

## 2.5D Algorithm for Tomographic Imaging of the Deep Electromagnetic Geophysical Measurement

A. Abubakar, T. M. Habashy, V. Druskin, D. Alumbaugh, P. Zhang  
M. Wilt, and L. Knizhnerman  
Schlumberger, USA

**Abstract**—We present a 2.5D inversion algorithm for the interpretation of electromagnetic data collected in a cross-well configuration. Some inversion results from simulated data as well as from field measurements are presented in order to show the efficiency and the robustness of the algorithm.

### 1. Introduction

Electromagnetic methods are essential tools for the appraisal of a reservoir because of their sensitivity to the resistivity (conductivity) which is a function of the fluid saturation. One of the traditional electromagnetic techniques for well logging is the induction single-well measurement. This technique is employed both as a wireline technology and as a measurement while drilling to estimate near well-bore resistivity. This induction logging measurement has a sensitivity of up to a few meters from the well and is a function of the separation between the transmitter and receiver and frequency of operation.

To reach deeper into the reservoir, a cross-well electromagnetic induction technology was developed, see Wilt et al., [6] and Spies and Habashy [4]. The system operates very similar to the single-well logging tool however with transmitter and receiver deployed in separate wells. During a cross-well survey the receivers are lowered into one well, initially to the bottom of the survey-depth interval. Then the transmitter is lowered into the second well and is moved to log the entire survey-depth interval. During logging the transmitter broadcasts electromagnetic signals at a number of pre-prescribed frequencies while at the receiver well these signals are recorded. After the transmitter run is completed the receiver array is moved to the next depth station in the survey interval and the process is then repeated until the entire depth interval has been covered. After the data set has been collected, an inversion process is applied to convert the electromagnetic signals to a resistivity distribution map of the region between the wells. Furthermore, since most of the survey involves only two wells, one can usually assume in the inversion that the geometry is 2D (the resistivity distribution is invariant along the direction perpendicular to the plane containing the wells).

This inverse process is one of the most challenging parts of the effort to make this cross-well technology work since it requires one to solve a full nonlinear inverse scattering problem, which is usually ill-conditioned and non-unique. Moreover, when the number of the model parameters to be inverted is large, the inversion can be very time-consuming.

In order to carry out the inversion within a reasonable time, we employ a finite-difference code as a forward simulator. In this forward code the configuration is numerically discretized using a small number of cells determined by the optimal grid technique, see Ingerman et al. [3]. The resulting linear system of equations representing the discretized forward problem has to be solved in each inversion step. To solve this system, we use a LU decomposition method that allows us to obtain the solution for all transmitters simultaneously. Furthermore, in order to be able to use the optimal grid without sacrificing accuracy we use an anisotropic material averaging formula. All these features help in reducing the computational time for constructing sensitivity kernel and for calculating the data misfit.

For the inversion method, we employ a constrained Gauss-Newton minimization scheme (see Habashy and Abubakar [2]) where the inverted model parameters are forced to lie within their physical bounds by using a nonlinear transformation procedure. We further enforce a reduction in the cost function after each iteration by employing a line search method. To improve on the conditioning of the inversion problem, we use two different regularizers. The first is a traditional  $L_2$ -norm regularizer, which allows a smooth solution. The second is the so-called weighted  $L_2$ -norm regularizer, which can provide a sharp reconstructed image, see van den Berg and Abubakar in [5]. The trade-off parameter which provides the relative weighting between the data and the regularization part of the cost function is determined automatically to enhance the robustness of the method. We will present results from simulated data as well as from field measurements to demonstrate the capabilities of the developed algorithm.

## 2. Methodology

We consider a general discrete nonlinear inverse problem described by the operator equation

$$\bar{\mathbf{d}}^{\text{obs}} = \bar{\mathbf{S}}(\bar{\mathbf{m}}), \quad (1)$$

