Chapter 1
A Bilinear Conjugate-Gradient Inversion Algorithm

1.1 Optimization via Nonlinear Least-Squares

Standard methods for minimizing a real-valued function of several variables can be divided into two general classes: those that require second derivative information, usually referred to as Newton-type methods, and those requiring only first derivative information, referred to as gradient methods. There are several excellent texts which, in addition to discussing many of these methods in detail, also give suggestions on when to use certain techniques. See, for example, the texts by Fletcher[36], Hestenes[52] or Luenberger[64].

In this chapter, we concentrate on gradient techniques for minimizing Φ, the norm of the residuals, for basically two reasons. First, the Töplitz-Hankel structure of the operators in the original volume-integral equation allow us to use fast Fourier transform techniques when doing matrix multiplications in solving the forward problem, and secondly, the bilinearity of the entire system allows us to find the gradient of Φ in closed form, as well as performing exact line searches when minimizing Φ in a particular direction. We first presented this method in [102–104]; it is known in the recent literature as the 'contrast source inversion method' [1, 10, 131].

1.2 A Bilinear Conjugate-Gradient Inversion Algorithm Using Volume-Integrals

Bilinear Inversion Algorithm Consider a T/R configuration, in which a fixed transmitting coil excites the anomaly, which is assumed to have a conductivity vector, $\boldsymbol{\sigma}$, and a receive coil scans the anomaly at positions, $i = 1, \ldots, N_v$, where N_v is the number of 'views'. Conversely, we could assume a single transmitting coil,

H. A. Sabbagh et al., *Advanced Electromagnetic Models for Materials Characterization and Nondestructive Evaluation*, Scientific Computation,
https://doi.org/10.1007/978-3-030-67956-9_1

and an array of N_v receive coils occupying the positions of the original scanned receive coil. Furthermore, we could actually scan fewer than N_v positions, and then interpolate to complete the N_v data points. (See [107] and [14] for further discussions of multiview and multifrequency reconstruction methods.)

In any case, the transmitter generates an anomalous scattering current, $\mathbf{J}^{(x)},\ \mathbf{J}^{(y)},\ \mathbf{J}^{(z)}$, which is computed by **VIC-3D®** and the receive coil(s) produces an incident field, $\mathbf{E}_{0x}^{(R)}(i),\ \mathbf{E}_{0y}^{(R)}(i),\ \mathbf{E}_{0z}^{(R)}(i)$. These combine to produce the change-in-transfer impedance due to the anomaly,

$$Z^{(i)}(\mathbf{J}) = \mathbf{E}_{0x}^{(R)}(i)\cdot\mathbf{J}^{(x)} + \mathbf{E}_{0y}^{(R)}(i)\cdot\mathbf{J}^{(y)} + \mathbf{E}_{0z}^{(R)}(i)\cdot\mathbf{J}^{(z)}\ , \tag{1.1}$$

that is also computed directly by **VIC-3D®**.[1] Note that, unlike our NLSE algorithm, we do not consider $\mathbf{J}^{(x)},\ \mathbf{J}^{(y)},\ \mathbf{J}^{(z)}$ to be secondary variables that are dependent upon the primary unknown, $\boldsymbol{\sigma}$. Rather, the current-vectors along with the cell *resistivities*, $\boldsymbol{\rho}$, are the primary unknowns.

To determine the model equations for the inversion algorithm, return to the fundamental volume-integral electric equation that **VIC-3D®** solves for the anomalous currents:

$$\begin{bmatrix}\mathbf{E}^{(0x)}\\ \mathbf{E}^{(0y)}\\ \mathbf{E}^{(0z)}\end{bmatrix} = \begin{bmatrix}\mathbf{Q}^{(x)}(\boldsymbol{\rho}) & 0 & 0\\ 0 & \mathbf{Q}^{(y)}(\boldsymbol{\rho}) & 0\\ 0 & 0 & \mathbf{Q}^{(z)}(\boldsymbol{\rho})\end{bmatrix}\begin{bmatrix}\mathbf{J}^{(x)}\\ \mathbf{J}^{(y)}\\ \mathbf{J}^{(z)}\end{bmatrix} + \begin{bmatrix}\mathbf{G}^{(xx)} & \mathbf{G}^{(xy)} & \mathbf{G}^{(xz)}\\ \mathbf{G}^{(yx)} & \mathbf{G}^{(yy)} & \mathbf{G}^{(yz)}\\ \mathbf{G}^{(zx)} & \mathbf{G}^{(zy)} & \mathbf{G}^{(zz)}\end{bmatrix}\begin{bmatrix}\mathbf{J}^{(x)}\\ \mathbf{J}^{(y)}\\ \mathbf{J}^{(z)}\end{bmatrix}\ . \tag{1.2}$$

The incident field on the left-hand side of (1.2) is due to the transmitting coil. The total electric-field moments are given by

$$\begin{aligned}\mathbf{E}^{(x)}(\boldsymbol{\rho},\mathbf{J}^{(x)}) &= \mathbf{Q}^{(x)}(\boldsymbol{\rho})\cdot\mathbf{J}^{(x)}\\ \mathbf{E}^{(y)}(\boldsymbol{\rho},\mathbf{J}^{(y)}) &= \mathbf{Q}^{(y)}(\boldsymbol{\rho})\cdot\mathbf{J}^{(y)}\\ \mathbf{E}^{(z)}(\boldsymbol{\rho},\mathbf{J}^{(z)}) &= \mathbf{Q}^{(z)}(\boldsymbol{\rho})\cdot\mathbf{J}^{(z)}\ ,\end{aligned} \tag{1.3}$$

so that when this is substituted back into (1.2), we get the basic constraint equation between the primary variables, $\boldsymbol{\rho}$ and $\mathbf{J}$:

[1] We are considering only 'electric-electric' interactions in this chapter. We assume that all hosts and flaws are nonmagnetic.

$$\begin{bmatrix} \mathbf{E}^{(0x)} \\ \mathbf{E}^{(0y)} \\ \mathbf{E}^{(0z)} \end{bmatrix} = \begin{bmatrix} \mathbf{E}^{(x)}(\boldsymbol{\rho}, \mathbf{J}^{(x)}) \\ \mathbf{E}^{(y)}(\boldsymbol{\rho}, \mathbf{J}^{(y)}) \\ \mathbf{E}^{(z)}(\boldsymbol{\rho}, \mathbf{J}^{(z)}) \end{bmatrix} + \begin{bmatrix} \mathbf{G}^{(xx)} & \mathbf{G}^{(xy)} & \mathbf{G}^{(xz)} \\ \mathbf{G}^{(yx)} & \mathbf{G}^{(yy)} & \mathbf{G}^{(yz)} \\ \mathbf{G}^{(zx)} & \mathbf{G}^{(zy)} & \mathbf{G}^{(zz)} \end{bmatrix} \begin{bmatrix} \mathbf{J}^{(x)} \\ \mathbf{J}^{(y)} \\ \mathbf{J}^{(z)} \end{bmatrix} . \tag{1.4}$$

Thus, the two constraint equations that define the inverse model are the linear system, (1.1), and *bilinear* system, (1.4). A bilinear system is linear in each variable, when the other is held constant. In this case, the variables are $\boldsymbol{\rho}$ and $\mathbf{J}^{(x)}, \mathbf{J}^{(y)}, \mathbf{J}^{(z)}$.

