Three-dimensional magnetotelluric inversion using conjugate gradients

Randall L. Mackie and Theodore R. Madden

Massachusetts Institute of Technology, Department of Earth, Atmospheric and Planetary Sciences, Cambridge, MA 02139-4307, *USA*

Accepted 1993 March 8. Received 1993 February 25; in original form 1992 February 26

SUMMARY

We have developed an inversion procedure that uses conjugate gradient relaxation methods. Although one can generalize the method to all inverse problems, we demonstrate its use to invert magnetotelluric data for 3-D earth models. This procedure allows us to bypass the actual computation of the sensitivity matrix **A** or the inversion of the $\mathbf{A}^T\mathbf{A}$ term. In fact, with the relaxation approach, one only needs to compute the effect of the sensitivity matrix or its transpose multiplying an arbitrary vector. We show that each of these requires one forward problem with a distributed set of sources either in the volume (for **A** multiplying a vector) or on the surface (for $\mathbf{A}^T$ multiplying a vector). This significantly reduces the computational requirements needed to do a 3-D inversion. For this paper, we have simplified the boundary conditions by assuming the model is repeated in the horizontal directions, but this is not a necessary constraint of the method. The algorithm reduces data errors to the 2 per cent level for noise-free synthetic 3-D magnetotelluric data.

Key words: magnetotellurics, 3-D inversion, conjugate gradient relaxation

INTRODUCTION

Inversion is the procedure of estimating physical earth parameters from a set of observed data, perhaps subject to certain constraints. Linearized inversion schemes provide one method for dealing with non-linear inversion problems (those problems where the data are non-linearly related to the model parameters). They involve expanding the model response in a Taylor series around some point in the model space, and then solving for the model changes that minimize the error between the model response and the observed data. Many such schemes exist, such as the non-linear least-squares method, but we prefer to use the maximum likelihood inversion procedure (Mackie, Bennett & Madden 1988). The maximum likelihood solution is the solution that maximizes the joint probability of fitting the observed data (subject to the data covariance) and adhering to an *a priori* model (subject to the model covariance). One obtains this solution with the help of the sensitivity matrix, which is the matrix that relates small changes in the model parameters to changes in the observed data.

For the 3-D magnetotelluric inversion problem, we apply conjugate gradient relaxation methods (Hestenes & Stiefel 1952) to solve the maximum likelihood system of equations instead of solving the system directly using matrix inversion routines. In doing so, we avoid having to explicitly construct the sensitivity matrix; indeed, we only need to know the effect of the sensitivity matrix, or its transpose, multiplying an arbitrary vector. We will show that each of these operations requires one forward problem, with sources distributed either in the volume, for the sensitivity matrix multiplying a vector, or on the surface, for its transpose multiplying a vector. This implementation greatly reduces the computational enormity of 3-D inversion, making practical 3-D inversions a possibility rather than an impossibility. One can generalize this method to any other inverse problem, but here we demonstrate the procedure for the magnetotelluric problem.

MAXIMUM LIKELIHOOD INVERSE

The maximum likelihood inverse is one method to obtain a solution to a non-unique inverse problem. There are many procedures for obtaining solutions, and each one biased, but the maximum likelihood inverse clearly exposes its biases. The maximum likelihood inverse is an example of a linearized inversion scheme. One can find the derivation of the maximum likelihood inverse in Mackie *et al.* (1988) and Madden (1990), and it closely follows the work of Tarantola & Valette (1982) and Tarantola (1987). Our maximum likelihood inverse, however, is somewhat different from that of Aki & Richards (1980). Their maximum likelihood inverse first minimizes the weighted data errors, then out of that resulting model space finds the minimum model. The maximum likelihood inverse we use,

$$(\mathbf{A}_k^{\mathrm{H}}\mathbf{R}_{\mathrm{dd}}^{-1}\mathbf{A}_k + \mathbf{R}_{\mathrm{mm}}^{-1})^{-1}\,\Delta\mathbf{m}_k = \mathbf{A}_k^{\mathrm{H}}\mathbf{R}_{\mathrm{dd}}^{-1}[\mathbf{d} - \mathbf{g}(\mathbf{m}_k)] + \mathbf{R}_{\mathrm{mm}}^{-1}(\mathbf{m}_0 - \mathbf{m}_k). \tag{1}$$

is similar to the stochastic inverse of Franklin (1970), and it gives the solution that minimizes a weighted sum of the variance of the difference between the model response and observed data, and the difference between the model parameters and an *a priori* model. In eq. (1), we make the following definitions:

$\mathbf{A}$ = sensitivity matrix ($A_{ij} = \partial d_i/\partial m_j$),
$\mathbf{d}$ = observed data vector,
$\mathbf{m}$ = model vector,
$\mathbf{g}$ = operator that maps a model $\mathbf{m}$ to the data space,
$\mathbf{R}_{\mathrm{dd}}$ = data covariance matrix,
$\mathbf{R}_{\mathrm{mm}}$ = model covariance matrix,
$\mathbf{m}_0$ = *a priori* model,
$\Delta\mathbf{m}$ = model changes for current iteration.

The sensitivity matrix is so named because its entries describe the perturbations in the data (or sensitivity of the data) due to perturbations in the model parameters. The superscript H stands for Hermitian, or complex conjugate transpose. Since the problem is non-linear, $\Delta\mathbf{m}$ describes only local changes so that one must iterate the inversion, each time updating the model. The model at the $k+1$ step is given by $\mathbf{m}_{k+1} = \mathbf{m}_k + \Delta\mathbf{m}_k$.

Equation (1) has a straightforward interpretation. The model changes calculated at each step represent a compromise between fitting the data and adhering to an *a priori* model. The compromise is weighted by the inverse of the data and model covariance matrices. We usually do not explicitly know $\mathbf{g}$, but we can calculate the model response $\mathbf{g}(\mathbf{m})$ by use of a forward modelling code.

In our implementation of the maximum likelihood inverse, we use logarithmic parameterization of the data and model parameters. This is useful for several reasons. First, it removes any bias associated with using either conductivity or resistivity as model parameters. Furthermore, it guarantees the positiveness of the model parameters (i.e. no negative resistivities or conductivities are allowed). In addition, it is the natural way of dealing with complex-valued data that are separated into log amplitude and phase ($\ln\mathbf{Z} = \ln|\mathbf{Z}| + i\theta$). Finally, logarithmic parametrization allows for larger changes in the model parameters as one iterates the inversion. For the electrical problem, where the resistivity can vary by several orders of magnitude, this reduces the total number of iterations needed to reach an acceptable solution.

The data covariance is a measure of the uncertainties in the data and can be computed or measured directly. The model covariance, on the other hand, is much more subjective because it cannot be measured directly. One can use the model covariance to apply a set of weights, filters, or constraints to the model parameters. Some simple constraints include imposing smoothing between neighbouring points, forcing certain parameters to be correlated with each other, or increasing/decreasing the freedom with which certain parameters can change. In the logarithmic parametrization scheme, the covariance matrices actually represent the covariances of the logarithms of the data and model parameters.

SENSITIVITY ANALYSIS AND RECIPROCITY

Madden (1990) and Mackie (1991) detail the sensitivity analysis and reciprocity relationship for the magnetotelluric problem. These are important issues in the relaxation inversion algorithm, and hence, we will review the salient points here.

One can write Maxwell's equations in matrix form as

$$\begin{bmatrix} -\sigma & \nabla x \\ \nabla x & -i\mu\omega \end{bmatrix}\begin{bmatrix} \mathbf{E} \\ \mathbf{H} \end{bmatrix} = \begin{bmatrix} \mathbf{J} \\ 0 \end{bmatrix} \tag{2}$$

where $\mathbf{J}$ are the media current sources, which for the MT problem are usually represented by a uniform current sheet far above the Earth's surface. If the media conductivity is perturbed by an amount $\delta\sigma$, then the $\mathbf{E}$ and $\mathbf{H}$ fields will also be perturbed by amounts δE and δH respectively

$$\begin{bmatrix} -(\sigma + \delta\sigma) & \nabla x \\ \nabla x & -i\mu\omega \end{bmatrix}\begin{bmatrix} \mathbf{E} + \delta\mathbf{E} \\ \mathbf{H} + \delta\mathbf{H} \end{bmatrix} = \begin{bmatrix} \mathbf{J} \\ 0 \end{bmatrix}. \tag{3}$$

Expanding this equation, subtracting it from eq. (2), and dropping second-order terms, we obtain

$$\begin{bmatrix} -\sigma & \nabla x \\ \nabla x & -i\mu\omega \end{bmatrix}\begin{bmatrix} \delta\mathbf{E} \\ \delta\mathbf{H} \end{bmatrix} = \begin{bmatrix} \mathbf{E}\,\delta\sigma \\ 0 \end{bmatrix}. \tag{4}$$

Thus, the $\mathbf{E}$ and $\mathbf{H}$ field perturbations satisfy the original EM equations except that the current sources are equal to the media perturbations multiplying the original $\mathbf{E}$ field. That is, one can solve for the field perturbations by doing forward problems with the proper current source distribution. We can rewrite eq. (2) in terms of Green's functions as (Kong 1986)

$$\begin{bmatrix} \mathbf{E}(r) \\ \mathbf{H}(r) \end{bmatrix} = \iiint d^3s\,\mathbf{G}(r, s)\cdot\mathbf{J}(s) \tag{5}$$

where $\mathbf{G}(r, s)$ is the dyadic Green's function. The elements of the dyadic Green's function are $G_i(r, s)_j$ where r is the observation point, s is the source point, i represents the observed field component ($i = 1, 2, \ldots, 6$ corresponding to $E_x, E_y, \ldots, H_z$) and j represents the source component ($j = 1, 2, 3$ corresponding to J_x, J_y, J_z). Using this notation, we can express eq. (4) as

$$\begin{bmatrix} \delta\mathbf{E}(r) \\ \delta\mathbf{H}(r) \end{bmatrix} = \iiint d^3s\,\mathbf{G}(r, s)\cdot\mathbf{E}(s)\,\delta\sigma(s). \tag{6}$$

Thus, it would appear that one needs to compute a dyadic Green's function for every point in the media where one wishes to determine the electrical properties.

Employing reciprocity, however, we find that we only need to compute Green's functions for every surface location where a measurement was made. One can derive the reciprocity relation from the bilinear identity along with the use of adjoint operators (Lanczos 1961), and it is given by

$$\underline{\mathbf{G}}_j^*(\mathbf{s}, \mathbf{r})_k = \mathbf{G}_k(\mathbf{r}, \mathbf{s})_j, \tag{7}$$

where $\underline{\mathbf{G}}$ refers to the Green's function for the adjoint operator. Since the curl operator and real constants are self adjoint (imaginary constants have a change in sign), the adjoint operator for Maxwell's equations is

$$\mathbf{D}^{\mathrm{H}} = \begin{bmatrix} -\sigma & \nabla x \\ \nabla x & i\mu\omega \end{bmatrix}. \tag{8}$$

Thus, the adjoint to Maxwell's equations is also the electromagnetic equations except with backwards time, or alternatively, negative frequencies. Since the adjoint in the frequency domain is simply the complex conjugated electromagnetic equations, $\underset{\sim}{\mathbf{G}}^* = \mathbf{G}$, we arrive at the desired reciprocity relation,

$$\mathbf{G}_j(\mathbf{s}, \mathbf{r})_k = \mathbf{G}_k(\mathbf{r}, \mathbf{s})_j. \tag{9}$$

This simple relation is of tremendous importance for solving the inverse problem. It means that the effect at the surface due to a source in the interior is equivalent to putting a source on the surface and solving for the effect in the interior (one must also take into account the proper interchange of vector components). In the magnetotelluric case, for example, the term $G_1(r, s)_2$ is the E_x effect at r (at the surface) due to a unit J_y dipole source at s (in the interior). By reciprocity, this is equal to $G_2(s, r)_1$, which is the E_y effect at s (in the interior) due to a unit J_x dipole source at r (at the surface). Similarly, the term $G_4(r, s)_2$ is the H_x effect at r (at the surface) due to a unit J_y dipole source at s (in the interior). Again by reciprocity, this is equal to $G_2(s, r)_4$, which is the E_y effect at s (in the interior) due to a unit H_x magnetic source at r (at the surface). Thus, if one were actually constructing the sensitivity matrix for the 3-D MT problem, one would only need to do forward modelling runs with sources at each surface measurement site instead of doing forward modelling runs with sources in each model block. This results in tremendous time savings. As we will see, however, our inversion algorithm does not actually require computing the sensitivity matrix, but the reciprocity relationship will still play an important role.

