Geophysical Journal International

Geophys. J. Int. (2012) **190,** 255–266 doi: 10.1111/j.1365-246X.2012.05523.x

Hybrid conjugate gradient-Occam algorithms for inversion of multifrequency and multitransmitter EM data

Gary D. Egbert

College of Oceanic and Atmospheric Sciences, Oregon State University, Corvallis, OR, USA. E-mail: egbert@coas.oregonstate.edu

Accepted 2012 April 24. Received 2012 March 14; in original form 2011 September 02

SUMMARY

We describe novel hybrid algorithms for inversion of electromagnetic geophysical data, combining the computational and storage efficiency of a conjugate gradient approach with an Occam scheme for regularization and step-length control. The basic algorithm is based on the observation that iterative solution of the symmetric (Gauss-Newton) normal equations with conjugate gradients effectively generates a sequence of sensitivities for different linear combinations of the data, allowing construction of the Jacobian for a projection of the original full data space. The Occam scheme can then be applied to this projected problem, with the tradeoff parameter chosen by assessing fit to the full data set. For EM geophysical problems with multiple transmitters (either multiple frequencies or source geometries) an extension of the basic hybrid algorithm is possible. In this case multiple forward and adjoint solutions (one each for each transmitter) are required for each step in the iterative normal equation solver, and each corresponds to the sensitivity for a separate linear combination of data. From the perspective of the hybrid approach, with conjugate gradients generating an approximation to the full Jacobian, it is advantageous to save all of the component sensitivities, and use these to solve the projected problem in a larger subspace. We illustrate the algorithms on a simple problem, 2-D magnetotelluric inversion, using synthetic data. Both the basic and modified hybrid schemes produce essentially the same result as an Occam inversion based on a full calculation of the Jacobian, and the modified scheme requires significantly fewer steps (relative to the basic hybrid scheme) to converge to an adequate solution to the normal equations. The algorithms are expected to be useful primarily for 3-D inverse problems for which the computational burden is heavily dominated by solution to the forward and adjoint problems.

Key words: Inverse theory; Magnetotelluric; Geomagnetic induction.

1 INTRODUCTION

Among the most widely applied, and practical, approaches to inversion of electromagnetic (EM) geophysical data (e.g., magnetotellurics; MT) in two and three dimensions are regularized schemes based on minimizing a penalty functional of the form

$$\Phi(\mathbf{m},\mathbf{d}) = (\mathbf{d}-\mathbf{f}(\mathbf{m}))^{\mathrm{T}}\mathbf{C}_{\mathbf{d}}^{-1}(\mathbf{d}-\mathbf{f}(\mathbf{m})) + \lambda(\mathbf{m}-\mathbf{m}_{\mathbf{0}})^{\mathrm{T}}\mathbf{C}_{\mathbf{m}}^{-1}(\mathbf{m}-\mathbf{m}_{\mathbf{0}}), \quad (1)$$

(e.g. see Avdeev (2005) and Siripunvaraporn (2012) for reviews). In (1) $\mathbf{C_d}$ and $\mathbf{C_m}$ are data and model covariances; as these are not central to our focus we assume the simplest form for both ($\mathbf{C_d} = \mathbf{I}$, $\mathbf{C_m} = \mathbf{I}$), and we take the *a priori* model parameter $\mathbf{m}_0 = 0$. Treatment of the more general case complicates notation, but presents no essential difficulty for the ideas discussed here (see the Appendix for details). We consider in particular methods for minimization of (1) based on linearization of the non-linear model-data mapping $\mathbf{f}(\mathbf{m})$, that is, that make use of the derivative of $\mathbf{f}$, the $N \times M$ Jacobian $\mathbf{J}$ (so $J_{ij} = \partial f_i/\partial m_j$; $N =$ #data; $M =$ # model parameters). Two general approaches, each with many variants, can be distinguished. In a Gauss-Newton approach (e.g. Parker 1994) the full Jacobian is used to approximate the second-order (Taylor series) expansion of the penalty functional around a current estimate of the model solution. The resulting quadratic form is then minimized, leading to a standard linear least-squares problem, defined (at least formally) by the system of normal equations

$$(\mathbf{J}^{\mathrm{T}}\mathbf{J} + \lambda\mathbf{I})\delta\mathbf{m} = \mathbf{J}^{\mathrm{T}}(\mathbf{d}-\mathbf{f}(\mathbf{m}_n)) - \lambda\mathbf{m}_n, \quad (2a)$$

which can be solved for the model update

$$\mathbf{m}_{n+1} = \mathbf{m}_n + \delta\mathbf{m}. \quad (2b)$$

The whole procedure must be iterated, with the Jacobian recomputed for the updated model parameter, to achieve the minimum of (1). As described in Parker (1994) some form of step-length control is required (e.g. setting $\mathbf{m}_{n+1} = \mathbf{m}_n + \mu\delta\mathbf{m}$ with $0 < \mu \leq 1$ determined by line search). The second approach is epitomized by non-linear conjugate gradients (NLCG; e.g. Rodi & Mackie 2001):

the minimum of (1) is found by direct optimization, computing the gradient of the penalty functional

$$\frac{1}{2}\frac{\partial \Phi}{\partial \mathbf{m}}\bigg|_{\mathbf{m}_n} = -\mathbf{J}^{\mathrm{T}}(\mathbf{d} - \mathbf{f}(\mathbf{m}_n)) + \lambda \mathbf{m}_n, \tag{3}$$

and using this to define a search direction in the model space. Φ is then minimized along this search direction, the model parameter is updated to $\mathbf{m}_{n+1}$ and the whole process is repeated. Both approaches are reviewed and compared in the context of EM geophysics problems of the sort considered here in Rodi & Mackie (2001). Limited memory quasi-Newton (Liu & Nocedal 1989) represents an alternative direct optimization approach, which has also been used for EM inverse problems (e.g. Avdeev & Avdeeva 2009).

The forward problem $\mathbf{f}(\mathbf{m})$ for a frequency-domain EM induction problem, such as 2-D or 3-D MT, involves solving elliptic partial differential equations (PDEs), derived from Maxwell's equations. For example, in quasi-static 3-D EM problems the governing equations formulated in terms of the electric field $\mathbf{E}$ are:

$$\nabla \times \nabla \times \mathbf{E} - \mathrm{i}\omega\mu\sigma\mathbf{E} = \mathbf{s}. \tag{4}$$

The forward mapping $\mathbf{f}(\mathbf{m})$ requires solving (4) subject to appropriate boundary conditions, and using the solution, evaluated at observation locations, to compute predicted data. In (4) σ is the spatially varying electrical conductivity of the medium, which we assume is defined through the unknown discrete model parameter $\mathbf{m}$, ω is angular frequency and $\mathbf{s}$ represents the sources (which may vanish, as for MT where the system is forced through the boundary conditions). In most realistic problems data are available for N_f frequencies, and N_s source geometries, so a total of $N_f N_s$ PDEs must be solved to evaluate $\mathbf{f}(\mathbf{m})$ for a single model parameter. As shown in general by Newman & Hoversten (2000), Pankratov & Kuvshinov (2010), and Egbert & Kelbert (2012) and previously for numerous specific examples referenced therein, computing one row (or one column) of $\mathbf{J}$ requires solving the governing PDE (or more precisely, its adjoint; though (4) is essentially self-adjoint) once. Evaluating a matrix-vector product such as $\mathbf{J}^{\mathrm{T}}\mathbf{r}$ (e.g. in the gradient of data misfit used in (3)) requires essentially the same computations as one forward problem.

A GN approach would appear at first blush to be much less efficient than NLCG: to implement (2) directly, one must apparently first compute all of $\mathbf{J}$ (requiring $N = N_f N_s N_r$ (where N_r is the number of receivers) solutions of the appropriate PDE, one for each row of the Jacobian), and then form and solve the $M \times M$ system of equations. In contrast, a single iteration with (3) requires a single gradient computation, followed by a line search to minimize over the search direction (generally requiring 2–4 additional solutions of the forward problem). However, as Rodi & Mackie (2001) show, NLCG requires many more iterations (typically 50–100 or more) compared to a GN scheme (typically 5–10 or less; see examples below). Furthermore, for 'multitransmitter' problems (i.e. with multiple frequencies and/or source geometries) each forward solution or gradient evaluation actually requires solving the governing PDE $N_f N_s$ times. Accounting for the significantly greater number of iterations needed for convergence (each requiring a line search) direct minimization with NLCG may require as many or more PDE solutions as a GN scheme based on full calculation of $\mathbf{J}$ (Siripunvaraporn & Egbert 2007; Siripunvaraporn & Sarakorn 2011). However, NLCG still avoids forming and solving the large system of normal equations of (2), so this and related approaches are now used in almost all implementations of 3-D inversion (e.g. Commer & Newman 2008; Avdeev & Avdeeva 2009); the efforts of Sasaki (2001), Siripunvaraporn *et al.* (2005) and Siripunvaraporn & Egbert (2009) are exceptions.

It is of course possible to use a GN approach without explicitly forming the normal equations of (2a), but instead solve this symmetric linear system of equations iteratively using conjugate gradients (CG). This approach, which has been used fairly extensively for EM inversion (e.g. Mackie & Madden 1993; Alumbaugh & Newman 1997; Rodi & Mackie 2001) is a variant on the truncated Newton approach to optimization (e.g. Dembo *et al.* 1982; Nash 2000), with the Hessian replaced by the GN approximation (e.g. Newman & Hoversten 2000).

To be concrete, and to set the stage for coming developments, we consider a variant on the GN equations of (2):

$$(\mathbf{J}\mathbf{J}^{\mathrm{T}} + \lambda\mathbf{I})\mathbf{b} = \hat{\mathbf{d}} = \mathbf{d} - \mathbf{f}(\mathbf{m}_n) + \mathbf{J}\mathbf{m}_n \tag{5a}$$

$$\mathbf{m}_{n+1} = \mathbf{J}^{\mathrm{T}}\mathbf{b}. \tag{5b}$$

This data space scheme (e.g. Siripunvaraporn & Egbert 2000; Siripunvaraporn & Sarakorn 2011), which requires solving the $N \times N$ system of normal equations in the data space (instead of the $M \times M$ system in the model parameter space), can be shown to be equivalent to (2). Instead of actually making the full dense matrix, one can again use CG, which requires multiplying an arbitrary vector by the coefficient matrix $(\mathbf{J}\mathbf{J}^{\mathrm{T}} + \lambda\mathbf{I})$. This in turn requires multiplication of data space vectors by $\mathbf{J}^{\mathrm{T}}$ and model space vectors by $\mathbf{J}$, essentially the same sort of computations as required by NLCG. This approach also avoids calculation of the full Jacobian and eliminates the need to form the normal equations. As shown in Siripunvaraporn & Egbert (2007) the total number of PDE solutions is, however, still typically comparable to that required for a full calculation of $\mathbf{J}$. And the CG scheme has an apparent disadvantage: once the full Jacobian is calculated, solving (5a) for different values of the tradeoff parameter λ is fairly fast—in particular no further PDE solutions are required.