Recall that the tri-diagonal matrices are symmetric, and have the following non-zero entries:

$$\begin{aligned} Q^{(x)}_{kK}(\boldsymbol{\rho}) &= \frac{\delta x \delta y \delta z}{6} \begin{cases} \rho_{klm} & \text{if } K = k-1 \\ 2(\rho_{klm} + \rho_{k+1,lm}) & \text{if } K = k \\ \rho_{k+1,lm} & \text{if } K = k+1 \end{cases} \\ Q^{(y)}_{lL}(\boldsymbol{\rho}) &= \frac{\delta x \delta y \delta z}{6} \begin{cases} \rho_{klm} & \text{if } L = l-1 \\ 2(\rho_{klm} + \rho_{k,l+1,m}) & \text{if } L = l \\ \rho_{k,l+1,m} & \text{if } L = l+1 \end{cases} \\ Q^{(z)}_{mM}(\boldsymbol{\rho}) &= \frac{\delta x \delta y \delta z}{6} \begin{cases} \rho_{klm} & \text{if } M = m-1 \\ 2(\rho_{klm} + \rho_{klm+1}) & \text{if } M = m \\ \rho_{klm+1} & \text{if } M = m+1 \end{cases} \end{aligned} \tag{1.5}$$

The first entry in each of these matrices is the lower diagonal, the second the main diagonal, and the third the upper diagonal. Recall that the resistivity of the klmth cell is the reciprocal of the conductivity, $\rho_{klm} = 1/\sigma_{klm}$. Note that if cell klm is filled with host material, then $\rho_{klm} = \infty$. This fact could be useful in allowing ρ_{klm} to act like a penalty term, forcing the associated current expansion coefficient, J_{klm}, to zero.

Now,

$$\begin{aligned} E^{(x)}_{klm} &= \sum_K Q^{(x)}_{kK} J^{(x)}_{Klm} \\ E^{(y)}_{klm} &= \sum_L Q^{(y)}_{lL} J^{(y)}_{kLm} \\ E^{(z)}_{klm} &= \sum_M Q^{(z)}_{mM} J^{(z)}_{klM} , \end{aligned} \tag{1.6}$$

is the form that is useful when singling out the currents at each cell. When we want to single out the resistivity at each cell, we simply rearrange (1.6):

$$E^{(x)}_{klm} = \sum_K P^{(x)}_{kK} \rho_{Klm}$$

$$E^{(y)}_{klm} = \sum_L P^{(y)}_{lL}\rho_{kLm}$$
$$E^{(z)}_{klm} = \sum_M P^{(z)}_{mM}\rho_{klM}\ , \tag{1.7}$$

where the matrix elements follow from (1.5) and (1.6)

$$P^{(x)}_{kK}(\mathbf{J}^{(x)}) = \frac{\delta x\delta y\delta z}{6}\begin{cases} J^{(x)}_{k-1lm} + 2J^{(x)}_{klm} \text{ if } K = k \\ 2J^{(x)}_{klm} + J^{(x)}_{k+1lm} \text{ if } K = k+1 \end{cases}$$

$$P^{(y)}_{lL}(\mathbf{J}^{(y)}) = \frac{\delta x\delta y\delta z}{6}\begin{cases} J^{(y)}_{kl-1m} + 2J^{(y)}_{klm} \text{ if } L = l \\ 2J^{(y)}_{klm} + J^{(y)}_{kl+1m} \text{ if } L = l+1 \end{cases} \tag{1.8}$$

$$P^{(z)}_{mM}(\mathbf{J}^{(z)}) = \frac{\delta x\delta y\delta z}{6}\begin{cases} J^{(z)}_{klm-1} + 2J^{(z)}_{klm} \text{ if } M = m \\ 2J^{(z)}_{klm} + J^{(z)}_{klm+1} \text{ if } M = m+1 \end{cases}$$

One approach to solving the system, (1.1) and (1.4), has been to use a linear ('Born') approximation, which means that the electric field within the flawed region is assumed to be the same as the electric field if the flaw were not present, namely the incident electric field. Thus, one would write $\mathbf{J} = \sigma\mathbf{E} \approx \sigma\mathbf{E}^0$, and then take moments of this equation. Substituting the resulting expressions for $\mathbf{J}^{(x,y,z)}_{klm}$ into (1.1) will produce a linear system for the unknown conductivities, σ_{klm}, that can be solved in a number of ways. See [105] for an example of this approach, in which the linear system is solved using the *algebraic reconstruction technique* (ART), and [24, 50, 51], as well as [111, pp. 282–285], for details on ART.

This approach, therefore, has the physical significance of ignoring the secondary sources produced by multiple scattering within the flaw. From a mathematical viewpoint, the second term of (1.4) (with the $\mathbf{G}$ matrices) is ignored. While linearization is occasionally accurate, it is not as general as the main subject of this chapter, bilinear conjugate-gradients. See [147] for a further discussion of the Born approximation, and [14] for an application of the Born approximation to solve a scattering problem of a three-dimensional flaw embedded in anisotropic composite materials.

Bilinear Conjugate-Gradients We use nonlinear conjugate-gradients to minimize the functional formed by the sum of squares of the residuals comprising (1.1) and (1.4):

$$\begin{aligned}\Phi(\boldsymbol{\rho}, \mathbf{J}) = \frac{1}{2}\Big\{ &\|\mathbf{E}^{(R)}_{0x}\cdot\mathbf{J}^{(x)} + \mathbf{E}^{(R)}_{0y}\cdot\mathbf{J}^{(y)} + \mathbf{E}^{(R)}_{0z}\cdot\mathbf{J}^{(z)} - \mathbf{Z}_{\text{meas}}\|^2 \\ &+\|\mathbf{E}^{(x)}(\boldsymbol{\rho}, \mathbf{J}^{(x)}) + \mathbf{G}^{(xx)}\cdot\mathbf{J}^{(x)} + \mathbf{G}^{(xy)}\cdot\mathbf{J}^{(y)} + \mathbf{G}^{(xz)}\cdot\mathbf{J}^{(z)} - \mathbf{E}^{(0x)}\|^2 \\ &+\|\mathbf{E}^{(y)}(\boldsymbol{\rho}, \mathbf{J}^{(y)}) + \mathbf{G}^{(yx)}\cdot\mathbf{J}^{(x)} + \mathbf{G}^{(yy)}\cdot\mathbf{J}^{(y)} + \mathbf{G}^{(yz)}\cdot\mathbf{J}^{(z)} - \mathbf{E}^{(0y)}\|^2\end{aligned}$$

$$
\begin{aligned}
&+\|\mathbf{E}^{(z)}(\boldsymbol{\rho},\mathbf{J}^{(z)})+\mathbf{G}^{(zx)}\cdot\mathbf{J}^{(x)}+\mathbf{G}^{(zy)}\cdot\mathbf{J}^{(y)}+\mathbf{G}^{(zz)}\cdot\mathbf{J}^{(z)}-\mathbf{E}^{(0z)}\|^2\Big\}\\
&=\frac{1}{2}\Big\{\|\mathscr{A}^{(R)}(\mathbf{J})-\mathbf{Z}_{\text{meas}}\|^2\\
&\quad+\|\mathscr{A}^{(x)}(\boldsymbol{\rho},\mathbf{J})-\mathbf{E}^{(0x)}\|^2+\|\mathscr{A}^{(y)}(\boldsymbol{\rho},\mathbf{J})-\mathbf{E}^{(0y)}\|^2+\|\mathscr{A}^{(z)}(\boldsymbol{\rho},\mathbf{J})-\mathbf{E}^{(0z)}\|^2\Big\}\\
&=\Phi^{(R)}(\mathbf{J})+\Phi^{(x)}(\boldsymbol{\rho},\mathbf{J})+\Phi^{(y)}(\boldsymbol{\rho},\mathbf{J})+\Phi^{(z)}(\boldsymbol{\rho},\mathbf{J}).
\end{aligned}
\tag{1.9}
$$