where  $\bar{\mathbf{d}}^{\text{obs}} = [d_1^{\text{obs}} d_2^{\text{obs}} \dots d_J^{\text{obs}}]^T$  is the vector of measured data and  $\bar{\mathbf{S}} = [S_1 S_2 \dots S_J]^T$  is the vector of data computed for the model parameters  $\bar{\mathbf{m}} = [m(x_q, z_r), q = 1, 2, \dots, Q; r = 1, 2, \dots, R]$ , where  $x_q$  and  $z_r$  denote the center of the 2D discretization cell. We use a lexicographical ordering of the unknowns to map the 2D array indices to 1D column indices  $(q, r) \rightarrow R \times (q - 1) + r$ . The superscript  $T$  denotes the transpose of a vector. We assume that there are  $J$  number of data points in the experiment and that the configuration can be described by  $I = Q \times R$  model parameters. In this cross-well electromagnetic problem the data are the component of the magnetic field which is parallel to the borehole axis. The unknown model parameter  $m(\bar{\mathbf{r}}) = \sigma(\bar{\mathbf{r}})/\sigma_0$  is the normalized conductivity where  $\sigma_0$  is a constant conductivity. In the implementation  $\sigma_0$  is chosen to be the average of the initial model used in the inversion.

We pose the inversion as the minimization problem. Hence at the  $n^{\text{th}}$  iteration we reconstruct  $\bar{\mathbf{m}}_n$  that minimizes

$$\Phi_n(\bar{\mathbf{m}}) = \phi^d(\bar{\mathbf{m}}) + \lambda_n \phi_n^m(\bar{\mathbf{m}}), \quad (2)$$

where  $\phi^d$  is a measure of data misfit:

$$\phi^d(\bar{\mathbf{m}}) = \frac{\sum_{j=1}^J |W_{j,j} [d_j^{\text{obs}} - S_j(\bar{\mathbf{m}})]|^2}{\sum_{j=1}^J |W_{j,j} d_j^{\text{obs}}|^2}, \quad (3)$$

in which  $|\cdot|$  denotes the absolute value and  $\bar{\bar{\mathbf{W}}}$  is a diagonal matrix whose elements are the estimates of the standard deviations of the noise. The symbol  $\lambda$  denotes the regularization parameter and  $\phi_n^m$  is a measure of the variation in the geometrical configuration:

$$\phi_n^m(\bar{\mathbf{m}}) = \int_D d\bar{\mathbf{r}} b_n^2(\bar{\mathbf{r}}) \left\{ |\nabla [m(\bar{\mathbf{r}}) - m^{\text{ref}}(\bar{\mathbf{r}})]|^2 + \delta_n^2 \right\}, \quad (4)$$

where  $\nabla = [\partial_x \partial_z]^T$  denotes spatial differentiation with respect to  $\bar{\mathbf{r}} = [x z]^T$ , and the weight  $b_n^2(\bar{\mathbf{r}})$  is given by

$$b_n^2(\bar{\mathbf{r}}) = \frac{1}{\int_D d\bar{\mathbf{r}} |\nabla [m_n(\bar{\mathbf{r}}) - m^{\text{ref}}(\bar{\mathbf{r}})]|^2 + \delta_n^2} \quad (5)$$

for the  $L_2$ -norm regularizer and

$$b_n^2(\bar{\mathbf{r}}) = \frac{1}{V} \frac{1}{|\nabla [m_n(\bar{\mathbf{r}}) - m^{\text{ref}}(\bar{\mathbf{r}})]|^2 + \delta_n^2} \quad (6)$$

for the weighted  $L_2$ -norm regularizer introduced in van den Berg and Abubakar [5]. The symbol  $V = \int_D d\bar{\mathbf{r}}$  denotes the volume of the computational domain and  $\bar{\mathbf{m}}^{\text{ref}}$  is the known reference model. Note that for the  $L_2$ -norm regularizer the weight  $b_n^2(\bar{\mathbf{r}})$  is independent of the spatial position  $\bar{\mathbf{r}}$ . The  $\delta_n^2$  is a constant which is chosen to be equal to:  $\delta_n^2 = \phi^d(\bar{\mathbf{m}}_n)/(\Delta_x \Delta_z)$ , where  $\Delta_x$  and  $\Delta_z$  are the widths of the discretization cell. The regularization parameter  $\lambda$  is determined automatically using the technique described in Habashy and Abubakar [2].