The Occam approach (Constable *et al.* 1987; see also Parker 1994) exploits this efficiency, varying λ both for step length control, and as a damping parameter, to search for minimum norm inverse solutions, which fit the data to a prescribed tolerance. Once $\mathbf{J}$ is computed (5) is used to compute a series of trial solutions corresponding to a range of λ, and the forward problem is then solved for each to evaluate the actual data misfit achieved as a function of λ. Initially, λ is chosen to minimize data misfit; as the scheme converges λ is chosen to minimize the model norm while keeping the misfit constant at the target value (Constable *et al.* 1987; Parker 1994). With this approach λ 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, subject to the data fit attained (Parker 1994). With a straightforward application of CG all of the PDE solution steps must be repeated for each new trial value of λ (Siripunvaraporn & Egbert 2007). The same situation holds for NLCG: the penalty functional is minimized with λ fixed, and the entire (or at least much of) the iterative process must be repeated with each new trial value. Thus, if one has to vary the regularization parameter—and often this is critical, even if one does not have the precise information about data error levels required to rigorously provide an *a priori* target misfit—GN schemes based on full calculation of $\mathbf{J}$ would appear to have some advantages.

We make two points in this paper. The first is in fact rather obvious: at the cost of a modest increase in memory requirements, CG schemes can be easily modified to allow the Occam approach to be implemented without computing the full Jacobian. The idea is closely related to the so-called hybrid algorithms, which have

previously been discussed in the context of damped least-squares problems (O'Leary & Simmons 1981; Kilmer & O'Leary 2001; Hanke 2001). It can also be viewed as a special case of the subspace inversion methods of Oldenburg *et al.* (1993), in which a small and effective model subspace is generated by the iterative CG solver.

Our second point is more novel, and is specific to multitransmitter inverse problems where computational costs are dominated by the need for multiple expensive forward solutions. Such problems are the norm in EM geophysics, and arise also in other geophysical problems, such as full waveform seismic inversion (e.g. Tape *et al.* 2010). We show that for such problems iterative solution of the data space normal eq. (5a) can be modified to achieve substantially more rapid convergence (in particular, with fewer required forward solutions). Both ideas follow from a more careful examination of the iterative CG algorithms used to solve (5a). We thus review the basis for the CG solution approach—that is, the Lanczos process—in Section 2, and demonstrate how the standard solution scheme can be easily modified to implement a hybrid CG-Occam scheme. In Section 3, we develop a modification to the Lanczos process that uses the multiplicity of forward and adjoint solutions required in multitransmitter EM geophysical inverse problems to accelerate convergence of the solution to the normal equations, leading to a modified hybrid CG-Occam algorithm. In Section 4, we demonstrate the efficacy of the new schemes using the 2-D MT inverse problem as a simple illustrative example. Although this simple problem is sufficient to demonstrate the effectiveness of the new algorithms, we stress that these schemes are likely to be most useful for 3-D problems where computational costs are dominated by expensive forward and adjoint solutions required for gradient calculations. Results and possible extensions are discussed in Section 5.

2 A HYBRID CG-OCCAM SCHEME

To motivate and describe the hybrid schemes we begin with a review of the Lanczos bi-diagonalization algorithm of Paige & Saunders (1982a), which forms the basis for standard CG solution methods. Here the algorithm 'BIDIAG1' is applied to the Jacobian $\mathbf{J}$, with the ultimate objective of solving the system of normal eq. (5a), initially taking $\lambda = 0$. In the first step of the Lanczos process unit vectors in the data and model space are computed

$$\beta_1 \mathbf{u}_1 = \hat{\mathbf{d}} \qquad \|\mathbf{u}_1\| = 1 \tag{6a}$$

$$\alpha_1 \mathbf{v}_1 = \mathbf{J}^{\mathrm{T}} \mathbf{u}_1 \qquad \|\mathbf{v}_1\| = 1. \tag{6b}$$

A key point to note here is that the model space vector $\alpha_1 \mathbf{v}_1 = \mathbf{J}^{\mathrm{T}} \mathbf{u}_1$ is just the sensitivity of a particular linear combination of data components, namely $\mathbf{u}_1^{\mathrm{T}} \mathbf{d}$ (ignoring noise $\partial \mathbf{u}_1^{\mathrm{T}} \mathbf{d} / \partial \mathbf{m} = \partial \mathbf{u}_1^{\mathrm{T}} \mathbf{f} / \partial \mathbf{m} = \mathbf{u}_1^{\mathrm{T}} \partial \mathbf{f} / \partial \mathbf{m} = \mathbf{u}_1^{\mathrm{T}} \mathbf{J} = \alpha_1 \mathbf{v}^{\mathrm{T}}$). Next compute

$$\mathbf{J}\mathbf{v}_1 = \alpha_1 \mathbf{u}_1 + \beta_2 \mathbf{u}_2, \tag{7}$$

where $\mathbf{u}_2$ is orthogonal to $\mathbf{u}_1$. Now if $\beta_2 = 0$, $\mathbf{J}\mathbf{J}^{\mathrm{T}}[\beta_1/\alpha_1^2]\mathbf{u}_1 = \beta_1 \mathbf{u}_1 = \hat{\mathbf{d}}$ so $\mathbf{b}_1 = (\beta_1/\alpha_1^2)\mathbf{u}_1$ would be an exact solution to (5a). In general, this will not be the case, and this initial estimate of $\mathbf{b}$ must be refined. We thus continue for $k = 2, \ldots, K$ (where K is determined by the stopping criterion discussed below)

$$\beta_k \mathbf{u}_k = \mathbf{J}\mathbf{v}_{k-1} - \alpha_{k-1}\mathbf{u}_{k-1} \qquad \|\mathbf{u}_k\| = 1 \tag{8a}$$

$$\mathbf{J}^{\mathrm{T}} \mathbf{u}_k - \beta_k \mathbf{v}_{k-1} = \alpha_k \mathbf{v}_k \qquad \|\mathbf{v}_k\| = 1, \tag{8b}$$

generating sequences of data and model space vectors (which can be saved as orthogonal matrices $\mathbf{U}_K = [\mathbf{u}_1 \cdots \mathbf{u}_K]$ and $\mathbf{V}_K = [\mathbf{v}_1 \cdots \mathbf{v}_K]$, respectively) and scalars α_k, β_k which can be organized as the bi-diagonal matrix

$$\mathbf{B}_K = \begin{bmatrix} \alpha_1 & \beta_2 & \cdots & 0 \\ 0 & \alpha_2 & \ddots & \vdots \\ \vdots & \ddots & \ddots & \beta_K \\ 0 & \cdots & 0 & \alpha_K \end{bmatrix}. \tag{9}$$

Then (6)–(8) can be expressed in matrix notation as

$$\mathbf{J}^{\mathrm{T}} \mathbf{U}_K = \mathbf{V}_K \mathbf{B}_K \tag{10}$$

$$\mathbf{J}\mathbf{V}_K = \mathbf{U}_K \mathbf{B}_K^{\mathrm{T}} + \beta_{K+1} \mathbf{u}_{K+1} \hat{\mathbf{e}}_K^{\mathrm{T}}, \tag{11}$$

where $\hat{\mathbf{e}}_K$ is the unit vector for coordinate K in $\mathbb{R}^K$ and $\mathbf{J}\mathbf{v}_K = \alpha_K \mathbf{u}_K + \beta_{K+1}\mathbf{u}_{K+1}$. The original system $\mathbf{J}\mathbf{J}^{\mathrm{T}}\mathbf{b} = \hat{\mathbf{d}}$ can be solved approximately by first projecting into the K-dimensional data subspace spanned by the columns of $\mathbf{U}_K$; i.e.

$$\mathbf{U}_K^{\mathrm{T}} \mathbf{J}\mathbf{J}^{\mathrm{T}} \mathbf{U}_K \tilde{\mathbf{b}}_K = \mathbf{U}_K^{\mathrm{T}} \hat{\mathbf{d}} = \beta_1 \hat{\mathbf{e}}_1. \tag{12}$$

The last equality follows from (6a) and orthonormality of the columns of $\mathbf{U}_K$. On the other hand, from the orthonormality of $\mathbf{V}_K$ and (10), the system to be solved can be seen to be symmetric, positive definite and tri-diagonal

$$\mathbf{B}_K^{\mathrm{T}} \mathbf{B}_K \tilde{\mathbf{b}} = \beta_1 \hat{\mathbf{e}}_1, \tag{13}$$

and hence easily solved. The vector $\mathbf{b}_K = \mathbf{U}_K \tilde{\mathbf{b}}$ then provides an approximate solution to the original system. Indeed we have from (10–13) and (6a)

$$\begin{aligned} \mathbf{J}\mathbf{J}^{\mathrm{T}} \mathbf{b}_K &= \mathbf{J}\mathbf{J}^{\mathrm{T}} \mathbf{U}_K \tilde{\mathbf{b}} = \mathbf{J}\mathbf{V}_K \mathbf{B}_K \tilde{\mathbf{b}} = \left[\mathbf{U}_K \mathbf{B}_K^{\mathrm{T}} + \beta_{K+1}\mathbf{u}_{K+1}\hat{\mathbf{e}}_K^{\mathrm{T}}\right]\mathbf{B}_K \tilde{\mathbf{b}} \\ &= \mathbf{U}_K \mathbf{B}_K^{\mathrm{T}} \mathbf{B}_K \tilde{\mathbf{b}} + \beta_{K+1}\mathbf{u}_{K+1}\left[\hat{\mathbf{e}}_K^{\mathrm{T}} \mathbf{B}_K \tilde{\mathbf{b}}\right] \\ &= \beta_1 \mathbf{U}_K \hat{\mathbf{e}}_1 + \beta_{K+1}\left[\alpha_K \hat{\mathbf{e}}_K^{\mathrm{T}} \tilde{\mathbf{b}}\right]\mathbf{u}_{K+1} \end{aligned} \tag{14}$$

$$= \beta_1 \mathbf{u}_1 + \alpha_K \beta_{K+1} \tilde{b}_K \mathbf{u}_{K+1} = \hat{\mathbf{d}} + \alpha_K \beta_{K+1} \tilde{b}_K \mathbf{u}_{K+1}, \tag{15}$$

where $\tilde{b}_K$ is the Kth component of the vector $\tilde{\mathbf{b}}$.