Now let's say a few words about the vector-matrix structure of these functionals. As shown in (1.1), $\mathscr{A}^{(R)}-\mathbf{Z}_{\text{meas}}$ is an $N_v\times 1$-vector, whose components are given by (1.1). This suggests that we should think of $\mathbf{E}^{(R)}_{0x}$, $\mathbf{E}^{(R)}_{0y}$, $\mathbf{E}^{(R)}_{0z}$, as $N_v\times N_c$-dimensional matrices, $E^{(R)}_{0x(i,klm)}$, $E^{(R)}_{0y(i,klm)}$, $E^{(R)}_{0z(i,klm)}$, $i=1,\ldots,N_v$, $klm=1,\ldots,N_c$, where N_c is the number of cells in the problem. Then, we have

$$
\begin{aligned}
\Phi^{(R)}(\mathbf{J})&=\frac{1}{2}\|\mathscr{A}^{(R)}(\mathbf{J})-\mathbf{Z}_{\text{meas}}\|^2\\
&=\frac{1}{2}\sum_{i=1}^{N_v}|\mathscr{A}^{(R)}_i(\mathbf{J})-Z_{\text{meas}}(i)|^2\\
&=\frac{1}{2}\sum_{i=1}^{N_v}\Big(\mathscr{A}^{(R)}_i(\mathbf{J})-Z_{\text{meas}}(i)\Big)\Big(\mathscr{A}^{(R)}_i(\mathbf{J})-Z_{\text{meas}}(i)\Big)^*\\
&=\frac{1}{2}\sum_{i=1}^{N_v}\Big(\mathbf{E}^{(R)}_{0x}(i)\cdot\mathbf{J}^{(x)}+\mathbf{E}^{(R)}_{0y}(i)\cdot\mathbf{J}^{(y)}+\mathbf{E}^{(R)}_{0z}(i)\cdot\mathbf{J}^{(z)}-Z_{\text{meas}}(i)\Big)\\
&\qquad\Big(\mathbf{E}^{(R)}_{0x}(i)\cdot\mathbf{J}^{(x)}+\mathbf{E}^{(R)}_{0y}(i)\cdot\mathbf{J}^{(y)}+\mathbf{E}^{(R)}_{0z}(i)\cdot\mathbf{J}^{(z)}-Z_{\text{meas}}(i)\Big)^*,
\end{aligned}
\tag{1.10}
$$

where * denotes the complex-conjugate, and the vector dot-product in the last expression implies a sum over the cell indices, klm.

It is a straightforward calculation, using the final form of (1.10), to show that the gradient of $\Phi^{(R)}(\mathbf{J})$ with respect to its primary variables, $J^{(x)}_{klm}$, $J^{(y)}_{klm}$, and $J^{(z)}_{klm}$ is

$$
\begin{aligned}
\frac{\partial\Phi^{(R)}}{\partial J^{(x)}_{klm}}&=\sum_i E^{(R)H}_{0x(klm,i)}\Big(\mathbf{E}^{(R)}_{0x}(i)\cdot\mathbf{J}^{(x)}+\mathbf{E}^{(R)}_{0y}(i)\cdot\mathbf{J}^{(y)}+\mathbf{E}^{(R)}_{0z}(i)\cdot\mathbf{J}^{(z)}-Z_{\text{meas}}(i)\Big)\\
\frac{\partial\Phi^{(R)}}{\partial J^{(y)}_{klm}}&=\sum_i E^{(R)H}_{0y(klm,i)}\Big(\mathbf{E}^{(R)}_{0x}(i)\cdot\mathbf{J}^{(x)}+\mathbf{E}^{(R)}_{0y}(i)\cdot\mathbf{J}^{(y)}+\mathbf{E}^{(R)}_{0z}(i)\cdot\mathbf{J}^{(z)}-Z_{\text{meas}}(i)\Big)\\
\frac{\partial\Phi^{(R)}}{\partial J^{(z)}_{klm}}&=\sum_i E^{(R)H}_{0z(klm,i)}\Big(\mathbf{E}^{(R)}_{0x}(i)\cdot\mathbf{J}^{(x)}+\mathbf{E}^{(R)}_{0y}(i)\cdot\mathbf{J}^{(y)}+\mathbf{E}^{(R)}_{0z}(i)\cdot\mathbf{J}^{(z)}-Z_{\text{meas}}(i)\Big),
\end{aligned}
\tag{1.11}
$$

where the superscript, H, on a matrix denotes the Hermitian of that matrix, namely the complex-conjugate of the transpose of the matrix. The term in parentheses is the ith component of the residual-vector, $\mathbf{RESID}(R)$.

The gradients of $\Phi^{(x)}$, $\Phi^{(y)}$, and $\Phi^{(z)}$ with respect to $\mathbf{J}^{(x)}$, $\mathbf{J}^{(y)}$, and $\mathbf{J}^{(z)}$ are given by:

$$\begin{aligned}
\frac{\partial \Phi^{(x)}}{\partial J^{(x)}_{klm}} &= \sum_{KLM} \left(Q^{(x)T}_{klm,KLM} + G^{(xx)H}_{klm,KLM} \right) (\mathbf{RESID}(X))_{KLM} \\
\frac{\partial \Phi^{(x)}}{\partial J^{(y)}_{klm}} &= \sum_{KLM} G^{(xy)H}_{klm,KLM} \, (\mathbf{RESID}(X))_{KLM} \\
\frac{\partial \Phi^{(x)}}{\partial J^{(z)}_{klm}} &= \sum_{KLM} G^{(xz)H}_{klm,KLM} \, (\mathbf{RESID}(X))_{KLM} \\
\frac{\partial \Phi^{(y)}}{\partial J^{(x)}_{klm}} &= \sum_{KLM} G^{(yx)H}_{klm,KLM} \, (\mathbf{RESID}(Y))_{KLM} \\
\frac{\partial \Phi^{(y)}}{\partial J^{(y)}_{klm}} &= \sum_{KLM} \left(Q^{(y)T}_{klm,KLM} + G^{(yy)H}_{klm,KLM} \right) (\mathbf{RESID}(Y))_{KLM} \\
\frac{\partial \Phi^{(y)}}{\partial J^{(z)}_{klm}} &= \sum_{KLM} G^{(yz)H}_{klm,KLM} \, (\mathbf{RESID}(Y))_{KLM} \\
\frac{\partial \Phi^{(z)}}{\partial J^{(x)}_{klm}} &= \sum_{KLM} G^{(zx)H}_{klm,KLM} \, (\mathbf{RESID}(Z))_{KLM} \\
\frac{\partial \Phi^{(z)}}{\partial J^{(y)}_{klm}} &= \sum_{KLM} G^{(zy)H}_{klm,KLM} \, (\mathbf{RESID}(Z))_{KLM} \\
\frac{\partial \Phi^{(z)}}{\partial J^{(z)}_{klm}} &= \sum_{KLM} \left(Q^{(z)T}_{klm,KLM} + G^{(zz)H}_{klm,KLM} \right) (\mathbf{RESID}(Z))_{KLM} \ , \qquad (1.12)
\end{aligned}$$

where

$$\begin{aligned}
\mathbf{RESID}(X) &= \mathbf{E}^{(x)}(\boldsymbol{\rho}, \mathbf{J}^{(x)}) + \mathbf{G}^{xx} \cdot \mathbf{J}^{(x)} + \mathbf{G}^{(xy)} \cdot \mathbf{J}^{(y)} + \mathbf{G}^{(xz)} \cdot \mathbf{J}^{(z)} - \mathbf{E}^{(0x)} \\
\mathbf{RESID}(Y) &= \mathbf{E}^{(y)}(\boldsymbol{\rho}, \mathbf{J}^{(y)}) + \mathbf{G}^{yx} \cdot \mathbf{J}^{(x)} + \mathbf{G}^{(yy)} \cdot \mathbf{J}^{(y)} + \mathbf{G}^{(yz)} \cdot \mathbf{J}^{(z)} - \mathbf{E}^{(0y)} \\
\mathbf{RESID}(Z) &= \mathbf{E}^{(z)}(\boldsymbol{\rho}, \mathbf{J}^{(z)}) + \mathbf{G}^{zx} \cdot \mathbf{J}^{(x)} + \mathbf{G}^{(zy)} \cdot \mathbf{J}^{(y)} + \mathbf{G}^{(zz)} \cdot \mathbf{J}^{(z)} - \mathbf{E}^{(0z)} \ . \qquad (1.13)
\end{aligned}$$

Note We use the notation for the gradient of a real function with respect to a complex variable to mean: $\dfrac{\partial \Phi}{\partial J} = \dfrac{\partial \Phi}{\partial R} + j \dfrac{\partial \Phi}{\partial I}$, where $R = \Re J$ and $I = \Im J$.