RELAXATION SOLUTION OF THE INVERSE PROBLEM

To carry out a linearized inversion using the maximum-likelihood inverse, one must determine the sensitivity matrix $\mathbf{A}$, compute $\mathbf{A}^{\mathrm{H}}\mathbf{A}$, and then invert $\mathbf{A}^{\mathrm{H}}\mathbf{A}$ (for the moment we are neglecting the data covariance weighting). In the 3-D case, computing $\mathbf{A}$ is a computationally enormous task even when using reciprocity. This is because the total number of forward problems needed to construct the sensitivity matrix is on the order of (no. measurement sites) x (no. frequencies). For a nominal 3-D problem, there might be 20 measurement sites and eight frequencies, thereby requiring on the order of 160 forward problems simply to set up the sensitivity matrix for one iteration of the inversion (of course one must also take into account source polarization and vector components of the fields so this estimate is just a lower bound). In addition, for this modest 3-D problem, there might be on the order of several thousand model parameters (20 × 20 × 10 model = 4000 model parameters). Inverting the matrix $\mathbf{A}^{\mathrm{H}}\mathbf{A}$, which has the dimension of the number of model parameters, is also a big computational task. To make 3-D inversions computationally tractable, a way must be found to circumvent these problems.

Smith & Booker (1991) derived an efficient iterative inversion technique that is called the rapid relaxation inverse (RRI). The important feature of their inversion scheme is that they approximate the lateral gradients of the electric and magnetic fields by the electric and magnetic field values at the previous iteration. Furthermore, they solve the forward problem with a relaxation scheme, thus yielding a quick and efficient algorithm. Their algorithm requires the solution of one forward problem per inversion iteration, but to date, has only been tested on 2-D models.

We take a somewhat different approach in that we use conjugate gradient relaxation techniques to solve the maximum likelihood equations (Madden & Mackie 1989; Madden 1990; Mackie 1991). At each iteration of the inversion, we use conjugate gradient relaxation to obtain an approximate solution for $\Delta\mathbf{m}$ in the maximum likelihood equations. This bypasses the need to do a large matrix inversion at each iteration of the inversion procedure. We can use standard conjugate gradient techniques because the system is positive definite and Hermitian. We should make clear that when using relaxation methods to solve the non-linear inversion, there are two levels of iteration involved. The outer loop is the iteration of the non-linear maximum likelihood equations. The inner loop is the iteration of the conjugate gradient relaxation procedure that is used to solve for the approximate $\Delta\mathbf{m}_k$ at each iteration of the inversion. It is our contention that only a few relaxation iterations are necessary at each inversion iteration since one must update the model and begin the whole process again. Furthermore, when using relaxation techniques, we never need to explicitly know $\mathbf{A}$, the sensitivity matrix. We only need to know the effect of $\mathbf{A}$ or $\mathbf{A}^{\mathrm{H}}$ multiplying a vector. We will show that one can do these operations without actually constructing the sensitivity matrix.

Our motivation for this procedure came from our experiences in implementing it in 1-D and 2-D geometries (Madden & Mackie 1989). In 1-D and 2-D geometries, there are no time savings involved in using relaxation methods because the models are not too large, and the sensitivity terms can be computed quickly and accurately. However, we found that in 1-D and 2-D, the relaxation technique gave results comparable to the direct results (those results obtained by explicitly computing the sensitivity matrix and solving the maximum likelihood equations by matrix inversion). We therefore extended these concepts to the 3-D inversion problem.

We will first outline our 3-D inversion algorithm before we discuss its details:

```
For $k = 1$ to max # inversion iterations                                      NON-LINEAR INVERSION
  $\mathbf{g}(\mathbf{m}_k)$                                                     response of current model
  $\mathbf{d} - \mathbf{g}(\mathbf{m}_k)$                                        data residuals
  $\mathbf{m}_0 - \mathbf{m}_k$                                                  model residuals
  $\mathbf{b} = \mathbf{A}^{\mathrm{H}}\mathbf{R}_{\mathrm{dd}}^{-1}[\mathbf{d} - \mathbf{g}(\mathbf{m}_k)] + \mathbf{R}_{\mathrm{mm}}^{-1}(\mathbf{m}_0 - \mathbf{m}_k)$      one forward problem with
                                                                                surface sources per frequency
  $\Delta\sigma_0 = 0, \ \mathbf{r}_0 = \mathbf{b}$                             initialize conjugate gradient
                                                                                algorithm
```

```
For $i = 1$ to max # relaxation iterations                    RELAXATION SOLUTION
    $\beta_i = \mathbf{r}_{i-1}^T\mathbf{r}_{i-1}/\mathbf{r}_{i-2}^T\mathbf{r}_{i-2}$                  $(\beta_0 = 0)$
    $\mathbf{p}_i = \mathbf{r}_{i-1} + \beta_i\mathbf{p}_{i-1}$                     $(\mathbf{p}_1 = \mathbf{r}_0)$ update search direction
    $\mathbf{Bp}_i = [\mathbf{A}^{\mathrm{H}}\mathbf{R}_{\mathrm{dd}}^{-1}\mathbf{A} + \mathbf{R}_{\mathrm{mm}}^{-1}]\mathbf{p}_i$        two forward problems per
                                                              frequency
    $\alpha_i = \mathbf{r}_{i-1}^T\mathbf{r}_{i-1}/\mathbf{p}_i^T\mathbf{Bp}_i$                 step length along search direction
    $\Delta\sigma_i = \Delta\sigma_{i-1} + \alpha_i\mathbf{p}_i$                   update model perturbations
    $\mathbf{r}_i = \mathbf{r}_{i-1} - \alpha_i\mathbf{Bp}_i$                      update residuals
end loop on relaxation iterations
    $\sigma_{k+1} = \sigma_k + \Delta\sigma$                        update model parameters
end loop on inversion iterations.
```

Mora (1987) developed a similar approach that used the conjugate gradient method of non-linear least squares to invert seismic reflection data for P-wave and S-wave velocities and density variations in two dimensions. He correctly showed that the operation of $\mathbf{A}^{\mathrm{H}}$ multiplying a vector was equivalent to doing one forward-modelling problem with sources distributed on the surface. In addition, he described how the operation of $\mathbf{A}^{\mathrm{H}}$ multiplying the surface data residuals is very similar to migration in that one properly images reflectors, but one does not obtain the correct amplitudes of the physical properties (Mora 1987). What he did not point out, however, is that the operation $\mathbf{Ax}$ is equivalent to doing one forward problem, except that in this case, as opposed to the $\mathbf{A}^{\mathrm{H}}$ problem, sources are distributed throughout the volume and not on the surface. Consequently, he approximated the term $[\mathbf{A}^{\mathrm{H}}\mathbf{R}_{\mathrm{dd}}^{-1}\mathbf{A} + \mathbf{R}_{\mathrm{mm}}^{-1}]$ in his stochastic inversion procedure by $\eta\mathbf{R}_{\mathrm{mm}}^{-1}$, where the model covariance matrix was assumed a diagonal matrix and η was a step length that ensured his error functional was minimized. Although Mora (1987) obtained good results with his algorithm for the inversion of seismic reflection data, he stated that for highly non-linear problems, the method probably would not work well. Electrical properties are very non-linear and can vary over several orders of magnitude, so we feel that it is important to treat the $\mathbf{A}^{\mathrm{H}}\mathbf{A}$ term more rigorously. Furthermore, since one can compute the $\mathbf{A}^{\mathrm{H}}\mathbf{A}$ effect on a vector with just two forward-modelling runs, as we will show, it is obviously worthwhile to proceed in this manner.

One can write the maximum likelihood eq. (1) as $\mathbf{Bu} = \mathbf{b}$ where $\mathbf{B} = (\mathbf{A}^{\mathrm{H}}\mathbf{R}_{\mathrm{dd}}^{-1}\mathbf{A} + \mathbf{R}_{\mathrm{mm}}^{-1})$, $\mathbf{u} = \Delta\mathbf{m}$, and $\mathbf{b} = \{\mathbf{A}^{\mathrm{H}}\mathbf{R}_{\mathrm{dd}}^{-1}(\mathbf{d} - \mathbf{g}(\mathbf{m})] + \mathbf{R}_{\mathrm{mm}}^{-1}(\mathbf{m}_0 - \mathbf{m})\}$. In the standard conjugate gradient procedure (Hestenes & Stiefel 1952), we need to know the initial residual, $\mathbf{r}_0 = \mathbf{b} - \mathbf{Bu}_0$, and we need to be able to compute what $\mathbf{B}$ multiplying an arbitrary vector yields at each iteration. Therefore, in terms of the maximum likelihood equation, we need to be able to compute the initial residual $\mathbf{r}_0 = \{\mathbf{A}^{\mathrm{H}}\mathbf{R}_{\mathrm{dd}}^{-1}[\mathbf{d} - \mathbf{g}(\mathbf{m})] + \mathbf{R}_{\mathrm{mm}}^{-1}(\mathbf{m}_0 - \mathbf{m})\}$, and at each relaxation step the effect of $(\mathbf{A}^{\mathrm{H}}\mathbf{R}_{\mathrm{dd}}^{-1}\mathbf{A} + \mathbf{R}_{\mathrm{mm}}^{-1})$ on the some vector. Since $\mathbf{R}_{\mathrm{dd}}^{-1}$ and $\mathbf{R}_{\mathrm{mm}}^{-1}$ are known, we only need to be able to compute quantities like $\mathbf{q} = \mathbf{A}^{\mathrm{H}}\mathbf{y}$ and $\mathbf{y} = \mathbf{Ax}$. We can compute these quantities using one forward-modelling run each without ever constructing the actual sensitivity matrix.

The sensitivity matrix describes the perturbation in a surface measurement due to a small perturbation in a physical property of the media. Although we have not stated explicitly which surface measurements we will invert for, they will involve derivatives of the surface $\mathbf{E}$ and $\mathbf{H}$ fields, and will consist of terms like $\partial E_x/\partial\sigma$, $\partial E_y/\partial\sigma$, $\partial H_x/\partial\sigma$, $\partial H_y/\partial\sigma$. The row space of the sensitivity matrix corresponds to the model parameter space. That is, the ith element in a particular row represents the perturbation in one of the surface fields at one site, one frequency, and one polarization due to a perturbation of the ith model parameter. If one were actually constructing the sensitivity matrix, one would put a unit current source at each surface measurement site for each frequency, and then compute the fields in the interior due to that current source (this utilizes the reciprocity relationship). Since from eq. (6) it follows that $\partial(\mathbf{E}, \mathbf{H})/\partial\sigma = \mathbf{G}(r, s) \cdot \mathbf{E}(s) \cdot (\text{volume})$, these fields multiplying the original $\mathbf{E}$ fields would yield the values $\partial\mathbf{E}/\partial\sigma$ and $\partial\mathbf{H}/\partial\sigma$. One could then compute the desired sensitivity terms.

