Progress In Electromagnetics Research, PIER 46, 265–312, 2004

A GENERAL FRAMEWORK FOR CONSTRAINT MINIMIZATION FOR THE INVERSION OF ELECTROMAGNETIC MEASUREMENTS

T. M. Habashy and A. Abubakar

Schlumberger-Doll Research
36 Old Quarry Road, Ridgefield, CT 06877-4108, USA

Abstract—In this paper, we developed a general framework for the inversion of electromagnetic measurements in cases where parametrization of the unknown configuration is possible. Due to the ill-posed nature of this nonlinear inverse scattering problem, this parametrization approach is needed when the available measurement data are limited and measurements are only carried out from limited transmitter-receiver positions (limited data diversity). By carrying out this parametrization, the number of unknown model parameters that need to be inverted is manageable. Hence the Newton based approach can advantageously be used over gradient-based approaches. In order to guarantee an error reduction of the optimization process, the iterative step is adjusted using a line search algorithm. Further unlike the most available Newton-based approaches available in the literature, we enhanced the Newton based approaches presented in this paper by constraining the inverted model parameters with nonlinear transformation. This constrain forces the reconstruction of the unknown model parameters to lie within their physical bounds. In order to deal with cases where the measurements are redundant or lacking sensitivity to certain model parameters causing non-uniqueness of solution, the cost function to be minimized is regularized by adding a penalty term. One of the crucial aspects of this approach is how to determine the regularization parameter determining the relative importance of the misfit between the measured and predicted data and the penalty term. We reviewed different approaches to determine this parameter and proposed a robust and simple way of choosing this regularization parameter with aid of recently developed multiplicative regularization analysis. By combining all the techniques mentioned above we arrive at an effective and robust parametric algorithm. As numerical examples we present results of electromagnetic inversion at induction frequency in the deviated borehole configuration.

1. INTRODUCTION

In inverse scattering problems, one aims to determine the shape, location and material parameters of the object from measurements of the (scattered) wavefield, when a number of waves are generated such that they successively illuminate the domain of interest. Since the wavefields itself depend on the material parameters, the inverse scattering problem is essentially nonlinear. For simple forward scattering models (mostly one-dimensional cases), a forward solver can generate a database containing a full collection of scattered-field data for a number of possible realizations of the scattering object, and one can then select a particular realization with the best fit to the actually measured data. But, in practice, the execution of such an enumerate inversion method is often impossible due to the large number of discrete unknown variables. Improvements of these Monte Carlo methods, see e.g., [1], are obtained when certain probability is taking into account, e.g., as in genetic algorithms [2]. The advantages of these inversion schemes are the simplicity and the feature that the algorithm will not trap into a local minimum. However, the major disadvantage is the large number of forward solutions to be generated and stored. Furthermore, presently, for a realistic full three-dimensional scattering problem (including multiple scattering), the generation of a full class of forward solutions is not feasible.

We therefore opt for an iterative approach, where the model parameters are updated iteratively by a consistent minimization of the misfit between the measured data and the predicted data. Since the updates of the model parameters are determined from the misfit, precaution has to be taken to overcome the ill-posed nature of the inverse problem and to avoid the local minima of the nonlinear problem. Two- and three-dimensional inverse scattering problem have been studied with various optimization methods, in particular, deterministic gradient type methods. Among the huge amount of literature we only list a few without intention of review: [3–11,

14, 15]. These gradient type methods are very attractive in case where we deal with a huge amount of unknown parameters (pixel-like/tomography inversion) and a large number of measurement data collected from (ideally) all illumination angle is available (enough data diversity) . When the number of available measurement data are limited and collected only from limited possible transmitter-receiver positions (limited data diversity), in order to arrive at a reliable solution, parametrization of the problem is needed. These are cases when one deal with Ground Penetrating Radar, geophysical borehole applications and non-destructive evaluation of corrosion in the metal pipe. By carrying out this parametrization, the number of unknown model parameters that need to be inverted is manageable. With other word, after this parametrization we end up with non under determined system. For this parametric inversion the Newton-based approaches are more appropriate than gradient-based methods because it have faster convergence rates.

In this paper first we reviewed the Newton based methods by putting them in a general framework. After that in order to enhance the performance of these Newton based methods, we constrained them with nonlinear transformations. This constrain forces the reconstruction of the unknown model parameters to lie within the physical bounds. In order to guarantee the error reducing nature of the algorithm, we adjusted the iterative step using a line search algorithm. In this way we prevent the algorithm to jumping around between two successive iteration. Further since the problem is essentially highly non-linear, some regularization is definitely needed. The main problem of adding a regularization term is how we can determine the appropriate regularization parameter or the Lagrange multiplier. In this paper we review a number of way of choosing this regularization parameter and proposed a new way for the Newton based method to choose this parameter. This new approach of choosing the regularization parameter is inspired by the work of [12, 13] on the multiplicative regularization for gradient-type algorithm. Our analysis shows that by posing our cost functional as a multiplicative functional we found that the regularization parameter can be chosen proportional to the original cost functional. This provides us with a robust way to determine the weight for the regularization term. Finally we also point out the relationship between the Newton based minimization method and the Bayesian inference approach. As numerical examples, we present inversion result from electromagnetic data at induction frequency in single-well deviated borehole configuration.

2. THE COST FUNCTION

We define the vector of residuals $\overline{\mathbf{e}}(\overline{\mathbf{x}})$ as a vector whose j-th element is the residual error (also referred to as the data mismatch) of the j-th measurement. The residual error is defined as the difference between the measured and the predicted normalized responses:

$$\overline{\mathbf{e}}(\overline{\mathbf{x}}) = \begin{bmatrix} e_1(\overline{\mathbf{x}}) \\ \vdots \\ e_M(\overline{\mathbf{x}}) \end{bmatrix} = \begin{bmatrix} S_1(\overline{\mathbf{x}}) - m_1 \\ \vdots \\ S_M(\overline{\mathbf{x}}) - m_M \end{bmatrix} = \overline{\mathbf{S}}(\overline{\mathbf{x}}) - \overline{\mathbf{m}}, \tag{1}$$

where M is the number of measurements. The symbol m_j is the normalized observed response (measured data) and S_j is corresponding the normalized simulated response as predicted by the vector of model parameters, $\overline{\mathbf{x}}^T$:

$$\overline{\mathbf{x}}^T = [x_1 \; x_2 \; \cdots x_{N-1} \; x_N] = (\overline{\mathbf{y}} - \overline{\mathbf{y}}_R)^T, \tag{2}$$

where N is the number of unknowns and the superscript ^{T} indicates transposition. We represent the vector of model parameters $\overline{\mathbf{x}}$ as the difference between the vector of the actual parameters $\overline{\mathbf{y}}$ and a reference/background model $\overline{\mathbf{y}}_R$. The reference model includes all *a priori* information on the model parameters such as those derived from independent measurements. We pose the inversion as the minimization of the following cost (or objective) function, $C(\overline{\mathbf{x}})$, of the form [20]:

$$C(\overline{\mathbf{x}}) = \frac{1}{2}\left[\mu\left\{\left\|\overline{\overline{\mathbf{W}}}_d \cdot \overline{\mathbf{e}}(\overline{\mathbf{x}})\right\|^2 - \chi^2\right\} + \left\|\overline{\overline{\mathbf{W}}}_x \cdot (\overline{\mathbf{x}} - \overline{\mathbf{x}}_p)\right\|^2\right] \tag{3}$$

The scalar factor $\mu\,(0 < \mu < \infty)$ is a Lagrange multiplier. Its inverse is called the regularization parameter or damping coefficient. It is a tradeoff parameter determining the relative importance of the two terms of the cost function. The determination of μ will produce an estimate of the model $\overline{\mathbf{x}}$ that has a finite minimum weighted norm (away from a prescribed model $\overline{\mathbf{x}}_p$) and which globally misfits the data to within a prescribed value χ (determined from *a priori* estimates of noise in the data). The second term of the cost function is included to regularize the optimization problem. It safeguards against cases when measurements are redundant or lacking sensitivity to certain model parameters causing non-uniqueness of solution. It also suppresses any possible magnification of errors in our parameter estimations due to noise which is unavoidably present in the measurements. These error magnifications may result in undesirable large variations in the model parameters which may cause instabilities in the inversion. $\overline{\overline{\mathbf{W}}}_x^T \cdot \overline{\overline{\mathbf{W}}}_x$ is

the inverse of the *model covariance matrix* representing the degree of confidence in the prescribed model $\overline{\mathbf{x}}_p$ and is also provided as *a priori* information. It can also be used to bias certain parts of the model $\overline{\mathbf{x}}$ towards the prescribed model $\overline{\mathbf{x}}_p$. $\overline{\overline{\mathbf{W}}}_d^T \cdot \overline{\overline{\mathbf{W}}}_d$ is the inverse of the *data covariance matrix*, which describes the estimated uncertainties (due to noise contamination) in the available data set. It describes not only the estimated variance for each particular data point, but also the estimated correlation between errors. It therefore provides a point by point weighting of the input data according to a prescribed criterion (hence, can be used to reduce the effect of outliers in the data). If the measurement noise is stationary and uncorrelated, then $\overline{\overline{\mathbf{W}}}_d = \text{diagonal}\{1/\sigma_j\}$ where σ_j is the root mean square deviation (standard deviation) of the noise for the j-th measurement.

3. CONSTRAINED MINIMIZATION

To impose *a priori* information, such as positivity or maximum and minimum bound (if they are available) on the inverted model parameters, $\overline{\mathbf{x}}$, we constrained them using a nonlinear transformation:

$$x_i = f(c_i, x_i^{\min}, x_i^{\max}), \qquad -\infty < c_i < \infty, \quad i = 1, 2, \ldots, N, \tag{4}$$

where $x_i^{\max}$ is an upper bound on the physical model parameter x_i and $x_i^{\min}$ is a lower bound. These nonlinear transformations map a constrained minimization problem to an unconstrained one. Note that this constraint will force the reconstruction of the model parameters to lie always within their physical bounds.

Obviously there is an infinite number of nonlinear transformations $f(c_i, x_i^{\min}, x_i^{\max})$ which can map a constrained minimization problem to an unconstrained one. Among the many that were tried, the three transformations, which were found to perform the best, are presented in Appendix A.

4. NORMALIZATION OF THE VECTOR OF RESIDUALS

We employ two possible forms of the cost function of Eq. (3) that puts the various measurements on equal footings. The two forms differ in the way we define the vector of residuals $\overline{\mathbf{e}}(\overline{\mathbf{x}})$. In the first form we define $\overline{\mathbf{e}}(\overline{\mathbf{x}})$ as follows:

$$e_j(\overline{\mathbf{x}}) = \frac{S_j(\overline{\mathbf{x}})}{m_j} - 1, \tag{5}$$

and hence

$$\|\overline{\mathbf{e}}(\overline{\mathbf{x}})\|^2 = \sum_{j=1}^{M} \left| \frac{S_j(\overline{\mathbf{x}})}{m_j} - 1 \right|^2 \tag{6}$$

In the second form we define:

$$e_j(\overline{\mathbf{x}}) = \frac{S_j(\overline{\mathbf{x}}) - m_j}{\|\overline{\mathbf{m}}\|/M}, \tag{7}$$

and hence

$$\|\overline{\mathbf{e}}(\overline{\mathbf{x}})\|^2 = \frac{\sum_{j=1}^{M} |S_j(\overline{\mathbf{x}}) - m_j|^2}{\left(\sum_{j=1}^{M} |m_j|^2\right)/M^2} \tag{8}$$

5. THE NEWTON MINIMIZATION APPROACH

To solve the above nonlinear optimization problem, we employ a Newton minimization approach [17] which is based on a local quadratic model of the cost function. The quadratic model is formed by taking the first three terms of the Taylor-series expansion of the cost function around the current k-th iteration ($\overline{\mathbf{x}}_k$), as follows:

$$C(\overline{\mathbf{x}}_k + \overline{\mathbf{p}}_k) \approx C(\overline{\mathbf{x}}_k) + \overline{\mathbf{g}}^T(\overline{\mathbf{x}}_k) \cdot \overline{\mathbf{p}}_k + \frac{1}{2}\overline{\mathbf{p}}_k^T \cdot \overline{\overline{\mathbf{G}}}(\overline{\mathbf{x}}_k) \cdot \overline{\mathbf{p}}_k, \tag{9}$$

where $\overline{\mathbf{p}}_k = \overline{\mathbf{x}}_{k+1} - \overline{\mathbf{x}}_k$ is the step in $\overline{\mathbf{x}}_k$ towards the minimum of the cost function $C(\overline{\mathbf{x}})$. The vector $\overline{\mathbf{g}}(\overline{\mathbf{x}}) = \nabla C(\overline{\mathbf{x}})$ is the gradient vector of the cost function $C(\overline{\mathbf{x}})$ and is given by the following expression:

$$\begin{aligned} \overline{\mathbf{g}}(\overline{\mathbf{x}}) &= \nabla C(\overline{\mathbf{x}}) = \left[g_n \equiv \frac{\partial C}{\partial x_n}, \quad n = 1, 2, 3, \ldots, N \right] \\ &= \mu\, \overline{\overline{\mathbf{J}}}^T(\overline{\mathbf{x}}) \cdot \overline{\overline{\mathbf{W}}}_d^T \cdot \overline{\overline{\mathbf{W}}}_d \cdot \overline{\mathbf{e}}(\overline{\mathbf{x}}) + \overline{\overline{\mathbf{W}}}_x^T \cdot \overline{\overline{\mathbf{W}}}_x \cdot (\overline{\mathbf{x}} - \overline{\mathbf{x}}_p), \end{aligned} \tag{10}$$

where x_n is the n-th component of the model vector $\overline{\mathbf{x}}$ and $\overline{\overline{\mathbf{J}}}(\overline{\mathbf{x}})$ is the $M \times N$ Jacobian matrix and is given by the following expression:

$$\begin{aligned} \overline{\overline{\mathbf{J}}}(\overline{\mathbf{x}}) &= \left[J_{mn} \equiv \frac{\partial e_m}{\partial x_n}, \qquad m = 1, 2, 3, \ldots, M; \quad n = 1, 2, 3, \ldots, N \right] \\ &= \begin{bmatrix} \partial S_1/\partial x_1 & \cdots & \partial S_1/\partial x_j & \cdots & \partial S_1/\partial x_N \\ \vdots & \ddots & \vdots & \ddots & \vdots \\ \partial S_\ell/\partial x_1 & \cdots & \partial S_\ell/\partial x_j & \cdots & \partial S_\ell/\partial x_N \\ \vdots & \ddots & \vdots & \ddots & \vdots \\ \partial S_M/\partial x_1 & \cdots & \partial S_M/\partial x_j & \cdots & \partial S_M/\partial x_N \end{bmatrix}, \end{aligned} \tag{11}$$

and $\overline{\overline{\mathbf{G}}}(\overline{\mathbf{x}}) = \nabla\nabla C(\overline{\mathbf{x}})$ is the Hessian of the cost function $C(\overline{\mathbf{x}})$ which is a real symmetric $N \times N$ matrix (not necessarily positive-definite) given by:

$$\begin{aligned}\overline{\overline{\mathbf{G}}}(\overline{\mathbf{x}}) \;&=\; \nabla\nabla C(\overline{\mathbf{x}}) = \left[G_{mn} \equiv \frac{\partial^2 C}{\partial x_n \partial x_m}, \qquad n, m = 1, 2, 3, \ldots, N\right] \\ &= \mu\left[\overline{\overline{\mathbf{J}}}^T(\overline{\mathbf{x}}) \cdot \overline{\overline{\mathbf{W}}}_d^T \cdot \overline{\overline{\mathbf{W}}}_d \cdot \overline{\overline{\mathbf{J}}}(\overline{\mathbf{x}}) + \overline{\overline{\mathbf{Q}}}(\overline{\mathbf{x}})\right] + \overline{\overline{\mathbf{W}}}_x^T \cdot \overline{\overline{\mathbf{W}}}_x, \qquad (12)\end{aligned}$$

where $\overline{\overline{\mathbf{Q}}}(\overline{\mathbf{x}}) = \sum_{m=1}^{M} f_m(\overline{\mathbf{x}})\overline{\overline{\mathbf{F}}}_m^T(\overline{\mathbf{x}})$ and $f_m(\overline{\mathbf{x}})$ is the m-th element of the weighted vector of residuals, $\overline{\mathbf{f}}(\overline{\mathbf{x}}) = \overline{\overline{\mathbf{W}}}_d \cdot \overline{\mathbf{e}}(\overline{\mathbf{x}})$, and $\overline{\overline{\mathbf{F}}}_m(\overline{\mathbf{x}}) = \nabla\nabla f_m(\overline{\mathbf{x}}) = [\partial^2 f_m/\partial x_i \partial x_j,\ i, j = 1, 2, 3, \ldots, N]$. Note that the full Hessian $\overline{\overline{\mathbf{G}}}(\overline{\mathbf{x}})$ of the cost function $C(\overline{\mathbf{x}})$ includes second order information that accounts for curvature. The minimum of the right-hand side of Eq. (9) is achieved if $\overline{\mathbf{p}}_k$ is a minimum of the quadratic function

$$\phi(\overline{\mathbf{p}}) = \overline{\mathbf{g}}^T(\overline{\mathbf{x}}_k) \cdot \overline{\mathbf{p}} + \frac{1}{2}\overline{\mathbf{p}}^T \cdot \overline{\overline{\mathbf{G}}}(\overline{\mathbf{x}}_k) \cdot \overline{\mathbf{p}} \qquad (13)$$

The function $\phi(\overline{\mathbf{p}})$ has a stationary point (a minimum, a maximum or a saddle point also called point of inflection) at $\overline{\mathbf{p}}_o$, only if the gradient vector of $\phi(\overline{\mathbf{p}})$ vanishes at $\overline{\mathbf{p}}_o$, i.e.,

$$\nabla\phi(\overline{\mathbf{p}}_o) = \overline{\overline{\mathbf{G}}} \cdot \overline{\mathbf{p}}_o + \overline{\mathbf{g}} = 0 \qquad (14)$$

Thus, the stationary point is the solution to the following set of linear equations:

$$\overline{\overline{\mathbf{G}}} \cdot \overline{\mathbf{p}}_o = -\overline{\mathbf{g}} \qquad (15)$$

5.1. Case (1): $\overline{\overline{\mathbf{G}}}$ is Singular

Equation (15) can be expressed as follows:

$$\overline{\mathbf{g}} = -\sum_{j=1}^{N} \overline{\mathbf{G}}_j p_j, \qquad (16)$$

where $\overline{\mathbf{G}}_j$ are the columns of the $N \times N$ matrix $\overline{\overline{\mathbf{G}}}$ and p_j is the j-th component of the vector $\overline{\mathbf{p}}_o$.