In standard implementations of CG the system (12) is not actually formed and solved. Rather, the approximate solution $\mathbf{b}_K$ is updated 'on the fly', starting from $\mathbf{b}_1 = (\beta_1/\alpha_1^2)\mathbf{u}_1$. Iterations can be terminated when the residual in the solution to (eq. 5a; i.e. $\mathbf{J}\mathbf{J}^{\mathrm{T}}\mathbf{b}_K - \hat{\mathbf{d}} = \alpha_K \beta_{K+1} \tilde{b}_K \mathbf{u}_{K+1}$) is sufficiently reduced, for example, when $\|\alpha_K \beta_{K+1} \tilde{b}_K \mathbf{u}_{K+1}\| / \|\hat{\mathbf{d}}\| < \varepsilon$. More generally, memory efficient and numerically stable schemes for damped least squares problems (e.g., LSQR) have been developed based on Lanczos bi-diagonalization (Paige & Saunders 1982b). With these approaches memory requirements are minimal—only the most recent $\mathbf{u}_k, \mathbf{v}_k$ need be retained, and solutions (and residuals) are updated at each step k. However, by actually saving all of $\mathbf{U}_K$, $\mathbf{V}_K$ and $\mathbf{B}_K$ (or in fact $\mathbf{U}_K$ and $\mathbf{J}^{\mathrm{T}}\mathbf{U}_K$) it is possible to form and solve the small $(K \times K)$ system $[\mathbf{U}_k^{\mathrm{T}} \mathbf{J}\mathbf{J}^{\mathrm{T}} \mathbf{U}_K + \lambda \mathbf{I}]\mathbf{b}_\lambda = \mathbf{U}_K^{\mathrm{T}} \hat{\mathbf{d}}$ (analogous to (12)) for any value of the regularization parameter λ. It is readily verified that the same error estimate (15) applies to this modified system. This approach, which allows an efficient implementation of the Occam scheme, is an example of a hybrid algorithm, of the sort previously discussed extensively in the numerical linear algebra literature (O'Leary & Simmons 1981; Kilmer & O'Leary 2001; Hanke 2001).

A hybrid Occam-CG scheme is thus obvious: (1) Apply Lanczos bi-diagonalization to $\mathbf{J}$, saving the orthonormal matrix $\mathbf{U}_K$, and the K model space vectors $\mathbf{J}^{\mathrm{T}}\mathbf{U}_K$. (2) Use these to form the $K \times K$

$\mathbf{m}_{prior}$ = prior model $\quad$ $\mathbf{m}_0$ = starting model

Outer loop: For $n = 0, 1, 2, \ldots$

$$\hat{\mathbf{d}}_n = \mathbf{d} - \mathbf{f}(\mathbf{m}_n) + \mathbf{J}[\mathbf{m}_n - \mathbf{m}_{prior}]$$

BIDIAG1:

$\beta_1 \mathbf{u}_1 = \hat{\mathbf{d}}$ $\quad$ $\|\mathbf{u}_1\| = 1$

$\alpha_1 \mathbf{v}_1 = \mathbf{J}^T \mathbf{u}_1$ $\quad$ $\|\mathbf{v}_1\| = 1$

for $k = 2, 3, \ldots$

$\beta_k \mathbf{u}_k = \mathbf{J}\mathbf{v}_{k-1} - \alpha_{k-1}\mathbf{u}_{k-1}$ $\quad$ $\|\mathbf{u}_k\| = 1$

if $|\beta_k \alpha_{k-1}| / \|\hat{\mathbf{d}}\| < \varepsilon$ exit loop

$\mathbf{J}^T \mathbf{u}_k - \beta_k \mathbf{v}_{k-1} = \alpha_k \mathbf{v}_k$ $\quad$ $\|\mathbf{v}_k\| = 1$

end BIDIAG1 (save $\mathbf{U}_K = [\mathbf{u}_1 \quad \cdots \quad \mathbf{u}_K]$, $\mathbf{J}^T \mathbf{U}_K$)

Optimize λ: for trial values of λ

Solve $K \times K$ system $\left[\mathbf{U}_K^T \mathbf{J}\mathbf{J}^T \mathbf{U}_K + \lambda \mathbf{I}\right] \mathbf{b}_\lambda = \mathbf{U}_K^T \hat{\mathbf{d}}$

$\mathbf{m}_\lambda = \mathbf{J}^T \mathbf{U}_K \mathbf{b}_\lambda$

Phase I:

choose λ to minimize $\|\mathbf{d} - \mathbf{f}(\mathbf{m}_\lambda)\|^2$

Phase II:

choose λ so that $\|\mathbf{d} - \mathbf{f}(\mathbf{m}_\lambda)\|^2 = Tol$

end outer loop

Figure 1. Pseudo-code for hybrid Occam-DCG.

cross-product matrix $\mathbf{R} = \mathbf{U}_K^{\mathrm{T}} \mathbf{J}\mathbf{J}^{\mathrm{T}} \mathbf{U}_K$. This matrix is in principal tri-diagonal, but round-off error will cause increasing large deviations as K increases, so it is best to retain and work with the matrix $\mathbf{J}^{\mathrm{T}}\mathbf{U}_K$. (3) Optimize the regularization parameter by solving the projected system $[\mathbf{R} + \lambda \mathbf{I}]\mathbf{b}_\lambda = \mathbf{U}_K^{\mathrm{T}} \hat{\mathbf{d}}$ for a series of values of λ, and minimize the misfit $\|\mathbf{d} - \mathbf{f}(\mathbf{m}_\lambda)\|^2$, where $\mathbf{m}_\lambda = \mathbf{J}^{\mathrm{T}} \mathbf{b}_\lambda$; after the target misfit is achieved, choose λ to minimize the penalty functional subject to achieving the target misfit. Pseudo-code for the scheme is given in Fig. 1. This hybrid scheme effectively uses the Lanczos process to generate a subset of sensitivities (i.e. the columns of $\mathbf{U}_K^{\mathrm{T}} \mathbf{J}$), corresponding to the data subspace spanned by $\mathbf{U}_K$. The Occam scheme is then applied to this projected problem, with the tradeoff parameter chosen by assessing fit to the full data set.

The basic hybrid algorithm solves the linear subproblem in the model subspace spanned by the columns of $\mathbf{V}_K$, and can thus be viewed as a special case of the subspace inversion methods discussed in Oldenburg *et al.* (1993). Although we have focused on a data space Occam approach, the same ideas are readily adapted to alternative G-N formulations in the model space, for example, to solve (2) for $\delta \mathbf{m}$. From this perspective the Lanczos bi-diagonalization can be viewed as a scheme for generating a particular model subspace, which approximates the row span of $\mathbf{J}$, and thus should be particularly efficient for finding approximate solutions to the full system of normal equations (with any value of the regularization parameter). Note that the Lanczos process already generates the sensitivity matrix-model parameter products $\mathbf{J}\mathbf{V}_K$ needed to generate the reduced normal equations for the subspace inversion approach (see Oldenburg *et al.* 1993), so a subspace inversion based on saving the full set of Lanczos vectors would be quite efficient.

3 A MODIFIED HYBRID SCHEME

In most EM inverse problems data are available for multiple frequencies, or more generally, with multiple transmitters (different frequencies and/or different source geometries). In this case the data vector and Jacobian can be decomposed into J (= number of transmitters) blocks as

$$\mathbf{d} = \begin{pmatrix} \mathbf{d}_1 \\ \vdots \\ \mathbf{d}_J \end{pmatrix}, \tag{16}$$

$$\mathbf{J}^{\mathrm{T}} = \begin{bmatrix} \mathbf{J}_1^{\mathrm{T}} & \cdots & \mathbf{J}_J^{\mathrm{T}} \end{bmatrix}. \tag{17}$$

A product such as $\mathbf{J}^{\mathrm{T}}\mathbf{r} = \sum_j \mathbf{J}_j^{\mathrm{T}} \mathbf{r}_j$ actually entails separate computations for each transmitter (each requiring solution of the governing PDE appropriate for that frequency), followed by summing the results (a sequence of J model space vectors). Details of the Jacobian calculation are somewhat different for the case of multiple transmitters (with a common frequency), but the decompositions of (16) and (17) remain valid and the required number of forward and adjoint solutions remains the same. Each one of the model space vectors $\mathbf{J}_j^{\mathrm{T}} \mathbf{r}_j$ gives the sensitivity of a linear combination of data

for a single transmitter. The basic idea behind the modified hybrid algorithm is to save all of these separate sensitivities, and use these to solve a projected data-space system analogous to (12). The key modification is actually to the Lanczos scheme, which we will show in Section 4 results in convergence of the normal equations in fewer iterations (i.e. with fewer matrix-vector products $\mathbf{J}^{\mathrm{T}}\mathbf{u}$ and $\mathbf{Jv}$).

The scheme is motivated by the observation that if we saved the individual transmitter data-space vectors and corresponding sensitivities $\mathbf{u}_{kj}, \mathbf{J}_j^{\mathrm{T}}\mathbf{u}_{kj}$ generated by the Lanczos bi-diagonalization discussed earlier, we could project (5) into a much larger (JK-dimensional) data subspace (which would contain the K-dimensional space spanned by the vectors $\mathbf{u}_k$), perhaps leading to a more accurate solution—or rather, allowing an equally accurate solution for a smaller value of K. We have found a modification to this simple idea to be significantly more effective.

