{\mathbf{m}_0} = -\mathbf{W}_{\text{TM}}[\text{diag}(\exp(-\mathbf{m}_0))]$ for log conductivity.

5.1.2 Example: 3-D MT

We again assume the simplest model parametrization, with conductivity, or the natural logarithm of conductivity, specified independently for each of the $M = N_x N_y N_z$ cells in the numerical grid. The discrete operator of (22) requires conductivity defined on cell edges, where the electric field components are defined. For physical consistency (current should be conserved), the edge conductivities should represent the volume weighted average of the surrounding four cells. Let \mathbf{W} be the $N_e \times M$ matrix representing this weighted averaging operator, a mapping from \mathcal{M} to \mathcal{S}_p . Then, the conductivity parameter mapping is given by $\sigma(\mathbf{m}) = \mathbf{W}\mathbf{m}$ or $\sigma(\mathbf{m}) = \mathbf{W}\exp(\mathbf{m})$, for the cases of linear and log conductivity, respectively.

Eq. (22) can be seen to be a special case of (27) with the identifications $\mathbf{S}_0 \equiv \mathbf{C}^\dagger \mathbf{C}$, $\mathbf{U} \equiv i\omega\mu\mathbf{I}$, $\mathbf{V} \equiv \mathbf{I}$ and $\pi(\mathbf{m}) \equiv \sigma(\mathbf{m})$, and we have

$$\mathbf{P} = \text{diag}(-i\omega\mu\mathbf{e}_0) \Pi_{\mathbf{m}_0}, \quad (37)$$

$$\mathbf{P}^T = \Pi_{\mathbf{m}_0}^T \text{diag}(i\omega\mu\mathbf{e}_0), \quad (38)$$

where $\Pi_{\mathbf{m}_0} = \mathbf{W}$ for linear conductivity, and $\Pi_{\mathbf{m}_0} = \mathbf{W}[\text{diag}(\exp(\mathbf{m}_0))]$ for logarithmic conductivity. Note that the transposes of the averaging operators \mathbf{W} and \mathbf{W}_{TM} represent mappings from cell edges to cells, a weighted sum of contributions from all edges that bound a cell.

5.2 Matrices \mathbf{L} and \mathbf{Q}

We turn now to the matrices \mathbf{L} and \mathbf{Q} , which represent the linearized observation process, as it is applied to the discrete numerical forward solution.

5.2.1 \mathbf{L} : general case

The very simplest sort of EM data is an observation of the primary field at a single location (e.g. $\epsilon = E_y(\mathbf{x})$), which can be represented as a local average of the modelled primary field

$$\epsilon = (\lambda^P)^T \mathbf{e}. \quad (39)$$

Here $\lambda^P \in \mathcal{S}_p$ is a sparse vector of interpolation coefficients, averaging from the discrete primary grid to the observation point \mathbf{x} . A point observation of the dual field (e.g. $\eta = H_x(\mathbf{x})$) is only slightly more complicated. Assuming, as will generally be the case, that the dual fields can be written as $\mathbf{h} = \mathbf{T}\mathbf{e}$, where $\mathbf{T} : \mathcal{S}_p \mapsto \mathcal{S}_D$ is a discrete differential operator (e.g. see 21), we have

$$\eta = (\lambda^D)^T \mathbf{T}\mathbf{e}, \quad (40)$$

where $\lambda^D \in \mathcal{S}_D$ is again a sparse vector of interpolation coefficients, now representing averaging on the dual grid. For some problems, $\mathbf{T} \equiv \mathbf{T}_{\pi(\mathbf{m})}$ will depend on the model parameter through $\pi(\mathbf{m})$ (see Section 5.2.4 for an example). It is also possible for the interpolation coefficients λ^P and/or λ^D to depend on the model parameter \mathbf{m} . We will return to these complications, which are accounted for in the operator \mathbf{Q} , below.

Note that for a finite-element formulation, where the solution is represented in terms of a discrete set of basis functions, field component evaluation functionals would have the same form (sparse vectors defined on the primary or dual space), but would have a slightly different interpretation—that is, the non-zero components of the evaluation functional for any location would be computed by evaluating (at this point) the basis functions for all degrees of freedom associated with the containing element.

Together, (39) and (40) give the basic evaluation functionals for the fundamental observables (point measurements of magnetic and electric fields) in any EM problem. For controlled source problems, where the data are typically just point measurements of the primary or dual field, these evaluation functionals are already the rows of \mathbf{L} . More generally, EM data are functions of both electric and magnetic components, at one or more locations. The most obvious example is the impedance, the local ratio of electric and magnetic fields. Other examples include interstation magnetic TFs, network MT accounting for the geometry of long dipoles (Siripunvaraporn *et al.* 2004), or horizontal spatial gradient methods based on array data (Schmücker 2003; Semenov & Shuman 2009). Inevitably, real data must be based on a discrete set of observations of the magnetic and electric fields, so the general EM data functional can be represented as

$$\begin{aligned} \psi_j(\mathbf{e}(\mathbf{m}), \mathbf{m}) &\equiv \gamma_j(\epsilon_1(\mathbf{m}), \dots, \epsilon_{K_P}(\mathbf{m}), \\ \eta_1(\mathbf{e}(\mathbf{m})), \dots, \eta_{K_D}(\mathbf{e}(\mathbf{m}))), \end{aligned} \quad (41)$$

where $\epsilon_k, k = 1, \dots, K_P$ and $\eta_k, k = 1, \dots, K_D$ are sets of primary and dual components computed at one or several points in the model domain, as $\epsilon_k = (\lambda_k^P)^T \mathbf{e}$ and $\eta_k = (\lambda_k^D)^T \mathbf{T} \mathbf{e}$, respectively.

From (10), the j th row of \mathbf{L} is then given by

$$\mathbf{l}_j = \left. \frac{\partial \psi_j}{\partial \mathbf{e}} \right|_{\mathbf{e}_0, \mathbf{m}_0} = \sum_{k=1}^{K_P} \left. \frac{\partial \gamma_j}{\partial \epsilon_k} \frac{\partial \epsilon_k}{\partial \mathbf{e}} \right|_{\mathbf{e}_0, \mathbf{m}_0} + \sum_{k=1}^{K_D} \left. \frac{\partial \gamma_j}{\partial \eta_k} \frac{\partial \eta_k}{\partial \mathbf{e}} \right|_{\mathbf{e}_0, \mathbf{m}_0}, \quad (42)$$

$$= \sum_{k=1}^{K_P} a_{jk}^P (\lambda_k^P)^T + \sum_{k=1}^{K_D} a_{jk}^D (\lambda_k^D)^T \mathbf{T}, \quad (43)$$

where a_{jk}^P and a_{jk}^D are the partial derivatives of the j th data functional with respect to the k th local field components. These coefficients depend only on the details of the data functional formulation, and the background EM solution \mathbf{e}_0 . Eq. (43) thus implies that we can decompose \mathbf{L} into two sparse matrices as

$$\mathbf{L} = \mathbf{A}^T \Lambda^T, \quad (44)$$

with

$$\mathbf{A} = \begin{bmatrix} \mathbf{A}_P \\ \mathbf{A}_D \end{bmatrix} \quad \text{and} \quad \Lambda = [\Lambda_P \quad \mathbf{T}^T \Lambda_D]. \quad (45)$$

Here, \mathbf{L} is a sparse $N_d \times N_e$ matrix that maps the EM solution to the data space, as in Eq. (14). \mathbf{A} is a $K \times N_d$ sparse matrix ($K = K_P + K_D$), such that the non-zero elements in its j th column are the coefficients a_{jk} , the derivatives of the data functionals with respect to each of the relevant local magnetic or electric field components. Finally, Λ is an $N_e \times K$ sparse matrix, with columns $\lambda_k \in \mathcal{S}_P$ containing the field component evaluation functionals, that is, the interpolation coefficients required to compute the k th electric/magnetic field component at a point from the primary EM field.

Thus, Λ depends only on the observation locations for each of the K local field components (and possibly on the model parameter \mathbf{m}_0). Observation functionals (non-linear or linearized) for any sort of EM data will be constructed from the same field component functionals, which are closely tied to the specific numerical discretization scheme used. \mathbf{A} , however, depends on details of the observation functionals (e.g. impedance versus apparent resistivity), and will also depend, in general, on the background EM solution used for linearization, \mathbf{e}_0 . However, \mathbf{A} (which is essentially a linearization of γ) does not depend on the details of the numerical implementation of the forward problem.

5.2.2 \mathbf{L} : multivariate TFs

Multivariate TFs are an important special case of non-linear data functionals which deserve a closer look. Plane wave source TFs provide the most important (and, in fact, only widely applied) example. In this case, two independent sources are assumed, corresponding to spatially uniform sources of a fixed frequency polarized in the x - and y -directions. As a consequence of the linearity of the induction equations, under this assumption any point observation of the EM fields can be linearly related to two reference components, through a frequency-dependent TF. Examples include the rows of the impedance tensor, such as

$$E_x = Z_{xx} H_x + Z_{xy} H_y, \quad (46)$$

vertical field TFs, and intersite magnetic TFs. TF components such as Z_{xx} and Z_{xy} , which are estimated from time-series of electric and magnetic fields observed at a single site, provide the basic input data for 3-D MT inversion.

For completeness, we consider the general case where a generic predicted component, which we denote as Y , is related to N_p predicting variables X_1, \dots, X_{N_p} via the TF

$$Y = \theta_1 X_1 + \dots + \theta_{N_p} X_{N_p}. \quad (47)$$

To evaluate the components of the complex TF vector $\Theta = (\theta_1, \dots, \theta_{N_p})^T$ it is necessary to solve forward problems for each of the assumed source configurations—that is, forward solutions $\mathbf{e}_1, \dots, \mathbf{e}_{N_p}$ for N_p transmitters are required. To compute the TF, we must evaluate Y and $X_j, j = 1, \dots, N_p$ for each of these forward solutions. Here, we represent this as

$$Y_i = \lambda_Y^T \mathbf{e}_i \quad X_{ij} = \lambda_{X_j}^T \mathbf{e}_i \quad i = 1, \dots, N_p. \quad (48)$$

Then, if \mathbf{Y} denotes the vector of predicted components for the N_p transmitters and \mathbf{X} denotes the corresponding $N_p \times N_p$ matrix of predicting variables, the TF can be computed as

$$\Theta = \mathbf{X}^{-1} \mathbf{Y}. \quad (49)$$

Note that, in general, the evaluation functionals $\lambda_{X_j} \in \mathcal{S}_P$ might be more complicated than the simple interpolation operators considered previously—for example, for the usual plane wave source case the predicting components are typically taken to be magnetic fields at the local site, which for the 3-D MT example we have considered would require computation of the secondary field (multiplication by the operator \mathbf{T}) followed by interpolation. And for more exotic cases such as the generalized horizontal spatial gradient (HSG) TF (Egbert 2002; Schmücker 2003, 2004; Semenov & Shuman 2009; Pankratov & Kuvshinov 2010) the predicting components would involve magnetic fields measured at multiple sites, used to form some sort of estimate of uniform and gradient field components. We thus assume only that these are sparse vectors representing linear functionals defined on \mathcal{S}_P .

