

Improved hydrogeophysical characterization and monitoring through parallel modeling and inversion of time-domain resistivity and induced-polarization data

Timothy C. Johnson¹, Roelof J. Versteeg¹, Andy Ward², Frederick D. Day-Lewis³, and André Revil⁴

ABSTRACT

Electrical geophysical methods have found wide use in the growing discipline of hydrogeophysics for characterizing the electrical properties of the subsurface and for monitoring subsurface processes in terms of the spatiotemporal changes in subsurface conductivity, chargeability, and source currents they govern. Presently, multichannel and multielectrode data collections systems can collect large data sets in relatively short periods of time. Practitioners, however, often are unable to fully utilize these large data sets and the information they contain because of standard desktop-computer processing limitations. These limitations can be addressed by utilizing the storage and processing capabilities of parallel computing environments. We have developed a parallel distributed-memory forward and inverse modeling algorithm for analyzing resistivity and time-domain induced polar-

ization (IP) data. The primary components of the parallel computations include distributed computation of the pole solutions in forward mode, distributed storage and computation of the Jacobian matrix in inverse mode, and parallel execution of the inverse equation solver. We have tested the corresponding parallel code in three efforts: (1) resistivity characterization of the Hanford 300 Area Integrated Field Research Challenge site in Hanford, Washington, U.S.A., (2) resistivity characterization of a volcanic island in the southern Tyrrhenian Sea in Italy, and (3) resistivity and IP monitoring of biostimulation at a Superfund site in Brandywine, Maryland, U.S.A. Inverse analysis of each of these data sets would be limited or impossible in a standard serial computing environment, which underscores the need for parallel high-performance computing to fully utilize the potential of electrical geophysical methods in hydrogeophysical applications.

INTRODUCTION

The use of low-frequency electrical geophysics (including standard DC resistivity, induced polarization (IP), and self-potential) for characterization, i.e., inversion of a single static data set, dates back to the early 1920s. The application of time-lapse electrical methods to investigate hydrological and biogeochemical processes began in the mid-1980s (Daily, 1984; Parra, 1988). Numerous studies have examined different aspects of electrical methods, including data acquisition, data processing (modeling and inversion), and interpretation, in characterization and time-lapse (or monitoring) mode. Electrical methods have been used for environmental, geotechnical, and

resource-recovery applications. Electrical geophysical signatures associated with hydrogeological, biogeochemical, and temperature-related processes motivate continued application of these methods (e.g., Legaz et al. [2009] for geothermal systems).

Multielectrode, multichannel instrumentation permits the rapid collection of spatially and temporally dense data sets. Although commercially available, desktop-based processing software is adequate for inverting 2D and moderately sized 3D data sets, such software is poorly suited for processing large 3D and 4D data sets, with hundreds or thousands of surveys. Data sets recently collected by the U. S. Geological Survey (USGS) contain more than 1200 2D time-lapse resistivity surveys (Henderson et al., 2009), and data collected

Manuscript received by the Editor 30 July 2009; revised manuscript received 6 December 2009; published online 30 September 2010.

¹Idaho National Laboratory, Energy Resource Recovery and Management, Idaho Falls, Idaho, U.S.A. E-mail: timothy.johnson@inl.gov; roelof.versteeg@inl.gov.

²Pacific Northwest National Laboratory, Energy and Environment Directorate, Richland, Washington, U.S.A. E-mail: andy.ward@pnl.gov.

³U. S. Geological Survey, Office of Groundwater, Branch of Geophysics, Storrs, Connecticut, U.S.A. E-mail: daylewis@usgs.gov.

⁴Colorado School of Mines, Department of Geophysics, Golden, Colorado, U.S.A. and Université de Savoie, INSU-CNRS LGIT, Le Bourget du Lac, France. E-mail: arevil@mines.edu.

© 2010 Society of Exploration Geophysicists. All rights reserved.

at the Brandywine, Maryland, U.S.A., site (Versteeg and Johnson, 2008) consist of approximately 200 3D resistivity and IP surveys with approximately 50,000 measurements each. Coupled or joint inversion approaches (e.g., electrical geophysical data inverted with geostatistical and/or hydrological data to estimate hydrological properties) also are computationally intensive (Johnson et al., 2009). For such applications, a computational environment is required that currently cannot be provided by desktop computers.

Here we discuss a parallel DC-resistivity and time-domain induced-polarization modeling and inversion code. First, we outline the forward equations and background theory for modeling DC-resistivity and time-domain IP measurements and the parallel implementation that involves distributing the pole potential computations over parallel processors. We then discuss the parallel inversion algorithm, which implements parallel Jacobian matrix computation and storage, and the application of a parallel matrix inversion algorithm. Three example applications demonstrate the computational advantages of parallel modeling/inversion and underscore the need for parallel computing in hydrogeophysics: (1) resistivity characterization of the Hanford 300 Area Integrated Field Research Challenge site in Hanford, Washington, U.S.A., (2) resistivity characterization of a volcanic island in the southern Tyrrhenian Sea in Italy, and (3) resistivity and IP monitoring of biostimulation at a Superfund site in Brandywine, Maryland, U.S.A.

FORWARD-MODELING ALGORITHM

Following the notation of Li and Oldenburg (2000), the DC potential Φ_σ at position \mathbf{r} induced by a point source of current I injected at position \mathbf{r}_0 within a medium with conductivity $\sigma(\mathbf{r})$ is given by

$$\nabla \cdot (\sigma(\mathbf{r}) \nabla \Phi_\sigma(\mathbf{r})) = -I\delta(\mathbf{r} - \mathbf{r}_0), \quad (1)$$

in the absence of IP effects and subject to appropriate boundary conditions. Equation 1 can be solved numerically for $\Phi_\sigma(\mathbf{r})$ (e.g., Dey and Morrison, 1979; Pridmore et al., 1981), resulting in a general system of equations given by

$$(\mathbf{A} + \mathbf{B})\Phi_\sigma = \mathbf{s}, \quad (2)$$

where \mathbf{A} represents a sparse system of domain equations, \mathbf{B} represents a sparse system of boundary equations, Φ_σ represents the discretized conductivity field vector, and \mathbf{s} represents the source-current distribution. Note that \mathbf{A} is a function of σ , and \mathbf{B} can be a function of σ and \mathbf{s} , depending on the implementation of the boundary conditions. A complete discussion of the evolution and state of the science for solving equation 1 in DC-resistivity modeling and inversion applications is beyond the scope of this paper, but readers are referred to Rücker et al. (2006) and references therein, which describe the finite-element formulations used in this work for computing \mathbf{A} and \mathbf{B} .

In the presence of IP effects analyzed in the time domain, the total potential field Φ_η is given by

$$\nabla \cdot (\sigma(\mathbf{r})(1 - \eta(\mathbf{r})) \nabla \Phi_\eta(\mathbf{r})) = -I\delta(\mathbf{r} - \mathbf{r}_0) \quad (3)$$

where $\eta(\mathbf{r})$ is the chargeability at position \mathbf{r} (Seigel, 1959). The secondary potential Φ_s and/or apparent chargeability η_a measured during an IP survey are given by (Oldenburg and Li, 1994)

$$\Phi_s = \Phi_\eta - \Phi_\sigma \quad (4)$$

and

$$\eta_a = \frac{\Phi_\eta - \Phi_\sigma}{\Phi_\eta}. \quad (5)$$

Thus, the simulated secondary potential and apparent chargeability can be computed with two separate forward runs solving equations 1 and 3.

Measured potential data can be simulated by solving equation 2 for each source/sink current pair and then taking the difference between each positive and negative measurement electrode collected for that current pair. This requires one forward computation for each modeled current/electrode pair. Commonly, there are more current pairs than electrodes. In this case, a more efficient approach for simulating the measured data is to first compute the pole solution for each electrode used as a current source or sink. The potential distribution for a source/sink current pair then is constructed by subtracting the pole potential associated with the current-sink electrode from the pole potential associated with the current-source electrode. In this case, one forward run is required for each electrode used as a source or current sink (which often is every electrode). Thus, the major computational effort in the forward problem is computing the pole solutions for a point-source current injected at each electrode location.

Because each pole solution can be computed independently, this calculation lends itself well to implementation in a parallel environment. Such an environment typically has multiple processors (e.g., 2400 processors in the Idaho National Laboratory Ice Storm Cluster). The processors, hereafter referred to as nodes, are organized in a master/slave configuration, where one node serves as the master node that controls execution of the program and instructs slaves concerning which routines to execute. The slave nodes receive commands from the master and handle most of the computing effort. The slaves operate using a data- or loop-level parallelism strategy, meaning that each slave executes the same code on different parts of the distributed data. In the forward mode, each slave is assigned one or more electrodes for which to compute a pole and submits the resulting pole solution back to the master node at completion. The parallel communications (i.e., communication between nodes) are implemented using the message passage interface (MPI) protocol (Gropp et al., 1999).

A flow diagram for the forward execution is shown in Figure 1. First, the master node reads the information necessary to execute the run (e.g., conductivity distribution, mesh configuration, survey configuration) and passes this information to the slave nodes. Next, the master node assigns the electrodes evenly to the slave nodes and instructs the slave nodes to compute the pole solutions for each assigned electrode. Finally, the master node collects the pole solutions from the slaves, assembles the simulated measurements, and outputs simulation results as requested by the user.

PARALLEL INVERSE MODELING ALGORITHM

Inverse equations

To present the inverse equations, we follow and implement the general theoretical development of Farquharson (2008), which accommodates most published measures of data and model misfit such as the ℓ_1 -through ℓ_p -norms, M-measure (Huber, 1964), and perturbed ℓ_p -norm (Farquharson, 2008). The objective of the inversion is to modify the estimated conductivity distribution σ_{est} in order to minimize the function Φ :

$$\Phi = \Phi_d(\mathbf{u}_d) + \beta \Phi_m(\mathbf{u}_m) \quad (6)$$

where Φ_d is an operator giving a scalar measure of the misfit between the observed and simulated resistivity (or chargeability) according to the desired norm, Φ_m is the corresponding scalar measure of the difference between σ_{est} and constraints placed upon the structure of σ_{est} , and β is a regularization (or trade-off) parameter that controls the contribution of Φ_m to Φ in comparison with Φ_d . The general forms of \mathbf{u}_d and \mathbf{u}_m are

$$\mathbf{u}_d = \mathbf{W}_d(\phi_{\text{obs}} - \phi_{\text{pred}}) \quad (7)$$

and

$$\mathbf{u}_m = \mathbf{W}_m(\sigma_{\text{est}} - \sigma_{\text{ref}}), \quad (8)$$

where ϕ_{obs} and ϕ_{pred} are the observed and predicted (i.e., simulated) resistivity (or chargeability) measurement vectors, respectively; σ_{est} and σ_{ref} are the estimated and reference electrical-log/conductivity-distribution vectors; \mathbf{W}_d is the data-weighting matrix; and \mathbf{W}_m is the model-weighting or regularization matrix. Note that we estimate the log-conductivity distribution to enforce positivity.