As for the standard Lanczos bi-diagonalization, the modified scheme generates a sequence of data space vectors which we denote as $\tilde{\mathbf{u}}_1, \ldots, \tilde{\mathbf{u}}_K$, subdivided into individual transmitter components as $\tilde{\mathbf{u}}_k^{\mathrm{T}} = [\mathbf{w}_{1k}^{\mathrm{T}} \cdots \mathbf{w}_{Jk}^{\mathrm{T}}]$, with each now normalized separately so $\|\mathbf{w}_{jk}^{\mathrm{T}}\| = 1$. As for standard Lanczos schemes the process is started from the right hand side $\hat{\mathbf{d}}$ of the system (5), but now with each block of the data vector normalized separately $\mathbf{w}_{j1} = \hat{\mathbf{d}}_j / \|\hat{\mathbf{d}}_j\|$. Leaving aside for the moment how the vectors $\mathbf{w}_{jk}$ are generated for $k > 1$, let $\Omega_{jk} = [\mathbf{w}_{j1} \cdots \mathbf{w}_{jk}]$ be the matrix constructed from the first $k$subvectors for transmitter j, and define the block diagonal matrix

$$\mathbf{W}_k = \mathbf{diag}\,(\Omega_{1k}, \ldots, \Omega_{Jk}). \tag{18}$$

Then the columns of $\mathbf{J}_j^{\mathrm{T}}\Omega_{jk}$ are model parameter vectors corresponding to the sensitivity for the k linear combinations of data defined by $\Omega_{jk}^{\mathrm{T}}\mathbf{d}_j$, and

$$\mathbf{J}^{\mathrm{T}}\mathbf{W}_K = \left[\mathbf{J}_1^{\mathrm{T}}\Omega_{1K} \cdots \mathbf{J}_J^{\mathrm{T}}\Omega_{JK}\right], \tag{19}$$

is the $M \times (KJ)$ matrix containing all of the sensitivities generated by the first K steps. We show by induction that, with the scheme for generating $\tilde{\mathbf{u}}_k$ described next, $\Omega_{jk}^{\mathrm{T}}\Omega_{jk} = \mathbf{I}$ for all k (i.e. for fixed j the vectors $\mathbf{w}_{jk}, k = 1, \ldots, K$ are orthonormal) so that $\mathbf{W}_K^{\mathrm{T}}\mathbf{W}_K = \mathbf{I}$.

Orthonormality of $\mathbf{W}_K$ certainly holds for $K = 1$. Supposing it holds also for K, we can use the computed sensitivities to solve the projected problem

$$\left(\mathbf{W}_K^{\mathrm{T}}\mathbf{J}\mathbf{J}^{\mathrm{T}}\mathbf{W}_K + \lambda_0\mathbf{I}\right)\tilde{\mathbf{b}} = \mathbf{W}_K^{\mathrm{T}}\hat{\mathbf{d}} \tag{20}$$

for any fixed λ_0. This is analogous to (12), but the matrix $\mathbf{W}_K$ has KJ instead of K columns, so the projected problem is solved in a larger subspace. Given the solution to (20) we next compute $\tilde{\mathbf{m}}_K = \mathbf{J}^{\mathrm{T}}\mathbf{b}_K = \mathbf{J}^{\mathrm{T}}\mathbf{W}_k\tilde{\mathbf{b}}$. If iterative solution of the linear subproblem (5a) were truncated at this point, $\tilde{\mathbf{m}}_K$ would be the model update for the next iteration given in eq. (5b). To continue iterations we compute

$$\mathbf{J}\tilde{\mathbf{m}}_K = \mathbf{J}\mathbf{J}^{\mathrm{T}}\mathbf{W}_K\tilde{\mathbf{b}}_K = \mathbf{W}_K\mathbf{W}_K^{\mathrm{T}}\mathbf{J}\mathbf{J}^{\mathrm{T}}\mathbf{W}_K\tilde{\mathbf{b}}_K + \mathbf{e}_{K+1}, \tag{21}$$

where $\mathbf{e}_{K+1}$ is orthogonal to all of the columns of $\mathbf{W}_K$, that is, $\mathbf{W}_K^{\mathrm{T}}\mathbf{e}_{K+1} = 0$. But then we have $\Omega_{jK}^{\mathrm{T}}\mathbf{e}_{j,K+1} = 0$, so setting $\mathbf{w}_{j,K+1} = \mathbf{e}_{j,K+1}/\|\mathbf{e}_{j,K+1}\|$, this vector is orthogonal to $\mathbf{w}_{jk}, k = 1, \ldots, K$. Thus, $\Omega_{j,K+1}, j = 1, \ldots, J$, and hence $\mathbf{W}_{K+1}$, are all orthonormal matrices, as claimed. Note that $\mathbf{e}_{K+1}$ is analogous to $\mathbf{u}_{K+1}$ in (8a)—that is, it represents the next data-space search direction, but blocks for each transmitter will be used separately.

Note that $\mathbf{W}_K\mathbf{W}_K^{\mathrm{T}}\hat{\mathbf{d}} = \hat{\mathbf{d}}$, and thus (20–21) imply that

$$\left(\mathbf{J}\mathbf{J}^{\mathrm{T}} + \lambda_0\mathbf{I}\right)\mathbf{b}_K = \hat{\mathbf{d}} + \mathbf{e}_{K+1}, \tag{22}$$

Replace BIDIAG1 with:

$$\mathbf{w}_{j1} = \hat{\mathbf{d}}_j / \left\|\hat{\mathbf{d}}_j\right\| \qquad j = 1, \ldots, J$$

for $k = 1, 2, ..., K$

Compute sensitivities:

$$\mathbf{J}_j^T\mathbf{w}_{jk} \qquad j = 1, \ldots, J$$

Accumulate model and data space vectors:

$$\Omega_{jk} = \left[\mathbf{w}_{j1} \quad \cdots \quad \mathbf{w}_{jk}\right]$$

$$\left[\mathbf{J}_1^T\Omega_{1k} \quad \cdots \quad \mathbf{J}_J^T\Omega_{Jk}\right] \qquad j = 1, \ldots, J$$

Denote:

$$\mathbf{W}_k = \mathbf{diag}\left(\Omega_{1k}, \cdots, \Omega_{Jk}\right)$$

$$\mathbf{Y}_k = \left[\mathbf{J}_1^T\Omega_{1k} \quad \cdots \quad \mathbf{J}_J^T\Omega_{Jk}\right] = \mathbf{J}^T\mathbf{W}_k$$

Solve for fixed λ_0 (=1) :

$$\left(\mathbf{Y}_k^T\mathbf{Y}_k + \lambda_0\mathbf{I}\right)\tilde{\mathbf{b}} = \mathbf{W}_k\hat{\mathbf{d}} = \begin{bmatrix} \Omega_{1k}^T\hat{\mathbf{d}}_1 \\ \vdots \\ \Omega_{Jk}^T\hat{\mathbf{d}}_J \end{bmatrix}$$

$$\tilde{\mathbf{m}}_k = \mathbf{Y}_k\tilde{\mathbf{b}}$$

Find next data space vectors for each transmitter:

$$\mathbf{c}_{jk} = \mathbf{J}_j\tilde{\mathbf{m}}_k \qquad j = 1, \ldots, J$$

$$\mathbf{e}_{jk} = \mathbf{c}_{jk} - \Omega_{jk}\Omega_{jk}^T\mathbf{c}_{jk}$$

$$\mathbf{w}_{j,k+1} = \mathbf{e}_{jk} / \left\|\mathbf{e}_{jk}\right\|$$

end

Figure 2. Pseudo-code for modified hybrid scheme.

so that $\mathbf{b}_K$ provides a good approximate solution to (5a) provided $\|\mathbf{e}_{K+1}\|$ is small enough. This can thus serve as a stopping criterion. If the residual is not sufficiently reduced, $\mathbf{e}_{K+1}$ can be used to generate the data space vectors $\mathbf{w}_{j,K+1}, j = 1, \ldots, J$, along with the corresponding model space vectors $\mathbf{J}_j^{\mathrm{T}}\mathbf{w}_{j,K+1}$ for the next iteration.

Pseudo-code for the modified hybrid scheme is given in Fig. 2. A key point to note is that a full solution to the projected linear subproblem is required at each step—that is, (20) is solved, and $\tilde{\mathbf{m}}_K$ formed for the computation of (21). In fact this is what is required to verify that the solution in the projected subspace (i.e. $\mathbf{b}_K$) solves the unprojected system (22) with sufficiently small residual. However, forming and solving the projected normal equations will not represent a serious computational challenge as long as KJ remains a small fraction of the total number of data. In particular, for 3-D problems where computational effort is dominated by solving the 3-D forward and adjoint problems, these extra steps will typically be negligible.

Note that the modified scheme depends on the initial tradeoff parameter selected λ_0. This is because the intermediate solution $\tilde{\mathbf{m}}_k$, which is used through (21) to compute the next data space search vectors $\mathbf{e}_{k+1}, \mathbf{w}_{j,k+1}$, depends on λ_0. The trade-off parameter should scale with the eigenvalues of the matrix $\mathbf{J}\mathbf{J}^{\mathrm{T}}$, and we can very roughly estimate this scale from

$$\mathbf{Tr}\left[\mathbf{W}_1^{\mathrm{T}}\mathbf{J}\mathbf{J}^{\mathrm{T}}\mathbf{W}_1\right] = \sum\nolimits_j \left\|\mathbf{J}_j^{\mathrm{T}}\mathbf{w}_{j1}\right\|^2 / J = \eta_0, \tag{23}$$

which is computed in the first step of the modified Lanczos process, before λ_0 is required.

In our tests we have taken $\lambda_0 = 0.01\eta_0$ for the first loop of the Occam scheme, and then used the optimal λ from the previous iteration for subsequent iterations.

As with the standard scheme, the modified Lanczos scheme generates a model subspace, now spanned by the KJ columns of $\mathbf{J}^{\mathrm{T}}\mathbf{W}_K$. However, the connection to the model subspace inversion methods of Oldenburg *et al.* (1993) is now somewhat weaker than for the standard hybrid scheme. To solve the model space equations of (2) projected into this subspace it would be necessary to compute $\mathbf{Jv}$ for all KJ model space vectors, but in the modified Lanczos scheme the Jacobian is only applied to the K intermediate model solution vectors $\tilde{\mathbf{m}}_k$. The modified hybrid scheme developed here is thus more clearly rooted in the data space perspective, with the inverse problem solved for a projection of the full data vector. It is also worth noting that the sequence of data subspaces that we solve the problem in are not the usual Krylov subspaces generated by repeated application of $\mathbf{JJ}^{\mathrm{T}}$. Indeed the actual sequence of projected spaces depends to some extent on λ_0.

4 EXAMPLE: 2-D MT

As an illustration of the above ideas we consider the 2-D MT inverse problem. While the more complicated modified hybrid scheme for multitransmitter problems can hardly be justified for such a problem, where computational costs associated with forward and adjoint calculations are relatively low, this simple problem is sufficient to demonstrate the two main points we wish to make: (1) even for modest values of K the hybrid schemes essentially reproduce results obtained with a full Occam scheme based on a full Jacobian calculation, and (2) the modified hybrid scheme accomplishes this with fewer iterations of the Lanczos schemes, and hence fewer forward and adjoint solutions.