For the relaxation solution, however, we do not need to construct the sensitivity matrix. Notice that the sensitivity matrix multiplying an arbitrary vector $\mathbf{x}$ is simply a sum over all model parameters of the sensitivity term multiplying the vector component for that model block:

$$\mathbf{p} = \mathbf{Ax} \sim \sum_{\substack{\text{model}\\ \text{parameters}\\ j}} \frac{\partial(E, H)}{\partial\sigma_j} x_j. \tag{10}$$

Each component of the vector $\mathbf{p}$ is defined at a particular surface measurement site and a particular frequency. We can use the principle of linear superposition to compute the vector $\mathbf{p}$ without computing the sensitivity matrix $\mathbf{A}$. The superposition principle states that if $\mathbf{T}$ is the linear system transformation, then for any two inputs in_1 and in_2 and any scalar constant c,

$$\begin{aligned} &\mathbf{T}[\mathrm{in}_1 + \mathrm{in}_2] = \mathbf{T}[\mathrm{in}_1] + \mathbf{T}[\mathrm{in}_2], \text{ and}\\ &\mathbf{T}[c\ \mathrm{in}_1] = c\mathbf{T}[\mathrm{in}_1]. \end{aligned} \tag{11}$$

If this is true for two inputs, then it can be shown to be true for any number of inputs. In our case, the linear system $\mathbf{T}$ is represented by Maxwell's equations, and the inputs are current and magnetic sources. We saw that the sensitivity in the surface fields is the response due to a unit current source in the media multiplying the original $\mathbf{E}$ field at that location. Therefore, the vector $\mathbf{p}$ goes like

$$\begin{aligned} &\mathbf{T}[J_0(1)E_0(1)]x(1) + \mathbf{T}[J_0(2)E_0(2)]x(2) + \cdots\\ &\quad + \mathbf{T}[J_0(n)E_0(n)]x(n) \end{aligned} \tag{12}$$

where $J_0(i)$ is a unit current source, $E_0(i)$ is the original $\mathbf{E}$ field, $x(i)$ is the component of the vector $\mathbf{x}$, and i specifies the model block. Employing the superposition principle allows us, however, to alternatively express the vector $\mathbf{p}$ as

$$\mathbf{T}[J_0(1)E_0(1)x(1)+J_0(2)E_0(2)x(2)+\cdots$$
$$+J_0(n)E_0(n)x(n)]. \tag{13}$$

We see that this is equivalent to putting all the sources in the media at the same time and computing one forward problem to determine the effect at the surface. The surface fields from this calculation give the desired result of the sensitivity matrix multiplying the vector $\mathbf{x}$ at every surface measurement site at one frequency, and this is done without ever actually computing the sensitivity matrix. We have defined one forward problem as corresponding to two polarizations at one frequency. We see that we have gained tremendously in terms of computation time by using this approach. Computing the sensitivity matrix by the traditional, 'non-reciprocal' method required doing one forward problem for each model parameter for each frequency. Utilizing the reciprocity relationships reduces the number of forward problems to the number of surface measurement sites for each frequency. However, our approach for computing $\mathbf{Ax}$ requires only one forward problem for each frequency.

Likewise, we can employ a similar procedure for $\mathbf{A}^{\mathrm{H}}$ multiplying an arbitrary vector $\mathbf{y}$. We express the vector $\mathbf{q}=\mathbf{A}^{\mathrm{H}}\mathbf{y}$ in terms of another vector $\mathbf{Q}$ where $\mathbf{Q}=\mathbf{A}^T\mathbf{y}^*$ and $\mathbf{q}=\mathbf{Q}^*=(\mathbf{A}^T\mathbf{y}^*)^*=\mathbf{A}^{\mathrm{H}}\mathbf{y}$. The column space of the sensitivity matrix corresponds to the data space, and each element in a particular column corresponds to a perturbation in one of the surface fields at one of the measurement sites and one frequency for one particular model parameter. The vector $\mathbf{Q}$ therefore goes as,

$$\mathbf{Q}\sim\sum_{\mathrm{freq}}\sum_{\mathrm{surf}}\frac{\partial(E,H)_{\mathrm{surf}}}{\partial\sigma}y^*_{\mathrm{surf}}, \tag{14}$$

which is a sum over all frequencies and all surface measurements of the sensitivity term multiplying the component of the vector $\mathbf{y}$ for that particular frequency and surface measurement, the result being given at an interior model block. As before, we can use the principle of linear superposition to compute the vector $\mathbf{Q}$. Using reciprocity, we saw that the sensitivity in the surface fields is the response due to unit current or magnetic sources on the surface, depending on whether one is solving for the perturbed $\mathbf{E}$ fields or $\mathbf{H}$ fields. Therefore, the vector $\mathbf{Q}$ goes like

$$\{\mathbf{T}[S(1)]y^*(1)+\boldsymbol{T}[S(2)]y^*(2)+\cdots+$$
$$+\mathbf{T}[S(k)]y^*(k)\}E_0(i) \tag{15}$$

where $S(k)$ represents a unit surface current or magnetic source, $y(k)$ is the component of the vector $\mathbf{y}$, and k specifies the surface location. As before, $E_0(i)$ is the original $\mathbf{E}$ field in the ith model block. The principle of linear superposition, however, allows us to rewrite the vector $\mathbf{Q}$ as

$$\{\mathbf{T}[S(1)y^*(1)+S(1)y^*(1)+\cdots+S(k)y^*(k)]\}E_0(i), \tag{16}$$

which corresponds to one forward-modelling run with all the sources distributed at the surface simultaneously. This computation is similar to a downwards propagation of surface fields, but it is actually a downwards propagation in *backwards* time. Since we are really computing $\mathbf{A}^{\mathrm{H}}\mathbf{y}$, and since by reciprocity $G_j(\mathbf{s},\mathbf{r})_i=G_i(\mathbf{r},\mathbf{s})_j$, the $\mathbf{A}^{\mathrm{H}}\mathbf{y}$ operation is actually a sum of the complex conjugate of the Green's function responses to the surface sources, and the complex conjugate of the Green's function involves negative frequencies, or equivalently, backwards time.

Therefore, we can carry out iterations of the conjugate gradient procedure without ever explicitly computing the sensitivity matrix. For each iteration we need to compute two forward problems per frequency with sources distributed throughout the volume and across the surface. At the initial iteration, one additional forward problem is required to compute $\mathbf{A}^{\mathrm{H}}$ multiplying the data residuals. The surface sources for this calculation are simply $\mathbf{R}_{\mathrm{dd}}^{-1}[\mathbf{d}-\mathbf{g}(\mathbf{m}_k)]$, which are given at each surface measurement location and frequency. For each iteration of the conjugate gradient procedure, one must first compute $\mathbf{Ap}_i$ where $\mathbf{p}_i$ is the search direction at the ith iteration. The sources for this calculation are therefore the vector $\mathbf{p}_i$, which has a value at each interior model block. One must then compute $\mathbf{A}^{\mathrm{H}}\mathbf{R}_{\mathrm{dd}}^{-1}\mathbf{Ap}_i$. The sources for this calculation therefore are $\mathbf{R}_{\mathrm{dd}}^{-1}\mathbf{Ap}_i$, which is a vector with values given at each surface measurement location and frequency.

FORWARD MODELLING

The forward-modelling routine used in our 3-D MT inversion algorithm is the direct solution described in Mackie (1991) and Mackie, Madden & Wannamaker (1993). This algorithm solves for the electromagnetic fields in a 3-D model by propagating an impedance matrix from the bottom of the model up to the Earth's surface (this is very similar to a Ricatti equation approach). The impedance matrix relates the horizontal electric fields within a layer to the magnetic fields within that same layer. Instead of doing a direct decomposition on the entire 3-D operator, this algorithm achieves its speed by doing a smaller matrix inversion for each layer in the model. The advantage of using this algorithm in the inversion routine is that once the forward problem has been solved to compare the model response to the observed data, one can then quickly compute the fields for any source configuration if the layer inverses have been stored. Thus, additional conjugate gradient relaxation steps, which comprise the inner loop of the inversion algorithm, come at little added time and cost above that required to solve the problem once for the current model response.

Alternatively, we have developed a conjugate direction relaxation algorithm for the 3-D MT forward problem (Mackie 1991; Mackie & Madden 1993). This is an approximate algorithm that is accurate, quick, and requires much less memory than the direct solution. The advantage of using this algorithm over the direct algorithm is that it is much quicker for solving the 3-D MT forward problem. The disadvantage is that completely independent forward solution relaxations are required to compute the responses for the different source configurations needed in the inversion relaxations (the inner loop of the inversion algorithm). This would start to be an issue if many relaxation steps (the inner loop of the inversion) were desired at each inversion iteration (the outer loop of the inversion).

3-D MT DATA PARAMETERS

In magnetotellurics, we typically measure the impedance tensor over a range of frequencies at each measurement site. One can write the impedance tensor, which relates the horizontal **E** fields to the horizontal **H** fields, as

$$\begin{bmatrix} E_x(\omega) \\ E_y(\omega) \end{bmatrix} = \begin{bmatrix} Z_{xx} & Z_{xy} \\ Z_{yx} & Z_{yy} \end{bmatrix} \begin{bmatrix} H_x(\omega) \\ H_y(\omega) \end{bmatrix}. \tag{17}$$

Assuming that there are two linearly independent source polarizations, one can express the impedance tensor as

$$\mathbf{Z} = \frac{\begin{bmatrix} (E_{x1}H_{y2} - E_{x2}H_{y1}) & (E_{x2}H_{x1} - E_{x1}H_{x2}) \\ (E_{y1}H_{y2} - E_{y2}H_{y1}) & (E_{y2}H_{x1} - E_{y1}H_{x2}) \end{bmatrix}}{H_{x1}H_{y2} - H_{x2}H_{y1}} \tag{18}$$

where the subscripts 1 and 2 refer to the fields from the two different source polarizations. At first glance, it might seem logical to just invert for the four complex impedance elements. The difficulty with this, however, is that in many situations, Z_{xx} and Z_{yy} are zero or close to zero. This is a problem when using logarithmic parameterzation. Since $\partial \ln Z/\partial \ln \sigma = \sigma/Z(\partial Z/\partial \sigma)$, we see the possibility for a division by zero if any of the impedance elements are zero.

Instead, we take an approach similar to the eigenstate formulations of Eggers (1982), and LaTorraca, Madden & Koringa (1986). They were mainly concerned with extracting physically meaningful scalar parameters from the complex impedance tensor, especially for 3-D geometries. We use a somewhat simpler approach and decompose the impedance tensor in terms of two complex vector fields (*cf.* Jackson 1975). We consider the **E** field due to a unit H_x field, which is $\mathbf{E}_{Hx} = [\hat{\mathbf{x}}Z_{xx} + \hat{\mathbf{y}}Z_{yx}]$, and the **E** field due to a unit H_y field, which is $\mathbf{E}_{H_y} = [\hat{\mathbf{x}}Z_{xy} + \hat{\mathbf{y}}Z_{yy}]$. These two vectors can serve as the basis set for the impedance tensor. Every complex vector can be described in terms of four scalar quantities: amplitude, phase, direction, and ellipticity. These quantities for the **E** vectors given above are closely related to the apparent resistivity, phase, direction, and ellipticity values usually computed from field data. In our 3-D inversion scheme, we invert for the amplitude, phase, and direction of the **E** vectors described above. Ellipticities are diffcult to determine accurately from field data, and therefore, we have excluded them from our inversion scheme. The derivatives of these quantities are found in Appendix 1.

BOUNDARY CONDITIONS

Up to this point, we have not dealt specifically with the issues of the boundary conditions. The 3-D forward modelling algorithm described in Mackie *et al.* (1993) assigns tangential **H** field values on the boundaries of the 3-D model. In this work, we use a modified version of the algorithm that assumes the model is repeated in the horizontal directions; that is, no side boundary terms are assigned. In this way, we avoid having to deal with the boundary values and their effect on the sensitivity terms. We made this assignment to simplify the programming and to test our concepts about relaxation inversion. In the future, however, we will include the boundaries and their effect on the sensitivity terms in order to deal with more realistic earth models.

RESULTS FOR THEORETICAL DATA

In this section, we present results using the relaxation inversion routine for inverting numerically computed, noise-free 3-D magnetotelluric data. We have not extensively tested the algorithm on a wide variety of models, but nonetheless, we can demonstrate that even at this stage in the development of the algorithm, it works as intended.

