REBOCC: An Efficient Data-Subspace Inversion for Two-Dimensional Magnetotelluric Data

Weerachai Siripunvaraporn and Gary Egbert*

(submitted to *Geophysics* January 15, 1999)

ABSTRACT

There are currently three types of algorithms in use for regularized 2-D inversion of magnetotelluric (MT) data. All seek to minimize some functional which penalizes data misfit and model structure. With the most straightforward approach (exemplified by OCCAM) the minimization is accomplished using some variant on a linearized Gauss-Newton approach to minimization of A second approach is to use a descent method (e.g., non-linear conjugate gradients, NLCG) to avoid the expense of constructing large matrices (e.g., the sensitivity matrix). Finally, approximate methods (e.g., RRI) have been developed which use cheaply computed approximations to the sensitivity matrix to search for a minimum of the penalty functional. Approximate approaches can be very fast, but in practice often fail to converge without significant expert user intervention. On the other hand, the more straightforward methods can be prohibitively expensive to use for even moderate sized data sets. Here we present a new and much more efficient variant on the OCCAM scheme. By expressing the solution as a linear combination of rows of the sensitivity matrix smoothed by the model covariance (the "representers") we transform the linearized inverse problem from the M-dimensional model space to the N-dimensional data space. Since generally $N << M$, this transformation by itself can result in significant computational saving. More importantly the data space formulation suggests a simple approximate method for constructing the inverse solution. Since MT data are smooth and "redundant", a subset of the representers is typically sufficient to form the model without significant loss of detail. Computations required for constructing sensitivities and the size of matrices to be inverted can be significantly reduced by this approximation. We refer to this inversion as REBOCC, the reduced basis OCCAM's inversion (REBOCC). Numerical experiments on synthetic and real data sets with REBOCC, NLCG, RRI and OCCAM show that REBOCC is significantly faster than the NLCG or OCCAM. In terms of speed REBOCC is not competitive with RRI. However, even with a simple synthetic data set we could not always get RRI to converge to a reasonable solution. The basic idea behind REBOCC should be more broadly applicable, in particular to 3-D MT inversion.

INTRODUCTION

The Magnetotelluric (MT) method for imaging crustal and upper mantle electrical conductivity has found increasing use in both geophysical exploration application (Orange, 1989; Vozoff, 1972) and in fundamental studies of large scale tectonics (Jones, 1992; Wannamaker et al., 1994; Chen et al., 1996; Unsworth et al., 1999). Initial applications of MT were based on local 1-D interpretations, for which theories (Weidelt, 1972; Parker, 1980) and inversion methods (Jupp and Vozoff, 1975; Constable et al., 1987; Smith and Booker, 1988) are well developed. It is now clear that two or even three dimensional interpretation is essential for most real MT data sets. Over the past decade very substantial progress has been made on the development of 2-D inversion methods. These have included straightforward extensions of linearized search methods developed previously for 1-D regularized inversion (deGroot-Hedlin and Constable, 1990; Uchida, 1993), efficient approximate methods (Smith and Booker, 1991; Farquharson and Oldenburg, 1996), the subspace method (Oldenburg et al., 1993) and methods based on direct iterative minimization of a regularized penalty functional (Rodi and Mackie, 1999). These program are freely available, and are widely used for interpretation of modern MT surveys. However the available inversion algorithms are not without flaws. The fastest (RRI) intrinsically limit the model space search and can fail to converge (without substantial user intervention); the most general and flexible (e.g. OCCAM) run very slowly and require substantial memory.

We begin our discussion with a review of 2-D MT inversion methods including a data space variant on OCCAM. We then introduce our inversion algorithm which is stable, flexible and general, and much faster than most previously proposed schemes. We demonstrate the stability and efficiency of our approach by inverting synthetic MT data, comparing our results to those obtained with other methods, including RRI (Smith and Booker, 1991) and NLCG

*College of Oceanic and Atmospheric Sciences, Oregon State University, Corvallis, OR 97331

(Rodi and Mackie, 1999). Finally we briefly consider an example application to field data from a dense profile across the San Andreas fault near Parkfield, CA (Unsworth et al., 1997).

OVERVIEW OF INVERSION METHODS

We begin with a broad overview of previously developed 2-D MT inversion methods. To be explicit we consider the Earth as discretized into a series of M constant resistivity blocks, $\mathbf{m} = [m_1, m_2, \ldots, m_M]$. There are N observed data $\mathbf{d} = [d_1, d_2, \ldots, d_N]$, with estimated uncertainties $\mathbf{e} = [e_1, e_2, \ldots, e_N]$. The fit of the theoretical model responses $\mathbf{F}[\mathbf{m}]$ to the observational data can be expressed as

$$X_d^2 = (\mathbf{d} - \mathbf{F}[\mathbf{m}])^T \mathbf{C_d}^{-1} (\mathbf{d} - \mathbf{F}[\mathbf{m}]), \tag{1}$$

where the superscript T represents matrix transpose and $\mathbf{C_d}$ is the data covariance matrix, which in practice is diagonal.

Because of the non-uniqueness of the inverse problem, an infinite number of models can produce the same misfit. Most modern MT inversion schemes resolve this non-uniqueness by seeking models that have minimum possible structure (in some sense) for a given level of misfit (Parker, 1994). This makes the inversion stable, with resulting models less likely to contain spurious features.

To quantify "model structure", we consider a model norm of the general form

$$X_m^2 = (\mathbf{m} - \mathbf{m_0})^T \mathbf{C_m}^{-1} (\mathbf{m} - \mathbf{m_0}), \tag{2}$$

where $\mathbf{m_0}$ is a base (or prior) model, and $\mathbf{C_m}$ is a model covariance matrix which characterizes the expected magnitude and smoothness of resistivity variations relative to $\mathbf{m_0}$. Other approaches to minimum structure inversion are similar, though in some cases $\mathbf{C_m}^{-1}$ is replaced by a model roughness operator. The minimum structure inverse problem is to minimize X_m^2 subject to $X_d^2 = X_*^2$, where X_*^2 is the desired level of misfit.

To solve this minimization problem, a Lagrange multiplier λ^{-1} can be introduced, resulting in an unconstrained functional $U(\mathbf{m}, \lambda)$

$$U(\mathbf{m}, \lambda) = (\mathbf{m} - \mathbf{m_0})^T \mathbf{C_m}^{-1} (\mathbf{m} - \mathbf{m_0}) + \lambda^{-1} \{ (\mathbf{d} - \mathbf{F}[\mathbf{m}])^T \mathbf{C_d}^{-1} (\mathbf{d} - \mathbf{F}[\mathbf{m}]) - X_*^2 \}, \tag{3}$$

for which we seek stationary points (with respect to both $\mathbf{m}$ and λ). Alternatively, we may consider the penalty functional $W_\lambda(\mathbf{m})$

$$W_\lambda(\mathbf{m}) = (\mathbf{m} - \mathbf{m_0})^T \mathbf{C_m}^{-1} (\mathbf{m} - \mathbf{m_0}) + \lambda^{-1} \{ (\mathbf{d} - \mathbf{F}[\mathbf{m}])^T \mathbf{C_d}^{-1} (\mathbf{d} - \mathbf{F}[\mathbf{m}]) \}. \tag{4}$$

In (4), λ acts to "trade-off" between minimizing the norm of data misfit and the norm of the model (Parker, 1994). When λ is large, the data misfit is de-emphasized, leading to a smoother model. In contrast, as $\lambda \to 0$, the inverse problem becomes closer to the ill conditioned least-square inversion problem, resulting in an erratic model (see Parker, 1980).

Note that both U and W have the same stationary points with respect to variation of the model, i.e. $\partial U / \partial \mathbf{m} = \partial W / \partial \mathbf{m}$, where λ is fixed. Parker (1994) uses this to show that stationary points of (3) can be found by minimizing (4) for a series of λ values, and then choosing λ so that the misfit satisfies the constraint $X_d^2 = X_*^2$.

For linear $\mathbf{F}[\mathbf{m}]$ this is straightforward, since in this case (for fixed λ) $\partial U / \partial \mathbf{m} = 0$ is a linear system of equations which may be solved for $\mathbf{m}$. Because $\mathbf{F}[\mathbf{m}]$ is non-linear, iterative solution methods are required. We briefly consider some of the approaches which have been taken by previous workers, and then outline our approach.

Rodi and Mackie (1999) provide a good review of several approaches, including a straightforward Gauss-Newton (GN) method. This approach is based on linearizing $\mathbf{F}[\mathbf{m}]$ with a Taylor series expansion,

$$\mathbf{F}[\mathbf{m}_{k+1}] = \mathbf{F}[\mathbf{m}_k + \delta \mathbf{m}] = \mathbf{F}[\mathbf{m}_k] + \mathbf{J}_k (\mathbf{m}_{k+1} - \mathbf{m}_k), \tag{5}$$

where k denotes iteration number, and $\mathbf{J}_k = \left. \frac{\partial \mathbf{F}}{\partial \mathbf{m}} \right|_{\mathbf{m}_k}$ is the $N \times M$ sensitivity matrix calculated at $\mathbf{m}_k$. Calculation of $\mathbf{J}_k$, which describes the perturbations in the data due to changes in the model, is described in detail by Rodi and Mackie (1999), Mackie and Madden (1993) and Rodi (1978). Substituting (5) in (4), we obtain

$$\begin{aligned} \widetilde{W} = {} & (\mathbf{m}_{k+1} - \mathbf{m_0})^T \mathbf{C_m}^{-1} (\mathbf{m}_{k+1} - \mathbf{m_0}) + \\ & \lambda^{-1} \{ (\hat{\mathbf{d}}_k - \mathbf{J}_k (\mathbf{m}_{k+1} - \mathbf{m_0}))^T \mathbf{C_d}^{-1} (\hat{\mathbf{d}}_k - \mathbf{J}_k (\mathbf{m}_{k+1} - \mathbf{m_0})) \}, \end{aligned} \tag{6}$$

where $\hat{\mathbf{d}}_k = \mathbf{d} - \mathbf{F}[\mathbf{m}_k] + \mathbf{J}_k(\mathbf{m}_k - \mathbf{m_0})$. This $\widetilde{W}$ is then quadratic in $\mathbf{m}_{k+1}$ and thus can be minimized exactly (for fixed λ). For numerical stability (Marquardt, 1963), an extra damping parameter ϵ is generally required to control step size for each iteration. The system of equation to be solved for each iteration then becomes

$$\mathbf{m}_{k+1} - \mathbf{m}_k = \left[\lambda \mathbf{C}_{\mathbf{m}}^{-1} + \mathbf{\Gamma}_k^{\mathbf{m}} + \epsilon_k \mathbf{I}\right]^{-1} \left[\mathbf{J}_k^T \mathbf{C}_{\mathbf{d}}^{-1}(\mathbf{d}_k - \mathbf{F}[\mathbf{m}_k]) - \lambda \mathbf{C}_{\mathbf{m}}^{-1}(\mathbf{m}_k - \mathbf{m_0})\right], \tag{7}$$

where the "model space cross-product" matrix $\mathbf{\Gamma}_k^{\mathbf{m}} = \mathbf{J}_k^T \mathbf{C}_{\mathbf{d}}^{-1} \mathbf{J}_k$ is an $M \times M$ positive semi-definite symmetric matrix, and $\mathbf{I}$ is the identity matrix. Note that with the GN approach λ is fixed. Therefore, the algorithm will converge to a stationary point of (4), not (3). To achieve the stationary point of (3) (with respect to both λ and $\mathbf{m}$), the process would have to be repeated with different values of λ until the constraint $X_d^2 = X_*^2$ was satisfied.

