Inverse Problems **20** (2004) S151–S170 PII: S0266-5611(04)78789-4

Solution accelerators for large-scale three-dimensional electromagnetic inverse problems

Gregory A Newman[1] **and Paul T Boggs**[2]

[1] Earth Sciences Division, Lawrence Berkeley National Laboratory, Berkeley, CA, USA
[2] Computational Sciences and Mathematics Research Department, Sandia National Laboratories, Livermore, CA, USA

Received 5 April 2004, in final form 12 July 2004
Published 8 November 2004
Online at stacks.iop.org/IP/20/S151
doi:10.1088/0266-5611/20/6/S10

Abstract

We provide a framework for preconditioning nonlinear three-dimensional electromagnetic inverse scattering problems using nonlinear conjugate gradient (NLCG) and limited memory (LM) quasi-Newton methods. Key to our approach is the use of an approximate adjoint method that allows for an economical approximation of the Hessian that is updated at each inversion iteration. Using this approximate Hessian as a preconditioner, we show that the preconditioned NLCG iteration converges significantly faster than the non-preconditioned iteration, as well as converging to a data misfit level below that observed for the non-preconditioned method. Similar conclusions are also observed for the LM iteration; preconditioned with the approximate Hessian, the LM iteration converges faster than the non-preconditioned version. At this time, however, we see little difference between the convergence performance of the preconditioned LM scheme and the preconditioned NLCG scheme. A possible reason for this outcome is the behaviour of the line search within the LM iteration. It was anticipated that, near convergence, a step size of one would be approached, but what was observed, instead, were step lengths that were nowhere near one. We provide some insights into the reasons for this behaviour and suggest further research that may improve the performance of the LM methods.

1. Introduction

Three-dimensional (3D) electromagnetic (EM) inversion has shown significant potential in hydrological and hazardous waste site characterization, as well as in oil and gas exploration. It also has important applications in mineral and geothermal exploration and general geologic mapping. 3D inversion has been used successfully to map subsurface transport pathways for contaminants [1] to delineate buried metallic waste, to define the extent of waste pits and to determine the safety of proposed long-term waste disposal sites [2]. In spite of these successes,

0266-5611/04/060151+20$30.00

3D inversion of data continues to be a cumbersome and complex process requiring significant time and computational resources [3–6], thus restricting its use.

Owing to the importance of the approach, we continue to investigate a variety of 3D inversion schemes for improved efficiency, emphasizing gradient-type techniques because they can effectively treat large data volumes that typically arise in 3D EM imaging experiments [7]. Such schemes include steepest decent, nonlinear conjugate gradients (NLCG) and limited memory (LM) quasi-Newton methods. To improve the efficiency of these methods, preconditioning can be applied with the goal of inducing these methods to perform more like Newton's method, which is considered to be the most robust nonlinear optimization scheme for large-scale inverse problems. Efficiency is paramount, because in this nonlinear inverse problem, hundreds to thousands of 3D forward-modelling solutions are required to fine-tune regularization parameters, quantify data noise and access the unique properties of the solution through an appraisal process [8].

Two properties are required of a good preconditioner. The first is to reduce the number of inversion iterations required for convergence, and the second is to do so in such a manner that the work required in the application of the preconditioner is small. A preconditioner that achieves the first goal, but is expensive to apply in terms of computational requirements, is neither effective nor desirable. Newman and Alumbaugh [7] explored the use of a diagonal preconditioner for the magnetotellurics based on a quasi-Newton formula, known as the Broyden–Fletcher–Goldfarb–Shanno (BFGS) update [9]. The advantage of this preconditioner is that it can be developed directly from the nonlinear conjugate gradient iteration and is continually updated within the inversion process. Its disadvantage is its erratic performance. In the best case, it offers a modest reduction in the number of inversion iterations compared with an inverse solution employing no preconditioning.

For the 2D magnetotelluric problem, Rodi and Mackie [10] introduced a preconditioner for the nonlinear conjugate gradient iteration that attempts to economically approximate the Hessian using the component of the Hessian that has been introduced to stabilize, or regularize, the inverse problem. In addition to this term, they add a scaled identity matrix, based on the size of the data component part of the Hessian, which is assumed to arise from a simplified homogeneous medium that is fixed during the inversion iteration. Since the approximate Hessian is highly sparse, it can be efficiently inverted to obtain a preconditioned descent/search direction using a linear conjugate gradient solver; hence its potential effectiveness as a preconditioner. Using this approach as a guide, we will seek to develop a more robust preconditioner using approximate data sensitivities based on the ideas of Farquharson and Oldenburg [11]. At each inversion iteration the idea is to more accurately estimate the data component of the Hessian so it can be applied as a preconditioner in both the NLCG and LM quasi-Newton schemes, where efficiency is paramount for 3D inverse problems. These later schemes have the potential to perform even better than NLCG schemes. These methods can be seen as extensions of the conjugate gradient method, in which additional storage is used to accelerate convergence. Unfortunately, in large-scale inverse problems, the quasi-Newton methods may not be practical due to excessive storage requirements. This problem can be avoided by implementing a limited memory variant of the scheme first proposed by Perry [12] and subsequently improved in the intervening years by Nocedal [13] among others. (See [14] for a good introduction.)

To set the stage for developing the preconditioner, we will first review the inverse and forward problem formulations, and gradient-type minimization methodologies. Following these reviews, we show how the preconditioner is implemented and will then demonstrate its effectiveness on a 3D cross well EM imaging experiment for the two different inversion schemes discussed above.

2. Inverse problem formulation

Following Newman and Hoversten, [15], we formulate the inverse problem by the minimization of the following error functional:

$$\phi = \tfrac{1}{2}\{(\mathbf{D}(\mathbf{d}^{p} - \mathbf{d}^{obs})\}^{\mathrm{T}*}\{\mathbf{D}(\mathbf{d}^{p} - \mathbf{d}^{obs})\}, \qquad (1)$$

where T^{*} denotes the transpose-conjugation operator. In the above expression, the predicted and observed data vectors are denoted by $\mathbf{d}^{p}$ and $\mathbf{d}^{obs}$, respectively, where each has n complex values. These vectors consist of electric or magnetic field values specified at the measurement points, where the predicted data are determined through solution of the forward modelling problem, discussed below. We have also introduced a diagonal weighting matrix, $\mathbf{D}$, into the error functional to compensate for noisy measurements; it is typically based on the inverse of the standard deviations of the measurements.

To stabilize the minimization of equation (1) we are required to add a regularization term. Many choices are available; in our earlier work we have focused on a class of conductivity models using Tikhonov regularization that exhibit smoothly varying properties. Thus, we introduce a matrix $\mathbf{W}$, based upon a finite-difference approximation to the Laplacian (∇^2) operator applied in Cartesian coordinates, to reduce model curvature in three dimensions. Further, we will divide the inversion domain into m cells and assign to each cell an unknown conductivity value. These quantities are real-valued and collectively stored in the model vector $\mathbf{m}$, which is piecewise constant. Hence, equation (1) is augmented as

$$\phi(\mathbf{m}) = \tfrac{1}{2}\{\mathbf{D}(\mathbf{d}^{obs} - \mathbf{d}^{p})\}^{\mathrm{T}*}\{\mathbf{D}(\mathbf{d}^{obs} - \mathbf{d}^{p})\} + \tfrac{1}{2}\lambda\{\mathbf{W}\mathbf{m}\}^{\mathrm{T}}\{\mathbf{W}\mathbf{m}\}, \qquad (2)$$

where the parameter λ attempts to balance data error and model smoothness, and T denotes the transpose operator. Note that $\mathbf{d}$ is now a function of $\mathbf{m}$, since these parameters will enter into the forward modelling problem. The common recipe for selecting λ is based upon a cooling approach (cf [16]); one carries out multiple solutions to the inverse problem starting with a large fixed value for λ. As λ is reduced, the data error, represented by the first term in equation (2), will decrease. We continue this process of reducing λ, until the data error agrees with a target misfit based upon the assumed noise content of the data.

3. Forward problem formulation

The error functional gradient, $\nabla\phi$, and the predicted data are linked directly to the forward modelling problem that is described by the time-harmonic Maxwell equations in the diffusive approximation (i.e. without displacement currents and assuming an $\mathrm{e}^{\mathrm{i}\omega t}$ time dependence),

$$\sigma\mathbf{E} - \nabla\times\mathbf{H} = -\mathbf{J}, \qquad \nabla\times\mathbf{E} + \mathrm{i}\omega\mu_o\mathbf{H} = -\mathbf{M}, \qquad (3)$$

where ω is the angular frequency and $\mathrm{i} = \sqrt{-1}$. Applied currents generate the electric and magnetic fields, $\mathbf{E}$ and $\mathbf{H}$, and are denoted by $\mathbf{J}$ for the electric source and $\mathbf{M}$ for the magnetic source. The Earth's electrical conductivity, σ, is a function of position that is allowed to vary three dimensionally. On the other hand, we set the magnetic permeability, μ, to its free-space value, μ_o. Variations in the magnetic permeability are rare, confined to magnetic ores and some volcanic soils.