The model used to generate the data is shown in Fig. 1 and consists of a conductive, 3-D inhomogeneity embedded at a shallow depth in an otherwise layered media. The

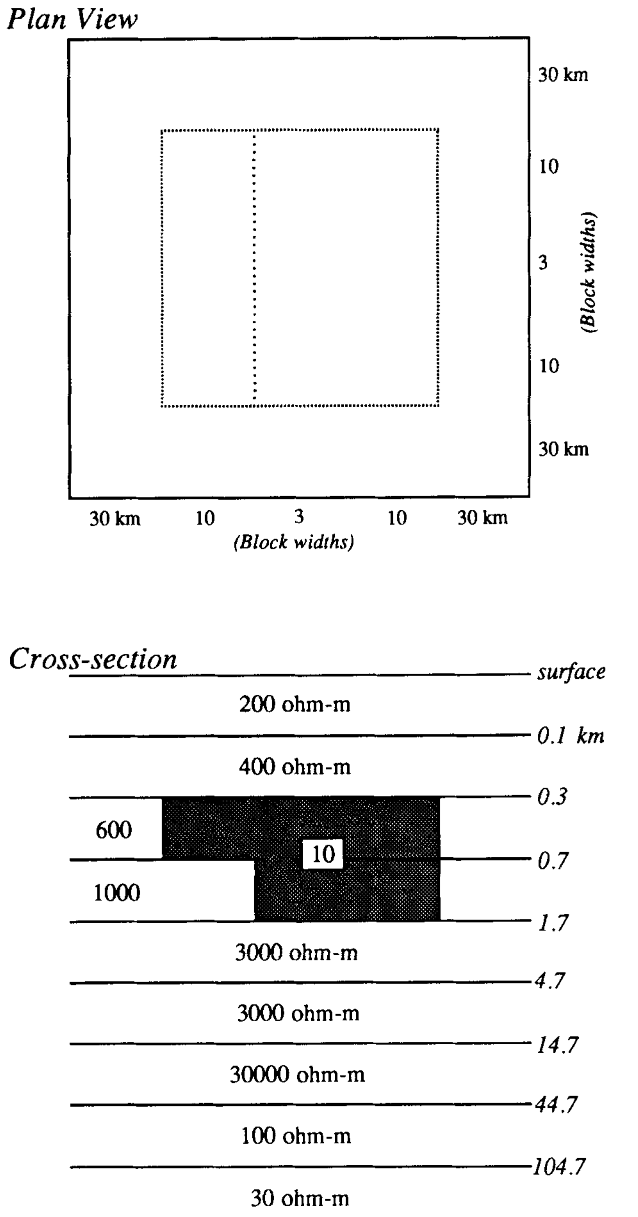

Figure 1. The 3-D model used to generate synthetic MT data. These data were used to test the 3-D MT inversion algorithm. The model was 5 blocks by 5 blocks by 9 layers. A 10 Ωm inhomogeneity was buried in an otherwise layered earth.

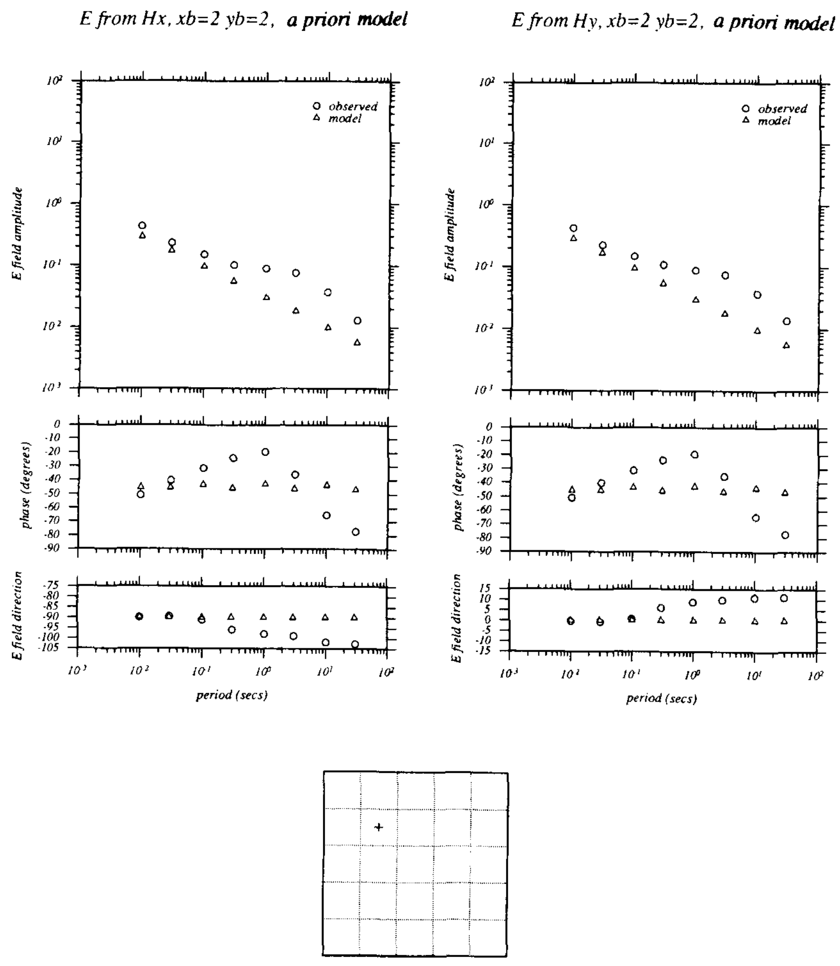

Figure 2. The observed and *a priori* model responses for the surface location $xb = 2$, $yb = 2$. Shown are the amplitudes, phases (in degrees), and directions (in degrees clockwise from the positive x-axis, which runs to the right) versus period. We define these values in the appendix.

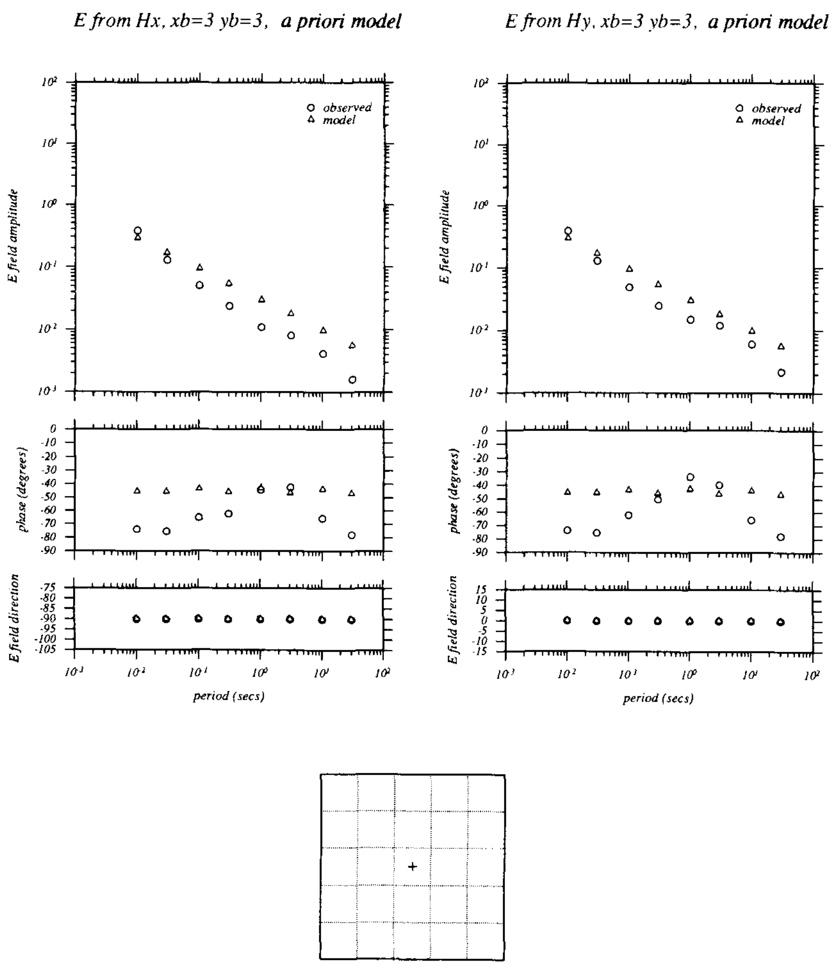

Figure 3. The observed and *a priori* model responses for the surface location $xb = 3$, $yb = 3$. Shown are the amplitudes, phases (in degrees), and directions (in degrees clockwise from the positive x-axis, which runs to the right) versus period. We define these values in the appendix.

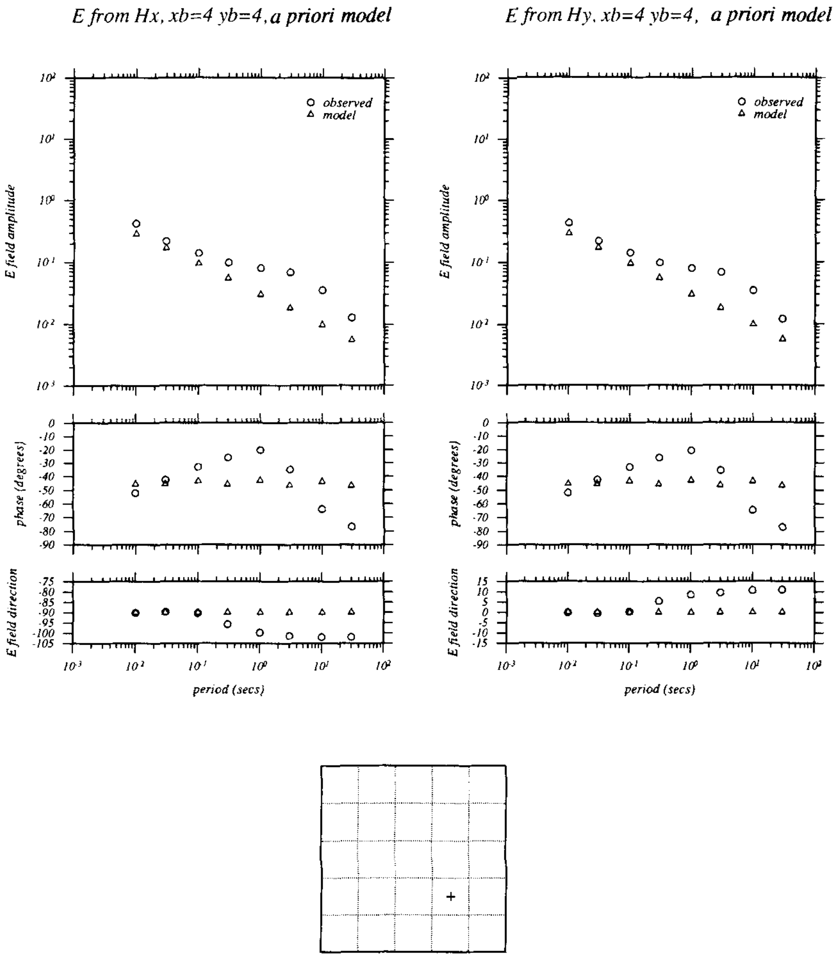

Figure 4. The observed and *a priori* model responses for the surface location $xb = 4$, $yb = 4$. Shown are the amplitudes, phases (in degrees), and directions (in degrees clockwise from the positive x-axis, which runs to the right) versus period. We define these values in the appendix.

model is composed laterally of five blocks in the x-direction and five blocks in the y-direction. Vertically, there are eight layers and a half-space. The impedance for a 30 Ωm halfspace is used at the bottom of the layers in the 3-D model, and it was held fixed throughout all the inversions. Data were output for eight frequencies equally spaced in the logarithm of period from 0.01 s to 30 s. The *a priori* model used for all inversions had the same dimensions as the one used to generate the 'observed data', but with a uniform resistivity of 100 Ωm and a 30 Ωm half-space.