The final gradients are those of $\Phi^{(x)}$, $\Phi^{(y)}$, and $\Phi^{(z)}$ with respect to $\boldsymbol{\rho}$:

$$\begin{aligned}
\frac{\partial \Phi^{(x)}}{\partial \rho_{klm}} &= \Re \sum_{KLM} P^{(x)H}_{klm,KLM} \, (\mathbf{RESID}(X))_{KLM} \\
\frac{\partial \Phi^{(y)}}{\partial \rho_{klm}} &= \Re \sum_{KLM} P^{(y)H}_{klm,KLM} \, (\mathbf{RESID}(Y))_{KLM} \\
\frac{\partial \Phi^{(z)}}{\partial \rho_{klm}} &= \Re \sum_{KLM} P^{(z)H}_{klm,KLM} \, (\mathbf{RESID}(Z))_{KLM} \; .
\end{aligned} \tag{1.14}$$

Remember that the P-matrices that are defined in (1.7) and (1.8) are complex, because they are explicit functions of the complex currents.

The gradient of $\Phi(\boldsymbol{\rho}, \mathbf{J})$ is the sum of the various sub-gradients:

$$\begin{aligned}
\frac{\partial \Phi}{\partial J^{(x)}_{klm}} &= \frac{\partial \Phi^{(R)}}{\partial J^{(x)}_{klm}} + \frac{\partial \Phi^{(x)}}{\partial J^{(x)}_{klm}} + \frac{\partial \Phi^{(y)}}{\partial J^{(x)}_{klm}} + \frac{\partial \Phi^{(z)}}{\partial J^{(x)}_{klm}} \\
\frac{\partial \Phi}{\partial J^{(y)}_{klm}} &= \frac{\partial \Phi^{(R)}}{\partial J^{(y)}_{klm}} + \frac{\partial \Phi^{(x)}}{\partial J^{(y)}_{klm}} + \frac{\partial \Phi^{(y)}}{\partial J^{(y)}_{klm}} + \frac{\partial \Phi^{(z)}}{\partial J^{(y)}_{klm}} \\
\frac{\partial \Phi}{\partial J^{(z)}_{klm}} &= \frac{\partial \Phi^{(R)}}{\partial J^{(z)}_{klm}} + \frac{\partial \Phi^{(x)}}{\partial J^{(z)}_{klm}} + \frac{\partial \Phi^{(y)}}{\partial J^{(z)}_{klm}} + \frac{\partial \Phi^{(z)}}{\partial J^{(z)}_{klm}} \\
\frac{\partial \Phi}{\partial \rho_{klm}} &= \frac{\partial \Phi^{(x)}}{\partial \rho_{klm}} + \frac{\partial \Phi^{(y)}}{\partial \rho_{klm}} + \frac{\partial \Phi^{(z)}}{\partial \rho_{klm}} \; .
\end{aligned} \tag{1.15}$$

Eventually we will need to minimize Φ along the line $(\rho, \mathbf{J}^{(x)}, \mathbf{J}^{(y)}, \mathbf{J}^{(z)}) + \alpha(\mathbf{u}, \mathbf{v}^{(x)}, \mathbf{v}^{(y)}, \mathbf{v}^{(z)})$ in function-space. α is a real number that parameterizes the line, $(\boldsymbol{\rho}, \mathbf{J}^{(x)}, \mathbf{J}^{(y)}, \mathbf{J}^{(z)})$ is a fixed starting point, and $(\mathbf{u}, \mathbf{v}^{(x)}, \mathbf{v}^{(y)}, \mathbf{v}^{(z)})$ is a direction-vector that will be determined by the conjugate-gradient algorithm. We will do this by differentiating Φ with respect to α, and then set the derivative equal to zero to determine the optimum α.

From (1.9), we have

$$\begin{aligned}
&\Phi(\boldsymbol{\rho} + \alpha\mathbf{u}, \mathbf{J}^{(x)} + \alpha\mathbf{v}^{(x)}, \mathbf{J}^{(y)} + \alpha\mathbf{v}^{(y)}, \mathbf{J}^{(z)} + \alpha\mathbf{v}^{(z)}) = \\
&\Phi^{(R)}(\mathbf{J}^{(x)} + \alpha\mathbf{v}^{(x)}, \mathbf{J}^{(y)} + \alpha\mathbf{v}^{(y)}, \mathbf{J}^{(z)} + \alpha\mathbf{v}^{(z)}) + \\
&\Phi^{(x)}(\boldsymbol{\rho} + \alpha\mathbf{u}, \mathbf{J}^{(x)} + \alpha\mathbf{v}^{(x)}, \mathbf{J}^{(y)} + \alpha\mathbf{v}^{(y)}, \mathbf{J}^{(z)} + \alpha\mathbf{v}^{(z)}) + \\
&\Phi^{(y)}(\boldsymbol{\rho} + \alpha\mathbf{u}, \mathbf{J}^{(x)} + \alpha\mathbf{v}^{(x)}, \mathbf{J}^{(y)} + \alpha\mathbf{v}^{(y)}, \mathbf{J}^{(z)} + \alpha\mathbf{v}^{(z)}) + \\
&\Phi^{(z)}(\boldsymbol{\rho} + \alpha\mathbf{u}, \mathbf{J}^{(x)} + \alpha\mathbf{v}^{(x)}, \mathbf{J}^{(y)} + \alpha\mathbf{v}^{(y)}, \mathbf{J}^{(z)} + \alpha\mathbf{v}^{(z)}) \; .
\end{aligned} \tag{1.16}$$

Clearly, $\Phi^{(R)}(\mathbf{J} + \alpha\mathbf{v})$ is quadratic in α, but the other three functionals are of the fourth-order in α, as can be seen by the bilinear function $\mathbf{E}(\boldsymbol{\rho}, \mathbf{J})$ that is defined in (1.3) and appears in (1.9).

The derivative of $\Phi^{(R)}$ is computed in a straight-forward (though slightly tedious) manner to be:

$$\frac{d\Phi^{(R)}(\mathbf{J}+\alpha\mathbf{v})}{d\alpha} = \Re\left\{\mathbf{v}^{(x)*}\cdot\frac{\partial\Phi^{(R)}(\mathbf{J})}{\partial\mathbf{J}^{(x)}} + \mathbf{v}^{(y)*}\cdot\frac{\partial\Phi^{(R)}(\mathbf{J})}{\partial\mathbf{J}^{(y)}} + \mathbf{v}^{(z)*}\cdot\frac{\partial\Phi^{(R)}(\mathbf{J})}{\partial\mathbf{J}^{(z)}} + \alpha\left(\mathbf{v}^{(x)*}\cdot\frac{\partial\Phi^{(R)}(\mathbf{v})}{\partial\mathbf{J}^{(x)}} + \mathbf{v}^{(y)*}\cdot\frac{\partial\Phi^{(R)}(\mathbf{v})}{\partial\mathbf{J}^{(y)}} + \mathbf{v}^{(z)*}\cdot\frac{\partial\Phi^{(R)}(\mathbf{v})}{\partial\mathbf{J}^{(z)}}\right)\right\} . \quad (1.17)$$

The vector dot-product denotes a sum over the cell indices, klm, as before, and the components of the gradient vectors are given in (1.11).