Taking partial derivatives of Θ with respect to \mathbf{e}_i we find, after some simplification

$$\frac{\partial \Theta}{\partial \mathbf{e}_i} = \mathbf{X}_0^{-1} \left[\frac{\partial \mathbf{Y}}{\partial \mathbf{e}_i} - \frac{\partial (\mathbf{X} \Theta_0)}{\partial \mathbf{e}_i} \right]. \quad (50)$$

In (50), the subscript zero denotes TFs and predicting components evaluated for the background forward solution. Note that the expression in brackets is a matrix of size $N_p \times N_e$ (N_e = dimension of \mathbf{e}_i), but only the i th row is non-zero (only the i th component of \mathbf{Y} and row of \mathbf{X} depend on solution \mathbf{e}_i). This row takes the form

$$\lambda_Y^T - \theta_1 \lambda_{X_1}^T - \dots - \theta_{N_p} \lambda_{X_{N_p}}^T, \quad (51)$$

which is independent of the source polarization index i .

As we noted at the end of Section 3, rows of \mathbf{L} for TF components couple the terms $\mathbf{S}_m^{-1} \mathbf{P}_i$ for multiple transmitters. We can now give an explicit form for this coupling, considering only a single predicted component Y , so that there are N_p complex rows of the matrix \mathbf{L} , one for each component of the TF. \mathbf{L} can also be divided into N_p blocks of columns, one for each transmitter as in (19). From (50) and (51), \mathbf{L} can be written in terms of \mathbf{X}_0^{-1} and block diagonal matrices as

$$\mathbf{L} = \mathbf{X}_0^{-1} \begin{bmatrix} \Psi & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \ddots & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \Psi \end{bmatrix} \begin{bmatrix} \Lambda^T & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \ddots & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \Lambda^T \end{bmatrix}, \quad (52)$$

where

$$\Psi = [1 \quad -\Theta^T] \quad \Lambda = [\lambda_Y^T \quad \lambda_{X_1}^T \quad \dots \quad \lambda_{X_{N_p}}^T]. \quad (53)$$

The product of the first two matrices corresponds to \mathbf{A}^T (and the rightmost of course to Λ) in (44). The more explicit form here more clearly defines the coupling between transmitters, and has important implications for efficient calculation of the full Jacobian, as we discuss further in Section 6.

5.2.3 Matrix \mathbf{Q}

When either the evaluation functionals or the field transformation operator \mathbf{T} have an explicit dependence on the model parameter (denoted in the latter case by $\mathbf{T}_{\pi(\mathbf{m})}$) there is an additional term in the sensitivity matrix, which we have denoted \mathbf{Q} . The j th row of this matrix is given by

$$\mathbf{q}_j = \frac{\partial \psi_j}{\partial \mathbf{m}} \Big|_{\mathbf{e}_0, \mathbf{m}_0} = \left[\sum_{k=1}^{K_P} a_{jk}^P \frac{\partial (\lambda_k^P)^T \mathbf{e}_0}{\partial \pi} \Big|_{\pi(\mathbf{m}_0)} + \sum_{k=1}^{K_D} a_{jk}^D \frac{\partial (\lambda_k^D)^T \mathbf{e}_0}{\partial \pi} \Big|_{\pi(\mathbf{m}_0)} \right] \Pi_{\mathbf{m}_0}, \quad (54)$$

where $\pi(\mathbf{m})$ is the (possibly non-linear) model parameter mapping to the dual or primary grid, and $\Pi_{\mathbf{m}_0}$ is the Jacobian of this mapping. Defining

$$\tilde{\mathbf{T}}_{\pi(\mathbf{m}_0), \mathbf{e}_0} = \frac{\partial}{\partial \pi} [\mathbf{T}_{\pi(\mathbf{m})} \mathbf{e}_0] \Big|_{\pi(\mathbf{m}_0)}, \quad (55)$$

$$\tilde{\Lambda}_P^T = \frac{\partial}{\partial \pi} [(\Lambda_P)^T \mathbf{e}_0] \Big|_{\pi(\mathbf{m}_0)} \quad (56)$$

and

$$\tilde{\Lambda}_D^T = \frac{\partial}{\partial \pi} [(\Lambda_D)^T \mathbf{T}_{\pi(\mathbf{m}_0)} \mathbf{e}_0] \Big|_{\pi(\mathbf{m}_0)}. \quad (57)$$

Eq. (54) can be given in matrix notation

$$\mathbf{Q} = [\mathbf{A}_P^T \ \mathbf{A}_D^T] \begin{bmatrix} \tilde{\Lambda}_P^T \\ \tilde{\Lambda}_D^T + \Lambda_D^T \tilde{\mathbf{T}} \end{bmatrix} \Pi_{\mathbf{m}_0}. \quad (58)$$

If the interpolation coefficients are independent of the model parameters (as will be most often the case) this reduces to

$$\mathbf{Q} = \mathbf{A}_D^T \Lambda_D^T \tilde{\mathbf{T}}_{\pi(\mathbf{m}_0), \mathbf{e}_0} \Pi_{\mathbf{m}_0}. \quad (59)$$

5.2.4 Example: 2-D MT

For 2-D MT, the fundamental observation is an impedance, the ratio E/B of orthogonal components of the electric and magnetic fields. For the TE mode, E_x corresponds to the primary (modelled) field \mathbf{e} , whereas H_y is the secondary field, which is computed as $\mathbf{h} = \mathbf{T}_E \mathbf{e}$. The secondary field mapping can be given explicitly as $\mathbf{T}_E = (-i\omega\mu)^{-1} \mathbf{O} \mathbf{G}$, where \mathbf{O} is a diagonal matrix with entries $+1$ and -1 for components corresponding to y - and z -edges, respectively. Columns of Λ_P and Λ_D now represent bilinear spline interpolation from the 2-D grid nodes and edges, respectively, to the data sites. These are independent of the model parameter, so $\mathbf{Q} \equiv 0$.

The impedance can be written explicitly as

$$Z \equiv \gamma_j(\mathbf{e}) = \frac{\lambda_E^T \mathbf{e}}{\lambda_H^T \mathbf{T}_E \mathbf{e}}, \quad (60)$$

where \mathbf{e} is the (primary) electric field, and λ_E and λ_H are, respectively, columns of Λ_P and Λ_D , and represent bilinear spline interpolation functionals on node (primary) and edge (dual) spaces. From (42), the row of \mathbf{L} corresponding to an impedance is found to be

$$\mathbf{l}_j \equiv \mathbf{l}_Z = (\lambda_H^T \mathbf{T}_E \mathbf{e}_0)^{-1} \lambda_E^T - \left[\lambda_E^T \mathbf{e}_0 / (\lambda_H^T \mathbf{T}_E \mathbf{e}_0)^2 \right] \lambda_H^T \mathbf{T}_E. \quad (61)$$

For the TM mode, Λ_P and Λ_D are the same as in the TE case, but the roles of primary and dual fields are reversed, so that

$$Z \equiv \gamma_j(\mathbf{e}) = \frac{\lambda_E^T \mathbf{T}_H \mathbf{e}}{\lambda_H^T \mathbf{e}}, \quad (62)$$

\mathbf{e} now denoting the (primary) magnetic field. Also the field transformation operator is now $\mathbf{T}_H = \text{diag}[\rho(\mathbf{m})] \mathbf{O} \mathbf{G}$, and thus depends on the model parameter, so \mathbf{Q} will be non-zero. Row j of \mathbf{L} is now

$$\mathbf{l}_j \equiv \mathbf{l}_Z = - \left[\lambda_E^T \mathbf{T}_H \mathbf{e}_0 / (\lambda_H^T \mathbf{e}_0)^2 \right] \lambda_H^T + (\lambda_H^T \mathbf{e}_0)^{-1} \lambda_E^T \mathbf{T}_H, \quad (63)$$

whereas the corresponding row of \mathbf{Q} is found to be

$$\mathbf{q}_j \equiv \mathbf{q}_Z = (\lambda_H^T \mathbf{e}_0)^{-1} \lambda_E^T \text{diag}[\mathbf{O} \mathbf{G} \mathbf{e}_0] \Pi_{\mathbf{m}_0}. \quad (64)$$

Note that the expressions for the scalar impedance for 2-D MT can also be derived as a special (degenerate) case of the multivariate TFs considered earlier.

Linearized data functionals for apparent resistivity and phase are discussed in Appendix C.

5.2.5 Example: 3-D MT

For the 3-D MT problem formulated in terms of the electric fields (Section 5.1.2), the discrete operator $\mathbf{T} = (i\omega\mu)^{-1} \mathbf{C}$ maps from edges to faces, computing magnetic fields through application of the discrete curl operator. Interpolation from edges and faces to an arbitrary location within the 3-D staggered grid model domain can be based on something simple such as trilinear splines. In this case, both Λ and \mathbf{T} are independent of \mathbf{m} , and so $\mathbf{Q} \equiv 0$.

\mathbf{L} can be readily derived as a special case of the multivariate TF with $N_p = 2$. Each row of the 2×2 impedance tensor is a separate TF—that is, Y in the general development of Section 5.2.2 corresponds to E_x for the first row and E_y for the second. The predictor variables X_1, X_2 correspond to the local horizontal magnetic field. Thus, $\lambda_{X_i}^T = \lambda_{H_i}^T \mathbf{T}$, $i = 1, 2$ are functionals for computing the two magnetic field components and $\lambda_Y^T = \lambda_{E_k}^T$ for rows $k = 1, 2$ of the impedance. The 2×2 matrix \mathbf{X} thus has elements $X_{ij} = \lambda_{H_i}^T \mathbf{T}_j$ (same for both rows of the impedance). From (52) and (53), the row of the (complex) \mathbf{L} corresponding to impedance element ki is

$$\begin{bmatrix} \mathbf{X}_{i1}^{-1} (\lambda_{E_k}^T - Z_{k1} \lambda_{H1}^T \mathbf{T} - Z_{k2} \lambda_{H2}^T \mathbf{T}) \\ \mathbf{X}_{i2}^{-1} (\lambda_{E_k}^T - Z_{k1} \lambda_{H1}^T \mathbf{T} - Z_{k2} \lambda_{H2}^T \mathbf{T}) \end{bmatrix} \quad (65)$$

where the components of \mathbf{X} , and the impedance components Z_{kj} are calculated from the background solution. Note that this row of \mathbf{L} has two blocks (each of length N_e), which multiply perturbations to the two polarizations $\delta \mathbf{e}_j$, $j = 1, 2$, and are summed to compute the total perturbation δZ_{ki} to the impedance element. Rows of \mathbf{L} for vertical field TFs, which relate H_z to the two local horizontal components of the magnetic field, have the same form, with λ_{H_z} replacing λ_{E_k} and the two components of the vertical field TF replacing λ_{E_j} , $j = 1, 2$.