Shown in Figs 2, 3 and 4 are the observed and *a priori* model responses for three different locations on the surface of the 3-D model. These three sites were chosen because it would be too voluminous to show the responses at all the sites, and the sites at $xb = 2$, $yb = 2$ and $xb = 4$, $yb = 4$ represent sites where 3-D effects are the severest. Fig. 5 shows the error progression versus inversion iteration for an inversion with five relaxation steps per inversion. We define the data error as

$$\sqrt{\frac{\sum_{ndat}\sum_{nfreq}\sum_{npol}\left[\left(\ln\frac{|E_o|}{|E_m|}\right)^2+(\theta_o-\theta_m)^2+(\Psi_o-\Psi_m)^2\right]}{ndat\times nfreq\times 2}}\times 100 \qquad (19)$$

where θ and ψ are given in radians, the subscripts o and m refer to observed and model respectively, $n$dat is the number of data locations, $n$freq is the number of frequencies, and $n$pol is the number of polarizations ($n\text{pol} = 2$). The right-hand-side error is the error in the right-hand side of the maximum likelihood equations, eq. (1). When the maximum likelihood equations are exactly satisfied, the right-hand side goes to zero. This does not necessarily correspond to the solution in which the model responses and observed data are exactly equal because the fit is actually a compromise between fitting the data and adhering to the *a priori* model. We computed the right-hand-side error plotted in Fig. 5 by the following equation, where $r(i)$ represents the element of the right-hand side corresponding to the ith model block:

$$\sqrt{\frac{\sum_{i=1}^{n\text{mod}} r(i)r(i)}{n\text{mod}}}\times 100. \qquad (20)$$

For this example, we used five relaxation steps per inversion iteration, and we assumed that both the model covariance and the data covariance matrices were of the form $\sigma^2\mathbf{I}$. That is, we set $\mathbf{R}_{\text{dd}} = \sigma_{\text{d}}^2\mathbf{I}$ and $\mathbf{R}_{\text{mm}} = \sigma_{\text{m}}^2\mathbf{I}$, where the ratio

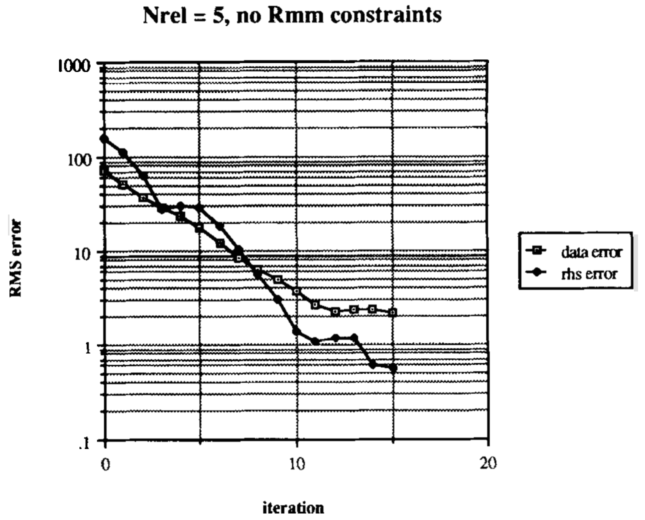

Figure 5. The RMS error progression as a function of the inversion iteration step. This particular run used five conjugate gradient relaxation steps per inversion iteration. The ratio of σ_d^2/σ_m^2 was equal to 3×10^{-5}. We defined the data error in eq. (19), and the right-hand-side error (rhs error) in eq. (20). The right-hand-side error should go to zero when one obtains the solution to the maximum likelihood equations.

```
Number of relaxation steps = 5                          Rmmfac=3.0e-5
Number of inversion iterations = 15                     Damping=1.0
No Rmm constraints

RESULTS FROM INVERSION PROGRAM                                  ACTUAL MODEL
==============================                          ==============================

 layer1                                                 layer 1
   239   237   242   240   235                              200   200   200   200   200
   245   263   266   267   234                              200   200   200   200   200
   231   260   254   250   240                              200   200   200   200   200
   246   263   266   269   235                              200   200   200   200   200
   236   235   240   240   240                              200   200   200   200   200
 layer2                                                 layer 2
   344   347   353   353   343                              400   400   400   400   400
   333   194   248   198   367                              400   400   400   400   400
   306   249   253   267   397                              400   400   400   400   400
   333   193   248   199   369                              400   400   400   400   400
   340   344   351   353   350                              400   400   400   400   400
 layer3                                                 layer 3
   531   537   491   510   545                              600   600   600   600   600
   578    11    11    10   485                              600    10    10    10   600
   586    10    13    13   458                              600    10    10    10   600
   578    11    11    10   486                              600    10    10    10   600
   527   536   488   509   553                              600   600   600   600   600
 layer4                                                 layer 4
  1196  1281  1218  1155  1278                             1000  1000  1000  1000  1000
  1280   154     9     9  1139                             1000  1000    10    10  1000
  1302   121    14    13  1228                             1000  1000    10    10  1000
  1281   154     9     9  1139                             1000  1000    10    10  1000
  1193  1290  1213  1149  1275                             1000  1000  1000  1000  1000
 layer5                                                 layer 5
  2346  2651  2681  2725  2444                             3000  3000  3000  3000  3000
  2792   490    99   126  2637                             3000  3000  3000  3000  3000
  2833   231    99    82  2630                             3000  3000  3000  3000  3000
  2801   491    99   125  2625                             3000  3000  3000  3000  3000
  2366  2676  2686  2710  2418                             3000  3000  3000  3000  3000
 layer6                                                 layer 6
  3041  4717  4813  4938  3033                             3000  3000  3000  3000  3000
  4811   829   356  1021  4787                             3000  3000  3000  3000  3000
  4980   192   154   111  4462                             3000  3000  3000  3000  3000
  4826   828   355  1016  4804                             3000  3000  3000  3000  3000
  3074  4713  4823  4919  3001                             3000  3000  3000  3000  3000
 layer7                                                 layer 7
 19461 14747  9627 14495 22294                            30000 30000 30000 30000 30000
 18683   373   173   365 12812                            30000 30000 30000 30000 30000
 17577   149    94   131  5318                            30000 30000 30000 30000 30000
 18702   368   172   361 12847                            30000 30000 30000 30000 30000
 19390 14820 10843 14680 22130                            30000 30000 30000 30000 30000
 layer8                                                 layer 8
    90    86    85    85    88                              100   100   100   100   100
    87   114   102   102    86                              100   100   100   100   100
    86   107    99    99    86                              100   100   100   100   100
    87   114   102   102    86                              100   100   100   100   100
    90    86    85    86    89                              100   100   100   100   100
```

Figure 6. On the left is the resulting model after 15 iterations of the inversion procedure, and on the right, for comparison, is the actual model. The resistivity values for each block are in Ωm. We held the 30 Ωm bottom half-space, which is not shown here, fixed during the inversion.

$\sigma_d^2/\sigma_m^2 = 3 \times 10^{-5}$. This simply means that the variance of the fit to the *a priori* model was assumed greater than the variance of the data errors. In other words, the data were trusted much more than the *a priori* model, and the fit to the observed data was more important in the inversion than was the fit to the *a priori* model. We added an additional damping term to stabilize the inversions and we tied it to the error in the right-hand side so that the damping decreased as the error decreased. The damping term was equal to $(1.0) * (1/\sigma_d^2) * (\text{right-hand-side error})$. Damping terms reduce the influence of the small eigenvalues in the early stages of the inversion, then allow them to become more influential at the latter stages. Damping terms should depend on the magnitudes of the eigenvalues, but the damping term added in these 3-D inversions came about largely from our experience with 2-D inversions and experimenting with the damping factor for 3-D inversions. This damping is somewhat larger than what we are use to dealing with in 2-D inversions, but this is probably due to the increased degrees of freedom in 3-D inversions as compared to 2-D inversions.

For this particular example, 15 iterations reduced the error in the fit to the data to approximately 2 per cent and the right-hand-side error to 0.6 per cent. We show both the resulting model and the actual model for this inversion in Fig. 6, where the numbers are the resistivity values in Ωm. Note that the resistivity structure of the top four layers is fairly well resolved, but that the image of the conductive feature is smeared into the lower layers. This is a common by-product of many inversion schemes where one only has data coverage on one side of the feature one is trying to image. The inherent difficulty magnetotellurics has of resolving resistive bodies, especially if they are underneath more conductive layers or bodies, enhances this phenomenon. Figs 7, 8 and 9 show the responses for the model of Fig. 6 for the three surface locations shown earlier. We see excellent agreement in both polarizations for amplitudes, phases and directions. We find similarly good fits to the observed data at all other locations, but we do not show them here.

As a further example, we inverted the same data starting from the same *a priori* model, except this time, we included $\mathbf{R}_{mm}$ constraints to keep the bottom four layers 1-D (that is, in each layer, all the resistivity values in that layer are tied together). We ran this inversion for different numbers of relaxation iterations per inversion iteration. Fig. 10 shows the resulting model after 15 inversion iterations using only one relaxation step per inversion iteration. Fig. 11 shows the resulting model after 15 inversion iterations using three relaxation steps per inversion iteration, and Fig. 12 shows the resulting model after 15 inversion iterations using 10 relaxation steps per inversion iteration. The results using only one relaxation step per iteration are surprisingly good, although the results using 10 relaxation steps per iteration are clearly the best. Tying together the resistivities within each of the bottom four layers has removed the smearing caused by the shallow conductive body and improved the estimates of the resistivities for these layers. Of course, in this example, we had the luxury of knowing beforehand that the bottom four layers should be uniform, but nonetheless, this demonstrates the usefulness of *a priori* information if it

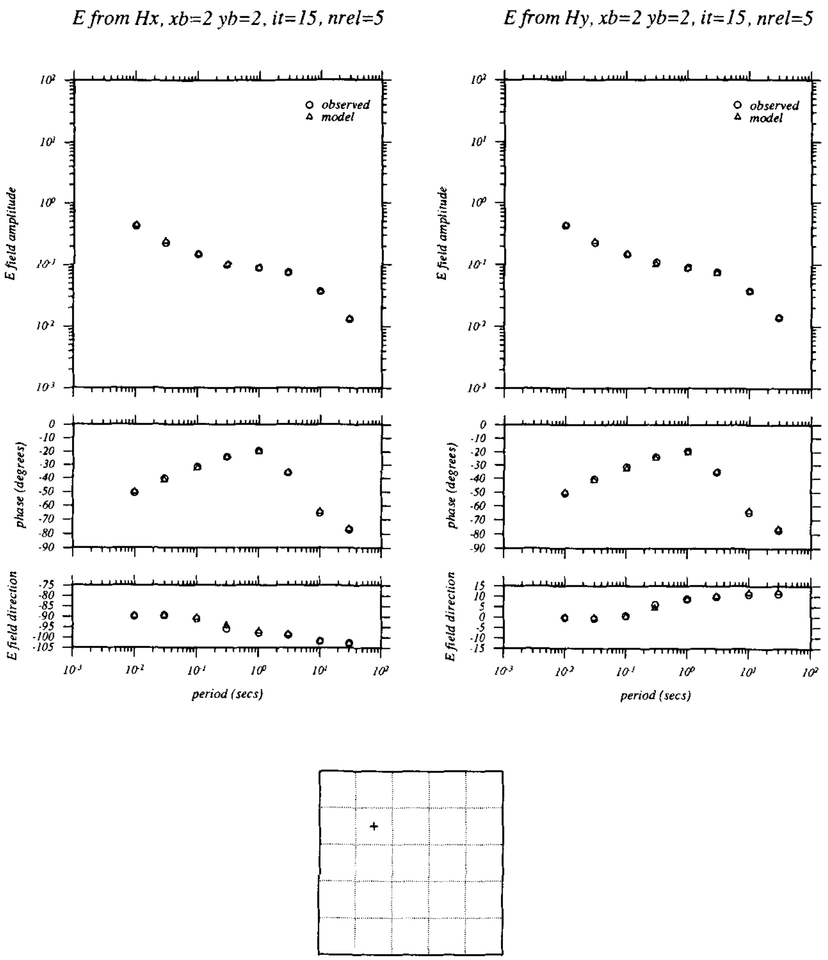

Figure 7. Here we plot the responses for the model of Fig. 6 for the surface location $xb = 2$, $yb = 2$ and the observed data for the same location.

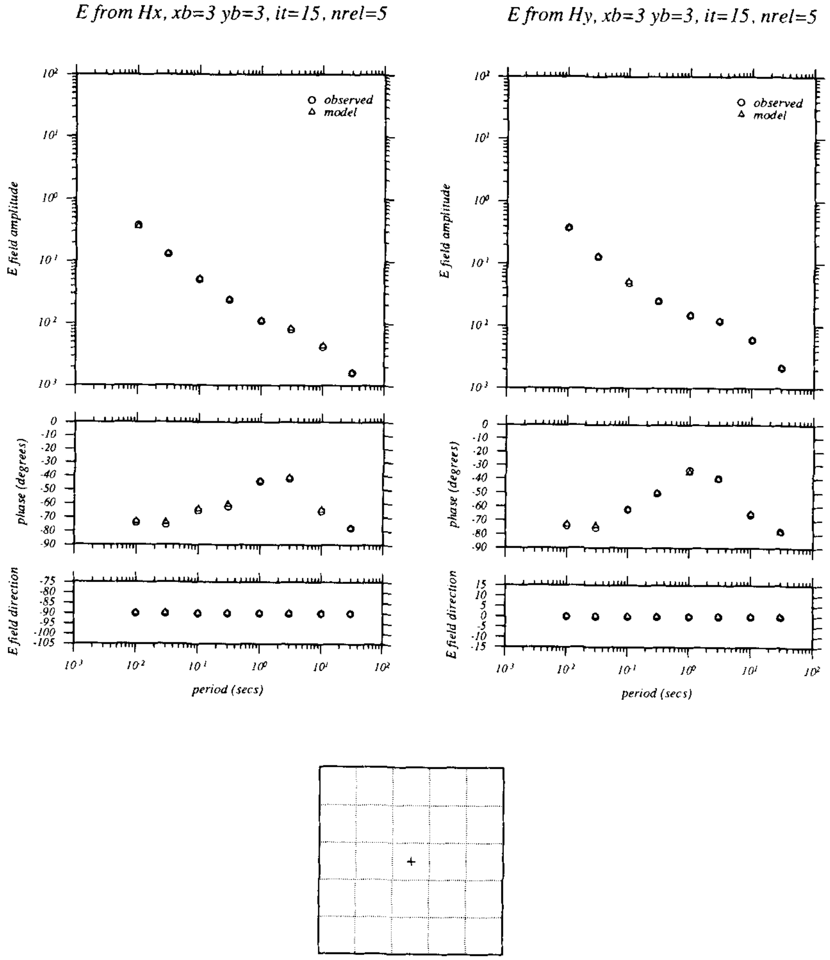

Figure 8. Here we plot the responses for the model of Fig. 6 for the surface location $xb = 3$, $yb = 3$ and the observed data for the same location.