The OCCAM approach, first proposed by Constable et al. (1987) (see also Parker, 1994; deGroot-Hedlin and Constable, 1993; Uchida, 1993), is also based on linearizing $\mathbf{F}[\mathbf{m}]$ and then solving for the stationary points of (6). Differentiating (6) with respect to $\mathbf{m}$ and setting the result to zero leads to an iterative sequence of approximate solutions

$$\mathbf{m}_{k+1}(\lambda) = \left[\lambda \mathbf{C}_{\mathbf{m}}^{-1} + \mathbf{\Gamma}_k^{\mathbf{m}}\right]^{-1} \mathbf{J}_k^T \mathbf{C}_{\mathbf{d}}^{-1} \hat{\mathbf{d}}_k + \mathbf{m_0}. \tag{8}$$

The unique feature of the OCCAM approach is that the parameter λ is used in each iteration both as a step length control and a smoothing parameter. That is, (8) is solved for a series of trial values of λ and the misfit $X_d^2(\mathbf{m}_{k+1}(\lambda))$ for each λ is evaluated by solving the 2-D forward problem. As for the linear problem, λ should be chosen so that the condition $X_d^2 = X_*^2$ is met. Usually, in the early iterations, the true misfit X_d^2 is higher than the desired X_*^2 for all possible λ. The OCCAM process thus chooses the model with the minimum misfit as the basis for the next iteration. The process is then repeated until the misfit reaches the desired level. Parker (1994) called this process of bringing the misfit down to the target level Phase I. Once the misfit reaches the desired level, Phase II begins by keeping the misfit at the desired level, but varying λ to search for the model with smallest norm. Since the problem is non-linear, the desired misfit may never be reached. However, in practice, improvement of the misfit from iteration to iteration can be expected, until a minimum is achieved.

Both GN and OCCAM share similar computational steps. For each iteration, $\mathbf{J}_k$ must be calculated, and an $M \times M$ system of equations, (7) for GN and (8) for OCCAM, must be solved. These methods are thus very time-consuming (e.g. Smith and Booker, 1991; Rodi and Mackie, 1999). Furthermore, these methods require much memory to store the sensitivity and cross-product matrices. These computational inefficiencies are the result of strongly dependence on the model space dimension M.

Several approaches have been proposed to avoid the heavy computational burden of the direct linearized search schemes. One approach is to use approximate sensitivities to eliminate calculation of the full sensitivity matrix. A good example in this category is the Rapid Relaxation Inversion (RRI) introduced by Smith and Booker (1991). RRI turns the 2-D inverse problem into a series of 1-D inverse problems, by computing the approximate sensitivity of data at each site to variations of resistivity directly below the site. The model is updated by solving a series of 1-D inverse problems and interpolating horizontally to form the 2-D resistivity model. Fit to the data is tested with a full 2-D forward calculation, and step length is adjusted if necessary. The process is repeated until the misfit condition is met. Note that this approach eliminates both the 2-D sensitivity calculation, and the need to solve a large $M \times M$ system of equations. Generally, RRI requires many iterations but overall is very fast. RRI can handle very large 2-D data sets, and has been applied to interpretation of many MT data sets. Oldenburg and Ellis (1991) suggested a very similar approach based on using a series of 1-D inversions as an approximate inverse mapping (AIM) to map the data back to model space, followed by full calculation of the 2-D forward problem to assess model fit.

The efficiency of approximate inversion schemes comes at a price. Because of the incomplete search of the model space, schemes based on 1-D inversions can be insensitive to features that are not directly beneath the locations of measurement (Farquharson and Oldenburg, 1996). For example, if the data is dominated by a significant feature outside of the profile, the inversion may have difficulty finding any models which actually fit the data (e.g. Unsworth et al., 1999), or the inversion may insert a geologically unreasonable feature beneath the profile. In addition, inversion of vertical magnetic transfer functions is difficult with this approach, since vertical magnetic fields are sensitive to nearby structures, rather than features directly beneath the sites.

Farquharson and Oldenburg (1996) proposed a somewhat different approximate approach based on using 2-D sensitivities of a homogeneous or layered half-space, instead of the exact sensitivities. This scheme eliminates the need for calculation of $\mathbf{J}_k$ but still requires inversion of large ($M \times M$) matrices. However, in many cases these simple approximations to the sensitivities are good enough to allow convergence of the inversion to an acceptable level of misfit. The effectiveness of this scheme depends on many factors: the complexity of structure of the true model, the closeness of the structure to the data sites and the magnitude of the resistivity contrasts.

Another way to reduce the computational burden in 2-D MT inversion is provided by the subspace approach of Oldenburg et al. (1993): the inverse solution is sought in a low dimensional subspace of the original M-dimensional model space. The success of this approach depends greatly on the choice of subspace basis vectors. Unfortunately, the proper choice is often not obvious, and a bad selection can lead to a poor solution. A combination of both approximate sensitivities and the subspace approach has been used by Oldenburg and Ellis (1993). Current implementation of OCCAM also allows for a simple sort of subspace approach, since the resistivity can be parameterized on a grid coarser than that used for numerical computations.

Rather than approximate the sensitivities, or impose prior restrictions on the model space, Mackie and Madden (1993) considered an approach based on a conjugate gradient (CG) relaxation solution of the linearized normal equations derived from (6). With the relaxation method, the actual computation of the sensitivity matrix can be avoided by using the fact that evaluating the gradient of the linearized penalty functional requires only one forward solution (per period) with a distributed set of sources either in the volume or on the surface. CG thus significantly reduces the computational requirements (both cpu time and memory) for each iteration, making attempts at even 3-D MT inversion feasible (Mackie and Madden, 1993). The model is updated for each iteration and the CG relaxation solution process is repeated until the stationary point of (4) is reached. Note however that since CG is a descent method which does not directly use any information about curvature of the penalty functional, many iterations may be required compared to GN or OCCAM. Also, as for the GN approach the entire process must be repeated for different values of λ to find a true minimum structure model which achieves a specified misfit X_*^2.

Rodi and Mackie (1999) considered a variant on this CG approach. These authors applied non-linear conjugate gradients (NLCG) directly to minimization of (4), with λ fixed. Similar to the CG method (Mackie and Madden, 1993), NLCG requires only a few forward solutions (per period) in each line minimization step. To improve the convergence rate, a simple preconditioner is used. Again, to obtain a norm minimizing solution with minimum structure at the desired misfit, one needs to minimize (4) for various values of λ. Rodi and Mackie (1999) show that the two descent methods (CG and NLCG) are comparable, and both are much more efficient than the GN method in terms of cpu time and memory requirements.

Here, we develop a variant on the OCCAM approach which is significantly more efficient than previously proposed methods. We begin by transforming the inverse problem from the model space into the data space, by expressing the solution as a linear combination of rows of the sensitivity matrix smoothed by the model covariance. This transformation reduces the size of the system of equations to be solved from $M \times M$ to $N \times N$. Since the number of model parameters M is often much larger than the number of data N, a significant decrease in both cpu time and memory can be achieved with this approach. More importantly, the data space formulation leads naturally to a simple approximation which can result in very significant computational savings in most cases.

Generally, MT data are smooth (in period, and for closely spaced sites, in space) and "redundant". Therefore, in the data space approach, there is no need to use all of the sensitivities as basis functions. A subset is typically sufficient to construct the model without significantly loss of detail. With this approximation it is unnecessary to compute all sensitivities, and the size of the system of equations that must be solved can be significantly reduced. We call this approach the **RE**duced **B**asis **OCC**am's (REBOCC) Inversion. Note that even though we construct the solution from subset of the smoothed sensitivities, the goal of the inversion remains to find the norm minimizing model subject to fitting all of the data well enough. As we shall discuss in more detail below, in the data space the choice of basis functions is very natural and is dictated by what features can be resolved by the available data. This is in contrast to the choice of a model subspace (Oldenburg et al., 1993), where the choice of subspace is rather arbitrary.

With careful implementation of forward modeling and sensitivity calculations, REBOCC runs in a fraction of the time required by methods such as GN or OCCAM and is also considerably faster than NLCG. In addition, memory requirements are significantly reduced so that large data sets can be inverted with REBOCC on a standard workstation. The basic idea behind REBOCC generalizes readily to the 3-D case.

THE DATA SPACE OCCAM METHOD (DASOCC)

Parker (1994) shows that the minimizer of (6) for iteration k can be expressed as a linear combination of rows of the smoothed sensitivity matrix $\mathbf{C_m}\mathbf{J}_k^T$,

$$\mathbf{m}_{k+1} - \mathbf{m_0} = \mathbf{C_m}\mathbf{J}_k^T\boldsymbol{\beta}_{k+1} = \sum_{j=1}^{N} \boldsymbol{\Upsilon}_{kj}\beta_{k+1,j} \tag{9}$$

where $\boldsymbol{\beta}_{k+1}$ is an unknown expansion coefficient vector of the basis functions $\boldsymbol{\Upsilon}_{kj}; j = 1, \ldots, N$. The basis vectors $\boldsymbol{\Upsilon}_{kj}$ are sometimes referred to as the "representers" of the linearized data functionals for iteration k (e.g., Parker, 1994). Substituting (9) into (6), we obtain

$$\widetilde{W} = \boldsymbol{\beta}_{k+1}^T\boldsymbol{\Gamma}_k^{\mathbf{n}}\boldsymbol{\beta}_{k+1} + \lambda^{-1}\{(\hat{\mathbf{d}}_k - \boldsymbol{\Gamma}_k^{\mathbf{n}}\boldsymbol{\beta}_{k+1})^T\mathbf{C_d}^{-1}(\hat{\mathbf{d}}_k - \boldsymbol{\Gamma}_k^{\mathbf{n}}\boldsymbol{\beta}_{k+1})\}. \tag{10}$$

Here $\mathbf{\Gamma}_k^{\mathbf{n}} = \mathbf{J}_k \mathbf{C_m} \mathbf{J}_k^T$ is the $N \times N$ "data space cross-product" matrix, which is again symmetric and positive semi-definite. Differentiating (10) with respect to $\boldsymbol{\beta}$ and rearranging, the unknown expansion coefficients can be obtained as

$$\boldsymbol{\beta}_{k+1} = (\lambda \mathbf{C_d} + \mathbf{\Gamma}_k^{\mathbf{n}})^{-1} \hat{\mathbf{d}}_k. \tag{11}$$

The inverse problem thus becomes a search for the N real expansion coefficients $\boldsymbol{\beta}_{k+1}$, instead of the M dimensional model, $\mathbf{m}_{k+1}$. Exactly as for the standard Occam, we can solve for $\boldsymbol{\beta}_{k+1}$, update the model, and then check the misfit for various values of λ. We again choose λ to achieve the minimum misfit if this exceeds the desired level X_*^2 (Phase I), and use this model as the basis for the next iteration. Once the desired misfit is achieved, Phase II begins to wipe out unnecessary features, while keeping the misfit at the desired level.