Equations 6–8 lead to a system of linear equations of the general form (Farquharson, 2008)

$$(\mathbf{J}^T \tilde{\mathbf{W}}_d \mathbf{J} + \beta \tilde{\mathbf{W}}_m) \delta \sigma_{\text{est}} = \mathbf{J}^T \tilde{\mathbf{W}}_d (\phi_{\text{obs}} - \phi_{\text{pred}}) + \beta \tilde{\mathbf{W}}_m (\sigma_{\text{ref}} - \sigma_{\text{est}}), \quad (9)$$

where \mathbf{J} is the Jacobian matrix, which contains the derivative of the measurements with respect to the estimated conductivity values; $\tilde{\mathbf{W}}_d$ is the generalized data-weighting matrix operator, including the effects of the data weighting specified in \mathbf{u}_d and the data misfit norm expressed by Φ_d ; and $\tilde{\mathbf{W}}_m$ is the generalized model-weighting matrix operator, including the model constraints specified in \mathbf{u}_m and the model norm specified by Φ_m .

The update vector $\delta \sigma_{\text{est}}$ represents a solution to equation 9 that, when added to σ_{est} , will decrease the value of the objective function (equation 6). Note that \mathbf{J} is dependent upon σ_{est} . Thus, the inversion operation is nonlinear and iterates by solving for $\delta \sigma_{\text{est}}$, adding it to the previous solution, calculating the data misfit, and repeating until the observed data are matched to within some criteria determined by the data noise. During this process, β is adjusted to provide a solution that maintains the data fit and also honors the model constraints to the extent possible. The value \mathbf{J} is recomputed during each iteration. When using a norm other than ℓ_2 as the measure of misfit, $\tilde{\mathbf{W}}_d$ and $\tilde{\mathbf{W}}_m$ are functions of σ_{est} and must be recomputed during each iteration. This method is known as *iteratively reweighted least squares* (Scales and Gersztenkorn, 1988) and is included in the parallel formulation shown in the next section for generality.

The generalized weighting matrices $\tilde{\mathbf{W}}_d$ and $\tilde{\mathbf{W}}_m$ are sparse with small storage requirements. Thus, operations involving $\tilde{\mathbf{W}}_d$ and $\tilde{\mathbf{W}}_m$ can be executed quickly using element sparse matrix operations. Conversely, \mathbf{J} is a large, dense matrix having n_{obs} rows and n_{est} columns, where n_{obs} is the number of measurements and n_{est} is the number of estimated parameters (i.e., cells or elements in the inversion mesh). Solving this inverse problem for large 3D problems poses two challenges. First, the memory requirement associated with storing \mathbf{J} rapidly exceeds the available memory on standard serial computers. Second, the total computational effort becomes extensive, leading to extreme run times.

There are two primary components to the computational effort. The first component consists of simulating the measurements and computing \mathbf{J} at each iteration, which requires one complete forward simulation of each source/sink-current pair and a corresponding set of adjoint state computations (McGillivray and Oldenburg, 1990; Sasaki, 1994). The second component involves solving equation 9. Both of these components (when solved in a serial environment) rapidly become cumbersome as the size of the problem increases.

In the next two sections, we discuss the parallel storage and computation of \mathbf{J} and a parallel conjugate-gradient least-squares (CGLS) algorithm (Hestenes and Stiefel, 1952) that we implement for solving equation 9.

Parallel Jacobian matrix storage and computation

Details concerning the adjoint-state computations required to compute \mathbf{J} are beyond the scope of this paper but are given in numerous manuscripts addressing DC resistivity inversion, including those suggested in the previous section and references therein. Spe-

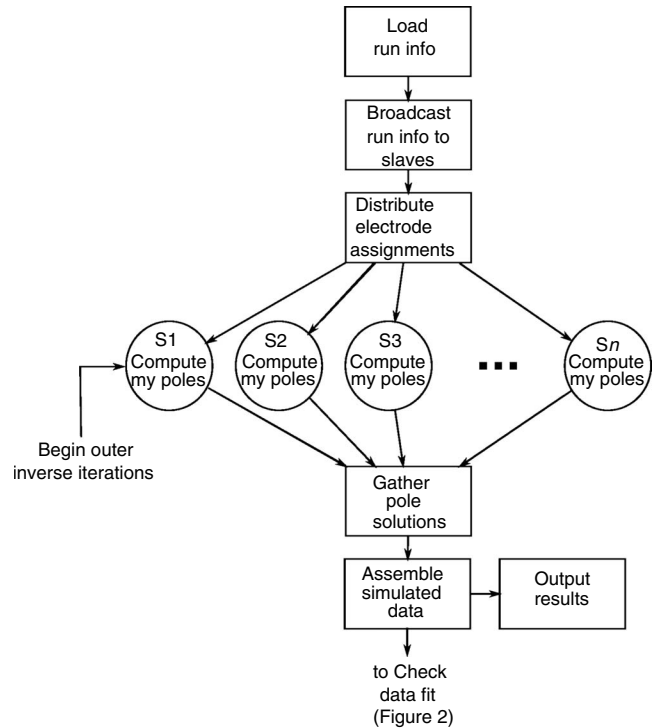


Figure 1. Generalized flow model, showing parallel implementation of forward computations. The primary computational burden (i.e., the pole-solution computation) is divided among slave processors (circles). Squares are master tasks; circles are slave tasks.

cifically, the finite-element adjoint-state formulation and computation presented by Günther et al. (2006) describe the formulation used in this paper.

A generalized flow diagram showing the parallel storage and computation of \mathbf{J} is shown in Figure 2. Once the forward computation for a particular iteration is complete, the master node computes the objective function and determines if the target data fit has been achieved. If so, the inversion is complete and the results are reported. If not, the master node adjusts the regularization parameter β as required (not shown in Figure 2) and broadcasts the full set of pole solutions to every slave, which is required for the adjoint computations. Next, the master computes (or updates) the generalized weighting $\tilde{\mathbf{W}}_d$ and $\tilde{\mathbf{W}}_m$ and waits for the slaves to complete the \mathbf{J} computation before proceeding to the parallel CGLS algorithm.

The \mathbf{J} computation is an iterative process that can be executed row by row. During the information-broadcast stage shown in Figure 1, the n_{obs} rows of \mathbf{J} are distributed evenly over the slave nodes, and each node is informed of the row interval to which it is assigned. Thus, assuming n slave nodes, each slave node will compute approximately n_{obs}/n rows of \mathbf{J} , requiring storage of a matrix measuring n_{obs}/n rows \times n_{est} columns. Upon receiving the pole solutions from the master, each slave allocates the required memory, executes the adjoint-state computations for its row interval, and waits for further instructions from the master. These instructions are provided during the execution of the parallel CGLS algorithm.

Parallel CGLS algorithm

The CGLS algorithm solves equation 9, which can be represented in a matrix form as

$$\begin{bmatrix} \tilde{\mathbf{W}}_d \mathbf{J} \\ \sqrt{\beta} \tilde{\mathbf{W}}_m \end{bmatrix} \delta \boldsymbol{\sigma}_{\text{est}} = \begin{bmatrix} \tilde{\mathbf{W}}_d (\boldsymbol{\phi}_{\text{obs}} - \boldsymbol{\phi}_{\text{pred}}) \\ \sqrt{\beta} \tilde{\mathbf{W}}_m (\boldsymbol{\sigma}_{\text{est}} - \boldsymbol{\sigma}_{\text{ref}}) \end{bmatrix} \quad (10)$$

or, in a more compact form, as

$$\tilde{\mathbf{J}} \delta \boldsymbol{\sigma}_{\text{est}} = \tilde{\mathbf{b}}. \quad (11)$$

Assuming $\tilde{\mathbf{J}}$ and $\tilde{\mathbf{b}}$ have been constructed, the CGLS algorithm may be expressed by the following operations in equations 12–28. Equa-

tions 18–28 represent the inner iterations of the inversion algorithm.

$$\mathbf{r}_e = \tilde{\mathbf{b}} \quad (12)$$

$$\mathbf{s}_e = \tilde{\mathbf{J}}^T \mathbf{r}_e \quad (13)$$

$$\mathbf{p} = \mathbf{s}_e \quad (14)$$

$$\gamma = \mathbf{s}_e^T \mathbf{s}_e \quad (15)$$

$$N_0 = \gamma^{1/2} \quad (16)$$

$$\delta \boldsymbol{\sigma}_{\text{est}} = 0 \quad (17)$$

Start loop

$$\mathbf{q} = \tilde{\mathbf{J}} \mathbf{p} \quad (18)$$

$$\Delta = \mathbf{q}^T \mathbf{q} \quad (19)$$

$$\alpha = \gamma \Delta \quad (20)$$

$$\delta \boldsymbol{\sigma}_{\text{est}} = \delta \boldsymbol{\sigma}_{\text{est}} + \alpha \mathbf{p} \quad (21)$$

$$\mathbf{r}_e = \mathbf{r}_e - \alpha \mathbf{q} \quad (22)$$

$$\mathbf{s}_e = \tilde{\mathbf{J}}^T \mathbf{r}_e \quad (23)$$

$$N = (\mathbf{s}_e^T \mathbf{s}_e)^{1/2} \quad (24)$$

$$\gamma_1 = \gamma \quad (25)$$

$$\gamma = N^2 \quad (26)$$

$$b = \gamma / \gamma_1 \quad (27)$$

$$\mathbf{p} = \mathbf{s}_e + b \mathbf{p} \quad (28)$$

if convergence criteria satisfied then exit loop
end loop.

The CGLS or inner iteration convergence criteria referenced after equation 28 typically is based upon the magnitude of the decrease in N with respect to N_0 between iterations or until some user-specified maximum number of iterations is reached. Each of equations 12–28 requires minimal computational effort except for equations 13, 18, and 23, which represent matrix/vector multiplications involving $\tilde{\mathbf{J}}$ and $\tilde{\mathbf{J}}^T$. Thus, each of these operations (except for equations 13, 18, and 23) can be computed efficiently by a single processor. To address the effort required to compute equations 13, 18, and 23, these equations are computed in parallel.