To solve (2) we employ a Gauss-Newton minimization approach. At the  $n^{\text{th}}$  iteration we obtain a set of linear equations for the search vector  $\bar{\mathbf{p}}_n$  that identifies the minimum of the approximated quadratic cost function, namely,

$$\bar{\bar{\mathbf{H}}}_n \cdot \bar{\mathbf{P}}_n = -\bar{\bar{\mathbf{g}}}_n, \quad (7)$$

where

$$\bar{\bar{\mathbf{H}}}_n = \bar{\mathbf{J}}_n^T \cdot \bar{\bar{\mathbf{W}}}^T \cdot \bar{\bar{\mathbf{W}}} \cdot \bar{\mathbf{J}}_n^T + \lambda_n \bar{\bar{\mathcal{L}}}(\bar{\mathbf{m}}_n), \quad (8)$$

$$\bar{\bar{\mathbf{g}}}_n = \bar{\mathbf{J}}_n^T \cdot \bar{\bar{\mathbf{W}}}^T \cdot [\bar{\mathbf{d}}^{\text{obs}} - \bar{\mathbf{S}}(\bar{\mathbf{m}}_n)] - \lambda_n \bar{\bar{\mathcal{L}}}(\bar{\mathbf{m}}_n) \cdot \bar{\mathbf{m}}_n, \quad (9)$$

in which

$$\bar{\bar{\mathcal{L}}}(\bar{\mathbf{m}}_n) \cdot \bar{\mathbf{m}}_n = \nabla \cdot [b_n^2(\bar{\mathbf{r}}) \nabla m_n(\bar{\mathbf{r}})]. \quad (10)$$

In (8) and (9),  $\bar{\mathbf{J}}_n = \bar{\mathbf{J}}(\bar{\mathbf{m}}_n)$  is the  $J \times I$  Jacobian matrix and is given by the following expression:

$$J_{j,i;n} = \eta \frac{\partial S_j(\bar{\mathbf{m}}_n)}{\partial m_{i;n}}, \quad \eta = \frac{1}{\sum_{k=1}^J |W_{k,k} d_k^{\text{obs}}|^2}. \quad (11)$$

This Jacobian matrix is calculated using an adjoint formulation, which only needs an extra forward problem solution at each Gauss-Newton search step. In this extra forward problem solution the roles of the transmitters and receivers are interchanged. However since we are using a 2.5D forward code with a LU decomposition solver, we need only one forward call to calculate both the data misfit and to generate the Jacobian matrix. Note that the use of the direct solver is possible, since we reduced the number of grids outside the inter-well region by employing the optimal grid technique in Ingerman et al. [3]. Furthermore, in order to be able to use the optimal grids without scarifying accuracy we use an anisotropic homogenization technique.

Since the size of the Hessian matrix  $\bar{\mathbf{H}}_n$  is large, we solve the linear system of equations (7) using a linear iterative method. To that end we first rewrite equation (7) as follows:

$$\bar{\mathcal{K}} \cdot \bar{\mathbf{p}}_n = \bar{\mathbf{f}}, \quad (12)$$

where  $\bar{\mathcal{K}} = \bar{\mathbf{H}}_n$  and  $\bar{\mathbf{f}} = -\bar{\mathbf{g}}_n$ . Since  $\bar{\mathcal{K}}$  is a self adjoint matrix, we employ a Conjugate Gradient Least Square (CGLS) scheme to solve this linear system of equations. This CGLS scheme starts with the initial values:

$$\bar{\mathbf{r}}^{(0)} = \bar{\mathbf{f}} - \bar{\mathcal{K}} \cdot \bar{\mathbf{p}}_n^{(0)}, \quad \text{ERR}^{(0)} = \frac{\|\bar{\mathbf{r}}^{(0)}\|}{\|\bar{\mathbf{f}}\|}, \quad (13)$$

where  $\bar{\mathbf{p}}_n^{(0)} = \bar{\mathbf{p}}_{n-1}$ . Next, we compute successively for  $N = 1, 2, \dots$ ,

$$\begin{aligned} A^{(N)} &= \langle \bar{\mathbf{r}}^{(N-1)}, \bar{\mathcal{K}} \cdot \bar{\mathbf{r}}^{(N-1)} \rangle, \\ \bar{\mathbf{u}}^{(N)} &= \bar{\mathbf{r}}^{(N-1)}, \quad N = 1, \\ &= \bar{\mathbf{r}}^{(N-1)} + \frac{A^{(N)}}{A^{(N-1)}} \bar{\mathbf{u}}^{(N-1)}, \quad N > 1, \\ B^{(N)} &= \|\bar{\mathcal{K}} \cdot \bar{\mathbf{u}}^{(N)}\|^2, \\ \bar{\mathbf{p}}_n^{(N)} &= \bar{\mathbf{p}}_n^{(N-1)} + \frac{A^{(N)}}{B^{(N)}} \bar{\mathbf{u}}^{(N)}, \\ \bar{\mathbf{r}}^{(N)} &= \bar{\mathbf{f}} - \bar{\mathcal{K}} \cdot \bar{\mathbf{p}}_n^{(N)}, \quad \text{ERR}^{(N)} = \frac{\|\bar{\mathbf{r}}^{(N)}\|}{\|\bar{\mathbf{f}}\|}, \end{aligned} \quad (14)$$