We emphasize here that we have only transformed the inverse problem solution method from the model space to the data space. Solutions obtained in both spaces will be identical if we choose all parameters (i.e., λ and $\mathbf{C_m}$) the same. For brevity we refer to this variant on Occam as the **Da**ta **S**pace **OCC**am (DASOCC) inversion. Note that a data space approach was also used by Smith and Booker (1988) in their treatment of the 1-D MT inverse problem.

The data space formulation offers several advantages. The most obvious is the reduction in the dimension of the system of equations which must be solved ($N \times N$ in the data space instead of $M \times M$ in the model space). Generally, $N << M$. This will be particularly true for the 3-D case.

Also, calculation in the model space requires $\mathbf{C_m}^{-1}$. Since it is not practical to specify a full $M \times M$ model covariance matrix $\mathbf{C_m}$ and then compute the inverse, $\mathbf{C_m}^{-1}$ is replaced by the first derivative roughness penalty in deGroot-Hedlin and Constable (1990). In the data space approach $\mathbf{C_m}$ is required, not its inverse. This offers some advantages since the model covariance can be readily used to include prior information such as an ocean, or faults which should be fixed in the model. We discuss these issues in more detail in the model covariance section.

REDUCED DATA SPACE OCCAM APPROACHES (REBOCC)

The data space formulation clearly shows that the solution is a linear combination of natural basis functions or representers. Each representer corresponds to a single data element (at a particular period, station, response and mode). Just as the MT data should be smooth and redundant, the representers vary slowly with period and site location for a given response and mode.

Figure 1 shows a plot of the representers of the apparent resistivity (TM mode) for three nearby periods. The plot reveals a smooth and, in fact, insignificant change from one period to another. These basis functions are thus highly redundant, so that an excellent approximation to the solution can be found in a subspace of much lower dimension (Parker and Shure, 1982; see also Parker, 1994). This simple but critical concept of data redundancy can significantly speed up the inversion while also substantially decreasing memory requirements.

Prior to solving the inverse problem, we will select a subset of L (out of N) data for which representers will be calculated at each iteration. As a simple example, we could choose all data for every other period (so $L = N/2$). Note that, L can typically be considerably smaller than this, as we shall show later. For iteration $k+1$, we seek solutions of the form,

$$\mathbf{m}_{k+1} = \mathbf{C_m} \mathbf{G}_k^T \boldsymbol{\alpha}_{k+1} + \mathbf{m_0}, \tag{12}$$

where $\boldsymbol{\alpha}_{k+1}$ is the L-dimensional unknown coefficient vector for the reduced basis, and $\mathbf{G}_k$ is the $L \times M$ subset sensitivity matrix.

To fit all of the data and to derive equations for $\boldsymbol{\alpha}_{k+1}$ analogous to (10) and (11), we require the linearized relationship between $\delta\mathbf{m}$ and $\hat{\mathbf{d}}_k$. In fact, we do not strictly have this relationship, unless we calculate all of the sensitivities. However, the data varies smoothly, and so a data value would be well approximated by interpolation of "nearby" data (e.g., adjacent frequencies from the same site). In the same way, sensitivities vary smoothly with frequency and/or site location (Fig. 1) and can be interpolated from nearby sensitivities. We thus express the approximation to the full sensitivity matrix $\mathbf{J}_k$ in terms of the subset sensitivity matrix $\mathbf{G}_k$ using an interpolation matrix, $\mathbf{B}$ of size $N \times L$, that is ,

$$\mathbf{J}_k \approx \mathbf{B} \mathbf{G}_k. \tag{13}$$

The interpolation matrix does not need to be very sophisticated. Recall that Farquharson and Oldenburg (1996) have had success in 2-D MT inversion using a very simple sensitivity matrix generated from either a homogeneous or layered half-spaced, while RRI uses only 1-D approximate sensitivities. By comparison, the approximate sensitivity matrix of (13) generated by even a crude interpolation of a sparse subset of representers will actually be quite close to the exact sensitivity matrix $\mathbf{J}_k$. Further details of the interpolation are given in the interpolation matrix section.

Substituting (12) and (13) into (6), we find

$$\widetilde{W} = \boldsymbol{\alpha}_{k+1}^T \boldsymbol{\Gamma}_k^{\mathbf{l}} \boldsymbol{\alpha}_{k+1} + \lambda^{-1}\{(\hat{\mathbf{d}}_k - \mathbf{B}\boldsymbol{\Gamma}_k^{\mathbf{l}}\boldsymbol{\alpha}_{k+1})^T \mathbf{C}_{\mathbf{d}}^{-1}(\hat{\mathbf{d}}_k - \mathbf{B}\boldsymbol{\Gamma}_k^{\mathbf{l}}\boldsymbol{\alpha}_{k+1})\}, \tag{14}$$

where $\boldsymbol{\Gamma}_k^{\mathbf{l}} = \mathbf{G}_k \mathbf{C}_{\mathbf{m}} \mathbf{G}_k^T$ is the $L \times L$ "data subspace cross-product" matrix. Following Egbert et al. (1994), we decompose $\mathbf{C}_{\mathbf{d}}^{-\frac{1}{2}}\mathbf{B}$ into the $N \times N$ orthonormal matrix $\mathbf{Q}$ where $\mathbf{Q}^T = \mathbf{Q}^{-1}$, and the $N \times L$ matrix $\mathbf{R}$

$$\mathbf{C}_{\mathbf{d}}^{-\frac{1}{2}}\mathbf{B} = \mathbf{Q}\mathbf{R} = \bar{\mathbf{Q}}\bar{\mathbf{R}}, \tag{15}$$

$$\text{where} \qquad \mathbf{Q} = [\ \bar{\mathbf{Q}} \mid \bar{\mathbf{Q}}_{\mathbf{0}}\ ] \qquad \text{and} \qquad \mathbf{R} = \begin{bmatrix} \bar{\mathbf{R}} \\ \mathbf{0} \end{bmatrix}.$$

Matrices $\bar{\mathbf{Q}}$ and $\bar{\mathbf{Q}}_{\mathbf{0}}$ have dimensions $N \times L$ and $N \times N - L$ respectively. Matrix $\bar{\mathbf{R}}$ is the square $L \times L$ upper triangular matrix, and $\mathbf{0}$ is the $N - L \times L$ zero matrix, i.e., all elements are zeros.

Equation (14) becomes

$$\widetilde{W} = \bar{\boldsymbol{\alpha}}_{k+1}^T \bar{\boldsymbol{\Gamma}}_k^{\mathbf{l}} \bar{\boldsymbol{\alpha}}_{k+1} + \lambda^{-1}\{(\mathbf{C}_{\mathbf{d}}^{-\frac{1}{2}}\hat{\mathbf{d}}_k - \mathbf{Q}\bar{\boldsymbol{\Gamma}}_k^{\mathbf{l}}\bar{\boldsymbol{\alpha}}_{k+1})^T(\mathbf{C}_{\mathbf{d}}^{-\frac{1}{2}}\hat{\mathbf{d}}_k - \mathbf{Q}\bar{\boldsymbol{\Gamma}}_k^{\mathbf{l}}\bar{\boldsymbol{\alpha}}_{k+1})\} \tag{16}$$

$$\text{where} \qquad \bar{\boldsymbol{\alpha}}_{k+1} = (\bar{\mathbf{R}}^{-1})^T \boldsymbol{\alpha}_{k+1} \tag{17}$$

$$\text{and} \qquad \bar{\boldsymbol{\Gamma}}_k^{\mathbf{l}} = \bar{\mathbf{R}}\boldsymbol{\Gamma}_k^{\mathbf{l}}\bar{\mathbf{R}}^T. \tag{18}$$

Inserting $\mathbf{Q}\mathbf{Q}^T = \mathbf{I}$ in between $\bar{\boldsymbol{\Gamma}}_k^{\mathbf{l}}$ and $\bar{\boldsymbol{\alpha}}_{k+1}$ on the right side of (16) and rearranging, this can be rewritten as

$$\widetilde{W} = \bar{\boldsymbol{\alpha}}_{k+1}^T \bar{\boldsymbol{\Gamma}}_k^{\mathbf{l}} \bar{\boldsymbol{\alpha}}_{k+1} + \lambda^{-1}\{X_{min}^2 + (\bar{\mathbf{d}}_k - \bar{\boldsymbol{\Gamma}}_k^{\mathbf{l}}\bar{\boldsymbol{\alpha}}_{k+1})^T(\bar{\mathbf{d}}_k - \bar{\boldsymbol{\Gamma}}_k^{\mathbf{l}}\bar{\boldsymbol{\alpha}}_{k+1})\}, \tag{19}$$

where $\bar{\mathbf{d}}_k = \bar{\mathbf{Q}}^T \mathbf{C}_{\mathbf{d}}^{-\frac{1}{2}}\hat{\mathbf{d}}_k$, and $X_{min}^2 = ||\bar{\mathbf{Q}}_{\mathbf{0}}\mathbf{C}_{\mathbf{d}}^{-\frac{1}{2}}\hat{\mathbf{d}}_k||^2 = ||\mathbf{C}_{\mathbf{d}}^{-\frac{1}{2}}\hat{\mathbf{d}}_k||^2 - ||\bar{\mathbf{Q}}^T\mathbf{C}_{\mathbf{d}}^{-\frac{1}{2}}\hat{\mathbf{d}}_k||^2$ is the approximate minimum achievable total square misfit for the selected basis. If we use all representers (i.e. $\mathbf{B} = \mathbf{I}$), then $X_{min}^2 = 0$. This corresponds to the fact that for a linear problem we can fit the data exactly if we use all representers. This will not be true for the non-linear MT problem. Thus X_{min}^2 only provides a very rough estimate of the magnitude of data misfit that might be achieved with the chosen reduced basis. In general, X_{min}^2 is high in the early iterations, and decreases to a constant in the later iterations.