Our solution method for the forward modelling problem is based upon the consideration that the number of model parameters required to simulate realistic 3D geology can typically exceed 10^7. Finite-difference modelling schemes are ideally suited for this task and can be parallelized to handle large-scale problems that cannot be easily treated otherwise [17]. After approximating the Maxwell equations on a staggered grid at a specific angular frequency using finite differencing and eliminating the magnetic field (see [17] for specific details), we obtain a linear system for the electric field,

$$\mathbf{KE} = \mathbf{S}, \tag{4}$$

where $\mathbf{K}$ is a sparse complex symmetric matrix of dimension $N \times N$, with 13 non-zero entries per row and its diagonal entries depend explicitly on the electrical conductivity, σ, which we desire to estimate through the inversion process. Since the electric field, $\mathbf{E}$, also depends upon the conductivity, implicitly, this gives rise to the nonlinearity of the inverse problem. The fields are sourced with a grounded wire or loop embedded within the modelling domain, described by the discrete source vector $\mathbf{S}$, and includes Dirichlet boundary conditions imposed on the problem. To avoid excessive meshing near the source, we favour a scattered-field formulation of the forward modelling problem. In this instance, $\mathbf{E}$ is replaced by $\mathbf{E}_s$ in equation (4). The source term, for a given transmitter, will now depend upon the difference between the 3D conductivity model and a simple background model, weighted by the background electric field, $\mathbf{E}_b$, where $\mathbf{E} = \mathbf{E}_b + \mathbf{E}_s$. We favour simple background models, such as whole space or layered half-space models that can be easily and rapidly simulated. Equation (4) is solved to a predetermined error level using iterative Krylov subspace methods. With the solution of the electric field, the magnetic field can be easily determined from a numerical implementation of Faraday's law

$$\mathbf{H} = \nabla \times \mathbf{E}/(-\mathrm{i}\omega\mu_o). \tag{5}$$

4. Gradient-type minimization methodologies

For large-scale nonlinear problems, as considered here, we shall minimize equation (2) using gradient-based optimization techniques, including NLCG and LM quasi-Newton schemes, because of their minimal storage requirements. We characterize these methods as gradient-based techniques because they employ only first-derivative information of the error functional in the minimization process. We first show how the gradients can be efficiently computed. Then we briefly describe the NLCG method that we have been using. Next, we discuss quasi-Newton methods and their implementation in this context. Finally, we introduce an approximate Hessian matrix based on an approximate data-sensitivity concept. This will be used as a preconditioner for the NLCG method and will be incorporated into the quasi-Newton inversion iteration as well, with the goal of improving the performance of both methods.

4.1. Gradient derivations

Gradient derivations for our problem can be found in Newman and Hoversten [15], where they are used to develop Newton/Gauss–Newton solution methodologies for the multi-dimensional electromagnetic inverse problem. Since we will investigate gradient-type solution methodologies, we shall re-derive gradients to make our paper self-contained.

We formally express the gradient of the error functional in equation (2) as

$$\boldsymbol{\nabla}\phi = \boldsymbol{\nabla}\phi_d + \lambda\boldsymbol{\nabla}\phi_m, \tag{6}$$

where $\boldsymbol{\nabla}\phi$ is split into the data-misfit and regularization terms appearing in equation (2). The model parametrization employed in the solution of the forward problem will also be used in the inverse solution. Hence the kth component of the gradient will be based upon the conductivity of the kth cell in the finite-difference mesh. Evaluation of $\boldsymbol{\nabla}\phi_m$ is straightforward:

$$\boldsymbol{\nabla}\phi_m = \mathbf{W}^{\mathrm{T}}\mathbf{W}\mathbf{m}. \tag{7}$$

Evaluation of $\nabla\phi_d$ is more complicated, but using the chain rule with some algebra it can be shown that

$$\nabla\phi_d = -\mathrm{Re}(\{\mathbf{DJ}\}^{\mathrm{T}}\{\mathbf{D}(\mathbf{d}^{obs} - \mathbf{d}^{p})\}^{*}), \tag{8}$$

where $*$ stands for complex conjugation, Re specifies the real part of its argument and $\mathbf{J}$ is the data-sensitivity matrix. To specify $\mathbf{J}$, consider a matrix $\mathbf{Q}$ of dimension $n \times N$, which operates on the electric field, $\mathbf{E}$, determined by the solution of equation (4). The operator $\mathbf{Q}$ interpolates the predicted electric field from the mesh to the receiver locations. It can also compute the discrete curl of the electric field (equation (5)) and interpolate predicted magnetic field results to specified receiver locations. Thus, the predicted data can be expressed as

$$\mathbf{d}^{p} = \mathbf{QE}. \tag{9}$$

With equation (9) the data sensitivity for the kth model parameter can be expressed as

$$\partial\mathbf{d}^{p}/\partial m_k = \mathbf{Q}\partial\mathbf{E}/\partial m_k \tag{10}$$

or

$$\partial\mathbf{d}^{p}/\partial m_k = \mathbf{Q}\partial\mathbf{E}_s/\partial m_k, \tag{11}$$

because the background electric field, $\mathbf{E}_b$, is fixed. Given a scattered field formulation of the forward modelling problem, we use equation (4) to express $\partial\mathbf{E}_s/\partial m_k$ as

$$\partial\mathbf{E}_s/\partial m_k = -\mathbf{K}^{-1}(\partial\mathbf{S}/\partial m_k - \partial\mathbf{K}/\partial m_k\mathbf{E}_s), \tag{12}$$

and the kth column of the data-sensitivity matrix/Jacobian matrix is written as

$$\mathbf{J}_k = \mathbf{QK}^{-1}(\partial\mathbf{S}/\partial m_k - \partial\mathbf{K}/\partial m_k\mathbf{E}_s). \tag{13}$$

Let us define an $N \times m$ matrix $\mathbf{G}$, where

$$\mathbf{G} = \{(\partial\mathbf{S}/\partial m_1 - \partial\mathbf{K}/\partial m_1\mathbf{E}_s), (\partial\mathbf{S}/\partial m_2 - \partial\mathbf{K}/\partial m_2\mathbf{E}_s), \ldots, (\partial\mathbf{S}/\partial m_m - \partial\mathbf{K}/\partial m_m\mathbf{E}_s)\}. \tag{14}$$

Hence, the Jacobian matrix is expressed using a triple matrix product,

$$\mathbf{J} = \mathbf{QK}^{-1}\mathbf{G}. \tag{15}$$

Moreover, since $\mathbf{K}^{-\mathrm{T}} = \mathbf{K}^{-1}$ ($\mathbf{K}$ is complex-symmetric), we also have

$$\mathbf{J}^{\mathrm{T}} = \mathbf{G}^{\mathrm{T}}\mathbf{K}^{-1}\mathbf{Q}^{\mathrm{T}}. \tag{16}$$

Substituting equation (16) into equation (8) and noting $\{\mathbf{DJ}\}^{\mathrm{T}} = \mathbf{J}^{\mathrm{T}}\mathbf{D}^{\mathrm{T}}$, we express the data component of the gradient in terms of the forward modelling problem:

$$\nabla\phi_d = -\mathrm{Re}(\mathbf{G}^{\mathrm{T}}\mathbf{K}^{-1}\mathbf{Q}^{\mathrm{T}}\mathbf{D}^{\mathrm{T}}\{\mathbf{D}(\mathbf{d}^{obs} - \mathbf{QE})\}^{*}). \tag{17}$$

We can now show that the number of forward modelling solutions needed to evaluate the gradient is two. The first is needed to specify the electric field, $\mathbf{E}$, at the current model at a fixed frequency. The additional solution arises from the source distribution,

$$\mathbf{v} = \mathbf{Q}^{\mathrm{T}}\mathbf{D}^{\mathrm{T}}\{\mathbf{D}(\mathbf{d}^{obs} - \mathbf{QE})\}^{*}, \tag{18}$$

and is determined by solving the linear system

$$\mathbf{Ku} = \mathbf{v}, \tag{19}$$

to a predetermined error level.

To conclude this section we remark that specification of the gradient, $\nabla\phi_d$, for multiple sources is simply a sum of gradients (equation (17)) due to each source at a discrete frequency, as applied in equation (4).

4.2. Nonlinear conjugate gradient iteration

Shown below is a pseudo-code of the preconditioned NLCG algorithm that could be used in the minimization of equation (2).