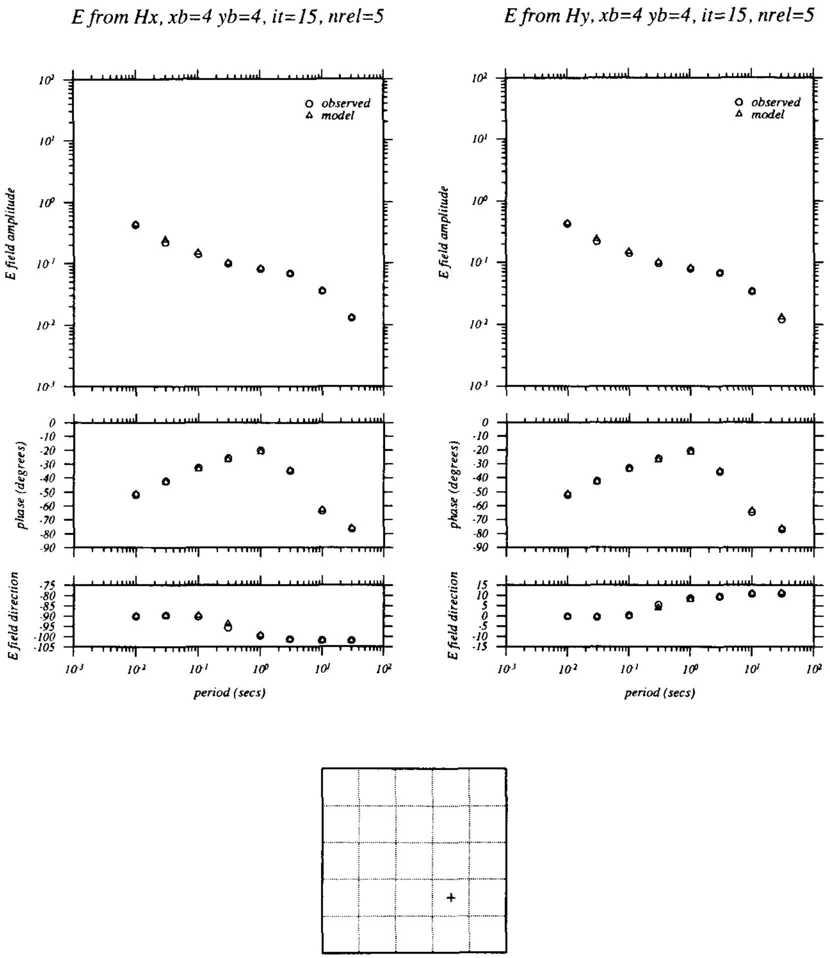

Figure 9. Here we plot the responses for the model of Fig. 6 for the surface location $xb = 4$, $yb = 4$ and the observed data for the same location.

```
Number of relaxation steps = 1                          Rmmfac=3.0e-5
Number of inversion iterations = 15                     Damping=1.0
Rmm constraints to keep layers 5,6,7,8 1D

RESULTS FROM INVERSION PROGRAM                  ACTUAL MODEL
==============================            ==============================

 layer1                                    layer 1
  204   218   218   220   194               200   200   200   200   200
  213   253   282   258   223               200   200   200   200   200
  205   270   207   267   219               200   200   200   200   200
  213   253   282   258   223               200   200   200   200   200
  204   218   218   220   194               200   200   200   200   200
 layer2                                    layer 2
  381   357   339   345   362               400   400   400   400   400
  345   196   249   198   356               400   400   400   400   400
  330   244   194   240   349               400   400   400   400   400
  345   196   249   198   356               400   400   400   400   400
  381   357   339   345   362               400   400   400   400   400
 layer3                                    layer 3
  690   565   508   497   662               600   600   600   600   600
  553    11    16     8   514               600    10    10    10   600
  543    16    24    14   533               600    10    10    10   600
  553    11    16     8   514               600    10    10    10   600
  690   565   508   497   662               600   600   600   600   600
 layer4                                    layer 4
 1291  1223  1113  1078  1268              1000  1000  1000  1000  1000
 1186    68    16     9  1108              1000  1000    10    10  1000
 1206    64    22    13  1128              1000  1000    10    10  1000
 1186    68    16     9  1108              1000  1000    10    10  1000
 1291  1223  1113  1078  1268              1000  1000  1000  1000  1000
 layer5                                    layer 5
 1795  2493  2179  2536  1753              3000  3000  3000  3000  3000
 2411   450   244   263  2400              3000  3000  3000  3000  3000
 2220   333   216   209  2112              3000  3000  3000  3000  3000
 2411   450   244   263  2400              3000  3000  3000  3000  3000
 1795  2493  2179  2536  1753              3000  3000  3000  3000  3000
 layer6                                    layer 6
 4626  6279  4062  5827  4559              3000  3000  3000  3000  3000
 6293   866   570   936  5873              3000  3000  3000  3000  3000
 4600   465   360   368  3722              3000  3000  3000  3000  3000
 6293   866   570   936  5873              3000  3000  3000  3000  3000
 4626  6279  4062  5827  4559              3000  3000  3000  3000  3000
 layer7                                    layer 7
15301  7318  2739  7194 14479             30000 30000 30000 30000 30000
 8984   617   339   604  6277             30000 30000 30000 30000 30000
 3604   325   222   297  2295             30000 30000 30000 30000 30000
 8984   617   339   604  6277             30000 30000 30000 30000 30000
15301  7318  2739  7194 14479             30000 30000 30000 30000 30000
 layer8                                    layer 8
   63    74    83    74    63               100   100   100   100   100
   74    88    92    88    74               100   100   100   100   100
   83    92    93    92    83               100   100   100   100   100
   74    88    92    88    74               100   100   100   100   100
   63    74    83    74    63               100   100   100   100   100
```

Figure 10. The resulting model after 15 iterations of the inversion procedure when we constrained the bottom four layers to be 1-D. For this run, we used 1 relaxation step per inversion iteration. We show the actual model on the right for comparison.

```
Number of relaxation steps = 3                  Rmmfac=3.0e-5
Number of inversion iterations = 15             Damping=1.0
Rmm constraints to keep layers 5,6,7,8 1D

RESULTS FROM INVERSION PROGRAM                      ACTUAL MODEL
==============================          ============================

 layer1                                 layer 1
   241   247   244   235   234            200   200   200   200   200
   247   262   276   260   231            200   200   200   200   200
   231   268   255   261   247            200   200   200   200   200
   247   256   278   267   250            200   200   200   200   200
   235   231   245   253   235            200   200   200   200   200
 layer2                                 layer 2
   342   366   356   339   342            400   400   400   400   400
   357   178   255   178   338            400   400   400   400   400
   327   255   255   262   383            400   400   400   400   400
   346   175   256   182   373            400   400   400   400   400
   334   334   357   374   343            400   400   400   400   400
 layer3                                 layer 3
   507   559   464   479   528            600   600   600   600   600
   577    10    11     8   467            600    10    10    10   600
   535    10    15    13   450            600    10    10    10   600
   538    11    11     8   517            600    10    10    10   600
   505   503   465   532   524            600   600   600   600   600
 layer4                                 layer 4
  1193  1410  1263  1247  1321           1000  1000  1000  1000  1000
  1441   145    10    10  1221           1000  1000    10    10  1000
  1438   102    14    13  1217           1000  1000    10    10  1000
  1382   148    10    10  1242           1000  1000    10    10  1000
  1222  1349  1260  1291  1259           1000  1000  1000  1000  1000
 layer5                                 layer 5
  2116  2347  2301  2346  2394           3000  3000  3000  3000  3000
  2459  1564  1070  1209  2301           3000  3000  3000  3000  3000
  2346  1076  1072   968  2351           3000  3000  3000  3000  3000
  2467  1538  1073  1213  2238           3000  3000  3000  3000  3000
  2262  2336  2303  2373  2144           3000  3000  3000  3000  3000
 layer6                                 layer 6
  3335  4135  4601  4251  3794           3000  3000  3000  3000  3000
  4211  3090  2503  2881  4119           3000  3000  3000  3000  3000
  4662  2168  2239  2035  4415           3000  3000  3000  3000  3000
  4285  3067  2517  2932  3984           3000  3000  3000  3000  3000
  3630  4202  4625  4281  3305           3000  3000  3000  3000  3000
 layer7                                 layer 7
 14505 11338  9014 11118 15182          30000 30000 30000 30000 30000
 12526  3787  3138  3708 10802          30000 30000 30000 30000 30000
 10454  3081  2800  3009  7924          30000 30000 30000 30000 30000
 12516  3791  3141  3718 10860          30000 30000 30000 30000 30000
 14621 11379  9166 11250 14952          30000 30000 30000 30000 30000
 layer8                                 layer 8
    87    84    82    83    86            100   100   100   100   100
    84    85    84    84    83            100   100   100   100   100
    82    84    84    84    82            100   100   100   100   100
    84    85    84    84    83            100   100   100   100   100
    87    84    82    83    86            100   100   100   100   100
```

Figure 11. The resulting model after 15 iterations of the inversion procedure when we constrained the bottom four layers to be 1-D. For this run, we used three relaxation steps per inversion iteration. We show the actual model on the right for comparison.

```
Number of relaxation steps = 10                 Rmmfac=3.0e-5
Number of inversion iterations = 15             Damping=1.0
Rmm constraints to keep layers 5,6,7,8 1D

RESULTS FROM INVERSION PROGRAM                      ACTUAL MODEL
==============================          ============================

 layer1                                 layer 1
   240   233   236   235   224            200   200   200   200   200
   239   269   258   266   226            200   200   200   200   200
   239   250   240   243   226            200   200   200   200   200
   244   267   255   264   225            200   200   200   200   200
   235   229   234   232   233            200   200   200   200   200
 layer2                                 layer 2
   340   335   352   344   344            400   400   400   400   400
   295   228   255   221   386            400   400   400   400   400
   306   258   241   286   400            400   400   400   400   400
   312   228   257   228   367            400   400   400   400   400
   337   332   353   343   349            400   400   400   400   400
 layer3                                 layer 3
   541   559   514   517   565            600   600   600   600   600
   577    12    12    11   527            600    10    10    10   600
   591    11    13    12   481            600    10    10    10   600
   587    12    11    13   505            600    10    10    10   600
   538   551   507   519   559            600   600   600   600   600
 layer4                                 layer 4
  1123  1272  1144  1117  1186           1000  1000  1000  1000  1000
  1328   164    10    10  1067           1000  1000    10    10  1000
  1295   140    13    12  1181           1000  1000    10    10  1000
  1333   168     9    10  1043           1000  1000    10    10  1000
  1109  1299  1164  1139  1181           1000  1000  1000  1000  1000
 layer5                                 layer 5
  2407  2561  2536  2619  2480           3000  3000  3000  3000  3000
  2625  2385  2223  1962  2551           3000  3000  3000  3000  3000
  2574  2204  2162  2074  2624           3000  3000  3000  3000  3000
  2672  2297  2219  1880  2611           3000  3000  3000  3000  3000
  2373  2611  2614  2565  2468           3000  3000  3000  3000  3000
 layer6                                 layer 6
  3276  3506  3693  3731  3276           3000  3000  3000  3000  3000
  3585  3534  3475  3099  3480           3000  3000  3000  3000  3000
  3625  3328  3356  3197  3688           3000  3000  3000  3000  3000
  3561  3429  3467  2968  3664           3000  3000  3000  3000  3000
  3266  3574  3756  3644  3367           3000  3000  3000  3000  3000
 layer7                                 layer 7
 17528 14841 13532 14859 17793          30000 30000 30000 30000 30000
 14980 12371 11607 12237 14863          30000 30000 30000 30000 30000
 13575 11586 11351 11538 13395          30000 30000 30000 30000 30000
 15017 12317 11592 12173 14874          30000 30000 30000 30000 30000
 17501 14822 13523 14787 17896          30000 30000 30000 30000 30000
 layer8                                 layer 8
    96    94    94    94    97            100   100   100   100   100
    94    93    93    93    95            100   100   100   100   100
    94    93    93    93    94            100   100   100   100   100
    94    93    93    93    95            100   100   100   100   100
    97    94    94    94    97            100   100   100   100   100
```

Figure 12. The resulting model after 15 iterations of the inversion procedure when we constrained the bottom four layers to be 1-D. For this run, we used 10 relaxation steps per inversion iteration. We show the actual model on the right for comparison.