If $\overline{\overline{\mathbf{G}}}$ is singular, then some of the columns of $\overline{\overline{\mathbf{G}}}$ can be expressed as a linear combination of others. Hence, the columns of $\overline{\overline{\mathbf{G}}}$ do not constitute a complete set of basis vectors that are sufficient to represent an arbitrary vector.

If the vector $\overline{\mathbf{g}}$ cannot be expressed as a linear combination of the columns of $\overline{\overline{\mathbf{G}}}$ (i.e., $\overline{\mathbf{g}}$ has a nonzero component in the null-space of $\overline{\overline{\mathbf{G}}}$), then the system of Eq. (15) is incompatible and does not have a solution. Thus, the stationary point $\overline{\mathbf{p}}_o$ does not exist and the function ϕ is unbounded above and below (i.e., has neither a maximum nor a minimum value). On the other hand, if the vector $\overline{\mathbf{g}}$ can be expressed as a linear combination of the columns of $\overline{\overline{\mathbf{G}}}$, then the system of Eq. (15) is compatible and has more than one solution.

5.2. Case (2): $\overline{\overline{\mathbf{G}}}$ Is Nonsingular

In this case, Eq. (15) has a unique solution given by:

$$\overline{\mathbf{p}}_o = -\overline{\overline{\mathbf{G}}}^{-1} \cdot \overline{\mathbf{g}} = -\sum_{j=1}^{N} \frac{1}{\lambda_j} \left(\overline{\mathbf{v}}_j^T \cdot \overline{\mathbf{g}} \right) \overline{\mathbf{v}}_j, \tag{17}$$

whose norm is given by:

$$\|\overline{\mathbf{p}}_o\| \equiv \left(\overline{\mathbf{p}}_o^T \cdot \overline{\mathbf{p}}_o \right)^{1/2} = \left[\sum_{j=1}^{N} \frac{1}{\lambda_j^2} \left(\overline{\mathbf{v}}_j^T \cdot \overline{\mathbf{g}} \right)^2 \right]^{1/2}, \tag{18}$$

where λ_j and $\overline{\mathbf{v}}_j$ are the eigenvalues and the corresponding orthonormal eigenvectors of the $N \times N$ Hessian matrix $\overline{\overline{\mathbf{G}}}$:

$$\overline{\overline{\mathbf{G}}} \cdot \overline{\mathbf{v}}_j = \lambda_j \overline{\mathbf{v}}_j, \tag{19}$$

with

$$\overline{\mathbf{v}}_i^T \cdot \overline{\mathbf{v}}_j = \delta_{ij} \tag{20}$$

At the stationary point $\overline{\mathbf{p}}_o$, given by Eq. (17), we have from Eq. (9):

$$\begin{aligned} C(\overline{\mathbf{x}}_k + \overline{\mathbf{p}}_o) &\approx C(\overline{\mathbf{x}}_k) - \frac{1}{2} \overline{\mathbf{g}}^T(\overline{\mathbf{x}}_k) \cdot \overline{\overline{\mathbf{G}}}^{-1}(\overline{\mathbf{x}}_k) \cdot \overline{\mathbf{g}}(\overline{\mathbf{x}}_k) \\ &= C(\overline{\mathbf{x}}_k) - \frac{1}{2} \sum_{j=1}^{N} \frac{1}{\lambda_j} \left(\overline{\mathbf{v}}_j^T \cdot \overline{\mathbf{g}} \right)^2 \end{aligned} \tag{21}$$

The stationary point $\overline{\mathbf{p}}_o$, given by Eq. (17), is either a minimum, a maximum or a saddle point, depending on the definiteness of the matrix $\overline{\overline{\mathbf{G}}}$. If $\overline{\overline{\mathbf{G}}}(\overline{\mathbf{x}}_k)$ is a positive definite matrix, then the quadratic model of Eq. (9) is guaranteed to have a unique stationary point (given by

Eq. (17)). Furthermore, and from Eq. (21), the direction of $\overline{\mathbf{p}}_o$ is guaranteed to be a descent direction, since

$$C(\overline{\mathbf{x}}_k + \overline{\mathbf{p}}_o) - C(\overline{\mathbf{x}}_k) \approx -\frac{1}{2}\sum_{j=1}^{N} \frac{1}{\lambda_j}\left(\overline{\mathbf{v}}_j^T \cdot \overline{\mathbf{g}}\right)^2 < 0 \tag{22}$$

In this case, we choose the search direction $\overline{\mathbf{p}}_k = \overline{\mathbf{p}}_o$. On the other hand, if $\overline{\overline{\mathbf{G}}}(\overline{\mathbf{x}}_k)$ is indefinite (but nonsingular), then the quadratic model of Eq. (9) is guaranteed to have a unique stationary point (given by Eq. (17)). However, this stationary point is not necessarily a minimum. Furthermore, and from Eq. (21), the direction of $\overline{\mathbf{p}}_o$ is not necessarily a descent direction, since $C(\overline{\mathbf{x}}_k + \overline{\mathbf{p}}_o) - C(\overline{\mathbf{x}}_k)$ can be non-negative.

As we will discuss in a subsequent section, the conditioning (singular or nonsingular nature) and the definiteness (positive/negative definiteness or indefiniteness) of the matrix $\overline{\overline{\mathbf{G}}}$ can be adjusted by the proper choice of μ.

The search direction $\overline{\mathbf{p}}_k$ which is given by the vector that solves Eq. (15) is called the Newton search direction. The method in which this vector is used as a search direction is called the full Newton search. The full Newton minimization approach is known to have a quadratic rate of convergence.

6. A MODIFIED NEWTON METHOD

In the case where $\overline{\overline{\mathbf{G}}}(\overline{\mathbf{x}}_k)$ is indefinite (but nonsingular), the direction of $\overline{\mathbf{p}}_o$, given by Eq. (17), is not necessarily a descent direction. One popular strategy in modifying Newtons method is to construct a related positive definite matrix $\overline{\overline{\mathbf{K}}}(\overline{\mathbf{x}}_k)$ when $\overline{\overline{\mathbf{G}}}(\overline{\mathbf{x}}_k)$ is indefinite. In this case the search direction is given by the solution of:

$$\overline{\overline{\mathbf{K}}} \cdot \overline{\mathbf{p}}_o = -\overline{\mathbf{g}} \tag{23}$$

One way of constructing such a positive definite matrix, is to construct $\overline{\overline{\mathbf{K}}}(\overline{\mathbf{x}}_k)$ such that it has the same eigenvectors as $\overline{\overline{\mathbf{G}}}(\overline{\mathbf{x}}_k)$ and with eigenvalues given by:

$$\mu_j = |\lambda_j| \tag{24}$$

This ensures that $\overline{\overline{\mathbf{K}}}(\overline{\mathbf{x}}_k) = \overline{\overline{\mathbf{G}}}(\overline{\mathbf{x}}_k)$ if $\overline{\overline{\mathbf{G}}}(\overline{\mathbf{x}}_k)$ is positive definite. The constructed positive definite matrix $\overline{\overline{\mathbf{K}}}(\overline{\mathbf{x}}_k)$ is hence given by the following expression:

$$\overline{\overline{\mathbf{K}}} = \sum_{j=1}^{N} |\lambda_j| \overline{\mathbf{v}}_j \overline{\mathbf{v}}_j^T, \tag{25}$$

and the corresponding search direction is given by:

$$\overline{\mathbf{p}}_o = -\overline{\overline{\mathbf{K}}}^{-1} \cdot \overline{\mathbf{g}} = -\sum_{j=1}^{N} \frac{1}{|\lambda_j|} \left(\overline{\mathbf{v}}_j^T \cdot \overline{\mathbf{g}}\right) \overline{\mathbf{v}}_j, \tag{26}$$

whose norm is given by:

$$\|\overline{\mathbf{p}}_o\| \equiv \left(\overline{\mathbf{p}}_o^T \cdot \overline{\mathbf{p}}_o\right)^{1/2} = \left[\sum_{j=1}^{N} \frac{1}{\lambda_j^2} \left(\overline{\mathbf{v}}_j^T \cdot \overline{\mathbf{g}}\right)^2\right]^{1/2}, \tag{27}$$

which is identical to that of the unmodified direction (see Eq. (18)). For this search direction, we have from Eq. (9):

$$C(\overline{\mathbf{x}}_k + \overline{\mathbf{p}}_o) - C(\overline{\mathbf{x}}_k) \approx \sum_{j=1}^{N} \frac{1}{|\lambda_j|} \left[1 - \frac{1}{2} sgn(\lambda_j)\right] \left(\overline{\mathbf{v}}_j^T \cdot \overline{\mathbf{g}}\right)^2 < 0, \tag{28}$$

thus, the direction of $\overline{\mathbf{p}}_o$, given by Eq. (26), is guaranteed to be a descent direction. In this case, we choose $\overline{\mathbf{p}}_k = \overline{\mathbf{p}}_o$.

7. THE GAUSS-NEWTON MINIMIZATION APPROACH

In the Gauss-Newton search method, one discards the second-order derivatives (to avoid the expensive computation of second derivatives), in which case the Hessian reduces to:

$$\overline{\overline{\mathbf{G}}}(\overline{\mathbf{x}}) = \overline{\overline{\mathbf{W}}}_x^T \cdot \overline{\overline{\mathbf{W}}}_x + \mu \overline{\overline{\mathbf{J}}}^T(\overline{\mathbf{x}}) \cdot \overline{\overline{\mathbf{W}}}_d^T \cdot \overline{\overline{\mathbf{W}}}_d \cdot \overline{\overline{\mathbf{J}}}(\overline{\mathbf{x}}), \tag{29}$$

which is a positive semi-definite matrix. If λ_j and $\overline{\mathbf{v}}_j$ are the eigenvalues and the corresponding orthonormal eigenvectors of the $N \times N$ real symmetric matrix $\overline{\overline{\mathbf{G}}}$

$$\overline{\overline{\mathbf{G}}} \cdot \overline{\mathbf{v}}_j = \lambda_j \overline{\mathbf{v}}_j, \qquad \overline{\mathbf{v}}_i^T \cdot \overline{\mathbf{v}}_j = \delta_{ij}, \tag{30}$$

then

$$\lambda_j = \frac{\overline{\mathbf{v}}_j^T \cdot \overline{\overline{\mathbf{G}}} \cdot \overline{\mathbf{v}}_j}{\|\overline{\mathbf{v}}_j\|^2} = \left\|\overline{\overline{\mathbf{W}}}_x \cdot \overline{\mathbf{v}}_j\right\|^2 + \mu \left\|\overline{\overline{\mathbf{W}}}_d \cdot \overline{\overline{\mathbf{J}}} \cdot \overline{\mathbf{v}}_j\right\|^2 \geq 0, \tag{31}$$

hence as previously mentioned $\overline{\overline{\mathbf{G}}}$ is a positive semi-definite matrix. As indicated before and as will be discussed in the next section, the Hessian $\overline{\overline{\mathbf{G}}}$ can be constructed to be a positive definite matrix by the proper choice of μ.

The search direction $\overline{\mathbf{p}}_k$ which is given by the vector that solves Eq. (15) with the Hessian approximated by Eq. (29) is called the Gauss-Newton search direction. The method in which this vector is used as a search direction is called the Gauss-Newton search. The Gauss-Newton minimization approach has a rate of convergence which is slightly less than quadratic but significantly better than linear. It provides quadratic convergence in the neighborhood of the minimum.

8. THE STEEPEST-DESCENT METHOD

In the case of large-scale optimization problems and to avoid the direct inversion of a large size matrix (the Hessian matrix $\overline{\overline{\mathbf{G}}}$) as is required by the Newton or Gauss-Newton minimization approaches, one can employ an iterative linear solver, e.g., a Lanczos method or a conjugate-gradient approach [18, 21]. However, these iterative approaches are more efficient for sparse systems and tend to loose their advantage for systems that have full matrices similar to the problem at hand where the Hessian is a full matrix. Furthermore, the most computationally expensive part of the inversion is not actually in inverting the Hessian matrix but is by far in the evaluation of the elements of the Hessian (or Jacobian).

An alternative approach to iterative solvers which does not require inverting the Hessian, is the steepest-descent method where the search direction is simply chosen to be along the negative of the gradient of the cost function, i.e.,

$$\overline{\mathbf{p}}_k = -\gamma_k \overline{\mathbf{g}}(\overline{\mathbf{x}}_k) \equiv -\gamma_k \overline{\mathbf{g}}_k, \tag{32}$$

where γ_k is a positive step-length which must be chosen so that the cost function $C(\overline{\mathbf{x}}_k + \overline{\mathbf{p}}_k)$ is "sufficiently decreased" after each iteration. It is thus clear that, unless the gradient vanishes, the search direction of Eq. (32) is a descent direction, since $\overline{\mathbf{p}}_k$ and $\overline{\mathbf{g}}_k$ are trivially bounded away from orthogonality ($\overline{\mathbf{g}}_k^T \cdot \overline{\mathbf{p}}_k = -\gamma_k |\overline{\mathbf{g}}_k|^2 \neq 0$).

We choose the step-length that maximizes the reduction in the cost function $C(\overline{\mathbf{x}}_k + \overline{\mathbf{p}}_k)$. Substituting Eq. (32) in Eq. (9), we obtain:

$$C(\overline{\mathbf{x}}_k + \overline{\mathbf{p}}_k) \approx C(\overline{\mathbf{x}}_k) - \gamma_k |\overline{\mathbf{g}}_k|^2 + \frac{1}{2}\gamma_k^2 \, \overline{\mathbf{g}}_k^T \cdot \overline{\overline{\mathbf{G}}}_k \cdot \overline{\mathbf{g}}_k, \tag{33}$$

whose minimum occurs when:

$$\gamma_k = \frac{|\overline{\mathbf{g}}_k|^2}{\overline{\mathbf{g}}_k^T \cdot \overline{\overline{\mathbf{G}}}_k \cdot \overline{\mathbf{g}}_k} \tag{34}$$

Thus the steepest-descent search direction is given by:

$$\begin{aligned}\overline{\mathbf{p}}_k &= \overline{\mathbf{x}}_{k+1} - \overline{\mathbf{x}}_k = -\frac{|\overline{\mathbf{g}}_k|^2}{\overline{\mathbf{g}}_k^T \cdot \overline{\overline{\mathbf{G}}}_k \cdot \overline{\mathbf{g}}_k}\overline{\mathbf{g}}_k \\ &= -\frac{|\overline{\mathbf{g}}_k|^2}{\left|\overline{\overline{\mathbf{W}}}_x \cdot \overline{\mathbf{g}}_k\right|^2 + \mu\left|\overline{\overline{\mathbf{W}}}_d \cdot \overline{\overline{\mathbf{J}}}_k \cdot \overline{\mathbf{g}}_k\right|^2}\overline{\mathbf{g}}_k \end{aligned} \tag{35}$$

This steepest-descent direction is guaranteed to yield a convergent algorithm, since:

- Unless the gradient vanishes, the search direction of Eq. (35) is a descent direction, since $\overline{\mathbf{p}}_k$ and $\overline{\mathbf{g}}_k$ are trivially bounded away from orthogonality:

$$\overline{\mathbf{g}}_k^T \cdot \overline{\mathbf{p}}_k = -\frac{|\overline{\mathbf{g}}_k|^4}{\left|\overline{\overline{\mathbf{W}}}_x \cdot \overline{\mathbf{g}}_k\right|^2 + \mu\left|\overline{\overline{\mathbf{W}}}_d \cdot \overline{\overline{\mathbf{J}}}_k \cdot \overline{\mathbf{g}}_k\right|^2} \neq 0 \tag{36}$$

- The cost function $C(\overline{\mathbf{x}}_k + \overline{\mathbf{p}}_k)$ is "sufficiently decreased" after each iteration:

$$C(\overline{\mathbf{x}}_k + \overline{\mathbf{p}}_k) - C(\overline{\mathbf{x}}_k) \approx -\frac{1}{2}\frac{|\overline{\mathbf{g}}_k|^4}{\left|\overline{\overline{\mathbf{W}}}_x \cdot \overline{\mathbf{g}}_k\right|^2 + \mu\left|\overline{\overline{\mathbf{W}}}_d \cdot \overline{\overline{\mathbf{J}}}_k \cdot \overline{\mathbf{g}}_k\right|^2} < 0 \tag{37}$$

Unfortunately, global convergence of an algorithm does not ensure that it is an efficient method. For the quadratic approximation (Eq. (9)) of the cost function, with $\overline{\overline{\mathbf{G}}}(\overline{\mathbf{x}})$ approximated by Eq. (29) (where it becomes a real symmetric positive-definite matrix), it can be shown that:

$$C(\overline{\mathbf{x}}_{k+1}) - C(\overline{\mathbf{x}}_o) \approx \frac{(\kappa-1)^2}{(\kappa+1)^2}\{C(\overline{\mathbf{x}}_k) - C(\overline{\mathbf{x}}_o)\}, \tag{38}$$

where $\overline{\mathbf{x}}_o$ is the value of the minimum of the cost function where $\overline{\mathbf{g}}(\overline{\mathbf{x}}_o)$ vanishes and $\kappa = \lambda_{max} = \lambda_{min}$ is the condition number of the Hessian $\overline{\overline{\mathbf{G}}}$. From Eq. (38), it is clear that the rate of convergence of the steepest-descent method is linear as opposed to the quadratic rate of convergence of the full Newton minimization approach. Furthermore, the factor of reduction in the error at each iteration step can be very close to unity, depending on the condition number of the Hessian matrix $\overline{\overline{\mathbf{G}}}$, so that there is a very small gain in accuracy at each iteration. With the steepest descent, the iteration is forced to traverse back and forth across the minimum rather than down towards the

minimum. In practice, the steepest-descent method may typically require hundreds of iterations to achieve very little progress towards the final solution.