The matrix $\tilde{\mathbf{J}}$ contains a dense upper matrix consisting of n_{obs} rows (i.e., the Jacobian matrix) augmented by a sparse lower matrix representing the model constraints. As discussed, the rows of the dense upper matrix (i.e., \mathbf{J}) are distributed evenly across slave nodes, and the lower sparse part is stored on the master node. Consequently, matrix/vector multiplications involving $\tilde{\mathbf{J}}$ are conducted in two parts. The dense part of the computation is computed in parallel on slave nodes and the sparse part is computed serially on the master node. This is accomplished by broadcasting the appropriate vector (i.e., \mathbf{p} or \mathbf{r}_e) from the master to the slave nodes. The slave nodes compute the portion of the matrix/vector multiplication corresponding to rows of the Jacobian matrix that each has computed and stored, and then they return the results to the master node. Thus, for equation 18, each slave multiplies \mathbf{p} by the rows for which it is responsible and returns the corresponding rows to the master node. The master node collects the rows from each slave node and assembles them to complete the dense part of the $\tilde{\mathbf{J}} \mathbf{p}$ multiplication.

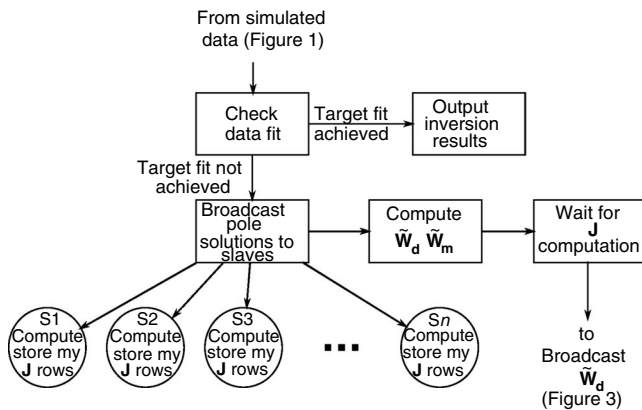


Figure 2. Generalized flow diagram demonstrating the Jacobian matrix computation for the inversion problem. The Jacobian matrix is divided evenly among slave nodes in terms of storage and computation, where each node stores and computes approximately the same number of rows. As the Jacobian computation proceeds, the master node computes the generalized data and model weighting matrices. Squares are master tasks; circles are slave tasks.

For equations 13 and 23 ($\tilde{\mathbf{J}}^T \mathbf{r}_e$), the slave nodes multiply the elements of \mathbf{r}_e corresponding to the columns of $\tilde{\mathbf{J}}^T$ for which each is responsible and then return the resulting vector to the master node. The master node sums the vectors from each slave to complete the dense part of the $\tilde{\mathbf{J}}^T \mathbf{r}_e$ multiplication. Note that the columns of $\tilde{\mathbf{J}}^T$ are the same as the rows of $\tilde{\mathbf{J}}$ stored by each slave node, so there is no difference in computational effort, reordering, or storage requirement for computing equations 13 and 23 compared to equation 18. The slave computations are performed by referencing the appropriate matrix and vector indices rather than reordering the matrices and vectors themselves for the matrix transpose operations. A detailed explanation of the parallel matrix vector multiplication operations is given in Appendix A.

After broadcasting the appropriate vector to the slave node, the master node executes the sparse part of the computation, gathers the dense part from the slaves, assembles the resulting vector (\mathbf{s}_e or \mathbf{q}), and proceeds with the CGLS algorithm.

A flow diagram of the parallel CGLS algorithm and the end of the outer iteration is shown in Figure 3. The algorithm begins by broadcasting the generalized data-weighting matrix $\tilde{\mathbf{W}}_d$ from the master node to the slave nodes, which then compute and store $\tilde{\mathbf{W}}_d \mathbf{J}$ for the rows of \mathbf{J} to which each is assigned. The master node then computes $\tilde{\mathbf{b}}$ and proceeds with the algorithm at equation 12, computing equations 13, 18, and 23 in parallel as described. When the parallel CGLS algorithm has converged, the master node updates σ_{est} and broadcasts the new estimate to the slaves. For our examples, we specify

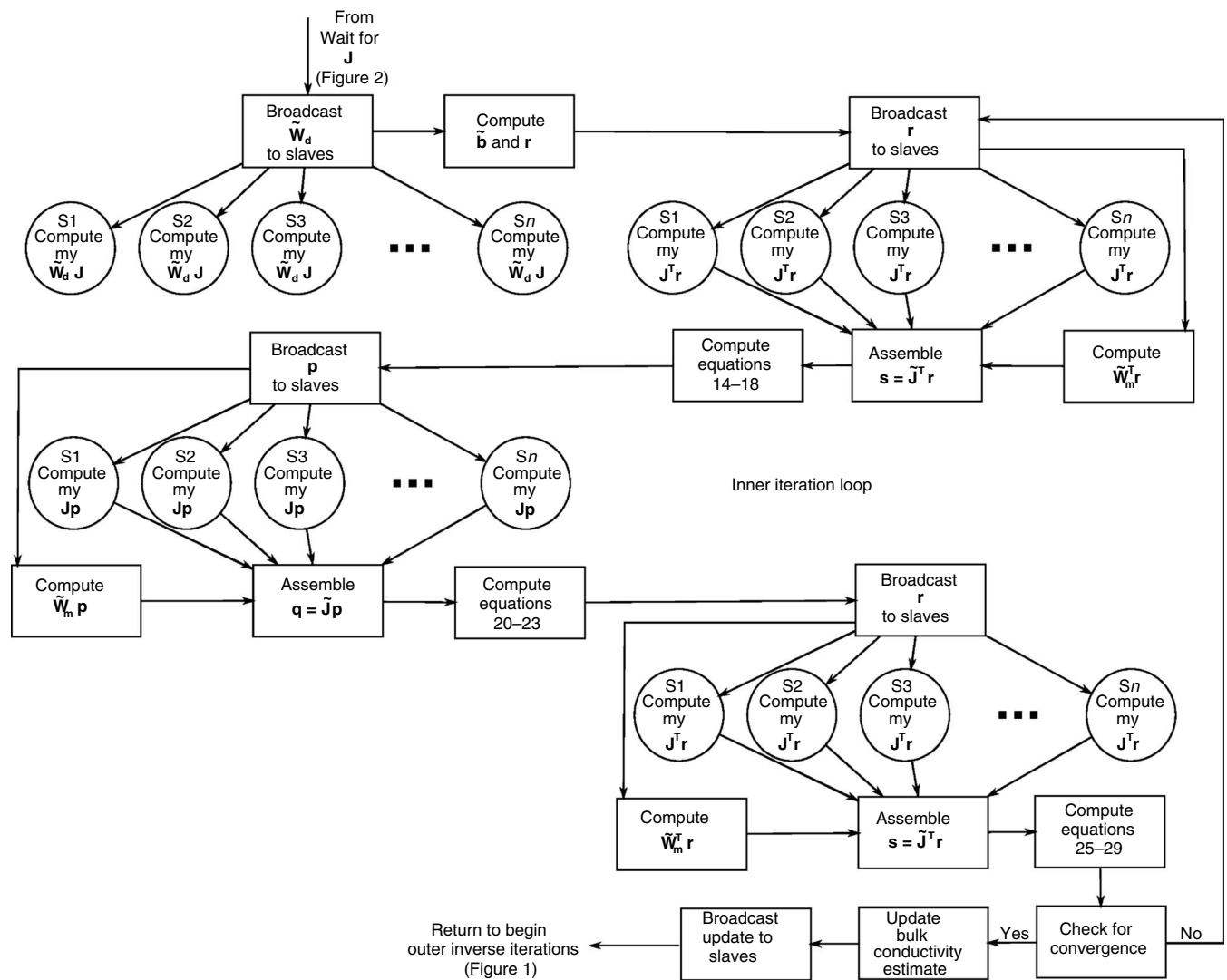


Figure 3. Generalized flow diagram demonstrating the parallel CGLS algorithm used to solve the inverse equations. Dense matrix-vector computations are distributed evenly among the slave nodes, but sparse matrix-vector, vector-vector, and lower-order operations are executed by the master node. Squares are master tasks; circles are slave tasks.

convergence when $n/n_0 = 0.1$ or 30 iterations are reached. This typically requires 15–30 inner iterations per outer iteration. Execution then returns to the outer-iteration starting point (Figure 1), and outer iterations continue until the resistivity data have been fit to within the user-specified criteria. If the maximum number of inner iterations is reached (i.e., the CGLS algorithm does not converge to the specified convergence criteria), the update vector still provides a decrease in the objective function because an exact solution to the linearized system is not required to produce a suitable update vector (Rodi and Mackie, 2001). It may or may not be possible to decrease the number of outer iterations required for convergence by decreasing n/n_0 , but it will increase the number of inner iterations required per outer iteration.

IP inversion and other considerations

The discussion thus far has focused on inversion of resistivity data. Inversion of IP data is a linear inversion executed after the conductivity distribution has been estimated and thus requires one Jacobian-matrix computation at the end of the resistivity inversion. The IP inversion algorithm iterates to find the optimal regularization parameter β without recomputing \mathbf{J} at each iteration. This is done by starting with a relatively large β value, computing the inverse solution, simulating the data, and checking the data misfit. If the data misfit is larger than that suggested by the data noise, the β value is reduced and the process repeats until the data are appropriately fit. Details of the IP inversion are found in Li and Oldenburg (2000).

We have omitted several other topics of interest concerning the parallel code because these topics are published elsewhere and/or exceed the scope of this manuscript. A few of these topics include advanced mesh-generation capabilities (Si, 2003), singularity removal

(Lowry et al., 1989), generalized model- and data-weighting constraints by iteratively reweighted least squares (Farquharson, 2008), the implementation of a line search for highly nonlinear problems, various options for directional model weighting and zonation, borehole constraints, and geostatistical constraints (Johnson et al., 2007).

EXAMPLE FIELD APPLICATIONS

In this section, we demonstrate the use of the parallel code for three field efforts: (1) resistivity characterization of the Hanford 300 Area Integrated Field Research Challenge site in Hanford, Washington, (2) resistivity characterization of a volcanic island in the southern Tyrrhenian Sea in Italy, and (3) resistivity and IP monitoring of biostimulation at a Superfund site in Brandywine, Maryland. Each inversion was computed on the Idaho National Laboratory Ice Storm cluster. A summary of the resources required for each inversion is shown in Table 1.