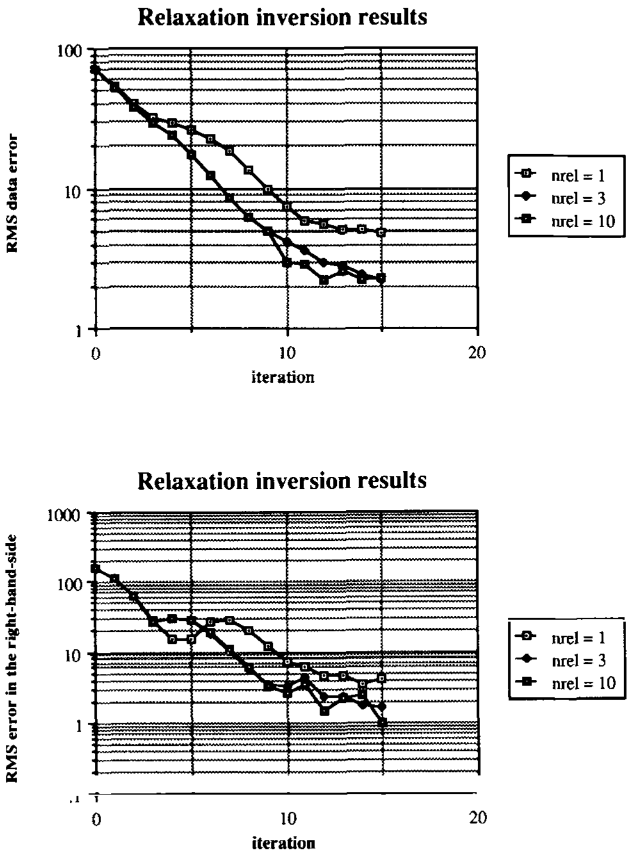

Figure 13. The RMS error progression as a function of the inversion iteration step for three different runs of the inversion algorithm. One run used one relaxation step per inversion iteration, one run used three relaxation steps per inversion iteration, and one run used 10 relaxation steps per inversion iteration. For each run, we constrained the bottom four layers of the model to be 1-D. We defined the data error and right-hand-side errors in the text, eqs (19) and (20).

is available. Fig. 13 shows the progression of the data errors and right-hand-side errors as a function of inversion iteration for the cases just described. Even though the inversions with three relaxation steps and 10 relaxation steps per inversion iteration wound up at about the same data error level, the one with 10 relaxation steps clearly did a better job at imaging the original model and adhering to the model covariance constraints.

We attempted to implement preconditioning of the relaxation scheme for the 3-D inversion as we did for the 2-D inversion (Madden & Mackie 1989). As in that case, we tried using the inverses of the 1-D sensitivity analyses for the vertical strip of blocks beneath each data site. Implementing this type of preconditioning in the 3-D case caused numerical instabilities that made the inversion diverge away from the correct solution. We do not know exactly what caused this behaviour, although we will hazard a guess. The 1-D sensitivities are more closely related to the actual 2-D sensitivities rather than the 3-D sensitivities because in both the 1-D and 2-D cases, there are sensitivity terms for the phase and resistivity, whereas in the 3-D case, there are sensitivity terms for amplitudes, phases, and directions. Furthermore, in the 3-D case, there are two electromagnetic modes that are coupled to each other. This coupling makes the 3-D inversion inherently more difficult, but may also be the reason for the numerical instabilities when we use preconditioning. This is because the preconditioning may be trying to drive the two separate electromagnetic modes in opposite directions, and this may cause the divergence we observed. Of course, the two modes increase the amount of data one has at each data site, and this fact alone should improve the convergence properties of the inversion so that the preconditioning may not be absolutely necessary for 3-D data and 3-D models. We saw in the examples presented earlier that we obtained extremely good fits to the data with just a few relaxation steps per inversion iteration without any sort of preconditioning.

CONCLUSIONS

In this paper, we discussed a method to invert magnetotelluric data for 3-D earth models. This method uses conjugate gradient relaxation to solve the maximum-likelihood inversion equations. Since at each iteration of the inversion we update the model and begin the procedure anew, we need only use a few relaxation steps at each step of the inversion. Because we are using relaxation methods, we do not need to explicitly construct and store the sensitivity matrix; rather, we only need to known the effect of the sensitivity matrix or its transpose multiplying an arbitrary vector. Each of these is equivalent to solving a forward problem with a given set of sources. This results in tremendous time savings over more traditional approaches, and makes 3-D inversions much more practical. We have demonstrated that the procedure works well for simple 3-D models.

ACKNOWLEDGMENTS

This work was supported in part by an industrial consortium of oil companies: Amoco, Chevron, Mobil, Standard Oil Production Co., Sun and Texaco. Comments from an anonymous reviewer helped improved the manuscript.

REFERENCES

Aki, K. & Richards, P. G., 1980. *Quantitative Seismology,* W. H. Freeman and Co., San Francisco.

Eggers, D. E., 1982. An eigenstate formulation of the magnetotelluric impedance tensor, *Geophysics,* **47,** 1204–1214.

Franklin, J. N., 1970. Well-posed stochastic extensions of ill-posed linear problems, *J. math. Anal. Appl.,* **31,** 682–716.

Hestenes, M. R. & Stiefel, E., 1952. Methods of conjugate gradients for solving linear systems, *J. Res. Nat. Bureau Stand.,* **49,** 409–436.

Jackson, J. D., 1975. *Classical Electrodynamics,* John Wiley & Sons, New New York.

Kong, J. A., 1986. *Electromagnetic Wave Theory,* John Wiley & Sons, New York.

Lanczos, C., 1961. *Linear Differential Operators,* D. Van Nostrand Co. Ltd, London.

LaTorraca, G. A., Madden, T. R. & Korringa, J., 1986. An analysis of the magnetotelluric impedance for three-dimensional conductivity structures, *Geophysics,* **51,** 1819–1829.

Mackie, R. L., 1991. Three-dimensional magnetotelluric modeling and inversion with applications to the California Basin and Range Province, *PhD dissertation,* MIT.

Mackie, R. L. & Madden, T. R., 1993. Conjugate gradient relaxation solutions for three-dimensional magnetotelluric modeling, *Geophysics*, **58**, 1052–1057.

Mackie, R. L., Bennett, B. R. & Madden, T. R., 1988. Long-period magnetotelluric measurements near the central California coast: a land-locked view of the conductivity structure under the Pacific Ocean, *Geophys. J.*, **95**, 181–194.

Mackie, R. L., Madden, T. R. & Wannamaker, P. E., 1993. Three-dimensional magnetotelluric modeling using difference equations—theory and comparisons to integral equation solutions, *Geophysics*, **58**, 215–226

Madden, T. R., 1990. Inversion of low-frequency electromagnetic data, in *Oceanographic and Geophysical Tomography*, pp. 377–408, Elsevier Science Publishers Amsterdam.

Madden, T. R. & Mackie, R. L., 1989. Three-dimensional magnetotelluric modeling and inversion, *Proc. IEEE*, **77**, 318–333.

Mora, P. R., 1987. Nonlinear two-dimensional elastic inversion of multioffset seismic data, *Geophysics*, **52**, 1211–1228.

Smith, J. T. & Booker, J. R., 1991. Rapid inversion of two- and three-dimensional magnetotelluric data, *J. geophys. Res.*, **96**, 3905–3922.

Tarantola, A., 1987. *Inverse Problem Theory: Method for Data fitting and Model Parameter Estimation*, Elsevier, New York.

Tarantola, A. & Valette, B., 1982. Generalized nonlinear inverse problems solved using the least squares criterion, *Rev. Geophys. Space Phys.*, **20**, 219–232.

APPENDIX 1: DERIVATION OF SENSITIVITY TERMS

We decompose the impedance tensor into two basis vectors given by $\mathbf{E}_{H_x} = [\hat{\mathbf{x}}Z_{xx} + \hat{\mathbf{y}}Z_{yx}]$ and $\mathbf{E}_{H_y} = [\hat{\mathbf{x}}Z_{xy} + \hat{\mathbf{y}}Z_{yy}]$. Any complex vector in the frequency domain, $\mathbf{E}(\omega) = \mathbf{a} + j\mathbf{b}$, can be expressed in the time domain, assuming an $e^{-i\omega t}$ dependency, as $\mathbf{e}(t) = \mathscr{Re}\,[\mathbf{E}(\omega)e^{-i\omega t}] = \mathbf{a}\cos\omega t + \mathbf{b}\sin\omega t$. The magnitude of the complex vector is simply $[\mathbf{E}^*\mathbf{E}]^{1/2}$. In the time domain, the vector traces our an ellipse, which in some cases degenerates to a circle. The major axis of the ellipse defines the direction corresponding to that vector field, and the phase is defined as the phase along that axis. The ellipticity is the ratio of the major axis to the minor axis. Ellipticities are difficult to determine accurately from field data, but magnitudes, phases, and directions are more robust in comparison. Consequently, we invert only for amplitudes, phases, and directions. Following Eggers (1982), we can write down the expressions for the magnitudes, phases (ϕ), and directions (ψ) in terms of our vector basis set:

$$|\mathbf{E}_{H_x}| = [\mathrm{Z}_{\mathrm{xx}}\mathrm{Z}^*_{\mathrm{xx}} + \mathrm{Z}_{\mathrm{yx}}\mathrm{Z}^*_{\mathrm{yx}}]^{1/2}$$

$$|\mathbf{E}_{H_y}| = [Z_{xy}Z^*_{xy} + Z_{yy}Z^*_{yy}]^{1/2}$$

$$\phi_{H_x} = \tfrac{1}{2}\tan^{-1}\frac{2(Z^{\mathrm{R}}_{xx}Z^{\mathrm{I}}_{xx} + Z^{\mathrm{R}}_{yx}Z^{\mathrm{I}}_{yx})}{(Z^{\mathrm{R}}_{xx})^2 + (Z^{\mathrm{R}}_{yx})^2 - (Z^{\mathrm{I}}_{xx})^2 - (Z^{\mathrm{I}}_{yx})^2}$$

$$\phi_{H_y} = \tfrac{1}{2}\tan^{-1}\frac{2(Z^{\mathrm{R}}_{xy}Z^{\mathrm{I}}_{xy} + Z^{\mathrm{R}}_{yy}Z^{\mathrm{I}}_{yy})}{(Z^{\mathrm{R}}_{xy})^2 + (Z^{\mathrm{R}}_{yy})^2 - (Z^{\mathrm{I}}_{xy})^2 - (Z^{\mathrm{I}}_{yy})^2} \qquad \text{(A1)}$$

$$\psi_{H_x} = \tfrac{1}{2}\tan^{-1}\frac{2(Z^{\mathrm{R}}_{xx}Z^{\mathrm{R}}_{yx} + Z^{\mathrm{I}}_{xx}Z^{\mathrm{I}}_{yx})}{(Z^{\mathrm{R}}_{xx})^2 + (Z^{\mathrm{I}}_{xx})^2 - (Z^{\mathrm{R}}_{yx})^2 - (Z^{\mathrm{i}}_{yx})^2}$$

$$\psi^{I}_{H_y} = \tfrac{1}{2}\tan^{-1}\frac{2(Z^{\mathrm{R}}_{xy}Z^{\mathrm{R}}_{yy} + Z^{\mathrm{I}}_{xy}Z^{\mathrm{I}}_{yy})}{(Z^{\mathrm{R}}_{xy})^2 + (Z^{\mathrm{I}}_{xy})^2 - (Z^{\mathrm{R}}_{yy})^2 - (Z^{\mathrm{I}}_{yy})^2}$$