Algorithm

```
(1) set $i = 1$, choose initial model $\mathbf{m}_i$ and compute $\mathbf{r}_i = -\nabla\phi(\mathbf{m}_i)$
(2) set $\mathbf{u}_i = \mathbf{M}_i^{-1}\mathbf{r}_i$
(3) find $\alpha_i$ that minimizes $\phi(\mathbf{m}_i + \alpha_i\mathbf{u}_i)$
(4) set $\mathbf{m}_{i+1} = \mathbf{m}_i + \alpha_i\mathbf{u}_i$ and compute $\mathbf{r}_{i+1} = -\nabla\phi(\mathbf{m}_{i+1})$
(5) stop when $|\mathbf{r}_{i+1}| < \varepsilon$, otherwise go to step (6)
(6) set $\beta_{i+1} = (\mathbf{r}_{i+1}^{\mathrm{T}}\mathbf{M}_{i+1}^{-1}\mathbf{r}_{i+1} - \mathbf{r}_i^{\mathrm{T}}\mathbf{M}_i^{-1}\mathbf{r}_i)/\mathbf{r}_i^{\mathrm{T}}\mathbf{M}_i^{-1}\mathbf{r}_i$
(7) set $\mathbf{u}_{i+1} = \mathbf{M}_{i+1}^{-1}\mathbf{r}_{i+1} + \beta_{i+1}\mathbf{u}$
(8) set $i = i + 1$ and go to step 3
```

The matrix operator $\mathbf{M}_i^{-1}$ in the algorithm is a preconditioner, which steers and scales the conjugate search direction $\mathbf{u}_i$ such that it more closely approximates the Newton direction. For now let us assume the preconditioner is simply the identity matrix. To use the NLCG algorithm sensibly requires efficient computation of the gradient, which we have already demonstrated. Most implementations of NLCG include a very accurate line search. Indeed, when applied to quadratic minimizations, an *exact* line search is necessary to ensure conjugacy. Since the evaluation of the function in our case is very expensive, we developed a procedure that gives an acceptable decrease of the function with a minimal number of evaluations. Newman and Alumbaugh [7] discuss the issue and show it is possible to achieve acceptable decreases in the error function using a line search based upon quadratic interpolation, safeguarded with back-tracking. Usually, an additional forward modelling application per source is all that is needed for the line search, thereby yielding three forward modelling applications per source per inversion iteration for the algorithm.

4.3. Background on LM quasi-Newton methods

It is well known that methods that only use first-order gradient information, for example, steepest descent, are often very slowly convergent in practice. To increase the speed of convergence, one would ideally like to use Newton's method, which is given by

$$\mathbf{m}_{i+1} = \mathbf{m}_i - \alpha_i(\mathbf{H}\phi_i)^{-1}\nabla\phi_i, \tag{20}$$

where $\mathbf{H}\phi_i$ is the (Hessian) matrix of second partial derivatives of ϕ with respect to $\mathbf{m}$, α_i is the steplength and the subscript i means that the function is evaluated at point $\mathbf{m}_i$. It is well known that Newton's method converges much faster than steepest descent. In fact, steepest descent is known to converge linearly with the constant of linearity close to one for ill-conditioned problems. Our problems are, of course, very ill-conditioned. The NLCG method discussed above also converges linearly, but its constant is often much smaller; hence reasonable performance has been achieved.

Newton's method, on the other hand, converges quadratically, which means, roughly, that near the solution (and with $\alpha_i = 1$) the number of significant digits doubles at each iteration. The problem with Newton's method, however, is that it is generally too expensive to compute and/or store the Hessian. Much research in optimization has been devoted to finding approximations to the Hessian that are inexpensive to compute, but effective in reducing the number of iterations required for solving the problem. See Nocedal and Wright [14] for a modern introduction to these concepts.

Quasi-Newton methods seek to approximate the Hessian by using differences in the gradients and differences in the iterates. In particular, these methods can be considered as a generalization of the secant method. Suppose we have an approximation, $\mathbf{H}_i$, to the Hessian at the current iterate. We use that to compute the next iterate,

$$\mathbf{m}_{i+1} = \mathbf{m}_i - \alpha_i \mathbf{H}_i^{-1} \nabla \phi_i. \tag{21}$$

Then by defining

$$\mathbf{s}_{i+1} = \mathbf{m}_{i+1} - \mathbf{m}_i, \tag{22}$$

$$\mathbf{y}_{i+1} = \nabla \phi_{i+1} - \nabla \phi_i, \tag{23}$$

quasi-Newton methods choose the next Hessian approximation to satisfy the so-called 'secant equation', i.e.

$$\mathbf{H}_{i+1}\mathbf{s}_{i+1} = \mathbf{y}_{i+1}. \tag{24}$$

There is wide latitude in choosing $\mathbf{H}_{i+1}$ to satisfy this equation and much research has been devoted to this. If the update is chosen appropriately, then one can achieve 'superlinear' convergence. The updating scheme that has emerged as the method of choice for many problems is the BFGS method given by

$$\mathbf{H}_{i+1} = \mathbf{H}_i + \mathbf{y}_i\mathbf{y}_i^{\mathrm{T}}/\mathbf{y}_i^{\mathrm{T}}\mathbf{s}_i - \mathbf{H}_i\mathbf{s}_i\mathbf{s}_i^{\mathrm{T}}\mathbf{H}_i/\mathbf{s}_i^{\mathrm{T}}\mathbf{H}_i\mathbf{s}_i. \tag{25}$$

Much experience over the past decades has shown that one often needs more iterations with a quasi-Newton method than with Newton's method, but the overall work is typically less with the quasi-Newton method; see [9]. An important feature of the BFGS method (and other low-rank methods) is that one can update the inverse of the Hessian rather than the Hessian itself and, thus, no linear systems have to be solved to obtain the next iterate. Let $\mathbf{H}_i^{-1}$ be the approximation to the inverse at iteration i. The formula for updating the inverse approximation is

$$\mathbf{H}_{i+1}^{-1} = (\mathbf{I} - \rho_i\mathbf{y}_i\mathbf{s}_i^{\mathrm{T}})\mathbf{H}_i^{-1}(\mathbf{I} - \rho_i\mathbf{y}_i\mathbf{s}_i^{\mathrm{T}} + \rho_i\mathbf{s}_i\mathbf{s}_i^{\mathrm{T}}), \tag{26}$$

where $\rho_i = 1/\mathbf{s}_i^{\mathrm{T}}\mathbf{y}_i$.

In the large-scale case, it is often too expensive to store the matrix $\mathbf{H}_i$ or its corresponding inverse since it will typically be dense, even if the underlying true Hessian is sparse. To overcome this, Perry [12] suggested a 'limited-memory' formulation of the quasi-Newton method. The idea is to store the pairs $\mathbf{s}_i$ and $\mathbf{y}_i$ for the past M iterations, depending on how much memory one has. With appropriate update strategies it can be shown that the quasi-Newton step can be computed without ever forming the matrices $\mathbf{H}_i$ or $\mathbf{H}_i^{-1}$. Taking this original idea of Perry, Nocedal extended the concept (see [14] for a good summary). Although there is no hope of achieving superlinear convergence using this idea, this strategy has been quite effective in practice. Indeed, the LM-BFGS method has become the method of choice for many large-scale problems and has been shown to be superior to both steepest descent and the NLCG methods in many applications.

As Nocedal and Wright [14] (section 9.1) show, the LM-BFGS method can be efficiently applied by using the above inverse update formula. This formula requires that, at each iteration, we compute an initial approximation to the Hessian and be able to apply its inverse to a vector. Typically, one uses a scaled identity matrix as the initial Hessian, but other possibilities exist. We discuss one such possibility below.

4.4. Line search

One of the keys to creating an effective implementation of a limited-memory method, or, in fact, any successful optimization algorithm, is the line search strategy, needed to determine an acceptable step length in equation (20). In the LM-BFGS method, commonly it is assumed that step lengths should be around 1 near the solution. However, if the underlying approximation to the Hessian is too crude, then it is conceivable that such step lengths will never be realized. Much testing has revealed that this is the case for our problems. Hence, we have adapted a line search strategy similar to that used in our NLCG scheme, with the exception that the scheme allows for a more aggressive step when following a direction of negative curvature. We have to be careful, however, since we do not want to take a step that will get us into the domain of attraction of a minimizer in which we have no interest. When the function is locally convex, the line search is based upon quadratic interpolation, safeguarded with back-tracking. In previous work, heuristics were developed to limit the step length in the context of the NLCG method and the same is done here. Basically, these procedures do not allow the change in any component of $\mathbf{m}_i$ to be more than a specified percentage of the value of the largest component of $\mathbf{m}_i$. This has the effect of not allowing model updates to move drastically away from a feasible modelling domain.