Similarly,

$$\begin{aligned}
&\frac{d\Phi^{(x)}}{d\alpha}(\boldsymbol{\rho}+\alpha\mathbf{u},\mathbf{J}+\alpha\mathbf{v})\\
&= \Re\left\{\mathbf{A}^*_{0x}\cdot\mathbf{B}_{0x} + \alpha\left(\mathbf{A}^*_{0x}\cdot\mathbf{B}_{1x} + \mathbf{A}^*_{1x}\cdot\mathbf{B}_{0x}\right)\right.\\
&\quad\left.+\alpha^2\left(\mathbf{A}^*_{0x}\cdot\mathbf{B}_{2x} + \mathbf{A}^*_{1x}\cdot\mathbf{B}_{1x}\right) + \alpha^3\mathbf{A}^*_{1x}\cdot\mathbf{B}_{2x}\right\}\\
&\frac{d\Phi^{(y)}}{d\alpha}(\boldsymbol{\rho}+\alpha\mathbf{u},\mathbf{J}+\alpha\mathbf{v})\\
&= \Re\left\{\mathbf{A}^*_{0y}\cdot\mathbf{B}_{0y} + \alpha\left(\mathbf{A}^*_{0y}\cdot\mathbf{B}_{1y} + \mathbf{A}^*_{1y}\cdot\mathbf{B}_{0y}\right)\right.\\
&\quad\left.+\alpha^2\left(\mathbf{A}^*_{0y}\cdot\mathbf{B}_{2y} + \mathbf{A}^*_{1y}\cdot\mathbf{B}_{1y}\right) + \alpha^3\mathbf{A}^*_{1y}\cdot\mathbf{B}_{2y}\right\}\\
&\frac{d\Phi^{(z)}}{d\alpha}(\boldsymbol{\rho}+\alpha\mathbf{u},\mathbf{J}+\alpha\mathbf{v})\\
&= \Re\left\{\mathbf{A}^*_{0z}\cdot\mathbf{B}_{0z} + \alpha\left(\mathbf{A}^*_{0z}\cdot\mathbf{B}_{1z} + \mathbf{A}^*_{1z}\cdot\mathbf{B}_{0z}\right)\right.\\
&\quad\left.+\alpha^2\left(\mathbf{A}^*_{0z}\cdot\mathbf{B}_{2z} + \mathbf{A}^*_{1z}\cdot\mathbf{B}_{1z}\right) + \alpha^3\mathbf{A}^*_{1z}\cdot\mathbf{B}_{2z}\right\} \quad (1.18)
\end{aligned}$$

where

$$\begin{aligned}
\mathbf{A}^*_{0x} &= \mathbf{J}^{(x)*}\cdot\mathbf{Q}^{(x)T}(\mathbf{u}) + \mathbf{v}^{(x)*}\cdot\mathbf{Q}^{(x)T}(\boldsymbol{\rho}) + \mathbf{v}^{(x)*}\cdot\mathbf{G}^{(xx)H}\\
&\quad+\mathbf{v}^{(y)*}\cdot\mathbf{G}^{(xy)H} + \mathbf{v}^{(z)*}\cdot\mathbf{G}^{(xz)H}\\
\mathbf{A}^*_{1x} &= 2\mathbf{v}^{(x)*}\cdot\mathbf{Q}^{(x)T}(\mathbf{u})\\
\mathbf{B}_{0x} &= \mathbf{Q}^{(x)}(\boldsymbol{\rho})\cdot\mathbf{J}^{(x)} + \mathbf{G}^{(xx)}\cdot\mathbf{J}^{(x)}\\
&\quad+\mathbf{G}^{(xy)}\cdot\mathbf{J}^{(y)} + \mathbf{G}^{(xz)}\cdot\mathbf{J}^{(z)} - \mathbf{E}^{(0x)}\\
\mathbf{B}_{1x} &= \mathbf{Q}^{(x)}(\mathbf{u})\cdot\mathbf{J}^{(x)} + \mathbf{Q}^{(x)}(\boldsymbol{\rho})\cdot\mathbf{v}^{(x)} + \mathbf{G}^{(xx)}\cdot\mathbf{v}^{(x)}\\
&\quad+\mathbf{G}^{(xy)}\cdot\mathbf{v}^{(y)} + \mathbf{G}^{(xz)}\cdot\mathbf{v}^{(z)}
\end{aligned}$$

$$
\begin{aligned}
\mathbf{B}_{2x} &= \mathbf{Q}^{(x)}(\mathbf{u}) \cdot \mathbf{v}^{(x)} \\
\mathbf{A}^*_{0y} &= \mathbf{J}^{(y)*} \cdot \mathbf{Q}^{(y)T}(\mathbf{u}) + \mathbf{v}^{(y)*} \cdot \mathbf{Q}^{(y)T}(\boldsymbol{\rho}) + \mathbf{v}^{(x)*} \cdot \mathbf{G}^{(yx)H} \\
&\quad + \mathbf{v}^{(y)*} \cdot \mathbf{G}^{(yy)H} + \mathbf{v}^{(z)*} \cdot \mathbf{G}^{(yz)H} \\
\mathbf{A}^*_{1y} &= 2\mathbf{v}^{(y)*} \cdot \mathbf{Q}^{(y)T}(\mathbf{u}) \\
\mathbf{B}_{0y} &= \mathbf{Q}^{(y)}(\boldsymbol{\rho}) \cdot \mathbf{J}^{(y)} + \mathbf{G}^{(yx)} \cdot \mathbf{J}^{(x)} \\
&\quad + \mathbf{G}^{(yy)} \cdot \mathbf{J}^{(y)} + \mathbf{G}^{(yz)} \cdot \mathbf{J}^{(z)} - \mathbf{E}^{(0y)} \\
\mathbf{B}_{1y} &= \mathbf{Q}^{(y)}(\mathbf{u}) \cdot \mathbf{J}^{(y)} + \mathbf{Q}^{(y)}(\boldsymbol{\rho}) \cdot \mathbf{v}^{(y)} + \mathbf{G}^{(yx)} \cdot \mathbf{v}^{(x)} \\
&\quad + \mathbf{G}^{(yy)} \cdot \mathbf{v}^{(y)} + \mathbf{G}^{(yz)} \cdot \mathbf{v}^{(z)} \\
\mathbf{B}_{2y} &= \mathbf{Q}^{(y)}(\mathbf{u}) \cdot \mathbf{v}^{(y)} \\
\mathbf{A}^*_{0z} &= \mathbf{J}^{(z)*} \cdot \mathbf{Q}^{(z)T}(\mathbf{u}) + \mathbf{v}^{(z)*} \cdot \mathbf{Q}^{(z)T}(\boldsymbol{\rho}) + \mathbf{v}^{(x)*} \cdot \mathbf{G}^{(zx)H} \\
&\quad + \mathbf{v}^{(y)*} \cdot \mathbf{G}^{(zy)H} + \mathbf{v}^{(z)*} \cdot \mathbf{G}^{(zz)H} \\
\mathbf{A}^*_{1z} &= 2\mathbf{v}^{(z)*} \cdot \mathbf{Q}^{(z)T}(\mathbf{u}) \\
\mathbf{B}_{0z} &= \mathbf{Q}^{(z)}(\boldsymbol{\rho}) \cdot \mathbf{J}^{(z)} + \mathbf{G}^{(zx)} \cdot \mathbf{J}^{(x)} \\
&\quad + \mathbf{G}^{(zy)} \cdot \mathbf{J}^{(y)} + \mathbf{G}^{(zz)} \cdot \mathbf{J}^{(z)} - \mathbf{E}^{(0z)} \\
\mathbf{B}_{1z} &= \mathbf{Q}^{(z)}(\mathbf{u}) \cdot \mathbf{J}^{(z)} + \mathbf{Q}^{(z)}(\boldsymbol{\rho}) \cdot \mathbf{v}^{(z)} + \mathbf{G}^{(zx)} \cdot \mathbf{v}^{(x)} \\
&\quad + \mathbf{G}^{(zy)} \cdot \mathbf{v}^{(y)} + \mathbf{G}^{(zz)} \cdot \mathbf{v}^{(z)} \\
\mathbf{B}_{2z} &= \mathbf{Q}^{(z)}(\mathbf{u}) \cdot \mathbf{v}^{(z)} \,, \qquad (1.19)
\end{aligned}
$$

and, once again, the vector dot-product denotes a sum over the cell indices, klm. Note that $\mathbf{B}_0$ is a residual-vector, which means that there may be some simplification in computing or using it, and finally, the order of some of the computations in (1.19) might be changed in order to take advantage of the convolutional or correlational nature of the matrices.