where  $\|\bar{\mathbf{u}}\| = \sqrt{\langle \bar{\mathbf{u}}, \bar{\mathbf{u}} \rangle}$  denotes the  $L_2$ -norm of a vector. This CGLS iteration process stops if the relative error  $\text{ERR}^{(N)}$  reaches a prescribed value, or when the total number of iterations  $N$  exceeds a prescribed maximum.

After the search vector  $\bar{\mathbf{p}}_n = \bar{\mathbf{p}}_n^{(N)}$  has been obtained, the unknown model parameters are updated as follows:

$$\bar{\mathbf{m}}_{n+1} = \bar{\mathbf{m}}_n + \nu_n \bar{\mathbf{p}}_n, \quad (15)$$

where  $\nu_n$  is a scalar constant parameter to be determined by a line search algorithm. In the implementation we always try first the full step, i.e.,  $\nu_n = 1$ , and check if it reduced the value of the cost function  $\Phi_n$ . If not, we backtrack along the Gauss-Newton step until we have an acceptable step. Since the Gauss-Newton step is a descent direction for  $\Phi_n$ , we are guaranteed to find an acceptable step. In this procedure  $\nu_n$  is selected such that:

$$\Phi_n(\bar{\mathbf{m}}_n + \nu_n \bar{\mathbf{p}}_n) \leq \Phi_n(\bar{\mathbf{m}}_n) + \alpha \nu_n \delta \Phi_{n+1}, \quad (16)$$

where  $0 < \alpha < 1$  is a fractional number, which is set to be quite small, i.e.,  $\alpha$  to  $10^{-4}$ , so that hardly more than a decrease in cost function value is required (see Dennis and Schnabel [1]). The parameter  $\delta \Phi_{n+1}$  is the rate of decrease of  $\phi(\bar{\mathbf{m}})$  at  $\bar{\mathbf{m}}_n$  along the direction  $\bar{\mathbf{p}}_n$  and is given by:

$$\delta \Phi_{n+1} = \left. \frac{\partial}{\partial \nu} \Phi_n(\bar{\mathbf{m}}_n + \nu \bar{\mathbf{p}}_n) \right|_{\nu=0} = \bar{\mathbf{g}}_n^T \cdot \bar{\mathbf{p}}_n. \quad (17)$$

If, at the  $(n+1)^{\text{th}}$  iteration,  $\nu_n^{(m)}$  is the current step-length that does not satisfy the condition (16), we compute the next backtracking step-length,  $\nu_n^{(m+1)}$ , by searching for the minimum of the cost function assuming a quadratic approximation in  $\nu$ . Hence  $\nu_k^{(m+1)}$  for  $m = 0, 1, 2, \dots$  is given by:

$$\nu_n^{(m+1)} = \frac{-0.5[\nu_k^{(m)}]^2 \delta\Phi_{(n+1)}}{\Phi_n(\bar{\mathbf{m}}_n + \nu_n^{(m)} \bar{\mathbf{p}}_n) - \Phi_n(\bar{\mathbf{m}}_n) - \nu_n^{(m)} \delta\Phi_{n+1}}. \quad (18)$$

In general, it is not desirable to decrease  $\nu_n^{(m+1)}$  too much since this may excessively slow down the iterative process. To prevent this slow down, we set  $\nu_n^{(m+1)} = 0.1\nu_n^{(m)}$  if  $\nu_n^{(m+1)} < 0.1\nu_n^{(m)}$  (but with  $\nu_n$  not to decrease below 0.1, i.e.,  $\nu_{\min} = 0.1$  to guard against a too small value of  $\nu$ ) and then proceed with the Gauss-Newton step.