One way to improve the convergence rate of the steepest descent method (also called gradient method) is by using a search direction which is a linear combination of the current gradient and the previous search direction. This type of method is called conjugate gradient. The nonlinear conjugate gradient method is modelled after the conjugate gradient method for linear equations see [22] or quadratic functionals, where it can be proven to converge in exact arithmetic in a number of steps N equal to number of unknowns. In classical conjugate gradient method (the so-called Fletcher-Reeves conjugate gradient direction) the search direction is given by

$$\overline{\mathbf{p}}_k = \gamma_k \, \overline{\mathbf{s}}_k, \tag{39}$$

where

$$\overline{\mathbf{s}}_k = -\overline{\mathbf{g}}_k + \frac{\left\|\overline{\mathbf{g}}_k\right\|^2}{\left\|\overline{\mathbf{g}}_{k-1}\right\|^2}\overline{\mathbf{s}}_{k-1}. \tag{40}$$

Again by substituting Eq. (39) in Eq. (33), the weighting parameter γ_k is given by

$$\gamma_k = \frac{\left|\overline{\mathbf{s}}_k\right|^2}{\overline{\mathbf{s}}_k^T \cdot \overline{\overline{\mathbf{G}}}_k \cdot \overline{\mathbf{s}}_k}. \tag{41}$$

Since our problem is nonlinear in term of the inverted model parameters, after a number of iterations, the effectiveness of these Fletcher-Reeves direction will reduce. This means that we must restart with the steepest descent direction and continue with new conjugate gradient directions. A method for avoiding these restarting procedures is the use of the Polak-Ribière conjugate gradient directions [22, 23]. They are obtained by changing Eq. (40) into

$$\overline{\mathbf{s}}_k = -\overline{\mathbf{g}}_k + \frac{< \overline{\mathbf{g}}_k, \overline{\mathbf{g}}_k - \overline{\mathbf{g}}_{k-1} >}{\left\|\overline{\mathbf{g}}_{k-1}\right\|^2}\overline{\mathbf{s}}_{k-1}. \tag{42}$$

When there are no changes in the inverted model parameters, we have $< \overline{\mathbf{g}}_k, \overline{\mathbf{g}}_{k-1} >= 0$ and the Polak-Ribière directions coincide with the Fletcher-Reeves directions. However, when the iterative scheme does not improve sufficiently, it means that $\overline{\mathbf{g}}_k \approx \overline{\mathbf{g}}_{k-1}$. In this case we have $\overline{\mathbf{s}}_k = -\overline{\mathbf{g}}_k$, the Polak-Ribière directions becomes the steepest descent directions, and the updating is restarted automatically.

9. LINE SEARCHES

The search vector $\overline{\mathbf{p}}_k$ of Eq. (15) is guaranteed to be a descent direction for the approximated quadratic form of the cost function of Eq. (3) [i.e., is a descent direction for the approximated cost functions of Eq. (9)]. However, this step [i.e., the new iterate $\overline{\mathbf{x}}_k + \overline{\mathbf{p}}_k$] may not sufficiently reduce the cost function of Eq. (3) and may not even decrease its value, indicating that $C(\overline{\mathbf{x}})$ is poorly modeled by a quadratic form in the vicinity of $\overline{\mathbf{x}}_k$. One approach to alleviate this problem [16] is to adopt a line-search algorithm where one searches for an appropriate real positive step-length ν_k along the search direction, $\overline{\mathbf{p}}_k$, which yields an acceptable next iterate, $\overline{\mathbf{x}}_{k+1} = \overline{\mathbf{x}}_k + \nu_k \overline{\mathbf{p}}_k$, that sufficiently decreases the cost function. The search direction can be either a Newton, a Gauss-Newton or a steepest-descent. However, in this section we will only consider the Gauss-Newton search approach.

Essentially in the line search algorithm one wants to solve the following problem:

$$\nu_k = \arg \min_{\nu} \{C(\overline{\mathbf{x}}_k + \nu \overline{\mathbf{p}}_k)\}. \tag{43}$$

Since ν is a scalar this minimization in principle can be carried out by using any nonlinear minimization routine [21]. However, if the function evaluation is expensive, a full nonlinear determination of ν_k might not be feasible. Since it will require multiple calculation of the cost function. It is therefore desirable to limit the number of such evaluations as much as possible. In this case we adopt an algorithm whereby a step-length $\nu_k > 0$ is selected which reduces the cost function such that the average rate of decrease from $C(\overline{\mathbf{x}}_k)$ to $C(\overline{\mathbf{x}}_k + \nu_k \overline{\mathbf{p}}_k)$ is at least some prescribed fraction, α, of the initial rate of decrease at $\overline{\mathbf{x}}_k$ along the direction $\overline{\mathbf{p}}_k$, i.e.,

$$C(\overline{\mathbf{x}}_k + \nu_k \overline{\mathbf{p}}_k) \leq C(\overline{\mathbf{x}}_k) + \alpha \nu_k \delta C_{k+1}, \tag{44}$$

where $0 < \alpha < 1$ is a fractional number which, in practice, is set quite small (we will set α to 10^{-4}) so that hardly more than a decrease in function value is required. δC_{k+1} is the rate of decrease of $C(\overline{\mathbf{x}})$ at $\overline{\mathbf{x}}_k$ along the direction $\overline{\mathbf{p}}_k$ and is given by:

$$\delta C_{k+1} = \frac{\partial}{\partial \nu} C(\overline{\mathbf{x}}_k + \nu \overline{\mathbf{p}}_k)\bigg|_{\nu=0} = \overline{\mathbf{g}}^T(\overline{\mathbf{x}}_k) \cdot \overline{\mathbf{p}}_k \tag{45}$$

The procedure we will adopt, is to first employ the full Gauss-Newton search step and if $\nu_k = 1$ fails to satisfy the criterion given by Eq. (44), then backtrack (i.e., reduce ν_k) along the direction of the Gauss-Newton step until an acceptable next iterate $\overline{\mathbf{x}}_{k+1} = \overline{\mathbf{x}}_k + \nu_k \overline{\mathbf{p}}_k$ is found.

If, at the $(k+1)$-th iteration, $\nu_k^{(m)}$ is the current step-length that does not satisfy the condition of Eq. (44), we compute the next backtracking step-length, $\nu_k^{(m+1)}$, by searching for the minimum of the following function:

$$f(\nu) \equiv C(\overline{\mathbf{x}}_k + \nu \overline{\mathbf{p}}_k), \tag{46}$$

which we approximate by a quadratic expression as

$$f(\nu) \approx a + b\nu + c\nu^2, \tag{47}$$

where the real constants a, b, and c are determined from the current information on the cost function $C(\overline{\mathbf{x}})$:

$$f(\nu = 0) \; = \; C(\overline{\mathbf{x}}_k), \tag{48}$$

$$\frac{df}{d\nu}(\nu = 0) \; = \; \delta C_{k+1}, \tag{49}$$

and

$$f\left(\nu = \nu_k^{(m)}\right) = C\left(\overline{\mathbf{x}}_k + \nu_k^{(m)} \overline{\mathbf{p}}_k\right), \tag{50}$$

from which we obtain

$$a \; = \; C(\overline{\mathbf{x}}_k), \tag{51}$$

$$b \; = \; \delta C_{k+1}, \tag{52}$$

and

$$c = \frac{1}{\{\nu_k^{(m)}\}^2}\left[C\left(\overline{\mathbf{x}}_k + \nu_k^{(m)} \overline{\mathbf{p}}_k\right) - C(\overline{\mathbf{x}}_k) - \nu_k^{(m)} \delta C_{k+1}\right] \tag{53}$$

Thus, $\nu_k^{(m+1)}$, which is the minimum of $f(\nu)$, for $m = 0, 1, 2, \ldots$ is given by:

$$\nu_k^{(m+1)} = -\frac{b}{2c} = -\frac{\{\nu_k^{(m)}\}^2}{2} \frac{\delta C_{k+1}}{C\left(\overline{\mathbf{x}}_k + \nu_k^{(m)} \overline{\mathbf{p}}_k\right) - C(\overline{\mathbf{x}}_k) - \nu_k^{(m)} \delta C_{k+1}}, \tag{54}$$

from which it is clear that if $C(\overline{\mathbf{x}}_k + \nu_k^{(m)} \overline{\mathbf{p}}_k) > C(\overline{\mathbf{x}}_k)$, then

$$0 < \nu_k^{(m+1)} < \frac{1}{2}\nu_k^{(m)} < \frac{1}{2^{m+1}}, \quad m = 0, 1, 2, \ldots \tag{55}$$

whereas if $C(\overline{\mathbf{x}}_k + \nu_k^{(m)} \overline{\mathbf{p}}_k) > C(\overline{\mathbf{x}}_k) + \alpha \nu_k^{(m)} \delta C_{k+1}$, then

$$0 < \nu_k^{(m+1)} < \frac{1}{2(1-\alpha)}\nu_k^{(m)} < \frac{1}{[2(1-\alpha)]^{m+1}}, \quad m = 0, 1, 2, \ldots \tag{56}$$

Thus, we start with $\nu_k^{(0)} = 1$ and proceed with the backtracking procedure of Eq. (54) until condition in Eq. (44) is satisfied. In general, it is not desirable to decrease $\nu_k^{(m+1)}$ too much since this may excessively slow down the iterative process, requiring many iterations to achieve very little progress towards the minimum. To prevent this slow down, we set $\nu_k^{(m+1)} = 0.1\,\nu_k^{(m)}$ if $\nu_k^{(m+1)} < 0.1\nu_k^{(m)}$ (but with ν_k not to decrease below 0.1, i.e., $\nu_{min} = 0.1$ to guard against too small a value of ν) and then proceed with the iteration.

To take advantage of the newly acquired information on the cost function beyond the first backtrack, one can replace the quadratic approximation of $f(\nu)$ of Eq. (47) by a cubic approximation. If ν_1 and ν_2 are two subsquent search steps, then according to the cubic approximation, the next search step is determined from:

$$\nu = \frac{-b + \sqrt{b^2 - 3a\,\delta C_{k+1}}}{3a} \tag{57}$$

where a and b are given by:

$$\begin{bmatrix} a \\ b \end{bmatrix} = \frac{1}{\nu_2 - \nu_1} \begin{bmatrix} 1/\nu_2^2 & -1/\nu_1^2 \\ -\nu_1/\nu_2^2 & \nu_2/\nu_1^2 \end{bmatrix} \begin{bmatrix} f(\nu_2) - \nu_2\,\delta C_{k+1} - C(\overline{\mathbf{x}}_k) \\ f(\nu_1) - \nu_1\,\delta C_{k+1} - C(\overline{\mathbf{x}}_k) \end{bmatrix}. \tag{58}$$

10. THE CHOICE OF THE LAGRANGE MULTIPLIER

There exist several criteria by which one can select the Lagrange multiplier μ. One such criterion [20] is to substitute $\mu = e^{\vartheta}$ or $\mu = \vartheta^2$ (i.e., any transform which enforces μ's positivity) and to assign ϑ a real solution of the following nonlinear algebraic equation:

$$\left\| \overline{\overline{\mathbf{W}}}_d \cdot \overline{\mathbf{e}}\,[\overline{\mathbf{x}}_k + \overline{\mathbf{p}}_k(\vartheta)] \right\| = \chi, \tag{59}$$

which results from setting $\partial C(\overline{\mathbf{x}}, \mu)/\partial\mu = 0$. This method is known as the discrepancy principle. A different strategy for selecting μ is also described in [19], in which a steepest descent method is applied in the initial steps of the iteration process. This corresponds to choosing small values for μ, hence, putting more weight on the second term of the cost function of Eq. (3) since the first term is only crudely approximated by the quadratic model of Eq. (9). As the iteration progresses, the reconstructed model approaches its true value, thus resulting in Eq. (9) becoming more accurate, and hence more weight (corresponding to larger values of μ) should be placed on minimizing

the first term of the cost function (this resembles the Gauss-Newton method).

One of the working criteria for choosing μ is to bound it by the following inequality

$$\max\{\text{small } \tau_m{}'s\} \ll 1/\mu \ll \min\{\text{large } \tau_m{}'s\}, \tag{60}$$

where τ_m are the eigenvalues of the positive-definite real symmetric matrix:

$$\overline{\overline{\mathbf{H}}} = \{\overline{\overline{\mathbf{W}}}_x^{-1}\}^T \cdot \overline{\overline{\mathbf{J}}}^T(\overline{\mathbf{x}}) \cdot \overline{\overline{\mathbf{W}}}_d^T \cdot \overline{\overline{\mathbf{W}}}_d \cdot \overline{\overline{\mathbf{J}}}(\overline{\mathbf{x}}) \cdot \overline{\overline{\mathbf{W}}}_x^{-1}. \tag{61}$$

The second part of the above inequality guarantees that the spectral content of the inversion operator remains unaltered, whereas the first part of the inequality regularizes the inversion problem by suppressing the null-space of the inversion operator. To illustrate this, we define $\overline{\mathbf{u}}_m$ as the orthonormal eigenvector corresponding to the eigenvalue τ_m of the matrix $\overline{\overline{\mathbf{H}}}$. In this case the solution to Eq. (15) with the Hessian approximated by Eq. (30) is given by:

$$\overline{\mathbf{p}}_k = \overline{\mathbf{x}}_{k+1} - \overline{\mathbf{x}}_k = -\sum_{j=1}^{N} \frac{1}{1+\mu\tau_j} \left(\overline{\mathbf{u}}_j^T \cdot \overline{\overline{\mathbf{W}}}_x^T \cdot \overline{\mathbf{g}}_k \right) \overline{\overline{\mathbf{W}}}_x \cdot \overline{\mathbf{u}}_j. \tag{62}$$

Hence, according to the criterion given in Eq. (60), we obtain:

$$\begin{aligned} \overline{\mathbf{p}}_k \;=\; & -\sum_{large\{\tau_j's\}} \frac{1}{\tau_j} \left(\overline{\mathbf{u}}_j^T \cdot \overline{\overline{\mathbf{W}}}_x^T \cdot \overline{\mathbf{g}}_k/\mu \right) \overline{\overline{\mathbf{W}}}_x \cdot \overline{\mathbf{u}}_j \\ & -\sum_{small\{\tau_j's\}} \mu \left(\overline{\mathbf{u}}_j^T \cdot \overline{\overline{\mathbf{W}}}_x^T \cdot \overline{\mathbf{g}}_k/\mu \right) \overline{\overline{\mathbf{W}}}_x \cdot \overline{\mathbf{u}}_j. \end{aligned} \tag{63}$$

From this expression it is clear that μ acts as a damping factor that suppresses (or filters out) the contributions of the small eigenvalues which have the tendency of magnifying the noise in the data and which may cause undesired large swings in $\overline{\mathbf{p}}_k$. Hence, the role of μ (which satisfies condition Eq. (60)) is to damp out the large deviations of the $(k+1)$-th iterate from the k-th iterate and to safeguard against unwanted large changes in $\overline{\mathbf{p}}_k$ due to noise in the data. In the implementation we choose the Lagrange multiplier as follows:

$$\frac{1}{\mu} = \beta \max_{\forall m}\{\tau_m\}, \qquad \text{if } \frac{\min_{\forall m}\{\tau_m\}}{\max_{\forall m}\{\tau_m\}} < \beta, \tag{64}$$

where β is a constant value which is to be determined by numerical implementation.

To illustrate the role of the ill-conditioning of the Hessian on the error magnification due to the noise in the data, we perturb the right-hand-side of Eq. (15) by $\delta\,\overline{\mathbf{g}}$ and compute the corresponding perturbation $\delta\,\overline{\mathbf{p}}$ in $\overline{\mathbf{p}}$:

$$\overline{\overline{\mathbf{G}}} \cdot \delta\,\overline{\mathbf{p}} \approx -\delta\,\overline{\mathbf{g}}, \tag{65}$$

from which we obtain:

$$\|\delta\,\overline{\mathbf{p}}\| \leq \|\overline{\overline{\mathbf{G}}}^{-1}\|\,\|\delta\,\overline{\mathbf{g}}\|\,, \tag{66}$$

and since:

$$\|\overline{\mathbf{g}}\| \leq \|\overline{\overline{\mathbf{G}}}\|\,\|\overline{\mathbf{p}}\|\,, \tag{67}$$

we get:

$$\frac{\|\delta\,\overline{\mathbf{p}}\|}{\|\overline{\mathbf{p}}\|} \leq \|\overline{\overline{\mathbf{G}}}\|\|\overline{\overline{\mathbf{G}}}^{-1}\|\frac{\|\delta\,\overline{\mathbf{g}}\|}{\|\overline{\mathbf{g}}\|} = \mathrm{cond}(\overline{\overline{\mathbf{G}}})\frac{\|\delta\,\overline{\mathbf{g}}\|}{\|\overline{\mathbf{g}}\|}, \tag{68}$$

which shows that the relative error in the estimator, $\|\delta\,\overline{\mathbf{p}}\|\,/\,\|\overline{\mathbf{p}}\|$, can be excessively magnified if the condition number of the Hessian is large.