6 MULTIPLE TRANSMITTERS

We now give a more explicit discussion of how all of the pieces of \mathbf{J} fit together in the case of multiple transmitters, allowing for the sort of coupling that occurs with multivariate TFs. In general, there will be N_T transmitters, corresponding to different source geometries and/or different frequencies. There will also be a total of N_R measured components of the EM field at some location. Note that these would correspond to the actual field components observed. Some or all of the data actually used for the inversion would be constructed from these, for example, through TFs, with possible further transformation to apparent resistivity and phase. In general, subsets of receiver locations may be used for each transmitter. The full Jacobian for all data can then be written

$$\mathbf{J} = \mathbf{A}^T \begin{bmatrix} \Lambda^T & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \ddots & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \Lambda^T \end{bmatrix} \begin{bmatrix} \mathbf{S}_1^{-1} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \ddots & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{S}_{N_T}^{-1} \end{bmatrix} \times \begin{bmatrix} \mathbf{P}_1 & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \ddots & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{P}_{N_T} \end{bmatrix} + \begin{bmatrix} \mathbf{Q}_1 & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \ddots & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{Q}_{N_T} \end{bmatrix}, \quad (66)$$

where \mathbf{A} is the measurement operator, evaluating solutions for each transmitter for all of the N_R receivers. The matrices \mathbf{P}_t and \mathbf{Q}_t are generally different for each transmitter t , as they depend on the forward solution \mathbf{e}_t , computed for the reference model parameter used for the Jacobian calculation (see 12). Two transmitters, indexed by t_1, t_2 , which only differ in the geometry of the source will typically have identical forward operators, that is, $\mathbf{S}_{t_1} = \mathbf{S}_{t_2}$ (though this is not true for the 2-D MT case, where the two source polarizations are decoupled, and different forward problems are solved for TE and TM modes). The solution operators will always be different for transmitters corresponding to different frequencies. Complications such as the possibility that not all receiver/transmitter pairs are observed, coupling between transmitters through TFs, and further non-linear transformations of data are embedded in the matrix \mathbf{A} . This matrix will generally be very sparse, with diagonal blocks coupling at most a few transmitters.

Perhaps the simplest specific example of (66) is the controlled source cross-well imaging problem (e.g. Alumbaugh & Newman 1997). In this case, transmitters are point magnetic dipoles in one well, and observations are point measurements of the magnetic field in another well. Assuming all transmitter–receiver pairs are observed, the total number of data is $N_d = N_T N_R$, and we may take $\mathbf{A} = \mathbf{I}$. Assuming further that all data are taken at a single frequency, the forward operators are all identical, $\mathbf{S}_t \equiv \mathbf{S}$. Then (assume $\mathbf{Q} \equiv \mathbf{0}$) the transpose of the full Jacobian can be computed as

$$\mathbf{J}^T = [\mathbf{P}_1(\mathbf{S}^T)^{-1} \Lambda \quad \mathbf{P}_2(\mathbf{S}^T)^{-1} \Lambda \quad \dots \quad \mathbf{P}_{N_T}(\mathbf{S}^T)^{-1} \Lambda]. \quad (67)$$

Thus, any of the $N_T N_R$ rows of \mathbf{J}^T can be constructed from N_T forward solutions (required to form \mathbf{P}_t , $t = 1, \dots, N_T$), and N_R adjoint solutions (one for each column λ_r of Λ). At the same time, the gradient of the data misfit can be written in terms of the residual vector (as in 6)

$$\mathbf{J}^T \mathbf{r} = \sum_t \mathbf{P}_t(\mathbf{S}^T)^{-1} \Lambda \mathbf{r}_t, \quad (68)$$

where \mathbf{r}_t are the components of the residual for transmitter t . Thus, calculation of the gradient (as required for each step in an NLCG or quasi-Newton search scheme) will require N_T adjoint solutions

(and again N_T forward solutions, for \mathbf{P}_t , $t = 1, \dots, N_T$). When $N_T \approx N_R$ (as, e.g., in the cross-well EM imaging example) the full Jacobian can thus be had for the same cost (at least in terms of calls to the forward/adjoint solver) as the gradient alone. Although storing the full Jacobian (of size $N_T N_R \times M$) might be prohibitively expensive in terms of memory, by computing and saving the N_T forward and N_R adjoint solutions, a GN scheme can be implemented, solving the normal equations with CG as in Alumbaugh & Newman (1997). This seems certain to be more practical and efficient than direct optimization schemes such as NLCG and quasi-Newton. Extensions of the simple case discussed here, to allow for multiple frequencies or more complex sampling patterns with only some transmitter–receiver pairs, would be straightforward.

In the simple cross-well example, the Jacobian ‘factors’ into components dependent on the transmitter and receiver with the sensitivity for data $d_{t,r}$ (where t and r are, respectively, the transmitter and receiver indices) is $\mathbf{P}_t(\mathbf{S}^T)^{-1} \lambda_r$. A similar factorization will apply to any problem where there are transmitters with a single frequency (more precisely, with identical forward solvers), but multiple source geometries. Many active source problems, in particular marine CSEM, would fall into this category.

This source–receiver factorization also applies to the case of multivariate TFs, and more complicated data derived from them. Consider the N_p rows of \mathbf{J} associated with the components of a single TF Θ . These rows of \mathbf{J} can be represented in the general form (66), with a single forward operator $\mathbf{S}_t \equiv \mathbf{S}$. From (52), we thus have

$$\mathbf{J}_\Theta = \mathbf{X}^{-1} \times \begin{bmatrix} \Psi \Lambda^T & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \ddots & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \Psi \Lambda^T \end{bmatrix} \begin{bmatrix} \mathbf{S}^{-1} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \ddots & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{S}^{-1} \end{bmatrix} \times \begin{bmatrix} \mathbf{P}_1 & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \ddots & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{P}_{N_p} \end{bmatrix} \quad (69)$$

or, for the transpose

$$\mathbf{J}_\Theta^T = [\mathbf{P}_1^T(\mathbf{S}^T)^{-1} \Lambda \Psi^T \quad \dots \quad \mathbf{P}_{N_p}^T(\mathbf{S}^T)^{-1} \Lambda \Psi^T] (\mathbf{X}^{-1})^T. \quad (70)$$

Thus, all N_p rows of \mathbf{J} require only a single adjoint solution, which must then be multiplied by each of the matrices \mathbf{P}_t , $t = 1, \dots, N_p$. The resulting model space vectors are then coupled, through the $N_p \times N_p$ matrix $(\mathbf{X}^{-1})^T$ to form the N_p rows of \mathbf{J}_Θ . For multivariate TF problems, there will generally be several predicted components at a single site, each associated with an N_p component TF. Each of these TFs will require a separate adjoint solution, $(\mathbf{S}^T)^{-1} \Lambda_j \Psi_j^T$ since Λ and Ψ will be different for each TF, but all share the same transmitter dependent matrices \mathbf{P}_t , and the same coupling matrix.

In the context of the 3-D MT problem, one has two TFs, corresponding to the two rows of the impedance tensor, and hence two adjoint solutions are required to compute sensitivities for the full impedance tensor. If vertical field TFs are also included, there would be a third TF, and a third adjoint solution would be required. A direct calculation of sensitivities through the transposed eq. (15), taking account of (19) but ignoring the factorization of \mathbf{L} used to derive (69) and (70), would suggest that two adjoint solutions are required for each of the four components of the

impedance tensor. This would imply a total of eight adjoint solutions to evaluate the full sensitivity for an impedance tensor at one location/frequency. Thus, the more careful analysis given here suggests substantial efficiencies, reducing the total number of adjoint solutions for a full calculation of the Jacobian by a factor of 4.

Sensitivities for any data derived from impedance tensor components can obviously be constructed from the adjoint solutions $(\mathbf{S}^T)^{-1} \mathbf{A}_j \Psi_j^T$, $j = 1, 2$ essentially as in (70) but with a modified coupling matrix, analogous to $(\mathbf{X}^{-1})^T$. An example would be the four components of the phase tensor (e.g. Caldwell *et al.* 2004), which is a non-linear function of the full impedance.

7 MODULAR IMPLEMENTATION

The mathematical developments of previous sections provide a framework for implementation of a general modular system for inversion of frequency-domain EM data. Here, we provide an overview of the organization and principal features of such a system, which we have developed in Fortran 95. A more detailed description of this modular system (hereinafter referred to as ModEM) will be provided in a future publication. Although a purist might argue that it is not strictly possible to write object-oriented code in Fortran 95 we have based our development on this programming paradigm, following approaches appropriate for the Fortran language as described in Akin (2003). We also use the terminology of this approach in our discussion here. As with most object-oriented programming our goals in ModEM are code reuse for multiple related applications, and providing templates for rapid development of new applications.

As discussed in detail earlier, the basic data objects which are manipulated in any inversion scheme include model (**m**) and data (**d**) vectors, and EM solution and source fields (**e** and **b**). These are treated in ModEM as essentially ‘abstract data types’, encapsulated data structures with details of the internal representation effectively hidden from higher level routines which manipulate them. For each of these classes, a standard series of methods must be defined (creation, destruction, vector space methods, dot products, etc.) with standardized interfaces. The inversion algorithms then apply operators such as **f**, **J**, **S**⁻¹, **L**, **P**, **Q**, **C**_m, which are implemented as methods that interact with the basic objects **m**, **d**, **e**, **b**. Standardizing type names and interfaces allows multiple instances of these operators and objects to be used interchangeably within the inversion system, and at the same time, simplifies development of any inversion algorithm that can be described in terms of these components.