The total cubic polynomial for $\dfrac{d\Phi}{d\alpha}(\boldsymbol{\rho} + \alpha\mathbf{u}, \mathbf{J} + \alpha\mathbf{v}) = 0$ is given by setting the sum of (1.17) and (1.18) equal to zero. Note that, because the coefficients of the polynomial are real, any complex roots must occur in conjugate pairs. This guarantees at least one real root, and possibly three. We will take the smallest real root.

1.3 The Algorithm

Step 0: Initialization The user creates a starting point

$$\begin{pmatrix} \mathbf{J}_0^{(x)} \\ \mathbf{J}_0^{(y)} \\ \mathbf{J}_0^{(z)} \\ \boldsymbol{\rho}_0 \end{pmatrix}.$$

Step 1: Steepest Descent First calculate the gradient

$$\nabla \Phi(\boldsymbol{\rho}_0, \mathbf{J}_0) = \begin{pmatrix} \dfrac{\partial \Phi}{\partial \mathbf{J}^{(x)}} \\ \dfrac{\partial \Phi}{\partial \mathbf{J}^{(y)}} \\ \dfrac{\partial \Phi}{\partial \mathbf{J}^{(z)}} \\ \dfrac{\partial \Phi}{\partial \boldsymbol{\rho}} \end{pmatrix} (\boldsymbol{\rho}_0, \mathbf{J}_0)$$

using (1.15). Then set the direction of movement at the first iteration to be

$$\begin{pmatrix} \mathbf{v}^{(x)} \\ \mathbf{v}^{(y)} \\ \mathbf{v}^{(z)} \\ \mathbf{u} \end{pmatrix} = -\nabla \Phi(\boldsymbol{\rho}_0, \mathbf{J}_0)$$

where $(\mathbf{v}^{(x)}, \mathbf{v}^{(y)}, \mathbf{v}^{(z)}) = \mathbf{v} \in \mathbf{C}^{N_c} \times \mathbf{C}^{N_c} \times \mathbf{C}^{N_c}$, and $\mathbf{u} \in \mathbf{R}^{N_c}$. Note that we will later use $\mathbf{f}_1$ to denote this direction. The problem now is to minimize Φ in the direction $\begin{pmatrix} \mathbf{v} \\ \mathbf{u} \end{pmatrix}$ from the point $(\boldsymbol{\rho}_0, \mathbf{J}_0)$. We should normalize the direction vector $\begin{pmatrix} \mathbf{v}^{(x)} \\ \mathbf{v}^{(y)} \\ \mathbf{v}^{(z)} \\ \mathbf{u} \end{pmatrix}$ before calculating the coefficients of the cubic equation that gives the minimum value. Therefore, define the new direction vector to be the unit vector

$$\frac{1}{\sqrt{(\|\mathbf{v}\|^2 + \|\mathbf{u}\|^2)}} \begin{pmatrix} \mathbf{v}^{(x)} \\ \mathbf{v}^{(y)} \\ \mathbf{v}^{(z)} \\ \mathbf{u} \end{pmatrix}$$

and note that we will use the same notation, $\begin{pmatrix} \mathbf{v}^{(x)} \\ \mathbf{v}^{(y)} \\ \mathbf{v}^{(z)} \\ \mathbf{u} \end{pmatrix}$, for the normalized (unit) direction vector. Now find the smallest positive α_0 that satisfies

$$a_0 + a_1\alpha + a_2\alpha^2 + a_3\alpha^3 = 0$$

where the coefficients are given by the appropriate sums of (1.17) and (1.18), with $\mathbf{J} = \mathbf{J}_0$ and $\boldsymbol{\rho} = \boldsymbol{\rho}_0$. Then define the new approximation to be

$$\begin{pmatrix} \mathbf{J}_1 \\ \boldsymbol{\rho}_1 \end{pmatrix} = \begin{pmatrix} \mathbf{J}_0 \\ \boldsymbol{\rho}_0 \end{pmatrix} + \alpha_0 \begin{pmatrix} \mathbf{v} \\ \mathbf{u} \end{pmatrix} .$$

Step $k+1$ Assume that the kth ($k \geq 1$) iteration, $\begin{pmatrix} \mathbf{J}_k \\ \boldsymbol{\rho}_k \end{pmatrix}$, has been determined. Let $\{\mathbf{f}_k\}$ be the sequence of direction vectors that is generated by defining $\mathbf{f}_1 = -\nabla\Phi(\mathbf{J}_0, \boldsymbol{\rho}_0)$ and for $k \geq 1$,

$$\mathbf{f}_{k+1} = -\nabla\Phi(\mathbf{J}_k, \boldsymbol{\rho}_k) + \beta_{k+1}\mathbf{f}_k ,$$

where $\nabla\Phi(\mathbf{J}_k, \boldsymbol{\rho}_k)$ and β_{k+1} are determined as follows:

$$\nabla\Phi(\boldsymbol{\rho}_k, \mathbf{J}_k) = \begin{pmatrix} \dfrac{\partial\Phi}{\partial\mathbf{J}^{(x)}} \\ \dfrac{\partial\Phi}{\partial\mathbf{J}^{(y)}} \\ \dfrac{\partial\Phi}{\partial\mathbf{J}^{(z)}} \\ \dfrac{\partial\Phi}{\partial\boldsymbol{\rho}} \end{pmatrix} (\boldsymbol{\rho}_k, \mathbf{J}_k) ,$$

and the parameter, β_{k+1}, depends upon one of the following iteration methods:

Steepest Descent This technique simply continues as **Step 1**, and defines the direction vector to always be the steepest descent direction. Hence, β_{k+1} will always be set equal to 0.

Fletcher-Reeves Here, β_{k+1} is defined as

$$\beta_{k+1} = \frac{\|\nabla\Phi(\mathbf{J}_k, \boldsymbol{\rho}_k)\|^2}{\|\nabla\Phi(\mathbf{J}_{k-1}, \boldsymbol{\rho}_{k-1})\|^2} .$$

Polak-Ribière In this algorithm, β_{k+1} is defined as

$$\beta_{k+1} = \frac{\left(\nabla\Phi(\mathbf{J}_k, \boldsymbol{\rho}_k) - \nabla\Phi(\mathbf{J}_{k-1}, \boldsymbol{\rho}_{k-1})\right)^* \cdot \nabla\Phi(\mathbf{J}_k, \boldsymbol{\rho}_k)}{\|\nabla\Phi(\mathbf{J}_{k-1}, \boldsymbol{\rho}_{k-1})\|^2} .$$

Fletcher-Reeves with Restart This method is similar to the Fletcher-Reeves algorithm except that after every r iterations, we restart the algorithm with the steepest descent step. Therefore, we can define β_{k+1} as

$$\beta_{k+1} = \begin{cases} 0 & \text{if } k \text{ is a multiple of } r \\ \dfrac{\|\nabla\Phi(\mathbf{J}_k, \boldsymbol{\rho}_k)\|^2}{\|\nabla\Phi(\mathbf{J}_{k-1}, \boldsymbol{\rho}_{k-1})\|^2} & \text{otherwise} \end{cases}$$

Hybrid 1 This hybrid algorithm tries to take into account the best aspects of some of the algorithms presented above.