The effect of the presence of the small $\{\tau_j' s\}$ in the spectrum of $\overline{\overline{\mathbf{H}}}$ is to cause the minimum of the cost function to be a flat valley (the presence of the noise in data will then result in multiple false local minima). The role of the regularizing parameter μ is to sharpen the minimum and suppress other false local minima.

The eigenvectors with small eigenvalues are the source of nonuniqueness in the inversion problem (i.e., the cause of the ill-posedness of the inversion). The addition of an arbitrary weighted sum of these eigenvectors to any solution (an inverted model) of the inverse problem may still make the resulting model fit the data. Thus the knowledge of the observed data will not enable us to construct the part of the model represented by these eigenvectors with small eigenvalues. Hence, these parts must be recovered from information other than that contained in the data. One approach to recover these components, is to incorporate all possible *a priori* information concerning the model of interest (in the construction of the reference model $\overline{\mathbf{y}}_R$ of Eq. (2) and in the introduction of the regularization term in the cost function of Eq. (3)), thus narrowing down the class of admissible solutions. If such *a priori* information is not available, one may choose to tolerate the nonuniqueness, if the class of admissible solutions still contains useful and decisive information about the unknown model.

It can be shown (using a stationary-phase argument) that the eigenvectors with small eigenvalues possess high spatial frequencies (they tend to be highly oscillatory). By suppressing these eigenvectors (as in Eq. (63)), in effect, we are constructing models which have the least degree of complexity and which best fit the data.

To enforce the inequality of Eq. (60) without evaluating the eigenvalues of the Hessian (which tend to be a computationally expensive operation), and guided by the above discussion, a possible alternative criterion which is similar to Eq. (60) but is different in implementation is to adaptively vary as the iteration proceeds. Recently [12] introduced an automatic way to choose the regularization parameter. This approach included the regularization as a multiplicative factor in the cost function. As the results the regularization parameter is founded to be proportional to the original/non-regularized cost function. Together with conjugate gradient type algorithm, this regularization technique has been shown to be very effective in inverting synthetic and experimental data, see [5] and [13]. In this paper, we adapted this multiplicative regularization technique for Newton type algorithm. To that end, we modified the cost function given in Eq. (3) as follows:

$$C_k(\overline{\mathbf{x}}) = F(\overline{\mathbf{x}})R_k(\overline{\mathbf{x}}), \tag{69}$$

where $F(\overline{\mathbf{x}})$ is the original cost functional,

$$F(\overline{\mathbf{x}}) = \frac{1}{2}\left\|\overline{\overline{\mathbf{W}}}_d \cdot \overline{\mathbf{e}}(\overline{\mathbf{x}})\right\|^2 \tag{70}$$

and $R_k(\overline{\mathbf{x}})$ is the regularization factor chosen to be

$$R_k(\overline{\mathbf{x}}) = \eta_k\left(\left\|\overline{\overline{\mathbf{W}}}_x(\overline{\mathbf{x}} - \overline{\mathbf{x}}_p)\right\|^2 + \|\delta\|^2\right), \quad \eta_k = \frac{1}{\left\|\overline{\overline{\mathbf{W}}}_x(\overline{\mathbf{x}}_k - \overline{\mathbf{x}}_p)\right\|^2 + \|\delta\|^2}, \tag{71}$$

in which δ is a constant parameter which is to be determined by numerical experimentation. One should note that the parameter δ if far more insensitive than the Lagrange multiplier. For all of our example in this paper we use only one value of parameter δ. From observation of Eq. (71) we expect that this parameter is only dependent on the type of measurements. Thus, for our numerical examples which are electromagnetic measurements at induction frequency we use the same value of δ for all the examples. The normalization in the regularization factor $R_k(\overline{\mathbf{x}})$ is chosen so that $R_k(\overline{\mathbf{x}} = \overline{\mathbf{x}}_k) = 1$. This means that at the end of the optimization process the value of the regularization parameter will be close to unity.

Following the analysis in Section 5, the gradient of the cost functional $C(\overline{\mathbf{x}})$ is given by:

$$\overline{\mathbf{g}}_k = \overline{\mathbf{g}}(\overline{\mathbf{x}} = \overline{\mathbf{x}}_k) = \overline{\overline{\mathbf{J}}}^T(\overline{\mathbf{x}}_k)\cdot\overline{\overline{\mathbf{W}}}_d^T\cdot\overline{\overline{\mathbf{W}}}_d\cdot\overline{\mathbf{e}}(\overline{\mathbf{x}}_k) + \eta_k F(\overline{\mathbf{x}}_k)\overline{\overline{\mathbf{W}}}_x^T\cdot\overline{\overline{\mathbf{W}}}_x\cdot(\overline{\mathbf{x}}_k - \overline{\mathbf{x}}_p), \tag{72}$$

and the Hessian of the cost function $C(\overline{\mathbf{x}})$ which is given by:

$$\begin{aligned}\overline{\overline{\mathbf{G}}}_k &= \overline{\overline{\mathbf{G}}}(\overline{\mathbf{x}} = \overline{\mathbf{x}}_k) \\ &= \overline{\overline{\mathbf{J}}}^T(\overline{\mathbf{x}}_k) \cdot \overline{\overline{\mathbf{W}}}_d^T \cdot \overline{\overline{\mathbf{W}}}_d \cdot \overline{\overline{\mathbf{J}}}(\overline{\mathbf{x}}_k) + \overline{\overline{\mathbf{Q}}}(\overline{\mathbf{x}}_k) + \eta_k F(\overline{\mathbf{x}}_k) \overline{\overline{\mathbf{W}}}_x^T \cdot \overline{\overline{\mathbf{W}}}_x \\ &\quad + \eta_k \left[\overline{\overline{\mathbf{W}}}_x^T \cdot \overline{\overline{\mathbf{W}}}_x \cdot (\overline{\mathbf{x}}_k - \overline{\mathbf{x}}_p) \right]^T \cdot \overline{\overline{\mathbf{J}}}^T(\overline{\mathbf{x}}_k) \cdot \overline{\overline{\mathbf{W}}}_d^T \cdot \overline{\overline{\mathbf{W}}}_d \cdot \overline{\mathbf{e}}(\overline{\mathbf{x}}_k) \\ &\quad + \eta_k \left[\overline{\overline{\mathbf{J}}}^T(\overline{\mathbf{x}}_k) \cdot \overline{\overline{\mathbf{W}}}_d^T \cdot \overline{\overline{\mathbf{W}}}_d \cdot \overline{\mathbf{e}}(\overline{\mathbf{x}}_k) \right]^T \cdot \overline{\overline{\mathbf{W}}}_x^T \cdot \overline{\overline{\mathbf{W}}}_x \cdot (\overline{\mathbf{x}}_k - \overline{\mathbf{x}}_p), \quad (73)\end{aligned}$$

where $\overline{\overline{\mathbf{Q}}}(\overline{\mathbf{x}}_k)$ is equal to the one given in Eq. (73). In our numerical implementation, due to lack of a priori information and constrained the optimization process so that it does not make a huge jump within two successive iteration, we choose $\overline{\mathbf{x}}_p$ equal to the value of $\overline{\mathbf{x}}$ at the previous iteration. For this particular choice of $\overline{\mathbf{x}}_p$, this multiplicative regularization technique is equivalent with the standard additive one if we choose the Lagrange multipliers μ to vary as the iteration proceeds according to:

$$\frac{1}{\mu} = \frac{F(\overline{\mathbf{x}}_k)}{\|\delta\|^2}. \quad (74)$$

The structure of this procedure is such that it will minimize the regularization factor with a large weighting parameter in the beginning of the optimization process, because the value of $F(\overline{\mathbf{x}}_k)$ is still large. In this case the search direction is predominantly steepest descent which is the more appropriate approach to use in the initial steps of the iteration process since it has the tendency of suppressing large swings in the search direction. As the iteration proceeds the optimization process will gradually minimize more the error in the original cost functional when the regularization factor $R_k(\overline{\mathbf{x}})$ remains a nearly constant value close to one. In this case the search direction corresponds to Newton search method which is the more appropriate approach to use as we get closer to the minimum of the original cost functional $F(\overline{\mathbf{x}})$ where the quadratic model for the cost functional becomes more accurate. if noise is present in the data, the original cost functional $F(\overline{\mathbf{x}})$ will remain at a certain value during the optimization process, the weight of the regularization factor will be more significant. Hence, the noise will, at all times, be suppressed in the inversion process and we automatically fulfill the need of a larger regularization when the data contains noise as suggested by [24] and [25].

11. UPDATE FORMULAS FOR THE HESSIAN

The methods discussed above are quite powerful, but it still has several disadvantage. One drawback is that the Jacobian matrix $\overline{\overline{\mathbf{J}}}(\overline{\mathbf{x}})$ is needed. In realistic problem, the geometry involved is complicated and hence the analytic expression for the derivatives in Eq. (11) are unavailable. It means that these derivatives should be computed numerically. If function evaluation is expensive, then the cost of finite-difference determination of the Jacobian might be prohibitive. There are quasi-Newton methods that provide cheap approximation to the Jacobian for zero finding. In this section we summarize various updating schemes of the Hessian without directly computing the Hessian [26].

Let $\overline{\mathbf{s}}_k = \nu_k \overline{\mathbf{p}}_k = \overline{\mathbf{x}}_{k+1} - \overline{\mathbf{x}}_k$ be the step taken from the k-th iterate, $\overline{\mathbf{x}}_k$, to obtain the $(k+1)$-th iterate, $\overline{\mathbf{x}}_{k+1}$. Expanding the gradient vector $\overline{\mathbf{g}}$ about the k-th iterate ($\overline{\mathbf{x}}_k$) in a Taylor series, we obtain:

$$\overline{\mathbf{g}}(\overline{\mathbf{x}}_k + \overline{\mathbf{s}}_k) = \overline{\mathbf{g}}(\overline{\mathbf{x}}_k) + \overline{\overline{\mathbf{G}}}(\overline{\mathbf{x}}_k) \cdot \overline{\mathbf{s}}_k + \cdots \tag{75}$$

or

$$\overline{\mathbf{g}}_{k+1} = \overline{\mathbf{g}}_k + \overline{\overline{\mathbf{G}}}_k \cdot \overline{\mathbf{s}}_k + \cdots \tag{76}$$

In any of the following update formulas, the updated Hessian, denoted by $\overline{\overline{\mathbf{U}}}_{k+1}$, will be required to satisfy Eq. (76) approximated by the first two terms in the Taylor series expansion, i.e.,

$$\overline{\mathbf{g}}_{k+1} = \overline{\mathbf{g}}_k + \overline{\overline{\mathbf{U}}}_{k+1} \cdot \overline{\mathbf{s}}_k, \tag{77}$$

or

$$\overline{\overline{\mathbf{U}}}_{k+1} \cdot \overline{\mathbf{s}}_k = \overline{\mathbf{q}}_k \equiv \overline{\mathbf{g}}_{k+1} - \overline{\mathbf{g}}_k, \tag{78}$$

This condition is referred to as the quasi-Newton condition. The updated Hessian, $\overline{\overline{\mathbf{U}}}_{k+1}$, is obtained by updating the previous approximate Hessian, $\overline{\overline{\mathbf{U}}}_k$, to take into account the newly acquired information on the curvature of the cost function (contained in Eq. (78)). The Hessian $\overline{\overline{\mathbf{U}}}_k$ is the approximate Hessian at the begining of the k-th iteration which reflects the curvature information that has already been accumulated. $\overline{\overline{\mathbf{U}}}_k$ was used to determine the k-th Gauss-Newton search direction, $\overline{\mathbf{p}}_k$, through Eq. (15):

$$\overline{\overline{\mathbf{U}}}_k \cdot \overline{\mathbf{p}}_k = -\overline{\mathbf{g}}_k \quad \text{or} \quad \overline{\overline{\mathbf{U}}}_k \cdot \overline{\mathbf{s}}_k = -\nu_k \overline{\mathbf{g}}_k. \tag{79}$$

11.1. The Rank-One Matrix Update

In this update, $\overline{\overline{\mathbf{U}}}_{k+1}$ is constructed from $\overline{\overline{\mathbf{U}}}_k$ by adding a symmetric matrix of rank-one (since the Hessian matrix is symmetric):

$$\overline{\overline{\mathbf{U}}}_{k+1} = \overline{\overline{\mathbf{U}}}_k + \overline{\mathbf{u}}\,\overline{\mathbf{u}}^T, \tag{80}$$

for some vector $\overline{\mathbf{u}}$. Substituting into the quasi-Newton condition Eq. (78), we get:

$$\left(\overline{\mathbf{u}}^T \cdot \overline{\mathbf{s}}_k\right) \overline{\mathbf{u}} = \overline{\mathbf{q}}_k - \overline{\overline{\mathbf{U}}}_k \cdot \overline{\mathbf{s}}_k, \tag{81}$$

from which we can easily deduce that:

$$\overline{\mathbf{u}} = \frac{1}{\left[\overline{\mathbf{s}}_k^T \cdot \left(\overline{\mathbf{q}}_k - \overline{\overline{\mathbf{U}}}_k \cdot \overline{\mathbf{s}}_k\right)\right]^{1/2}} \left(\overline{\mathbf{q}}_k - \overline{\overline{\mathbf{U}}}_k \cdot \overline{\mathbf{s}}_k\right), \tag{82}$$

and hence, the update formula for $\overline{\overline{\mathbf{U}}}_{k+1}$ is given by:

$$\begin{aligned}\overline{\overline{\mathbf{U}}}_{k+1} &= \overline{\overline{\mathbf{U}}}_k + \frac{1}{\overline{\mathbf{s}}_k^T \cdot \left(\overline{\mathbf{q}}_k - \overline{\overline{\mathbf{U}}}_k \cdot \overline{\mathbf{s}}_k\right)} \left(\overline{\mathbf{q}}_k - \overline{\overline{\mathbf{U}}}_k \cdot \overline{\mathbf{s}}_k\right) \left(\overline{\mathbf{q}}_k - \overline{\overline{\mathbf{U}}}_k \cdot \overline{\mathbf{s}}_k\right)^T \\ &= \overline{\overline{\mathbf{U}}}_k + \frac{1}{\overline{\mathbf{s}}_k^T \cdot (\overline{\mathbf{q}}_k + \nu_k \overline{\mathbf{g}}_k)} (\overline{\mathbf{q}}_k + \nu_k \overline{\mathbf{g}}_k) (\overline{\mathbf{q}}_k + \nu_k \overline{\mathbf{g}}_k)^T, \end{aligned} \tag{83}$$

where we assume that $\overline{\overline{\mathbf{U}}}_k \cdot \overline{\mathbf{s}}_k \neq \overline{\mathbf{q}}_k$ and $\overline{\mathbf{s}}_k^T \cdot (\overline{\mathbf{q}}_k - \overline{\overline{\mathbf{U}}}_k \cdot \overline{\mathbf{s}}_k) \neq 0$. This update is called the **Broyden symmetric rank-one update**.

Note that the quasi-Newton condition in Eq. (78) will continue to hold even if further rank-one matrices of the form $\overline{\mathbf{v}}\,\overline{\mathbf{v}}^T$ are added to $\overline{\overline{\mathbf{U}}}_{k+1}$ as long as the vector $\overline{\mathbf{v}}$ is orthogonal to $\overline{\mathbf{s}}_k$ (i.e., $\overline{\mathbf{v}}^T \cdot \overline{\mathbf{s}}_k = 0$). Of course, the elements of $\overline{\overline{\mathbf{U}}}_{k+1}$ will change by each additional matrix.

11.2. The Rank-Two Matrix Updates

In the following updates, $\overline{\overline{\mathbf{U}}}_{k+1}$ is constructed from $\overline{\overline{\mathbf{U}}}_k$ by adding a symmetric matrix of rank-two. The general form of a symmetric rank-two matrix update is given by:

$$\overline{\overline{\mathbf{U}}}_{k+1} = \overline{\overline{\mathbf{U}}}_k + \overline{\mathbf{u}}\,\overline{\mathbf{u}}^T + \alpha \left\{\overline{\mathbf{u}}\,\overline{\mathbf{v}}^T + \overline{\mathbf{v}}\,\overline{\mathbf{u}}^T\right\} + \overline{\mathbf{v}}\,\overline{\mathbf{v}}^T, \tag{84}$$

where $\overline{\mathbf{u}}$ and $\overline{\mathbf{v}}$ are any two different vectors and α is a scalar. The Hessian updates discussed in this section can all be derived from

the following formula which satisfies the quasi-Newton condition in Eq. (78):

$$\overline{\overline{\mathbf{U}}}_{k+1} = \overline{\overline{\mathbf{U}}}_k + \frac{1}{\overline{\mathbf{s}}_k^T \cdot \overline{\mathbf{v}}} \left\{ (\overline{\mathbf{q}}_k + \nu_k \overline{\mathbf{g}}_k) \overline{\mathbf{v}}^T + \overline{\mathbf{v}} (\overline{\mathbf{q}}_k + \nu_k \overline{\mathbf{g}}_k)^T \right\} - \frac{\overline{\mathbf{s}}_k^T \cdot (\overline{\mathbf{q}}_k + \nu_k \overline{\mathbf{g}}_k)}{(\overline{\mathbf{s}}_k^T \cdot \overline{\mathbf{v}})^2} \overline{\mathbf{v}}\, \overline{\mathbf{v}}^T, \tag{85}$$

for any arbitrary vector $\overline{\mathbf{v}}$ which is not orthogonal to $\overline{\mathbf{s}}_k$ (i.e., $\overline{\mathbf{s}}_k^T \cdot \overline{\mathbf{v}} \neq 0$).

Note that the rank-one matrix update of Eq. (83) can be derived from the rank-two matrix update of Eq. (85) by setting $\overline{\mathbf{v}} = \overline{\mathbf{q}}_k + \nu_k \overline{\mathbf{g}}_k$.