Differentiating (19) with respect to $\bar{\boldsymbol{\alpha}}$ and setting the result to zero, the unknown expansion coefficients can be obtained in a form similar to (11) ,

$$\bar{\boldsymbol{\alpha}}_{k+1} = (\lambda\mathbf{I} + \bar{\boldsymbol{\Gamma}}_k^{\mathbf{l}})^{-1}\bar{\mathbf{d}}_k. \tag{20}$$

Again, just as in the model and the data space methods, after solving (20), we update the model using (12) and (17), then solve the forward problem to evaluate X_d^2. The procedure is repeated to find the appropriate λ. The outer loop of the iterative minimization of (4) proceeds exactly as for OCCAM or for DASOCC.

Representer Subsets for REBOCC

The success of the data subspace approach depends to some extent on the selection of data points which determine the representers used in the model expansion. Clearly it is necessary to select the basis to uniformly cover the full data set, so that the simple interpolation scheme used here is effective. Beyond this basic criteria, the optimal choice of data subsets is an issue that needs further study. Here we offer some simple schemes based on our experience so far.

Two classes of data subsets which generally seem to work well are shown in Figure 2. In the first example, every p*th* period is chosen for all sites ("p*th*-stripe" pattern). This pattern is safe to apply in almost all cases because for physically consistency the data must be a smooth function of period (e.g. Weidelt, 1972). Selecting at least two periods per decade is our recommendation for this stripe pattern. Test runs with additional periods should be made (if feasible, given available computer resources) to verify that the data space has been adequately resolved.

This pattern offers a significant computation advantage: calculations for the sensitivity matrix need only be done for a reduced set of periods. For the 2-D problem, these sensitivities can be calculated by direct factorization of the coefficient matrix (see forward modeling section). With the p*th*-stripe pattern this factorization need only be done for a small subset of periods. This idea is clearly directly transferable to the 3-D case.

In the second example every p*th* period and every s*th* station are selected in a staggered pattern to build the basis ("p*th*:s*th*-checker" pattern). This pattern is probably reasonable only in the case where the site spacing is small (e.g., continuous EM profiling). As noted above, because of the way we compute sensitivities, it is generally most efficient to use as many sites as possible (i.e. small s). However, this pattern can be used to reduce storage requirements for very large data sets.

Comparison of Computational Resource Requirements

In this section we summarize the computer resources required by each method (OCCAM, DASOCC and REBOCC). In all three methods, most of the major cost lies in calculating the sensitivity matrices, computing the cross-product matrices, and, then (for each λ) solving equations (8), (11) or (20) respectively, followed by forward modeling. Similarly, the memory required by each method is dominated by storage of two matrices: the sensitivity matrices and the cross-product matrices. For REBOCC, extra memory is required to store the $N \times L$ interpolation matrix $\mathbf{B}$, and an additional calculation is required to factor $\mathbf{B}$ (but this only be done once at the start of the inversion), and to compute the inner products of (18) (for every iteration).

Table 1 summarizes the number of operations (multiplication and additions) needed, along with the minimum storage requirements for each method. As L approaches N, the extra calculations required with the REBOCC method become significant. However, our experience shows that L need only be 10 - 30 % of N to ensure convergence of REBOCC. Thus REBOCC can reduce memory requirements by at least 60 % and cpu time by more than 80 % compared to DASOCC.

REBOCC : ALGORITHM DETAILS

The overall stability and efficiency of the REBOCC scheme depends on many details including forward modeling, the model covariance, the 1-D line search for the Lagrange multiplier λ, the interpolation matrix, and static shift corrections. In this section we describe our implementations of these parts of the inversion algorithms.

Forward Modeling

Forward modeling is the heart of the inversion and thus must be reliable, fast and accurate. It is used in two parts of the inversion: to compute the sensitivity matrix, and to compute responses for calculating the misfit.

Assuming a time dependence of $exp(-i\omega t)$, the second order Maxwell's equations to be solved are

$$\nabla \times \nabla \times \mathbf{E} = i\omega\mu\sigma\mathbf{E} \tag{21}$$

for the transverse electric (TE) mode in which the electric currents flow parallel to the strike of the structure, and

$$\nabla \times \rho\nabla \times \mathbf{H} = i\omega\mu\mathbf{H} \tag{22}$$

for the transverse magnetic (TM) mode in which the electric currents flow perpendicular to the strike. Here $\mathbf{E}$ and $\mathbf{H}$ are the electric and magnetic fields, σ is the conductivity (inverse of the resistivity ρ), μ is magnetic permeability for free space and ω is the angular frequency.

We use a finite difference (FD) method with the TE and TM mode differential equations discretized as in Smith and Booker (1991). As usual, the discrete form of the differential equations can be expressed as $\mathbf{Ax} = \mathbf{b}$, where $\mathbf{b}$ contains the terms associated with the known boundary values and the source fields, and $\mathbf{x}$ represents the unknown fields ($\mathbf{E}$ for TE and $\mathbf{H}$ for TM). The matrix $\mathbf{A}$ is sparse (5-nonzero diagonals) and symmetric, but not Hermitian since the diagonal elements are not real. Boundary conditions for the top, right and left sides of the model domain are as in Smith and Booker (1991). At the bottom of the model domain, we extend the last layer at least one skin depth and set the field at the bottom of this extra layer to zero. The accuracy of the solution is controlled by the quality of the mesh. The reader is referred to Rodi (1976), deGroot-Hedlin and Constable (1990) and Smith and Booker (1991) for discussion of these issues.

Sensitivities for MT data (e.g., the apparent resistivity or phase) can be readily calculated in terms of sensitivities of the electric and magnetic field components at the surface (Rodi, 1976; Mackie and Madden, 1993). The surface electric and magnetic field components in turn can always be expressed in the general form $\mathbf{a}^T\mathbf{x}$ (Rodi, 1976) where $\mathbf{a}^T$ may depend upon $\mathbf{m}$, and $\mathbf{x}$ is the discrete electric (TE) or magnetic (TM) field solution. These sensitivities may be calculated from

$$\frac{\partial(\mathbf{a}^T\mathbf{x})}{\partial m_j} = \frac{\partial \mathbf{a}^T}{\partial m_j}\mathbf{x} - \mathbf{a}^T\mathbf{A}^{-1}[\frac{\partial \mathbf{A}}{\partial m_j}\mathbf{x}] \qquad j = 1, \ldots, M. \tag{23}$$

The second term of the right hand side in (23) can be computed by solving the same system of equations required for the forward problem, but with a different right hand side, $(\partial\mathbf{A}/\partial m_j)\mathbf{x}$.

Rodi (1976) shows that there are two ways to calculate the field component sensitivities of (23): one requires solving the forward problem M times per period, once for each m_j. The other uses the reciprocity property of the forward problem and requires N_s (number of stations) forward solutions per period. Good reviews of the sensitivity calculations can be found in Rodi (1976), Rodi and Mackie (1999) and Mackie and Madden (1993). Since $N_s << M$ the second approach is generally much more efficient. We use this approach for REBOCC.

The sparse system of linear equations $\mathbf{Ax} = \mathbf{b}$ can be solved in two general ways: by a direct method using the LU decomposition or with an iterative method (Press et al., 1992), such as preconditioned conjugate gradients (PCG). Both approaches have advantages. With a direct method, after $\mathbf{A}$ is decomposed into lower ($\mathbf{L}$) and upper ($\mathbf{U}$) triangular matrices, solution (by forward and back substitution) is extremely fast. The direct approach is thus best for constructing the sensitivity matrix where the same system of equations must be solved for multiple right hand sides. To solve for a single right hand side (e.g., when calculating the misfit during search for λ) an iterative method which takes advantage of sparseness is more efficient. We use an iterative approach in these circumstances.

PCG is an extremely effective solution method for positive definite symmetric systems (Barrett et al., 1994). PCG requires less storage and fewer operations per iteration than other iterative methods such as the Biconjugate gradient method. In our case, the coefficient matrix $\mathbf{A}$ is symmetric but the diagonal is not real. However, since the imaginary parts of the diagonal terms are considerably smaller than the real parts, it is still possible that this algorithm will work reasonably well (Mackie and Madden, 1993). To use the classical PCG on the complex symmetric system (non-Hermitian), the conjugate transpose is not applied when computing the the inner product, i.e. we use $<\mathbf{x},\mathbf{y}> = \mathbf{x}^T\mathbf{y}$ (Barrett et. al., 1994). To speed up convergence, we use the incomplete LU decomposition level 3, ILU(3), (Kershaw D.S., 1978; Smith and Booker, 1991) as a preconditioner.

Model Covariance $\mathbf{C_m}$

In the penalty functional (4), the inverse of the model covariance is required to evaluate the norm $\mathbf{m}^T\mathbf{C_m}^{-1}\mathbf{m}$. Inversion of a non-diagonal $M \times M$ model covariance would not be computationally practical. Thus, for model space inversion approaches it is conventional to formulate the model norm in terms of a "roughness penalty", instead of the inverse of a covariance matrix (for example, deGroot-Hedlin and Constable (1990) used the first derivative roughness penalty, in equation (1)). The relationship of some common roughness penalties to a generalized sort of model covariance is discussed by Wahba (1990).

With a data space approach, inversion of $\mathbf{C_m}$ is not required (see (9)). This gives us a great deal of freedom to define a model covariance which can include prior information of various sorts. However, it is not practical to even form an arbitrary full (non-diagonal) $M \times M$ model covariance matrix for typical 2-D problems. We thus consider a fairly general class of model covariance functions which allow reasonable flexibility for including prior information, and at the same time allow for efficient computation of the matrix product $\mathbf{C_m}\mathbf{J}^T$ needed in (9) and (12) without actually computing $\mathbf{C_m}$ explicitly.

Initially we consider covariances of the general form

$$\mathcal{C}_{\mathcal{M}}(\mathbf{r},\mathbf{r}') = \eta^2(\mathbf{r})\varrho(\mathbf{r}-\mathbf{r}') \tag{24}$$

where $\eta^2(\mathbf{r})$ is a prior model variance at $\mathbf{r}$ in the model domain and $\varrho(\mathbf{r}-\mathbf{r}') = exp(-[(\mathbf{r}-\mathbf{r}')/r_e]^2)$ is the model correlation (with length scale r_e). For our initial discussion we assume $\eta^2(\mathbf{r})$ is a constant. $\mathbf{C_m}$ is the discrete representation of the covariance function $\mathcal{C}_{\mathcal{M}}(\mathbf{r},\mathbf{r}')$. Multiplication of a vector $\mathbf{u}$ in the discrete model parameter space (e.g. $\mathbf{u}$ could be a column of $\mathbf{J}^T$) by $\mathbf{C_m}$ is a discrete representation of the integral

$$\int \mathcal{C}_{\mathcal{M}}(\mathbf{r},\mathbf{r}')\mathcal{U}(\mathbf{r}')d\mathbf{r}' \tag{25}$$

in which $\mathcal{U}(\mathbf{r}')$ is smoothed by convolution with $\mathcal{C}_{\mathcal{M}}$, and $\mathbf{u}$ is discrete representation of $\mathcal{U}$.