For 2-D MT electrical conductivity is assumed to be a function of depth z and 'cross-strike' distance y, with no variation along the x-direction (e.g. Fig. 3a), and source magnetic fields are assumed to be uniform (constant in x and y at $z = -\infty$). The magnetic source can be polarized either perpendicular or parallel to strike, corresponding to TE and TM modes, with induced currents flowing along and across strike, respectively (i.e. in the x direction and in the y–z plane). Data for this problem are complex impedances ($Z_{xy} = E_x/B_y$ for TE mode; $Z_{yx} = E_y/B_x$ for TM mode) observed at a series of N_s y-locations at the surface $z = 0$, for a set of N_f frequencies. With this setup the total number of 'transmitters', each requiring solution of a separate forward problem, is $J = 2N_f$, and the total number of (complex) data is $N = 2N_fN_s$. We have implemented the inversion schemes outlined above (a standard data space Occam approach, plus the hybrid scheme of Fig. 1 and the modified hybrid scheme of Fig. 2) and tested these on a range of synthetic data sets; we show results from two cases here. Forward and adjoint problems were solved numerically using a finite difference approach, essentially identical to that used in Siripunvaraporn & Egbert (2000), with the actual inversion procedures implemented in Matlab. In our implementation the regularization term was essentially as in (1), with deviations from a prior model $(\mathbf{m} - \mathbf{m}_0)$ penalized using a model space covariance similar to that described by Siripunvaraporn & Egbert (2000). See the Appendix for further details.

The test case I (Fig. 3a) is fairly simple, consisting of a series of blocks (three relatively conductive, one resistive) buried in a 100 ohm-m half-space. TE and TM mode data were generated for $N_s = 40$ sites evenly spaced between $-30 \le y \le 30$ km, at $N_f = 16$ frequencies logarithmically spaced between 0.00033 and 3.3 Hz. We thus have a total of 1280 complex (2560 real) synthetic observations, to which we add 5 per cent random noise. Results of applying the data space Occam inversion scheme, using a 100 ohm-m half-space as a prior (and starting) model, are shown in Fig. 3(b). The algorithm converges in four outer loop iterations, and structures in the synthetic model are recovered accurately. Trade-off curves, showing misfit as a function of the regularization parameter λ used in (5), are shown for each of the four outer-loop Occam iterations in Fig. 4(a). Note that the minimum in the trade-off curve occurs because of nonlinearity of the inverse problem; for a linear problem the misfit would converge to zero as λ is reduced.

Test case II (Fig. 5a) presents greater challenges, with a more complex pattern of near-surface heterogeneity, and more spatially extensive deep conductivity variations. Synthetic data for this model were generated in the same way as for case one ($N_s = 40$, $N_f = 16$, 5 per cent noise added). Starting from the 100 ohm-m prior the initial misfit is much greater (650 vs. 23.5 normalized rms), and the Occam scheme does not quite achieve the target misfit, stalling with a normalized rms of 2.4 after eight iterations (Fig. 6a). The model achieving this misfit is shown in Fig. 5(b). Many features are recovered (e.g. the alternating pattern of conductive and resistive near-surface blocks, the deep vertical conductor in the middle of

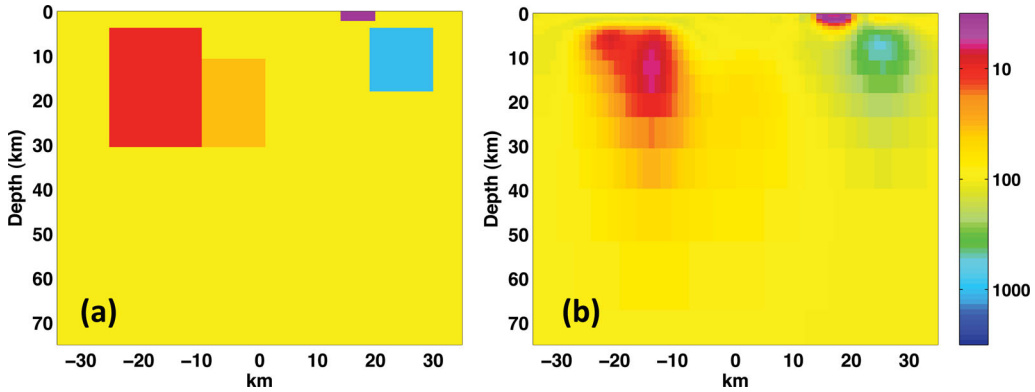

Figure 3. Synthetic 2-D MT test case I. (a) Resistivity model used to generate synthetic data. (b) Resistivity model recovered by Occam algorithm, based on full Jacobian calculation. Results obtained with the hybrid and modified hybrid schemes are indistinguishable, and are not plotted.

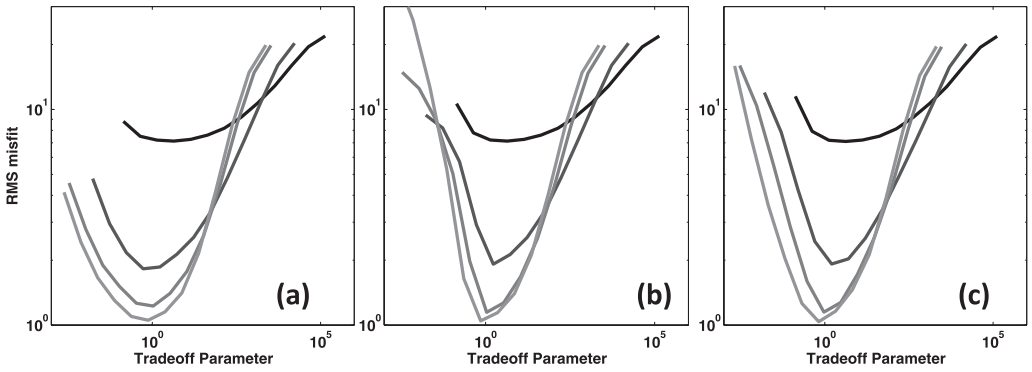

Figure 4. Trade-off curves for test case I, for (a) full Occam scheme; (b) hybrid scheme; (c) modified hybrid scheme.

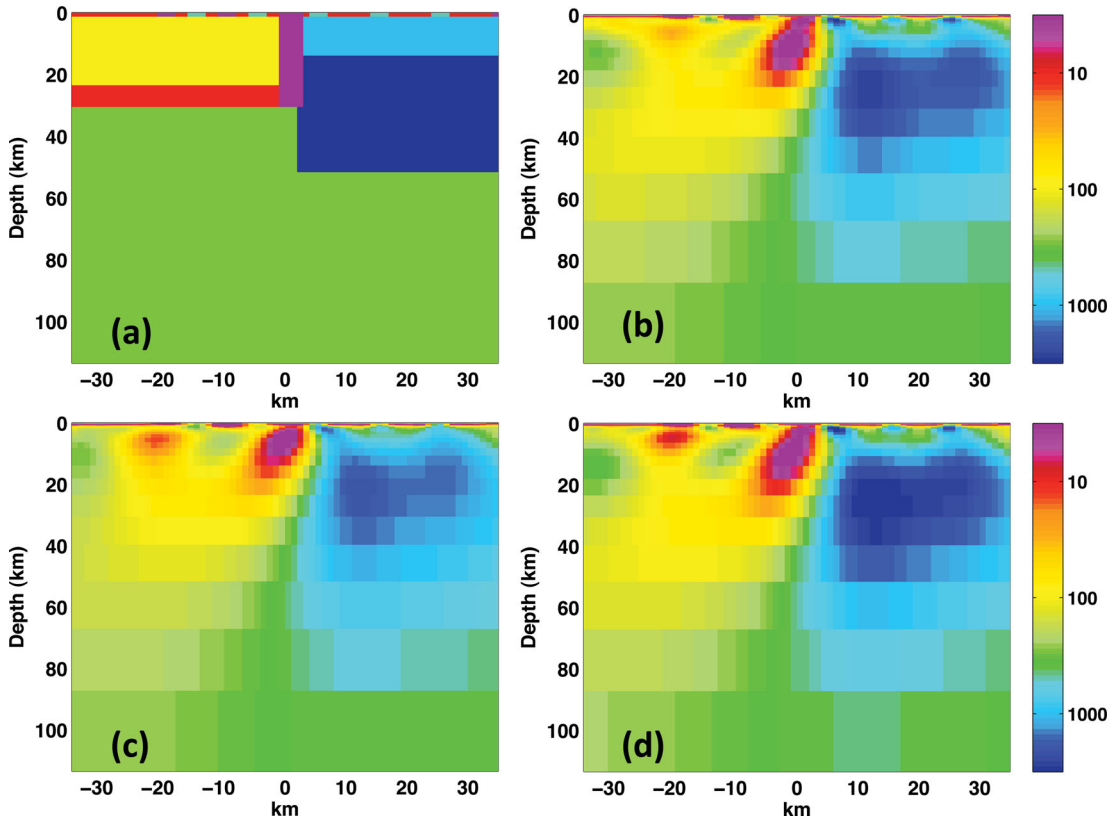

Figure 5. Synthetic 2-D MT test case II. Resistivity models (a) used to generate synthetic data; (b) recovered by Occam algorithm; (c) with the Hybrid scheme, and (d) with modified hybrid scheme. Essentially the same solution is recovered in all cases.

the domain, lateral variations in deep resistivity), although in detail the result deviates somewhat from the model used to generate the data,

For the hybrid Occam scheme we terminated the inner-loop (BIDIAG1) algorithm when the relative error in the solution to (5) (i.e. $\|(\mathbf{J}\mathbf{J}^{\mathrm{T}} + \lambda\mathbf{I})\mathbf{b} - \hat{\mathbf{d}}\|/\|\hat{\mathbf{d}}\|$ dropped below $\varepsilon = 10^{-2}$, or the number of iterations exceeded $K_{\max} = 30$. Using these convergence criteria the hybrid scheme reproduced results obtained with the standard Occam scheme based on the full Jacobian for both test cases. The final hybrid-scheme solution for case II (also fitting to a normalized rms of 2.4) is shown in Fig. 5(c). For case I results from the hybrid scheme are indistinguishable from the full Occam solution, and are not shown. Trade-off curves for the hybrid scheme are shown in Figs 4(b) and 6(b) for the two synthetic test cases. The behaviour as a function of iteration is very similar to that obtained with the full Jacobian, though the minima of the trade-off curves become somewhat narrower with the hybrid scheme.