To impose *a priori* information of maximum and minimum bounds on the unknown parameters, we constrained them using a nonlinear transformation of the form:

$$m_i = \frac{m_i^{\max} + m_i^{\min}}{2} + \frac{m_i^{\max} - m_i^{\min}}{2} \sin(c_i), \quad (19)$$

where  $m_i^{\max}$  and  $m_i^{\min}$  are upper and lower bounds on the physical model parameter  $m_i$ . It is clear that  $m_i \rightarrow m_i^{\min}$ , as  $\sin(c_i) \rightarrow -1$  and  $m_i \rightarrow m_i^{\max}$ , as  $\sin(c_i) \rightarrow +1$ . This nonlinear transformation will force the reconstruction of the model parameters to lie always within their prescribed bounds. Formally by using this nonlinear transformation we should be updating the auxiliary unknown parameters  $c_i$  instead of the model parameters  $m_i$ . However by using the relation  $p_i = q_i dm_i/dc_i$  where  $q_i$  is the Gauss-Newton search step with respect to  $c_i$ , we obtain the following relationships between the two successive iterates  $m_{i,n+1}$  and  $m_{i,n}$  of  $m_i$ :

$$m_{i,n+1} = \frac{m_i^{\max} + m_i^{\min}}{2} + \alpha_n \sin\left(\frac{\nu_n p_{i,n}}{\alpha_n}\right) + \left(m_{i,n} - \frac{m_i^{\max} + m_i^{\min}}{2}\right) \cos\left(\frac{\nu_n p_{i,n}}{\alpha_n}\right), \quad (20)$$

where  $\alpha_n = \sqrt{(m_i^{\max} - m_{i,n})(m_{i,n} - m_i^{\min})}$ .

The iteration process will be terminated if one of the following conditions occurs: (1) The misfit  $\phi^d(\bar{\mathbf{m}}_n)$  is within a prescribed tolerance factor; (2) The difference between the misfit at two successive iterates  $n$  is within a prescribed tolerance factor; (3) the difference between the model parameters  $\bar{\mathbf{m}}$  at two successive iterates  $n$  is within a prescribed tolerance factor; (4) The total number of iterations exceeds a prescribed maximum.

### 3. Numerical Example

As a test example we employ a model shown in Fig. 1(a). This model was originally used to study a  $CO_2$  injection operation and is employed here as it includes smoothly varying dipping stratigraphy as well as sharp boundaries and deviated wells. The background model shown in Fig. 1(b) is obtained using single-well logs interpolated between the two wells. The hypothesized  $CO_2$  injection region is shown in red in Fig. 1(a). The change between the true model and the background model is shown in Fig. 1(c) given in percentage difference (%). The data are collected using 41 transmitters and 41 receivers. The locations of the transmitters and receivers are denoted by 'T' and 'R' in Fig. 1. Thus we have 1681 complex-valued data points. After generating the synthetic data, we corrupted the data with random white noise that corresponds to 2% of the maximum amplitude of all data points. The inversion domain is from  $x = -30$  m to  $x = 350$  m and  $z = 950$  m to  $z = 1250$  m and is discretized into cells of dimensions 5 m by 5 m, hence the total number of unknown model parameters is 4636.

First we run our inversion algorithm using the  $L_2$ -norm regularizer given in (4) and (5). As the initial estimate we use the background model given in Fig. 1(b). Using this regularization term, the scheme took 15 iterations to converge. Figs. 1(d) and 1(e) show the percentage difference between the inverted resistivity and the background resistivity. The image obtained using the  $L_2$ -norm regularizer is shown in Fig. 1(d). The image obtained in this case has the appearance of a spatially smoothed version of the model changes in Fig. 1(c). Next we rerun our inversion code, however now we use the weighted  $L_2$ -norm regularization term given in (4) and (6). The inversion results after 19 iterations are shown in Fig. 1(e). By using the weighted  $L_2$ -norm regularizer we obtain a significant improvement in the reconstruction of the geometry and the amplitude of the change due to the  $CO_2$  injection. Finally we note that one iteration of the scheme takes only 180 seconds on a PC with a Pentium IV 3.04 GHz processor.