Egbert et al. (1994) show that this integral can be computed (up to a scalar factor) by introducing a "pseudo-time" t and solving the diffusion equation

$$\frac{\partial \mathcal{U}}{\partial t} = \gamma \; \nabla^2 \mathcal{U} \tag{26}$$

with initial condition $\mathbf{u}(\mathbf{r}')$. Here ∇^2 is the 2-D Laplacian operator and γ is a diffusion parameter. If the pseudo diffusion time τ is chosen so that $r_e = \sqrt{4\gamma\tau}$ the integral (25) is given by $\mathbf{u}(\mathbf{r}, t = \tau)$. The matrix product $\mathbf{C_m u}$ can thus be computed by explicit time stepping of (26) on the model grid from $t = 0$ to τ. In general, allowing for a spatially varying variance $\eta^2(\mathbf{r})$ the scheme can be expressed as

$$\mathbf{C_m} = \mathbf{\Sigma D_0^{-\frac{1}{2}} D^{\tau} D_0^{-\frac{1}{2}} \Sigma}, \tag{27}$$

where $\mathbf{\Sigma}$ is the discrete model space variance matrix, $\mathbf{D}$ is the sparse diffusion operator matrix and $\mathbf{D_0}$ is a diagonal normalization matrix. The normalization factor of each block is calculated from solving (26), by replacing $\mathcal{U}$ with the delta function as an initial condition. Therefore, this normalization matrix requires solving the diffusion equation M times, but only once at the beginning of the process. Note that each step (i.e., multiplication by $\mathbf{D}$) represents a local smoothing of the field which requires minimal storage or computation time.

Egbert et al (1994) applied this approach to an oceanographic inverse problem. Here, we consider a slightly different approach which is more efficient and better suited to the MT inverse problem. Stability considerations limit the size of γ (see Press et al., 1992), and force τ to be a very large number if r_e is large. This makes a direct application of (27) impractical.

For REBOCC we have developed a more efficient approach based on a simple application of operator splitting methods (Press et al., 1992). With this approach the 2-D diffusion equation is replaced by a series of 1-D problems alternating between vertical and horizontal directions.

$$\mathbf{C_m} = \mathbf{\Sigma D_0^{-\frac{1}{2}}} [\mathbf{D_H^{\frac{1}{2}} D_V D_H^{\frac{1}{2}}}]^{\tau} \mathbf{D_0^{-\frac{1}{2}} \Sigma}, \tag{28}$$

Solutions of each 1-D diffusion equation (i.e. multiplication by $\mathbf{D_H}$ and $\mathbf{D_V}$) is rapid with a fully implicit method which is trivially implemented for a 1-D problem. This scheme is stable for arbitrarily long "time steps". Since we do not need an accurate solution of the diffusion equation to define a reasonable model correlation function, we use only a small number of pseudo-time steps, τ.

With this approach, it is not necessary to use a constant correlation length scale (or variances) throughout the model. Different length scales can be used in vertical and horizontal directions. The model covariance approach is thus very flexible, and allows rather general statistical specification of prior information. For example, known faults can be allowed for by letting the smoothing length scale go to zero at the known fault location, or part of the resistivity structure can be frozen by letting variances go to zero. However, the choice of proper correlation length scale is important, but difficult to justify rigorously. Making r_e too large can result in difficulties in finding any models that fit the data. Choosing r_e too small can also result in problems (e.g., confusion of static shifts with deep lateral structure). Inevitably some experimentation with correlation length scales will be required.

As a default strategy we make the vertical correlation length scale of each layer proportional to the depth of that layer, with the horizontal correlation length scale set to the maximum of the depth and the gap between stations. On the edges of the model domain, the length scale is set equal to the distance from sides of the model to the edge stations. This choice of correlation length scales coincides with the loss of the resolving power of the data at depth, and near the boundaries of the model.

The model covariance of (28) allows only for perturbations around the assumed background model, which may in fact be grossly wrong. We can make some allowance for this in the model covariance by adding a constant matrix $\mathbf{K}$ (all elements are κ) so that the covariance matrix $\mathbf{C_m}$ becomes

$$\mathbf{C_m} = \mathbf{\Sigma}(\mathbf{K} + \mathbf{D_0^{-\frac{1}{2}}} [\mathbf{D_H^{\frac{1}{2}} D_V D_H^{\frac{1}{2}}}]^{\tau} \mathbf{D_0^{-\frac{1}{2}}}) \mathbf{\Sigma}.$$

The addition of $\mathbf{K}$ to the model covariance corresponds to allowing for uncertainties in the level of a constant background resistivity. In our experience $\kappa = 1$ usually works well (with $\eta^2(\mathbf{r}) = 1$), however, the limits of this choice need to be better understood.

Finally, note that the data subspace cross-product matrix can be written

$$\begin{aligned} \mathbf{\Gamma^l} &= \mathbf{J C_m J}^T \\ &= \mathbf{J \Sigma K \Sigma J}^T + ([\mathbf{D_H^{\frac{1}{2}} D_V D_H^{\frac{1}{2}}}]^{\frac{\tau}{2}} \mathbf{D_0^{-\frac{1}{2}} \Sigma J})^T ([\mathbf{D_H^{\frac{1}{2}} D_V D_H^{\frac{1}{2}}}]^{\frac{\tau}{2}} \mathbf{D_0^{-\frac{1}{2}} \Sigma J}). \end{aligned} \tag{29}$$

Computation of the cross-product matrix is thus accomplished most efficiently by doing half the smoothing of the sensitivities to compute $([\mathbf{D_H^{\frac{1}{2}} D_V D_H^{\frac{1}{2}}}]^{\frac{\tau}{2}} \mathbf{D_0^{-\frac{1}{2}} \Sigma J})$ followed, by computation of the cross products.

Line Search for λ

At every iteration, we search for the λ that gives the model (defined by (12) and (20)) with the minimum misfit (Phase I), or at the desired misfit (Phase II). For each trial value of λ, the system of equation (20) must be solved, the model updated, and the misfit computed by solving the forward problem. Therefore, minimizing the number of λs tried can help us to further reduce computational costs.

We use a relatively simple search method which takes advantage of several facts: (a) the misfit is a smooth function of λ, (b) the range of log10 λ is generally within the interval [0,6] (for our default model variance $\eta^2(\mathbf{r}) = 1$), and (c) the optimal choice of λ changes little between iterations.

For the first iteration, we start with three different equally spaced λs covering one decade of λ. Misfits computed for these initial λs will tell us whether we should go left or right, or stop if a minimum has already been bracketed. Once a minimum is bracketed, we use a parabolic interpolation (Press et al., 1992) on these three points to estimate the minimum of the misfit. This estimated misfit might be significantly lower than the misfits from the three λs. If not, we just keep the best λ so far and move on to the next iteration. Usually, this scheme requires about 3 to 8 trials values of λ, about half of the 8 to 12 values of λ per iteration reported for the 2-D Occam's inversion by deGroot-Hedlin and Constable, (1990).

With the prior knowledge of the previous iteration, the previous three bracketing points can be used as a starting point for the next iteration, and the process is repeated. Generally, the optimal λ does not change much between iterations, so only a small number of trial values of λ (but at least 3) are required.

Once the desired misfit is achieved within the range of trial λs bracketing the minimum, parabolic interpolation is used to locate λ providing the desired level of misfit. If two or more values of λ bracketing the minimum have the same (desired) misfit, we choose the larger λ, which usually corresponds to a smaller model norm.

In general, the desired misfit may never be reached, so that the smoothing process is not performed. The model with the minimum misfit (higher than the desired misfit) might then contain some unnecessary features inserted by the inversion to make the fit better. Therefore, we would recommend an additional run with a higher target misfit to find the minimum norm model corresponding to a larger (and this time achievable) misfit.

Interpolation Matrix

As described above, we require an approximation to the full sensitivity matrix $\mathbf{J}_k$ to use all of the data in the approximate penalty functional (19). This approximation can be made via the subset sensitivity matrix $\mathbf{G}_k$ and an $N \times L$ interpolation matrix $\mathbf{B}$.

Here, we describe our interpolation scheme which is fairly simple, while still allowing for a general pattern of data subsets. Each type of data (e.g. TM phase) is tabulated into a 2-dimensional array ordered by period and station location (see Figure 3). To interpolate the sensitivity to a site/period for which the sensitivity is not computed (j_s, j_p) (filled circle) we first divide the area surrounding it into eight parts, $A_1, \cdots, A_8$, as indicated in Figure 3.

In each area, we search for the nearest calculated sensitivity, indexed by (i_s, i_p). The interpolated sensitivity is then a weighted average of the nearest calculated sensitivities. The weighting coefficients for the eight nearest elements are given as

$$\Pi(i_s, j_s; i_p, j_p) = \frac{\Pi(i_s, j_s)\Pi(i_p, j_p)}{\sum \Pi(i_s, j_s)\Pi(i_p, j_p)}$$

where $\Pi(i_s, j_s) = exp[-\frac{\Delta^s_{ij}}{\sum \Delta^s_{ij}}]$, $\Pi(i_p, j_p) = exp[-\frac{\Delta^p_{ij}}{\sum \Delta^p_{ij}}]$, and Δ^s_{ij} and Δ^p_{ij} are the distance between stations and log10 period difference respectively, and the summation is over the eight subareas of figure 3.

This simple interpolation scheme can fail when the station spacing is large. However, in such a case, it would probably be best to avoid interpolation between stations, and use a pth-stripe pattern.

Static Shift Correction

Shallow local inhomogeneities can distort the regional electric field, and cause a frequency-independent shift in the log apparent resistivity while leaving the phase unaffected. For REBOCC, static shifts are incorporated as additional model parameters, which are automatically estimated by the inversion (if requested by the user). For each iteration static shifts for each site are estimated using the median residual (observed minus (undistorted) calculated) log10 apparent resistivity. Then, the misfits are recalculated using the (distorted) calculated responses. More sophisticated ways can be used to obtain the static shift factor (e.g., Wu et. al., 1993; deGroot-Hedlin and Constable, 1993; Ogawa and Uchida, 1996), but tests with synthetic distorted data indicate that this simple scheme is effective.

NUMERICAL EXPERIMENTS

Synthetic Data

To test REBOCC, we generated synthetic data from a 2-D model adapted from the COPROD2 inversion results of Wu et. al. (1993). The model (Figure 4) consists of four layers, with three 1 $\Omega-m$ rectangular conductors embedded (see Figure 4 for details). The grid used to form the synthetic data was well discretized at 170×183 (with an additional 10 air layers for the TE mode) to ensure accuracy of the solution. Model responses, including apparent resistivity and phases for transverse magnetic (TM) and transverse electric (TE) modes, and also vertical magnetic field transfer functions (which we will refer to as tipper, or TP), were generated using the finite element forward modeling program of Wannamaker et al. (1986). Data for 36 stations spaced at 3 km intervals, and 31 periods increasing logarithmically from 1 s to 1000 s (about 10 periods per decade) were used for the inversion test. Two percent Gaussian noise was added to the data prior to the inversion. Note that the three conductors are not clearly evident in either TE or TM model responses (Figure 4).