For each iteration the hybrid solution is constructed as a linear combination of the model space vectors $\mathbf{J}^{\mathrm{T}}\mathbf{u}_k$, $k = 1, \ldots, K$. The first three of these, computed for the first iteration of test case I (i.e. with the Jacobian calculated for a 100 ohm-m half-space) are plotted in Figs 7(a)–(c). Note this Jacobian depends only on the uniform distribution of sites (and the frequencies), and the spatial patterns that dominate the basis functions are determined by the data (which determine the data-space vectors $\mathbf{u}_k$). In particular, the large positive feature in Fig. 7(a) coincides with the near-surface conductor between kilometres 10 and 20 (Fig. 1a), which has severely distorted the synthetic data from nearby sites.

The basis for the modified hybrid scheme is illustrated through the two lower rows of Fig. 7, where some of the individual transmitter

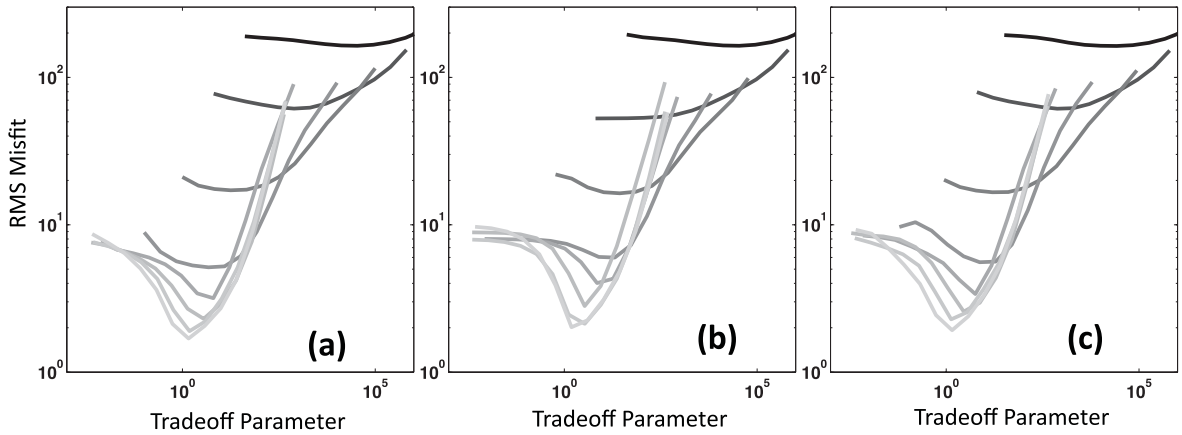

Figure 6. Trade-off curves for test case II, for (a) full Occam scheme; (b) Hybrid scheme; (c) Modified hybrid scheme.

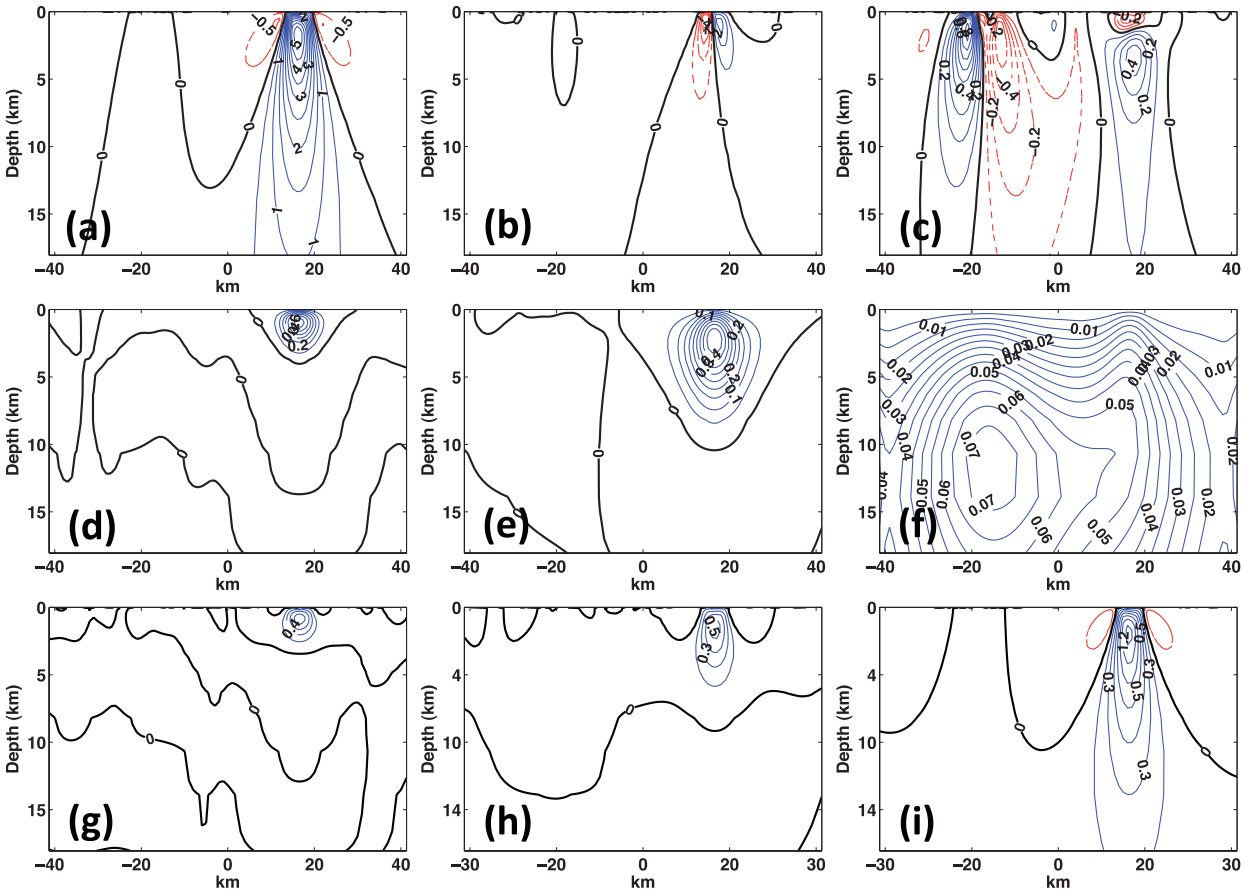

Figure 7. Sensitivity components $\mathbf{J}^{\mathrm{T}}\mathbf{u}$ generated by the hybrid and modified hybrid schemes. (a–c) First three model space vectors generated by BIDIAG1 on the first (outer loop) iteration for test case I (i.e. with the Jacobian calculated for a 100 ohm-m half-space). In the lower two rows selected individual transmitter component sensitivities (again for the first iteration test case) are plotted for three frequencies (3.3, 0.5, 0.02 Hz) for (d–f) the TE mode and (g–i) the TM mode.

component sensitivities are plotted. More specifically, the model space vector derived at the first step (plotted for test case I in Fig. 7a) can be expanded

$$\mathbf{J}^{\mathrm{T}}\mathbf{u}_1 = \sum_{j=1}^{J} \mathbf{J}_j^{\mathrm{T}}\mathbf{u}_{1j}, \tag{24}$$

where the index $j = 1, \ldots, J$ indicates transmitter number. For our examples, with TE and TM mode data for 16 periods, the total number of transmitters is $J = 32$. Component sensitivities for three frequencies (3.3, 0.5, 0.02 Hz) are plotted for the TE mode in Figs 7(d)–(f). Sensitivities for the same three frequencies for the TM mode are shown in Figs 7(g)–(i). Note that the sensitivities generally vary fairly smoothly with frequency. The lowest frequency TM mode sensitivities (e.g. Fig. 7i) are very similar by themselves to the sum of (24). Evidently, fitting the large static shifts associated with the near-surface conductor is the first priority in the iterative CG solution. Other model features are evident in the larger set of basis functions available to the multitransmitter scheme. In particular the large conductive block on the left side of the model at 5–30 km

depth (Fig. 1a), corresponds to a clear peak in the same area in the longest period TE mode sensitivity (Fig. 7f). We anticipate that the additional model space basis functions will allow better approximation of the solution after fewer inner-loop steps.

This expectation is confirmed in Fig. 8, where we plot convergence to the solution of (5) for the inner loop of the standard and modified CG Occam schemes. For both test cases I and II the modified scheme converges more rapidly, with comparable reduction in the normal equation residual in roughly half the number of iterations required of the standard CG scheme. Convergence of the outer loop of the modified hybrid Occam scheme remains comparable to the standard data space Occam implementation based on the full Jacobian (Figs 4c and 6c). Final model results are also virtually identical for both case I (not shown) and case II (Fig. 5d).

It is well know that numerical round-off causes orthogonality of the sequence of vectors $\mathbf{u}_k, \mathbf{v}_k$ generated by the Lanczos process to break down as k increases, degrading convergence of the CG solver (e.g. Gollub & Van Loan 1989). The modified hybrid scheme explicitly enforces orthogonality of the sequence $\tilde{\mathbf{u}}_k$, and perhaps this is at least in part responsible for the more rapid convergence seen in Fig. 8. To test this we repeat the Lanczos bi-diagonalization, modified so that the sequence $\mathbf{u}_k, k = 1, \ldots, K$ remains exactly orthonormal, as in, for example, the generalized conjugate residual scheme of Eisenstat *et al.* (1983). For both cases I and II convergence of the CG scheme with explicit orthogonalization at each step shows significant improvement, but is still significantly slower compared to the modified scheme of Section 3 (Fig. 9).

