Gauss–Newton and full Newton methods in frequency–space seismic waveform inversion

R. Gerhard Pratt,[1,*] Changsoo Shin[2] and G. J. Hicks[3]

[1] *Department of Geological Sciences, Queen's University in Kingston, Ontario,* K7L 3N6, *Canada*
[2] *School of Civil, Urban and Geosystem Engineering, College of Engineering, Seoul National University, Seoul* 151-742, *South Korea*
[3] *Department of Geology, Imperial College, London* SW7 2BP, *UK*

Accepted 1997 October 21. Received 1997 August 7; in original form 1997 January 21

SUMMARY

By specifying a discrete matrix formulation for the frequency–space modelling problem for linear partial differential equations ('FDM' methods), it is possible to derive a matrix formalism for standard iterative non-linear inverse methods, such as the gradient (steepest descent) method, the Gauss–Newton method and the full Newton method. We obtain expressions for each of these methods directly from the discrete FDM method, and we refer to this approach as frequency-domain inversion (FDI). The FDI methods are based on simple notions of matrix algebra, but are nevertheless very general. The FDI methods only require that the original partial differential equations can be expressed as a discrete boundary-value problem (that is as a matrix problem).

Simple algebraic manipulation of the FDI expressions allows us to compute the gradient of the misfit function using only three forward modelling steps (one to compute the residuals, one to backpropagate the residuals, and a final computation to compute a step length). This result is exactly analogous to earlier backpropagation methods derived using methods of functional analysis for continuous problems. Following from the simplicity of this result, we give FDI expressions for the approximate Hessian matrix used in the Gauss–Newton method, and the full Hessian matrix used in the full Newton method. In a new development, we show that the additional term in the exact Hessian, ignored in the Gauss–Newton method, can be efficiently computed using a backpropagation approach similar to that used to compute the gradient vector.

The additional term in the Hessian predicts the degradation of linearized inversions due to the presence of first-order multiples (such as free-surface multiples in seismic data). Another interpretation is that this term predicts changes in the gradient vector due to second-order non-linear effects. In a numerical test, the Gauss–Newton and full Newton methods prove effective in helping to solve the difficult non-linear problem of extracting a smooth background velocity model from surface seismic-reflection data.

Key words: diffraction, finite-difference methods, inversion, numerical techniques, seismic velocities, wave equation.

INTRODUCTION

During the last decade, seismic waveform inversion has been tackled by applied mathematicians and geophysicists with an increasing degree of success. To our knowledge, the first attempt at waveform inversion was attempted by Lines & Kelly in 1983 (L. Lines, personal communication, 1992). Lines & Kelly computed partial derivatives of the seismogram with respect to the coordinates of a wedge-shaped model, using a numerical difference scheme via a finite-difference forward-modelling technique. At that time, the research was considered too expensive to pursue. An important step was taken by Lailly (1983) and Tarantola (1984), who recognized that the steepest descent direction for the inverse problem (the negative gradient) could be computed without computing the partial derivatives explicitly. They showed (for the acoustic wave equation) how the gradient of the misfit function could be computed by 'backpropagating' the data residuals and correlating the result with forward-propagated wavefields, in a manner very similar to many pre-stack migration algorithms. Numerical results using this approach

[*] Formerly at: Department of Geology, Imperial College, London SW7 2BP, UK.

were given by Kolb, Collino & Lailly (1986) and Gauthier, Virieux & Tarantola (1986). Mora (1987a) extended the technique to elastic problems and provided synthetic numerical examples using complex geological structures. These authors all formulated their methods in the time domain; Pratt & Worthington (1990) and Pratt (1990) applied the same idea to inverse problems in the frequency domain and used an implicit frequency-domain finite-difference technique to provide the forward model.

Several researchers have applied these methods to ray theoretical solutions of the wave equation (Beydoun *et al.* 1989; Lambare *et al.* 1992). The local nature of the ray approximation makes this approach computationally attractive, and ray-based techniques have enjoyed considerable application to real reflection data (Beydoun *et al.* 1989; Beydoun *et al.* 1990). Some progress has also been made outside the ray paradigm in terms of application to reflection data (Crase *et al.* 1990). Some success has been achieved in tomographic (transmission) imaging from cross-borehole data by utilizing gradient methods for waveform inversion in conjunction with finite-difference modelling. Results with real cross-borehole data have been demonstrated by Zhou *et al.* (1995), who utilized a time-domain approach, and Song, Williamson & Pratt (1995) and Pratt *et al.* (1995) who utilized a frequency-domain approach.

For large problems such as those summarized in the previous paragraph, gradient methods seem to be the method of choice, as they do not require the inversion of large matrices. Although gradient methods, such as the conjugate gradient method, can be formulated that have quadratic convergence, it can take a significant amount of iterations before quadratic convergence is established. Shin (1988) applied a Gauss–Newton method to invert seismic data using a frequency-domain finite-element method. The possibility of using the full Newton method for small problems has been investigated by Santosa (1987). When the number of model parameters is large, Newton algorithms involve the computation and inversion of large matrices. This can be circumvented, particularly during the early stages of iterative inversions, by restricting the number of model parameters, in which case the matrices are considerably smaller. As forward-modelling techniques are refined, it is now feasible to consider the more rapidly converging Gauss–Newton method and the full Newton method using workstation computational facilities.

In this paper we present a new formalism for posing the seismic waveform inversion problem, based on a discrete frequency–space forward-modelling procedure that we term frequency-domain modelling (FDM). FDM methods can be posed as either finite-difference problems or finite-element problems. It was shown by Marfurt (1984) that the multiple-source numerical modelling problem was best approached using FDM methods; further developments in FDM methods (Jo, Shin & Suh 1996; Stekl & Pratt 1997) have recently made this approach still more attractive. By specifying the FDM method from the outset, we are able to introduce a discrete matrix formalism for the seismic-waveform inverse problem directly in the frequency–space domain. We term this approach frequency-domain inversion (FDI). The combined FDM/FDI approach allows us to replace the notions of functional analysis of, for example, Tarantola (1987), with the simpler notions of matrix algebra. This does not depend on the utilization of a specific wave equation, or on a specific parametrization. Many partial derivative equations, including the simplest 1-D scalar wave equation, the full 3-D viscoelastic anisotropic wave equation, or even electromagnetic and potential field equations can (in principle) be treated with the same formalism.

The FDM/FDI formalism leads to a matrix algebra demonstration of Lailly's (1983) fast method for computing the gradient of the misfit function for the waveform inversion problem. Leading on from this, we show how the approximate Hessian matrix is structured and used in Gauss–Newton inversion, and we provide a new, fast method for computing the Hessian matrix required for the full Newton inversion technique. In this paper we present two numerical examples. The first is a 2-D imaging example. The gradient method for 2-D seismic imaging and inversion has been demonstrated by many authors; here we too produce such a demonstration, but we extend the demonstration to illustrate the application of the Gauss–Newton method to the same problem. In our second, more significant example we illustrate the utility of the full Newton method (using the exact Hessian matrix), in helping to solve a classic, but difficult, problem in seismic waveform inversion, the problem of determining the low-wavenumber (or 'background') seismic velocity variation from surface reflection data.

Our paper is organized into three sections. In the first section we present the FDM method, the discrete forward-modelling method on which all our results are based. In the second section, we recast the classical techniques of iterative inversion using the FDI formalism, we illustrate these methods with a numerical, 2-D inversion example, and we show how the additional (non-linear) term in the full Hessian can be efficiently computed by backpropagation techniques. In the third and final section of this paper, we present our second, more detailed numerical example illustrating the application of the full Newton method to reflection seismology.

Our paper contains two appendices. Our first appendix gives explicit formulae for the results of this paper when the model is parametrized using alternative linear basis functions, and compares these approaches to the subspace search method of Kennett, Sambridge & Williamson 1988) and the multiscale approach advocated by Bunks *et al.* (1995). Our second appendix compares the use of the additional term in the Hessian to a similar (but not identical) non-linear approach developed by Snieder (1990).

FORWARD MODELLING IN THE SPACE–FREQUENCY DOMAIN

The discretized equations for the acoustic or elastic wave equations using either a finite-difference or a finite-element approach can be written as

$$\mathbf{M}\ddot{\tilde{\mathbf{u}}}(t)+\mathbf{K}\tilde{\mathbf{u}}(t)=\tilde{\mathbf{f}}(t) \tag{1}$$

(see for example Marfurt 1984), where $\tilde{\mathbf{u}}(t)$ is the discretized wavefield (that is the pressure or the displacement) arranged as a column vector, $\mathbf{M}$ is the mass matrix, $\mathbf{K}$ is the stiffness matrix and $\tilde{\mathbf{f}}(t)$ are the source terms, also arranged as a column vector. Eq. (1) includes the boundary conditions implicitly—the actual form of the boundary conditions will alter the various matrix coefficients. If viscous damping is included, eq. (1) becomes

$$\mathbf{M}\ddot{\tilde{\mathbf{u}}}(t)+\mathbf{C}\dot{\tilde{\mathbf{u}}}(t)+\mathbf{K}\tilde{\mathbf{u}}(t)=\tilde{\mathbf{f}}(t)\,, \tag{2}$$

where $\mathbf{C}$ is the damping matrix. Details of the finite-element and finite-difference approaches can be found in many textbooks (Bathe & Wilson 1976; Zienkiewicz & Taylor 1989). The mass, stiffness and damping matrices are computed by forming a discrete representation of the underlying (spatial) partial differential equations and the physical parameters (for example, the seismic velocities, the bulk density and the attenuation parameters). [Our description is in terms of wave propagation modelling, but the modelling method, and the following inverse methods, can be applied to any physical system that can be represented using eq. (2).] It is convenient (but not required) that source–receiver reciprocity holds. The reciprocity property depends on the underlying physics of the problem as well as on the details of the numerical approach. In particular, reciprocity depends on the boundary conditions of the problem.

Eqs (1) or (2) are expressed in the time–space domain. We now choose to implement a frequency-domain solution. This allows distinct computational advantages for multisource problems (Marfurt 1984; Pratt 1990). As we shall show, this leads to a straightforward formalism for the inverse methods employed by Tarantola (1984) and others. Taking the temporal Fourier transform of eq. (2) yields

$$\mathbf{K}\mathbf{u}(\omega)+i\omega\mathbf{C}\mathbf{u}(\omega)-\omega^2\mathbf{M}\mathbf{u}(\omega)=\mathbf{f}(\omega)\,, \tag{3}$$

where

$$\mathbf{u}(\omega)=\int_{-\infty}^{\infty}\tilde{\mathbf{u}}(t)\,\mathrm{e}^{-i\omega t}dt \quad \text{and} \quad \mathbf{f}(\omega)=\int_{-\infty}^{\infty}\tilde{\mathbf{f}}(t)\,\mathrm{e}^{-i\omega t}dt\,. \tag{4}$$

For simplicity we rewrite eq. (3) as

$$\mathbf{S}\,\mathbf{u}=\mathbf{f} \quad \text{or} \quad \mathbf{u}=\mathbf{S}^{-1}\mathbf{f}\,, \tag{5}$$

where the complex 'impedance' matrix, $\mathbf{S}$, is given by $\mathbf{S}=\mathbf{K}-\omega^2\mathbf{M}+i\omega\mathbf{C}$. Frequency-domain modelling is an implicit finite-difference method (Marfurt 1984); the second, explicit, form shown in eq. (5) is only representational, as it is not generally possible (or desirable) to actually invert the very large impedance matrix $\mathbf{S}$. Eq. (5) is often solved using direct matrix factorization methods, such as *LU* decomposition (Press *et al.* 1992; Pratt 1990). If *LU* decomposition is used to solve eq. (5), the matrix factors can be re-used to solve rapidly the forward problem for any new source vector, $\mathbf{f}$ [for typical problem sizes, the additional number of floating point operations required to generate the solution for a new source term is less than 1 per cent of the number of operations required to generate the *LU* matrix factors—see Stekl & Pratt (1997) for details]. This point is especially important in the iterative solution of the inverse problem, in which many forward solutions for real sources and 'virtual' sources will be required at each iteration. It is critical to use ordering schemes that allow maximum advantage to be taken of the sparsity of both $\mathbf{S}$ and its *LU* factorization; nested dissection (Liu & George 1981; Marfurt & Shin 1989) is such a method.

We shall refer to any modelling approach based on eq. (5) as 'frequency-domain modelling', or FDM. The analysis given in this paper will be generally applicable to the inversion of any forward problem, geophysical or otherwise, that can be cast in the form of eq. (5), although the physical interpretation of the results herein will be specific to the seismic problem. We now introduce a specific discretization, depicted in Fig. 1, in which the wavefield is to be computed at $n_x \times n_z = l$ nodal points on a regular grid (the grid is 2-D for illustration purposes, but could be 1-, 2- or 3-D). The model can be thought of as being specified at each of these node points, but such a parametrization is not necessary, nor even particularly desirable. Since finite-difference or finite-element methods generally use a discretization interval that is far smaller than the resolution of most realistic earth models, it is often sensible to define the model in terms of an alternative parameter set, $\mathbf{p}$, an $m\times 1$ column vector, where $m \neq l$ in general (and usually $m \leq l$). In terms of generating forward-modelled data, it is only necessary to be able to compute the $l\times l$ impedance matrix $\mathbf{S}(\mathbf{p})$. As we shall see later, when considering the inverse problem we will also require expressions for $\partial\mathbf{S}/\partial p_i$ for all m model parameters. (In Appendix A we discuss parametrization in more general terms.) We also assume that all model parameters are real valued. Certain physical parameters can be represented as complex numbers (for example velocity and attenuation can be jointly represented by a complex velocity); here we represent the imaginary parts explicitly as additional real-valued parameters.

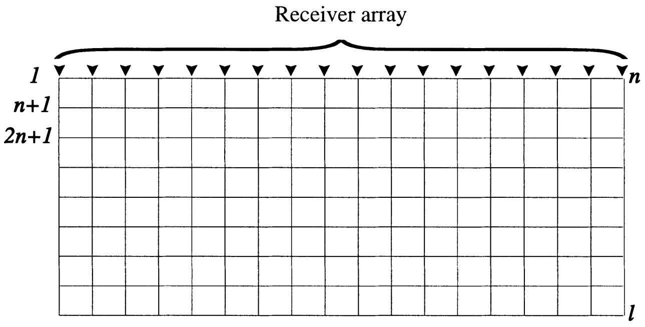


Figure 1. A discrete representation of the forward-modelling problem. The representation is schematic; the assumption of two dimensions is not required, nor is this ordering of the node points necessary. The wavefield (either a scalar or a vector quantity) is sampled at each of the $n_x \times n_z = l$ node points. Receiver data is synthesized at the first n of these node points.

The wavefield vector, $\mathbf{u}$, and the source vector, $\mathbf{f}$, are $l\times 1$ column vectors; the complex impedance matrix, $\mathbf{S}$ is an $l\times l$ matrix. All quantities except the model parameters are complex quantities. Note that, although we will treat eq. (5) as if it describes forward modelling for a single source position, additional source locations can be incorporated simply by increasing the number of elements in $\mathbf{u}$ by l for each additional source; $\mathbf{S}$ and $\mathbf{S}^{-1}$ then have block diagonal structures, in which each diagonal block is an identical submatrix. We could also represent additional frequency components in the same manner, although the diagonal block submatrices of $\mathbf{S}$ would then no longer be identical. The same comment applies to the 2.5-D method of Song & Williamson (1995), in which a new diagonal block would be generated for each wavenumber considered. Ultimately we could end up with a complete time-domain representation in this manner; usually, however, it would seem to be prohibitively expensive, and unnecessary, to actually proceed in this manner.