Results of REBOCC Inversion

Our first experiment is to invert the synthetic data using different subsets of calculated representers for TM and combined TM+TE mode data sets. We consider one checker pattern and three stripe patterns in this comparison. All inversions are run on a Sun UltraSparc I workstation. The desired RMS is set to 1 (i.e., 2 % misfit).

Due to limitations of computer resources, the model grid used for inverting the data is necessarily coarser. Different forward modeling programs and different grids will help to reveal any systematic errors occurring in our forward program. The model grid used for the inversion was thus discretized at 100×31, plus 10 air layers for the TE mode. A 100 $\Omega-m$ half space was used as a starting model for all inversions.

The correlation length scale r_e for all runs was set as described in the model covariance section. With a station spacing of 3 km this leads to larger horizontal length scales from the surface to 3 km depth, and a 1:1 ratio below.

Memory requirements for storing the cross-product and sensitivity matrices for all three methods (OCCAM, DASOCC and REBOCC) and the interpolation matrix (for REBOCC only) are calculated directly from Table 1, and are shown in Figure 5.

For TM+TE inversion memory requirements of the data space (DASOCC) approach exceed those of the model space (OCCAM) approach, because N is actually larger than M for this joint inversion case. However, using the reduced basis method, memory requirements can be significantly reduced (Figure 5). For example, only 12 MBytes of memory are needed for a TM mode inversion using the $6th$:$2nd$-checker pattern of calculated representers. These computations would be practical on even a small PC.

Although memory requirements for OCCAM are comparable to those of DASOCC, joint TM and TE inversion with OCCAM is not practical. Cpu time on a Sun UltraSparc I workstation for the TM only inversion ($N=2232$ and $M=3100$) required about 22 hours per iteration, significantly longer than required by data space methods (DASOCC and REBOCC). The long run times result from larger matrices in the model space, and from direct calculation of the sensitivity matrix elements without the reciprocity theorem. Because of the long run times required, we do not consider OCCAM further in our comparison.

The convergence rates of DASOCC and REBOCC are plotted in Figure 6. For both the TM and TM+TE inversions, the full basis inversion (DASOCC) requires much longer per iteration than any of the REBOCC inversions (by a factor of 6 - 35). Both the $9th$-stripe and the $6th$:$2nd$-checker data subset patterns have comparable numbers of representers (L), and require about the same amount of cpu time per iteration. However, the 6th:2nd-checker converges while the $9th$-stripe does not. Clearly representers for more than one period per decade is required. For the TM mode case, all of the inversions converge (except the $9th$-stripe) to the desired misfit (5 iterations for the $6th$:$2nd$-checker subset; 3 iterations for the rest). Models resulting from the different inversions are indistinguishable from one another. Only the models from the $6th$-stripe subset are shown in figure 7. For TM+TE inversion, none of the inversions are able to find a model with an RMS of 1. Except for the $9th$-stripe, all converge to some minimum level, which tends to get larger as L gets smaller. The minimum misfit is around 1.15 RMS for the full basis, and 2.5 RMS for the $6th$:$2nd$-checker subset.

We used a $6th$-stripe representer subset for additional single and joint mode inversions, using the same synthetic data with the same starting model. Figure 7 and 8 show the results of these runs. The TM mode inversion reveals the layered host resistivity structure with little lateral variation. In contrast, the TE inversion reveals two conductors beneath the resistive layer. The second conductor on the east side of the model from $y = 0$ to 40 km corresponds to a combination of conductors B and C in the synthetic model. The conductive features extend continuously to the deeper part of the model, and the layered host resistivity structure is not well-resolved. Interestingly, the tipper inversion displays the boundaries of the conductors more accurately than either single mode MT inversion. Even the small third conductor (C) can be distinguished. However, the layered host resistivity structure is very poorly resolved with the tipper inversion.

The joint inversion of TM and TE modes shows that the MT data is fit adequately with only two conductors inside the resistive layer (although the desired level of misfit is not quite achieved). Using tipper and the TE mode, the inversion finds all three conductors (although the last one is not well-separated from the middle one). However, the upper resistive layer is not well-defined. Similarly, with tipper and the TM mode, all three conductors inside the resistive layer are recovered. All of these joint inversions show that the tipper is needed for the inversions to resolve the smallest conductor. The model from tipper and TM mode inversion seems to be closest to the true model of Figure 4.

Finally, we inverted all data, TM, TE and tipper using the REBOCC inversion with a $6th$-stripe pattern ($N = 6696$, $M = 3100$, and $L = 1296$). The inversion requires about 108 MBytes of memory and approximately 11 hours of run time (on a Sun UltraSparc 1) to obtain a model with a RMS of 1.0 at the 18th iteration. One can stop the inversion after the 8th iteration (about 5 hrs) to obtain a model with an RMS of about 2 (or 4 % error), which is in fact indistinguishable from the fully converged run. The model and the fit are shown in Figure 8.

Comparison with Other Inversions

Using the same data set, we also compared REBOCC with several other inversion programs: NLCG of Rodi and Mackie (1999) and RRI of Smith and Booker (1991). The same starting model (100 $\Omega - m$ half space) and the same model mesh ($M = 100 \times 31 = 3100$) were used for all inversions. For timing comparisons, we run the inversions on the same machine, a Sun UltraSparc I. We use the $6th$-stripe subset for the REBOCC inversion. For other inversion programs, we generally used the default parameter values provided with the programs.

Figure 9 shows the models produced by NLCG and RRI for TM, TE and TM+TE data sets (Note that NLCG and RRI do not presently allow for inversion of tippers). These plots should be compared with the REBOCC results of Figure 7 (a), (b) and (d). In Figure 10 we plot RMS misfit against cpu time (and iteration number) of RRI, REBOCC ($6th$-stripe) and NLCG algorithms for TM inversion and joint inversion of TM and TE modes. The plot of RMS misfit and cpu time for the TE inversion is very similar to that of the TM case, but cpu times are greater.

RRI which is incredibly fast per iteration requires 12 iterations to converge to the solution at the desired level of misfit for TM mode (27 iterations for TE mode). However, for the joint inversion of TM and TE, RRI fails to converge to a reasonable model. The minimum misfit achieved is quite large (4.53), and the final model is very rough. The small scale features (D+ like) between $y = -15$ km and 0 km (between conductor A and B in the synthetic model; Figure 4) are inserted in the early iterations and, once there, are very difficult to get rid of in the subsequent iterations (Smith and Booker, 1991). This confusion of the inversion is perhaps the result of the 1-D sensitivity approximation used by RRI. Beneath the sites from $y = -15$ to 0 km, the TM data (which is not sensitive to the isolated conductors) suggests a resistive structure. On the other hand, the TE data which is highly sensitive to the nearby conductors, tries to place conductive features beneath these sites.

NLCG converges to the desired misfit in 31 iterations (about 4000 seconds) for the TM mode, and about 27500 seconds (97 iterations to reach an RMS of 1.1) for the TE mode. The same misfits of both single mode inversions can be accomplished much more quickly with REBOCC about 600 seconds (3 iterations) for the TM mode (Figure 10) and about 1600 seconds (5 iterations) for the TE mode.

For joint inversion, NLCG requires about 51000 seconds (120 iterations) to reach an RMS of 1.4 while REBOCC only need 9500 seconds (in 11 iterations) to reach a similar level of misfit. However, in this example NLCG can reduce the RMS to a lower minimum (1.2 RMS after 165 iterations) than the the $6th$-stripe REBOCC (1.37 RMS after 11 iterations). To reduce the RMS further, more basis function may be required for REBOCC.

Note that the convergence times plotted in Figure 10 for REBOCC reflect only Phase I iterations where the goal is to bring the misfit down to the desired level (1.0 RMS). Since for the joint (TM+TE) inversion the desired level cannot be reached, one should restart the inversion so that Phase II can be completed with the desired misfit level set to a higher value (e.g., 1.5 RMS starting with the model from the 9th iteration of Phase I). Phase II is necessary in order to obtain the minimum structure model (Parker, 1994). At the same time, for NLCG, to obtain a model with minimum structure, several trial values of λ would be required.

It is difficult to compare the models produced by NLCG, RRI and REBOCC (Figure 9 and Figure 7) because of the difference of the smoothing parameters (model covariance). However, all of these inversions (except the joint mode inversion for RRI) reveal the main features and produce responses that fit the data adequately.

Example with Field Data

Finally we briefly consider application of REBOCC to a real data set, a high density MT profile across the San Andreas fault (SAF) near Parkfield, CA (Unsworth et al., 1997). This is a large data set with 55 TM mode responses, 37 TE mode responses and 15 tippers each at 41 periods. We discretized the model at 200×74 blocks (plus 10 air layers for the TE mode), and used a 10 $\Omega - m$ half space as a starting model. The static shift distortion parameters are set free so that the program can automatically adjust the values. An $8th$-stripe subset ($L = .15N$) is used for joint inversion of TM, TE and vertical magnetic transfer functions. The results of inversion are shown on Figure 11. After the 4th iteration (about 12 hrs of cpu time on Sun UltraSparc I), the inversion finds a model with a misfit of about 13 % (an RMS of about 2.7). A slightly better misfit can be obtained after a few more iterations. However, the norm of the model increases significantly when the data is fit better. Dimensionality analysis of the full impedance tensor suggests that the misfit of the model of Figure 11 is reasonable, given the degree of 3-D complications in this data set (Siripunvaraporn et al., 1998). Thus, we prefer the model with the misfit at the 4th iteration. These cpu times are large, but this is a very large data set ($N = 8774$) and big model ($M = 14800$).

DISCUSSIONS AND CONCLUSIONS

The REBOCC inversion has been shown to be effective in practice. By using a relatively small subset of the representers (i.e., smoothed sensitivities) computational requirements (both memory and cpu time) can be substantially reduced. The pth-stripe subset requires sensitivity calculations for only every pth frequency. Our results show that even with p up to about 8 the inversion usually converges to a reasonable model, which does not depend in any significant way upon p.

In our comparisons with other inversions, we found that REBOCC was significantly faster than both OCCAM and NLCG. None of these inversions (including REBOCC) are competitive with RRI in terms of speed. However, we could not get RRI to converge for joint mode (TM+TE) inversion of our test data set. Possibly, a more experienced user of this program might be able to vary or adjust some parameters to make the inversion work successfully. Similarly for NLCG, the convergence rate might vary with different values of λ. Perhaps a reasonable model could be obtained more rapidly with a different choice of λ. However, to find a true minimum structure solution (as REBOCC does) NLCG would have to be run with several trial values of λ (some of which might converge more slowly).

CG and NLCG are descent methods which make no use of the second derivative (Hessian) of the penalty functional. GN and OCCAM essentially calculate the full Hessian, while REBOCC make a very good (but not perfect) approximation. Our results indicate that this approximate calculation is worth the effort. This is particularly true for the 2-D case considered here, where direct LU factorization of the differential equation coefficient matrix is feasible, since in this case many sensitivities can be computed quickly once the factorization is complete.