3D characterization of the Hanford 300 Area IFRC

The Hanford 300 Area Integrated Field Research Challenge (IFRC) is a Department of Energy — Office of Science, Environmental Remedial Science Division-sponsored multiyear, multi-institution, multiresearcher field research project. The objective of this project is to develop a better field-based understanding of the behavior of a uranium plume at the Hanford 300 Area. This better understanding requires integrating and using data from multiple geoscience disciplines (geophysics, hydrogeology, and geochemistry, to name a few). As part of this project, a well field was installed, consisting of 28 wells arranged in a triangle of approximately 90 m (300 ft) per side. Each well is approximately 18 m (60 ft) deep and has 30 electrodes installed at 0.61-m (2-ft) spacing between electrodes. All electrodes are connected to a central outbuilding from which data can be collected using multichannel, multielectrode resistivity instrumentation.

The locations of the IFRC boreholes and electrodes are shown in Figure 4. The monitoring/characterization array consists of 28 boreholes with 30 electrodes each for a total of 840 electrodes. Electrical resistivity tomography (ERT) and IP data were collected sequentially using a multichannel, 240-electrode measurement system. For each survey, a group of eight wells was used with two duplicate wells between subsequent surveys to ensure data continuity. Full reciprocal measurements were collected for noise analysis and data filtering. Data with reciprocal errors greater than 10% were removed from the inverted data set. After filtering, 149,706 resistivity measurements were included in the resistivity inversion.

The full computational mesh for the resistivity characterization inversion is shown in Figure 5. The mesh is extended beyond the IFRC area in the center of the mesh to reduce boundary effects. In addition, the mesh is refined near electrode

Table 1. Inverse problem sizes and computational resources required for each example.

Resource	IFRC	La Fossa Cone	Brandywine
Number of mesh elements	640,046	398,208	142,447
Number of electrodes	840	958	105
Number of data points	149,706	9525	10,039 Res. 2354 IP
Number of processors	421	480	106
Memory per processor (forward/inverse)	116 Mb/676 Mb	72 Mb/378 Mb	38 Mb/89 Mb
Master memory (forward/inverse)	547 Mb/1050 Mb	613 Mb/998 Mb	133 Mb/182 Mb
Run time to convergence (approximate)	5 hours	ℓ_1 -norm, 3 hours ℓ_2 -norm, 2 hours	2 hours/5 min. ^a

^aApproximately 2 hours for background inversion and approximately 5 minutes for time-lapse inversion using background as the starting model.

locations to accommodate large potential gradients near current-source and -sink electrodes. The refinement is accomplished by placing several mesh nodes near electrode nodes and instructing the mesh-generation software to construct a high-quality tetrahedral mesh (Si, 2003). In total, the mesh contains 106,366 finite-element nodes and 640,046 tetrahedral elements. Each element is used as an inversion parameter in this case.

The data were inverted on 421 nodes (one master and 420 slave nodes) using a homogeneous starting model, requiring approximately 5 hours of computation time on the Idaho National Laboratory Ice Storm cluster. The regularization constraints are based upon a weighted directional derivative of the log-conductivity distribution between neighboring cells as described in Appendix B. These constraints seek to minimize the directional derivative between element centers with a horizontal-to-vertical weighting of 10:1 to foster the layered structure expected at the site. The inverse solution is shown in Figure 6. Increasing levels of resistivity have been removed from the images to show the internal structure of the estimated resistivity.

A histogram of the data fit is shown in Figure 7. Here and in the next two examples, the data-fit errors are defined as the residual error (i.e., predicted resistance minus observed resistance) divided by the observed resistance, expressed as a percentage.

Characterization of Vulcano Island

Vulcano Island is a small volcanic island (3×7 km) located at the southernmost margin of the Aeolian Islands in the southern Tyrrhenian Sea in Italy ($38^\circ 24'N$, $14^\circ 58'E$). La Fossa cone occupies a 3-km-wide caldera at the northwest end of the island. This edifice constitutes the location of most of the historical eruptions of Vulcano Island. Because of the relatively modest dimensions of this edifice and its strong hydrothermal activity, the volcano is an excellent natural laboratory to test the ability of 3D resistivity to image the conductive substructure of an active volcano. The description of the data acquisition is described in Revil et al. (2008) and is not detailed here except to note that the data consist of nine 2D surface-resistivity lines positioned over the cone at varying azimuths to optimize coverage.

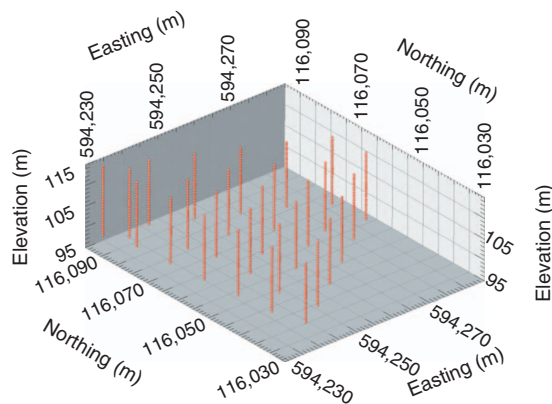


Figure 4. Location of boreholes and borehole electrodes (red dots) at the IFRC area. The ERT/IP characterization/monitoring array consists of 28 boreholes with 30 electrodes each for a total of 840 electrodes.

In contrast to the other examples, inversion of the Vulcano Island data requires modeling surface topography to compute subsurface potentials accurately. Figure 8a shows the surface topography for the computational mesh. To construct the mesh, known topography was used in the vicinity of the volcano and survey area. Away from the volcano where elevations were unknown, topography was inferred using the nearest known point. Figure 8b shows the surface of the mesh with refinement about the volcano. Figure 8c shows a close-up of the volcano and mesh. Here, the mesh is refined about the 2D surface electrode lines to accommodate large potentials near current-source and -sink electrodes. In all, the computational mesh consists of 958 electrode locations, 74,459 finite-element nodes, and 398,208 tetrahedral elements. The relatively large number of elements with respect to the number of data points in this case was required to model the surface topography accurately. Singularity removal was not used in this inversion.

The full inverted data set consists of 9525 2D resistivity measurements and the inverse computations were conducted on 480 nodes (one master and 479 slave nodes). The data were inverted using the model constraints described in Appendix B, with equal weighting in the horizontal and vertical directions. For comparison, the data were inverted with ℓ_1 - and ℓ_2 -norm weighting on the model (ℓ_2 -norm on the data in both cases) using iteratively reweighted least squares for the ℓ_1 -norm solution under the formulation described by Farquharson (2008). Starting with the same homogeneous model and regularization weighting, the ℓ_1 - and ℓ_2 -norm solutions required approximately 3 and 2 hours, respectively, to converge. Convergence

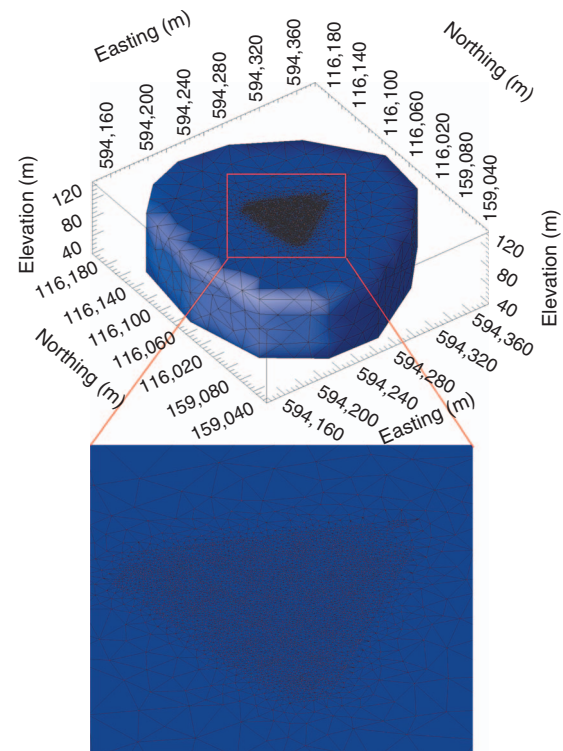


Figure 5. IFRC ERT/IP finite-element characterization and monitoring computational mesh. The mesh is composed of tetrahedral elements containing 106,366 nodes and 640,046 elements. IFRC borehole electrodes are located in the center refined mesh area.

curves showing the total, data, and model parts of the objective function per iteration are shown in Figure 9. The regularization parameter β also is shown for comparison. For this inversion, β is reduced

by one-half of its previous value if the total objective function decreases by less than 5% during the previous outer iteration.

The corresponding solutions are shown in Figure 10, with vertical cross sections progressing toward the center of the volcano to illustrate the estimated subsurface conductivity structure. In comparison to the ℓ_2 -norm, the ℓ_1 -norm weighting on the model eases constraint errors at sharp boundaries and thus helps to foster sharp interfaces in the inverse solution as suggested in Figure 10. Because there are no crossline measurements, the surface-resistivity structure in many areas is uncertain and may contain artifacts. The subsurface structure is better resolved; it shows a high-conductivity area below the center of the edifice where fumarolic activity can be observed. This high-conductivity subvolume can be associated with the high temperature of the hydrothermal fluids ($> 200^\circ\text{C}$), the high salinity of the brines, and the presence of clays resulting from alteration.

A histogram of the data fit for the ℓ_2 -norm inversion is shown in Figure 11 (the ℓ_1 -norm histogram is similar). In this case, reciprocal measurements were not available to help identify and remove erroneous data from the inversion and estimate the appropriate data fit. Thus, the appropriate data fit was estimated. With respect to a normal distribution, the error histogram displays long tails and larger errors. The high error data are likely from changing subsurface conductivity between 2D surveys as a result of the dynamic processes occurring within the volcano. These higher error data might have been removed had reciprocal or repeat measurements been available.

Monitoring bioremediation, Brandywine field site

The former Brandywine Defense Reutilization and Marketing Office (DRMO) Yard is an inactive facility administratively controlled by Andrews Air Force Base (AFB). The 3.3-ha (8-acre) site is approximately 13 km (8 miles) south-southeast of Andrews AFB in the town of Brandywine, Maryland. While in operation from 1943–1987, the Brandywine DRMO yard was used for temporary storage of scrap materials and hazardous waste generated from various Department of Defense (DOD) facilities in the region. The primary groundwater contaminant at the site is trichloroethylene (TCE), a cleaning solvent.