By examining the solutions to eq. (5) when the components of the source vector, f_i, are replaced by a Kronecker delta, δ_{ij}, it is clear that the columns of $\mathbf{S}^{-1}$ must contain the discrete approximations to the Green's functions. Thus,

$$\mathbf{S}^{-1}=[\mathbf{g}^{(1)}\,\mathbf{g}^{(2)}\,\ldots\,\mathbf{g}^{(l)}]\,, \tag{6}$$

GJI000 21/4/98 09:11:42 3B2 version 5.20
The Charlesworth Group, Huddersfield 01484 517077

where the column vectors $\mathbf{g}^{(j)}$ approximate the discretized Green's function for an impulse at the jth node. If the numerical problem is exactly reciprocal with respect to an interchange of source and receiver elements, then both $\mathbf{S}$ and $\mathbf{S}^{-1}$ are symmetric matrices, and not Hermitian (self-adjoint) matrices. [In implementation $\mathbf{S}$ is often not perfectly symmetric when certain (unphysical) absorbing boundary conditions are implemented (Pratt 1990). This does not cause problems in the iterative inverse solutions we will present in the next section.]

THE INVERSE PROBLEM IN THE SPACE–FREQUENCY DOMAIN

In this section we shall review methods for solving the non-linear seismic-waveform inversion problem using local methods (for the size of problems we are interested in, it is still not feasible to consider a global search for the optimal parameter set). By posing the inverse problem in the frequency domain, and specifically assuming the forward problem can be solved by FDM as in eq. (5), we shall obtain specific algorithms for FDI that are straightforward to understand and implement, and additionally are perfectly general for any forward problem that can be represented by FDM.

We suppose we have n experimental observations, $\mathbf{d}$, at a subset of nodal points corresponding to receiver locations. It is convenient to assume the node points are ordered in such a way that the first $n \le l$ node points are receiver locations (see Fig. 1, but the results we obtain are not specific to such an ordering scheme). The inverse problem is to infer a set of model parameters, $\mathbf{p}$, that would predict the observations using our forward-modelling algorithm. We also assume we have a suitable initial model, $\mathbf{p}^{(0)}$, that is close enough to the global solution to allow successive relinearizations. (In seismic data analysis, such models can often be generated by traveltime analysis.) Given an initial model, we can calculate the response, using the FDM method, $\mathbf{u}=\mathbf{S}^{-1}\mathbf{f}$.

The residual error at the n receiver node points, $\delta\mathbf{d}$, is defined as the difference between the initial model response and the observed data at the receiver locations. Thus

$$\delta d_i = u_i - d_i\,, \qquad i=(1, 2, \dots, n)\,, \tag{7}$$

where the subscripted quantities are the individual components of $\delta\mathbf{d}$, $\mathbf{u}$ and $\mathbf{d}$, and the subscript i represents the receiver number.

As is common in many inverse problems, we seek to minimize the l_2 norm of the data residuals. Thus we seek to minimize the 'misfit' function (or 'objective' function)

$$E(\mathbf{p}) = \frac{1}{2}\,\delta\mathbf{d}^{\mathrm{t}}\delta\mathbf{d}^{*}\,, \tag{8}$$

where the superscript t represents the ordinary matrix transpose and the superscript $*$ represents the complex conjugate, introduced to ensure the misfit function is a true (real-valued) norm for complex-valued data.

The simple misfit function in eq. (8) neglects any incorporation of *a priori* statistical information on the data or on the model in the form of covariance matrices. We have, in effect, assumed an identity data covariance matrix and assumed infinite *a priori* model variances. More realistic covariance matrices may of course be incorporated, but we deliberately omit them as they tend to obscure the simplicity of the algorithms. In the solution of real inverse problems, these terms provide numerical stability, allow meaningful prior statistical information to be incorporated, and allow meaningful *a posteriori* statistical information to be extracted (see e.g. Tarantola 1987).

The gradient method of inversion

The gradient method is a recipe for reducing the l_2 norm (8) by iteratively updating the parameter vector according to

$$\mathbf{p}^{(k+1)} = \mathbf{p}^{(k)} - \alpha^{(k)}\nabla_p E^{(k)}\,, \tag{9}$$

where k is an iteration number and α is a step length (a positive scalar) chosen to minimize the l_2 norm in the direction given by the gradient of $E(\mathbf{p})$. The role of the step length can also be thought of as converting the units of the gradient vector to model dimensions.

The gradient of the misfit function represents the direction in which the misfit function is increasing most rapidly. The misfit function can always be reduced by pursuing the negative of this direction. The iteration in eq. (9) is performed until some suitable stopping criteria is reached. The convergence rate of the gradient method is generally quite slow; convergence can be improved by adopting a conjugate gradient approach (see for example Mora 1987a), which does not require any significant additional computations.

We may evaluate the gradient direction by taking partial derivatives of eq. (8) with respect to the model parameters, $\mathbf{p}$, yielding

$$\nabla_p E = \frac{\partial E}{\partial \mathbf{p}} = \mathscr{Re}\{\mathbf{J}^{\mathrm{t}}\delta\mathbf{d}^{*}\}\,, \tag{10}$$

where $\mathbf{J}^{\mathrm{t}}$ is the transpose of the $n \times m$ Frechét derivative matrix, the (complex-valued) elements of which are given by

$$J_{ij} = \frac{\partial u_i}{\partial p_j}\,, \qquad i=(1, 2, \dots, n)\,; \;\; j=(1, 2, \dots, m)\,. \tag{11}$$

In eq. (10) we assume there are m model parameters, so that $\mathbf{p}$ and $\nabla_p E$ are column vectors of length m. The mathematics demand that the real part of the complex-valued vector $\mathbf{J}^{\mathrm{t}}\delta\mathbf{d}^{*}$ be taken—this ensures the gradient of the real-valued misfit function with respect to real-valued parameters remains real. (It is also possible to obtain a real-valued result automatically if each frequency component of the data is accompanied by the corresponding negative frequency component. This makes use of the conjugate symmetry of the Fourier transform of real-valued, time-domain data.)

The computation of the step length, α, required in eq. (9) is straightforward. For linear forward problems the step length is given by the formula

$$\alpha^{(k)} = \frac{|\nabla_p E|^2}{|\mathbf{J}\nabla_p E|^2}\,, \tag{12}$$

where $|\,|$ represents the Euclidean length of the vectors. For non-linear forward problems (such as the seismic problem), the step length must be found using line-search techniques along the direction opposite to the gradient.

We now wish to link explicitly the computation of the gradient vector for the FDI problem to the forward FDM problem given in eq. (5). To do this we first augment the $m \times n$ matrix $\mathbf{J}$ with the additional terms required to define partial

GJI000 21/4/98 09:12:16 3B2 version 5.20
The Charlesworth Group, Huddersfield 01484 517077

derivatives at *all* node points, not just at the receiver locations, to obtain a new $m \times l$ matrix $\hat{\mathbf{J}}$. We may then write a new equation, equivalent to eq. (10):

$$\nabla_p E = \mathscr{Re}\{\hat{\mathbf{J}}^{\mathrm{t}}\,\delta\hat{\mathbf{d}}^*\}\,, \tag{13}$$

where $\delta\hat{\mathbf{d}}$ is the data residual vector, of length n, augmented with $(l-n)$ zero values to produce a new vector of length l. Explicitly, eq. (13) represents

$$\begin{bmatrix} \dfrac{\partial E}{\partial p_1} \\ \dfrac{\partial E}{\partial p_2} \\ \vdots \\ \dfrac{\partial E}{\partial p_m} \end{bmatrix} = \mathscr{Re}\left\{ \begin{bmatrix} \dfrac{\partial u_1}{\partial p_1} & \cdots & \dfrac{\partial u_n}{\partial p_1} & \dfrac{\partial u_{n+1}}{\partial p_1} & \cdots & \dfrac{\partial u_l}{\partial p_1} \\ \dfrac{\partial u_1}{\partial p_2} & \cdots & \dfrac{\partial u_n}{\partial p_2} & \dfrac{\partial u_{n+1}}{\partial p_2} & \cdots & \dfrac{\partial u_l}{\partial p_2} \\ \vdots & \ddots & \vdots & \vdots & \ddots & \vdots \\ \dfrac{\partial u_1}{\partial p_m} & \cdots & \dfrac{\partial u_n}{\partial p_m} & \dfrac{\partial u_{n+1}}{\partial p_m} & \cdots & \dfrac{\partial u_l}{\partial p_m} \end{bmatrix} \begin{bmatrix} \delta d_1^* \\ \vdots \\ \delta d_n^* \\ 0 \\ \vdots \\ 0 \end{bmatrix} \right\}$$

$$= \mathscr{Re}\left\{ \begin{bmatrix} \dfrac{\partial \mathbf{u}^{\mathrm{t}}}{\partial p_1} \\ \dfrac{\partial \mathbf{u}^{\mathrm{t}}}{\partial p_2} \\ \vdots \\ \dfrac{\partial \mathbf{u}^{\mathrm{t}}}{\partial p_m} \end{bmatrix} \begin{bmatrix} \delta d_1^* \\ \vdots \\ \delta d_n^* \\ 0 \\ \vdots \\ 0 \end{bmatrix} \right\} \tag{14}$$

(recall, n is the number of receiver points, l is the number of node points, and we have ordered the node points in such a manner that the first n node points correspond to the n receiver points).

An expression for any of the partial derivatives in eq. (14) in terms of the forward-modelling matrix eq. (5) can now be obtained by taking the partial derivative of both sides of eq. (5) with respect to the ith parameter p_i:

$$\mathbf{S}\frac{\partial \mathbf{u}}{\partial p_i} = -\frac{\partial \mathbf{S}}{\partial p_i}\mathbf{u} \quad \text{or} \quad \frac{\partial \mathbf{u}}{\partial p_i} = \mathbf{S}^{-1}\mathbf{f}^{(i)}\,, \tag{15}$$

where we have introduced the ith 'virtual' source term

$$\mathbf{f}^{(i)} = -\frac{\partial \mathbf{S}}{\partial p_i}\mathbf{u}\,, \tag{16}$$

itself an $l \times 1$ vector. This development is the same as that used by Oristaglio & Worthington (1980) and Rodi (1976) to develop partial derivatives for the electromagnetic problem. By analogy with eq. (5), the partial derivatives in eq. (15) are the solution to a new forward-modelling problem, now driven by the virtual sources at the location of the ith parameter. The virtual source represents the interaction (or scattering) of the predicted (or background) wavefield, $\mathbf{u}$, with the parameter p_i. We will therefore refer to $\partial\mathbf{u}/\partial p_i$ as the 'partial derivative wavefield from the ith node' (p_i does not necessarily represent a nodal parameter, but it is convenient to think of it this way). As shown in eq. (14), each row of $\hat{\mathbf{J}}^{\mathrm{t}}$ (i.e. each column of $\hat{\mathbf{J}}$) contains a partial derivative wavefield from a single physical parameter; there are m such columns. If the inversion parameters are defined as the (discrete) values of a single physical parameter at the node points (the 'point collocation' scheme), there will be $m=l$ columns and $\hat{\mathbf{J}}$ is a square matrix. Furthermore, the virtual sources will then be highly local, approximating point sources, with a seismic moment that depends on the details of the parametrization. In this case the computation is directly comparable with the Born method for computing wavefield perturbations due to first-order scattering.

The computation of the partial derivatives of the impedance matrix, $\partial\mathbf{S}/\partial p_i$, required in the computation of the virtual sources (15), will depend on the specific details of the finite approximation method used (that is on the type of wave equation, the parametrization, the order of the differencing scheme, the basis functions, etc.). However, in all cases these partial derivatives are relatively trivial to compute [see Shin (1988) and our Appendix A].

Efficient calculation of the gradient direction

Since we could generate an equation similar to eq. (15) for any choice of i, we can represent all the partial derivatives simultaneously by the matrix equation

$$\hat{\mathbf{J}} = \begin{bmatrix} \dfrac{\partial \mathbf{u}}{\partial p_1} & \dfrac{\partial \mathbf{u}}{\partial p_2} & \cdots & \dfrac{\partial \mathbf{u}}{\partial p_m} \end{bmatrix} = \mathbf{S}^{-1}\,[\mathbf{f}^{(1)} \quad \mathbf{f}^{(2)} \quad \cdots \quad \mathbf{f}^{(m)}]$$

$$\text{or} \quad \hat{\mathbf{J}} = \mathbf{S}^{-1}\mathbf{F}\,, \tag{17}$$

where $\mathbf{F}$ is an $l \times m$ matrix, the columns of which are the virtual source terms for each of the m physical parameters. Eq. (17) gives an explicit formula for the Frechét derivative matrix $\mathbf{J}$ (being the first $n \le l$ rows of $\hat{\mathbf{J}}$). Computation of the elements of $\mathbf{J}$ using eq. (17) would require m forward-propagation problems to be solved, in addition to the one required to compute the virtual sources using eq. (16). However, in order to compute the gradient using eqs (10) or (13) it is not necessary to compute the elements of $\mathbf{J}$ explicitly. Substituting (17) into (13) we obtain

$$\nabla_p E = \mathscr{Re}\{\hat{\mathbf{J}}^{\mathrm{t}}\delta\hat{\mathbf{d}}^*\} = \mathscr{Re}\{\mathbf{F}^{\mathrm{t}}[\mathbf{S}^{-1}]^{\mathrm{t}}\delta\hat{\mathbf{d}}^*\} \tag{18}$$

or

$$\nabla_p E = \mathscr{Re}\{\mathbf{F}^{\mathrm{t}}\mathbf{v}\}\,, \tag{19}$$

where

$$\mathbf{v} = [\mathbf{S}^{-1}]^{\mathrm{t}}\delta\hat{\mathbf{d}}^* \tag{20}$$

is the 'backpropagated wavefield'. If the inverse impedance matrix $\mathbf{S}^{-1}$ is symmetric, as we expect for reciprocal problems (see eq. 1), then $[\mathbf{S}^{-1}]^{\mathrm{t}} = \mathbf{S}^{-1}$ and

$$\mathbf{v} = \mathbf{S}^{-1}\delta\hat{\mathbf{d}}^*\,. \tag{21}$$

(If the inverse impedance matrix is not symmetric, the transpose of the LU decomposition, $U^{\mathrm{t}}L^{\mathrm{t}}$ may be used to generate $\mathbf{v}$ by the usual forward-reduction and back-substitution processes.) In either eq. (20) or eq. (21), computing the new, backpropagated wavefield, $\mathbf{v}$ requires only one additional forward problem to be solved. Since the computation of the step length (in the linear approximation) also requires the solution of a forward problem, this brings the total number of forward solutions required to three. The forward and backpropagation problems are solved in the same model, and hence the stored LU factors of $\mathbf{S}$ can still be used to compute these forward solutions rapidly.

GJI000 21/4/98 09:12:40 3B2 version 5.20
The Charlesworth Group, Huddersfield 01484 517077

In the development above the original impedance matrix (and not its adjoint) is used to compute the backpropagated field, and the conjugate of the data residuals is used in eqs (20) and (21). An alternative development can be obtained by recognizing that

$$\nabla E_p = \mathscr{Re}\{\hat{\mathbf{J}}^{t}\delta\hat{\mathbf{d}}^{*}\} = \mathscr{Re}\{\hat{\mathbf{J}}^{t^{*}}\delta\hat{\mathbf{d}}\}\,. \tag{22}$$

This leads to an alternative definition of the backpropagated field,

$$\mathbf{w} = \mathbf{v}^{*} = [\mathbf{S}^{-1}]^{t^{*}}\delta\hat{\mathbf{d}}, \tag{23}$$

or, for reciprocal problems,

$$\mathbf{w} = \mathbf{v}^{*} = [\mathbf{S}^{-1}]^{*}\delta\hat{\mathbf{d}}. \tag{24}$$

In eq. (23) or (24) we do not conjugate (time reverse) the data residuals, but we use the adjoint of the impedance matrix to compute the backpropagated field. We still use eq. (19) to compute the gradient.