The basic idea behind the REBOCC algorithm could also be applied to the 3-D inversion problem. We thus do not agree with the statement made by Rodi and Mackie (1999) that any sort of inversion based on a sensitivity calculation will not be practical for realistic 3-D EM problems, even allowing for improvement of the computer hardware. A straightforward extension of REBOCC to 3-D inversion is readily apparent, although for calculating the sensitivity matrix the LU decomposition would probably be impractical (except on a supercomputer) and have to be replaced by a relaxation method. The possibility of adapting the reduced basis method to other inversion approaches also deserves consideration.

For example, one could try to solve (20) using conjugate gradients. As in the model space conjugate gradient approach used by Mackie and Madden (1993) the matrix multiplications required can be reduced to solving the forward problem twice per inner loop iteration. Using the reduced basis idea, forward calculations could be made for only a subset of frequencies. Sensitivity and cross-product matrices would not have to be constructed or stored. However, a full conjugate gradient solution would be required for each λ, so a true OCCAM type approach would probably not be practical. A similar data space conjugate gradient scheme has been applied to oceanographic inversion problems by Egbert (1997) and Bennett et al. (1996).

We believe that REBOCC is a significant advance in practical methods for solution of 2-D MT inverse problems, and that the basic ideas of our implementation should be highly relevant to the 3-D problem in some form.

ACKNOWLEDGMENTS

This research has been partially supported by the Royal Thai Embassy via the Development and Promotion of the Science and Technology (DPST) scholarship to WS, and by grants from the US DOE (DE-FG03-96ER1495) and NSF (9614411-EAR).

REFERENCES

Barret, R., Berry, M., Chan, T. F., Demmel, J., Donato, J., Dongarra, J., Eijkhout, V., Pozo, R., Romine, C., and Van der Vorst, H., 1994, Templates for the Solution of Linear Systems: Building Blocks for Iterative Methods, Soc. Ind. Appl. Math.

Bennett, A. F., Chua, B. S., and Leslie, L. M., 1996, Generalized inversion of a global numerical weather prediction model, Meteorology and Atmospheric Physics, **60**, 165-178.

Chen, L., Booker, J. R., Jones, A. G., Wu, N., Unsworth, M., Wei, W., and Tan, H., 1996, Electrically conductive crust in southern Tibet from INDEPTH magnetotelluric surveying, Science, **274**, 1694-1696.

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

deGroot-Hedlin, C., and Constable, S., 1990, Occam's inversion to generate smooth, two-dimensional models from magnetotelluric data, Geophysics, **55**, 1613-1624.

deGroot-Hedlin, C., and Constable, S., 1993, Occam's Inversion and the North American Central Plains Electrical Anomaly, J. Geomag. Geoelectr., **45**, 985-999.

Egbert, G. D., Bennett, A. F., and Foreman, M. G., 1994, TOPEX/POSEIDON tides estimated using a global inverse model, J. Geophys. Res., **99**, 24821-24852.

Egbert, G. D, 1997, Tidal data inversion: interpolation and inference, Prog. Oceanog., **40**, 53-80.

Farquharson, C. G., and Oldenburg, D. W., 1996, Approximate sensitivities for the electromagnetic inverse problem, Geophys. J. Internat., **126**, 235-252.

Jones, A. G., 1992, Electrical conductivity of the continental lower crust, *in* Fountain, D. M., Arculus, R. J., and Kay, R. W., Eds., Continental Lower Crust: Elsevier Science Publ. Co., Inc., 81-143.

Jupp, D. L. B., and Vozoff, K., 1975, Stable iterative methods for the inversion of geophysical data, Geophys. J. Roy. Astr. Soc., **42**, 957-976.

Kershaw, D.S., 1978, The Incomplete Cholesky-Conjugate Gradient Method for the Iterative Solution of Systems of Linear Equations, J. Comput. Phys., **26**, 43-65.

Mackie, R. L., and Madden, T. R., 1993, Three-dimensional magnetotelluric inversion using conjugate gradients, Geophys. J. Internat., **115**, 215-229.

Marquardt, D. W., 1963, An Algorithm for least-squares estimation of nonlinear parameters, J. Soc. Indust. Appl. Math, **11**, 431-441.

Oldenburg, D. W., and Ellis, R. G., 1991, Inversion of geophysical data using an approximate inverse mapping, Geophys. J. Internat., **105**, 325-353.

Oldenburg, D. W., McGillivray, P. R., and Ellis, R. G., 1993, Generalized subspace methods for large scale inverse problems, Geophys. J. Internat., **114**, 12-20.

Oldenburg, D. W., and Ellis, R. G., 1993, Efficient inversion of magnetotelluric data in two dimensions, *Phys. Earth Planet. Internat.*, **81**, 177-200.

Ogawa, Y. and T. Uchida, 1996, A two-dimensional magnetotelluric inversion assuming Gaussian static shift, *Geophysics*, **126**, 69-76.

Orange, A. S., 1989, Magnetotelluric Exploration for Hydrocarbons, Proc. IEEE, **77**, 287-317.

Parker, R. L., 1980, The inverse problem of electromagnetic induction: existence and construction of solutions based upon incomplete data, J. Geophys. Res., **85**, 4421-4425.

Parker, R. L., and Shure, L., 1982, Efficient Modeling of the Earth's Magnetic Field with Harmonics Splines, Geophys. Res. Lett., **9**, 812-815.

Parker, R. L., 1994, Geophysical Inverse Theory: Princeton Univ. Press.

Press, W. H., Teukolsky, S. A., Vetterling, W. T., and Flannery, B. P., 1992, NUMERICAL RECIPES in FORTRAN: The Art of Scientific Computing - 2nd edition: Cambridge Univ. Press.

Rodi, W. L., 1976, A technique for improving the accuracy of finite element solutions for magnetotelluric data, Geophys. J. Roy. Astr. Soc., **44**, 483-506.

Rodi, W. L., and Mackie, R. L., 1999, Nonlinear Conjugate Gradients Algorithm for 2-D Magnetotelluric Inversion, Geophysics, submitted.

Siripunvaraporn, W., Egbert, G. D., Eisel, M., and Unsworth, M., 1998, A High Resolution EM Survey of the San Andreas Fault (SAF): Local Conductivity Structure in a Regional Context, Eos, Trans. Am. Geophys. Union, **79**, 45, Fall Meet. Suppl., F227.

Smith, J. T, and Booker, J. R., 1988, Magnetotelluric inversion for minimum structure, Geophysics, **53**, 1565-1576.

Smith, J. T, and Booker, J. R., 1991, Rapid Inversion of Two- and Three-Dimensional Magnetotelluric Data, J. Geophys. Res., **96**, 3905-3922.

Uchida, T., 1993, Smooth 2-D Inversion for magnetotelluric data based on statistical criterion ABIC, J. Geomag. Geoelectr., **45**, 841-858.

Unsworth, M. J., Bedrosian, P., Egbert, G. D., and Eisel, M., 1997, 3-D Electrical Resistivity Structure of the San Andreas Fault at Parkfield, California, Eos, Trans. Am. Geophys. Union, **78**, 46, Fall Meet. Suppl., F696.

Unsworth, M., Egbert, G., and Booker, J., 1998, High resolution electromagnetic imaging of the San Andreas fault in Central California, J. Geophys. Res., **104**, 1131-1150.

Wannamaker, P. E., Stodt, J. A., and Rijo, L., 1986, Two-dimensional topographic responses in magnetotelluric modeled using finite elements, Geophysics, **51**, 2131-2144.

Wannamaker, P. E., Booker, J. R., Filloux, J. H., Jones, A. G., Jiracek, G. R., Chave, A. D., Tarits, P., Waff, H. S., Egbert, G. D., Young, C. T., Stodt, J. A., Martinez, M. G., Lwaw, L. K., Yukutake, T., Segawa, J. S., White, A., and Green, A. W., Jr, 1994, Magnetotelluric Observations Across the Juan de Fuca Subduction System in the EMSLAB Project, J. Geophys. Res., **94**, 14111-14125.

Wahba, G., 1990, Spline Methods for Observational Data, Soc. Ind. Appl. Math.

Vozoff, K., 1972, The Magnetotelluric Method in the Exploration of Sedimentary Basins, Geophysics, **37**, 98-141.

Weidelt, P., 1972, The inverse problem of geomagnetic induction, Z. Geophys., **38**, 257-289.

Wu, N., Booker, J. R., and Smith, J. T., 1993, Rapid Two-Dimensional Inversion of COPROD2 Data, J. Geomag. Geoelectr., **45**, 1073-1087.

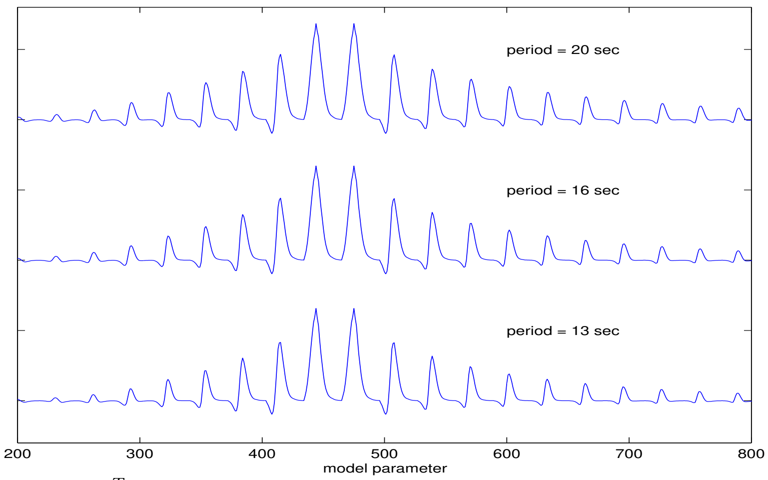

FIG. 1. Plots of representers $\mathbf{C_m J}^T$ for TM mode apparent resistivity at one station for nearby periods, for the synthetic data discussed later. The representers change very slowly as a function of period.