Components in ModEM can be usefully organized into three layers, as illustrated in Fig. 3. On the left-hand side of the figure are components which define the basic discretization and numerical solution approach used for the forward problem, whereas the components on the right-hand side are more generic, constructed to be directly applicable to a wide range of EM inverse problems. These are separated by an interface layer, which serves to hide problem and implementation specific details from the more generic inversion modules. Each layer in the figure contains several boxes (representing modules or groups of modules in our actual implementation) which are worth distinguishing at the level of this overview.

Two boxes represent the core of the numerical implementation layer. The first includes the grid, data structures that define the primary and dual-field spaces \mathcal{S}_p , \mathcal{S}_D , field component interpolation functionals (Λ), and the primary to dual mapping **T**—everything

needed to define the discrete formulation of the forward problem. The second provides the actual solver for these discretized equations. To be useful for the inversion system this solver, which will be used for both forward and sensitivity calculations, must allow for general sources and boundary conditions, and for solution of the transposed or adjoint system, as well as the usual forward problem. As noted earlier, the PDEs of EM are intrinsically symmetric, so supporting adjoint solutions is typically almost trivial, although there are some details (e.g. associated with non-uniform grids) that may require some care (e.g. Kelbert *et al.* 2008).

No specific data type or procedure names from the core numerical implementation modules are referenced by more generic components of ModEM, so there is a great deal of flexibility in actual implementation at this base level. We have so far used ModEM with three distinct numerical models: the 3-D (electric field) and 2-D (TE and TM mode) Cartesian coordinate FD models discussed earlier, and a 3-D spherical coordinate FD model for global induction studies formulated in terms of the magnetic fields. Source code from previously developed applications were used for the 2-D MT and spherical models, which are described by Siripunvaraporn & Egbert (2000) and Uyeshima & Schultz (2000), respectively. Relatively minor modifications to these codes were required to ensure the required generality of the solver, and to simplify interfacing with other components of ModEM.

The model space is also placed on the left-hand side of Fig. 3, as important components of this module—in particular the mappings π and Π —are strongly dependent on details of the numerical formulation of the forward problem. At the same time, the model space is heavily used by higher level components of ModEM, including the generic inversion modules, and possibly the data functionals (see Section 5.2). Thus, any implementation of the model space module must follow certain conventions to maintain consistency with the rest of the system, for example, providing methods with standardized names and interfaces for linear algebra, dot products and covariance operators. We view the model parametrization and regularization (also part of this module) as something that should be very easy to extend and modify to accommodate a diversity of interpretation problems. For example, the simple conductivity parametrizations discussed earlier could be modified to enforce bounds on conductivities, for example, by replacing the logarithm by a different conductivity transformation as in Avdeev & Avdeeva (2009), or additional parameters to allow explicitly for near-surface distortion (de Groot-Hedlin 1995) could be added. Completely different model parametrizations (e.g. in terms of interface positions between bodies of known conductivity; Smith *et al.* 1999; de Groot-Hedlin & Constable 2004) or regularization approaches may be appropriate in specific situations. To simplify modification and extension, we adopt a strict object-oriented approach for the model parameter space module, hiding all details of a specific instantiation from the rest of the modular system (i.e. in Fortran 95 all attributes of **m** are ‘private’). Note that only the model parameter mappings depend explicitly on the numerical discretization of the EM fields; the rest of the model space implementation is independent of these details and could in principle be used with multiple numerical modelling approaches.

The generic inversion layer is represented by the three boxes on the right-hand side in Fig. 3. The inversion box represents the actual search algorithms, which are written in a generic way using methods from data space, model space and sensitivity modules. Several of the algorithms discussed in Section 2 have been implemented, including the NLCG scheme (e.g. Rodi & Mackie 2001) and the Data space CG scheme of Siripunvaraporn & Egbert (2007). Other inversion

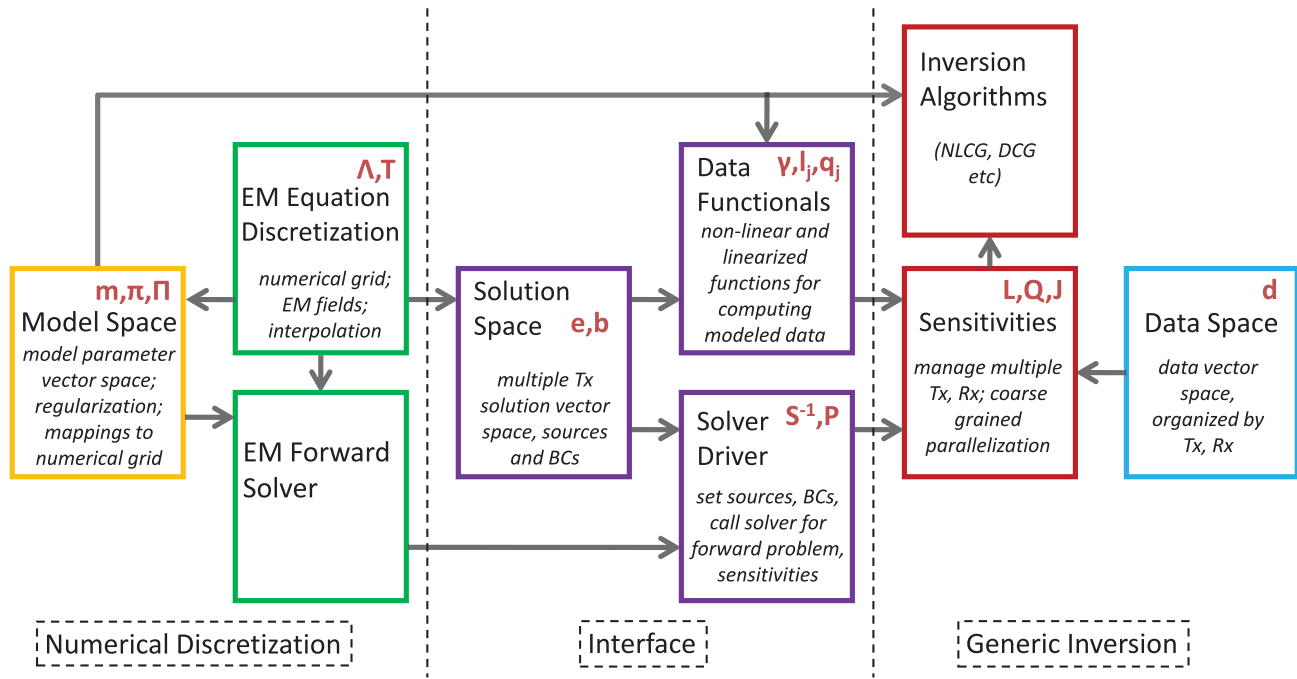


Figure 3. Schematic overview of the Modular Electromagnetic Inversion (ModEM) system. Boxes represent modules (or groups of modules, in actual implementation), with dependencies defined by arrows. Data objects and operators, as defined in Sections 3–5, are listed in the appropriate module, along with a brief summary of function. Tx, Rx denote the transmitter and receiver indices, respectively.

approaches can easily be added. Of course, the same inversion routines can be used for multiple problems: the NLCG code has been applied to 2-D and 3-D MT, simple controlled source EM, and global induction problems in spherical coordinates.

The data space is also part of the generic layer. This is organized, following the discussion of Sections 5.2 and 6, to allow for multicomponent data, observed with multiple receivers, and with sources generated by multiple transmitters. Elements of the data vector \mathbf{d} are thus described by three attributes: *transmitter*, *data type* and *receiver*. *Transmitter* uniquely defines the forward problem that must be solved, including both the specific PDE as well as the sources and boundary conditions. *Receiver* defines, in conjunction with *data type*, the measurement process that must be applied to the forward solution to allow comparison between model and data. The three attributes are treated abstractly at the level of the generic inversion modules, with data vector components carrying only pointers to the actual metadata associated with these attributes (e.g. site location, source polarization, transmitter location, etc.) which are stored as entries in lists, or *dictionaries*. This approach allows a generic format for data storage, hides extraneous details from the inversion modules, and still provides enough information about the transmitter/receiver structure so that forward modelling and sensitivity computations can be organized efficiently.

These tasks are managed by routines in the sensitivity module, which implement the full forward mapping \mathbf{f} and operations with the Jacobian \mathbf{J} or its transpose. For example, the transmitter, receiver and data type attributes can be used to ensure that each required forward problem is solved once (and only once), and then used to compute predicted data (or implement appropriate sensitivity calculations) for all necessary receivers and data types. For some cases (CSEM, and even to some extent 3-D MT; see Section 6), computations with the Jacobian can be ‘factored’ for efficiency into components that depend on the receiver and on the transmitter. In ModEM, such efficiencies can be implemented through specialized versions of the

sensitivity module. A coarse grained parallelization (over transmitters, or unique forward problems, similar to the approach used in Siripunvaraporn & Egbert 2009) is also implemented through the sensitivity module. This allows the parallel version to be used with only minor modifications for a wide range of different applications, and to some extent different search algorithms, including those to be developed in future.

The middle layer in Fig. 3 provides an interface between the generic inversion modules, and the problem and numerical implementation specific base modules. In particular, the EM solution and source terms \mathbf{e} and \mathbf{b} are defined at this level in the solution space module. These objects must always meet the interface standards of the generic layer, but the implementation of a particular instance of these objects will be problem-specific, and built on base layer routines. Source and receiver details for each specific application are also defined in this interface layer. Thus, inversions for different EM methods may be developed using the same base of numerical discretization modules (and of course the same inversion modules) through modifications to the interface layer.

For example, we have used the 3-D Cartesian FD code base for both MT and CSEM. The fundamental EM solution and source objects have distinct implementations for the two methods. For a single transmitter (frequency) in the 3D MT problem, \mathbf{b} represents boundary conditions for two orthogonal plane wave sources, and \mathbf{e} represents the corresponding pair of solutions, each a 3-D vector field. For the CSEM problem, \mathbf{b} represents a single dipole source, and \mathbf{e} is just a single vector field. These differences are implemented in the solution space module. A secondary field formulation (e.g. Alumbaugh *et al.* 1996), which is essential for accurate forward modelling for the CSEM problem, but less critical for MT, is readily implemented through the solver interface module. For the CSEM case, the interface includes routines to compute the primary and scattered fields and hence the source term needed for the FD solver; for the MT case appropriate boundary conditions are

simply generated (and the secondary field approach is not used). In both cases, the same base-level FD solver is then called (once for CSEM, twice for 3-D MT) to do the core computations. Data for the MT and CSEM problems also differ, requiring modifications to the data functional module: for 3-D MT data are TFs or impedances, whereas for CSEM they are just simple observations of individual electric or magnetic field components.