4.5. Using an approximate Hessian

Although our experience using the above procedures has been generally satisfactory, to solve the very large problems of interest requires us to obtain a higher level of efficiency. Below we show how to compute an approximation to the Hessian that has some useful properties when used with our approach. Specifically, a Hessian approximation can be used with the NLCG method as a preconditioner or with the LM-BFGS method as the initial Hessian approximation for each inversion iterate. We discuss each of these now.

Recall that our description of the NLCG method included the use of a preconditioner and that two properties are required for a good preconditioner. The first is to reduce the number of inversion iterations, and the second is to do so economically. As noted above in the discussion of the LM-BFGS method, one normally uses a scaled identity matrix as the initial Hessian approximation. However, if we have a better approximation that incorporates more information about the Hessian, it seems appropriate to try this approximation for the initial Hessian matrix for each inversion iterate. Here, as in the NLCG case, the initial approximation has to be inexpensive to compute and to apply for it to be effective.

The starting point in designing our Hessian approximation is to consider the Gauss–Newton method for the case where the data are complex, but where the model parameters are strictly real. In this case (see [18]), the model update is given by

$$\mathbf{m}_{i+1} = \{(\mathbf{DJ})^{\mathrm{T}*}(\mathbf{DJ}) + \lambda\mathbf{W}^{\mathrm{T}}\mathbf{W}\}^{-1}(\mathbf{DJ})^{\mathrm{T}*}\{(\mathbf{d}_i - \mathbf{d}_{obs}) - (\mathbf{DJ})\mathbf{m}_i\}. \qquad (27)$$

Here the quantity $(\mathbf{DJ})$ describes the weighted data sensitivity or Jacobian matrix based upon the current model $\mathbf{m}_i$. Its specification to the forward problem has already been discussed previously. Now in the Gauss–Newton method, the inverse of the full Hessian (which includes second derivative terms) is approximated by

$$\mathbf{M}_i^{-1} \approx \{(\mathbf{DJ})^{\mathrm{T}*}(\mathbf{DJ}) + \lambda\mathbf{W}^{\mathrm{T}}\mathbf{W}\}^{-1}, \qquad (28)$$

where the approximation only becomes exact in the case of zero residuals as we approach a minimum of equation (2). A good choice for preconditioning the NLCG iteration is to consider a further simplification to equation (28). Specifically, we could use

$$\mathbf{M}_i^{-1} \approx \{\mathrm{diag}[(\mathbf{DJ})^{\mathrm{T}*}(\mathbf{DJ})] + \lambda\mathbf{W}^{\mathrm{T}}\mathbf{W}\}^{-1}. \qquad (29)$$

Here the operator diag extracts the diagonal of the matrix $[(\mathbf{DJ})^{\mathrm{T}*}(\mathbf{DJ})]$. Equation (29) is now in a form that can be useful for preconditioning when approximate data sensitivities (cf [11]) are used in its evaluation; Mackie [19] has also investigated this approach as a preconditioner for 3D magnetotelluric inverse problems.

These sensitivities can be obtained from the electric field integral equation

$$\mathbf{d}_i^p(\mathbf{r}) = \mathbf{d}_{i-1}^p(\mathbf{r}) + \int_v \mathrm{d}r' \, {}^{\mathbf{j}}\mathbf{G}_{i-1}^{\mathbf{d}}(\mathbf{r};\mathbf{r}')\mathbf{E}_i(\mathbf{r}')\delta\sigma(\mathbf{r}') \tag{30}$$

which determines the predicted data for the model update at iteration i for a given source (cf [20]). Here ${}^{\mathbf{j}}\mathbf{G}_{i-1}^{\mathbf{d}}$ is the dyadic Green's function arising from an electric current impulse, denoted as $\mathbf{j}$, where the data type, $\mathbf{d}$, can be set to either electric field, $\mathbf{E}$, or magnetic field, $\mathbf{H}$, in equation (30). From [21], the dyadic Green's functions ${}^{\mathbf{j}}\mathbf{G}_{i-1}^{\mathbf{E}}$ and ${}^{\mathbf{j}}\mathbf{G}_{i-1}^{\mathbf{H}}$ can be shown to satisfy the pair of tensor equations

$$\sigma_{i-1}{}^{\mathbf{j}}\mathbf{G}_{i-1}^{\mathbf{E}} - \nabla x\,{}^{\mathbf{j}}\mathbf{G}_{i-1}^{\mathbf{H}} = -\boldsymbol{\zeta}\delta(\mathbf{r}-\mathbf{r}'), \qquad \mathrm{i}\omega\mu_o{}^{\mathbf{j}}\mathbf{G}_{i-1}^{\mathbf{H}} + \nabla x\,{}^{\mathbf{j}}\mathbf{G}_{i-1}^{\mathbf{E}} = 0, \tag{31}$$

where $\boldsymbol{\zeta}$ is the identity dyadic.

Now the electric field in equation (30) also depends upon the updated model; hence it is not known. However, using the Born approximation, we can replace the electric field with the corresponding field from the previous update. That is, we assume $\mathbf{E}_{i-1}(\mathbf{r}') = \mathbf{E}_i(\mathbf{r}')$; however an additional complication remains. Even though the dyadic Green's function depends upon the current model that is known, its determination or specification requires the solution of equations (31), which are a tensor version of the 3D Maxwell's equations with impulsive sourcing. To avoid this complication, we now replace the 3D Green's function in equation (30) with one arising from a much simpler 1D layered half-space background model that will be fixed during the inversion process and can be easily determined, semi-analytically. It is conceivable that such a simple model would also be used to launch the inversion process. The approximation using a fixed Green's function is also employed in inverse scattering problems, where it is known as the Born iterative method [22], and allows us to further simplify equation (30), obtaining

$$\mathbf{d}_i^p(\mathbf{r}) \approx \mathbf{d}_{i-1}^p(\mathbf{r}) + \int_v \mathrm{d}r' \, {}^{\mathbf{j}}\mathbf{G}_{1\mathrm{D}}^{\mathbf{d}}(\mathbf{r};\mathbf{r}')\mathbf{E}_{i-1}(\mathbf{r}')\delta\sigma(\mathbf{r}'). \tag{32}$$

At this point, we are now able to use equation (32) to specify the approximate data sensitivities per unit volume. Setting $\delta\mathbf{d}_i^p(\mathbf{r}) = \mathbf{d}_i^p(\mathbf{r}) - \mathbf{d}_{i-1}^p(\mathbf{r})$, we have

$$\frac{\delta\mathbf{d}_i^p(\mathbf{r})}{\delta\sigma(\mathbf{r}')} \approx {}^{\mathbf{j}}\mathbf{G}_{1\mathrm{D}}^{\mathbf{d}}(\mathbf{r};\mathbf{r}')\mathbf{E}_{i-1}(\mathbf{r}'). \tag{33}$$

In order to specify the approximate data sensitivity for the kth model parameter, m_k, we integrate equation (33) over the cell volume, Δv_k, that is assigned to m_k in the finite-difference mesh. Thus

$$\partial\mathbf{d}^p/\partial m_k \approx \int_{\Delta v_k} \mathrm{d}r' \, {}^{\mathbf{j}}\mathbf{G}_{1\mathrm{D}}^{\mathbf{d}}(\mathbf{r};\mathbf{r}')\mathbf{E}_{i-1}(\mathbf{r}'). \tag{34}$$

For our purposes we approximate the above integration as follows:

$$\partial\mathbf{d}^p/\partial m_k \approx {}^{\mathbf{j}}\mathbf{G}_{1\mathrm{D}}^{\mathbf{d}}(\mathbf{r};\mathbf{r}_k)\mathbf{E}_{i-1}(\mathbf{r}_k)\Delta v_k, \tag{35}$$

where $\mathbf{r}_k$ corresponds to the position at the centre of the kth cell. Since there are far more image points than data, we can also make equation (35) more efficient, computationally, by employing reciprocity. This avoids sourcing the 1D tensor Green's functions at each cell centre within the imaging volume. Depending upon the data type specified in equation (34) we set ${}^{\mathbf{j}}\mathbf{G}_{1\mathrm{D}}^{\mathbf{d}}(\mathbf{r};\mathbf{r}_k)$

to ${}^{\mathbf{j}}\mathbf{G}^{\mathbf{E}}_{1\mathrm{D}}(\mathbf{r};\mathbf{r}_k)$ or to ${}^{\mathbf{j}}\mathbf{G}^{\mathbf{H}}_{\mathbf{1D}}(\mathbf{r};\mathbf{r}_k)$ and employ the following reciprocity relationships that can be derived from [21]:

$$ {}^{\mathbf{j}}\mathbf{G}^{\mathbf{E}}_{1\mathrm{D}}(\mathbf{r};\mathbf{r}_k) = {}^{\mathbf{j}}\widetilde{\mathbf{G}}^{\mathbf{E}}_{1\mathrm{D}}(\mathbf{r}_k;\mathbf{r}) \tag{36} $$