This site is representative of many other DOD sites with organic contaminants. For such contaminants, enhanced bioremediation is a preferred restorative practice. However, a primary challenge for

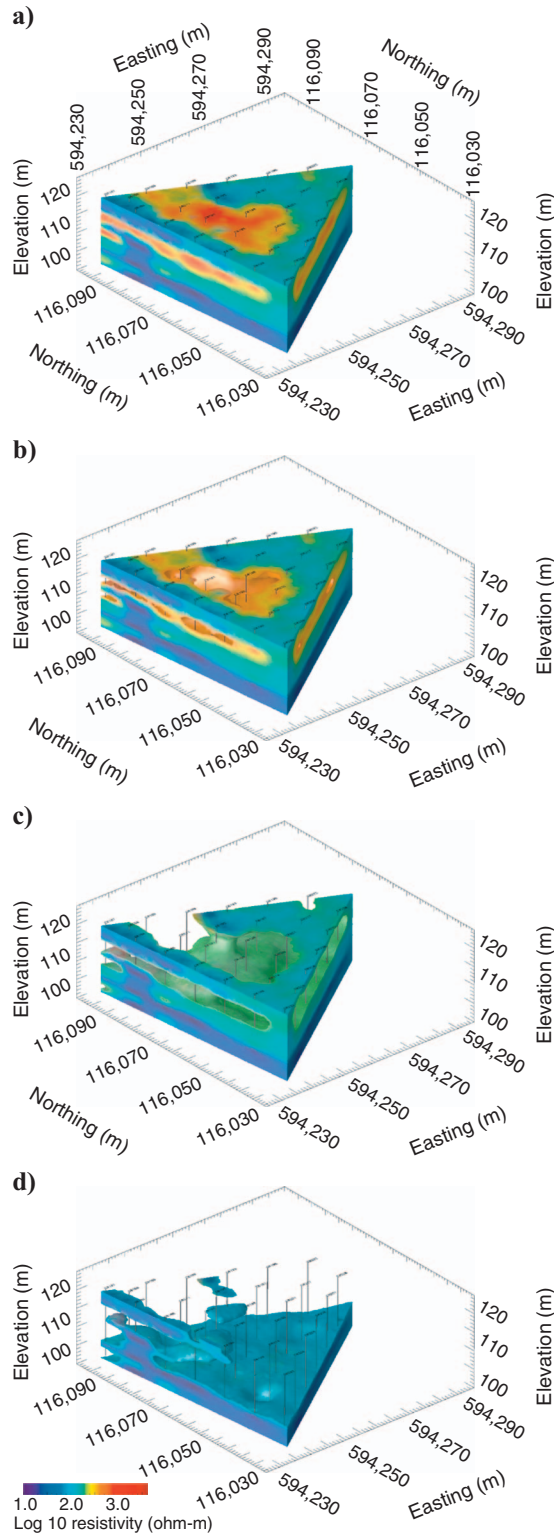


Figure 6. IFRC ERT characterization inverse solutions. (a) Full solution in the IFRC area. (b–d) Solutions showing all resistivities at stated isosurface and below: (b) 400 ohm-m; (c) 200 ohm-m; (d) 100 ohm-m.

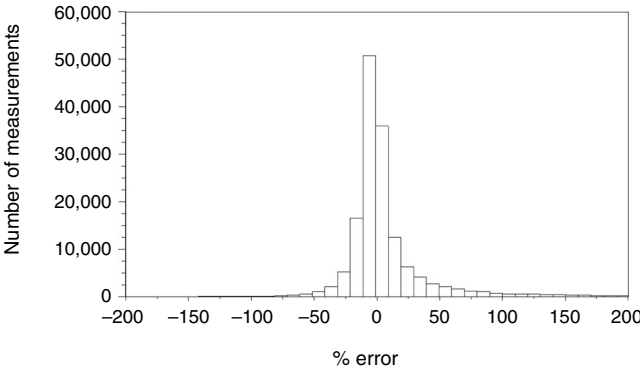


Figure 7. Histogram of resistance-data misfits for IFRC inversion shown in Figure 6.

bioremediation is the verification of amendment transport and the resulting geochemical processes (e.g., Lane et al., 2006). The objective of the ongoing research effort is to demonstrate the feasibility of using a geophysical monitoring system to provide information on the spatiotemporal behavior of biostimulants injected into the subsurface. For this purpose, an autonomous electrical geophysical monitoring system has been operating on the site since March 2008 and has been monitoring the injection and subsequent movement of a proprietary mixture of lactates, fatty acids, and a phosphate buffer. Data are collected continuously, with about 200 3D data sets collected as of August 2009. Our work focuses on monitoring two injection points out of more than 1000 planned injection locations.

The Brandywine resistivity/IP monitoring system electrode configuration is shown in Figure 12a. The system consists of seven boreholes, each with 15 vertically spaced electrodes, and three surface electrode lines for 165 total electrodes. Only the borehole electrode data were used (i.e., 105 electrodes) in the monitoring inversion shown here. The modeling/inversion mesh (not shown) consists of 26,189 finite-element nodes and 142,447 tetrahedral elements with refinement about electrodes as in the previous examples.

Preinjection (or background) data were collected to establish the estimated background bulk conductivity and chargeability distributions. These data consist of 2D (i.e., hole to hole) and 3D measurements, including full reciprocals. After noise analysis, 10,039 resistivity measurements and 2354 IP measurements were used in the background characterization inversions.

The background data were inverted on 106 nodes (one master and 105 slave) and required approximately 2 hours of computation time. The estimated preinjection bulk conductivity and chargeability are shown in Figure 12b and c, and the corresponding data misfit histograms are shown in Figure 12d and e. Figure 12b and c shows the higher bulk conductivity and moderate chargeability associated with a fine-grained deep confining unit observed at the site. This confining unit is important when analyzing the behavior of the bioamendment-induced changes in bulk conductivity and chargeability.

Bioamendment injection in the rightmost injection well was executed on 11 March 2008. After injection, resistivity and IP data were collected continuously with a full data sequence (i.e., one time-lapse data set) requiring approximately 1.5 days. The time-lapse data sets were reduced to the same measurements as the background data set for consistency. Each data set was inverted using the background solution as the starting model and using the same data-weighting and regularization constraints as those used to produce the background model. After calculating each time-lapse inverse solution, the background solution was subtracted to produce the estimated change in bulk conductivity and chargeability with time due to the biogeochemical changes initiated by introducing the bioamendment. Each time-lapse inversion was executed on 106 processors and required approximately 5 minutes for the combined resistivity/IP inversion.

The monthly change in bulk conductivity from March 2008 to January 2009 is shown in Figure 13. The time sequence shows a high-conductivity plume emanating from the injection well and sinking with time to the lower bounding unit. The bulk-conductivity change peaks 5–6 months after injection and then begins to return to background conditions toward January 2009. The time-lapse change in chargeability is shown in Figure 14. The change in chargeability generally shows the same behavior as the change in bulk conductivity but has remained more persistent over time.

SCALABILITY

Because of their modular nature and subsequent relatively low cost/performance ratio, most parallel environments use a distributed memory structure. In comparison to shared-memory parallel computing systems (each processor has access to all memory), distributed-memory systems (each processor has dedicated memory) de-

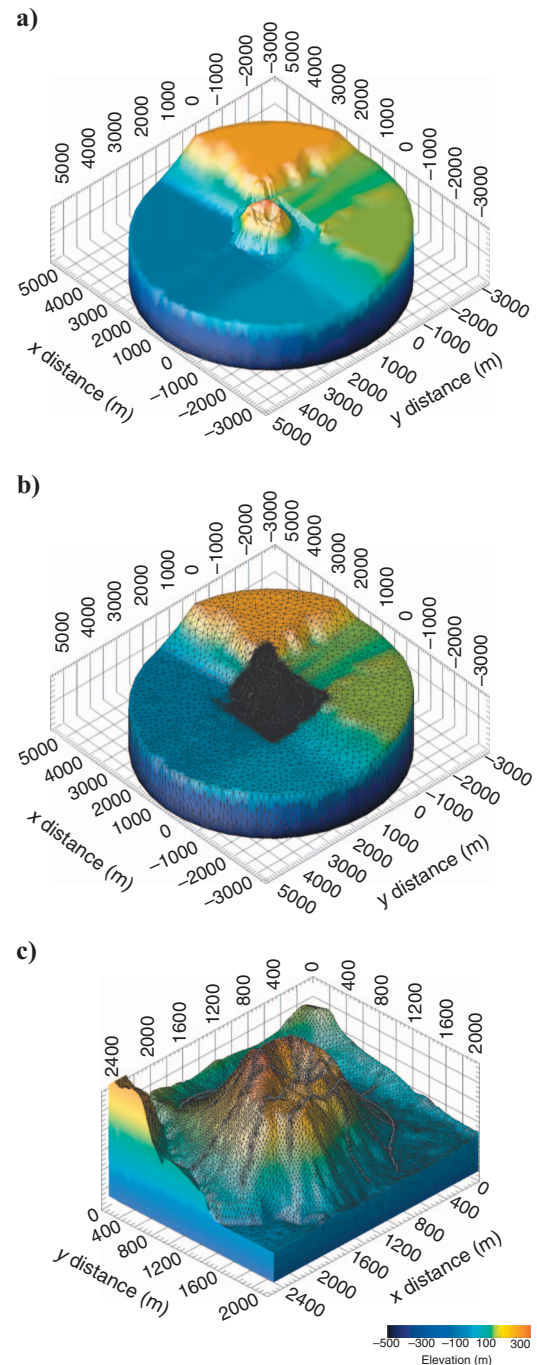


Figure 8. (a) Surface topography and boundaries of computational mesh. (b) Computational mesh showing refined the volcano area. (c) Computational mesh in the volcano vicinity. Refined areas show locations of 2D electrode arrays. Mesh consists of 958 electrode locations, 74,459 nodes, and 398,208 elements. Shown at 3:1 vertical/horizontal scaling.

grade performance when information must be passed between processors. Because processors do not share memory, they must transfer information to other processors over a network interface that increases run time. In general, the ratio of message passing to computation time increases with the number of processors used. The scalability of a program is a measure of the run time required with respect to the number of nodes used. For instance, perfect scalability allows a twofold decrease in computation time, resulting from a two-fold increase in the number of nodes. Near-perfect scaling sometimes is possible in so-called embarrassingly parallel problems, where no communication is required between processors.

Figure 15 shows the scalability of the forward code when computing 512 pole solutions on the Hanford IFRC mesh. In this case, the largest number of slave nodes possible is 512 (one for each pole solution). Figure 15a shows the total run time and the pure computation time (total run time minus message passing time) with respect to the number of slave nodes. Figure 15b shows the percentage of total run time used in message passing (overhead time) with respect to the number of slave nodes. For the forward problem, the overhead time includes the time required for the master node to broadcast run information to the slave nodes plus the time required for the slave nodes to transmit the pole solutions back to the master node as shown in Figure 1. The percent overhead time increases to about 45% of the total run time at 512 slave nodes, and the total computing time decreases from 30,880 s with two slave nodes to 219 s with 512 slave nodes.