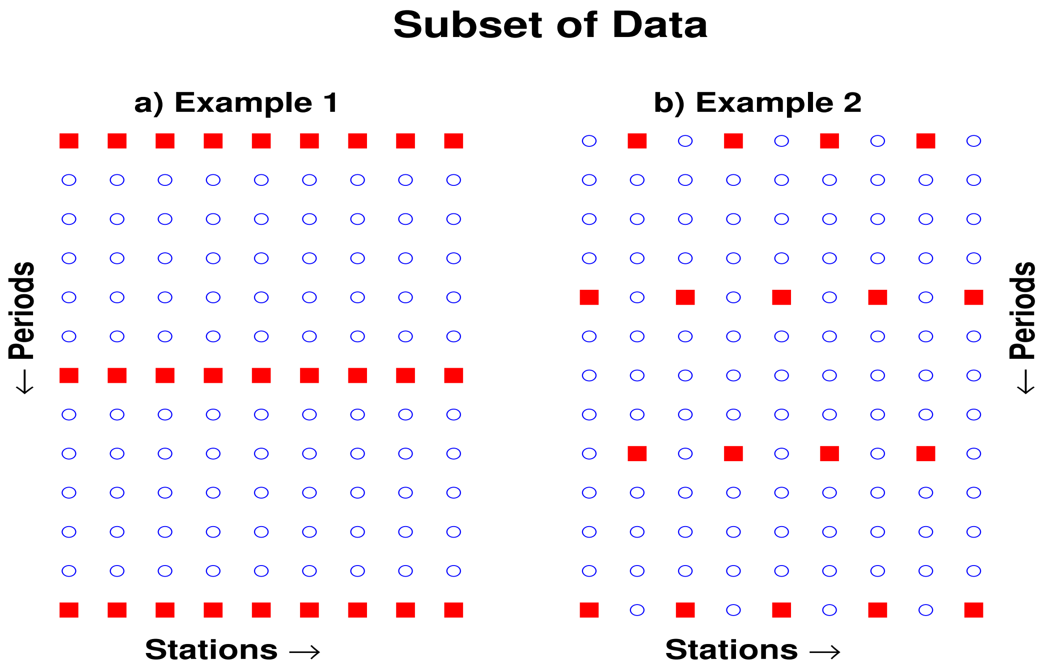

FIG. 2. a) First example: "pth-stripe" pattern where every pth period is selected for all sites (here p = 6). Representers are calculated for data corresponding to the filled squares; the open circles represent the remaining data. b) Second example: "pth:sth-checker" (with p = 4 and s = 2) pattern where the selected data form a checker pattern.

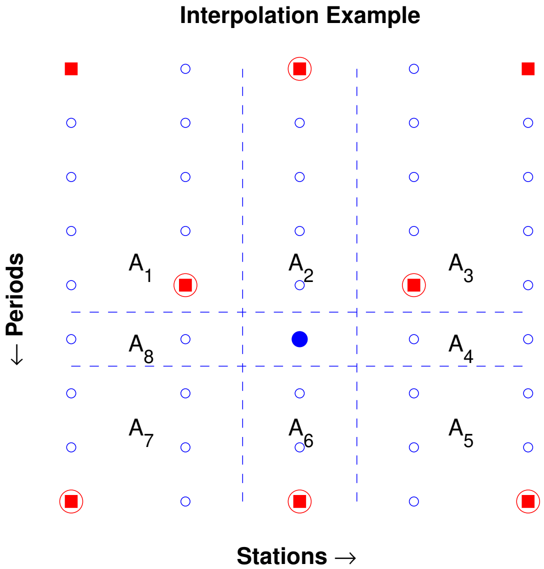

FIG. 3. Schematic interpolation example. Filled squares correspond to calculated sensitivities, and open circles the remaining data. To interpolate the calculated sensitivities to the filled circle, the data distribution is divided into eight areas, $A_1, \cdots, A_8$, as indicated. The nearest calculated sensitivities (filled squares inside the circle) are then chosen from each area to form the interpolated sensitivity at the target point. Sub-areas where there are no calculated sensitivities, e.g. A_4 and A_8, are omitted from the weighted average.

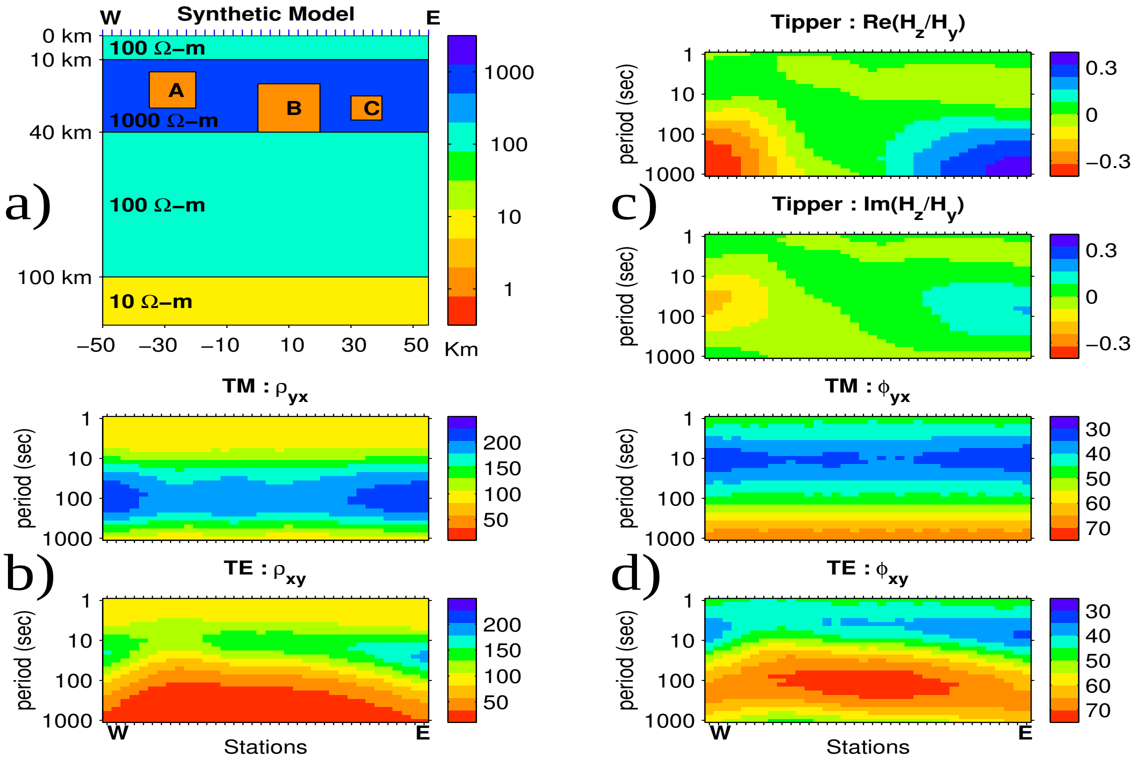

FIG. 4. (a) model used to generate synthetic data responses. Inside the 1000 $\Omega - m$ resistive layer, there are three 1 $\Omega - m$ conductors, A, B and C. Conductor A is 15 km × 15 km, buried at 15 km depth. Conductor B is 20 km × 20 km, buried at 20 km depth. Conductor C is 10 km × 10 km, buried at 25 km depth. A and B are separated by 20 km, while B and C are separated by only 10 km. Apparent resistivities and phases from TM and TE mode and Tippers generated from the model are shown in (b) - (d).

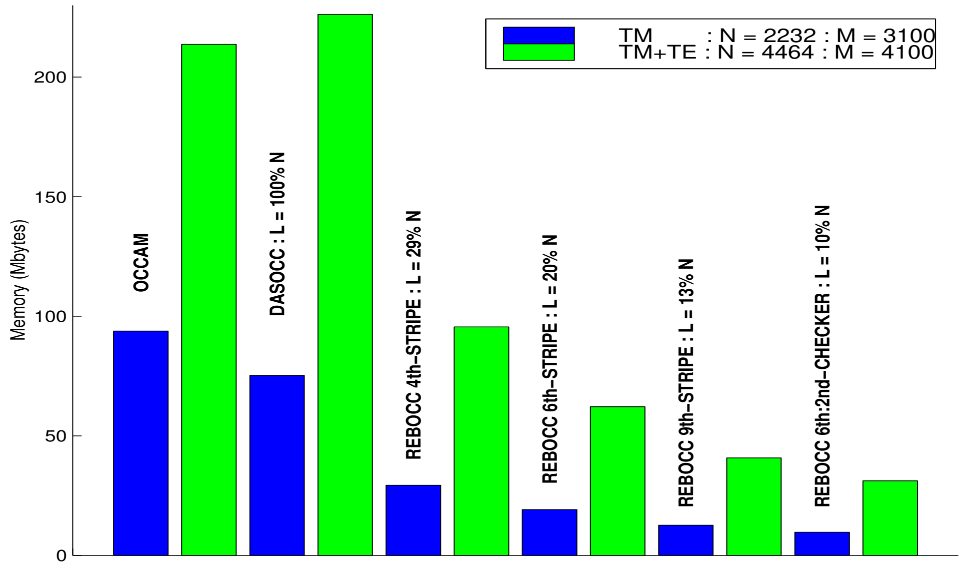

FIG. 5. Memory requirements for OCCAM, DASOCC and REBOCC for different representer subsets calculated from Table 1. for TM and TM+TE modes. Note that double precision is used.

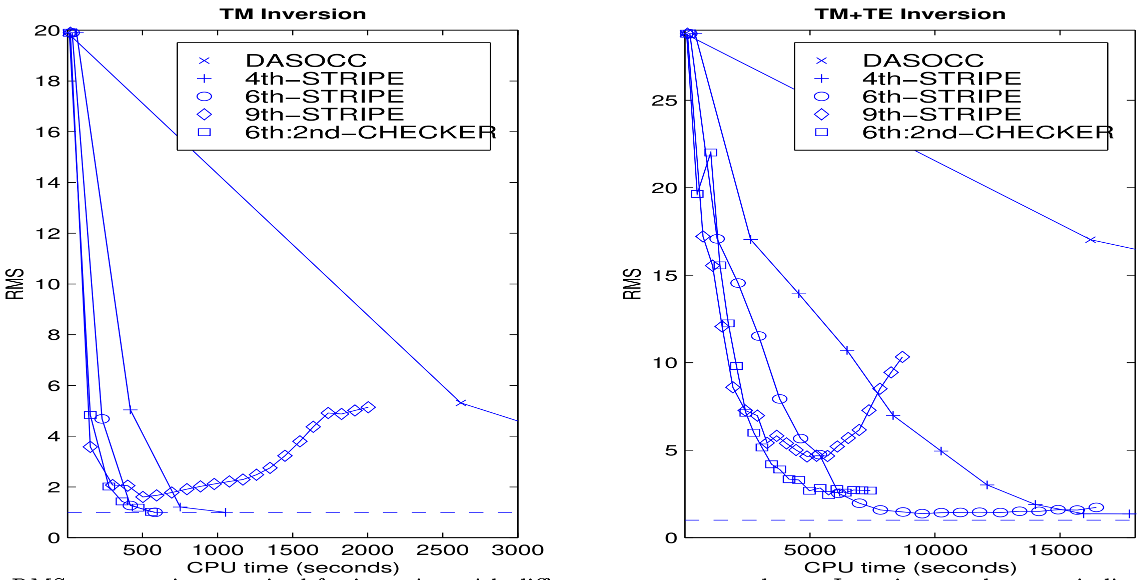

FIG. 6. RMS vs. cpu time required for inversion with different representer subsets. Iteration numbers are indicated by the plotted symbols.