and

$$ {}^{\mathbf{j}}\mathbf{G}^{\mathbf{H}}_{1\mathrm{D}}(\mathbf{r};\mathbf{r}_k) = -{}^{\mathbf{m}}\widetilde{\mathbf{G}}^{\mathbf{E}}_{1\mathrm{D}}(\mathbf{r}_k;\mathbf{r})/(\mathrm{i}\omega\mu_o), \tag{37} $$

where $\sim$ denotes the transpose of a dyadic. Here, ${}^{\mathbf{m}}\mathbf{G}^{\mathbf{E}}_{1\mathrm{D}}$ is the 1D dyadic Green's function arising from a magnetic current impulse, denoted as $\mathbf{m}$. The 1D dyadic Green's functions satisfy equations similar to their 3D counterparts (cf equation (31)),

$$ \sigma_{1\mathrm{D}}{}^{\mathbf{j}}\mathbf{G}^{\mathbf{E}}_{1\mathrm{D}} - \nabla x{}^{\mathbf{j}}\mathbf{G}^{\mathbf{H}}_{1\mathrm{D}} = -\boldsymbol{\zeta}\delta(\mathbf{r}-\mathbf{r}'), \qquad \mathrm{i}\omega\mu_o{}^{\mathbf{j}}\mathbf{G}^{\mathbf{H}}_{1\mathrm{D}} + \nabla x{}^{\mathbf{j}}\mathbf{G}^{E}_{1\mathrm{D}} = 0 \tag{38} $$

and

$$ \sigma_{1\mathrm{D}}{}^{\mathbf{m}}\mathbf{G}^{\mathbf{E}}_{1\mathrm{D}} - \nabla x\,{}^{\mathbf{m}}\mathbf{G}^{\mathbf{H}}_{1\mathrm{D}} = 0, \qquad \mathrm{i}\omega\mu_o\,{}^{\mathbf{m}}\mathbf{G}^{\mathbf{H}}_{1\mathrm{D}} + \nabla x\,{}^{\mathbf{m}}\mathbf{G}^{\mathbf{E}}_{1\mathrm{D}} = -\boldsymbol{\zeta}\delta(\mathbf{r}-\mathbf{r}'), \tag{39} $$

except the conductivity model is 1D instead of 3D; this makes fast semi-analytical solution methods viable.

Equations (35)–(37) provide for an economical approximation for entries of the Jacobian matrix $\mathbf{J}$ and, therefore, diag$[(\mathbf{DJ})^{\mathrm{T}*}(\mathbf{DJ})]$, given that the electric field $\mathbf{E}_{i-1}$ has already been determined at the current model in equation (17). We can also consider equations (35)–(37) to be based upon an approximate adjoint method, because reciprocal or adjoint Green tensors have been conveniently utilized in the sensitivity approximation.

To complete the preconditioning step in the NLCG iteration, we determine $\mathbf{v}_i$ from

$$ \mathbf{r}_i = \mathbf{M}_i\mathbf{v}_i \tag{40} $$

by using a linear conjugate gradient solver. A similar type of solver would also be required when we are using LM-BFGS scheme. Recall that at each LM-BFGS iteration, we compute an initial approximation to the Hessian and then apply its inverse to a vector.

5. Demonstration

Several features of the preconditioning scheme will now be demonstrated. First we will show that it does indeed accelerate the convergence of the inversion iteration, achieving smaller misfits than those obtained without preconditioning over a fixed computational time interval of 24 h; all demonstrations were carried out using 64 processors on Sandia National Laboratories ASC RED parallel computing platform. Also of interest is to investigate the impact of the preconditioner on the line search and the effectiveness of the LM-BFGS iteration for different memory sizes. Finally, it is important that the robustness of the preconditioner be verified by applying it not only to a synthetic data example, but to a field example as well.

Over the last 15 years a series of salt-water injection monitoring experiments has been conducted at the University of California's Richmond Field Station. These experiments have been undertaken to simulate enhanced oil-recovery water flood and injection of contaminants/tracers into an aquifer. Alumbaugh and Newman [1] discuss one such set of experiments using cross-well EM to image the salt-water injection. Well logs of the site show that the geology of the near surface consists of unconsolidated alluvium (mud and gravel) whose electrical conductivity ranges from 0.2 to 0.02 S m^{-1}. Below 33 m depth, less conductive basement rock, consisting of sandstone and shale is encountered. Conductivity of the basement can range as low as 0.001 S m^{-1}. Cross-well electromagnetic measurements were taken by placing a vertical magnetic dipole antenna, operating at 18.5 kHz, in the centre/injection well. Vertical field measurements were then taken in the four surrounding wells at 5 m intervals

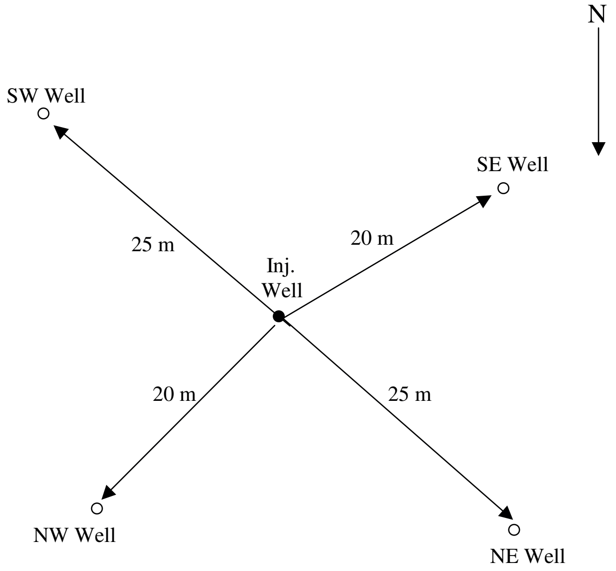

Figure 1. Layout of the Richmond-field site experiment.

from 5 to 55 m depth. A similar range of source depths, sampled at 2.5 m intervals, was also employed in the experiment. The well layout of the Richmond-field site is shown in figure 1.

In the process of analysing the field data, Alumbaugh and Newman [1] carried out a design and resolution analysis of the experiment by considering probable models of the site. We have taken the data for one of these models and inverted them using the inversion schemes previously described. The electric properties of this model that we will attempt to recover consist of a 30 m thick conductive overburden, which includes a 4 m thick aquifer or sand channel, 20 m south-east of the injection well, near 22 m depth. The plume corresponding to the injected salt-water was modelled to be several metres thick, and placed near the injection well at 30 m depth. It was assigned a conductivity of 0.2 S m^{-1}, with lateral dimensions of about 5 m on a side. Below 30 m depth, an electrical basement is included in the model. The basement is insulating, exhibiting low conductivities, less than 0.02 S m^{-1} and includes a vertical contact or fault, based upon well log data; a detailed graphic of the model is shown in figure 2. Note that four additional wells were used in the design to study the resolving power of the data through better well coverage. The additional wells are hypothetical and do not actually exist at the Richmond field site.

The amount of data analysed consists of 3654 data points, simulated in eight wells that surround the injection well. The noise model assumed for the data is Gaussian and is based upon a standard deviation equal to 2% of the magnetic field for amplitudes greater than 1×10^{-6} A m^{-1}, and a standard deviation equal to 2×10^{-8} when the field drops below this value. The data were weighted within the inversion processes with this noise distribution. The inversion of the 3D cross well data set was launched with a 0.0333 S m^{-1} half-space using a mesh of 121 495 cells. Figure 3 shows the image recovered at 30 m depth using a preconditioned NLCG scheme at the 48th iteration. The plume is clearly imaged, off the centre of the injection well. Also shown are the eight receiver wells that surround the injection well. Cross-sections of the recovered model along the x and y directions are shown in figure 4. These cross-sections indicate a resistive electrical basement, but are not able to precisely