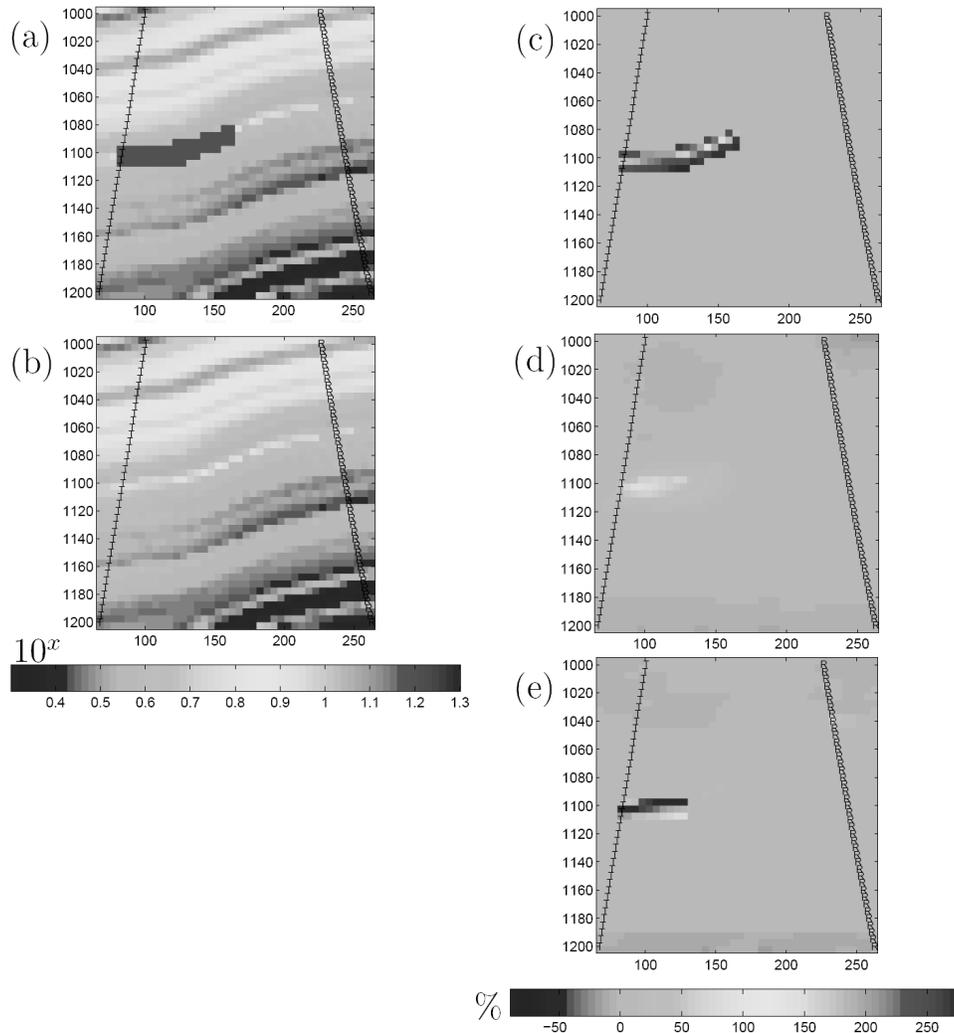


Figure 1: The resistivity distribution of the true model (a), of the initial model (b), the changes between (a) and (b) given in percentage (c), the inverted resistivity plotted as the change with respect to the model in (b) obtained using a  $L_2$ -norm regularizer (d) and a weighted  $L_2$ -norm regularizer (e).

## REFERENCES

1. Dennis JR, J. E. and R. B. Schnabel, *Numerical Methods for Unconstrained Optimization and Nonlinear Equations*, Prentice Hall, Inc., New Jersey, 1983.
2. Habashy, T. M. and A. Abubakar, "A general framework for constraint minimization for the inversion of electromagnetic measurements," *Progress in Electromagnetic Research*, Vol. 46, 265–312, 2004.
3. Ingerman, D., V. L. Druskin, and L. Knizhnerman, "Optimal finite difference grids and rational approximations of the square root I. Elliptic problems," *Communications on Pure and Applied Mathematics*, LIII, 1039–1066, 2000.
4. Spies, B. R. and T. M. Habashy, "Sensitivity analysis of cross-well electromagnetics," *Geophysics*, Vol. 60, 834–845, 1995.
5. Van den Berg, P. M. and A. Abubakar, "Contrast source inversion method: state of art," *Progress in Electromagnetics Research*, Vol. 34, 189–218, 2001.
6. Wilt, M. J., D. L. Alumbaugh, H. F. Morrison, A. Becker, K. H. Lee, and M. Deszcz-Pan, "Crosshole electromagnetic tomography: system design considerations and field results," *Geophysics*, Vol. 60, 871–885, 1995.