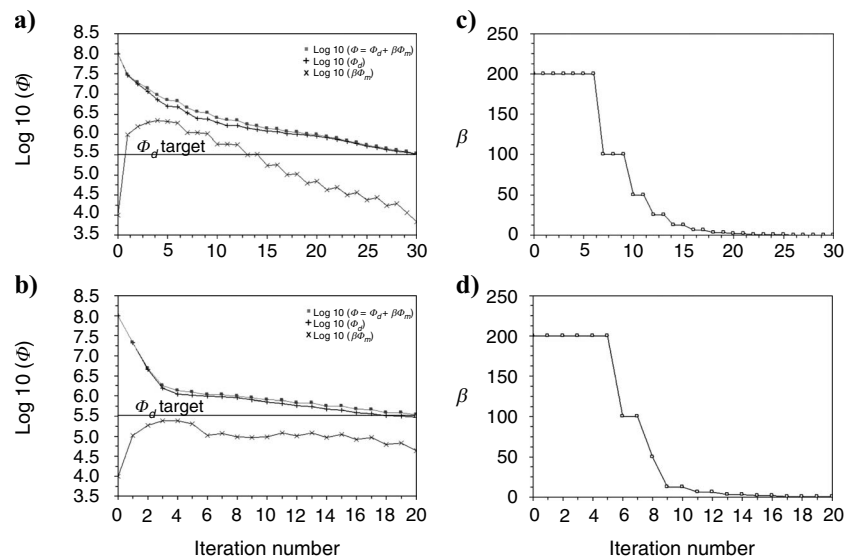
Figure 16 shows the corresponding scalability of the inverse code executed for one time-lapse solution of the Brandywine data. Here,

there are 105 pole solutions, so a maximum of 105 slave nodes. With respect to the forward problem, the overhead time required for the inversion problem increases considerably. This increase in overhead time occurs because of the master-to-slave pole solution broadcast shown in Figure 2 and particularly because of the broadcast and gather operations required in the parallel CGLS algorithm shown in Figure 3. Nevertheless, total computing time decreases significantly, requiring 17,593 s for a single processor and 461 s for 105 processors.

DISCUSSION

Although standard 2D and moderate 3D data sets can be inverted readily using commercially available software on standard desktops, such inversion is not feasible for the large 3D and 4D data sets that are being collected routinely. Memory limitations on standard desktop computers may require users to rely on limited data, coarsen the inversion mesh, or both, each of which may degrade the inverse solution compared to what would otherwise be possible. Günther et al. (2006) demonstrate significantly improved memory efficiency by neglecting small sensitivities and converting the resulting Jacobian matrix to sparse form, allowing larger inversion problems on serial computers. Although this is helpful and would decrease memory requirements in a parallel implementation, it is ultimately unscalable in serial mode and will reach a useful limit for larger problems. In addition, performance of the inversion algorithm degrades as more elements are removed from the Jacobian matrix.

Figure 9. Convergence curves showing (a, b) the objective function component values and (c, d) regularization-parameter values for La Fossa cone inversion. Views (a) and (c) apply to the ℓ_1 -norm inversion; (b) and (d) apply to the ℓ_2 -norm inversion. Each inversion started with the same homogeneous model and regularization parameter value (i.e., 200). The convergence parameters were set such that the regularization parameter decreased by 0.5 times the previous value if the decrease in total objective function was less than 5% of the previous value.



In cases where memory resources are adequate, run times can quickly become excessive for large inversion problems. Excessive run time reduces the utility of time-lapse inversion and inhibits apractitioner's ability to test inversion options and noise assumptions, which often are critical for optimizing the inverse solution. In general, the full utility of electrical geophysical characterization and monitoring is underutilized because standard computing environments and codes are unable to fully process large data sets often produced by modern multichannel and multielectrode resistivity/IP data-collection systems.

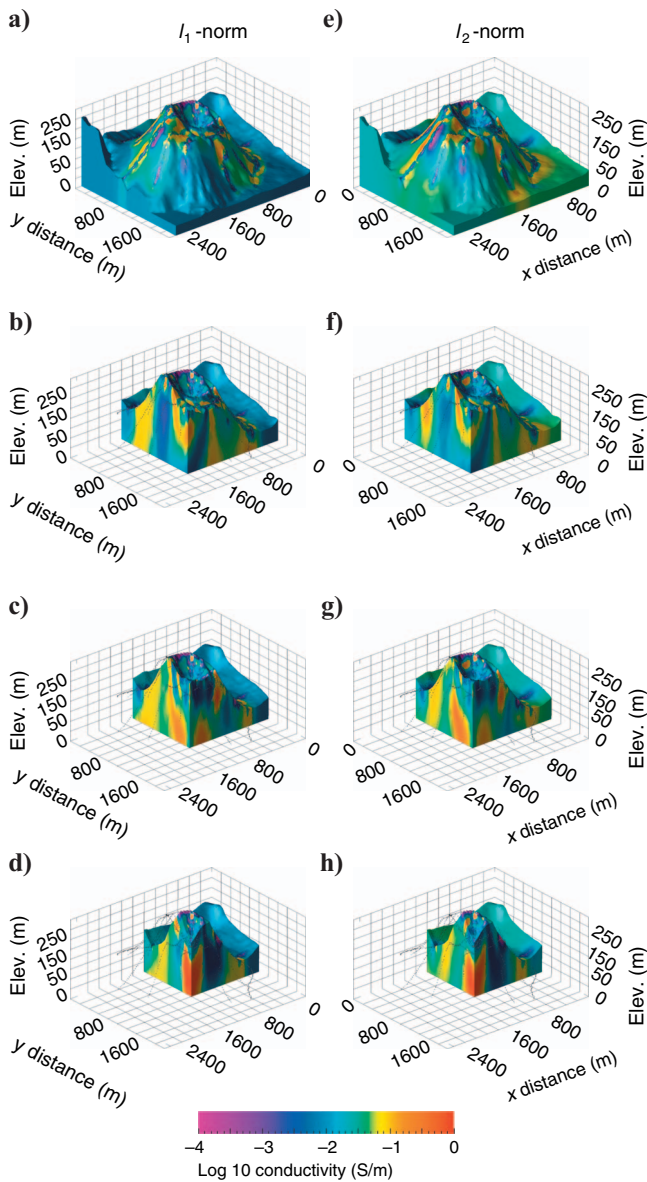


Figure 10. Inverse (a–d) ℓ_1 - and (e–h) ℓ_2 -norm solutions, showing interior bulk conductivity structure at vertical cross sections moving toward the center of the volcano from top to bottom. Electrode locations are shown as black dots. Images are shown at 3:1 vertical/horizontal scaling.

Our three examples demonstrate the utility of and need for parallel resistivity/IP inversion. The Hanford IFRC example demonstrates a large data set containing information about the bulk-conductivity structure over a large area with high resolution, given the scale of the survey. Effective extraction of this information allows scientists to better understand the distribution of hydrogeologic variations at the site, which presumably are correlated to bulk-conductivity variations. This information aids in the development of hypothetical flow and transport models used to study uranium transport and fate at the site.

For the La Fossa data inversion, parallel modeling and inversion will make it feasible to conduct 4D resistivity monitoring of active volcanoes with high spatiotemporal resolution at the kilometer scale (see Finizola et al. [2009] for a recent field example using 2D data sets). Coupled with self-potential, CO_2 , and temperature measurements, resistivity/IP monitoring data could be used to better understand the behavior of volcanoes and other hydrothermal systems in near real time. In addition, real-time monitoring of volcanic activity could provide actionable information for hazard management and could support decisions regarding evacuation.

The Brandywine example demonstrates the possibility of achieving high-resolution and near-real-time images of transient conductivity and/or chargeability variations caused by natural and engineered hydrologic processes, enabling new scientific insights and improved decision-making. Although past geophysical efforts have provided after-the-fact insight into processes such as aquifer-storage recovery injection (Singha et al., 2007) and bioremediation (Lane et al., 2006), ongoing technology and software developments may soon allow geophysicists to provide information to decision-makers in an actionable timeframe — while injections are being performed and field procedures can be modified. In the case of biostimulation, as at the Brandywine site, such information could provide remedial operation managers the means to verify, immediately after injection, that amendments reach the target zones and to modify injection rates or locations if these objectives are not met.

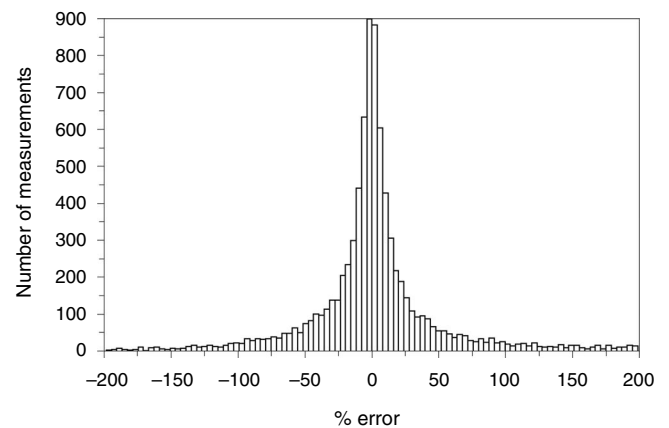


Figure 11. Histogram of resistance data-misfit errors for ℓ_2 -norm inverse solution shown in Figure 10.