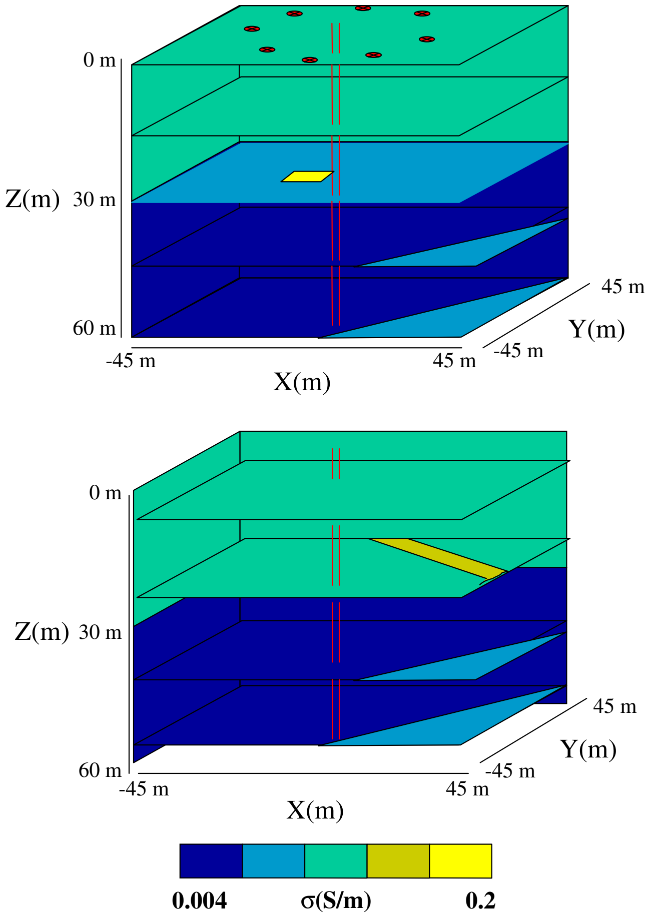

Figure 2. Test model used to test the preconditioned NLCG and LM-BFGS inversion schemes. The model is based on a resolution analysis of a cross-well EM imaging experiment at the Richmond Field Station. Two views of the same model are included such that horizontal conductivity slices can be shown at nine different depths, with back and left-side panels showing how the conductivity varies continuously with depth. The location of the eight observation wells is shown at the surface of the model. The injection well is indicated in red. The figure shows a 4 m thick channel deposit at 22 m depth, a 4 m thick plume centred near 30 m depth, the contact between the conductive overburden and resistive basement and a vertical contact within the basement.

recover all features of the basement, because of the limited resolving power of the data, even with the augmented well coverage. Regarding this last point, we refer the interested reader to resolution analysis found in [1] for more details on this point.

Convergence of the preconditioned NLCG inversion iteration is shown in figure 5. In this figure we have also plotted the data component of the error functional as this measure can be used to indicate when we have fitted the data to within the assumed data errors. Ideally, we want to fit the data to a target level of one, but have only achieved a data fit, based upon the weighted squared error, which is 3.28 times above our target misfit. To achieve the target misfit, it is necessary to reduce the amount of regularization or model smoothing added to stabilize the inverse problem; the tradeoff parameter λ appearing in equation (2) was fixed at 10. Nevertheless, even in this example, where λ is set too large, we are still able to resolve the

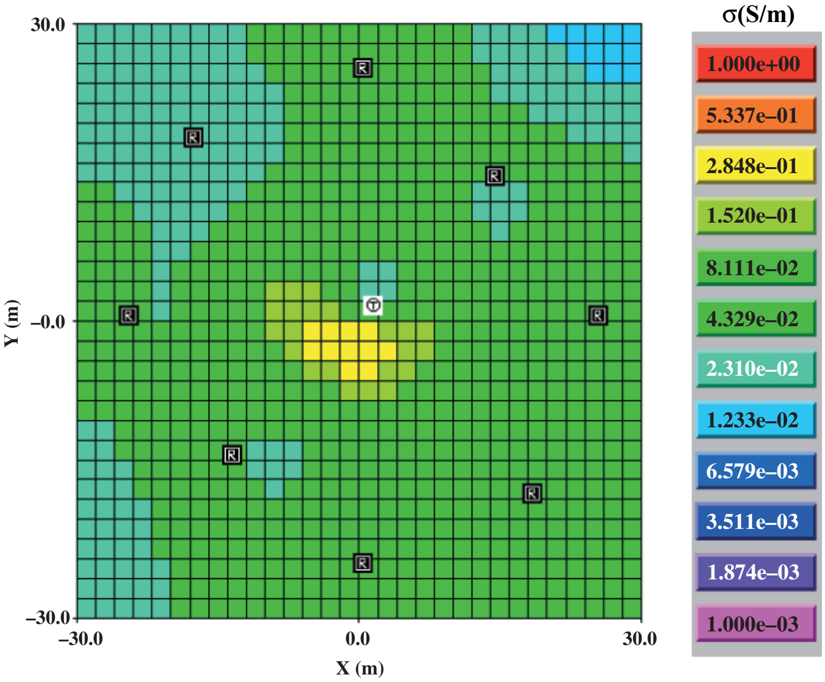

Figure 3. Reconstructed image of the test model at 30 m depth. The injected plume is clearly indicated in the image. The eight observation wells and injection well are indicated in the figure. T denotes a transmitter position in the injection well and R a detector position in the observations wells.

injected salt-water plume with an accuracy comparable with when we fit the data to the target misfit level of one (see [1]).

While all the above-mentioned details are important for this part of the demonstration, our focus in on accelerating the inverse iteration and figure 5 shows clearly that we have achieved this goal. The non-preconditioned NLCG iteration converges significantly slower than the preconditioned one, eventually converging to misfit level above that observed for the preconditioned problem over the same 24 h period in which the calculations were carried out (see table 1 as well). Similar solution acceleration is observed for the preconditioned LM-BFGS iteration in figure 6 with a memory size of 5; the recovered model is very similar to reconstructions illustrated in figures 3 and 4 for the preconditioned NLCG iteration and is not shown. We also investigated the performance of the algorithm on this test problem for different memory sizes. The results are summarized in tables 2 and 3 for the preconditioned and non-preconditioned inversion iterations. These tables show that increasing the memory size slightly degrades the performance of the algorithm. Best results are obtained with smaller memory sizes. This may be because the number of iterations with the smaller memory size is larger in the allotted time; it may also be due to the fact that using data from iterations too long ago is not helpful, especially at the early stages of the process. More investigation of these issues, including an adaptive strategy, where the memory size varies as the iteration proceeds, is clearly warranted.

To further verify the robustness of the preconditioner, we now demonstrate it on field data acquired at the Richmond field site. The cross-well measurements to be analysed were

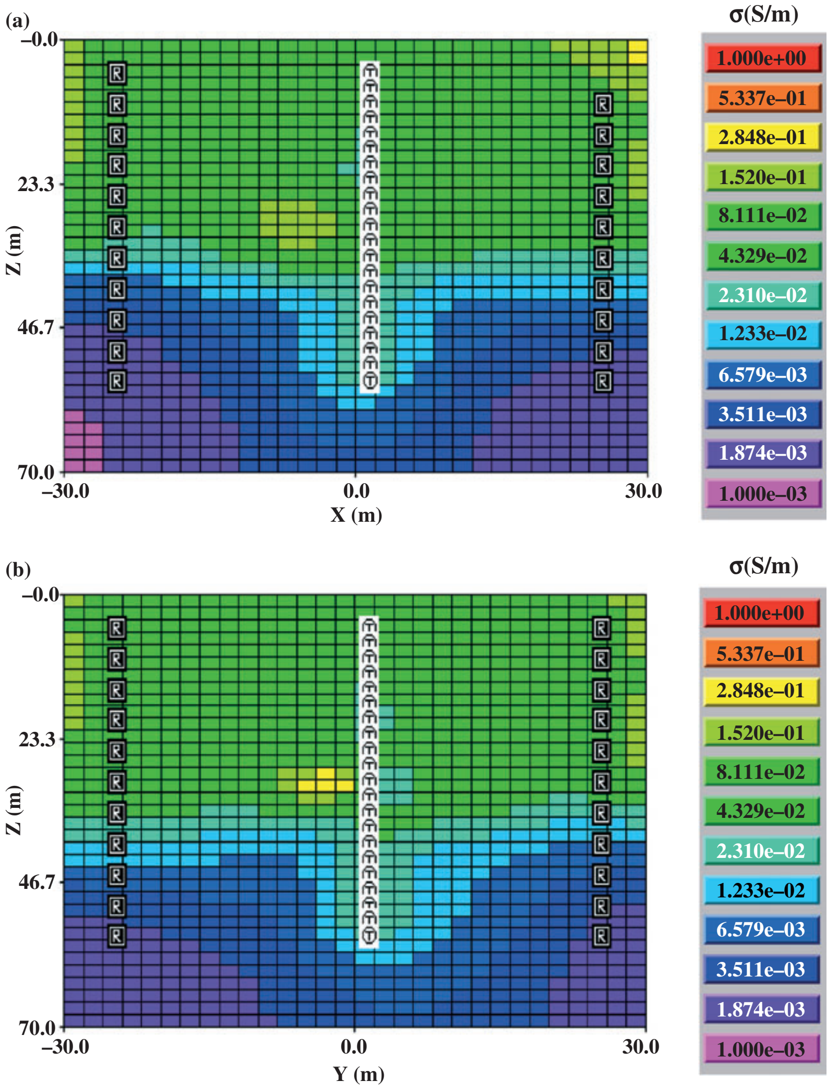

Figure 4. Cross-sections of the reconstructed test model: (a) *X–Z* cross-sectional image and (b) *Y–Z* cross-sectional image. Cross-sections indicate a resistive electrical basement consistent with the test model. T and R denote transmitter and detector locations.