In Fig. 10 we further compare the convergence behaviour of the hybrid schemes for a set of eight test cases (including cases I and II). Here we plot the number of inner-loop iterations required for each outer-loop step in the Occam scheme for the Lanczos bi-diagonalization with explicit orthogonalization, and for the modified multitransmitter scheme. Black filled symbols are used for the first scheme, and grey open symbols for the modified scheme,

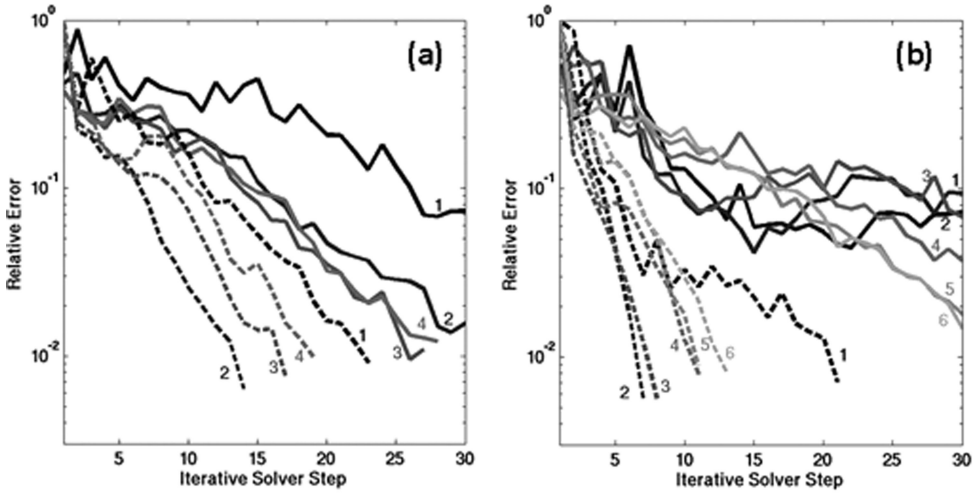

Figure 8. Convergence of the solution to the linear subproblem (5), for the hybrid scheme of Section 2 (solid lines) and for the modified hybrid scheme of Section 3 (dashed lines). Different line shadings correspond to different outer-loop Occam iterations, which are numbered near the end of each curve. Panels (a) and (b) give results for test cases I and II, respectively. In both cases the iterative solution is terminated when the relative error in the solution to eq. (5a; defined as $\|(\mathbf{J}\mathbf{J}^{\mathrm{T}} + \lambda_0\mathbf{I})\mathbf{b}_k - \hat{\mathbf{d}}\|/\|\hat{\mathbf{d}}\|$) drops below 10^{-2}, or k exceeds 30. In all cases the multifrequency scheme converges significantly faster than the standard CG iterative solution.

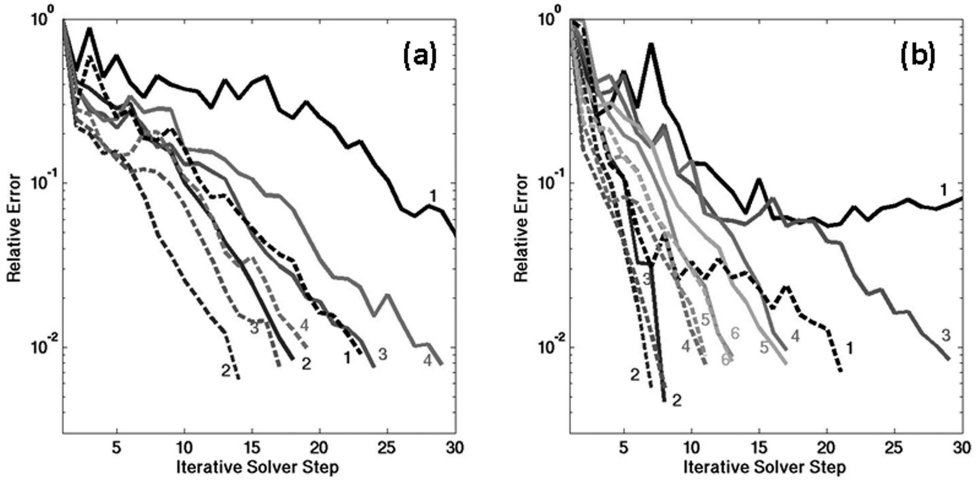

Figure 9. As in Fig. 8, but using a modified Lanczos bi-diagonalization scheme with explicit orthogonalization of all saved data-space vectors $\mathbf{u}_k$ for the standard hybrid scheme (solid lines). Dashed lines are as in Fig. 8. Explicit orthogonalization improves the convergence, although the multifrequency scheme still performs better.

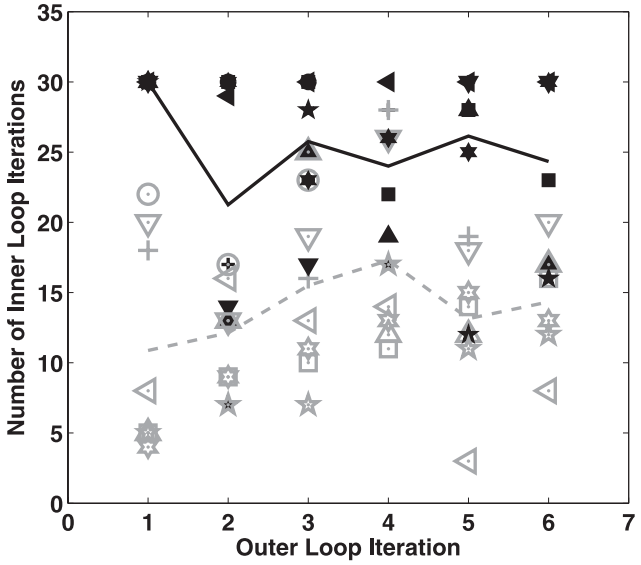

Figure 10. Total number of inner-loop iterations required for each outer-loop step in the Occam scheme for eight different synthetic model test cases. Different symbol styles are used for each case, with black filled and grey open symbols used for the standard (with explicit orthogonalization) and modified hybrid schemes, respectively. The lines give the averages (over test cases) for each outer-loop iteration: solid black line denotes standard hybrid scheme, dashed grey denotes modified scheme.

with different symbol styles used for each of the different synthetic model tests. The lines give the averages (over test cases) for each of the outer-loop iterations. The greatest increase in efficiency for the modified approach occurs on the first iteration, where the average number of steps decreases from 30 to 11. More modest, but still significant, improvement is seen for later iterations. Overall, the modified multitransmitter scheme reduces the total number of inner-loop iterations by a bit less than half, compared to Lanczos bi-diagonalization with explicit re-orthogonalization.

5 DISCUSSION AND CONCLUSIONS

We have discussed two hybrid schemes, which approximate the Occam scheme almost exactly without full calculation of the forward data mapping Jacobian. Both are based on the observation that iterative solution of the symmetric normal equations in the Gauss-Newton scheme effectively generates a sequence of sensitivities for different linear combinations of data, allowing construction of the Jacobian for a projection of the full data space. The Occam scheme can then be applied to this projected problem, with trade-off parameters chosen by assessing fit to the full data set. For EM geophysical problems with multiple transmitters (either multiple frequencies or source geometries) multiple forward solutions are required for a search step in the Lanczos process. Each of these solutions generates the sensitivity for a linear combination of data from the corresponding transmitter. From the perspective of the hybrid approach, with the Lanczos process generating an approximation to the full Jacobian, it is advantageous to save all of the component sensitivities, and use these to solve the projected problem in a larger subspace. This forms the basis for our second scheme, the modified hybrid algorithm.

Compared to standard CG schemes the proposed hybrid methods require substantially more storage, as the full sequence of data and model space vectors generated by the Lanczos process must be saved ($K(M+N)$ real numbers). For the modified approach storage requirements are even greater, as separate model space vectors are saved for each transmitter and each step in the solution process ($KJM + KN$ real numbers). However, as long as $KJ \ll N$ the additional memory required even for the modified scheme will be small compared to the MN real numbers required for storage of the full Jacobian. For our examples we have $KJ \approx 500$ while $N = 2560$.

Note also that a key component of the modified hybrid scheme is to explicitly solve the normal equations for the projected problem at each step in a modified Lanczos process, construct a 'trial' solution $\tilde{\mathbf{m}}$, and then apply the Jacobian $\mathbf{J}$ to this solution (i.e. compute $\mathbf{J}_j\tilde{\mathbf{m}}$, $j = 1, \ldots, J$) to generate the next set of data-space search vectors. Thus, additional computation is also required with the modified scheme [to compute $\tilde{\mathbf{m}}$; the equivalent multiplication by $\mathbf{J}$ is already required for the Lanczos process, e.g. in (8a)]. However, the projected system of normal eqs (12) or (20) will generally be small enough to be solved very rapidly—the largest system in our test cases was about 500 × 500, and the size of this system would not change significantly for a large 3-D inverse problem. Even so, this extra computation probably only makes sense when a single vector matrix multiply such as $\mathbf{J}^{\mathrm{T}}\mathbf{r}$ is sufficiently expensive, as it would be for something like the 3-D-MT inverse problem, where this single multiplication represents solving J independent 3-D PDEs. Indeed, for the 2-D-MT example we have used for illustration, solution of forward problems is sufficiently fast that justification for the modified scheme is at best marginal. Note also that one could apply the modified algorithm of Fig. 2 to any matrix $\mathbf{J}$, artificially divided into row blocks. However the extra computations required for each step of this scheme would in general overwhelm any saving due to reduction in the number of Lanczos steps that could be achieved.

The hybrid schemes described here are likely to be especially useful for joint inversion, for example, of MT and controlled source EM (e.g. Commer & Newman 2009), or EM and seismic traveltime data (e.g. Gallardo & Meju 2007). In the first place, multiple data types require running multiple forward models, and this can also be exploited within the framework developed here (as it was in the 2-D MT example, where TE and TM model solutions are computed). Furthermore, experience inverting multiple data types (e.g. Commer & Newman 2009) demonstrates that multiple trade-off parameters may be required to allow for differential weighting of disparate data types. And, one approach to joint inversion is to enforce structural similarity between two or more distinct physical parameters (e.g. conductivity and seismic velocity) by minimizing the norm of parameter gradient cross products (Gallardo & Meju 2004). Structural similarity defined in this way can be enforced by introducing another term into the penalty functional (1), with yet another adjustable weight. Efficient schemes for choosing these weights, as may be offered by hybrid schemes, are thus likely to prove valuable for joint inversion.

There are a number of potential extensions and refinements of the ideas presented here. First, we have focused on basic ideas, ignoring details that might make the schemes numerically more stable or efficient. For example, the cross product matrices in the projected normal eqs (12) and (20) need not be formed explicitly. Instead the singular value decomposition of the projected sensitivity matrix (e.g. $\mathbf{W}_K^{\mathrm{T}}\mathbf{J}$) could be used, both for efficient and stable solution of the normal equations, and to reduce storage requirements. And with the projected Jacobian saved, forming an approximation to the linearized resolution matrix would be straightforward (e.g. Minkoff 1996). In this application the additional sensitivity vectors provided by the modified scheme would improve the approximation of the

resolution matrix (e.g. see discussion on approximations to the resolution matrix in Deal & Nolet 1996). Another possible extension worth exploring would be to use the approximated Jacobian computed in a hybrid scheme as a preconditioner for the next outer loop iteration of the Occam inversion scheme.

Finally, we have focused on making the data-space Occam scheme efficient for even very large problems. The basic idea behind this scheme could be adapted to a more general truncated Gauss–Newton scheme. More generally, it would be worth considering how (or if) the individual transmitter gradient components generated in each evaluation of the penalty functional gradient might be used in other search algorithms such as NLCG or quasi-Newton.