A joint MT-CSEM inversion could also be implemented with very minor changes to the interface layer: solution space, solver driver and data functional modules for MT and CSEM can be merged, with the appropriate case (one or two source polarizations, secondary field solver or MT boundary value problem, impedances or field components) selected based on the transmitter index. This idea can be extended to develop joint inversion for EM with other sorts of geophysical data (seismic, gravity, etc.). In a joint inversion setting, the base layer might include two or more numerical discretization and forward solver modules, with physical parameters that define the forward problems coupled (explicitly or structurally) through a joint model parameter module. These forward problem solvers would then be interfaced to the generic inversion layer through merged solution space, data functional and solver driver modules. The structure of the data space module provides a good basis for developing modified inversion search algorithms as might be appropriate for joint inversion, such as allowing for control over trade-offs between fitting data of different types.

As a brief illustration of some of the capabilities of ModEM, we consider synthetic data inversion tests for three of the EM inverse problems discussed in previous sections: 2-D MT, 3-D MT and global induction. In all of the tests discussed here, we generated synthetic data using some variant on a ‘checkerboard’ conductivity distribution, of the sort often used for resolution tests in seismic tomography, added Gaussian random noise and used the NLCG algorithm implemented in ModEM for inversion.

For the 2-D MT tests, we inverted TE and TM mode data for 12 periods evenly spaced on a logarithmic scale from 0.3–3000 s. Data were generated for 30 sites, with error standard deviation 3 per cent of impedance magnitude. The conductivity model consisted of a checkerboard pattern of 10 and 1000 ohm-m blocks embedded in a 100 ohm-m half-space (Fig. 4a). The same grid ($N_y = 106$

with nominal resolution 1.5 km; $N_z = 40$ increasing logarithmically, starting from 0.5 km) was used for generating the synthetic data, and for the inversion. The covariance used was similar to that of Siripunvaraporn & Egbert (2000), and the prior model was a 100 ohm-m half-space. The NLCG inversion converged from a normalized root-mean-square (rms) misfit of 15.9 to below 1.05 in 68 iterations. The resulting solution, which fits the data to within the expected errors, and captures the main features of the synthetic model, is shown in Fig. 4(b).

For the 3-D MT tests, we used a 3-D variant on the checkerboard, as illustrated in Fig. 5(a). For data we used the full impedance (all four complex components), plus the vertical magnetic field TFs, for 12 periods logarithmically spaced between 10–10 000 s. Error levels were set at 3 per cent of $|Z_{xy}Z_{yx}|^{1/2}$ for all impedance components, and at 0.03 for the non-dimensional vertical magnetic TF components. The grid (again used both for computing the synthetic data and for inversion) was $67 \times 67 \times 60$, with a nominal resolution in the core of 20 km horizontally (see Fig. 5a). A total of 225 sites, on a 15×15 regular 80 km grid were used for the inversion. The covariance was similar to that used for the 2-D tests, and the prior was again a 100 ohm-m half-space. The NLCG algorithm converged from a normalized rms misfit of 12.32 to below 1.05 in 51 iterations, resulting in the inverse solution shown in Fig. 5(b). Again, major model features are well recovered, with some degradation in imaging capability evident below shallower conductive features, as would be expected.

As a final example, we show a simple global induction example. As discussed earlier, the ModEM implementation for this case is based on the spherical coordinate forward solver of Uyeshima & Schultz (2000), which is formulated in terms of the magnetic fields. Data for this global problem are so-called C-responses, ratios of the vertical (H_z) and north (H_x) components of the magnetic field, computed under the assumption that external sources can be approximated well by a zonal (geomagnetic coordinates) dipole. A stand-alone inversion code for this sort of data, based on the same solver, is described by Kelbert *et al.* (2008). An application of the inversion to observatory data is given in Kelbert *et al.* (2009). Here, we only demonstrate our new ModEM version, using a simple synthetic example based on a four-layer 1-D Earth (0–100 km depth:

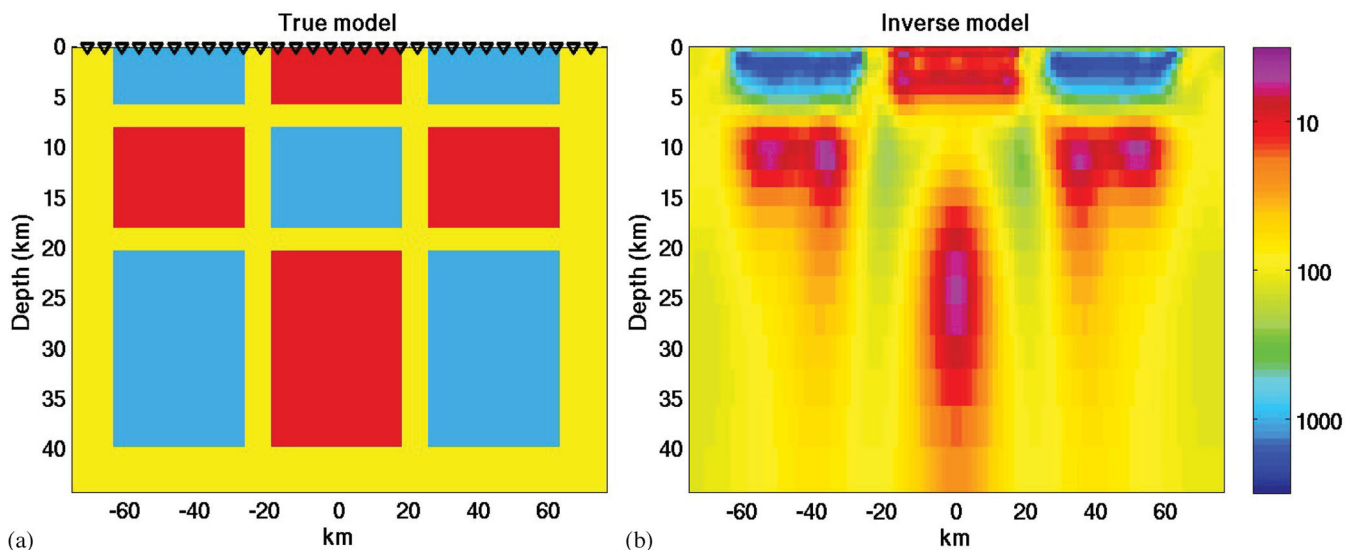


Figure 4. (a) Resistivity model used to generate synthetic data for 2-D MT test, with site locations shown at top. (b) Inverse solution obtained with ModEM, fitting TE and TM mode impedances with a normalized rms misfit of 1.05.

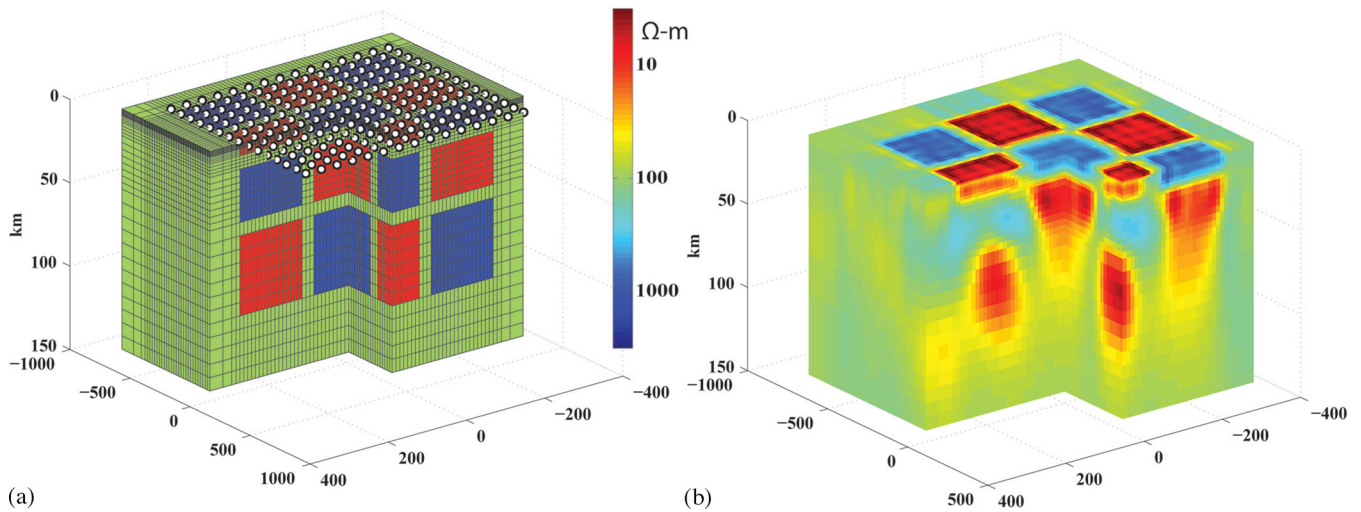


Figure 5. (a) Resistivity model used to generate synthetic data for 3D MT test. The centre of the model grid (used for generating data and for inversion) is shown, along with the regular grid of sites on the surface. (b) Inverse solution obtained with ModEM, fitting full impedance tensor plus vertical magnetic TFs to a normalized rms misfit of 1.05. Note that in the cut-away view the upper surface shown is at 2 km depth, but the structures shown extend to the surface.

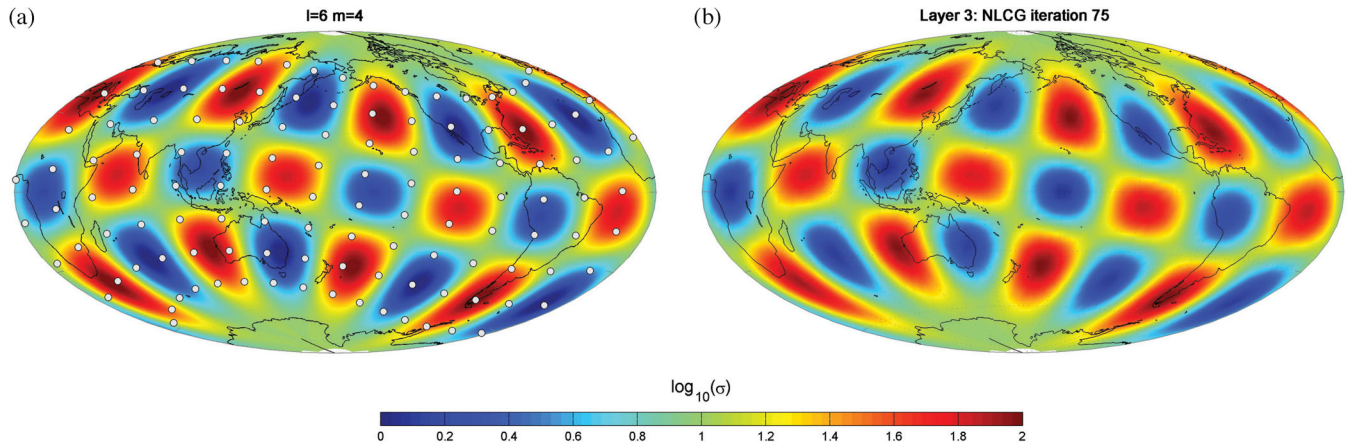


Figure 6. (a) Heterogeneous conductivity in layer 3 (400–650 km depth) of global model used to generate test data for the global induction inversion, with sites shown as filled circles. (b) Conductivity variations in the same layer recovered by the NLCG inversion, implemented with ModEM.