$$\beta_{k+1} = \begin{cases} 0 \quad \text{if } \|\nabla\Phi(\mathbf{J}_k, \boldsymbol{\rho}_k)\|^2 > (0.2)^k \times 10^8 \\ \beta_{\text{FR}} \text{ if } \beta_{\text{PR}} < 0 \\ \beta_{\text{PR}} \text{ if } \beta_{\text{PR}} \le 5\dfrac{\|\nabla\Phi(\mathbf{J}_k, \boldsymbol{\rho}_k)\|^2}{\|\nabla\phi(\mathbf{J}_{k-1}, \boldsymbol{\rho}_{k-1})\|^2} = 5\beta_{\text{FR}} \\ \beta_{\text{FR}} \text{ otherwise} \end{cases}$$

Hybrid 2 In this algorithm, we try to control the orthogonality of the gradients and define β_{k+1} as

$$\beta_{k+1} = \begin{cases} 0 & \text{if } |\nabla\Phi(\mathbf{J}_{k-1}, \boldsymbol{\rho}_{k-1})^* \cdot \nabla\Phi(\mathbf{J}_k, \boldsymbol{\rho}_k)| > 0.2 \\ \dfrac{\|\nabla\Phi(\mathbf{J}_k, \boldsymbol{\rho}_k)\|^2}{\|\nabla\Phi(\mathbf{J}_{k-1}, \boldsymbol{\rho}_{k-1})\|^2} & \text{otherwise} \end{cases}$$

After determining β_{k+1}, we set the new direction vector

$$\begin{pmatrix} \mathbf{v}^{(x)} \\ \mathbf{v}^{(y)} \\ \mathbf{v}^{(z)} \\ \mathbf{u} \end{pmatrix} = \mathbf{f}_{k+1} = -\nabla\Phi(\boldsymbol{\rho}_k, \mathbf{J}_k) + \beta_{k+1}\mathbf{f}_k$$

where, as before $(\mathbf{v}^{(x)}, \mathbf{v}^{(y)}, \mathbf{v}^{(z)}) \in \mathbf{C}^{N_c} \times \mathbf{C}^{N_c} \times \mathbf{C}^{N_c}$, and $\mathbf{u} \in \mathbf{R}^{N_c}$. As in **Step 1**, we should normalize $\begin{pmatrix} \mathbf{v} \\ \mathbf{u} \end{pmatrix}$ to make it a unit vector, and then find the smallest positive α_k that satisfies

$$a_0 + a_1\alpha + a_2\alpha^2 + a_3\alpha^3 = 0$$

with coefficients gotten from (1.17)–(1.19) and the various functions evaluated at the current point, $(\mathbf{J}_k, \boldsymbol{\rho}_k)$.

Since we are able to use exact line searches in this algorithm, most of the methods which we use will always yield a direction in which Φ is decreasing. In fact, since we normalize the direction vector, the constant in the cubic equation, a_0, is equal to the directional derivative of Φ at the point $(\mathbf{J}_k, \boldsymbol{\rho}_k)$ in the direction $\begin{pmatrix} \mathbf{v} \\ \mathbf{u} \end{pmatrix}$. Therefore, a_0 should always be negative. We have found, however, that even in the methods which theoretically always yield descent directions, we sometimes will obtain a positive a_0. Usually this phenomenon occurs when we are 'close' to the actual solution. We suggest that if $a_0 \geq 0$, then the algorithm should be restarted by going to the initialization step, and setting $(\mathbf{J}_0, \boldsymbol{\rho}_0) = (\mathbf{J}_k, \boldsymbol{\rho}_k)$. This is equivalent to using the steepest descent direction (i.e., letting $\beta_{k+1} = 0$) at **Step** $k + 1$.

Now define the new iteration to be

$$\begin{pmatrix} \mathbf{J}_{k+1}^{(x)} \\ \mathbf{J}_{k+1}^{(y)} \\ \mathbf{J}_{k+1}^{(z)} \\ \boldsymbol{\rho}_{k+1} \end{pmatrix} = \begin{pmatrix} \mathbf{J}_{k}^{(x)} \\ \mathbf{J}_{k}^{(y)} \\ \mathbf{J}_{k}^{(z)} \\ \boldsymbol{\rho}_{k} \end{pmatrix} + \alpha_k \begin{pmatrix} \mathbf{v}^{(x)} \\ \mathbf{v}^{(y)} \\ \mathbf{v}^{(z)} \\ \mathbf{u} \end{pmatrix},$$

check the residuals for convergence, and either quit or continue with the iteration. This requires a stopping rule, which could be a variation of what we do now in **VIC-3D®**, or perhaps stopping when the relative error, $\dfrac{\Phi(\mathbf{J}, \boldsymbol{\rho})}{\|(\mathbf{J}, \boldsymbol{\rho})\|}$, is less than some prescribed value.

1.4 Example: Raster Scan at Three Frequencies[2]

Consider Fig. 1.1, which shows a $4 \times 4 \times 4$ mm^3 through-wall anomaly and a T/R-scan system in which the transmitter occupies two positions, $(-14, -14)$mm and $(14, -14)$ mm, and the receiver undergoes a two-dimensional raster scan of 11 points, with equal intervals of 0.75 mm in each direction. The anomaly grid consists of $16 \times 16 \times 4 = 1024$ cells, with the four z-layers numbered 0,1,2,3. The flaw that is to be reconstructed is shown in Fig. 1.2.

The model coil parameters are given in Table 1.1. It is clear that the transmit coil is much larger than the grid cell-size, and is quite remote from the anomalous region. The receive coil scans over the region with dimensions that are comparable to the cell-size, though this is not a requirement.

The transmitter is excited at three frequencies of 10^2, 10^4 and 10^5 Hz, which, with the two-point transmitter scan, produces six 'experiments,' in the language of set-theoretic estimation. Thus, the total impedance data set that is submitted to the bilinear conjugate-gradient inversion algorithm comprises $6 \times 11 \times 11 = 726$ values.

[2] See [102–104] for additional examples.

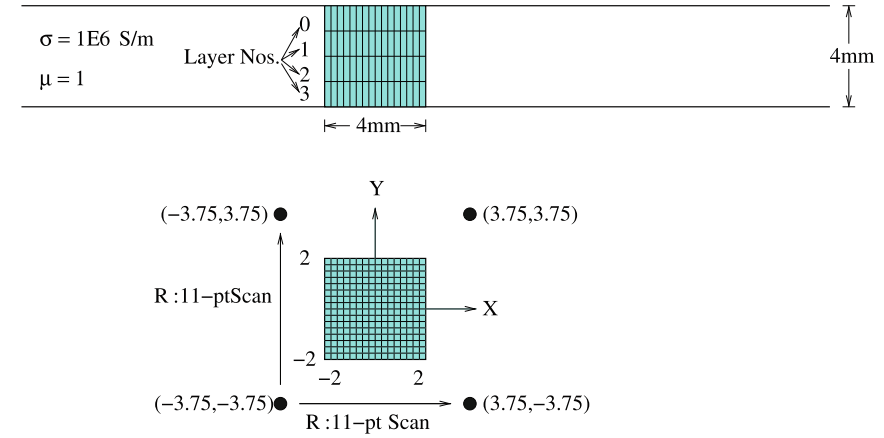

Fig. 1.1 Illustrating the first example. All unlabeled coordinates are in mms. The 11×11-point scan of the receiver probe is shown, as well as the 2-point scan of the transmitter probe. Top: Side view. Bottom: Top view