collected in the preinjection phase of the experiment. The amount of data analysed consists of 2024 data points, acquired in the four wells that surround the injector well (figure 1). A finite-difference mesh of 133 632 cells was used to invert the data. A 0.0333 S m^{-1} half-space was again employed as the starting model, where we fix the regularization parameter at 10. Below we will discuss our findings when this parameter is allowed to vary. However, the noise model is now different from the synthetic example because the field data have far greater noise content in the longer source–receiver offsets (see [1] for more details). Instead, following Alumbaugh and Newman [1], the data are weighted by 2% of the maximum amplitude for each source relative to all receivers in a given observation well. This type of weighting puts

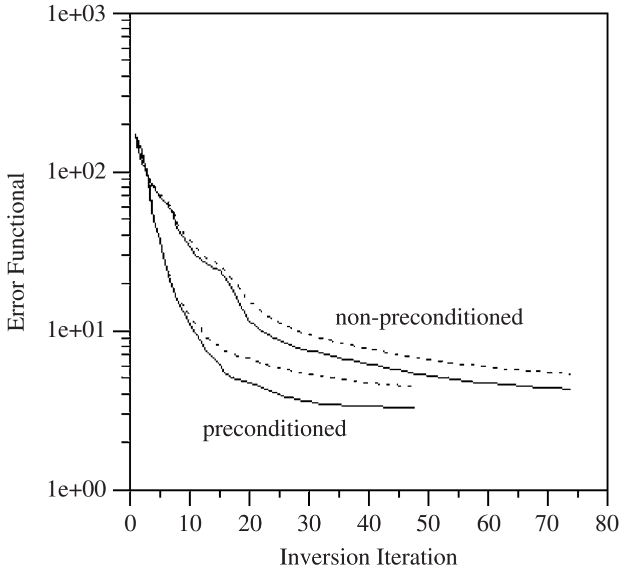

Figure 5. Convergence of the NLCG iteration is shown for the test model problem of figure 2. Preconditioned and non-preconditioned convergence is illustrated. We have also plotted the error function with (– – –) and without (——) the stabilization or regularization term for both types of iterations. Beyond the first iteration, the data component of the error functional is always smaller than the total error functional.

Table 1. NLCG algorithm with/without preconditioning; ϕ_d corresponds to the data component of the error functional. All results based on 24 h of CPU time using 64 processors on ASC_RED for synthetic test data.

	Iterations	ϕ	ϕ_d
Preconditioned	48	4.46	3.28
Non-preconditioned	74	5.33	4.29

more emphasis on the short offset positions than the longer ones. Figure 7 illustrates the convergence properties of the NLCG inversion iteration, with and without preconditioning. The benefits of preconditioning are clear and convincing with the preconditioned iteration converging to the target misfit of one, while the non-preconditioned iteration has yet to do so. These results are further summarized in table 4. Although not shown, similar behaviour has been found to hold for the LM-BFGS inversion iteration with best results, again, obtained for small memory sizes. Figure 8 shows cross-sections of the model recovered with the NLCG preconditioned inversion iteration. The contact between the conductive overburden and resistive basement are clearly indicated below 35 m depth, which is in good agreement with the geological section based on the well log information. However, one should not place much confidence in the variations of the basement conductivity, given our previous findings from the design of the experiment. A conductive anomaly is also observed in the overburden, right of the injection well in the '*YZ*' cross-section image. Alumbaugh and Newman [1] have verified that this anomaly is an image artifact, arising from measurements that have not been corrected for horizontal deviations in the boreholes.

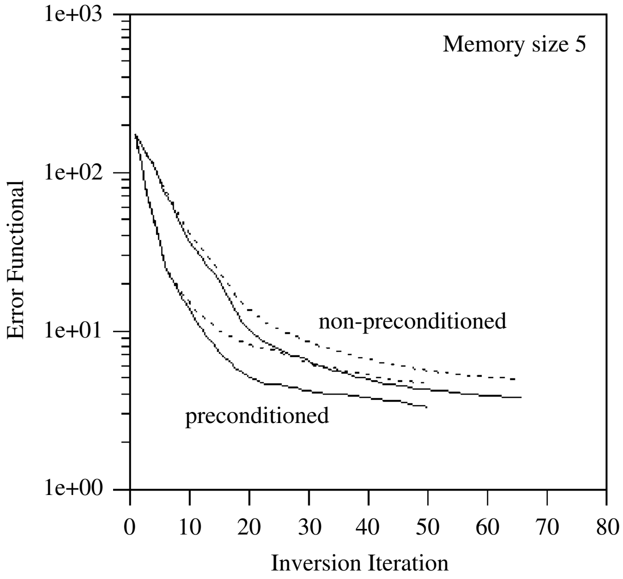

Figure 6. Convergence of the LM-BFGS iteration is shown for the test model problem of figure 2. Preconditioned and non-preconditioned convergence is illustrated. We have also plotted the error function with and without the stabilization (regularization) term for both types of iterations.

Table 2. Performance of the preconditioned LM-BFGS algorithm for different memory sizes; all results based on 24 h of CPU time using 64 processors on ASC_RED for synthetic test data.

Memory size	Iteration	ϕ	ϕ_d
5	50	4.61	3.31
10	48	4.59	3.30
20	45	4.79	3.48
40	50	4.66	3.72
All	37	4.70	3.39

Table 3. Performance of the LM-BFGS algorithm without preconditioning for different memory sizes; all results based on 24 h of CPU time using 64 processors on ASC_RED for synthetic test data.

Memory size	Iteration	ϕ	ϕ_d
5	66	4.89	3.76
10	61	5.06	3.89
20	62	5.02	3.86
40	62	5.02	3.86
All	58	5.17	3.96

It is worthwhile to point out that when the post-injection data were inverted, there was no clear indication of the injected salt-water plume even when the post- and pre-injection images were differenced. Convergence to the target misfit was easily achieved using the preconditioned NLCG scheme for the same type of noise assumptions that were employed with

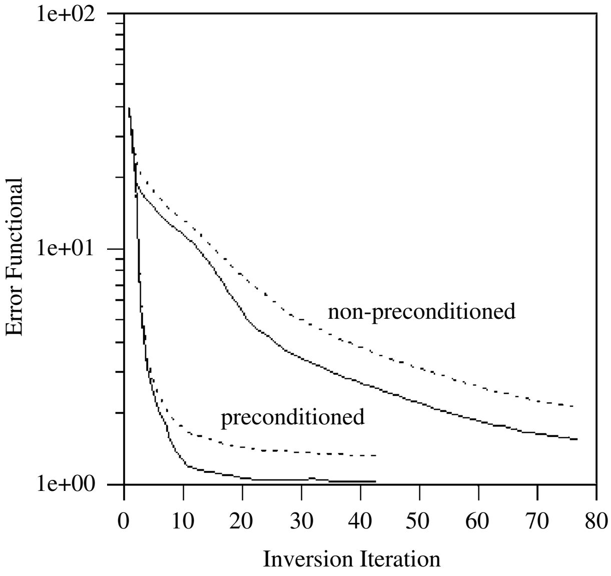

Figure 7. Convergence of the NLCG iteration, with and without preconditioning, for the pre-injection phase of the Richmond-field experiment.

Table 4. NLCG algorithm with/without preconditioning as applied to the preinjection Richmond cross-well data set. All results based on 24 h of CPU time using 64 processors on ASC_RED for synthetic test data.

	Iterations	ϕ	ϕ_d
Preconditioned	43	1.33	1.02
Non-preconditioned	77	1.54	2.12

the pre-injection data. When Alumbaugh and Newman [1] inverted the Richmond field data with the same noise model, but using a much more computationally intensive Gauss–Newton scheme, the plume was clearly imaged. This curious result reminds us of the non-uniqueness issues that can plague non-linear inverse problems, even when these problems are regularized.