11.2.1. The Powell-Symmetric-Broyden (PSB) Update

In this update, the arbitrary vector $\overline{\mathbf{v}}$ is chosen to be $\overline{\mathbf{s}}_k$ to obtain:

$$\overline{\overline{\mathbf{U}}}_{k+1} = \overline{\overline{\mathbf{U}}}_k + \frac{1}{\|\overline{\mathbf{s}}_k\|^2} \left\{ (\overline{\mathbf{q}}_k + \nu_k \overline{\mathbf{g}}_k) \overline{\mathbf{s}}_k^T + \overline{\mathbf{s}}_k (\overline{\mathbf{q}}_k + \nu_k \overline{\mathbf{g}}_k)^T \right\} - \frac{\overline{\mathbf{s}}_k^T \cdot (\overline{\mathbf{q}}_k + \nu_k \overline{\mathbf{g}}_k)}{\|\overline{\mathbf{s}}_k\|^4} \overline{\mathbf{s}}_k \overline{\mathbf{s}}_k^T. \tag{86}$$

11.2.2. The Davidson-Fletcher-Powell (DFP) Update

In this update, the arbitrary vector $\overline{\mathbf{v}}$ is chosen to be $\overline{\mathbf{q}}_k$ to obtain:

$$\overline{\overline{\mathbf{U}}}_{k+1} = \overline{\overline{\mathbf{U}}}_k + \frac{\nu_k}{\overline{\mathbf{s}}_k^T \cdot \overline{\mathbf{g}}_k} \overline{\mathbf{g}}_k\, \overline{\mathbf{g}}_k^T + \frac{1}{\overline{\mathbf{s}}_k^T \cdot \overline{\mathbf{q}}_k} \overline{\mathbf{q}}_k\, \overline{\mathbf{q}}_k^T - \nu_k (\overline{\mathbf{s}}_k^T \cdot \overline{\mathbf{g}}_k) \overline{\mathbf{w}}_k\, \overline{\mathbf{w}}_k^T, \tag{87}$$

where

$$\overline{\mathbf{w}}_k \equiv \frac{1}{\overline{\mathbf{s}}_k^T \cdot \overline{\mathbf{q}}_k} \overline{\mathbf{q}}_k - \frac{1}{\overline{\mathbf{s}}_k^T \cdot \overline{\mathbf{g}}_k} \overline{\mathbf{g}}_k. \tag{88}$$

Note that $\overline{\mathbf{w}}_k$ is orthogonal to $\overline{\mathbf{s}}_k$ (i.e., $\overline{\mathbf{s}}_k^T \cdot \overline{\mathbf{w}}_k = 0$).

11.2.3. The Broyden-Fletcher-Goldfarb-Shanno (BFGS) Update

Since $\overline{\mathbf{w}}_k$ is orthogonal to $\overline{\mathbf{s}}_k$, consequently, any multiple of the rank-one matrix $\overline{\mathbf{w}}_k\, \overline{\mathbf{w}}_k^T$ can be added to $\overline{\overline{\mathbf{U}}}_{k+1}$ without violating the quasi-Newton condition in Eq. (78). This leads to the following update formula:

$$\overline{\overline{\mathbf{U}}}_{k+1} = \overline{\overline{\mathbf{U}}}_k + \frac{\nu_k}{\overline{\mathbf{s}}_k^T \cdot \overline{\mathbf{g}}_k} \overline{\mathbf{g}}_k \overline{\mathbf{g}}_k^T + \frac{1}{\overline{\mathbf{s}}_k^T \cdot \overline{\mathbf{q}}_k} \overline{\mathbf{q}}_k \overline{\mathbf{q}}_k^T - \alpha \nu_k (\overline{\mathbf{s}}_k^T \cdot \overline{\mathbf{g}}_k) \overline{\mathbf{w}}_k \overline{\mathbf{w}}_k^T, \tag{89}$$

where α is any scalar. In the case of the Broyden-Fletcher-Goldfarb-Shanno (BFGS) update, α is set to zero, to obtain:

$$\overline{\overline{\mathbf{U}}}_{k+1} = \overline{\overline{\mathbf{U}}}_k + \frac{\nu_k}{\overline{\mathbf{s}}_k^T \cdot \overline{\mathbf{g}}_k} \overline{\mathbf{g}}_k \, \overline{\mathbf{g}}_k^T + \frac{1}{\overline{\mathbf{s}}_k^T \cdot \overline{\mathbf{q}}_k} \, \overline{\mathbf{q}}_k \, \overline{\mathbf{q}}_k^T. \tag{90}$$

Although it is possible to start the approximation of the Hessian $\overline{\overline{\mathbf{U}}}_k$ using simply the identity matrix, in the implementation we prefer to spend the first N function evaluations on finite-difference approximation to initialize $\overline{\overline{\mathbf{U}}}_k$. Finally one should note that since $\overline{\overline{\mathbf{U}}}_k$ is not the exact Jacobian, we are not guaranteed that $\overline{\mathbf{p}}_k$ is a descent direction for the cost function in Eq. (9). Thus the line search algorithm can fail to return a suitable step if $\overline{\overline{\mathbf{U}}}_k$ wanders far from the true Jacobian. In this case we reinitialize $\overline{\overline{\mathbf{U}}}_k$ using finite-difference approximation through function evaluations.

12. CRITERION FOR TERMINATING THE ITERATION

The iteration process will stop if one of the following conditions occurs first:

- The root mean square of the relative error reaches a prescribed value η determined from estimates of noise in the data, i.e.,

$$\left\{ \frac{1}{M} \left\| \overline{\mathbf{e}} \right\|^2 \right\}^{1/2} \leq \eta, \tag{91}$$

  where η is a predetermined *a priori* information that has to be provided by the user. In the hypothetical case of noise-free data, $\eta = 0$.
- The differences between two successive iterates, $(k+1)$-th and k-th, of the model parameters are within a prescribed tolerance factor, *tol*, of the current iterate:

$$\sum_{j=1}^{N} |x_{j,k+1} - x_{j,k}| \leq tol \times \sum_{j=1}^{N} |x_{j,k+1}|. \tag{92}$$

- The number of iterations exceeds a prescribed maximum.
- The difference between the cost function at two successive iterates, $(k+1)$-th and k-th, of the model parameters is within a prescribed tolerance factor, *tole*, of the cost function at the current iterate:

$$|C(\overline{\mathbf{x}}_{k+1}) - C(\overline{\mathbf{x}}_k)| \leq tole \times C(\overline{\mathbf{x}}_{k+1}). \tag{93}$$

13. REGULARIZATION

As previously mentioned, the inversion problem is invariably ill-posed. One approach of narrowing down the solution of the inverse problem is to introduce *a priori* information through a regularization term in the cost function, as was done in the cost function of Eq. (3). In this section we will discuss several other choices of the regularization term. We generalize the cost function of Eq. (3), by redefining it as follows:

$$C(\overline{\mathbf{x}}) = \frac{1}{2}\left[\mu\left\{\left\|\overline{\overline{\mathbf{W}}}_d \cdot \overline{\mathbf{e}}(\overline{\mathbf{x}})\right\|^2 - \chi^2\right\} + R(\overline{\mathbf{x}}, \overline{\mathbf{x}}_p)\right], \tag{94}$$

where $R(\overline{\mathbf{x}}, \overline{\mathbf{x}}_p)$ is a generalized regularization term whose selection will bias the class of models to be inverted.

13.1. L_1-Norm

The weighted L_q-norm of a vector $\overline{\mathbf{u}}$ is defined as:

$$\left\|\overline{\overline{\mathbf{W}}} \cdot \overline{\mathbf{u}}\right\|_q = \left[\sum_{i=1}^{N}\left|\sum_{j=1}^{N} W_{ij}u_j\right|^q\right]^{1/q} \tag{95}$$

The L_1-norm regularization term is hence given by:

$$R(\overline{\mathbf{x}}, \overline{\mathbf{x}}_p) = \left\|\overline{\overline{\mathbf{W}}}_x \cdot (\overline{\mathbf{x}} - \overline{\mathbf{x}}_p)\right\|_1 = \sum_{i=1}^{N}\left|\sum_{j=1}^{N} W_{ij}(x_j - x_{pj})\right| \tag{96}$$

Unlike the L_2-norm regularization term of Eq. (3), the L_1-norm will allow the inverted parameters to acquire large differences (large contrasts) relative to each other. However, it should be cautioned that such a regularization term may introduce spurious artifacts due to the ill-posedness of the inversion problem combined with the presence of noise in data.

The L_1-norm regularization term, has the disadvantage that its derivative does not exist and therefore such an approach is not compatible with the Newton-type methods where derivatives are required. An alternative approach to Newton-type methods is to employ linear programming schemes.

13.2. Maximum Entropy

In the maximum entropy method, the regularization term is defined by:

$$R(\overline{\mathbf{x}}, \overline{\mathbf{x}}_p) = -\sum_{j=1}^{N} r_j \ln(r_j), \tag{97}$$

where

$$r_j = \frac{t_j}{T}, \qquad T = \sum_{j=1}^{N} t_j \quad \Longrightarrow \quad \sum_{j=1}^{N} r_j = 1,$$

and the vector $\overline{\mathbf{t}}$ is given by:

$$\overline{\mathbf{t}} = \overline{\overline{\mathbf{W}}}_x \cdot (\overline{\mathbf{x}} - \overline{\mathbf{x}}_p)$$

The advantage of the maximum entropy regularization term is that it provides the most probable estimation which is consistent with the measured data. Similar to the L_1-norm regularization term, but to a lesser extent (because of its inherent smoothing effect), it can also allow the inverted parameters to acquire relatively large differences (large contrasts), however, without the appearance of spurious artifacts resulting from the ill-posedness of the inversion problem and the presence of noise in data. Another property which the maximum entropy of Eq. (97) possess is that it automatically enforces positivity on the model parameters without the necessity of imposing additional constraints. This can be seen from the observation that $R(\overline{\mathbf{x}}, \overline{\mathbf{x}}_p)$ of Eq. (97) has an infinite slope as any of the model parameters approaches zero.

14. THE WEIGHTED LEAST SQUARES MINIMIZATION IN THE FRAMEWORK OF STOCHASTIC ESTIMATION

14.1. Preliminaries

In the stochastic framework, each data point m_i is assumed to be different from the corresponding simulated response $S_i(\overline{\mathbf{x}})$ by a measurement error (or noise) denoted by e_i (see Eq. (1)):

$$\overline{\mathbf{e}} = \overline{\mathbf{S}}(\overline{\mathbf{x}}) - \overline{\mathbf{m}}. \tag{98}$$

One assumes that the measurement noise $\overline{\mathbf{e}}$ is a random variable represented by a normal (Gaussian) distribution with zero mean. In

this case the probability distribution function for the measurement noise $\overline{\mathbf{e}}$ is given by:

$$\begin{aligned} P(\overline{\mathbf{e}}) &= P_e \exp\left(-\frac{1}{2}\overline{\mathbf{e}}^T \cdot \overline{\overline{\mathbf{\Lambda}}}_e^{-1} \cdot \overline{\mathbf{e}}\right) \\ &= P_e \exp\left\{-\frac{1}{2}\left[\overline{\mathbf{m}} - \overline{\mathbf{S}}(\overline{\mathbf{x}})\right]^T \cdot \overline{\overline{\mathbf{\Lambda}}}_e^{-1} \cdot \left[\overline{\mathbf{m}} - \overline{\mathbf{S}}(\overline{\mathbf{x}})\right]\right\}, \end{aligned} \quad (99)$$

where P_e is a normalization constant and $\overline{\overline{\mathbf{\Lambda}}}_e$ is the *noise covariance matrix*. The above Gaussian distribution assumption will breakdown if the simulated response, $\overline{\mathbf{S}}(\overline{\mathbf{x}})$, does not exactly and fully represent the physics of the measurement $\overline{\mathbf{m}}$. In this case, the error will be biased by the degree of inaccuracy in the simulated response and hence will not possess a normal distribution.

The probability distribution function of Eq. (99) is also the conditional probability density $P(\overline{\mathbf{m}}|\overline{\mathbf{x}})$ of the data $\overline{\mathbf{m}}$ given the model $\overline{\mathbf{x}}$:

$$P(\overline{\mathbf{m}}|\overline{\mathbf{x}}) = P_e \exp\left\{-\frac{1}{2}\left[\overline{\mathbf{m}} - \overline{\mathbf{S}}(\overline{\mathbf{x}})\right]^T \cdot \overline{\overline{\mathbf{\Lambda}}}_e^{-1} \cdot \left[\overline{\mathbf{m}} - \overline{\mathbf{S}}(\overline{\mathbf{x}})\right]\right\}. \quad (100)$$

In Bayesian inference approaches, this probability density function is also referred to as the *Likelihood function* and is denoted by $L(\overline{\mathbf{x}}|\overline{\mathbf{m}})$. Maximizing the probability distribution function of Eq. (100) is equivalent to minimizing the negative of its logarithm which is also equivalent to the weighted least squares minimization of the first term of the cost function of Eq. (3) with:

$$\overline{\overline{\mathbf{\Lambda}}}_e^{-1} = \overline{\overline{\mathbf{W}}}_d^T \cdot \overline{\overline{\mathbf{W}}}_d$$

This form of parameter estimation is called the *maximum likelihood estimation*:

$$\begin{aligned} \overline{\mathbf{x}}_{ML} &= \arg\max_{\overline{\mathbf{x}}}\{L(\overline{\mathbf{x}}|\overline{\mathbf{m}})\} = \arg\max_{\overline{\mathbf{x}}}\{P(\overline{\mathbf{m}}|\overline{\mathbf{x}})\} \\ &= \arg\min_{\overline{\mathbf{x}}}\left\{\frac{1}{2}\left[\overline{\mathbf{S}}(\overline{\mathbf{x}}) - \overline{\mathbf{m}}\right]^T \cdot \overline{\overline{\mathbf{\Lambda}}}_e^{-1} \cdot \left[\overline{\mathbf{S}}(\overline{\mathbf{x}}) - \overline{\mathbf{m}}\right]\right\}. \end{aligned} \quad (101)$$

On the other hand, the joint probability density function $P(\overline{\mathbf{x}}, \overline{\mathbf{m}})$, can be represented by Bayes' rule as follows:

$$P(\overline{\mathbf{x}}|\overline{\mathbf{m}}) = P(\overline{\mathbf{m}})P(\overline{\mathbf{x}}|\overline{\mathbf{m}}) = P(\overline{\mathbf{x}})P(\overline{\mathbf{m}}|\overline{\mathbf{x}}) = P(\overline{\mathbf{x}})L(\overline{\mathbf{x}}|\overline{\mathbf{m}}), \quad (102)$$

from which we obtain the following expression for the conditional probability density of the model $\overline{\mathbf{x}}$ given the data $\overline{\mathbf{m}}$:

$$P(\overline{\mathbf{x}}|\overline{\mathbf{m}}) = \frac{P(\overline{\mathbf{x}})L(\overline{\mathbf{x}}|\overline{\mathbf{m}})}{P(\overline{\mathbf{m}})}, \quad (103)$$

where $P(\overline{\mathbf{x}})$ and $P(\overline{\mathbf{m}})$ are the prior (or marginal) probability density functions of the model $\overline{\mathbf{x}}$ and the data $\overline{\mathbf{m}}$, respectively. In the terminology of the Bayesian approach, $P(\overline{\mathbf{x}}|\overline{\mathbf{m}})$ is referred to as the *posterior probability density function*, $P(\overline{\mathbf{x}})$ as the *prior probability density function* and $P(\overline{\mathbf{m}})$ as the *evidence*. The prior probability density function $P(\overline{\mathbf{m}})$ can be regarded as a normalization factor which makes the integral of the posterior probability density function $P(\overline{\mathbf{x}}|\overline{\mathbf{m}})$ with respect to the model vector $\overline{\mathbf{x}}$ equal to unity, hence:

$$P(\overline{\mathbf{m}}) = \int d\,\overline{\mathbf{x}}\, P(\overline{\mathbf{x}}) L(\overline{\mathbf{x}}|\overline{\mathbf{m}}). \tag{104}$$

The estimation problem of maximizing the posterior probability density function, $P(\overline{\mathbf{x}}|\overline{\mathbf{m}})$, is often called the *maximum a posteriori probability estimation*:

$$\overline{\mathbf{x}}_{MAP} = \arg \max_{\overline{\mathbf{x}}} \{P(\overline{\mathbf{m}}|\overline{\mathbf{x}})\}, \tag{105}$$

which is equivalent to minimizing the negative of the logarithm of $P(\overline{\mathbf{x}}|\overline{\mathbf{m}})$, i.e.,

$$\overline{\mathbf{x}}_{MAP} = \arg \min_{\overline{\mathbf{x}}} \left\{ \frac{1}{2} \left[\overline{\mathbf{S}}(\overline{\mathbf{x}}) - \overline{\mathbf{m}} \right]^T \cdot \overline{\overline{\mathbf{\Lambda}}}_e^{-1} \cdot \left[\overline{\mathbf{S}}(\overline{\mathbf{x}}) - \overline{\mathbf{m}} \right] - \ln \left[\frac{P_e P(\overline{\mathbf{x}})}{P(\overline{\mathbf{m}})} \right] \right\}, \tag{106}$$

which is also equivalent to the weighted least squares minimization problem defined by the cost function of Eq. (94) with the regularization term given by:

$$R(\overline{\mathbf{x}}, \overline{\mathbf{x}}_p) = -2 \ln \left[\frac{P_e P(\overline{\mathbf{x}})}{P(\overline{\mathbf{m}})} \right]. \tag{107}$$

From Eq. (101) and Eq. (106), $\overline{\mathbf{x}}_{MAP}$ will reduce to $\overline{\mathbf{x}}_{ML}$ in the absence of any *a priori* information (i.e., with a uniform prior probability density function $P(\overline{\mathbf{x}})$).