Either form of backpropagation can used, eq. (20) or (23). The former is easier, as the original impedance matrix may be used, rather than its adjoint. The latter form [the backpropagation method given in eq. (23) or (24)] exactly parallels the backpropagation method derived by Lailly (1983) using continuous methods of functional analysis. The FDI formalism replaces the notions of functional analysis with the simpler notions of matrix algebra; this will prove useful in the next section when we consider the Newton methods of inversion. Ultimately, computer programs for gradient methods can be developed from either formalism without any fundamental differences. Tarantola (1986), Mora (1987a), Kolb *et al.* (1986), Pratt (1990), Pratt & Worthington (1990) and Chavent & Jacewitz (1995) have all exploited this technique in computing the gradient of the misfit function for non-linear waveform inversion.

The backpropagation technique given by Lailly (1983) makes use of the adjoint state method, which has been used extensively in optimal control theory since at least 1956 (Goodman & Lance 1956; Jurovics & McIntyre 1962; Lee & Marcus 1967). The matrix derivation of the backpropagation technique given in eqs (18) to (21) was derived by one of us (Shin 1988) several years ago, but the result has not received much attention in the literature. An identical result has since been independently obtained by Griewank (1989) in the field of photon diffusion.

We may summarize the backpropagation method of eq. (21) as follows. The gradient is computed in two steps: (1) The ‘backpropagated’ wavefield, **v**, is computed by solving a forward problem with the original impedance matrix, and with the source terms replaced by the conjugate (time-reversed) residuals. (2) The backpropagated field is multiplied by the virtual sources generated by the original predicted wavefield, **u**. Finally, we take the real part of the result. These operations can be compared with the equivalent time-domain operations presented by Mora (1987a): Mora computed the gradient using a zero-lag cross-correlation of each virtual source time series with the backpropagated wavefield. The operation represented by eq. (18) is in fact a convolution of the backpropagated field with the virtual sources—the difference is due to the time reversal of the data residuals implied in eq. (21), and due to the fact that we use the original wave equation to propagate these. Mora (1987a) formed a different backpropagated wavefield by using the unreversed data residuals and the time-reversed (adjoint) wave equation.

It is informative to use eqs (16) and (19) to express the *i*th component of the gradient vector as

$$(\nabla_p E)_i = \mathscr{Re}\{\mathbf{f}^{(i)^{t}}\mathbf{v}\} = \mathscr{Re}\left\{\mathbf{u}^{t}\left[\frac{\partial\mathbf{S}^{t}}{\partial p_i}\right]\mathbf{v}\right\}, \tag{25}$$

from which it is evident that where $\partial\mathbf{S}^{t}/\partial p_i$ consists of only highly local non-zero values at, or near, the diagonal of the *i*th row (as it will for the point collocation scheme), the gradient can be computed by a scaled multiplication (convolution in the time domain) of forward and backpropagated wavefields. This is the description usually given for the computation of the gradient vector, and it is clearly closely related to reverse time migration algorithms, and to Claerbout’s (1976) U/D imaging principle. The analogy with pre-stack reverse time migration has been made by many authors, and is further illustrated in the next section of this paper.

A physical interpretation of the gradient

In this section we examine the role the partial derivative wavefields play in the gradient method, and we illustrate the effectiveness (or otherwise) of the gradient vector as an estimate of the parameter updates by using a point diffractor model. Although the illustration we provide here is rather classical, and does not rely on the FDI formalism, we provide this illustration in order to introduce the physical concepts of virtual sources and partial derivative wavefields that are crucial in order to understand the FDI implementations of the Gauss–Newton method and the full Newton method to follow.

In order to illustrate these concepts, we use the small point diffractor model depicted in Fig. 2. A single source is located centrally just below the surface, and 21 receivers are distributed just below the surface from one end of the model to the other. Synthetic frequency-domain data were computed using the FDM method, followed by Fourier synthesis to create time-domain data, using 16 frequencies between 0 and 25 Hz. We computed a residual wavefield by forward modelling twice (with and without the anomaly), and we computed a number of representative partial derivative wavefields with respect to (point) perturbations of the velocities at the nodes indicated in Fig. 3. The partial derivative wavefields were computed using eqs (15) and (16). Owing to the weak nature of the anomaly, we expect the partial derivative wavefields to be a good representation of the wavefield perturbations that would be observed for a true perturbation of the velocity.

Since the behaviour of seismic data is easier to visualize in the time domain than in the frequency domain, we will illustrate the operation of the gradient method using the resultant time-domain wavefields (i.e. seismograms) depicted in Figs 3(a) to (e). These figures show the time-domain surface expressions of five partial derivative seismograms. The partial differentiation was computed with respect to the velocity parameters at each of the five scattering locations shown in Fig. 3. From the preceding analysis (see eq. 17), we recognize that these partial derivative seismograms represent forward propagation using virtual sources at the scattering locations. The virtual sources are excited, or activated, by the background wavefield. In the space–time domain, the virtual sources have numerical support only *at* the scattering locations and *when* the background wavefield arrives at the scattering location. In the

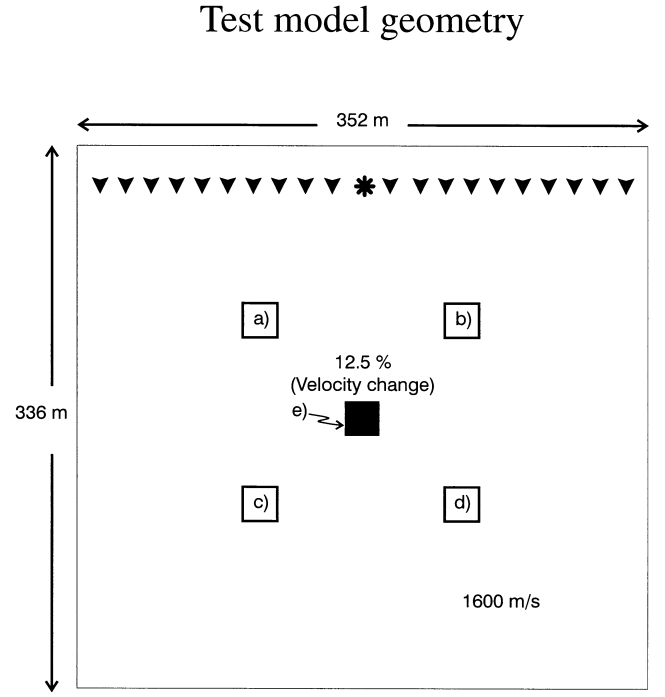


Figure 2. Test model containing a single anomaly embedded in a 1600 m s^{-1} homogeneous velocity model. The velocity of the anomaly is 1800 m s^{-1}. The single-source, multiple-receiver reflection geometry used to test the gradient and Gauss–Newton methods is shown in the figure. The numerical experiment used 16 frequencies between 1 and 25 Hz. The five node points (a), (b), (c), (d) and (e) used to compute the partial derivative wavefields in Fig. 3 are also identified.

absence of other events in the background wavefield, the virtual sources will return to zero after the background wavefield has passed through the cell. Because of this, the partial derivative seismograms display characteristic hyperbolic point diffraction responses. (In the frequency domain, the phase of the complex Fourier component associated with each trace in Fig. 3 changes across the receiver array hyperbolically.)

The final seismogram in Fig. 3 depicts the actual data residuals for the point diffractor seismogram, using an initial guess of a homogeneous model. The classical hyperbolic response of this point diffractor is clear. The negative polarity of this seismogram is due to our definition of the data residuals as $u_i - d_i$ in eq. (7).

Using this insight into the appearance of the partial derivative seismograms, we now return to the computation of the gradient vector (19). This consists of forming the $m \times 1$ vector $\mathcal{Re}\{\mathbf{J}^t \delta\mathbf{d}^*\}$, where the elements of $\mathbf{J}$ are the partial derivative wavefields. As can be seen from eq. (14), in the frequency domain each element of the vector $\mathbf{J}^t\delta\mathbf{d}^*$ contains a partial derivative wavefield, sampled at the receiver points, multiplied with the conjugated (time-reversed) residual wavefield, $\delta\mathbf{d}$, following which the real part is extracted. The corresponding operation in the time domain is zero-lag correlation (of two real-valued time-series). The residual seismogram (the time-domain expression of $\delta\mathbf{d}$) will have a maximum correlation with the partial derivative seismogram emanating from the central node, and lesser correlations with other seismograms. This process of correlation with the point diffractor responses is closely related to the early migration work of Hagedoorn (1954) in his discussion of maximum convexity surfaces. Thus, Hagedoorn's migration of pre-stack data is equivalent to the first iteration of a gradient method.

For band-limited data, as the partial derivative wavefield is progressively time delayed and shifted in space with respect to the residual wavefield, the wavefields go in and out of phase, leading to decaying oscillations in their correlation. Thus, band-limited data cause 'leakage' in the correlations, resulting in a gradient that is not strictly localized at the true scattering location. The wider the aperture, the less leakage will occur in the correlations away from the central node.

For our test example, the resulting gradient vector using all 16 frequency components (after scaling using the correct step length) is shown in Fig. 4. From the previous discussion, the maximum element in the $\mathcal{Re}\{\mathbf{J}^t\delta\mathbf{d}^*\}$ vector should occur at the location of the point scatterer. We also expect oscillatory side lobe features as the partial derivative wavefields progressively correlate, then anti-correlate with the residual wavefield. Both these features are indeed observed in Fig. 4. The gradient shown in Fig. 4 can be seen to represent an image of the original diffracting point, with the location of the point diffractor receiving the largest contribution (except at the source point where the source and receiver singularities combine). However, the image is not a particularly good one. This is partly because the image is equivalent to pre-stack migration of data from only a single source, but also because an update based simply on a single computation of the gradient, without any preconditioning, is fundamentally limited as a migration operator.

Spurious correlations between the partial derivative wavefields and the residuals is not the only source of misplaced parameter perturbations in the gradient vector. Events that arise from multiple scattering, although not present in our test example, would also spuriously correlate with the (single scattered) partial derivative wavefields, further corrupting the inversion results. In the next section we investigate to what extent Newton methods can improve this first iteration result by accounting for finite frequency effects and for irregular coverage of the target.

Newton methods of inversion

Newton methods are derived by considering an expansion of the misfit function in eq. (8) as a Taylor series and retaining terms up to quadratic order (Bertsekas 1982; Tarantola 1987):

$$E(\mathbf{p}+\delta\mathbf{p}) = E(\mathbf{p}) + \delta\mathbf{p}^t\nabla_p E(\mathbf{p}) + \frac{1}{2}\,\delta\mathbf{p}^t\ \mathbf{H}\delta\mathbf{p} + O(|\delta\mathbf{p}|^3)\,. \qquad (26)$$

$\mathbf{H}$ is the $m \times m$ Hessian second-derivative matrix, the elements of which are given by

$$H_{ij} = \frac{\partial^2 E(\mathbf{p})}{\partial p_i \partial p_j}\,, \quad i=(1, 2, \ldots, m)\,; \quad j=(1, 2, \ldots, m)\,. \qquad (27)$$

We seek a vector $\delta\mathbf{p}$ that will locate the minimum within the quadratic approximation. For linear forward problems this approach will converge in one iteration. Minimizing with respect to all components of $\delta\mathbf{p}$, we find the solution is characterized by

$$\mathbf{H}\delta\mathbf{p} = -\nabla_p E \quad \text{or} \quad \delta\mathbf{p} = -\mathbf{H}^{-1}\,\nabla_p E\,. \qquad (28)$$

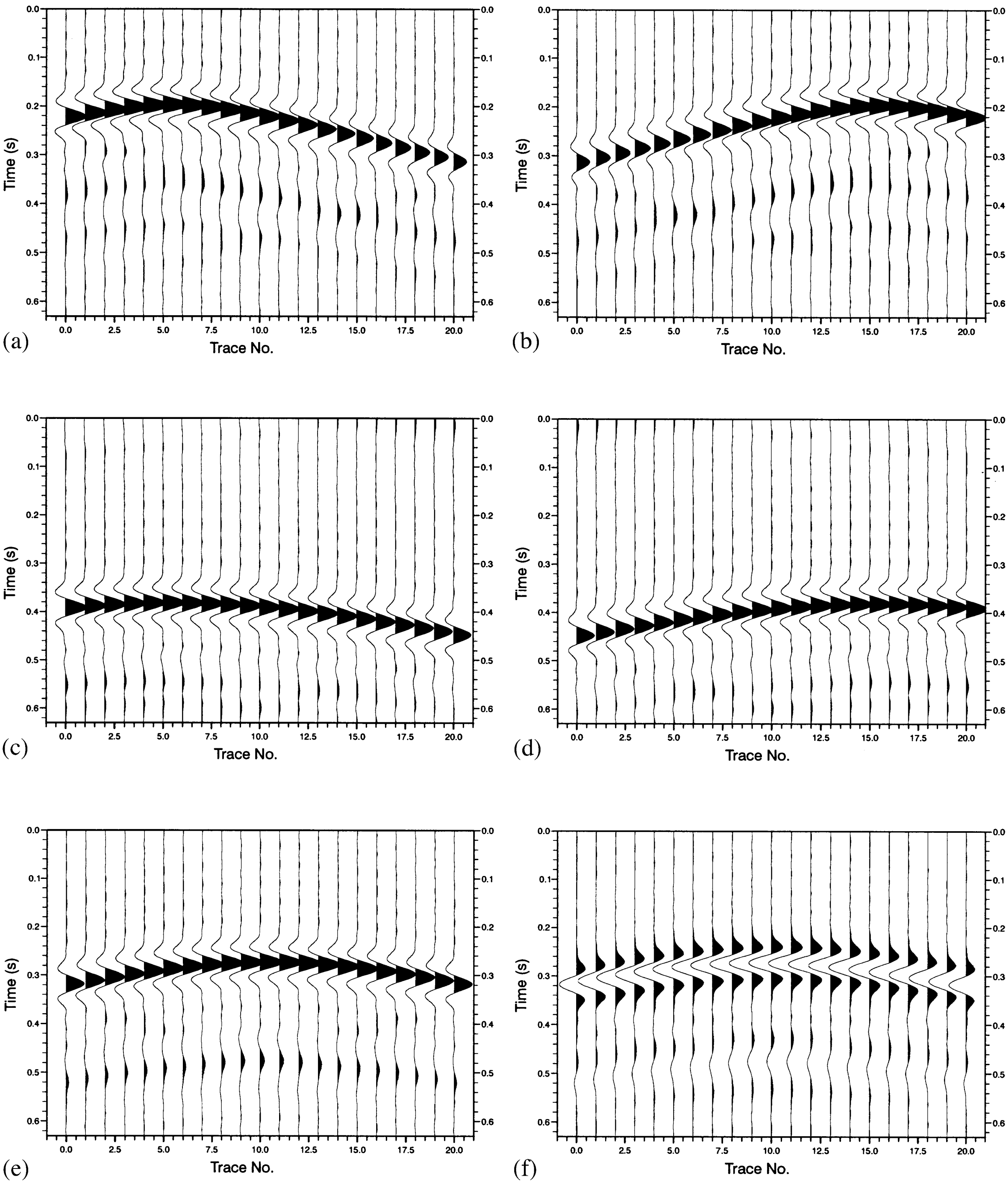


Figure 3. Time-domain wavefields related to the model depicted in Fig. 2. These were computed by frequency-domain modelling (FDM) and Fourier synthesis. 16 frequencies between 0 and 25 Hz were used. (a)–(e) The 'partial derivative seismograms', computed with respect to perturbations at the five corresponding node points in Fig. 2. (f) The residual seismogram, that is the difference between the observed data in the true model and the predicted data in the homogeneous model. The residual wavefield has the opposite polarity from the partial derivative wavefields; this results from the definition of the data residuals as $u_i - d_i$ in eq. (7).