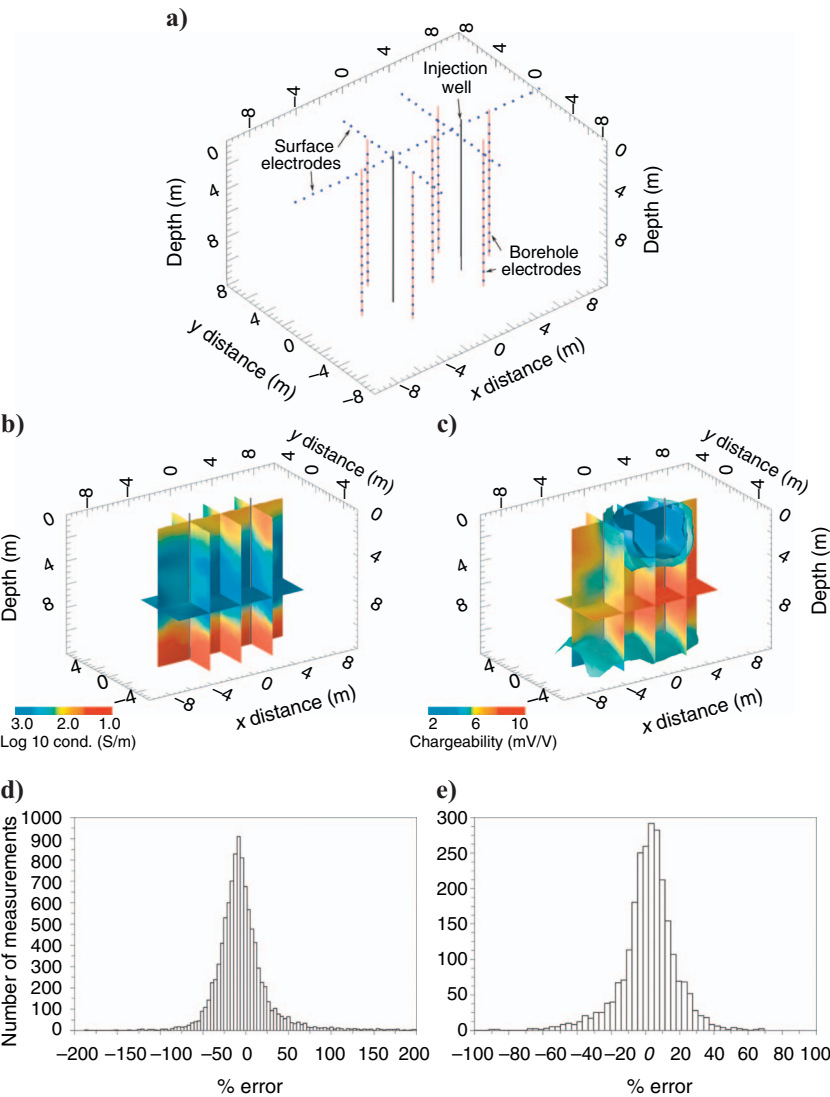


Figure 12. (a) Diagram of Brandywine surface and borehole electrode configuration, including bioamendment injection well. (b) Preinjection estimated background bulk-conductivity distribution. (c) Preinjection estimated background chargeability distribution. Isosurfaces shown are at chargeability values of 0.004 and 0.005 V/V. (d) Resistivity misfit-error histogram corresponding to conductivity estimate shown in (b). (e) Apparent chargeability misfit histogram for chargeability estimate shown in (c). Histograms in (d) and (e) are typical of those for each time-lapse inversion.

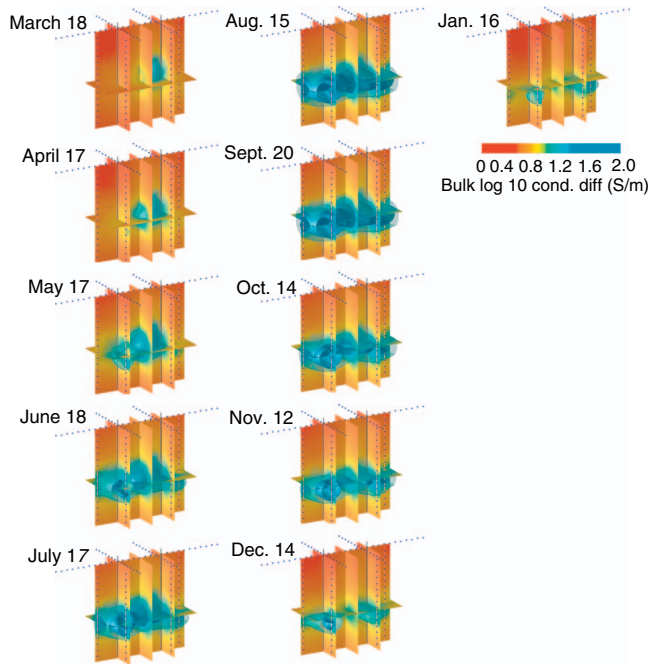


Figure 13. Monthly time-lapse estimates of change in bulk conductivity from background caused by bioamendment injection and associate biogeochemical activity.

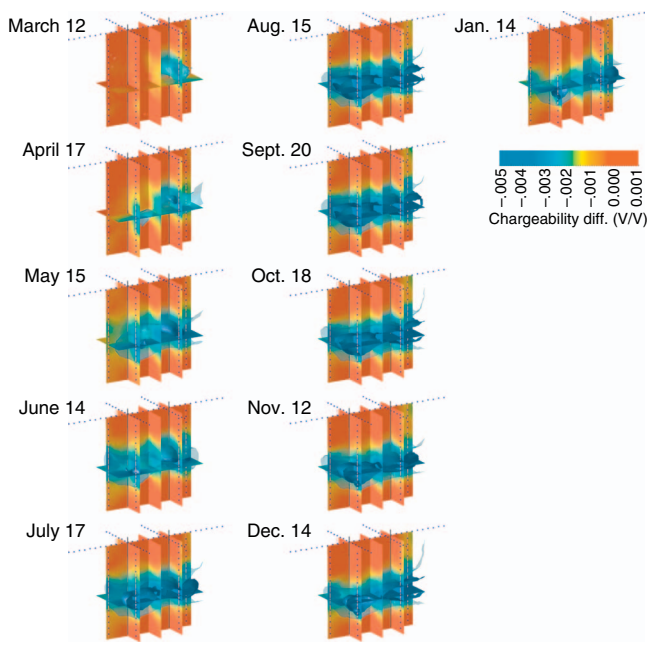


Figure 14. Monthly time-lapse estimates of change in chargeability from background caused by bioamendment injection and associate biogeochemical activity.

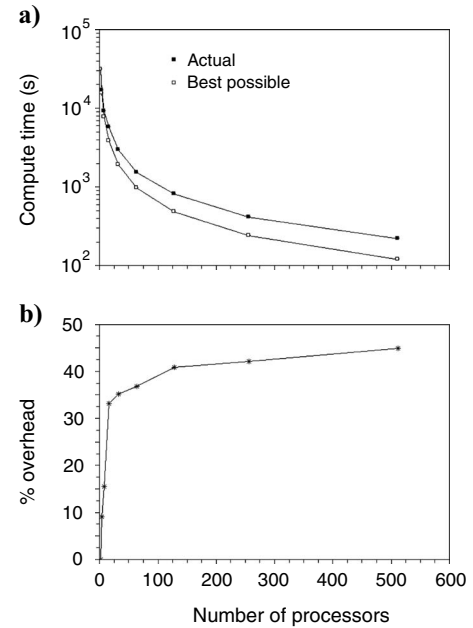


Figure 15. (a) Comparison of run time versus number of processors for the forward-resistivity/IP code simulating 512 pole solutions with the Hanford 300 IFRC mesh. The open dots indicate perfect scaling, or the pure computation time (i.e., no message-passing time). The solid dots indicate total run time, including pure computation and message-passing time. (b) Comparison of the percentage of message-passing time (i.e., overhead time) required with respect to total run time for the forward problem in (a).

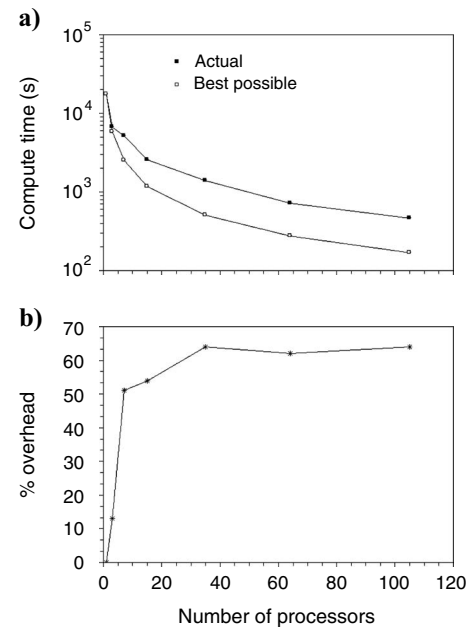


Figure 16. (a) Comparison of run time versus number of processors for the inverse-resistivity/IP code estimating one time-lapse solution on the Brandywine mesh. The open dots indicate perfect scaling, or the pure computation time (i.e., no message-passing time). The solid dots indicate total run time, including pure computation and message-passing time. (b) Comparison of the percentage of message-passing time (i.e., overhead time) required with respect to total run time for the forward problem in (a). The increase in overhead with respect to the forward problem is primarily because of the message-passing required in the CGLS algorithm.

CONCLUSION

We have presented a parallel modeling and inversion approach for analyzing time-domain resistivity and IP data, and we have applied the code in three hydrogeophysical applications. The capability to model and invert electrical geophysical data in a parallel computing environment allows full utilization of current resistivity/IP data-collection capabilities and significantly improves the capability to characterize the electrical geophysical properties of the subsurface and to monitor spatiotemporal variations in electrical geophysical properties caused by evolving subsurface processes. We anticipate that the continued use of parallel inversion will improve understanding of the links between subsurface properties and associated electrical geophysical signatures, giving electrical geophysics an increasing role in solving subsurface problems and understanding subsurface behavior.

ACKNOWLEDGMENTS

We thank Partha Routh for helpful discussions and advice concerning the serial version of the code presented. We acknowledge and appreciate helpful reviews by Colin Farquharson, Adam Pidlisecky, Rory Henderson, and an anonymous reviewer. This research was supported in part by the Environmental Remediation Sciences Program (ERSP), Office of Biological & Environmental Research (OBER), U. S. Department of Energy (DOE), as part of the Hanford 300 Area Integrated Field Research Challenge Project. Brandywine characterization and monitoring is supported by grant ER-0717 from the Environmental Security Technology Certification Program (ESTCP) and by the USGS Toxic Substances Hydrology Program. Financial support for the work performed at Vulcano Island was provided by INSU-CNRS and the Laboratoire GéoSciences Réunion in France, the Instituto di Metodologie per l'Analisi Ambientale del CNR, the Istituto Nazionale di Geofisica e Vulcanologia, and the Dipartimento per la Protezione Civile in Italy (Project V3.5 Vulcano, 2005–2007). This work is supported by the DOE, Office of Science, under DOE Idaho Operations Office contract DE-AC07-05ID14517.

APPENDIX A

EXPLANATION OF PARALLEL MATRIX/VECTOR MULTIPLICATION OPERATIONS

Let \mathbf{Jp} be the dense part of the $\tilde{\mathbf{Jp}}$ multiplication shown in equation 18, and let k, i represent row i of \mathbf{J} that is stored on slave k . Let $Jp_{k,i}$ be element i of \mathbf{Jp} , which is computed by slave k . Once the master node has broadcast \mathbf{p} to each slave, $Jp_{k,i}$ is computed by slave k as

$$Jp_{k,i} = \sum_{j=1}^m J_{i,j} p_j. \quad (\text{A-1})$$

After each slave computes its elements of $Jp_{k,i}$, it returns the results to the master, which collects the elements from each slave and assembles them in the correct order to complete the \mathbf{Jp} computation.