Introducing the additional assumption that the model $\overline{\mathbf{x}}$ is a random variable represented by a normal distribution with mean $\overline{\mathbf{x}}_p$, the prior probability density function is thus given by:

$$P(\overline{\mathbf{x}}) = P_x \exp \left\{ -\frac{1}{2} [\overline{\mathbf{x}} - \overline{\mathbf{x}}_p]^T \cdot \overline{\overline{\mathbf{\Lambda}}}_x^{-1} \cdot [\overline{\mathbf{x}} - \overline{\mathbf{x}}_p] \right\}, \tag{108}$$

where P_x is a normalization constant and $\overline{\overline{\mathbf{\Lambda}}}_x$ is the *model covariance matrix*. In this case:

$$\begin{aligned} \overline{\mathbf{x}}_{MAP} \;=\; \arg \min_{\overline{\mathbf{x}}} \frac{1}{2} \Bigg\{ & \left[\overline{\mathbf{S}}(\overline{\mathbf{x}}) - \overline{\mathbf{m}} \right]^T \cdot \overline{\overline{\mathbf{\Lambda}}}_e^{-1} \cdot \left[\overline{\mathbf{S}}(\overline{\mathbf{x}}) - \overline{\mathbf{m}} \right] \\ & + [\overline{\mathbf{x}} - \overline{\mathbf{x}}_p]^T \cdot \overline{\overline{\mathbf{\Lambda}}}_x^{-1} \cdot [\overline{\mathbf{x}} - \overline{\mathbf{x}}_p] \Bigg\}, \end{aligned} \tag{109}$$

which is equivalent to the weighted least squares minimization problem defined by the cost function of Eq. (3) with $\mu = 1$ and with:

$$\overline{\overline{\mathbf{\Lambda}}}_e^{-1} = \overline{\overline{\mathbf{W}}}_d^T \cdot \overline{\overline{\mathbf{W}}}_d, \qquad \overline{\overline{\mathbf{\Lambda}}}_x^{-1} = \overline{\overline{\mathbf{W}}}_x^T \cdot \overline{\overline{\mathbf{W}}}_x. \tag{110}$$

Before closing this section, we note that the above stochostic framework does not readily provide a means to accommodate a Lagrange multiplier μ as in the least squares approach defined by the cost function of Eq. (3). As we discussed before, the role of the Lagrange multiplier μ is important in eliminating degenerate inversions which are likely to achieve unrealistically small values of the residual errors to levels below the noise plateau χ (see Section 10). By judicially adjusting the Lagrange multiplier μ, the residual errors will be brought into their expected statistical range. In the rest of this analysis we will (unjustifiably) introduce the Lagrange multplier, μ, in the exponent of the likelihood function, to obtain:

$$P(\overline{\mathbf{x}}|\overline{\mathbf{m}}) = \frac{P_e P_x}{P(\overline{\mathbf{m}})} \exp\{-C(\overline{\mathbf{x}}, \overline{\mathbf{m}})\}, \tag{111}$$

where $C(\overline{\mathbf{x}}, \overline{\mathbf{m}})$ is given by:

$$\begin{aligned} C(\overline{\mathbf{x}}, \overline{\mathbf{m}}) \;=\; & \frac{1}{2}\left\{\mu\left[\overline{\mathbf{S}}(\overline{\mathbf{x}}) - \overline{\mathbf{m}}\right]^T \cdot \overline{\overline{\mathbf{\Lambda}}}_e^{-1} \cdot \left[\overline{\mathbf{S}}(\overline{\mathbf{x}}) - \overline{\mathbf{m}}\right]\right. \\ & \left. + \left[\overline{\mathbf{x}} - \overline{\mathbf{x}}_p\right]^T \cdot \overline{\overline{\mathbf{\Lambda}}}_x^{-1} \cdot \left[\overline{\mathbf{x}} - \overline{\mathbf{x}}_p\right]\right\}. \end{aligned} \tag{112}$$

Finally, it should be noted that the choice of which probability density, $P(\overline{\mathbf{e}})$, may truly represent the experimental uncertainties for the data is not a straightforward matter. In principle, a careful examination of the experimental conditions under which the data were gathered can suggest an appropriate choice of the probability density for representing the data uncertainties, however, such a task may not always be easy or even possible. The difficulty that arises in attempting to describe the statistics of the data, is that some of the uncertainties affecting the data are not statistical in nature. Because of these difficulties, the tendency some times is to assume a probability density which may result in biasing the estimation process leading to erroneous results.

14.2. The Fisher Information Matrix

We start with the definition of the *score function* defined as the gradient with respect to the model parameters of the logarithm of the joint

probability distribution function [27]:

$$\overline{\mathbf{s}}(\overline{\mathbf{x}}, \overline{\mathbf{m}}) = \nabla_{\overline{\mathbf{x}}} \ln[P(\overline{\mathbf{x}}, \overline{\mathbf{m}})]. \tag{113}$$

The score function "scores" values of $\overline{\mathbf{x}}$ as the random vector $\overline{\mathbf{m}}$ assumes its various values. Score values which are near to zero are "good" scores and scores which are different from zero are "bad" scores. The score function has mean zero:

$$E\{\overline{\mathbf{s}}(\overline{\mathbf{x}}, \overline{\mathbf{m}})\} = \int d\,\overline{\mathbf{m}} P(\overline{\mathbf{x}}, \overline{\mathbf{m}}) \nabla_{\overline{\mathbf{x}}} \ln\{P(\overline{\mathbf{x}}, \overline{\mathbf{m}})\} = \nabla_{\overline{\mathbf{x}}} \int d\,\overline{\mathbf{m}} P(\overline{\mathbf{x}}, \overline{\mathbf{m}}) = 0, \tag{114}$$

where E denotes the expected value. The Fisher Information Matrix is the covariance matrix of the score function and is denoted by $\overline{\overline{\mathbf{\Gamma}}}(\overline{\mathbf{x}})$:

$$\begin{aligned} \overline{\overline{\mathbf{\Gamma}}}(\overline{\mathbf{x}}) &= E\left\{\overline{\mathbf{s}}(\overline{\mathbf{x}}, \overline{\mathbf{m}}) \overline{\mathbf{s}}^T(\overline{\mathbf{x}}, \overline{\mathbf{m}})\right\} \\ &= E\left\{\left[\nabla_{\overline{\mathbf{x}}} \ln\{P(\overline{\mathbf{x}}, \overline{\mathbf{m}})\}\right]\left[\nabla_{\overline{\mathbf{x}}} \ln\{P(\overline{\mathbf{x}}, \overline{\mathbf{m}})\}\right]^T\right\}. \end{aligned} \tag{115}$$

From the identity:

$$\begin{aligned} \nabla_{\overline{\mathbf{x}}} \nabla_{\overline{\mathbf{x}}} \ln\{P(\overline{\mathbf{x}}, \overline{\mathbf{m}})\} &= \frac{1}{P(\overline{\mathbf{x}}, \overline{\mathbf{m}})} \nabla_{\overline{\mathbf{x}}} \nabla_{\overline{\mathbf{x}}} P(\overline{\mathbf{x}}, \overline{\mathbf{m}}) \\ &\quad - \left[\nabla_{\overline{\mathbf{x}}} \ln\{P(\overline{\mathbf{x}}, \overline{\mathbf{m}})\}\right]\left[\nabla_{\overline{\mathbf{x}}} \ln\{P(\overline{\mathbf{x}}, \overline{\mathbf{m}})\}\right]^T \end{aligned} \tag{116}$$

and from the equality:

$$E\left\{\frac{1}{P(\overline{\mathbf{x}}, \overline{\mathbf{m}})} \nabla_{\overline{\mathbf{x}}} \nabla_{\overline{\mathbf{x}}} P(\overline{\mathbf{x}}, \overline{\mathbf{m}})\right\} = 0, \tag{117}$$

it follows that $\overline{\overline{\mathbf{\Gamma}}}(\overline{\mathbf{x}})$ is also given by:

$$\overline{\overline{\mathbf{\Gamma}}}(\overline{\mathbf{x}}) = -E\left\{\nabla_{\overline{\mathbf{x}}} \nabla_{\overline{\mathbf{x}}} \ln[P(\overline{\mathbf{x}}, \overline{\mathbf{m}})]\right\}. \tag{118}$$

Using expression (118) and for the posterior probability density function of Eq. (111), the Fisher information matrix is given by the Hessian matrix of Eq. (12):

$$\overline{\overline{\mathbf{\Gamma}}}(\overline{\mathbf{x}}) = \overline{\overline{\mathbf{G}}}(\overline{\mathbf{x}}) = \overline{\overline{\mathbf{\Lambda}}}_x^{-1} + \mu\, \overline{\overline{\mathbf{J}}}^T(\overline{\mathbf{x}}) \cdot \overline{\overline{\mathbf{\Lambda}}}_e^{-1} \cdot \overline{\overline{\mathbf{J}}}(\overline{\mathbf{x}}), \tag{119}$$

where we have discarded second-order derivatives (consistent with the Gauss-Newton optimization method). This assumption is justified since the residual error at the minimum of the cost function is small enough such that the first-order term $\overline{\overline{\mathbf{J}}}^T(\overline{\mathbf{x}}) \cdot \overline{\overline{\mathbf{W}}}_d^T \overline{\overline{\mathbf{W}}}_d \cdot \overline{\overline{\mathbf{J}}}(\overline{\mathbf{x}})$ of Eq. (12)

will dominate the second-order term $\overline{\overline{\mathbf{Q}}}(\overline{\mathbf{x}})$ which is weighted by the residual errors.

The Fisher information matrix is a measure of the information content in the data. It provides a sensitivity map of the data with respect to the model parameters.

14.3. The Estimator's Covariance Matrix and the Cramer-Rao Lower Bound

After estimating the vector of the model parameters $\overline{\mathbf{x}}$, one is interested in computing the estimator's covariance matrix denoted by $\overline{\overline{\mathbf{\Sigma}}}$ and defined by:

$$\overline{\overline{\mathbf{\Sigma}}} = E\left\{[\overline{\mathbf{x}}^* - E(\overline{\mathbf{x}}^*)]\,[\overline{\mathbf{x}}^* - E(\overline{\mathbf{x}}^*)]^T\right\}, \tag{120}$$

where $\overline{\mathbf{x}}^*$ is the estimator of the model parameter vector $\overline{\mathbf{x}}$. A related parameter that attempts to quantifies errors is the *error covariance matrix* denoted by $\overline{\overline{\mathbf{\Omega}}}$ and defined by:

$$\overline{\overline{\mathbf{\Omega}}} = E\left\{[\overline{\mathbf{x}}^* - \overline{\mathbf{x}}][\overline{\mathbf{x}}^* - \overline{\mathbf{x}}]^T\right\}. \tag{121}$$

The diagonal element, $E\{(x_n^* - x_n)^2\}$, is the mean-squared error between the estimator x_n^* and the true model parameter x_n. The off-diagonal element is the cross-covariance between the errors of two different parameters of the model. The error covariance matrix is related to the estimator's covariance matrix by:

$$\overline{\overline{\mathbf{\Omega}}} = \overline{\overline{\mathbf{\Sigma}}} + \overline{\overline{\mathbf{B}}}, \tag{122}$$

where $\overline{\overline{\mathbf{B}}}$ is the *bias-squared matrix* for the estimator, given by:

$$\overline{\overline{\mathbf{B}}} = [E\{\overline{\mathbf{x}}^*\} - \overline{\mathbf{x}}][E\{\overline{\mathbf{x}}^*\} - \overline{\mathbf{x}}]^T. \tag{123}$$

The diagonal term of $\overline{\overline{\mathbf{\Omega}}}$ (the mean-squared error of the estimator) is therefore given by:

$$\begin{aligned} E\left\{(x_n^* - x_n)^2\right\} &= E\left\{[x_n^* - E\{x_n\}]^2\right\} + (E\{x_n^*\} - x_n)^2 \\ &= \mathrm{var}(x_n^*) + (E\{x_n^*\} - x_n)^2, \end{aligned} \tag{124}$$

which is the sum of the variance of the estimator plus its bias-squared. The estimator $\overline{\mathbf{x}}^*$ is said to be unbiased if:

$$E\{\overline{\mathbf{x}}^*\} = \overline{\mathbf{x}}, \tag{125}$$

in this case $\overline{\overline{\mathbf{B}}} = 0$ and the error covariance matrix, $\overline{\overline{\mathbf{\Omega}}}$, is equal to the estimator covariance matrix, $\overline{\overline{\mathbf{\Sigma}}}$, while the mean-squared error becomes the variance of the estimator. In this case (for an unbiased estimator), it can be shown that [27]:

$$\overline{\overline{\mathbf{\Sigma}}} = \overline{\overline{\mathbf{\Omega}}} \geq \overline{\overline{\mathbf{\Gamma}}}^{-1}, \tag{126}$$

which is the *Cramer-Rao* inequality which bounds the covariance matrix of an unbiased estimator from below by the inverse of the Fisher information matrix.

A similar and an approximate result can be obtained by expanding the exponent, $C(\overline{\mathbf{x}}, \overline{\mathbf{m}})$, of the posterior probability density function, $P(\overline{\mathbf{x}}|\overline{\mathbf{m}})$, of Eqs. (111)–(112) around the estimator $\overline{\mathbf{x}}^*$, to obtain (see Eq. (9)):

$$P(\overline{\mathbf{x}}|\overline{\mathbf{m}}) \approx \frac{P_e P_x}{P(\overline{\mathbf{m}})} \exp[-C(\overline{\mathbf{x}}^*, \overline{\mathbf{m}})] \exp\left[\frac{1}{2}(\overline{\mathbf{x}} - \overline{\mathbf{x}}^*)^T \cdot \overline{\overline{\mathbf{G}}}(\overline{\mathbf{x}}^*) \cdot (\overline{\mathbf{x}} - \overline{\mathbf{x}}^*)\right], \tag{127}$$

which shows that when the cost function is approximated by a local quadratic form, the posterior probability density function is Gaussian in the model space. Furthermore, from Eq. (127) one can trivially deduce that the estimator covariance matrix, $\overline{\overline{\mathbf{\Sigma}}}$, is approximated by:

$$\begin{aligned}\overline{\overline{\mathbf{\Sigma}}} &\approx \overline{\overline{\mathbf{G}}}^{-1}(\overline{\mathbf{x}}^*) = \overline{\overline{\mathbf{\Gamma}}}^{-1} = \left[\overline{\overline{\mathbf{\Lambda}}}_x^{-1} + \mu\, \overline{\overline{\mathbf{J}}}^T(\overline{\mathbf{x}}^*) \cdot \overline{\overline{\mathbf{\Lambda}}}_e^{-1} \cdot \overline{\overline{\mathbf{J}}}(\overline{\mathbf{x}}^*)\right]^{-1} \\ &= \overline{\overline{\mathbf{\Lambda}}}_x - \mu\, \overline{\overline{\mathbf{\Lambda}}}_x \cdot \overline{\overline{\mathbf{\Lambda}}}^T(\overline{\mathbf{x}}^*) \cdot \left[\overline{\overline{\mathbf{\Lambda}}}_e + \mu\, \overline{\overline{\mathbf{J}}}(\overline{\mathbf{x}}^*) \cdot \overline{\overline{\mathbf{\Lambda}}}_x \cdot \overline{\overline{\mathbf{J}}}^T(\overline{\mathbf{x}}^*)\right]^{-1} \cdot \overline{\overline{\mathbf{J}}}(\overline{\mathbf{x}}^*) \cdot \overline{\overline{\mathbf{\Lambda}}}_x.\end{aligned} \tag{128}$$

In deriving Eq. (127), the gradient $\overline{\mathbf{g}}(\overline{\mathbf{x}}^*)$ is set to zero, since it vanishes at the minimum of the cost function $C(\overline{\mathbf{x}}, \overline{\mathbf{m}})$.

In the case when the measurement noise is uncorrelated and with a uniform standard deviation, σ, then:

$$\begin{aligned}\overline{\overline{\mathbf{\Sigma}}} &\approx \sigma^2 \left[\sigma^2 \overline{\overline{\mathbf{\Lambda}}}_x^{-1} + \mu\, \overline{\overline{\mathbf{J}}}^T(\overline{\mathbf{x}}^*) \cdot \overline{\overline{\mathbf{J}}}(\overline{\mathbf{x}}^*)\right]^{-1} \\ &= \overline{\overline{\mathbf{\Lambda}}}_x - \mu\, \overline{\overline{\mathbf{\Lambda}}}_x \cdot \overline{\overline{\mathbf{J}}}^T(\overline{\mathbf{x}}^*) \cdot \left[\sigma^2 \overline{\overline{\mathbf{I}}} + \mu\, \overline{\overline{\mathbf{J}}}(\overline{\mathbf{x}}^*) \cdot \overline{\overline{\mathbf{\Lambda}}}_x \cdot \overline{\overline{\mathbf{J}}}^T(\overline{\mathbf{x}}^*)\right]^{-1} \cdot \overline{\overline{\mathbf{J}}}(\overline{\mathbf{x}}^*) \cdot \overline{\overline{\mathbf{\Lambda}}}_x.\end{aligned} \tag{129}$$

The square root of the diagonal terms (variances) of the covariance matrix provide "error bars" describing the uncertainties in the

estimated values of the model parameters. On the average, the true value of the i-th model parameter x_i will fall, 68% of the time, within $\pm\sqrt{\Sigma_{ii}}$ of the estimated value x_i^*, within $\pm 2\sqrt{\Sigma_{ii}}$, 95% of the time and within $\pm 3\sqrt{\Sigma_{ii}}$, 99.7% of the time.