In the gradient method of eq. (9) it is assumed that $\delta\mathbf{p}$ can be estimated by a scalar multiplied by the gradient $\nabla_p E$. Eq. (28) shows that a better estimate is the gradient, preconditioned, or filtered by the inverse Hessian. For this reason, in spite of the quasi-linearity of the problem examined using the gradient method in Fig. 4, the image is not optimal. Naturally, if gradient methods are applied iteratively, the inverse Hessian is effectively reconstructed during the iterative process. However, preconditioning using the inverse Hessian, or an approximation of the inverse Hessian, can yield improved convergence rates in iterative solutions.

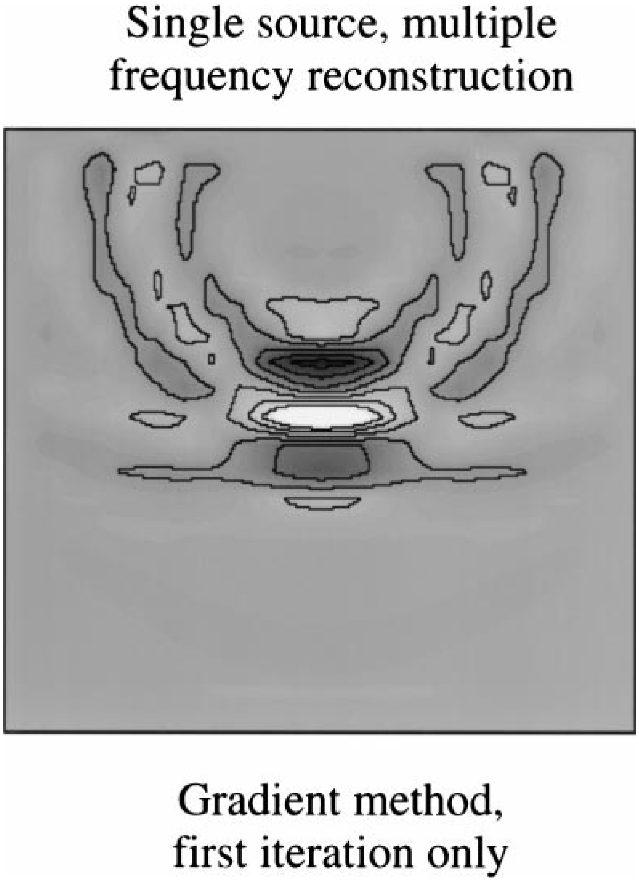


Figure 4. The gradient vector (scaled using the correct step length) for the first iteration of an inversion of the data from the single anomaly model shown in Fig. 2. The gradient was computed using 16 frequencies between 0 and 25 Hz from a single source and 21 receivers (as shown in Fig. 2). This image can be considered to be roughly equivalent to pre-stack depth migration. Some of the artefacts that appear may be removed by iteration or by preconditioning using the Gauss–Newton method (see Fig. 6).

Eq. (28) suggests the Newton method for iterative solution

$$\mathbf{p}^{(k+1)} = \mathbf{p}^{(k)} - \mathbf{H}^{-1}\nabla_p E\,. \tag{29}$$

Newton methods are normally avoided in large inverse problems due to the large cost of computing and inverting the $m\times m$ Hessian matrix. Nevertheless it is instructive to pursue the analysis, since eq. (29) *can* be of use in problems that are deliberately underparametrized so that m is small (Xu, McMechan & Sun 1995). Also, as computers evolve, and as increasingly accurate FDM techniques are developed, the seismic inverse problem, when cast as an FDI problem, is no longer the insurmountably large problem it used to be. As we shall see, it will also be possible to recast eq. (29) in such a manner as to require the inversion of only an $n\times n$ matrix, instead of an $m\times m$ matrix given by eq. (27). This latter approach has relevance for problems in which the number of data is considerably smaller than the number of parameters [Delprat-Jannuad & Lailly (1992) have advocated using a large number of parameters in conjunction with appropriate regularization techniques in order to avoid discretization errors].

We can explicitly express each element of the Hessian matrix in eq. (27) by differentiating eq. (8):

$$H_{ij} = \frac{\partial^2 E}{\partial p_i \partial p_j}$$

$$= \mathcal{R}e\left\{ \begin{bmatrix} \dfrac{\partial u_1}{\partial p_i} & \dfrac{\partial u_2}{\partial p_i} & \cdots & \dfrac{\partial u_n}{\partial p_i} \end{bmatrix} \begin{bmatrix} \dfrac{\partial u_1^*}{\partial p_j} \\ \dfrac{\partial u_2^*}{\partial p_j} \\ \vdots \\ \dfrac{\partial u_n^*}{\partial p_j} \end{bmatrix} + \begin{bmatrix} \dfrac{\partial^2 u_1}{\partial p_i \partial p_j} & \dfrac{\partial^2 u_2}{\partial p_i \partial p_j} & \cdots & \dfrac{\partial^2 u_n}{\partial p_i \partial p_j} \end{bmatrix} \begin{bmatrix} \delta d_1^* \\ \delta d_2^* \\ \vdots \\ \delta d_n^* \end{bmatrix} \right\}, \tag{30}$$

from which, using eq. (11), one obtains

$$\mathbf{H} = \mathcal{R}e\{\mathbf{J}^{\mathrm{t}}\mathbf{J}^*\} + \mathcal{R}e\left\{\left[\left(\frac{\partial}{\partial p_1}\mathbf{J}^{\mathrm{t}}\right)\delta\mathbf{d}^* \quad \left(\frac{\partial}{\partial p_2}\mathbf{J}^{\mathrm{t}}\right)\delta\mathbf{d}^* \quad \cdots \quad \left(\frac{\partial}{\partial p_m}\mathbf{J}^{\mathrm{t}}\right)\delta\mathbf{d}^*\right]\right\} \tag{31}$$

or

$$\mathbf{H} = \mathbf{H}_a + \mathbf{R}\,, \tag{32}$$

where

$$\mathbf{H}_a = \mathcal{R}e\{\mathbf{J}^{\mathrm{t}}\mathbf{J}^*\} \tag{33}$$

is the ‘approximate Hessian’, and

$$\mathbf{R} = \mathcal{R}e\left\{\left(\frac{\partial}{\partial \mathbf{p}^{\mathrm{t}}}\mathbf{J}^{\mathrm{t}}\right)(\delta\mathbf{d}^* \quad \delta\mathbf{d}^* \quad \ldots \quad \delta\mathbf{d}^*)\right\}. \tag{34}$$

There are m column vectors in the last term in eq. (34), each equal to $\delta\mathbf{d}^*$ [note that the first term in eq. (34) implies a specific meaning for the partial differentiation of a matrix with respect to a row vector].

The Gauss–Newton method of inversion

Of the two terms in the expression for the Hessian matrix (32), the first term, $\mathbf{H}_a$, is straightforward to compute, whereas the second term, $\mathbf{R}$, is often difficult to compute. Moreover, quoting from Tarantola (1987): ‘The last term is small if: i) the residuals are small, or ii) the forward equation is quasi linear. As in Newton methods we never need to know the Hessian with great accuracy, and as the last term in [eq. (32)] is, in general, difficult to handle, it is generally dropped off’. If we neglect the second term (which clearly is only important if changes in the parameters cause a change in the partial derivative wavefields), we obtain the Gauss–Newton formula

$$\mathbf{p}^{(k+1)} = \mathbf{p}^{(k)} - \mathbf{H}_a^{-1}\nabla_p E \quad \text{or} \qquad \delta\mathbf{p} = -\mathbf{H}_a^{-1}\nabla_p E \tag{35}$$

(eq. 35 can also be obtained by linearizing the forward problem and applying the least squares 'normal equation'). The full Newton method, to be discussed in the next section, differs only from the Gauss–Newton method by the inclusion of the second term in the Hessian.

In eq. (35) we have assumed that the Hessian matrix has full column rank, and is thus invertible. Generally this will not be the case. The Hessian matrix is often either ill-conditioned or actually singular. To improve and stabilize the Gauss–Newton method for non-linear problems, some form of regularization will be required. Perhaps the simplest form of regularization is to apply a damping term to the Hessian before inverting it:

$$\delta\mathbf{p} = -(\mathbf{H}_a + \lambda\mathbf{I})^{-1}\,\nabla_p E \tag{36}$$

[suggested originally by Levenberg (1944) and Marquardt (1963)]. However, a more general approach is the stochastic non-linear least squares approach of Tarantola & Vallette (1982), which reduces to the Levenberg–Marquardt algorithm when diagonal covariance matrices are used to describe the (Gaussian) data and model statistics. When strong damping is used ($\lambda \gg 0$), the damping term dominates and the Gauss–Newton method approaches the gradient method.

Where we invoke regularization, we can represent the approximate inverse thus obtained as the 'pseudo-inverse', $\mathbf{H}^{\dagger}$. The parameter estimates $\delta\hat{\mathbf{p}}$ can be related to the true $\delta\mathbf{p}$ using

$$\delta\hat{\mathbf{p}} = -\mathbf{H}^{\dagger}\,\nabla_p E = \mathbf{Y}\delta\mathbf{p}\,, \tag{37}$$

where

$$\mathbf{Y} = -\mathbf{H}^{\dagger}\mathcal{R}e\{\mathbf{J}^{t}\mathbf{J}^{*}\} \tag{38}$$

is the resolution matrix in the linear approximation (see e.g. Ory & Pratt 1995).

A physical interpretation of the approximate Hessian

In this section we examine the role the partial derivative seismograms play in the Gauss–Newton method by returning to the point diffractor model of Fig. 2. We have already examined the role the partial derivative wavefields play in the computation of the gradient vector. These concepts can also be used to understand the structure of the approximate Hessian used in the Gauss–Newton method and the exact Hessian used in the full Newton method (see next section). In order to solve eq. (35), in addition to the gradient vector we must also form the $m \times m$ approximate Hessian matrix $\mathbf{H}_a = \mathcal{R}e\{\mathbf{J}^{t}\mathbf{J}^{*}\}$.

A similar argument to that used in our discussion of the physical meaning of the gradient vector can be applied to understand the structure of the $\mathbf{H}_a$ matrix. As can be seen from eq. (30), each element in $\mathbf{H}_a = \mathcal{R}e\{\mathbf{J}^{t}\mathbf{J}^{*}\}$ is the scalar product of two partial derivative wavefields at the receiver locations, one of which is conjugated. This operation corresponds to zero lag correlation in the time domain. From Fig. 3, it should be clear that the partial derivative wavefields are largely uncorrelated with each other (and of course perfectly self-correlated). In the high-frequency limit, these wavefields would be perfectly uncorrelated with each other. However, because the frequency content is finite, the partial derivative wavefields from adjacent nodes are in fact correlated to some extent. Thus the $\mathbf{H}_a$ matrix is diagonally dominant, due to the autocorrelations occurring on the main diagonal, and banded due to the finite frequency effects. Furthermore, as the scattering points are removed from the source location, the partial derivative wavefields drop in amplitude, and so too the corresponding elements of the approximate Hessian will drop in amplitude.

Due to excessive cost, the Hessian matrix is not normally computed when the forward model is computed using time-domain finite-difference methods. However, by using the FDM/FDI method, and by taking advantage of the recent efficiency advances made in FDM methods (as well as advances made recently in the speed and memory configurations available on desktop workstations), we are able to compute the approximate Hessian exactly for the FDM/FDI problem. Fig. 5 shows the (symmetric) approximate Hessian matrix, $\mathbf{H}_a$, computed using the homogeneous model in Fig. 2. (The structure of the matrix naturally depends on the ordering used for the node points, which, for the purposes of illustration here is a straightforward row ordering scheme.) The diagonal dominance of the matrix with the off-diagonal bands is clear. The source and receiver locations also affect the structure of the approximate Hessian, due to the geometrical spreading in amplitudes of the partial derivative wavefields. Fig. 5 was computed with a single source in the model, resulting in large correlation values for all elements of the approximate Hessian associated with the source location and its nearest neighbours. The situation would be more complicated if the background velocity model were not homogeneous; however, it is generally true that the approximate Hessian matrix will be diagonally dominant and banded.

The approximate Hessian matrix thus predicts the defocusing that affects the gradient vector due to the incomplete and uneven illumination of the target, and the natural defocusing that occurs due to limited bandwidth of the seismic experiment. The structure of the approximate Hessian matrix (Fig. 5) is similar to a convolutional smoothing operator. The application of the *inverse* Hessian thus sharpens, or focuses, the gradient vector. One could argue that much subsequent work in migration since Hagedoorn's work has been directed at approximating the inverse Hessian to improve the focusing properties of migration. In the work of Lambaré *et al.* (1992) and others, the inverse Hessian is computed under the ray theoretical approximation, and this is used as an operator that focuses the gradient estimates.

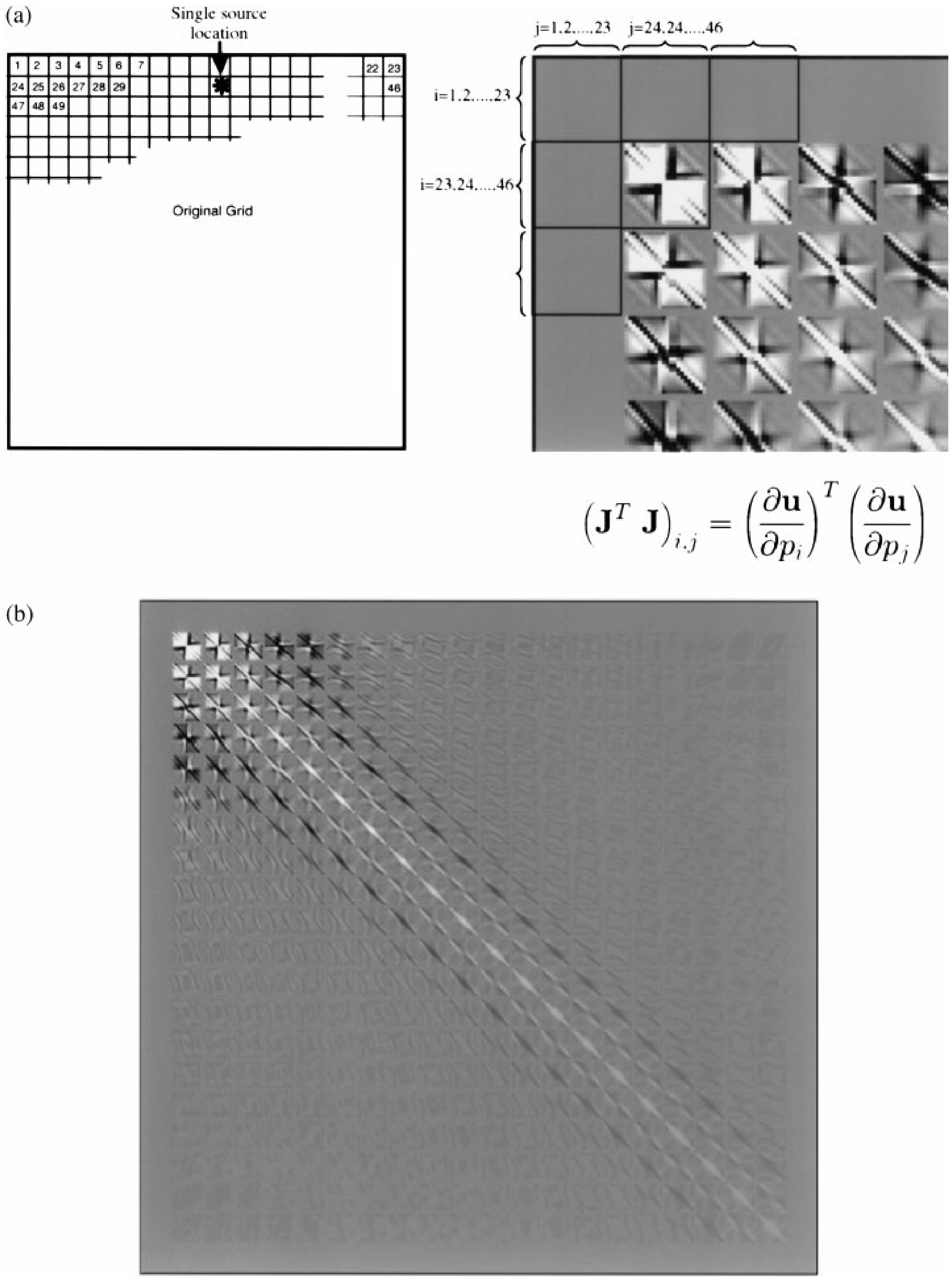
The filtering effect of the inverse of the approximate Hessian on the gradient for our test model (Fig. 2) is shown in Fig. 6. This image depicts the first iteration of a Gauss–Newton inversion. The removal of some of the artefacts seen on the gradient image (Fig. 4) is clear, although the image still suffers from inadequate source–receiver coverage and low-frequency content. It is important to point out that a similar result to that shown in Fig. 2 could have been obtained using the gradient method by iterating sufficiently.