To illustrate the parallel part of the $\tilde{\mathbf{J}}^T \mathbf{r}_e$ computation shown in equations 13 and 23, let $\mathbf{J}^T \mathbf{r}_e$ represent the dense part of $\tilde{\mathbf{J}}^T \mathbf{r}_e$. Let k_i and k_e represent the starting and ending rows of \mathbf{J} , which are stored by slave k . Let $(\mathbf{J}^T \mathbf{r}_e)_k$ represent a vector computed by slave k , and let $(\mathbf{J}^T \mathbf{r}_e)_{k,j}$ be the j th element of that vector. Then

$$(\mathbf{J}^T \mathbf{r}_e)_{k,j} = \sum_{i=k_s}^{k_e} J_{i,j} r_{e,i}, \quad (\text{A-2})$$

where $r_{e,i}$ is element i of vector \mathbf{r}_e . After receiving \mathbf{r}_e from the master node, each slave computes equation A-2 and returns $(\mathbf{J}^T \mathbf{r}_e)_k$ to the master node. The computation is then completed by the master node as

$$\mathbf{J}^T \mathbf{r}_e = \sum_{k=1}^{nslaves} (\mathbf{J}^T \mathbf{r}_e)_k, \quad (\text{A-3})$$

where $nslaves$ is the number of slave processors.

APPENDIX B

WEIGHTED DIRECTIONAL DERIVATIVE OF LOG-CONDUCTIVITY DISTRIBUTION BETWEEN NEIGHBORING CELLS

The regularization constraints specified in the examples seek to minimize the directional derivative of the conductivity distribution between adjacent elements, weighted according to the relative components of the derivative in each direction. The regularization constraints may be specified mathematically as

$$(\nabla \sigma(\mathbf{r}) \cdot \mathbf{v})(\mathbf{W} \cdot \hat{\mathbf{v}}) = 0, \quad (\text{B-1})$$

where \mathbf{r} is a position vector and $\sigma(\mathbf{r})$ represents the conductivity distribution. The vector \mathbf{v} represents some vector such that the first term of equation A-1 represents the first spatial derivative of σ in the direction of \mathbf{v} . The vector $\hat{\mathbf{v}}$ is a unit vector in the direction of \mathbf{v} . The vector $\mathbf{W} = W_x \mathbf{i} + W_y \mathbf{j} + W_z \mathbf{k}$ specifies the relative weighting in the x -(W_x), y -(W_y), and z -(W_z) directions, so that the second term on the right-hand side of equation B-1 scales the derivative according to \mathbf{W} .

For example, if \mathbf{r}_i is the position of the center of element i and \mathbf{r}_j is the position of the center of an element adjacent to element i , then let $\mathbf{v}_{ij} = \mathbf{r}_j - \mathbf{r}_i$ be the vector pointing from the center of element i to the center of element j . In this case, equation B-1 can be approximated as

$$(\nabla \sigma(\mathbf{r}_i) \cdot \mathbf{v}_{ij})(\mathbf{W} \cdot \hat{\mathbf{v}}_{ij}) \approx \left(\frac{\sigma_j - \sigma_i}{\|\mathbf{v}_{ij}\|} \right) (\mathbf{W}_x \hat{v}_{x,ij} + \mathbf{W}_y \hat{v}_{y,ij} + \mathbf{W}_z \hat{v}_{z,ij}) = 0, \quad (\text{B-2})$$

where $v_{x,ij}$, $v_{y,ij}$, and $v_{z,ij}$ are the components of the unit vector in the \mathbf{v}_{ij} direction, i.e.,

$$\frac{\mathbf{v}_{ij}}{\|\mathbf{v}_{ij}\|} = v_{x,ij} \mathbf{i} + v_{y,ij} \mathbf{j} + v_{z,ij} \mathbf{k}.$$

Equation B-2 is represented in matrix notation for all elements as

$$\mathbf{W}_m (\sigma - \sigma_{\text{ref}}) = 0, \quad (\text{B-3})$$

where row l and column j of \mathbf{W}_m is given by

$$\mathbf{W}_{m,l,j} = \frac{1}{\|\mathbf{r}_{ij}\|} (\mathbf{W}_x r_{x,ij} + \mathbf{W}_y r_{y,ij} + \mathbf{W}_z r_{z,ij}) \quad (\text{B-4})$$

and

$$\mathbf{W}_{\mathbf{m},1,i} = -\frac{1}{\|\mathbf{r}_{ij}\|}(\mathbf{W}_{\mathbf{x}}r_{x,ij} + \mathbf{W}_{\mathbf{y}}r_{y,ij} + \mathbf{W}_{\mathbf{z}}r_{z,ij}). \quad (\text{B-5})$$

Note in equation B-3 the reference model is included for generality.

REFERENCES

- Daily, W., 1984, Underground oil-shale retort monitoring using geotomography: *Geophysics*, **49**, 1701–1707.
- Dey, A., and H. F. Morrison, 1979, Resistivity modeling for arbitrarily shaped three-dimensional structures: *Geophysics*, **44**, 753–780.
- Farquharson, C. G., 2008, Constructing piecewise-constant models in multi-dimensional minimum-structure inversions: *Geophysics*, **73**, no. 1, K1–K9.
- Finizola, A., M. Aubert, A. Revil, C. Schütze, and F. Sortino, 2009, Importance of structural history in the summit area of Stromboli during the 2002–2003 eruptive crisis inferred from temperature, soil CO₂, self-potential, and electrical resistivity tomography: *Journal of Volcanology and Geothermal Research*, **183**, 213–227.
- Gropp, W., E. Lusk, and A. Skjellum, 1999, *Using MPI: Portable Parallel Programming with the Message-Passing Interface*: MIT Press.
- Günther, T., C. Rücker, and K. Spitzer, 2006, Three-dimensional modeling and inversion of DC resistivity data incorporating topography — II. Inversion: *Geophysical Journal International*, **166**, 506–517.
- Henderson, R. D., F. D. Day-Lewis, E. Abarca, C. F. Harvey, H. N. Karam, L. Liu, and J. W. Lane, Jr., 2009, Marine electrical resistivity imaging of submarine groundwater discharge: Sensitivity analysis and application in Waquoit Bay, Massachusetts, USA: *Hydrogeology Journal*, **18**, 173–185.
- Hestenes, M. R., and E. Stiefel, 1952, Methods of conjugate gradients for solving linear systems: *Journal of Research of the National Bureau of Standards*, **49**, 409–436.
- Huber, P. J., 1964, Robust estimation of a local parameter: *Annals of Mathematical Statistics*, **35**, 73–101.
- Johnson, T. C., P. S. Routh, T. Clemo, W. Barrash, and W. P. Clement, 2007, Incorporating geostatistical constraints in nonlinear inversion problems: *Water Resources Research*, **43**, W10422.
- Johnson, T. C., R. J. Versteeg, H. Huang, and P. S. Routh, 2009, Data-domain correlation approach for joint hydrogeologic inversion of time-lapse hydrogeologic and geophysical data: *Geophysics*, **74**, no. 6, F127–F140.
- Lane, J. W., Jr., F. D. Day-Lewis, and C. C. Casey, 2006, Geophysical monitoring of a field-scale biostimulation pilot project: *Ground Water*, **44**, 430–443.
- Legaz, A., J. Vandemeulebrouck, A. Revil, A. Kemna, A. W. Hurst, R. Reeves, and R. Papasin, 2009, A case study of resistivity and self-potential signatures of hydrothermal instabilities, Inferno Crater Lake, Waimangu, New Zealand: *Geophysical Research Letters*, **36**, L12306.
- Li, Y., and D. W. Oldenburg, 2000, 3-D inversion of induced polarization data: *Geophysics*, **65**, 1931–1945.
- Lowry, T., M. B. Allen, and P. N. Shive, 1989, Singularity removal: A refinement of resistivity modeling techniques: *Geophysics*, **54**, 766–774.
- McGillivray, P. R., and D. W. Oldenburg, 1990, Methods for calculating Fréchet derivatives and sensitivities for the non-linear inverse problem: A comparative study: *Geophysical Prospecting*, **38**, 499–524.
- Oldenburg, D. W., and Y. Li, 1994, Inversion of induced polarization data: *Geophysics*, **59**, 1327–1341.
- Parra, J. O., 1988, Electrical response of a leak in a geomembrane liner: *Geophysics*, **53**, 1445–1452.
- Pridmore, D. F., G. W. Hohmann, S. H. Ward, and W. R. Sill, 1981, An investigation of finite-element modeling for electrical and electromagnetic data in three dimensions: *Geophysics*, **46**, 1009–1024.
- Revil, A., A. Finizola, S. Piscitelli, E. Rizzo, T. Ricci, A. Crespy, B. Angeletti, M. Balasco, S. Barde Cabusson, L. Bennati, A. Bolève, S. Byrdina, N. Carzaniga, F. Di Gangi, J. Morin, A. Perrone, M. Rossi, E. Roulleau, and B. Suski, 2008, Inner structure of La Fossa di Vulcano (Vulcano Island, southern Tyrrhenian Sea, Italy) revealed by high-resolution electric resistivity tomography coupled with self-potential, temperature, and CO₂ diffuse degassing measurements: *Journal of Geophysical Research*, **113**, B07207.
- Rodi, W., and R. L. Mackie, 2001, Nonlinear conjugate gradients algorithm for 2-D magnetotelluric inversion: *Geophysics*, **66**, 174–187.
- Rücker, C., T. Günther, and K. Spitzer, 2006, Three-dimensional modeling and inversion of DC resistivity data incorporating topography — I. Modeling: *Geophysical Journal International*, **166**, 495–505.
- Sasaki, Y., 1994, 3-D resistivity inversion using the finite-element method: *Geophysics*, **59**, 1839–1848.
- Scales, J. A., and A. Gersztenkorn, 1988, Robust methods in inverse theory: *Inverse Problems*, **4**, 1071–1091.
- Seigel, H. O., 1959, Mathematical formulation and type curves for induced polarization: *Geophysics*, **24**, 547–565.
- Si, H., 2003, TetGen: A quality tetrahedral mesh generator and a 3D Delaunay triangulator: <http://tetgen.berlios.de>, accessed 15 June 2010.
- Singha, K., F. D. Day-Lewis, and J. W. Lane, Jr., 2007, Geoelectrical evidence of bicontinuum transport in groundwater: *Geophysical Research Letters*, **34B**, L12401.
- Versteeg, R., and T. Johnson, 2008, Using time-lapse electrical geophysics to monitor subsurface processes: *The Leading Edge*, **27**, 1488–1497.