The interpretation of the off-diagonal elements (covariances) of the covariance matrix is not straightforward. A more meaningful indicator of error correlations is the *correlation coefficient matrix*, $\overline{\overline{\rho}}$, defined by:

$$\rho_{ij} = \frac{\Sigma_{ij}}{\sqrt{\Sigma_{ii}\Sigma_{jj}}}, \tag{130}$$

which is bounded by the inequality $-1 \leq \rho_{ij} \leq +1$. If the correlation coefficient, ρ_{ij}, between the estimated parameters x_i^* and x_j^* is close to zero, then the uncertainties in estimating these two model parameters are uncorrelated. On the other hand, the uncertainties will be highly correlated (or anti-correlated), if the correlation coefficient is close to $+1$ (or -1). A strong correlation on uncertainties, means that the two parameters have not been independently resolved by the data set, but that only some linear combination of the two parameters has been resolved.

15. NUMERICAL EXAMPLES

In order to illustrate the strength and weakness of the presented parametric inversion algorithm, we show some representative inversion results of electromagnetic measurements at induction frequency (diffusive regime) in the geophysical borehole applications. Since from the point of view of CPU time, the numerical computation of the second derivative in the Jacobian matrix is not feasible, in the inversion we use Gauss-Newton approach using constrained minimization and line search. Due to lack of *a priori* information the weighting matrixes given in Eq. (3), $\overline{\overline{\mathbf{W}}}_d$ and $\overline{\overline{\mathbf{W}}}_x$, are chosen to be identity matrixes. The parameter $\overline{\mathbf{x}}_p$ in the cost function of Eq. (3) is chosen to be equal to the inverted parameter in the previous iterative step $\overline{\mathbf{x}}_k$. In this way we reduce the possibility of the optimization process to make a huge jump around within two successive iteration. The regularization parameter $1/\mu$ is determined using the results of multiplicative regularization analysis given in Eq. (74). Note that the parameter in Eq. (74) is needed to be determined one time, and then this value is used for all the examples.

15.1. Example 1

In this first example, the normalization given in Eq. (6) is used. The motivation behind this normalization choice is that the sensitivity of the apparent resistivity measurement (the measured data) is boosted at point close to the bed boundaries and away from the center bed. This boosting is a result of the fact that the apparent resistivity readings at point close to the bed boundaries are, in general smaller in magnitude than those closer to the center of the bed. Data points that are close to the bed interfaces are important for radial resolution in each bed. The measured data are assumed to be collected using AIT (Array Induction Tool), a Schlumbergers tool, which has three operating frequency (26, 52 and 105 kHz). As the forward code we use a three-dimensional finite difference code the so-called SLDMINV developed by [28]. Although the computation of one logging point using this finite-difference scheme can be obtained within a few seconds, in our inversion scheme we need to run it for a great number of time (proportional to number of unknown and logging point) in order to generate the Jacobian matrix. Hence, in order to make inversion results ready within acceptable amount of time, we use the BFGS update given in Eq. (90) to construct the Hessian matrix.

As a test case we consider geometry consisting four beds (three with invasion) as shown in Figure 1. This geometry was first proposed

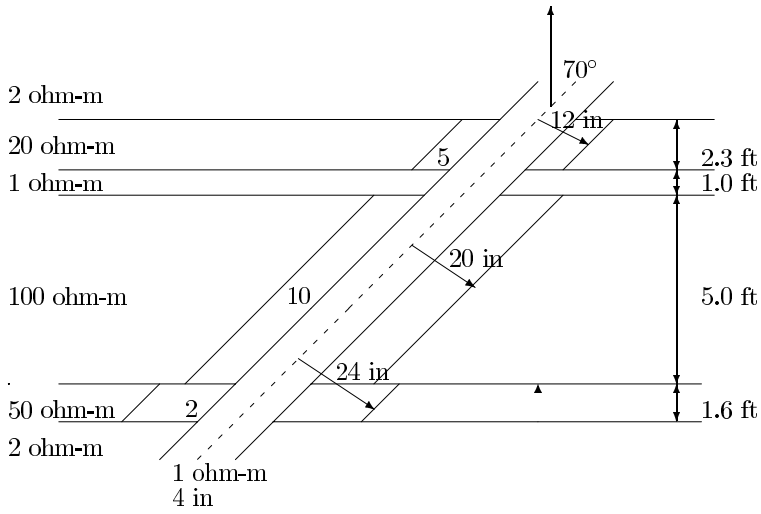

Figure 1. Four-bed invaded formation. The dip angle is 70°. Invasion radii are perpendicular to the borehole axis.

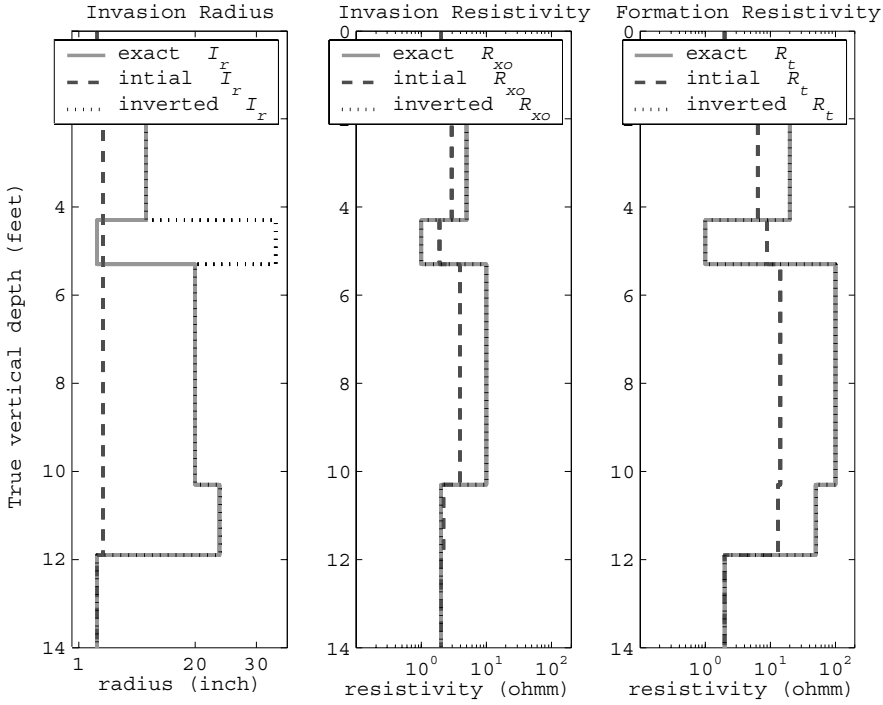

Figure 2. Inversion results of the four layers example.

by [29]. The well is highly deviated; the deviation angle is 70°. The borehole radius is 4 inch and the resistivity of the mudcake is 1 ohm-m.

In the inversion we aim to invert the formation resistivity (R_t), invasion resistivity (R_{xo}) and invasion radius (I_r). Since we do not have enough sensitivity to invert for the bed boundaries and the borehole, they are assumed to be known from other independent measurements. The measurement data, which are the apparent resistivities, are collected within 10 logging points distributed uniformly from $z = 2$ ft until $z = 12$ ft. In each logging point the AIT tool collected 14 data points. The plot of the inversion results after 25 iterations are given in Figure 2. In this figure we show the exact model parameters (solid lines), initial model parameters (dashed lines) and the inverted model parameters (dotted lines). The initial model parameters are the ones obtained from apparent resistivity reading of the tool. Note that after 25 iterations the value of the square root of the cost function reduced to 0.055%. Further, one should note that in Figure 2, since $R_t \approx R_{x0}$ of the 2nd layer from the top, the value of the inverted invasion radius

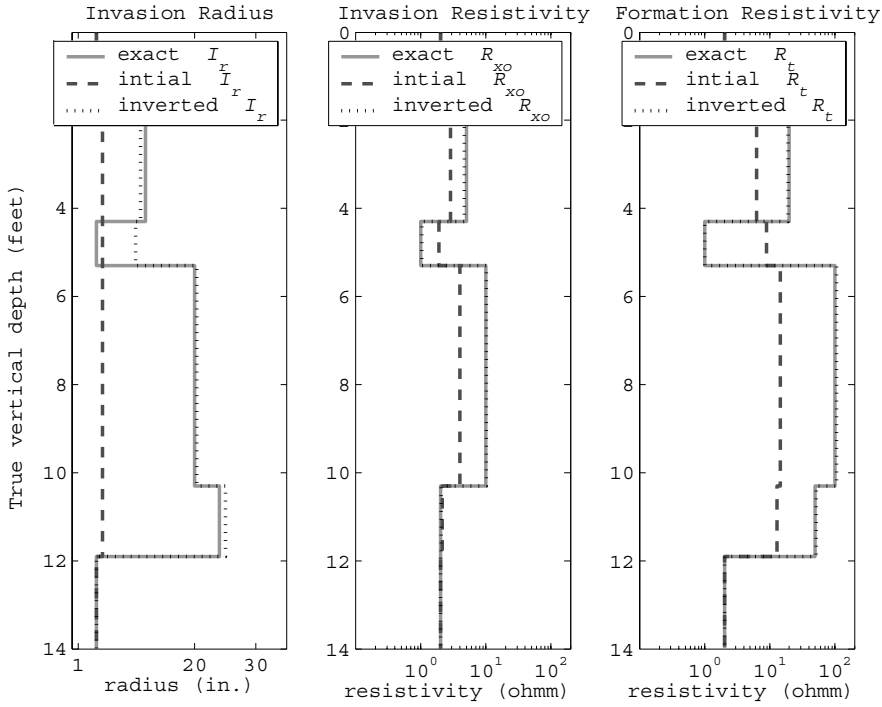

Figure 3. Inversion results of the four layers example, but now the data are corrupted with 3% random white noise.

I_r is not important.

Next, we add 3% pseudo random white noise to the simulated data by using the following procedure:

$$\overline{\mathbf{m}}_{\text{noise}} = (1 + \beta f_{\text{noise}})\overline{\mathbf{m}}, \tag{131}$$

where f_{noise} is a random number between -1 and 1, and $\beta = 0.03$ is the amount of noise. The inversion results after 42 iterations are given in Figure 3. After 42 iterations the square root of the cost functional reduced approximately to the noise level 1.6%. We observe that the all the inverted resistivity parameters are excellent. The effect of noise just appeared on the reconstructed invasion radius parameter of the fourth bed from the top. The invasion radius of this fourth bed is obviously overestimated.

15.2. Example 2

As the next example we consider inversion of non-invaded TI anisotropy layering formation in deviated borehole configuration. In this example, we aim to reconstruct the horizontal formation resistivity R_h, vertical formation resistivity R_v and the bed boundary of every layer. Furthermore the dip angle θ will also be reconstructed. Hence the number of unknown model parameters is $N = 3L$ where L is the number of layers.

In order to obtain enough sensitivity to carry out this inversion, the measured data are assumed to be obtained from triaxial induction measurement, see [30]. This type of measurement incorporates three mutually orthogonal transmitter and receiver coils. Then, for each receiver position we measure nine orthogonal magnetic field components (i.e., an x-directed transmitter with x, y and z-directed receivers, a y-directed transmitter with x, y and z-directed receivers, and a z-directed transmitter with x, y and z-directed receivers). In the inversion we assumed that the azimuth angle is solved first by rotating the data matrix so that the cross coupling xy (x transmitter and y receiver), yx, yz and zx are zero. This puts the coordinate y axis along the relative strike of the formation. Hence, we have only xx, yy, zz, xz and zx date left to carry out inversion of the remaining parameters. The ability to simply rotate the data is one of the significance advantages of a fully triaxial tool with all sensors located at a common point. The transmitter and receiver coils are modelled as points magnetic dipoles because it has been demonstrated both theoretically [31] and [32] that a point magnetic dipole is accurate at observation distances greater than several coil radii. Since in this case we have different type of data the normalization given in Eq. (7) is used. The data are assumed to be collected using Schlumberger AIT-H tool configuration. With this tool, for each logging point, we recorded the triaxial vector magnetic fields in six different positions with respect to the transmitter. The transmitter and receiver separations are varied from 15 inch up to 72 inch. For this simulation a fast semi analytical forward model is used [33]. Hence, the Jacobian are generated with the aid of finite difference calculation. The data are collected at 178 logging points, distributed uniformly from $z = -10\,\text{ft}$ up to $z = 167\,\text{ft}$.

As a test case we consider a twenty eight layer model where the well is deviated $\theta = 30^\circ$. The model has been adapted from the standard Oklahoma formation to replicate TI anisotropy model. The details description of the model is given in Table 1.

The true, initial and inverted model after 15 iterations are given in Figure 4. In this figure the exact model is given by the solid lines while the initial model and the inverted model are given by the dashed

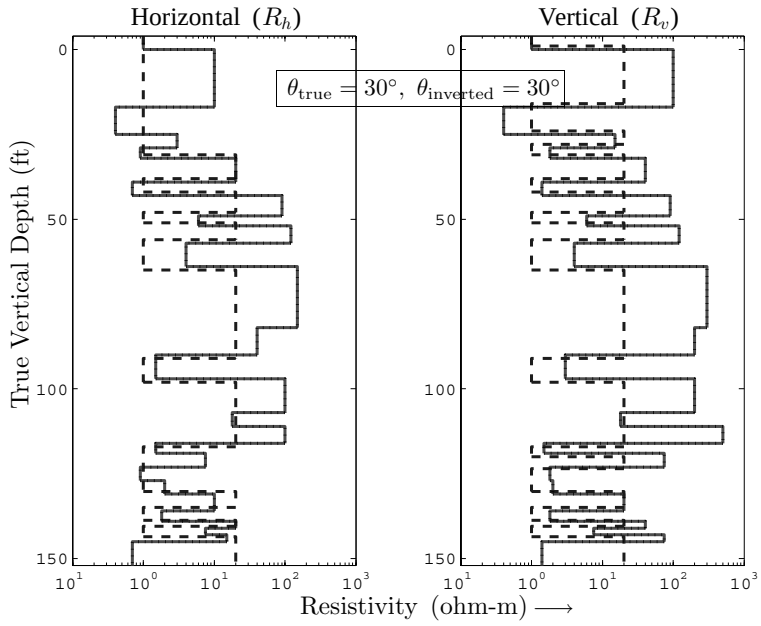

Figure 4. Inversion results of the twenty-eight non-invaded TI anisotropy layers example. The true model, the initial estimate and the inverted model are given by the solid, dashed and dotted lines.

and dotted lines. We observe that all the unknown parameters are reconstructed very well. Note that after 15 iterations the square root of the cost function is reduced to 0.0018%. In Figure 5, we plot the square root of the cost functional $C(\overline{\mathbf{x}})$ as function of iteration for the case without noise (solid line) and with 3% pseudo random white noise (dashed line). The inversion results from data corrupted with 3% pseudo random white noise is given in Figure 6.

In Figure 6 we observe only small effect of noise in the reconstructions of the vertical resistivities. One should also note that the value of the cost functional in the inversion of noisy data converge to a certain value which corresponds to the noise level, see Figure 5.

Finally we note that the approximate time needed for this parametric inversion procedure is given by

$$T = (N + 2)KMt, \tag{132}$$

where N is the number of unknowns, K is the total number of iterations, M is the number of sources (logging points) and t is the computation time of the forward solver for each source. Note that

Table 1. Description of the non-invaded TI anisotropy resistivity model to perform numerical simulations shown in next figures. The well is deviated $\theta = 30°$.

Nr.	Layer to (ft)	R_h (ohm-m)	R_v (ohm-m)	Thickness (ft)
1	0	1	1	∞
2	17	10	100	17
3	25	0.4	0.4	8
4	29	3	15	4
5	32	0.9	1.8	3
6	39	20	40	7
7	43	0.7	1.4	4
8	49	90	90	6
9	52	6	6	3
10	57	120	120	5
11	64	4	4	7
12	64	150	300	18
13	90	40	200	8
14	97	1.5	3	7
15	107	100	200	10
16	111	18	18	4
17	116	100	500	5
18	119	1.5	1.5	3
19	123	7.5	75	4
20	127	0.9	1.8	4
21	131	2	2	4
22	136	10	20	5
23	139	1.8	1.8	3
24	141	20	40	2
25	143	7.5	7.5	2
26	145	15	75	2
27	157	0.7	1.4	12
28	$+\infty$	1.1	1.1	∞

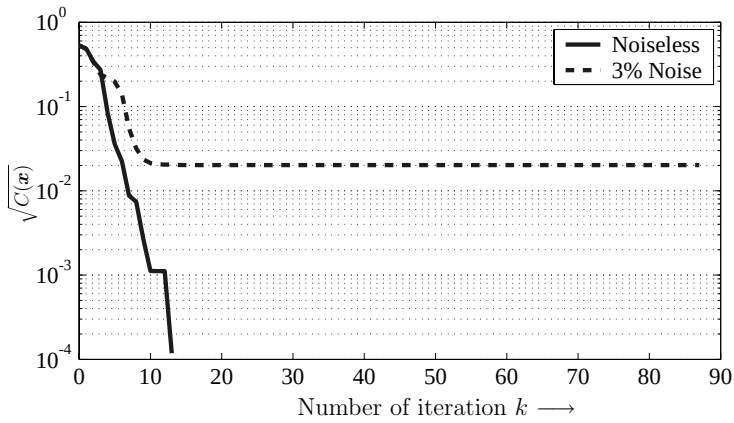

Figure 5. The square root of the cost functional as function of iteration of the twenty eight layers example.