Eq. (36) can thus be interpreted as a $m \times m$ filtering operation that is applied to focus the gradient vector, with a damping term to improve stability. However, there is another interpretation, which becomes evident if we examine an equation equivalent to eq. (36):

$$\delta\mathbf{p} = -\left(\mathbf{K}^{t}\mathbf{K} + \lambda\mathbf{I}\right)^{-1}\mathbf{K}^{t}\delta\mathbf{d}'\,, \tag{39}$$

where

$$\mathbf{K} = \begin{bmatrix} \mathcal{R}e\{\mathbf{J}\} \\ \mathcal{I}m\{\mathbf{J}\} \end{bmatrix} \quad \text{and} \quad \delta\mathbf{d}' = \begin{bmatrix} \mathcal{R}e\{\delta\mathbf{d}\} \\ \mathcal{I}m\{\delta\mathbf{d}\} \end{bmatrix}$$



$$\left(\mathbf{J}^T\,\mathbf{J}\right)_{i,j} = \left(\frac{\partial \mathbf{u}}{\partial p_i}\right)^T \left(\frac{\partial \mathbf{u}}{\partial p_j}\right)$$

Figure 5. A representation of the approximate Hessian matrix, $\mathscr{Re}\{\mathbf{J}^t\mathbf{J}^*\}$, for the model shown in Fig. 2. (a) The original $22\times 23=506$ element grid, and a detailed section of the approximate Hessian showing the organization of the matrix. (b) The total 506×506 element approximate Hessian for the problem. Each pixel in the matrix measures the correlation between partial derivative wavefields emerging from the corresponding node points and another node point. The wavefields are best correlated along the main diagonal, but non-zero, oscillatory behaviour is observed off the main diagonal.

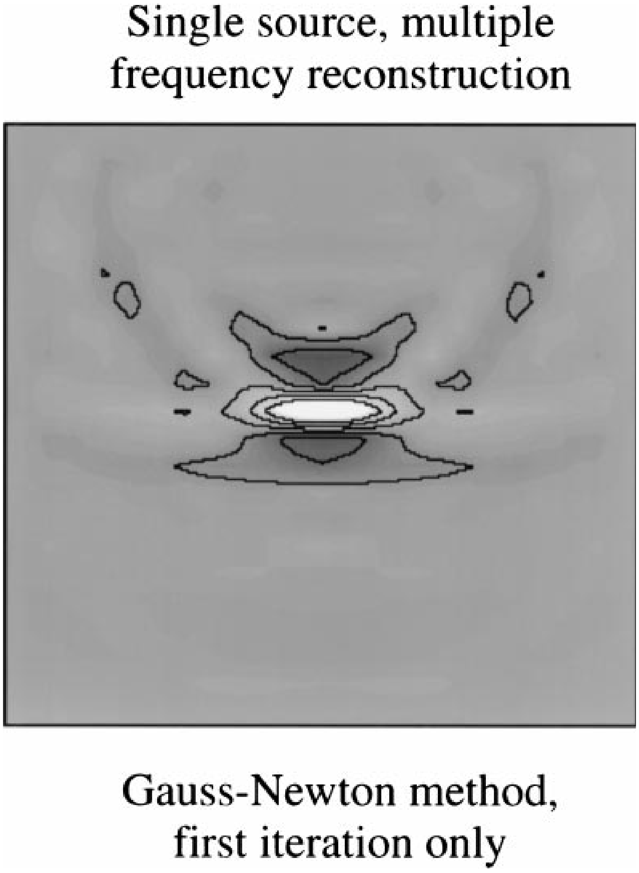


Figure 6. Image computed from data in the model shown in Fig. 2 using a single iteration of the Gauss–Newton method. This image was computed from the same data as the single-iteration gradient image shown in Fig. 4. The artefacts in the gradient image have been effectively removed by filtering the gradient with the inverse of the approximate Hessian matrix (Fig. 5).

are real-valued matrices. Eq. (39) can be recast as

$$\delta\mathbf{p} = -\mathbf{K}^{\mathrm{t}}(\mathbf{K}\mathbf{K}^{\mathrm{t}} + \lambda\mathbf{I})^{-1}\,\delta\mathbf{d}' \tag{40}$$

[see Tarantola (1987) (problem 1.19) for a proof of this manipulation]. In eq. (40) a $2n \times 2n$ (real-valued) filter is applied directly to the $2n$ (n real and n imaginary) residuals before backpropagation. In problems in which $n \leq m$ (underdetermined problems) the use of this approach will reduce the size of the matrix that needs to be inverted, although the computation of all m partial derivatives is still required. This is very closely related to the filtered backpropagation algorithm developed by Devaney (1982) (see also Wu & Toksöz 1987) in order to implement diffraction tomography. Eq. (40) thus appears to be the generalization of diffraction tomography to any background medium, with any source–receiver geometry and for any wave equation (acoustic, elastic, electromagnetic, etc.) that can be approximated by an FDM method as in eq. (5).

The full Newton method of inversion

We now return to the exact expression of the Hessian matrix, eq. (32). If we use this exact expression in the Newton algorithm for inversion, eq. (29), we obtain the full Newton method

$$\delta\mathbf{p} = -(\mathbf{H}_a + \mathbf{R})^{-1}\,\nabla_p E\,, \tag{41}$$

where

$$\mathbf{R} = \mathscr{R}e\left\{\left(\frac{\partial}{\partial\mathbf{p}^{\mathrm{t}}}\,\mathbf{J}^{\mathrm{t}}\right)(\delta\mathbf{d}^* \quad \delta\mathbf{d}^* \quad \ldots \quad \delta\mathbf{d}^*\,)\right\}. \tag{42}$$

Most researchers tend to avoid the computation of the second term, which is awkward to calculate and which in any case should be small (provided the problem is approximately linear, which, in practice, implies that the starting model is sufficiently close to the true answer). However, we have found that in the FDI problem it is no more difficult to compute the second term than the first. In the next two sections we show how a backpropagation approach similar to that used to compute the gradient vector can be used. The analysis leads directly to an interpretation of the matrix $\mathbf{R}$ in terms of second-order scattering (i.e. first-order multiples).

Calculation of the exact Hessian

The exact Hessian consists of two terms, the approximate Hessian $\mathbf{H}_a$, which we have already discussed, and $\mathbf{R}$, defined in eq. (42). From eq. (30) the elements of $\mathbf{R}$ are

$$R_{ij} = \mathscr{R}e\left\{\left[\frac{\partial^2\mathbf{u}^{\mathrm{t}}}{\partial p_i\partial p_j}\right]\delta\hat{\mathbf{d}}^*\right\}$$
$$i = (1, 2, \ldots, m); \quad j = (1, 2, \ldots, m)\,, \tag{43}$$

where we now augment $\delta\mathbf{d}$ with $l - n$ zeros, as in eq. (14). It is clear from this expression [and by analogy with eq. (13) for the gradient] that elements in $\mathbf{R}$ are computed by cross-correlating second-order partial derivatives with the data residuals. The first-order partial derivatives are related to first-order scattering; it follows that the second-order partial derivatives are related to second-order scattering. We will show the exact form of this relationship in the next few paragraphs.

In an earlier section we gave analytical formula for the first-order derivatives in terms of first-order virtual sources. We now develop an analytical formula for the second-order partial derivatives in terms of second-order virtual sources. The second-order partial derivatives may be computed by pursuing the analysis begun earlier:

$$\mathbf{S}\,\frac{\partial\mathbf{u}}{\partial p_i} = -\frac{\partial\mathbf{S}}{\partial p_i}\,\mathbf{u} \tag{44}$$

(eq. 15 again). Taking the derivative of both sides of eq. (44) with respect to p_j yields

$$\mathbf{S}\,\frac{\partial^2\mathbf{u}}{\partial p_j\partial p_i} + \left(\frac{\partial\mathbf{S}}{\partial p_j}\right)\left(\frac{\partial\mathbf{u}}{\partial p_i}\right) = -\left(\frac{\partial\mathbf{S}}{\partial p_i}\right)\left(\frac{\partial\mathbf{u}}{\partial p_j}\right) - \frac{\partial^2\mathbf{S}}{\partial p_j\partial p_i}\,\mathbf{u}\,. \tag{45}$$

Solving eq. (45) for the required second derivatives yields

$$\mathbf{S}\,\frac{\partial^2\mathbf{u}}{\partial p_j\partial p_i} = -\mathbf{f}^{(ij)} \quad \text{or} \quad \frac{\partial^2\mathbf{u}}{\partial p_j\partial p_i} = -\mathbf{S}^{-1}\mathbf{f}^{(ij)}\,, \tag{46}$$

where we have introduced a new, second-order virtual source term

$$\mathbf{f}^{(ij)} = \left(\frac{\partial\mathbf{S}}{\partial p_i}\right)\left(\frac{\partial\mathbf{u}}{\partial p_j}\right) + \left(\frac{\partial\mathbf{S}}{\partial p_j}\right)\left(\frac{\partial\mathbf{u}}{\partial p_i}\right) + \frac{\partial^2\mathbf{S}}{\partial p_j\partial p_i}\,\mathbf{u}\,. \tag{47}$$

Thus, as for the first derivatives, the second-order partial derivatives can also be generated by solving a forward problem using a virtual source term. There are three terms in the virtual sources defined by eq. (47). The first term, $(\partial \mathbf{S}/\partial p_i)(\partial \mathbf{u}/\partial p_j)$ is a virtual source at the ith node, excited by the partial derivative wavefield from the jth node. Since the partial derivative wavefield is itself a first-order scattered wavefield, this new virtual source term generates the double scattered wavefield or first-order multiple from these two node points. Similarly, $(\partial \mathbf{S}/\partial p_j)(\partial \mathbf{u}/\partial p_i)$ is a virtual source at the jth node, excited by the partial derivative wavefield from the ith node. The final term, $(\partial^2 \mathbf{S}/\partial p_j \partial p_i)\mathbf{u}$, depends on the parametrization and the exact details of the finite-approximation scheme. In most cases, where orthogonal basis functions are used to parametrize the model (such as the point collocation scheme), this term will equal zero unless $i=j$. We may therefore think of $\partial^2 \mathbf{u}/\partial p_j \partial p_i$ as the sum of two doubly scattered wavefields, the first representing scattering from node j, then from node i, and the second representing scattering from node i, then from node j.

Returning to eq. (43), we see that the i, jth element of $\mathbf{R}$ represents the sum of the doubly scattered wavefields, each conjugated, and multiplied by the data residuals. Thus R_{ij} is a measure of the correlation of the residuals with second-order scattered events. The effect on the gradient of first-order multiples, as a specific class of double scattered events, will therefore be included in the correlation information contained in $\mathbf{R}$. It should also be noted that the diagonal elements R_{ii} can take on non-zero values when the scattered waves are reflected back to the scattering node to be rescattered, such as when there is a free surface in the model.

With this understanding of the elements of $\mathbf{R}$, let us re-examine the full Newton estimate of the parameter updates,

$$(\mathbf{H}_a + \mathbf{R})\,\delta\mathbf{p} = -\nabla_p E \quad \text{or} \quad \delta\mathbf{p} = -(\mathbf{H}_a + \mathbf{R})^{-1}\,\nabla_p E\,. \tag{48}$$

The two components of the Hessian sum to act as a filter that relates the parameter updates to the gradient vector. We have seen that the first term, $\mathbf{H}_a = \mathscr{Re}\{\mathbf{J}^{t}\mathbf{J}^{*}\}$, predicts the defocusing inherent in the correlation technique due to finite frequencies and uneven source–receiver coverage. We now see that the second term, $\mathbf{R}$, predicts the artefacts in the gradient vector that occur due to double scattered events. The inverse of the full Hessian therefore acts as a deconvolution operator that now includes terms that operate on first-order multiples. The second term in the Hessian, usually neglected, therefore has the potential for acting as a de-multiple operator, even in the first iteration. Gradient methods applied iteratively would eventually take care of the same effects, but the full Newton method has the potential for converging more rapidly. Nevertheless, the convergence of any of these methods in the presence of multiple energy is entirely dependent on being within the region of the global minimum of the misfit function—and the higher the order of multiples in the data, the more difficult it is to ensure that this condition is met.

As previously discussed, the correlation technique for the gradient vector produces false anomalies by mistakenly correlating energy from multiple scattering in the residuals with the partial derivative wavefields. Similarly, in the computation of $\mathbf{R}$, single scattered energy in the residuals is mistakenly correlated with the second-order partial derivative wavefields. This erroneous correlation is far more significant for the latter case, as single scattered energy is much larger in amplitude than double scattered energy. Consequently, although $\mathbf{R}$ does correctly predict the presence of first-order multiple energy in the gradient vector, it will also produce significant false predictions, especially near source or receiver locations in the model. These spurious parameter perturbations may be such that the misfit cannot be further reduced when incorporating the additional term in the Hessian, in which case the matrix $\mathbf{R}$ is not helpful. In such a case, the matrix $\mathbf{R}$ can destroy the positive definiteness of the approximate Hessian. If the Hessian is no longer positive definite, then our estimate of the misfit function $E(\mathbf{p})$ is not locally convex, and gradient methods are preferable.

Efficient calculation of the exact Hessian

Computing all possible second-order derivatives using eq. (46) would require m^2 forward problems to be solved, in addition to the m forward problems that are required to compute the virtual sources (one for each partial derivative wavefield) in eq. (47). However, eq. (43) only requires that we know the action of the second derivative matrix on the residuals, $\delta\mathbf{d}$. Taking the transpose of eq. (46) yields

$$\left[\frac{\partial^2 \mathbf{u}}{\partial p_i \partial p_j}\right]^{t} = -[\mathbf{f}^{(ij)}]^{t}[\mathbf{S}^{-1}]^{t}\,. \tag{49}$$

Substituting eq. (49) into eq. (43) yields

$$R_{ij} = -\mathscr{Re}\left\{\left[\mathbf{f}^{(ij)}\right]^{t}\,\mathbf{v}\right\}, \tag{50}$$

where $\mathbf{v}$ is defined, as before, by

$$\mathbf{v} = [\mathbf{S}^{-1}]^{t}\delta\hat{\mathbf{d}}^{*} \tag{51}$$

(eq. 20 again). By analogy with the computation of the gradient vector, each element of $\mathbf{R}$ is computed in two steps. (1) The wavefield, $\mathbf{v}$, is computed by backpropagation (this field may already be available following the computation of the gradient vector). (2) The backpropagated wavefield is multiplied by the second-order virtual sources (created by the first-order scattered wavefields), following which the real (zero phase) component is extracted. Thus this step requires the solution of m additional forward problems, required to compute the second-order virtual sources from the first-order partial derivatives using eq. (47) (and these first-order partial derivatives may already be available following the computation of the approximate Hessian). Although m may be a large number of forward problems to solve, it is much less than m^2. We noted earlier that there are cases in which the problem may be initially characterized by a reduced parameter set, so that m may be quite small. We illustrate such a problem in the next section of this paper.