ACKNOWLEDGMENTS

This work was supported by DOE grant DE-FG0302ER15318. I thank Weerachai Siripunvaraporn and an anonymous reviewer for comments that helped improved clarity of the manuscript.

REFERENCES

Alumbaugh, D.L. & Newman, G.A., 1997. Three-dimensional massively parallel electromagnetic inversion: II. Analysis of a cross-well electromagnetic experiment, *Geophys. J. Int.,* **128,** 355–363, doi:10.1111/j.1365246X.1997.tb01560.x.

Avdeev, D., 2005. Three-dimensional electromagnetic modeling and inversion from theory to application, *Surv. Geophys.,* **26,** 767–799.

Avdeev, D. & Avdeeva, A., 2009. 3D Magnetotelluric inversion using a limited-memory quasi-Newton optimization, *Geophysics,* **74,** F45–F57.

Commer, M. & Newman, G.A., 2008. New advances in three-dimensional controlled-source electromagnetic inversion, *Geophys. J. Int.,* **172,** 513–535.

Commer, M. & Newman, G.A., 2009. Three-dimensional controlled-source electromagnetic and magnetotelluric joint inversion, *Geophys. J. Int.,* **178,** 1305–1316, doi:10.1111/j.1365-246X.2009.04216.x.

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

Deal, M.M. & Nolet, G., 1996. Comment on 'Estimation of resolution and covariance for large matrix inversions' by J. Zhang and G. A. McMechan, *Geophys. J. Int.,* **127,** 245–250, doi:10.1111/j.1365-246X.1996.tb01548.x.

Dembo, R.S., Eisenstat, S.C. & Steihaug, T., 1982. Inexact Newton methods, *SIAM J. Numer. Anal.,* **19,** 400–408.

Egbert, G.D. & Kelbert, A., 2012. Computational recipes for electromagnetic inverse problems, *Geophys. J. Int.,* **189,** 251–267, doi:10.1111/j.1365-246X.2011.05347.x.

Egbert, G.D., Bennett, A.F. & Foreman, M.G.G., 1994. TOPEX/POSEIDON Tides estimated using a global inverse model, *J. geophys. Res.,* **99,** 24 821–24 852.

Eisenstat, S.C., Elman, H.C. & Schultz, M.H., 1983. Variational iterative methods for nonsymmetric systems of linear equations, *SIAM J. Numer. Anal.,* **20,** 345–357.

Gallardo, L.A. & Meju, M.A., 2004. Joint two-dimensional inversion with cross-gradient constraints, *J. geophys. Res.,* **109,** B03311, doi:10.1029/2003JB002716.

Gallardo, L.A. & Meju, M.A., 2007. Joint two-dimensional cross-gradient imaging of magnetotelluric and seismic traveltime data for structural and lithologic classification, *Geophys. J. Int.,* **169,** 1261–1272.

Gollub, G.H. & Van Loan, C., 1989. *Matrix Computations,* 2nd edn, Johns Hopkins University Press, Baltimore, MD.

Hanke, M., 2001. On Lanczos based methods for the regularization of discrete ill-posed problems, *BIT,* **41,** 1008–1018.

Kilmer, M.E. & O'Leary, D.P., 2001. Choosing regularization parameters in iterative methods for ill-posed problems, *SIAM J. Matrix Anal. Appl.,* **22,** 1204–1221.

Liu, D.C. & Nocedal, J., 1989. On the limited memory method for large scale optimization, *Math. Prog. B,* **45,** 503–528.

Mackie, R.L. & Madden, T.R., 1993. Conjugate direction relaxation solutions for 3-D magnetotelluric modeling, *Geophysics,* **58,** 1052–1057.

Minkoff, S.E., 1996. A computationally feasible approximate resolution matrix for seismic inverse problems, *Geophys. J. Int.,* **126,** 345–359.

Nash, S.G., 2000. A survey of truncated-Newton methods, *J. Comput. appl. Math.,* 45–59.

Newman, G.A. & Hoversten, G.M., 2000. Solution strategies for two- and three-dimensional electromagnetic inverse problems, *Inverse Probl.,* **16,** 1357, doi:10.1088/0266-5611/16/5/314.

O'Leary, D.P. & Simmons, J.A., 1981. A bidagonalization-regularization procedure for large scale discretization of ill-posed problems, *SIAM J. Sci. Statist. Comput.,* **2,** 474–489.

Oldenburg, D.W., McGillivray, P.R. & Ellis, R.G., 1993. Generalized subspace methods for large-scale inverse problems, *Geophys. J. Int.,* **114,** 12–20, doi:10.1111/j.1365-246X.1993.tb01462.x.

Paige, C.C. & Saunders, M.A., 1982a. LSQR: An algorithm for sparse linear equations and sparse least squares, *ACM Trans. Math. Softw.,* **8,** 43–71.

Paige, C.C. & Saunders, M.A., 1982b. Algorithm 583 LSQR: sparse linear equations and least squares problems, *ACM Trans. Math. Softw.,* **8,** 195–209.

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,** 229–249.

Parker, R.L., 1994. *Geophysical Inverse Theory,* Princeton University Press, Princeton, NJ.

Purser, R.J., Wu, W.-S., Parrish, D.F. & Roberts, N.M., 2003a. Numerical aspects of the application of recursive filters to variational statistical analysis. part I: spatially homogeneous and isotropic Gaussian covariances, *Monthly Wea. Rev.,* **131,** 1524–1535.

Purser, R.J., Wu, W.-S., Parrish, D.F. & Roberts, N.M., 2003b. Numerical aspects of the application of recursive filters to variational statistical analysis. part II: spatially inhomogeneous and anisotropic general covariances, *Monthly Wea. Rev.,* **131,** 1536–1548.

Rodi, W.L. & Mackie, R.L., 2001. Nonlinear conjugate gradients algorithm for 2-D Magnetotelluric inversion, *Geophysics,* **66,** 174–187.

Sasaki, Y., 2001. Full 3-D inversion of electromagnetic data on PC, *J. appl. Geophys.,* **46,** 45–54.

Siripunvaraporn, W., 2012. Three-dimensional magnetotelluric inversion: an introductory guide for developers and users, *Surv. Geophys.,* **33,** 5–27, doi:10.1007/s10712-011-9122-6.

Siripunvaraporn, W. & Egbert, G., 2000. An efficient data-subspace inversion method for 2-D magnetotelluric data, *Geophysics,* **65,** 791–803.

Siripunvaraporn, W. & Egbert, G.D., 2007. Data space conjugate gradient inversion for 2-D magnetotelluric data, *Geophys. J. Int.,* **170,** 986–994.

Siripunvaraporn, W. & Egbert, G., 2009. WSINV3DMT: Vertical magnetic field transfer function inversion and parallel implementation, *Phys. Earth planet. Inter.,* **173,** 317–329.

Siripunvaraporn, W. & Sarakorn, W., 2011. An efficient data space conjugate gradient Occam's method for three-dimensional magnetotelluric inversion, *Geophys. J. Int.,* **186,** 567–579, doi:10.1111/j.1365-246X.2011.05079.x.

Siripunvaraporn, W., Egbert, G., Lenburi, Y. & Uyeshima, M., 2005. Three-dimensional magnetotelluric inversion: data space method, *Phys. Earth planet. Inter.,* **140,** 3–14.

Tape, C., Liu, Q., Maggi, A. & Tromp, J., 2010. Seismic tomography of the southern California crust based on spectral-element and adjoint methods, *Geophys. J. Int.,* **180,** 433–462, doi:10.1111/j.1365-246X.2009.04429.x.

APPENDIX: INCLUDING MODEL AND DATA COVARIANCES

Here we briefly sketch treatment of the general form of the penalty functional (1) where model and data covariances are not the identity. We follow the approach used by Siripunvaraporn & Egbert (2000) where the model covariance $\mathbf{C_m} = \mathbf{C_m^{1/2}}\mathbf{C_m^{1/2}}$ is implemented as a

positive definite symmetric smoothing operator; applying half the smoothing steps essentially provides the square root of the operator. Examples of such covariance operators are given in Egbert *et al.* (1994), Siripunvaraporn & Egbert (2000), and Purser *et al.* (2003a,b). Defining a transformed model parameter $\tilde{\mathbf{m}}$ implicitly through

$$\mathbf{m} = \mathbf{C}_{\mathbf{m}}^{1/2}\tilde{\mathbf{m}} + \mathbf{m}_0, \tag{A1}$$

and transforming the data vector in the usual way as $\tilde{\mathbf{d}} = \mathbf{C}_{\mathbf{d}}^{-1/2}\mathbf{d} = \mathbf{C}_{\mathbf{d}}^{-1/2}\mathbf{f} + \mathbf{C}_{\mathbf{d}}^{-1/2}\varepsilon$ (so that the data error covariance is the identity), the Jacobian for the transformed problem can be written as

$$\tilde{\mathbf{J}} = \frac{\partial \tilde{\mathbf{f}}}{\partial \tilde{\mathbf{m}}} = \mathbf{C}_{\mathbf{d}}^{-1/2}\frac{\partial \mathbf{f}}{\partial \mathbf{m}}\mathbf{C}_{\mathbf{m}}^{1/2} = \mathbf{C}_{d}^{-1/2}\mathbf{J}\mathbf{C}_{\mathbf{m}}^{1/2}, \tag{A2}$$

where $\mathbf{J}$ is the original sensitivity for the untransformed problem. Dropping the tildes the penalty functional (1) is reduced to the simpler form used throughout the paper, and the methods described can be applied to invert the transformed data for the transformed model parameter $\tilde{\mathbf{m}}$. This can be converted back to the physical model parameter using (A1). Note that only multiplication by the model covariance operator $\mathbf{C}_{\mathbf{m}}^{1/2}$ and the inverse data error covariance square root $\mathbf{C}_{\mathbf{d}}^{-1/2}$ are required; the inverse model covariance is never directly used. Although we have focused on data space solution methods, the same approach can be used for model space solution approaches, such as NLCG—that is, the penalty functional can be minimized with respect to $\tilde{\mathbf{m}}$, with the gradient derived from the transformed Jacobian $\tilde{\mathbf{J}}$.

The principal limitation of the approach described here is that multiplication by $\mathbf{C}_{\mathbf{m}}^{1/2}$ must be implemented, and for some classical regularization operators this may not be so straightforward. For example, if the regularization term is taken to be $\|\nabla^2\mathbf{m}\|^2$, applying the smoothing operator $\mathbf{C}_{\mathbf{m}}^{1/2}$ amounts to solving Poisson's equation; boundary conditions are a complicating, although not insurmountable, issue. Such details are beyond the scope of this paper.