0.0001 S m^{-1} ; 100–400 km: 0.01 S m^{-1} ; 400–650 km: 0.1 S m^{-1} ; 650–4000 km: 2.0 S m^{-1}) with an $l = 6$, $m = 4$ spherical harmonic perturbation (in geomagnetic dipole coordinates) imposed in layer 3 (400–650 km). The amplitude of the perturbation (Fig. 6a) is equivalent to one order of magnitude variation around the 0.1 S m^{-1} background.

Data were distributed on a regular spherical grid (eight latitudes, from 56S to 56N, 15 evenly spaced longitudes, 120 sites total), for four periods: 6 hr, 1, 4 and 16 d. The synthetic C-responses were computed on a $3^\circ \times 3^\circ$ grid, and again 3 per cent Gaussian errors were added. The inversion assumed the same 1-D prior, and a relatively low-dimensional model parametrization: only the third layer was allowed to deviate from uniform, with variations parametrized by spherical harmonics up to degree and order 9. A diagonal (in the spherical harmonic domain) model error covariance was used for the inversion, which was run on a 5° resolution spherical grid. For this case, the inversion converged from a normalized rms misfit of 14.43 to 1.46 in 76 NLCG iterations. Although the fit is not quite to within the expected errors (presumably because of numerical errors associated with the coarser grid used for the inversion)

conductivity variations in layer 3 (Fig. 6b) are recovered almost perfectly.

8 CONCLUSIONS

We have derived general recipes for the Jacobian calculations that are central to a wide range of EM inversion algorithms. Our analysis is based on the discrete formulation of the forward problem, including explicit treatment of parameter mappings and data functionals in the numerical implementation. Through this analysis, we show how the Jacobian can be decomposed into simpler operators, and we analyse the dependence of these operators on the specific EM problem (e.g. through the transmitter and receiver configuration), or on implementation specific details, such as the model parametrization or the nature of the numerical discretization. Based on the general formulation, we provide explicit expressions for Jacobian calculations for several example problems, including 2-D and 3-D MT, and 3-D controlled source problems with multiple transmitter locations. A key result of our general analysis is the ‘factorization’ of the Jacobian into components dependent only on transmitters, and on

receivers. This has important implications for efficient implementation of inversion algorithms, which will be explored more thoroughly elsewhere. To the extent that we have discussed numerical and discretization details of the forward problem, we have focused on FD methods. However, much of our theory is more generally applicable—for example, the division of the Jacobian into components, and the dependencies of these components on details of the EM problem and model parametrization—and will provide a useful guide to development of inversion algorithms for any numerical implementations of the EM forward problem.

Building on the general theoretical framework, we have sketched our development of ModEM, a modular system of computer codes for EM inversion. ModEM allows inversion codes developed for one purpose to be rapidly adapted to other problems, and simplifies development of new capabilities. For example, the 3-D MT inversion discussed earlier can be extended to include intersite magnetic TFs through very minor modifications to the data functional module (essentially adding rows to the matrix **A** in 44). Only slightly greater modifications were required for initial development of an inversion for CSEM data, for which both sources and receivers are different. Flexibility and ease of modification of the model parametrization, and interchangeable inversion search algorithms are other noteworthy features of ModEM.

REFERENCES

- Akin, J.E., 2003. *Object-Oriented Programming via Fortran 90/95*, Vol. 1317, Issue 1, Cambridge University Press, Cambridge, 348pp.
- Alumbaugh, D.L. & Newman, G.A., 1997. Three-dimensional massively parallel electromagnetic inversion: II. Analysis of a crosswell electromagnetic experiment, *Geophys. J. Int.*, **128**, 355–363, doi:10.1111/j.1365-246X.1997.tb01560.x.
- Alumbaugh, D.L., Newman, G.A., Prevost, L. & Shadid, J.N., 1996. Three-dimensional wide band electromagnetic modeling on massively parallel computers, *Radio Sci.*, **33**, 1–23.
- Avdeev, D.B., 2005. Three-dimensional electromagnetic modelling and inversion from theory to application, *Surv. Geophys.*, **26**(6), 767–799.
- Avdeev, D.B. & Avdeeva, A., 2009. 3D magnetotelluric inversion using a limited-memory quasi-Newton optimization, *Geophysics*, **74**(3), F45–F57, doi:10.1190/1.3114023.
- Bennett, A.F., 2002. *Inverse Modeling of the Ocean and Atmosphere*, Cambridge University Press, Cambridge.
- Caldwell, T.G., Bibby, H.M. & Brown, C., 2004. The magnetotelluric phase tensor, *Geophys. J. Int.*, **158**(2), 457–469.
- Chua, B., 2001. An inverse ocean modeling system, *Ocean Modelling*, **3**, 137–165, doi:10.1016/S1463-5003(01)00006-3.
- Commer, M. & Newman, G.A., 2008. New advances in three-dimensional controlled-source electromagnetic inversion, *Geophys. J. Int.*, **172**, 513–535, doi:10.1111/j.1365-246X.2007.03663.x.
- Constable, S.C., Parker, R.L. & Constable, C.G., 1987. Occam's inversion: a practical algorithm for generating smooth models from electromagnetic sounding data, *Geophysics*, **52**(3), 289–300.
- Egbert, G.D., 1994. A new stochastic process on the sphere: application to characterization of long-period global scale external sources, in *14th Workshop on Electromagnetic Induction in the Earth and Moon*, Brest, France.
- Egbert, G.D., 2002. Processing and interpretation of electromagnetic induction array data, *Surv. Geophys.*, **23**(2–3), 207–249.
- Egbert, G.D. & Bennett, A.F., 1996. Data assim., *Modern Approaches to Data Assimilation in Ocean Modeling*, p. 147, Elsevier Science, Amsterdam.
- de Groot-Hedlin, C., 1995. Inversion for regional 2-D resistivity structure in the presence of galvanic scatterers, *Geophys. J. Int.*, **122**(3), 877–888, doi:10.1111/j.1365-246X.1995.tb06843.x.
- de Groot-Hedlin, C. & Constable, S.C., 2004. Inversion of magnetotelluric data for 2D structure with sharp resistivity contrasts, *Geophysics*, **69**(1), 78, doi:10.1190/1.1649377.
- Haber, E., 2005. Quasi-Newton methods for large-scale electromagnetic inverse problems, *Inverse Probl.*, **21**(1), 305–323.
- Kelbert, A., 2006. Geophysical inverse theory applied to reconstruction of large-scale heterogeneities in electrical conductivity of Earth's mantle, *PhD thesis*, Cardiff University.
- Kelbert, A., Egbert, G.D. & Schultz, A., 2008. Non-linear conjugate gradient inversion for global EM induction: resolution studies, *Geophys. J. Int.*, **173**(2), 365–381, doi:10.1111/j.1365-246X.2008.03717.x.
- Kelbert, A., Schultz, A. & Egbert, G.D., 2009. Global electromagnetic induction constraints on transition-zone water content variations, *Nature*, **460**(7258), 1003–1006, doi:10.1038/nature08257.
- de Lugao, P., Portniaguine, O. & Zhdanov, M.S., 1997. Fast and stable two-dimensional inversion of magnetotelluric data, *J. Geomag. Geoelectr.*, **49**(11–12), 1437–1454.
- McGillivray, P.R., Oldenburg, D.W., Ellis, R.G. & Habashy, T.M., 1994. Calculation of sensitivities for the frequency-domain electromagnetic problem, *Geophys. J. Int.*, **116**(1), 1–4, doi:10.1111/j.1365-246X.1994.tb02121.x.
- Mackie, R.L. & Madden, T.R., 1993. 3-dimensional magnetotelluric inversion using conjugate gradients, *Geophys. J. Int.*, **115**(1), 215–229.
- Mackie, R.L., Smith, J.T. & Madden, T.R., 1994. 3-dimensional electromagnetic modeling using finite-difference equations—the magnetotelluric example, *Radio Sci.*, **29**(4), 923–935.
- Marquardt, D.W., 1963. An algorithm for least-squares estimation of non-linear parameters, *J. Soc. Ind. Appl. Math.*, **11**, 431–441.
- Nedelec, J.C., 1980. Mixed finite elements in R3, *Numer. Math.*, **35**(3), 315–341, doi:10.1007/BF01396415.
- Newman, G.A. & Alumbaugh, D.L., 1997. Three-dimensional massively parallel electromagnetic inversion I. Theory, *Geophys. J. Int.*, **128**, 345–354, doi:10.1111/j.1365-246X.1997.tb01559.x.
- Newman, G.A. & Alumbaugh, D.L., 2000. Three-dimensional magnetotelluric inversion using non-linear conjugate gradients, *Geophys. J. Int.*, **140**, 410–424.
- Newman, G.A. & Boggs, P.T., 2004. Solution accelerators for large-scale three-dimensional electromagnetic inverse problems, *Inverse Probl.*, **20**, 151–170, doi:10.1088/0266-5611/20/6/S10.
- Newman, G.A. & Hoversten, G.M., 2000. Solution strategies for two- and three-dimensional electromagnetic inverse problems, *Inverse Probl.*, **16**, 1357–1375, doi:10.1088/0266-5611/16/5/314.
- Nocedal, J. & Wright, S.J., 1999. *Numerical Optimization*, Springer-Verlag, New York, NY.
- Pankratov, O. & Kuvshinov, A., 2010. General formalism for the efficient calculation of derivatives of EM frequency-domain responses and derivatives of the misfit, *Geophys. J. Int.*, **181**(1), 229–249.
- Parker, R.L., 1994. *Geophysical Inverse Theory*, Princeton University Press, Princeton, NJ.
- Rodi, W.L., 1976. A technique for improving the accuracy of finite element solutions for magnetotelluric data, *Geophys. J. Int.*, **44**(2), 483–506, doi:10.1111/j.1365-246X.1976.tb03669.x.
- Rodi, W.L. & Mackie, R.L., 2001. Nonlinear conjugate gradients algorithm for 2-D magnetotelluric inversion, *Geophysics*, **66**(1), 174–187.
- Rodrigue, G. & White, D., 2001. A vector finite element time-domain method for solving Maxwell's equations on unstructured hexahedral grids, *SIAM J. Sci. Comput.*, **23**(3), 683, doi:10.1137/S1064827598343826.
- Sasaki, Y., 2001. Full 3-D inversion of electromagnetic data on PC, *J. appl. Geophys.*, **46**, 45–54, doi:10.1016/S0926-9851(00)00038-0.
- Schmücker, U., 2003. Horizontal spatial gradient sounding and geomagnetic depth sounding in the period range of daily variation, in *Protokoll Aber das Kolloquium elektromagnetische Tiefenforschung*, Kolloquium: Königstein, pp. 228–237.
- Schmücker, U., 2004. Multivariate magneto-variational soundings (MVS), in *Proceedings of the 17th EM Induction Workshop*, Hyderabad.
- Semenov, V.Y. & Shuman, V.N., 2009. Impedances for induction soundings of the Earth's mantle, *Acta Geophys.*, **58**(4), 527–542.
- Siripunvaraporn, W. & Egbert, G.D., 2000. An efficient data-subspace inversion method for 2-D magnetotelluric data, *Geophysics*, **65**(3), 791–803.