THE FULL NEWTON METHOD IN ESTIMATING BACKGROUND VELOCITIES

The introduction of the approximate and full Hessian matrices in the previous sections, and the efficiency of the FDM/FDI methods, allow one to consider applying the full Newton method to realistic, full-sized seismic-waveform inversion problems. For the full Newton method to be of utility, the problem should meet the following two criteria: (1) it must be possible to parametrize the problem using only a small number of parameters, in order to avoid having to compute and invert large Hessian matrices, and (2) the problem should be

GJI000 21/4/98 09:15:58 3B2 version 5.20
The Charlesworth Group, Huddersfield 01484 517077

quasi-quadratic, with significant second-order non-linearities. A classical problem that meets both these conditions is that of estimating the smoothly varying, low-wavenumber 'background' velocity in the inversion of seismic-reflection data. Such problems have been the subject of numerous investigations (Mora 1987b, 1989; Snieder *et al.* 1989; Cao *et al.* 1990; Symes & Carazzone 1991; Chavent & Jacewitz 1995; Jervis, Sen & Stoffa 1996; Varela, Stoffa & Sen 1996). In this section we illustrate the use of the gradient, Gauss–Newton and full Newton methods in helping to solve this problem effectively.

The velocity model used in this study is shown in Fig. 7(a). The velocity varies only with depth, and consists of a low-wavenumber (background) component, combined with a higher-wavenumber (reflective) component. For simplicity, in this study the density is assumed to be constant (and known). We synthesized a reflection experiment over the top of this model using a single source and a split spread of 28 receivers, with a maximum offset of 700 m. These synthetic data are shown in Fig. 7(b), along with several traces representing the high-wavenumber model in time, after convolving with the source wavelet. The source wavelet has a dominant frequency of 10 Hz and a maximum frequency of 25 Hz. There is a good correlation between the zero-offset trace and the high-wavenumber model, although there is a decay in the data amplitudes with time due to geometrical spreading, and there is some evidence of interbed multiples in the data.

The objective in this simple numerical experiment is to extract as much information as possible about the full spectrum of the 1-D velocity model from the recorded data, starting from a single, reasonable velocity gradient model. One might initially expect that the problem is amenable to a straightforward application of iterative inversion methods. Both high and low wavenumbers are represented in the data: the high wavenumbers generate the reflections in the first place, and control the zero-offset time and the amplitudes of the reflections; the low wavenumbers control the change in traveltimes of the reflections with offset (the seismic 'moveout'). However, it has long been recognized that simple gradient-type methods applied to reflection data of this type either fail to converge, or fail to converge sufficiently rapidly on the long-wavelength components of the velocity field (Gauthier *et al.* 1986; Mora 1987a,b, 1989).

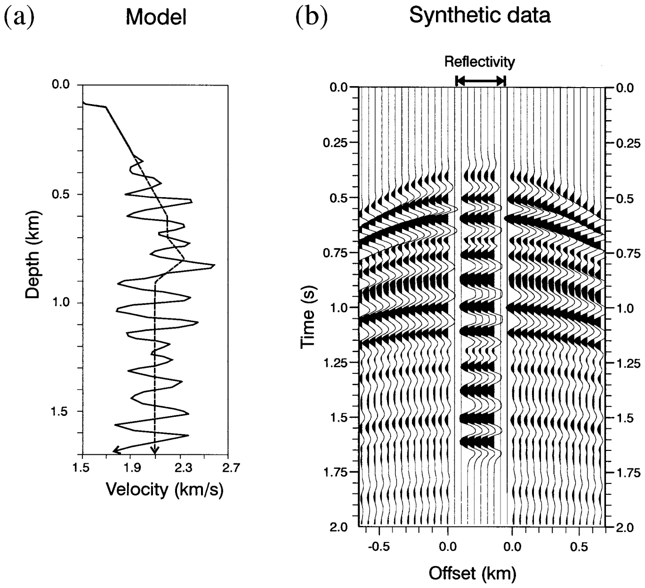


Figure 7. (a) Depth model used to test the performance of the gradient and Newton methods in estimating the 'background' velocity. The model is built using a combination of a low-wavenumber (background) velocity field (dashed) and a high-wavenumber (reflectivity) velocity field. The sum of these two components is the total velocity–depth function (solid). (b) Synthetic seismic-reflection data computed for the model shown in (a) (the direct arrivals have been suppressed on this plot for clarity). The central traces in the panel represent the high-wavenumber velocity model in two-way time after convolution with the source signature.

A partial solution to the low-wavenumber convergence problem is to formulate an additional and entirely separate inversion step designed to solve only for low wavenumbers. However, the low wavenumbers may fail to converge on the correct solution due to the existence of local extrema in the misfit function. Local extrema may be encountered, for example, with an incorrect velocity model for which the predicted zero-offset time of one arrival coincides with the measured zero-offset time of a different arrival. Sen & Stoffa (1991) and Varela *et al.* (1996) suggested using simulated annealing to overcome this difficulty. Simulated annealing, however, requires a large number of forward models to be evaluated. The low-wavenumber inversion problem is also strongly coupled to the high-wavenumber problem, since perturbing the background velocity perturbs not only the moveout of the reflections, but also the zero-offset two-way traveltimes. Snieder *et al.* (1989) showed that a reparametrization of the high-wavenumber function from depth to two-way vertical traveltime can be used to help decouple the low- and high-wavenumber inversion problems. In this approach, the zero-offset times remain unchanged as the low wavenumbers are altered, and for this reason local extrema are much less likely to be encountered. However, the problem is still strongly non-linear. Snieder *et al.* (1989) and Cao *et al.* (1990) suggested a simplex search algorithm for the low wavenumbers.

In this example we adopt two of the suggestions described in the previous paragraph: we separate the high- and low-wavenumber problems by formulating distinct inversion steps for each of these, and we reparametrize the high-wavenumber inversion problem from depth to time, in order to decouple the two inverse problems. With such an approach, we now show how Newton methods can be effectively used to converge on the low wavenumbers in the problem. We parametrize the low-wavenumber velocity field using a small number of cubic spline node points (after Cao *et al.* 1990, for example). In order to avoid complication, we assume from here on that the high-wavenumber component of the velocity field is known in two-way time (but not in depth). This is not a drastic assumption, as a good reflectivity estimate in two-way time can usually be recovered in a straightforward manner, either by using a simple stack of the data, or, in the case of lateral velocity variations, by applying the standard method of seismic time migration.

The algorithm used to compute the update to the low-wavenumber velocity field for a given iteration is outlined below.

(1) We begin by explicitly calculating the required partial derivatives. The elements of the $n \times m$ Frechét derivative matrix are defined as

$$J_{ij} = \frac{\partial u_i}{\partial p_j}\,, \quad i = (1, 2, \ldots, n)\,; \quad j = (1, 2, \ldots, m) \tag{52}$$

(eq. 11 again), where there are n data and m model parameters. In this study we use five frequency components of the data (equally spaced between 5 and 25 Hz), thus for the 28 receivers there are $n = 28 \times 5 = 140$ data. We parametrize the low-wavenumber model using the velocity values at each of four cubic spline node points, thus for the low-wavenumber problem there are only $m = 4$ model parameters. We compute the partial derivatives (the columns of $\mathbf{J}$) directly, by individually perturbing each of the four low-wavenumber parameters. During the perturbation, the high-wavenumber velocity model is stretched vertically in such a manner as to maintain the vertical two-way traveltimes (this effectively reparametrizes the problem from depth, to time, after Snieder *et al.* 1989). In this manner, the data perturbations contain perturbed moveouts, but unperturbed zero-offset traveltimes. The effect of the high- and low-wavenumber components of the velocity field on the data are thus decoupled.

(2) The (low-wavenumber) gradient vector is computed using,

$$\nabla_p E = \mathscr{Re}\{\mathbf{J}^{\mathrm{t}}\,\delta\mathbf{d}^*\} \tag{53}$$

(eq. 10 again). If we are using the gradient method to solve for the low wavenumbers, our model update is obtained by stopping here, computing a step length, and applying it to the gradient vector $\nabla_p E$ using

$$\delta\mathbf{p} = -\alpha^{(k)} \nabla_p E^{(k)}\,, \tag{54}$$

as in eq. (9).

(3) If we are using a Gauss–Newton method, we now compute the approximate Hessian matrix, given by

$$\mathbf{H}_a = \mathscr{Re}\{\mathbf{J}^{\mathrm{t}}\mathbf{J}^*\} + \lambda\mathbf{I}\,, \tag{55}$$

where we include a damping factor λ, estimated from the square root of the sum of the squares of the elements of $\mathbf{J}$ (Dimri 1992) . The partial derivatives are available, as we have already computed them explicitly when computing the gradient vector in the previous step.

(4) The (unscaled) Gauss–Newton parameter perturbations can then be computed:

$$\delta\hat{\mathbf{p}}^{(1)} = -\mathbf{H}_a^{-1}\,\nabla_p E \tag{56}$$

(eq. 35 again, with the superscripts in the notation as in eq. B9). The damping used in eq. (52) causes the length of $\delta\hat{\mathbf{p}}^{(1)}$ to be underestimated. Therefore, as for the gradient perturbation, we need to compute a step length and apply it to this perturbation vector.

(5) In order to extend the inversion to the full Newton method, an additional perturbation direction needs to be computed. This contains the contribution of the second (non-linear) term in the exact Hessian. We use the (unscaled) second-order perturbation

$$\delta\hat{\mathbf{p}}^{(2)} \approx -\mathbf{H}_a^{-1}\mathbf{R}\,\delta\hat{\mathbf{p}}^{(1)}\,. \tag{57}$$

This equation is obtained from the series expansion in the last line of eq. (B14) of Appendix B. Eqs (56) and (57) relate to the first and second terms in this expansion, respectively. The matrix $\mathbf{R}$ is computed using the backpropagation method of eq. (48), requiring us to solve (only) one more forward problem to compute the backpropagated wavefield.

(6) As yet the relative scale factors for the two terms, $\delta\hat{\mathbf{p}}^{(1)}$ and $\delta\hat{\mathbf{p}}^{(2)}$, are undetermined. The optimal full Newton parameter perturbation vector lies in the plane defined by the vectors given by eqs (56) and (57). It may be fully described by

$$\delta\hat{\mathbf{p}} = \alpha\delta\hat{\mathbf{p}}^{(1)} + \beta\delta\hat{\mathbf{p}}^{(2)}\,. \tag{58}$$

The direction of this perturbation vector depends on the relative scale factors, α and β. These are obtained by solving a 2-D subspace Gauss–Newton inversion:

$$\begin{bmatrix}\alpha\\ \beta\end{bmatrix} = (\mathscr{Re}\{\mathbf{A}^{\mathrm{t}}\mathbf{A}^*\} + \lambda\mathbf{I})^{-1}\;\mathscr{Re}\{\mathbf{A}^{\mathrm{t}}\,\delta\mathbf{d}^*\}\,, \tag{59}$$

where $\mathbf{A}$ is an $n \times 2$ matrix containing the two partial derivative seismograms produced by perturbing the current velocity field in the directions of $\delta\hat{\mathbf{p}}^{(1)}$ and $\delta\hat{\mathbf{p}}^{(2)}$. Damping is again introduced due to the ill conditioning of the projected Hessian, $\mathbf{A}^{\mathrm{t}}\mathbf{A}^*$. As a result, we again require a step-length computation for this perturbation vector, before applying it to the low-wavenumber model.

(7) The final low-wavenumber velocity update given by eqs (53) (for the gradient method), (56) (for the Gauss–Newton method) and (58) (for the full Newton method) is found (after the computation of a step length in each case) by interpolating between the m node points, using cubic splines. In each case, after the low-wavenumber perturbation has been used to update the low-wavenumber velocity field component, the high-wavenumber velocity model is stretched so as to preserve the vertical two-way traveltime information, forming a final new velocity–depth function.

The results of applying these algorithms to the example problem are shown in Fig. 8. The first panel, Fig. 8(a), shows the unscaled, low-wavenumber perturbation functions computed for (1) the gradient, (2) the Gauss–Newton perturbation, $\delta\hat{\mathbf{p}}^{(1)}$, and (3) the second-order perturbation function, $\delta\hat{\mathbf{p}}^{(2)}$ (that is the terms given by eqs 53, 56 and 57, respectively, before the computation of a step length). The perturbation functions are constrained to zero within the water layer (we assume we know the water depth and velocity). The gradient term is clearly dominated by the near-surface contribution, a direct result of the decreasing sensitivity of the data to velocity perturbations with depth. The perturbation function derived from the Gauss–Newton method is more hopeful: here the perturbation function peaks close to the depth at which the starting model is most in error, and is negative below the centre of the model. However, the Gauss–Newton perturbation function, like the gradient term, does not contain a significant contribution from the velocity node in the deepest part of the model. We interpret this error as originating in the non-linearity of the problem; that is, the gradient term in the deepest part of the model varies significantly if the velocity field is changed in the shallower part of the model—higher-order terms are required to account for this variation. This increase in error with depth can be understood with reference to the effect of a velocity perturbation on the propagating wavefields used to compute the gradient. As we perturb the velocity field, we will perturb the downgoing wavefields. Since the gradient