Overall we see superior performance when the preconditioner is employed in the inversion iterations for both NLCG and LM-BFGS schemes. For the most part, a quadratic line search used in conjunction with the preconditioned gradients produces satisfactory results. However, at the early stages of the preconditioned inversion iteration, we have also observed that the quadratic line search can fail because of negative curvature, particularly if the regularization parameter is set too small. A remedy to this situation is to use a more careful line search. At this time, we see little difference between using a preconditioned LM-BFGS scheme employing small memory sizes and preconditioned NLCG; overall the performances of both preconditioned schemes appear to be very similar. We also found that using large memory sizes degrades, somewhat, the performance of the LM-BFGS scheme. Nevertheless, several improvements for the LM-BFGS method are still possible, and we are continuing to pursue these. For example, the general expectation for LM methods is that a step length of one will eventually be acceptable. We have not observed this at all. In fact, in our tests, the step lengths are almost never near one. This suggests again that a more careful line search at the first iteration

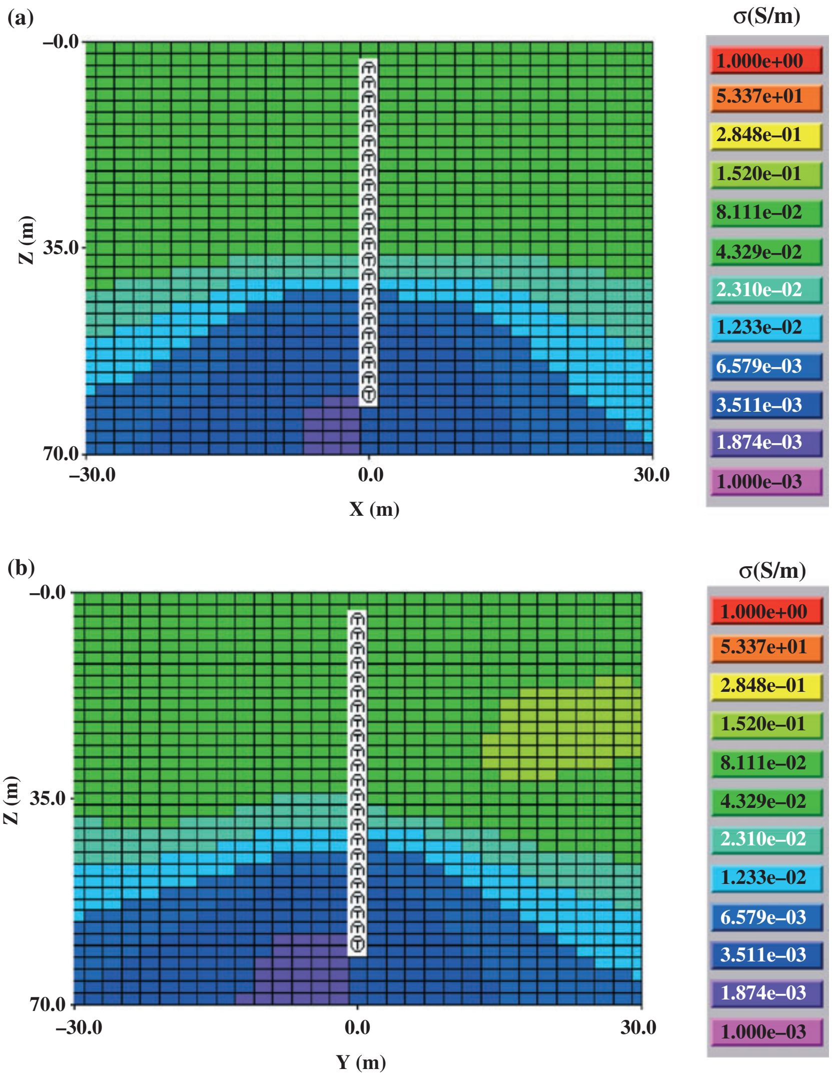

Figure 8. Cross-sections of the model recovered with the NLCG preconditioned inversion iteration for pre-injection phase of the Richmond-field experiment: (a) '*XZ*' or east–west cross-section, (b) '*YZ*' or north–south cross-section. Note the receiver wells in figure 1 are outside the planes of the two cross-sections.

would be helpful in establishing the approximate scale of the problem. This step length could then be saved. At subsequent iterations, we could use the saved step for evaluation and then try to adjust this, in a trust-region like strategy to adapt the step length as we progress. Also, we could try to improve the preconditioner for its specific use in the quasi-Newton framework by adding another scaled diagonal term that would serve as the initial approximation for all of the omitted terms in the Hessian. On the other hand, using only a few dozen inversions iterations in conjunction with a low rank method for updating the Hessian, it may be unrealistic to expect faster convergence compared with conjugate gradients. More research is needed in the application of limited memory quasi-Newton methods for large-scale inverse problems.

6. Conclusions

We have provided a framework for preconditioning the inversion iteration for nonlinear conjugate gradient and limited memory quasi-Newton methods. Key to our approach is the use of an approximate adjoint concept that allows for an economical approximation of the Hessian. We anticipate that better approximations to the Hessian than the type employed here may lead to better preconditioners. Allowing for more fill-in would be one possibility. Another would use the adjoint fields of a 3D background model that is also used to launch the inversion process. While it would be necessary to save or store all the adjoint fields within the imaging volume, for each of the different detectors/receivers that source these fields, the computation of these adjoint fields needs to be done only once.

Acknowledgments

During the last three years, in which preconditioning ideas discussed in the paper were developed, R Mackie has been conducting similar investigations, focusing on application to 3D magnetotelluric inverse problems. Technical discussions with R Mackie were helpful in confirming the approximate adjoint concept as presented here. This work was carried out at Lawrence Berkeley and Sandia National Laboratories with funding provided by United States Department of Energy, Office of Basic Energy Sciences. Additional support was provided by Sandia Laboratories Laboratory Directed Research Development (LDRD) office. Sandia is a multi-program laboratory operated by the Sandia Corporation, a Lockhead Martin Company, for the United States Department of Energy under Contract DE-AC04-94AL85000.

References

[1] Alumbaugh D L and Newman G A 1997 3-D massively parallel electromagnetic inversion—Part II. Analysis of a cross well experiment *Geophys. J. Int.* **128** 355–63

[2] Newman G A, Recher S, Tezkan B and Neubauer F M 2003 3D inversion of a scalar radio magnetotelluric field data set *Geophys.* **68** 782–90

[3] Mackie R L and Madden T R 1993 Three-dimensional magnetotelluric inversion using conjugate gradients *Geophys. J. Int.* **115** 215–29

[4] Newman G A and Alumbaugh D L 1997 Three-dimensional massively parallel electromagnetic inversion—I. Theory *Geophys. J. Int.* **128** 345–54

[5] Dorn O, Bertete-Aguirre H, Berryman J G and Papanicolaou G C 1999 A nonlinear inversion method for 3D electromagnetic imaging using adjoint fields *Inverse Problems* **15** 1523–58

[6] Haber E, Ascher U and Oldenburg D W 2004 Inversion of 3D electromagnetic data in frequency and time domain using an inexact all-at-once approach *Geophys.* in press

[7] Newman G A and Alumbaugh D L 2000 Three-dimensional magnetotelluric inversion using non-linear conjugate gradients *Geophys. J. Int.* **140** 410–24

[8] Alumbaugh D L and Newman G A 2000 Image appraisal for 2D and 3D electromagnetic inversion *Geophys.* **65** 1455–67

[9] Dennis J and Schnabel R 1996 *Numerical Methods for Unconstrained Optimization and Nonlinear Equations* (Philadelphia, PA: SIAM)

[10] Rodi W and Mackie R L 2001 Nonlinear conjugate gradients algorithm for 2-D magnetotelluric inversion *Geophys.* **66** 174–87

[11] Farquharson C G and Oldenburg D W 1996 Approximate sensitivities for the electromagnetic inverse problem *Geophys. J. Int.* **126** 235–52

[12] Perry J M 1977 A class of conjugate gradient algorithms with a two step variable-metric memory *Discussion Paper 269* Center of Mathematical Studies in Economics and Management Science, Northwestern University

[13] Nocedal J 1980 Updating quasi-Newton matrices with limited storage *Math. Comput.* **35** 773–82

[14] Nocedal J and Wright S J 1999 *Numerical Optimization* (New York: Springer)

[15] Newman G A and Hoversten G M 2000 Solution strategies for 2D and 3D electromagnetic inverse problems *Inverse Problems* **16** 1357–75

[16] Haber E and Oldenburg D 1997 Joint inversion a structural approach *Inverse Problems* **13** 63–77

[17] Alumbaugh D L, Newman G A, Prevost L and Shadid J N 1996 Three dimensional, wideband electromagnetic modeling on massively parallel computers *Radio Sci.* **31** 1–23

[18] Menke W 1984 *Geophysical Data Analysis: Discrete Inverse Theory* (New York: Academic)

[19] Mackie R L 2004 personal communication

[20] Newman G A 1995 Cross well electromagnetic inversion using integral and differential equations *Geophys.* **60** 899–911

[21] Zhdanov M 1988 *Integral Transforms in Geophysics* (New York: Springer)

[22] Chew W C 1991 *Waves and Fields in Inhomogeneous Media* (Oxford: Oxford University Press)