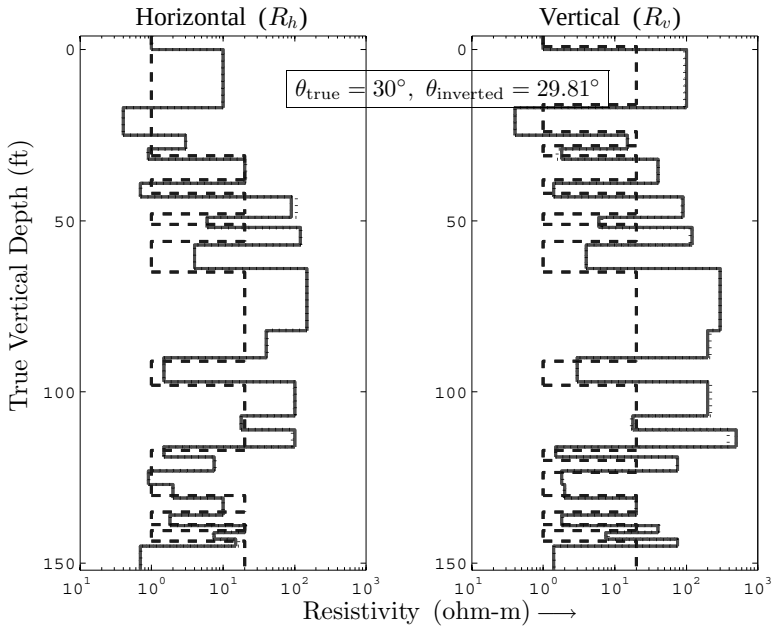

Figure 6. Inversion results of the twenty-eight non-invaded TI anisotropy layers example from data with 3% random white noise. The true model, the initial estimate and the inverted model are given by the solid, dashed and dotted lines.

in Eq. (132), the number of two comes out from the fact that in each iteration the scheme need at least one call to forward solver to calculate the cost functional and each time the line search is used, more calls are required.

16. CONCLUSIONS

In this paper we have reviewed a number variant of Newton type methods. This Newton type methods will be a good candidate when one aim on inversion where the number of inverted model parameters is limited. Further enhancements of this Newton type method are done by using nonlinear transformation and a line search procedure. In this way the inverted model parameters are forced to be lied within their physical bounds. Furthermore as shown in the numerical example by using the multiplicative regularization technique to determine the regularization parameter we are arrive at an efficient and robust parametric inversion algorithm.

ACKNOWLEDGMENT

The authors thank Vladimir Druskin and Barbara Anderson for many valuable discussions on forward and inverse modelling.

APPENDIX A. NONLINEAR TRANSFORMATIONS FOR CONSTRAINED MINIMIZATION

If x_{max} is an upper bound on the model parameter x and x_{min} is a lower bound, then in order to ensure that $x_{min} < x < x_{max}$ at all iterations, we introduce one of the following transformations.

A.1. First Transformation

$$x = f(c, x_{min}, x_{max}) = x_{min} + \frac{x_{max} - x_{min}}{c^2 + 1} c^2, \quad -\infty < c < +\infty \tag{A1}$$

It is clear that:

$$x \to x_{min}, \qquad \text{as } c \to 0, \tag{A2}$$

$$x \to x_{max}, \qquad \text{as } c \to \pm\infty \tag{A3}$$

It is straightforward to show that:

$$\frac{\partial S_j}{\partial c} = \frac{dx}{dc}\frac{\partial S_j}{\partial x} = 2\frac{x_{max} - x}{x_{max} - x_{min}}\sqrt{(x_{max} - x)(x - x_{min})}\ \frac{\partial S_j}{\partial x}, \tag{A4}$$

where S_j is the j-th measurement. The two successive iterates x_{k+1} and x_k of x are related by:

$$x_{k+1} = x_{min} + \frac{x_{max} - x_{min}}{c_{k+1}^2 + 1} c_{k+1}^2 = x_{min} + \frac{x_{max} - x_{min}}{(c_k + q_k)^2 + 1}(c_k + q_k)^2, \tag{A5}$$

where

$$c_k = \left(\frac{x_k - x_{min}}{x_{max} - x_k} \right)^{1/2}, \tag{A6}$$

and $q_k = c_{k+1} - c_k$ is the Gauss-Newton search step in c towards the minimum of the cost function. Defining:

$$p = 2\frac{x_{max} - x}{x_{max} - x_{min}} \sqrt{(x_{max} - x)(x - x_{min})}\ q = \frac{dx}{dc} q, \tag{A7}$$

we obtain the following relationship between the two successive iterates x_{k+1} and x_k of x (assuming an adjustable step-length ν_k along the search direction):

$$x_{k+1} = x_{min} + \frac{x_{max} - x_{min}}{\alpha_k^2 + (x_k - x_{min})(x_{max} - x_k)^3} \alpha_k^2, \tag{A8}$$

where

$$\alpha_k = (x_k - x_{min})(x_{max} - x_k) + \frac{1}{2}(x_{max} - x_{min}) \nu_k p_k \tag{A9}$$

Note that:

$$x_{k+1} \to x_{max}, \qquad \text{if } x_k \to x_{max} \text{ or } x_{min} \tag{A10}$$

The variable p defined by Eq. (A7) is the solution of Eq. (15). Finally, one should note that this transformation of Eq. (A1) introduces false minima at $x = x_{max}$ and $x = x_{min}$ since $\partial S_j / \partial c$ vanishes at both $x = x_{max}$ and $x = x_{min}$. Notice that [from Eq. (A10)] this transformation skews the emphasis towards x_{max} rather than towards x_{min}.

A.2. Second Transformation

$$x = f(c, x_{min}, x_{max}) = \frac{x_{max} \exp(c) + x_{min} \exp(-c)}{\exp(c) + \exp(-c)}, \quad -\infty < c < +\infty \tag{A11}$$

It is clear that:

$$x \to x_{min}, \qquad \text{as } c \to -\infty, \tag{A12}$$

$$x \to x_{max}, \qquad \text{as } c \to +\infty \tag{A13}$$

It is straightforward to show that:

$$\frac{\partial S_j}{\partial c} = \frac{dx}{dc}\frac{\partial S_j}{\partial x} = 2\frac{(x_{max} - x)(x - x_{min})}{x_{max} - x_{min}}\frac{\partial S_j}{\partial x}, \tag{A14}$$

The two successive iterates x_{k+1} and x_k of x are related by:

$$\begin{aligned} x_{k+1} &= \frac{x_{max}\exp(c_{k+1}) + x_{min}\exp(-c_{k+1})}{\exp(c_{k+1}) + \exp(-c_{k+1})} \\ &= \frac{x_{max}\exp(c_k)\exp(q_k) + x_{min}\exp(-c_k)\exp(-q_k)}{\exp(c_k)\exp(q_k) + \exp(-c_k)\exp(-q_k)} \end{aligned} \tag{A15}$$

where

$$c_k = \frac{1}{2}\ln\left(\frac{x_k - x_{min}}{x_{max} - x_k}\right) \tag{A16}$$

Defining:

$$p = 2\frac{(x_{max} - x)(x - x_{min})}{x_{max} - x_{min}}\ q = \frac{dx}{dc}\ q, \tag{A17}$$

we obtain the following relationship between the two successive iterates x_{k+1} and x_k of x (assuming an adjustable step-length ν_k along the search direction):

$$x_{k+1} = \frac{x_{max}(x_k - x_{min})\exp(\alpha_k\nu_k p_k) + x_{min}(x_{max} - x_k)}{(x_k - x_{min})\exp(\alpha_k\nu_k p_k) + (x_{max} - x_k)}, \quad \text{for } p_k < 0, \tag{A18}$$

and

$$x_{k+1} = \frac{x_{max}(x_k - x_{min}) + x_{min}(x_{max} - x_k)\exp(-\alpha_k\nu_k p_k)}{(x_k - x_{min}) + (x_{max} - x_k)\exp(-\alpha_k\nu_k p_k)}, \quad \text{for } p_k > 0, \tag{A19}$$

where

$$\alpha_k = \frac{x_{max} - x_{min}}{(x_{max} - x_k)(x_k - x_{min})} \tag{A20}$$

Note that:

$$x_{k+1} \to x_{min}, \qquad \text{if } x_k \to x_{min}, \tag{A21}$$

$$x_{k+1} \to x_{max}, \qquad \text{if } x_k \to x_{max}, \tag{A22}$$

Hence, this transformation of Eq. (A11) introduces false minima at $x = x_{min}$ and $x = x_{max}$ since $\partial S_j/\partial c$ vanishes at both $x = x_{min}$ and $x = x_{max}$. Notice that this transformation symmetrizes the emphasis on both x_{min} and x_{max}.

A.3. Third Transformation

$$x = f(c, x_{min}, x_{max}) = \frac{x_{max} + x_{min}}{2} + \frac{x_{max} - x_{min}}{2} \sin c, \qquad -\infty < c < +\infty \quad \text{(A23)}$$

It is clear that:

$$x \to x_{min}, \qquad \text{as } \sin c \to -1, \quad \text{(A24)}$$

$$x \to x_{max}, \qquad \text{as } \sin c \to +1 \quad \text{(A25)}$$

It is straightforward to show that:

$$\frac{\partial S_j}{\partial c} = \frac{dx}{dc}\frac{\partial S_j}{\partial x} = \sqrt{(x_{max} - x)(x - x_{min})}\ \frac{\partial S_j}{\partial x}, \quad \text{(A26)}$$

The two successive iterates x_{k+1} and x_k of x are related by:

$$\begin{aligned} x_{k+1} &= \frac{x_{max} + x_{min}}{2} + \frac{x_{max} - x_{min}}{2} \sin c_{k+1} \\ &= \frac{x_{max} + x_{min}}{2} + \frac{x_{max} - x_{min}}{2} \sin\{c_k + q_k\}, \end{aligned} \quad \text{(A27)}$$

where

$$c_k = \arcsin\left\{\frac{2x_k - x_{max} - x_{min}}{x_{max} - x_{min}}\right\} \quad \text{(A28)}$$

Defining:

$$p = \sqrt{(x_{max} - x)(x - x_{min})}\ q = \frac{dx}{dc}\ q, \quad \text{(A29)}$$

we obtain the following relationship between the two successive iterates x_{k+1} and x_k of x (assuming an adjustable step-length ν_k along the search direction):

$$x_{k+1} = \frac{x_{max}+x_{min}}{2} + \left(x_k - \frac{x_{max}+x_{min}}{2}\right)\cos\left(\frac{\nu_k p_k}{\alpha_k}\right) + \alpha_k \sin\left(\frac{\nu_k p_k}{\alpha_k}\right), \quad \text{(A30)}$$

where

$$\alpha_k = \sqrt{(x_{max} - x_k)(x_k - x_{min})} \quad \text{(A31)}$$

REFERENCES

1. Tarantola, A., *Inverse Problem Theory*, Elsevier, Amsterdam, 1987.

2. Goldberg, D. E., *Genetic Algorithms in Search, Optimization and Machine Learning*, Addison-Wesley, Reading, MA, 1989.
3. Abubakar, A. and P. M. van den Berg, "Three dimensional nonlinear inversion in cross-well electrode logging," *Radio Science*, Vol. 33, No. 4, 989–1004, 1998.
4. Abubakar, A., P. M. van den Berg, and B. J. Kooij, "A conjugate gradient contrast source technique for 3D profile inversion," *IEICE Transactions on Electronics*, E83-C, 1864–1877, 2000.
5. Abubakar, A., P. M. van den Berg, and S. Y. Semenov, "Two- and three-dimensional algorithms for microwave imaging and inverse scattering," *Journal of Electromagnetic Waves and Applications*, Vol. 17, No. 2, 209–231, 2003.
6. Barkeshli, S. and R. G. Lautzenheiser, "An iterative method for inverse scattering problems based on an exact gradient search," *Radio Science*, Vol. 29, 1119–1130, 1994.
7. Chew, W. C. and Y. M. Wang, "Reconstruction of two-dimensional permittivity distribution using the distorted Born iterative method," *IEEE Transactions on Medical Imaging*, Vol. 9, 218–225, 1990.
8. Isernia, T., V. Pascazio, and R. Pierri, "On the local minima in a tomographic imaging technique," *IEEE Transactions on Geoscience and Remote Sensing*, Vol. 39, 1596–1607, 2001.
9. Kleinman, R. E. and P. M. van den Berg, "A modified gradient method for two-dimensional problems in tomography," *Journal of Computational and Applied Mathematics*, Vol. 42, 17–35, 1992.
10. Liu, Q. H. and W. C. Chew, "Applications of the CG-FFHT method with an improved FHT algorithm," *Radio Science*, Vol. 29, 1009–1022, 1994.
11. Zhdanov, M. S. and G. Hursan, "3D electromagnetic inversion based on quasi-analytical approximation," *Inverse Problems*, Vol. 16, 1297–1322, 2000.
12. Van den Berg, P. M. and A. Abubakar, "Contrast source inversion method: State of art," *Progress in Electromagnetics Research*, PIER 34, 189–218, 2001.
13. Van den Berg, P. M., A. Abubakar, and J. T. Fokkema, "Multiplicative regularization for contrast profile inversion," *Radio Science*, Vol. 38, No. 2, 23.1–23.10, 2003.
14. Franchois, A. and C. Pichot, "Microwave imaging — Complex permittivity reconstruction with Levenberg-Marquardt method," *IEEE Transactions on Antennas and Propagation*, Vol. 45, 203–215, 1997.

15. Habashy, T. M., M. L. Oristaglio, and A. T. de Hoop, "Simultaneous nonlinear reconstruction of two-dimensional permittivity and conductivity," *Radio Science*, Vol. 29, 1101–1118, 1994.
16. Dennis Jr., J. E. and R. B. Schnabel, *Numerical Methods for Unconstrained Optimization and Nonlinear Equations*, Prentice Hall, Inc., New Jersey, 1983.
17. Gill, P. E., W. Murray, and M. H. Wright, *Practical Optimization*, Academic Press, London, 1981.
18. Golub, G. H. and C. F. Van Loan, *Matrix Computations*, The Johns Hopkins University Press, London, 1993.
19. Habashy, T. M., W. C. Chew, and E. Y. Chow, "Simultaneous reconstruction of permittivity and conductivity profiles in a radially inhomogeneous slab," *Radio Science*, Vol. 21, 635–645, 1986.
20. Oldenburg, D. W., "Inversion of electromagnetic data: An overview of new techniques," *Surveys in Geophysics*, Vol. 11, 231–270, 1990.
21. Press, W. H., S. A. Teukol, W. T. Vetterling, and B. P. Flannery, *Numerical Recipes in FORTRAN*, 77, Cambridge University Press, England, 1992.
22. Fletcher, R., *Practical Methods of Optimization: Volume I, Unconstrained Optimization*, Wiley, 1980.
23. Luenberger, D. G., *Linear and Nonlinear Programming*, Addison-Wesley, Reading, MA, 1984.
24. Rudin, L., S. Osher, and C. Fatemi, "Nonlinear total variation based on noise removal algorithm," *Physica*, Vol. 30D, 259–268, 1992.
25. Chan, T. F. and C. K. Wong, "Total variation blind deconvolution," *IEEE Transactions on Image Processing*, Vol. 7, 370–375, 1998.
26. Broyden, C. G., *Mathematics of Computation*, Vol. 19, 577–593, 1965.
27. Scharf, L. L., *Statistical Signal Processing, Detection, Estimation, and Time Series Analysis*, Addison-Wesley Publishing Company, Massachusetts, 1991.
28. Druskin, V., L. A. Knizhnerman, and P. Lee, "New spectral Lanczos decomposition method for induction modeling in arbitrary 3-D geometry," *Geophysics*, Vol. 64, No. 3, 701–706, 1999.
29. Van der Horst, M., V. Druskin, and L. Knizhnerman, "Modeling induction logs in 3D geometres," *Three-Dimensional*

Electromagnetics, M. Oristaglio and B. Spies (eds.), 611–622, Society of Exploration Geophysicists, Tulsa, USA, 1999.

30. Anderson, B. I., T. D. Barber, and T. M. Habashy, "The interpretation and inversion of fully triaxial induction data; A sensitivity study," *SPWLA 43rd Annual Logging Symposium*, Oiso, Japan, June 2–5, 2003.
31. Spies, B. R. and T. M. Habashy, "Sensitivity analysis of crosswell electromagnetics," *Geophysics*, Vol. 60, No. 3, 834–845, 1995.
32. Anderson, B. I., S. Bonner, M. Luling, and R. Rosthal, "Response of 2-MHz LWD resistivity and wireline induction tools in dipping beds and laminated formations," *The Log Analyst*, Vol. 33, No. 5, 461–475.
33. Habashy, T. M. and M. G. Luling, "A point magnetic dipole radiator in a layered, TI-Anisotropic medium," *Schlumberger-Doll Research Note*, EMG-001-94-14, 87, 1994.

Tarek M. Habashy received the B.Sc. degree from Cairo University, Egypt, and the M.Sc. and Ph.D. degrees from Massachusetts Institute of Technology (MIT), Cambridge, in 1976, 1980, and 1983, respectively, all in electrical engineering. During the academic year 1982–1983, he was a Research Associate in the Department of Electrical Engineering and Computer Science, MIT. Since September 1983, he has been with Schlumberger-Doll Research, Ridgefield, CT, and is now the Director of Mathematic and Modeling Department, where he is also a scientific advisor. He served as a Editor of Radio Science and as a member of the advisory committee board of the book series Progress in Electromagnetic Research. He is currently a member of the editorial boards of Inverse Problems and the Journal of Electromagnetic Waves and Applications.

Aria Abubakar was born in Bandung, Indonesia, on August 21, 1974. He received M.Sc. degree (Cum Laude) in electrical engineering and the Ph.D. degree (Cum Laude) in technical sciences, both from the Delft University of Technology, Delft, The Netherlands, in 1997 and 2000, respectively. In 1996, he was a Research Student at Shell Research B.V., Rijswijk, The Netherlands. He was a summer intern at Schlumberger-doll Research, Ridgefield, Connecticut, USA, From the year 2000 until the beginning of year 2003, he was a Postdoctoral Applied Researcher at the Laboratory of Electromagnetic Research and Section of Applied Geophysics, Delft University of Technology.