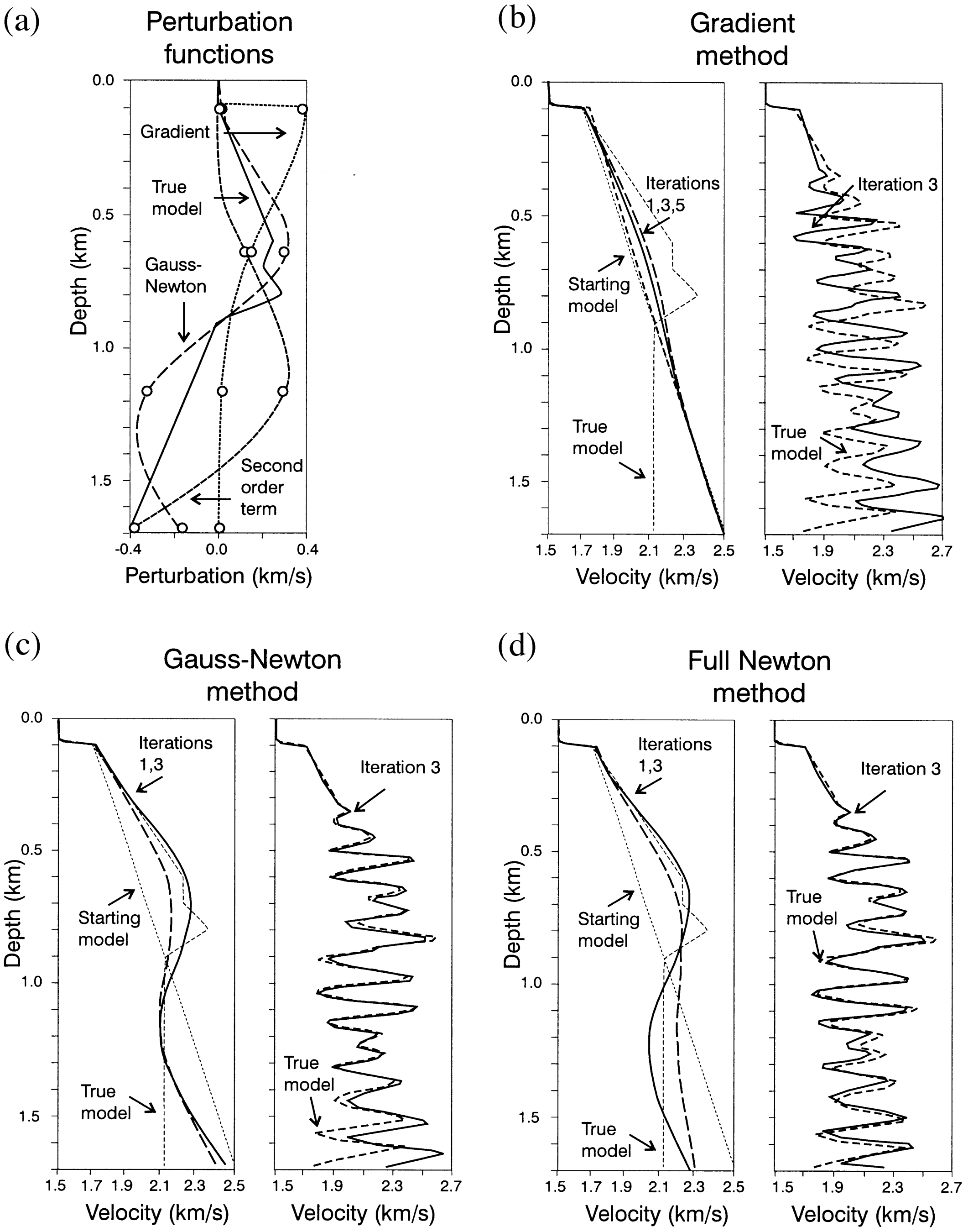


Figure 8. (a) Unscaled perturbation functions for the low-wavenumber problem (normalized to the true maximum error of 400 m s^{-1}) for the gradient method (dotted line), the Gauss–Newton method (short-dashed line) and the contribution from the non-linear term in the full Newton method (long-dashed line). The circles depict the spline node points used. (b) Left: the low-wavenumber result of one, three and five iterations of the gradient method. Right: the total velocity model after three iterations of the gradient method. (c) As in (b), but using the Gauss–Newton method. (d) As in (b) and (c), but using the full Newton method.

term represents, effectively, the scaled multiplication of two such propagating wavefields, **u** and **v** (see eq. 22), the gradient will therefore be particularly affected at large depths. The second-order perturbation term $\delta\hat{\mathbf{p}}^{(2)}$ shown in Fig. 8(a) clearly responds to the mismatch in the velocities at the bottom of the model.

In Figs 8(b), (c) and (d) we depict the results of applying the gradient, Gauss–Newton and full Newton methods to recover the low wavenumbers of the velocity model. In each case, we show the first and third iteration results; for the gradient method we also show the fifth iteration result. For completeness, we also show the final results including the high-wavenumber components. Recall that we assumed that the high-wavenumber component was known as a function of two-way vertical traveltime—the high-wavenumber component was depth-corrected in each case using the current low-wavenumber velocity estimate. It is evident from Fig. 8(b) that the gradient method, even after five iterations, fails to produce any significant correction to the deeper part of the low-wavenumber model. As a result, the total velocity–depth model, after inclusion of the high wavenumbers, is significantly in error. The Gauss–Newton approach, shown in Fig. 8(c), succeeds in restoring the low-wavenumber velocities everywhere except at the very bottom of the model in three iterations. The full Newton approach, in Fig. 8(d), succeeds in restoring the velocities to within approximately 80 m s^{-1} down to a depth of 1600 m, still with only three iterations. As a result, the total velocity–depth model from the full Newton inversion is the closest to the true velocity–depth model. It should be noted that none of the methods succeeds in perfectly reconstructing the depth model between 600 and 900 m, due to the inadequacy of the four node spline parametrization—naturally in a real inversion we would proceed by introducing more nodes into the parametrization at this stage.

In Fig. 9 we depict the data residuals for the three results shown in Figs 8(b), (c) and (d), on the same scale as the original data for the experiment (in each case the result after three iterations is used). Although our inversions were carried out in the frequency domain, we computed these residuals by full time-domain simulation in the final models, and subtracting the result from the original synthetic data. It can be observed that the gradient method fails to account for much of the data—it has succeeded in accounting for some near-offset, shallow data, but the moveouts are not well accounted for, and the residuals tend to increase with offset. The Gauss–Newton method performs significantly better after three iterations, accounting for much more of the data, leaving only some energy in the residuals at about 800 ms and at about 1600 ms. The residuals for the full Newton method do not reduce the data residuals at 800 ms, which is due to the inability of the parametrization to fit the low-wavenumber velocity model (remarked on in the previous paragraph). However, the full Newton method does succeed in reducing the residuals at 1600 ms, nearly to zero. This success appears to be due to the ability of the full Newton method to account for the changes in the partial derivatives of the low-wavenumber inversion problem due to the velocity perturbations.

In theory, both the gradient method and the Gauss–Newton method should eventually converge to the same result as the full Newton method, given a sufficient number of iterations. This, however, did not prove to be the case for the gradient method, which was iterated further. We showed the result after five iterations in Fig. 8(b); very little improvement was observed. This failure is in part due to the difficulty in accurately computing the step length—as the gradient method iterates, we find ourselves on either side of a long, narrow valley in the misfit function, unable to locate the narrow base of the valley (a more accurate, more expensive, step-length computation might alleviate this to some extent, as would a conjugate gradient implementation). The Gauss–Newton and full Newton methods succeed in locating a direction that points along the valley, rather than down the steep slope of the valley. The cost of one iteration of each of the three methods is comparable: we compute the partial derivatives explicitly in each case (that is we do not use backpropagation to compute the gradient, as this would not allow us to implement the time–depth conversion in the high wavenumbers). Since we have four model parameters, four forward models are required to compute the partial derivatives. The step length in each case requires one additional forward model to compute. The

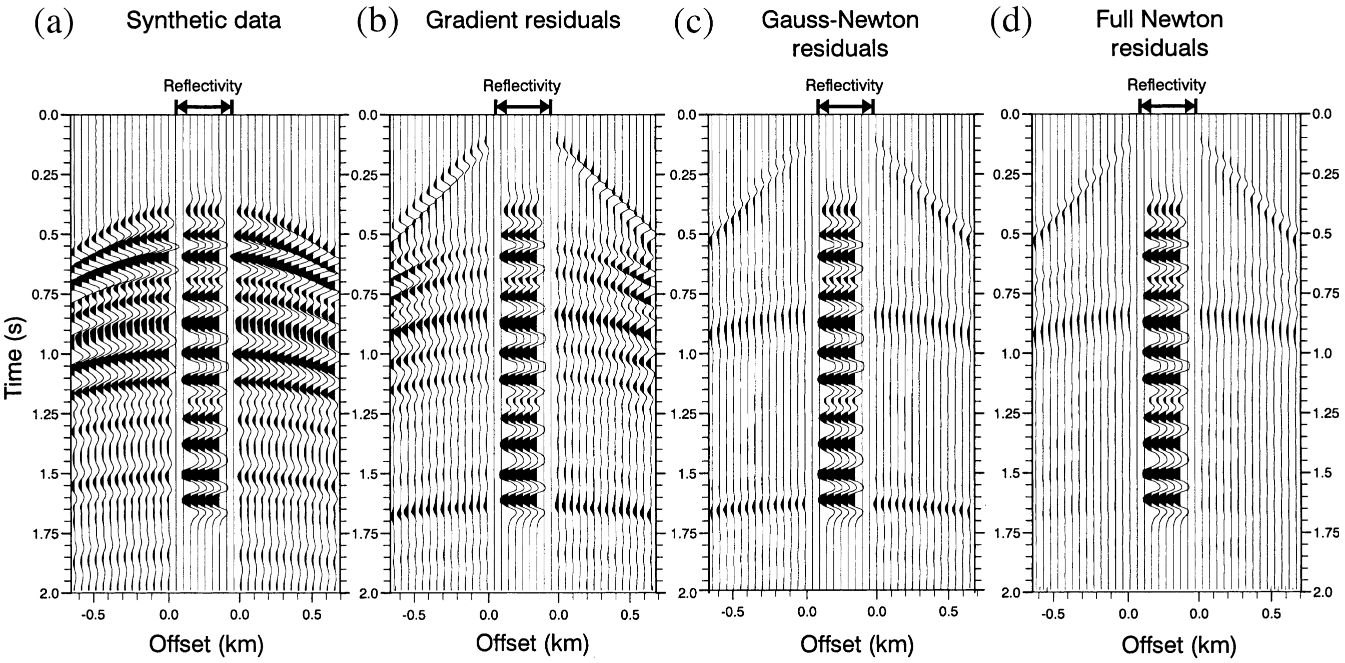


Figure 9. (a) Reflection data for the model shown in Fig. 7(a) (Fig. 7b again). For clarity, the direct arrivals have been suppressed. (b) Data residuals after three iterations of the gradient method. (c) Data residuals after three iterations of the Gauss–Newton method. (d) Data residuals after three iterations of the full Newton method. All seismic traces are displayed in true amplitude, with the same scaling factor being used for all four panels. For reference purposes in each of these panels the central traces depict the high-wavenumber part of the velocity model, convolved with the source signature.

Gauss–Newton method thus requires exactly the same work as the gradient method (neglecting the trivial number of operations required for matrix inversion and matrix multiplication using the projected Hessian); the full Newton method requires an additional forward model to compute the backpropagation, plus two more forward models to compute the relative scaling factors, α and β in eq. (59).

CONCLUSIONS

In this paper we have introduced a link between a discrete, frequency–space method (FDM) for modelling seismic-wave propagation, and the frequency–space inversion methods (FDI) that make use of the discrete modelling method. This linkage allows us to pursue a matrix formalism for the analysis of the inverse problem. Specifically, we have presented a new derivation of Lailly's (1983) fast method for computing the gradient of the misfit function by backpropagation, we have demonstrated the computation of the approximate Hessian, and we have given a new, fast backpropagation method for computing the additional term in the exact Hessian matrix. These theoretical results are applicable to any partial differential equation that can be discretized using a complex-valued impedance matrix, including viscoacoustic and visco-elastic equations (potentially including anisotropy), or, more generally, potentially useful for problems based on Maxwell's equations for electromagnetic fields or for potential field problems. The results hold for 1-D, 2-D and 3-D problems, and for 2.5-D problems. We have shown that the inverse Hessian can be manipulated to yield a filtered backpropagation algorithm, similar to that used in 'diffraction tomography' by Devaney (1982) and by Wu & Toksöz (1987), but generally applicable to all the physical problems listed above.

For the acoustic 2-D seismic problem, we were able to compute numerically the approximate Hessian matrix for a realistic-sized problem in reflection seismology. We showed in a numerical example how the use of the inverse of the approximate Hessian would improve convergence in iterative methods by refocusing the gradient image of single point scatters. Convergence is a critical issue in iterative inversion for large seismic problems; since the FDM method can compute partial derivatives rapidly by forward propagation from virtual sources, we believe this approach may find some application.

In the final section of this paper we presented a second numerical example. We used gradient, Gauss–Newton and full Newton methods to compute the low-wavenumber (background) velocity structure from simulated, multi-offset, surface seismic-reflection data. This is a rather classic problem in reflection seismology, of considerable importance. In our example we showed the potential that the Gauss–Newton and full Newton methods have in accelerating convergence of iterative solutions in such problems, without introducing excessive additional costs.

ACKNOWLEDGMENTS

We would like to acknowledge the helpful comments and suggestions made by Patrick Lailly and by an anonymous reviewer. GJH is supported by the Natural Environmental Research Council (UK).

REFERENCES

Bathe, K.J. & Wilson, E.L., 1976. *Numerical Methods in Finite Element Analysis,* Prentice-Hall, NJ.

Bertsekas, D.P., 1982. Enlarging the region of convergence of Newton's method for constrained optimization, *J. Optimization Theory Applications,* **36,** 221–251.

Beydoun, W.B., Delvaux, J., Mendes, M., Noual, G. & Tarantola, A., 1989. Practical aspects of an elastic migration/inversion of crosshole data for reservoir characterization: a Paris basin example, *Geophysics,* **54,** 1587–1595.

Beydoun, W.B., Mendes, M., Blanco, J. & Tarantola, A., 1990. North sea reservoir description: benefits of an elastic migration/inversion applied to multicomponent vertical seismic profile data, *Geophysics,* **55,** 209–218.

Bunks, C., Saleck, F.M., Zaleski, S. & Chavent, G., 1995. Multiscale seismic waveform inversion, *Geophysics,* **60,** 1457–1473.

Cao, D., Beydoun, W.B., Singh, S.C. & Tarantola, A., 1990. A simultaneous inversion for background velocity and impedence maps, *Geophysics,* **55,** 458–469.

Chavent, G. & Jacewitz, C.A., 1995. Determination of background velocities by multiple migration fitting, *Geophysics,* **60,** 476–490.

Claerbout, J.F., 1985. *Fundamentals of Geophysical Data Processing,* Blackwell Scientific Publications, Oxford.

Crase, E., Pica, A., Noble, M., McDonald, J. & Tarantola, A., 1990. Robust elastic nonlinear waveform inversion: application to real data, *Geophysics,* **55,** 527–538.

Delprat-Jannaud, F. & Lailly, P., 1992. Ill-posed and well-posed formulations of the reflection traveltime tomography problem, *J. geophys. Res.,* **97,** 19 827–19 844.

Devaney, A.J., 1982. A filtered backpropagation algorithm for diffraction tomography, *Ultrasonic Imaging,* **4,** 336–350.

Dimri, V., 1992. *Deconvolution and Inverse Theory: Application to Geophysical Problems, Methods in Geochemistry and Geophysics,* Vol. 29, Elsevier, Amsterdam.

Farra, V. & LeBegat, S., 1995. Sensitivity of *qp*-wave traveltimes and polarization vectors to heterogeneity, anisotropy and interfaces, *Geophys. J. Int.,* **121,** 371–384.

Gauthier, O., Virieux, J. & Tarantola, A., 1986. Two-dimensional non-linear inversion of seismic waveforms: numerical results, *Geophysics,* **51,** 1387–1403.

Goodman, T.R. & Lance, G.N., 1956. The numerical integration of two-point boundary value problems: mathematical tables and other aids to computation, **10,** 82–86.

Grechka, V.Y. & McMechan, G.A., 1997. Analysis of reflection traveltimes in 3-d transversely-isotropic heterogeneous media, *Geophysics,* **62,** 1884–1895.

Griewank, A., 1989. On automatic differentiation, in *Programming: Recent Developments and Applications,* pp. 83–108, eds Iris, M. & Tanabe, K., Kluwer Academic, Dordrecht.

Hagedoorn, J.G., 1954. A process of seismic reflection interpretation, *Geophys. Pospect.,* **2,** 85–127.

Jervis, M., Sen, M.K. & Stoffa, P.L., 1996. Prestack migration velocity estimation using nonlinear methods, *Geophysics,* **60,** 138–150.

Jo, C.H., Shin, C.S. & Suh, J.H., 1996. Design of an optimal 9 point finite difference frequency-space acoustic wave equation scheme for inversion and modeling, *Geophysics,* **61,** 529–537.

Jurovics, S.A. & McIntyre, J.E., 1962. The adjoint method and its application to trajectory optimization, *ARS J.,* **32,** 1354.

Kennett, B., Sambridge, M.S. & Williamson, P.R., 1988. Subspace methods for large inverse problems with multiple parameter classes, *Geophys. J.,* **94,** 237–247.

Kolb, P., Collino, F. & Lailly, P., 1986. Pre-stack inversion of a 1-D medium, *Proc. IEEE,* **74,** 498–508.