- Siripunvaraporn, W. & Egbert, G.D., 2007. Data space conjugate gradient inversion for 2-D magnetotelluric data, *Geophys. J. Int.*, **170**, 986–994, doi:10.1111/j.1365-246X.2007.03478.x.
- Siripunvaraporn, W. & Egbert, G., 2009. WSINV3DMT: vertical magnetic field transfer function inversion and parallel implementation, *Phys. Earth planet. Inter.*, **173**(3–4), 317–329, doi:10.1016/j.pepi.2009.01.013.
- Siripunvaraporn, W., Egbert, G.D. & Lenbury, Y., 2002. Numerical accuracy of magnetotelluric modeling: a comparison of finite difference approximations, *Earth Planets Space*, **54**(6), 721–725.
- Siripunvaraporn, W., Uyeshima, M. & Egbert, G.D., 2004. Three-dimensional inversion for Network-Magnetotelluric data, *Earth Planets Space*, **56**(9), 893–902.
- Siripunvaraporn, W., Egbert, G.D., Lenbury, Y. & Uyeshima, M., 2005. Three-dimensional magnetotelluric inversion: data-space method, *Phys. Earth planet. Inter.*, **150**(1–3), 3–14.
- Smith, J.T., 1996. Conservative modeling of 3-D electromagnetic fields: 1. Properties and error analysis, *Geophysics*, **61**(5), 1308–1318.
- Smith, T., Hoversten, M., Gasperikova, E. & Morrison, F., 1999. Sharp boundary inversion of 2D magnetotelluric data, *Geophys. Prospect.*, **47**(4), 469–486, doi:10.1046/j.1365-2478.1999.00145.x.
- Spitzer, K., 1998. The three-dimensional DC sensitivity for surface and subsurface sources, *Geophys. J. Int.*, **134**, 736–746, doi:10.1046/j.1365-246x.1998.00592.x.
- Uyeshima, M. & Schultz, A., 2000. Geomagnetic induction in a heterogeneous sphere: a new three-dimensional forward solver using a conservative staggered-grid finite difference method, *Geophys. J. Int.*, **140**(3), 636–650.
- Yee, K., 1966. Numerical solution of initial boundary value problems involving Maxwell's equations in isotropic media, *IEEE Trans. Antennas Propag.*, **14**, 302–307, doi:10.1109/TAP.1966.1138693.
- Zhang, J., Mackie, R.L. & Madden, T.R., 1995. 3-D resistivity forward modeling and inversion using conjugate gradients, *Geophysics*, **60**(5), 1313–1325.

APPENDIX A: DEPENDENCE OF SOURCE TERMS ON MODEL PARAMETERS

In some cases (in particular for active source problems), it is appropriate to use a so-called ‘secondary field’ approach to solve the forward problem (e.g. Alumbaugh *et al.* 1996). In this case, a background (typically 1-D) conductivity is assumed, allowing quasi-analytic computation of a background solution, with the ‘secondary’ field due to deviation from the background conductivity then computed numerically. More precisely, the total field solution is represented as $\mathbf{e} = \hat{\mathbf{e}} + \delta\mathbf{e}$, where the background field $\hat{\mathbf{e}}$ satisfies the 1-D equation defined by conductivity parameter $\hat{\mathbf{m}}$. It is readily verified that the secondary field $\delta\mathbf{e}$ satisfies the induction equation with a modified source. Assuming the 3-D operator can be expressed as in (27) this takes the form

$$\mathbf{S}_m \delta\mathbf{e} = -\mathbf{U}[(\pi(\mathbf{m}) - \pi(\hat{\mathbf{m}})) \circ \mathbf{V}\hat{\mathbf{e}}]. \quad (\text{A1})$$

The RHS in (A1) depends on the model parameter \mathbf{m} , suggesting that an additional term should be included in eq. (12).

However, if we differentiate both sides of (A1) with respect to \mathbf{m} and use (27) again we find

$$\frac{\partial}{\partial \mathbf{m}} [\mathbf{S}_0 \delta\mathbf{e} + \mathbf{U}(\pi(\mathbf{m}) \circ \mathbf{V}\delta\mathbf{e})] = -\frac{\partial}{\partial \mathbf{m}} [\mathbf{U}(\pi(\mathbf{m}) \circ \mathbf{V}\hat{\mathbf{e}})], \quad (\text{A2})$$

implying

$$0 = \frac{\partial}{\partial \mathbf{m}} [\mathbf{S}_0 \delta\mathbf{e} + \mathbf{U}(\pi(\mathbf{m}) \circ \mathbf{V}\hat{\mathbf{e}})] = \frac{\partial}{\partial \mathbf{m}} [\mathbf{S}_m \mathbf{e}], \quad (\text{A3})$$

the last equality following from the fact that $\mathbf{S}_0 \hat{\mathbf{e}}$ does not depend on \mathbf{m} . Thus, as long as the RHS of the original problem is in-

dependent of the model parameter, $\partial\mathbf{e}/\partial\mathbf{m} = \partial[\delta\mathbf{e}]/\partial\mathbf{m}$ satisfies (12) without any additional terms, even if the equation for the secondary field does depend on \mathbf{m} . Note also that even if the forward problem is solved with a secondary field approach, the Jacobian calculation (either through 14 or 15) involves only the standard discrete solver \mathbf{S}_m^{-1} . Use of a secondary field approach only affects the derivative indirectly through its dependence on the forward solution.

APPENDIX B: 3-D STAGGERED GRID DETAILS

Here, we give a more precise definition of the discrete finite difference (FD) operator corresponding to $\nabla \times \nabla \times + i\omega\mu\sigma$ and its adjoint, and clarify implementation of boundary conditions for the 3-D magnetotelluric (MT) problem. Similar considerations apply to other cases considered in the text. To do this, we need to distinguish more precisely between interior and boundary nodes in the grid. In the main text, \mathcal{S}_p (\mathcal{S}_D) have been used to denote the space of discrete complex vector fields defined on all edges (faces) of the staggered grid. Here, we use the same symbols with tildes ($\tilde{\mathcal{S}}_p$, $\tilde{\mathcal{S}}_D$) to indicate the restriction to interior edges or faces. The discrete curl operator is naturally defined as a mapping from all edges to all faces, but we need only consider the partial mapping which computes the curl for interior faces (see e.g. Kelbert (2006) for details). Denote this as

$$\mathbf{C} : \mathcal{S}_p \mapsto \tilde{\mathcal{S}}_D \quad (\text{B1})$$

and partition $\mathbf{e} \in \mathcal{S}_p$ and \mathbf{C} into interior and boundary edge components

$$\mathbf{e} = \begin{bmatrix} \tilde{\mathbf{e}} \\ \mathbf{e}_b \end{bmatrix} \quad \mathbf{C} = \begin{bmatrix} \tilde{\mathbf{C}} & \mathbf{C}_b \end{bmatrix}, \quad (\text{B2})$$

so that $\tilde{\mathbf{C}} : \tilde{\mathcal{S}}_p \mapsto \tilde{\mathcal{S}}_D$ and $\mathbf{C}\mathbf{e} = \tilde{\mathbf{C}}\tilde{\mathbf{e}} + \mathbf{C}_b\mathbf{e}_b$.

To define adjoints precisely, we need to specify inner products. The natural inner products for the primary and dual spaces (interior nodes only) are

$$\langle \tilde{\mathbf{e}}_1, \tilde{\mathbf{e}}_2 \rangle_p = \tilde{\mathbf{e}}_1^* \mathbf{V}_E \tilde{\mathbf{e}}_2 \quad \langle \tilde{\mathbf{h}}_1, \tilde{\mathbf{h}}_2 \rangle_D = \tilde{\mathbf{h}}_1^* \mathbf{V}_F \tilde{\mathbf{h}}_2. \quad (\text{B3})$$

In (B3), \mathbf{V}_E and \mathbf{V}_F are real diagonal matrices of edge and face volume elements. Edge volumes, for example, are defined as one-fourth of the total volume of the four cells sharing the edge, so that the first discrete inner product in (B3) approximates the integral L_2 inner product for vector fields $\int \int \int \mathbf{E}_1^*(\mathbf{x}) \cdot \mathbf{E}_2(\mathbf{x}) dV$. The adjoint of the interior curl operator $\tilde{\mathbf{C}}^\dagger : \tilde{\mathcal{S}}_D \mapsto \tilde{\mathcal{S}}_p$ satisfies, by definition,

$$\langle \tilde{\mathbf{h}}, \tilde{\mathbf{C}}\tilde{\mathbf{e}} \rangle_D = \langle \tilde{\mathbf{C}}^\dagger \tilde{\mathbf{h}}, \tilde{\mathbf{e}} \rangle_p \quad \forall \tilde{\mathbf{e}} \in \tilde{\mathcal{S}}_p, \tilde{\mathbf{h}} \in \tilde{\mathcal{S}}_D. \quad (\text{B4})$$