where the terms like Z^{R}_{xx} stand for the real part of Z_{xx}, and so on. We derive the sensitivity terms by algebraically differentiating the above expressions. We first need the partial derivatives of the components of the impedance tensor, which are

$$\begin{aligned}
\frac{\partial Z_{xx}}{\partial\sigma} &= \frac{1}{\det HH}\left[E_{x1}\frac{\partial H_{y2}}{\partial\sigma} + H_{y2}\frac{\partial E_{x1}}{\partial\sigma} - E_{x2}\frac{\partial H_{y1}}{\partial\sigma} - H_{y1}\frac{\partial E_{x2}}{\partial\sigma} - Z_{xx}\,\partial HH\right]\\
\frac{\partial Z_{xy}}{\partial\sigma} &= \frac{1}{\det HH}\left[E_{x2}\frac{\partial H_{x1}}{\partial\sigma} + H_{x1}\frac{\partial E_{x2}}{\partial\sigma} - E_{x1}\frac{\partial H_{x2}}{\partial\sigma} - H_{x2}\frac{\partial E_{x1}}{\partial\sigma} - Z_{xy}\,\partial HH\right]\\
\frac{\partial Z_{yx}}{\partial\sigma} &= \frac{1}{\det HH}\left[E_{y1}\frac{\partial H_{y2}}{\partial\sigma} + H_{y2}\frac{\partial E_{y1}}{\partial\sigma} - E_{y2}\frac{\partial H_{y1}}{\partial\sigma} - H_{y1}\frac{\partial E_{y2}}{\partial\sigma} - Z_{yx}\,\partial HH\right]\\
\frac{\partial Z_{yy}}{\partial\sigma} &= \frac{1}{\det HH}\left[E_{y2}\frac{\partial H_{x1}}{\partial\sigma} + H_{x1}\frac{\partial E_{y2}}{\partial\sigma} - E_{y1}\frac{\partial H_{x2}}{\partial\sigma} - H_{x2}\frac{\partial E_{y1}}{\partial\sigma} - Z_{yy}\,\partial HH\right]
\end{aligned} \qquad \text{(A2)}$$

where we have made the following definitions:

$$\begin{aligned}
\det HH &= H_{x1}H_{y2} - H_{x2}H_{y1}\\
\partial HH &= H_{x1}\frac{\partial H_{y2}}{\partial\sigma} + H_{y2}\frac{\partial H_{x1}}{\partial\sigma} - H_{x2}\frac{\partial H_{y1}}{\partial\sigma} - H_{y1}\frac{\partial H_{x2}}{\partial\sigma}.
\end{aligned} \qquad \text{(A3)}$$

With these definitions, one finds the sensitivity terms for the magnitudes are

$$\begin{aligned}
\frac{\partial\,|\mathbf{E}_{H_x}|}{\partial\sigma} &= \frac{1}{|\mathbf{E}_{H_x}|}\left[\mathscr{Re}\left(Z^*_{xx}\frac{\partial Z_{xx}}{\partial\sigma}\right) + \mathscr{Re}\left(Z^*_{yx}\frac{\partial Z_{yx}}{\partial\sigma}\right)\right]\\
\frac{\partial\,|\mathbf{E}_{H_y}|}{\partial\sigma} &= \frac{1}{|\mathbf{E}_{H_y}|}\left[\mathscr{Re}\left(Z^*_{xy}\frac{\partial Z_{xy}}{\partial\sigma}\right) + \mathscr{Re}\left(Z^*_{yy}\frac{\partial Z_{yy}}{\partial\sigma}\right)\right]
\end{aligned} \qquad \text{(A4)}$$

where we have used relationships such as

$$Z_{xx}\frac{\partial Z^*_{xx}}{\partial\sigma} + Z^*_{xx}\frac{\partial Z_{xx}}{\partial\sigma} = 2\,\mathscr{Re}\left[Z^*_{xx}\frac{\partial Z_{xx}}{\partial\sigma}\right] \qquad \text{(A5)}$$

in simplifying the expressions for the derivatives. The expressions for the derivatives of the phase terms are a bit more complicated, and are

$$\begin{aligned}
\frac{\partial\phi_{H_x}}{\partial\sigma} = \frac{1}{1+u^2}\Bigg[&\left(\frac{Z^{\mathrm{I}}_{xx}}{u_{\mathrm{d}}} - \frac{2Z^{\mathrm{R}}_{xx}u_{\mathrm{n}}}{u^2_{\mathrm{d}}}\right)\frac{\partial Z^{\mathrm{R}}_{xx}}{\partial\sigma}\\
&+ \left(\frac{Z^{\mathrm{I}}_{yx}}{u_{\mathrm{d}}} - \frac{2Z^{\mathrm{R}}_{yx}u_{\mathrm{n}}}{u^2_{\mathrm{d}}}\right)\frac{\partial Z^{\mathrm{R}}_{yx}}{\partial\sigma}\\
&+ \left(\frac{Z^{\mathrm{R}}_{xx}}{u_{\mathrm{d}}} - \frac{2Z^{\mathrm{I}}_{xx}u_{\mathrm{n}}}{u^2_{\mathrm{n}}}\right)\frac{\partial Z^{\mathrm{I}}_{xx}}{\partial\sigma}\\
&+ \left(\frac{Z^{\mathrm{R}}_{yx}}{u_{\mathrm{d}}} - \frac{2Z^{\mathrm{I}}_{yx}u_{\mathrm{n}}}{u^2_{\mathrm{d}}}\right)\frac{\partial Z^{\mathrm{I}}_{yx}}{\partial\sigma}\Bigg]
\end{aligned}$$

$$\begin{aligned}\frac{\partial \phi_{H_y}}{\partial \sigma} = \frac{1}{1+v^2}\bigg[&\left(\frac{Z^{\mathrm{I}}_{xy}}{v_{\mathrm{d}}} - \frac{2Z^{\mathrm{R}}_{xy}v_{\mathrm{n}}}{v^2_{\mathrm{d}}}\right)\frac{\partial Z^{\mathrm{R}}_{xy}}{\partial \sigma}\\ &+\left(\frac{Z^{\mathrm{I}}_{yy}}{v_{\mathrm{d}}} - \frac{2Z^{\mathrm{R}}_{yy}v_{\mathrm{n}}}{v^2_{\mathrm{d}}}\right)\frac{\partial Z^{\mathrm{R}}_{yy}}{\partial \sigma}\\ &+\left(\frac{Z^{\mathrm{R}}_{xy}}{v_{\mathrm{d}}} - \frac{2Z^{\mathrm{I}}_{xy}v_{\mathrm{n}}}{v^2_{\mathrm{d}}}\right)\frac{\partial Z^{\mathrm{I}}_{xy}}{\partial \sigma}\\ &+\left(\frac{Z^{\mathrm{R}}_{yy}}{v_{\mathrm{d}}} - \frac{2Z^{\mathrm{I}}_{yy}v_{\mathrm{n}}}{v^2_{\mathrm{d}}}\right)\frac{\partial Z^{\mathrm{I}}_{yy}}{\partial \sigma}\bigg]\end{aligned} \tag{A6}$$

where we have made the following definitions

$$\begin{aligned} u &= \frac{2(Z^{\mathrm{R}}_{xx}Z^{\mathrm{I}}_{xx} + Z^{\mathrm{R}}_{yx}Z^{\mathrm{I}}_{yx})}{(Z^{\mathrm{R}}_{xx})^2 + (Z^{\mathrm{R}}_{yx})^2 - (Z^{\mathrm{I}}_{xx})^2 - (Z^{\mathrm{I}}_{yx})^2}\\ v &= \frac{2(Z^{\mathrm{R}}_{xy}Z^{\mathrm{I}}_{xy} + Z^{\mathrm{R}}_{yy}Z^{\mathrm{I}}_{yy})}{(Z^{\mathrm{R}}_{xy})^2 + (Z^{\mathrm{R}}_{yy})^2 - (Z^{\mathrm{I}}_{xy})^2 - (Z^{\mathrm{I}}_{yy})^2},\end{aligned} \tag{A7}$$

with u_{d} and u_{n} representing the denominator and numerator respectively of the expression given above for u, and v_{d} and v_{n} likewise representing the denominator and numerator of the expression given above for v. The sensitivity terms for the directions of the major axes are

$$\begin{aligned}\frac{\partial \psi_{H_x}}{\partial \sigma} = \frac{1}{1+s^2}\bigg[&\left(\frac{Z^{\mathrm{R}}_{xx}}{s_{\mathrm{d}}} + \frac{2Z^{\mathrm{R}}_{yx}s_{\mathrm{n}}}{s^2_{\mathrm{d}}}\right)\frac{\partial Z^{\mathrm{R}}_{yx}}{\partial \sigma}\\ &+\left(\frac{Z^{\mathrm{R}}_{yx}}{s_{\mathrm{d}}} - \frac{2Z^{\mathrm{R}}_{xx}s_{\mathrm{n}}}{s^2_{\mathrm{d}}}\right)\frac{\partial Z^{\mathrm{R}}_{xx}}{\partial \sigma}\\ &+\left(\frac{Z^{\mathrm{I}}_{xx}}{s_{\mathrm{d}}} + \frac{2Z^{\mathrm{I}}_{yx}s_{\mathrm{n}}}{s^2_{\mathrm{d}}}\right)\frac{\partial Z^{\mathrm{I}}_{yx}}{\partial \sigma}\\ &+\left(\frac{Z^{\mathrm{I}}_{yx}}{s_{\mathrm{d}}} - \frac{2Z^{\mathrm{I}}_{xx}s_{\mathrm{n}}}{s^2_{\mathrm{d}}}\right)\frac{\partial Z^{\mathrm{I}}_{xx}}{\partial \sigma}\bigg]\\ \frac{\partial \Psi_{H_y}}{\partial \sigma} = \frac{1}{1+t^2}\bigg[&\left(\frac{Z^{\mathrm{R}}_{xy}}{t_{\mathrm{d}}} + \frac{2Z^{\mathrm{R}}_{yy}t_{\mathrm{n}}}{t^2_{\mathrm{d}}}\right)\frac{\partial Z^{\mathrm{R}}_{yy}}{\partial \sigma}\\ &+\left(\frac{Z^{\mathrm{R}}_{yy}}{t_{\mathrm{d}}} - \frac{2Z^{\mathrm{R}}_{xy}t_{\mathrm{n}}}{t^2_{\mathrm{d}}}\right)\frac{\partial Z^{\mathrm{R}}_{xy}}{\partial \sigma}\\ &+\left(\frac{Z^{\mathrm{I}}_{xy}}{t_{\mathrm{d}}} + \frac{2Z^{\mathrm{I}}_{yy}y_{\mathrm{n}}}{t^2_{\mathrm{d}}}\right)\frac{\partial Z^{\mathrm{I}}_{yy}}{\partial \sigma}\\ &+\left(\frac{Z^{\mathrm{I}}_{yy}}{t_{\mathrm{d}}} - \frac{2Z^{\mathrm{I}}_{xy}t_{\mathrm{n}}}{t^2_{\mathrm{d}}}\right)\frac{\partial Z^{\mathrm{I}}_{xy}}{\partial \sigma}\bigg]\end{aligned} \tag{A8}$$

where we have made the following definitions

$$\begin{aligned} s &= \frac{2(Z^{\mathrm{R}}_{xx}Z^{\mathrm{R}}_{yx} + Z^{\mathrm{I}}_{xx}Z^{\mathrm{I}}_{yx})}{(Z^{\mathrm{R}}_{xx})^2 + (Z^{\mathrm{I}}_{xx})^2 - (Z^{\mathrm{R}}_{yx})^2 - (Z^{\mathrm{I}}_{yx})^2}\\ t &= \frac{2(Z^{\mathrm{R}}_{xy}Z^{\mathrm{R}}_{yy} + Z^{\mathrm{I}}_{xy}Z^{\mathrm{I}}_{yy})}{(Z^{\mathrm{R}}_{xy})^2 + (Z^{\mathrm{I}}_{xy})^2 - (Z^{\mathrm{R}}_{yy})^2 - (Z^{\mathrm{I}}_{yy})^2},\end{aligned} \tag{A9}$$

with s_{d} and s_{n} representing the denominator and numerator respectively of the above expression for s and t_{d} and t_{n} representing the denominator and numerator of the above expression for t.