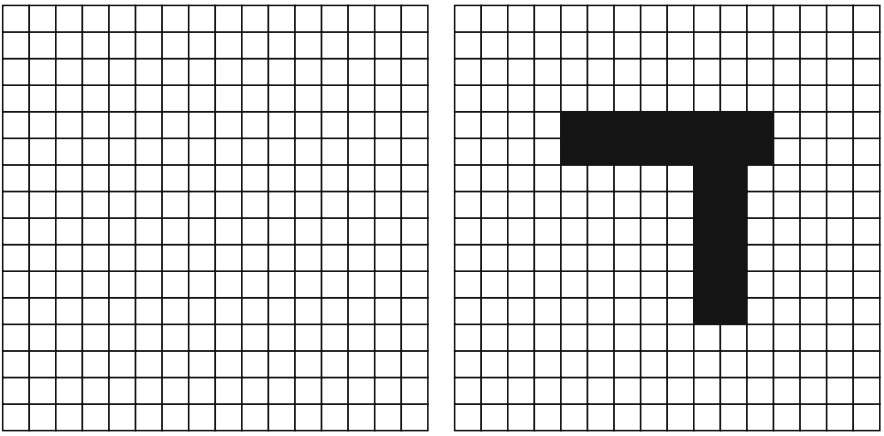

Fig. 1.2 Illustrating the flaw that is to be reconstructed. Left: Layers 0 and 3. Right: Layers 1 and 2. A solid cell has a volume-fraction of 1 (empty), and a blank cell a volume-fraction of 0 (host material)

Table 1.1 Coil parameters for Example 1

Parameter	Transmit	Receive
Inner radius (mm)	15.0	0.25
Outer radius (mm)	25.0	0.50
Height (mm)	3.0	1.0
Turns	100	100

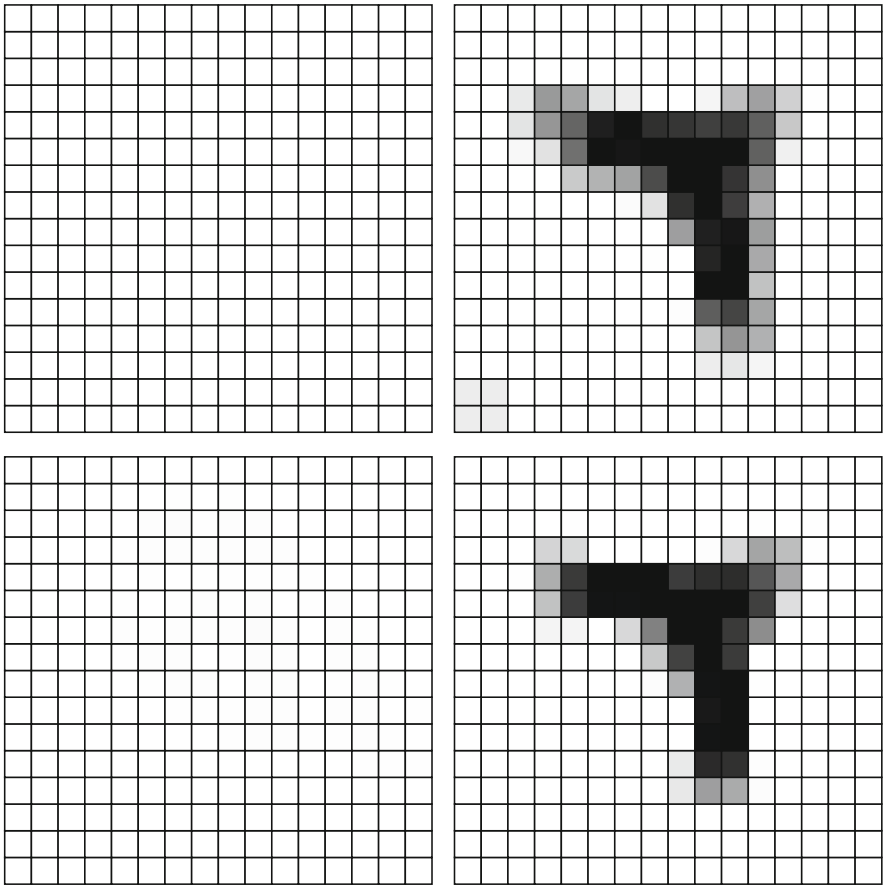

Fig. 1.3 Reconstruction of the flaw in Fig. 1.2. Left: Layers 0 and 3. Right: Layers 1 and 2

The unknowns are 1024 values of conductivity and 3072 values of anomalous current for each experiment, giving a total of 19,456 unknowns to be determined by the inversion algorithm.

In starting the inversion process, we assume that the anomaly is nonexistent, which means that the initial conductivity is that of the host region, and the initial anomalous currents are zero for each experiment. The Polak-Ribière algorithm is used to compute β_{k+1}. The results are shown in Fig. 1.3.

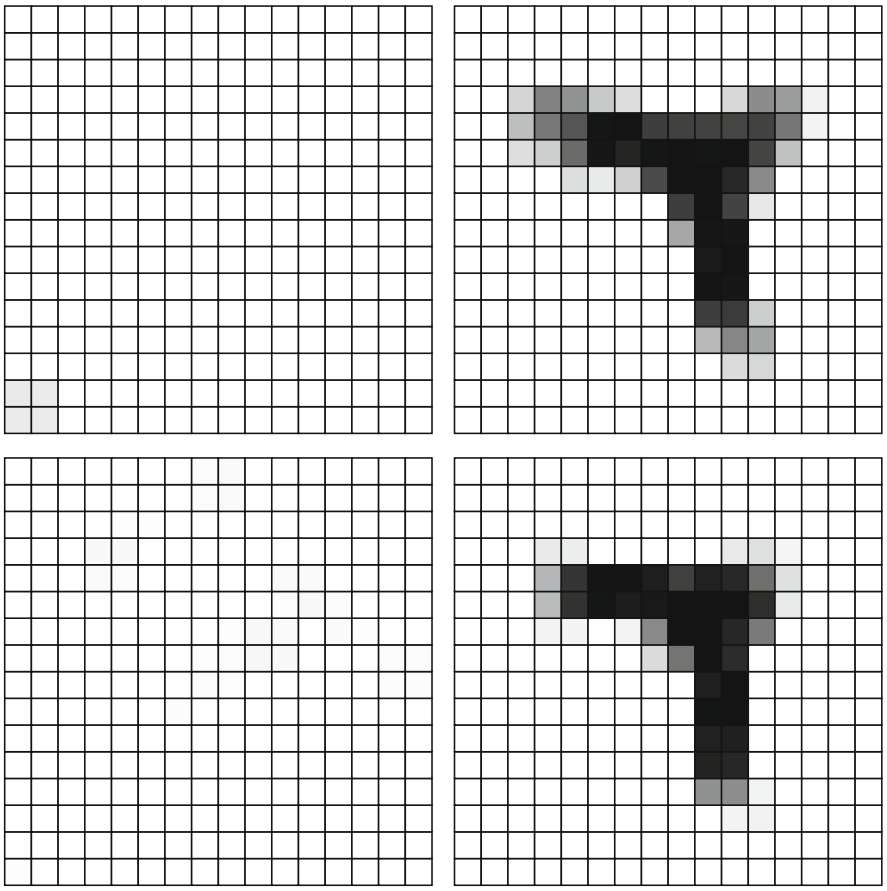

Fig. 1.4 Reconstruction of the flaw in Fig. 1.2 with 5% Gaussian noise added to the input impedance data. Left: Layers 0 and 3. Right: Layers 1 and 2

When we repeat the experiment with the original input data corrupted by the addition of 5% Gaussian noise, we get a reconstruction shown in Fig. 1.4. Clearly, even with a strongly underdetermined system the inversion algorithm performs robustly in the presence of a large noise component.

Though we have not done it here, it is possible to introduce adaptive preconditioning to reduce the effects of noise during the inversion process by introducing a scaling operator, $\mathscr{B}$, that selectively eliminates those components of the reconstructed vector that are less than a certain value. The threshold value is determined early in the iterative process, and requires a certain amount of intuition. It can be made more precise by using statistical decision theory. This is in contrast to post-reconstruction image processing, where weighted averaging filters are used to remove unwanted artifacts after the image has been reconstructed. An example of this approach is given in [14] in the context of a linearized Born approximation to the conjugate-gradient algorithm.