Noting that that $\tilde{\mathbf{C}}$ is real, one then readily derives

$$\tilde{\mathbf{C}}^\dagger = \mathbf{V}_E^{-1} \tilde{\mathbf{C}}^T \mathbf{V}_F. \quad (\text{B5})$$

From the definitions of \mathbf{V}_E and \mathbf{V}_F one can verify that $\tilde{\mathbf{C}}^\dagger$ indeed corresponds to the appropriate geometric definition of the curl operator defined on cell faces. Thus, the electric field eq. (22) with source \mathbf{j}_s

$$\nabla \times \nabla \times \mathbf{E} + i\omega\mu\sigma \mathbf{E} = \mathbf{j}_s \quad (\text{B6})$$

can be approximated on the discrete grid as

$$\tilde{\mathbf{C}}^\dagger \mathbf{C}\mathbf{e} + i\omega\mu\sigma \tilde{\mathbf{e}} = [\tilde{\mathbf{C}}^\dagger \tilde{\mathbf{C}} + i\omega\mu\sigma] \tilde{\mathbf{e}} + \tilde{\mathbf{C}}^\dagger \mathbf{C}_b \mathbf{e}_b = \tilde{\mathbf{b}}, \quad (\text{B7})$$

where $\tilde{\mathbf{b}} \in \tilde{\mathcal{S}}_p$ gives the discrete approximation for the source current \mathbf{j}_s inside the domain; these currents (and hence $\tilde{\mathbf{b}}$) vanish for

the 3-D MT example we have focused on. The discrete system (B7) has one equation for each of the \tilde{N}_e interior edges, but N_e (= total number of edges) unknowns. Boundary conditions are thus required, most simply specification of tangential electric field components on the boundary edges. Then, the full system of equations ($\mathbf{S}\mathbf{e} = \mathbf{b}$) can be decomposed into interior and boundary components as

$$\begin{bmatrix} \tilde{\mathbf{C}}^\dagger \tilde{\mathbf{C}} + i\omega\mu\sigma & \tilde{\mathbf{C}}^\dagger \mathbf{C}_b \\ 0 & \mathbf{I} \end{bmatrix} \begin{bmatrix} \tilde{\mathbf{e}} \\ \mathbf{e}_b \end{bmatrix} = \begin{bmatrix} \mathbf{S}_{ii} & \mathbf{S}_{ib} \\ 0 & \mathbf{I} \end{bmatrix} \begin{bmatrix} \tilde{\mathbf{e}} \\ \mathbf{e}_b \end{bmatrix} = \begin{bmatrix} \tilde{\mathbf{b}} \\ \mathbf{b}_b \end{bmatrix}, \quad (\text{B8})$$

where \mathbf{b}_b represents the specified boundary data. Eliminating the boundary edges results in a well-posed $\tilde{N}_e \times \tilde{N}_e$ problem for electric fields restricted to interior edges

$$[\tilde{\mathbf{C}}^\dagger \tilde{\mathbf{C}} + i\omega\mu\sigma \mathbf{I}] \tilde{\mathbf{e}} = \mathbf{S}_{ii} \tilde{\mathbf{e}} = \tilde{\mathbf{b}} - \tilde{\mathbf{C}}^\dagger \mathbf{C}_b \mathbf{b}_b, \quad (\text{B9})$$

with the RHS determined from the boundary data, and any source terms in the domain. Using (B5), we see that the discrete operator in (B9) can be written as $\mathbf{S}_{ii} = \mathbf{V}_E^{-1} \tilde{\mathbf{C}}^T \mathbf{V}_E \tilde{\mathbf{C}} + i\omega\mu\sigma \mathbf{I}$. Thus, as sketched in Section 3, the system $\mathbf{S}\mathbf{e} = \mathbf{b}$ can be reduced to symmetric form by eliminating the boundary nodes, and then multiplying both sides of the resulting eq. (B9) by \mathbf{V}_E .

We emphasize that in our treatment of the discrete forward problem we take \mathbf{e} , \mathbf{S} and \mathbf{b} to include both interior and boundary nodes. Thus to be precise in our application of (27) to the 3-D FD equations considered here, we should take

$$\mathbf{S}_0 = \begin{bmatrix} \tilde{\mathbf{C}}^\dagger \tilde{\mathbf{C}} & \tilde{\mathbf{C}}^\dagger \mathbf{C}_b \\ 0 & \mathbf{I} \end{bmatrix}, \quad (\text{B10})$$

and we should define $\pi(\mathbf{m}) \equiv \sigma(\mathbf{m}) \equiv 0$ on boundary edges. This is a general property of $\pi(\mathbf{m})$, since the boundary conditions do not depend on the model parameter. This implies that the columns of \mathbf{P} corresponding to boundary nodes will all vanish. Also, accounting for the boundary conditions in the transpose of \mathbf{S} we have, in the notation of (B8),

$$\mathbf{S}^T \mathbf{e} = \begin{bmatrix} \mathbf{S}_{ii} & 0 \\ \mathbf{S}_{ib}^T & \mathbf{I} \end{bmatrix} \begin{bmatrix} \tilde{\mathbf{e}} \\ \mathbf{e}_b \end{bmatrix} = \begin{bmatrix} \tilde{\mathbf{b}} \\ \mathbf{b}_b \end{bmatrix}. \quad (\text{B11})$$

The transposed solution operator $(\mathbf{S}^T)^{-1} \mathbf{b}$, which appears extensively throughout the main text, can thus be interpreted as solution of the homogeneous problem (for interior nodes)

$$\mathbf{S}_{ii}^T \tilde{\mathbf{e}} = \tilde{\mathbf{b}} \quad (\text{B12})$$

followed by computation of the boundary terms

$$\mathbf{e}_b = \mathbf{b}_b - \mathbf{S}_{ib}^T \tilde{\mathbf{e}}. \quad (\text{B13})$$

In fact, solutions to the adjoint problem $(\mathbf{S}^T)^{-1} \mathbf{b}$ are always multiplied by \mathbf{P}^T , and because the rows of \mathbf{P}^T corresponding to boundary components are zero, the boundary terms in (B13) are never actually required for our purposes.

APPENDIX C: TRANSFORMATION OF JACOBIAN TO REAL FORM

To allow for the fact that the model parameter \mathbf{m} is typically real, and in some cases data are also real, we have assumed that \mathbf{d} and \mathbf{J} are real, with any complex observations (e.g. an impedance) represented as two real elements of the data vector. However, throughout the text, we have used complex notation for \mathbf{L} , $\mathbf{S}_{m_0}^{-1}$, \mathbf{P} and \mathbf{Q} , so \mathbf{J} computed from (14) would also be complex. In fact, for complex observations it is readily verified that the real and imaginary parts of a row of the complex expression for the Jacobian give the sensitivity (a vector in the real model parameter space) for the corresponding real and imaginary parts of one observation. Thus, to keep the Jacobian and the data vector strictly real, we can set

$$\bar{\mathbf{d}} = \begin{bmatrix} \Re(\mathbf{d}) \\ \Im(\mathbf{d}) \end{bmatrix} \quad \bar{\mathbf{J}} = \begin{bmatrix} \Re(\mathbf{J}) \\ \Im(\mathbf{J}) \end{bmatrix} = \Re \left[\begin{bmatrix} \mathbf{L} \\ -i\mathbf{L} \end{bmatrix} \mathbf{S}^{-1} \mathbf{P} + \begin{bmatrix} \mathbf{Q} \\ -i\mathbf{Q} \end{bmatrix} \right] \quad (\text{C1})$$

with the convention that for any observations that are intrinsically real the rows corresponding to the imaginary component are omitted. From (C1), $\bar{\mathbf{J}}^T \bar{\mathbf{d}} = \Re(\mathbf{J}^T) \Re(\mathbf{d}) + \Im(\mathbf{J}^T) \Im(\mathbf{d})$. It is easily seen that

$$\bar{\mathbf{J}}^T \bar{\mathbf{d}} = \Re[\mathbf{J}^T \mathbf{d}^*] = \Re[\mathbf{P}^T \mathbf{S}^{T-1} \mathbf{L}^T \mathbf{d}^* + \mathbf{Q} \mathbf{d}^*], \quad (\text{C2})$$

where the superscript asterisk denotes the complex conjugate. Thus, the complex component matrices can be used to construct the real Jacobian $\bar{\mathbf{J}}$, and to implement multiplication by this matrix and its transpose. Note also that while we assume the data vector is real, real and imaginary parts of sensitivities for a complex observation are computed (e.g. via 15) with a single adjoint solution.

Apparent resistivity and phase provide examples of observations that are intrinsically real. In terms of the impedance, the apparent resistivity is defined as

$$\rho_a = (\omega\mu)^{-1} |Z|^2 = (\omega\mu)^{-1} [Z_r^2 + Z_i^2], \quad (\text{C3})$$

where Z_r and Z_i are real and imaginary parts of the impedance Z , and ω is angular frequency. Applying the chain rule,

$$\frac{\partial \rho_a}{\partial \mathbf{m}} = \frac{\partial \rho_a}{\partial Z_r} \frac{\partial Z_r}{\partial \mathbf{m}} + \frac{\partial \rho_a}{\partial Z_i} \frac{\partial Z_i}{\partial \mathbf{m}} = \frac{2}{\omega\mu} \left[Z_r \frac{\partial Z_r}{\partial \mathbf{m}} + Z_i \frac{\partial Z_i}{\partial \mathbf{m}} \right] \quad (\text{C4})$$

$$\begin{aligned} &= \frac{2}{\omega\mu} \left[Z_r \Re \frac{\partial Z}{\partial \mathbf{m}} + Z_i \Im \frac{\partial Z}{\partial \mathbf{m}} \right] = \Re \left[\frac{2Z^*}{\omega\mu} \frac{\partial Z}{\partial \mathbf{m}} \right] \\ &= \Re \left[\frac{2Z^* \mathbf{I}_Z^T}{\omega\mu} \frac{\partial \mathbf{e}}{\partial \mathbf{m}} \right]. \end{aligned} \quad (\text{C5})$$

Thus, $\mathbf{l}_\rho = 2Z^* \mathbf{I}_Z^T / \omega\mu$ gives the (complex) row of \mathbf{L} for an apparent resistivity, again with the convention that the real part of the product in (14) is taken for the corresponding row of the real Jacobian. Similarly for the phase $\phi = \tan^{-1}(Z_i/Z_r)$, we find that the row of \mathbf{L} takes the form $\mathbf{l}_\phi = iZ^* \mathbf{I}_Z^T / |Z|^2$.