Lailly, P., 1983. The seismic inverse problem as a sequence of before stack migrations, in *Conference on Inverse Scattering: Theory and Application,* eds Bednar, J.B., Redner, R., Robinson, E. & Weglein, A., Soc. Industr. appl. Math., Philadelphia, PA.

GJI000 21/4/98 09:17:07 3B2 version 5.20
The Charlesworth Group, Huddersfield 01484 517077

Lambare, G., Virieux, J., Mandariaga, R. & Jin, S., 1992. Iterative asymptotic inversion in the acoustic approximation, *Geophysics,* **57,** 1138–1154.

Lee, E.B. & Markus, L., 1967. *Foundations of Optimal Control Theory,* John Wiley, New York, NY.

Levenberg, K., 1944. A method for the solutoin of certain nonlinear problems in least squares, *Q. appl. Math,* **2,** 164–168.

Liu, J.W. & George, A., 1981. *Computer Solution of Large Sparse Positive Definite Systems,* Prentice-Hall, NJ.

Marfurt, K.J., 1984. Accuracy of finite-difference and finite-element modeling of the scalar and elastic wave-equations, *Geophysics,* **49,** 533–549.

Marfurt, K.J. & Shin, C.S., 1989. The future of iterative modeling of geophysical exploration, in *Supercomputers in Seismic Exploration,* ed. Eisner, E., *Seis. Expl.,* **21,** 203–228.

Marquardt, D.W., 1963. An algorithm for least squares estimation of nonlinear parameters, *J. Soc. Industr. appl. Math.,* **11,** 431–441.

Mora, P.R., 1987a. Nonlinear two-dimensional elastic inversion of multioffset seismic data., *Geophysics,* **52,** 1211–1228.

Mora, P.R., 1987b. Elastic wavefield inversion for low and high wave-numbers of the P- and S-wave velocities, a possible solution, in *Deconvolution and Inversion,* pp. 321–337, eds Bernabini, M., Carrion, P., Jacovetti, G., Rocca, F., Treitel, S. & Worthington, M., Blackwell Scientific Publications, Oxford.

Mora, P., 1989. Inversion = migration + tomography, *Geophysics,* **54,** 1575–1586.

Oristaglio, M.L. & Worthington, M.H., 1980. Inversion of surface and borehole electromagnetic data for two-dimensional electrical conductivity models, *Geophys. Prospect.,* **28,** 633–657.

Ory, J. & Pratt, R.G., 1995. Are our parameter estimates biased? The significance of finite difference operators, *Inverse Problems,* **11,** 397–424.

Pratt, R.G., 1990. Inverse theory applied to multi-source cross-hole tomography. Part II: elastic wave-equation method, *Geophys. Prospect.,* **38,** 311–330.

Pratt, R.G. & Goulty, N.R., 1991. Combining wave-equation imaging with traveltime tomography to form high-resolution images from crosshole data, *Geophysics,* **56,** 208–224.

Pratt, R.G. & Worthington, M.H., 1990. Inverse theory applied to multi-source cross-hole tomography. Part I: acoustic wave-equation method, *Geophys. Prospect.,* **38,** 287–310.

Pratt, R.G., McGaughey, W.G. & Chapman, C.H., 1993. Anisotropic velocity tomography: a case study in a near-surface rock mass, *Geophysics,* **58,** 1748–1763.

Pratt, R.G., Shipp, R.M., Song, Z.M. & Williamson, P.R., 1995. Fault delineation by wavefield inversion of cross-borehole seismic data, *57th Mtg. Eur. Assoc. Expl Geophys., Extended Abstracts EAEG,* 95, Session D001.

Press, W.H., Teukolsky, S.A., Vettering, W.T. & Flannery, B.P., 1992. *Numerical Recipes in FORTRAN: The Art of Scientific Computing,* 2nd edn, Cambridge University Press, Cambridge.

Rodi, W.L., 1976. A technique for improving the accuracy of finite element solutions for magnetotelluric data, *Geophys. J. R. astr. Soc.,* **44,** 483–506.

Santosa, F., 1987. Inversion of band-limited reflection seismograms using stacking velocities as constraints, *Inverse Problems,* **3,** 477–499.

Sen, M.K. & Stoffa, P.L., 1991. Nonlinear one-dimensional seismic waveform inversion using simulated annealing, *Geophysics,* **56,** 1624–1638.

Shin, C.S., 1988. Nonlinear elastic wave inversion by blocky parameterization, *PhD thesis,* University of Tulsa, OK.

Snieder, R., 1990. A perturbative analysis of non-linear inversion, *Geophys. J. Int.,* **101,** 545–556.

Snieder, R., Xie, M.Y., Pica, A. & Tarantola, A., 1989. Retrieving both the impedance contrast and background velocity: a global strategy for the seismic reflection problem, *Geophysics,* **54,** 991–1000.

Song, Z.-M. & Williamson, P.R., 1995. Frequency-domain acoustic-wave modeling and inversion of crosshole data: Part I– 2.5-D modeling method, *Geophysics,* **60,** 784–795.

Song, Z.-M., Williamson, P.R. & Pratt, R.G., 1995. Frequency-domain acoustic-wave modeling and inversion of crosshole data: Part II– inversion method, synthetic experiments and real-data results, *Geophysics,* **60,** 796–809.

Stekl, I. & Pratt, R.G., 1997. Accurate visco-elastic modeling by frequency-domain finite diffferences using rotated operators, *Geophysics,* **63,** in press.

Symes, W.W. & Carazzone, J.J., 1991. Velocity inversion by differential semblance optimization, *Geophysics,* **56,** 654–663.

Tarantola, A., 1984. Inversion of seismic reflection data in the acoustic approximation, *Geophysics,* **49,** 1259–1266.

Tarantola, A., 1986. A strategy for nonlinear elastic inversion of seismic reflection data, *Geophysics,* **51,** 1893–1903.

Tarantola, A., 1987. *Inverse Problem Theory: Methods for Data Fitting and Parameter Estimation,* Elsevier, Amsterdam.

Tarantola, A. & Valette, B., 1982. Generalized nonlinear inverse problems solved using the least-squares criterion, *Rev. Geophys. Space Phys.,* **20,** 219–232.

Varela, C.L., Stoffa, P.L. & Sen, M.K., 1996. Automatic background velocity estimation in 2D media, *58th Mtg. Eur. Assoc. Expl Geophys., Extended Abstracts—Geophysical Division EAEG,* session X009.

Wang, Y. & Housemann, G.A., 1995. Tomographic inversion of reflection seismic amplitude data for velocity variation, *Geophys. J. Int.,* **123,** 355–372.

Williamson, P.R., 1990. Tomographic inversion in reflection seismology, *Geophys. J. Int.,* **100,** 255–274.

Wu, R.S. & Toksoz, M.N., 1987. Diffraction tomography and multisource holography applied to seismic imaging, *Geophysics,* **52,** 11–25.

Xu, T., McMechan, G.A. & Sun, R., 1995. 3-D prestack full-wavefield inversion, *Geophysics,* **60,** 1805–1818.

Zhou, C., Cai, W., Luo, Y., Schuster, G.T. & Hassanzadeh, S., 1995. Acoustic wave-equation traveltime and waveform inversion of crosshole seismic data, *Geophysics,* **60,** 765–773.

Zienkiewicz, O.C. & Taylor, R.L., 1989. *The Finite Element Method,* 4th edn, McGraw-Hill, London.

APPENDIX A: ALTERNATIVE PARAMETRIZATION

In this appendix we clarify the relationship between the variables used to parametrize the inverse problem, and the variables used to compute the elements of the complex impedance matrix, **S**. In the finite-difference method, the elements of **S** are computed using standard schemes from a knowledge of the model at a discrete set of l nodal points. However, it is not necessary or even desirable to use these nodal parameters as the inversion parameters. We shall represent the nodal parameters using the $q{\cdot}l \times 1$ column vector $\mathbf{m}$, from which the impedance matrix is computed directly. The number q represents the number of different physical parameters required at each node point (for example for a problem parametrized in terms of velocity and density, $q=2$). We shall represent the inversion parameters using the $m \times 1$ column vector $\mathbf{p}$ (m is independent of l). In order to compute a forward result from a given inversion result, we require a method for computing the elements of **S** from a given $\mathbf{p}$. It is also necessary to be able to compute the gradient of the misfit function with respect to these parameters, and, for the Hessian matrix, to be able to compute the partial derivatives of the data with respect to these parameters.

If one chooses inversion parameters that can be used to compute modelling variables through linear combinations of basis (or interpolation) functions, then further simplification of the quantities required in the FDI problem is possible. For continuous functions m($\mathbf{r}$), such a scheme can be represented by

$$\mathrm{m}(\mathbf{r})=\sum_{i}^{m} a^{(i)}(\mathbf{r})p_i\,, \tag{A1}$$

where $\{a^{(i)}(\mathbf{r})\}$ is a set of m basis functions (the interpolation functions), and p_i are the components of the vector $\mathbf{p}$. Eq. (A1) yields a continuous expression for the model parameter m($\mathbf{r}$). However, in order to generate the impedance matrix S we only require the values of the model parameters at the nodal points. Evaluating (A1) at the l nodal point locations $\{\mathbf{r}_i\}$, we obtain

$$\mathbf{m}=\sum_{i}^{m}\mathbf{a}^{(i)}p_i=\mathsf{A}\mathbf{p}\,, \tag{A2}$$

where $\mathbf{m}$ is the $q\times l$ vector of modelling parameters, $\mathbf{a}^{(i)}$ are the values of the basis functions at the nodal points (the basis vectors), and A is a (real-valued) projection matrix whose columns comprise these basis vectors.

Many examples of this class of parametrization have been suggested. For example, this subsumes the use of the modelling parameters themselves as inversion parameters (i.e. $\mathbf{p}=\mathbf{m}$). This scheme is referred to as the 'point collocation' scheme, for which the basis functions are spatial delta functions and for which there is a one-to-one mapping of the elements of $\mathbf{p}$ to the elements of $\mathbf{m}$. This choice is the most obvious one. It has been used frequently in waveform inversion (e.g. Mora 1987, but see Xu *et al.* 1995 for a counter-example). However, the degree of discretization required to model accurately the forward problem (multiple node points per wavelength) greatly exceeds the resolution of most geophysical measurement techniques. Using the point collocation scheme is an extreme choice that is wasteful of computing resources and leads to unnecessary non-uniquenesses for the inverse problem. This is especially important when the early stages of a non-linear inverse scheme are only used to solve for the large-scale (smooth) parameters [as advocated by Williamson (1990) and Bunks *et al.* (1995)].

At the other extreme, one may choose to solve only for a single parameter, say the average velocity (in which case the interpolation function is simply a constant). In between these extremes are a variety of possible schemes, including cubic splines (as in Farra & LeBegat 1995), Chebyshev polynomials (as in Grechka & McMechan 1997), truncated Fourier series (as in Wang & Houseman 1995), or bilinear interpolation of parameters on a coarse grid (as in Pratt, McGaughey & Chapman 1993). All of these schemes can be expressed in the form (A1).

Eq. (A2) allows us to generate the impedance matrix S from the current inversion parameters in two stages: first, compute the nodal parameters, $\mathbf{m}$, from the inversion parameters, $\mathbf{p}$, using (A2), then generate the impedance matrix using the appropriate finite-difference scheme. In order to compute the elements of the Frechét derivative matrix, J, we observe from eq. (17) in the body of the paper that we require the virtual source matrix, F, the columns of which comprise the m virtual source vectors $\mathbf{f}^{(i)}$ (one for each inversion parameter), given by

$$\mathbf{f}^{(i)}=-\frac{\partial\mathsf{S}}{\partial p_i}\,\mathbf{u}\,, \tag{A3}$$

where $\mathbf{u}$ is the predicted (forward-propagated) wavefield in the current model, at each of the l node points. Let us use eqs (A2) and (A3) to obtain a relationship between the required virtual source vectors and the virtual source vectors for a point collocation scheme. We have

$$\begin{aligned}\mathbf{f}^{(i)}&=-\frac{\partial\mathsf{S}}{\partial p_i}\,\mathbf{u}=-\left[\sum_{j}^{q\times l}\frac{\partial\mathsf{S}}{\partial m_j}\cdot\frac{\partial m_j}{\partial p_j}\right]\mathbf{u}\\&=-\left[\sum_{j}^{q\times l}\frac{\partial\mathsf{S}}{\partial m_j}\cdot a_j^{(i)}\right]\mathbf{u}\\&=-\sum_{j}^{q\times l}\frac{\partial\mathsf{S}}{\partial m_j}\,\mathbf{u}\cdot a_j^{(i)}\\&=\sum_{j}^{q\times l}\mathbf{g}^{(j)}a_j^{(i)}\end{aligned} \tag{A4}$$

(no implied summation), where we have introduced the virtual sources under the point collocation scheme,

$$\mathbf{g}^{(j)}=-\frac{\partial\mathsf{S}}{\partial m_j}\,\mathbf{u}\,, \tag{A5}$$

which are trivial to compute from the finite-difference formulation (or from the finite-element formulation). From eq. (A4) we may conclude that the virtual source matrix is given by

$$\mathsf{F}=\mathsf{GA}\,, \tag{A6}$$

where G is the virtual source matrix for the point collocation scheme, the columns of which comprise the $q\times l$ virtual source vectors $\mathbf{g}^{(i)}$ (one for each of q physical parameters at each of l nodes). Eqs (A4) and (A5) show that the virtual sources required for the scheme (A2) may be obtained by interpolating the virtual sources for the point collocation scheme.

From eqs (19) and (A6) we can generate expressions for the gradient vector,

$$\nabla_p E=\mathscr{Re}\{\mathsf{F}^{\mathrm{t}}\mathbf{v}\}=\mathscr{Re}\{\mathsf{A}^{\mathrm{t}}\mathsf{G}^{\mathrm{t}}\mathbf{v}\}=\mathsf{A}^{\mathrm{t}}\,\mathscr{Re}\{\nabla_m E\}\,, \tag{A7}$$

and for the approximate Hessian,

$$\mathsf{H}_a=\mathscr{Re}\{\mathsf{J}^{\mathrm{t}}\mathsf{J}^*\}=\mathsf{A}^{\mathrm{t}}\,\mathscr{Re}\{\mathsf{K}^{\mathrm{t}}\mathsf{K}^*\}\mathsf{A} \tag{A8}$$

[which is obvious on consideration of eqs (17 and (A6)], where K is the Frechét derivative matrix for the point collocation scheme. The equations show that the gradient vector and the approximate Hessian matrix can be obtained from the equivalent quantities for the point collocation scheme, using the matrix A^{t} as a projection matrix from the space of modelling parameters to the space of inversion parameters.

Relationship to other schemes

The relationships between the point collocation scheme and model parametrizations given by eqs (A2), (A7) and (A8) can be directly related to the subspace search method developed by Kennett *et al.* (1988). If we express the modelling-parameter updates at each iteration for the Gauss–Newton method by inserting eqs (A7) and (A8) into eq. (35) in the body of this paper, and using eq. (A2) to project onto the modelling parameters, we obtain

$$\delta\mathbf{m}=\mathsf{A}(\mathsf{A}^{\mathrm{t}}\mathscr{Re}\{\mathsf{K}^{\mathrm{t}}\mathsf{K}^*\}\mathsf{A}+\lambda\mathsf{I})^{-1}\mathsf{A}^{\mathrm{t}}\,\mathscr{Re}\{\mathsf{K}^{\mathrm{t}}\delta\mathbf{d}^*\}\,, \tag{A9}$$

GJI000 21/4/98 09:17:19 3B2 version 5.20
The Charlesworth Group, Huddersfield 01484 517077

which is identical to eq. (2.16) of Kennett *et al.* (1988), except that we are working with complex-valued data, and that we have again omitted a rigorous consideration of the role of the *a priori* model covariance matrix. We reiterate that the covariances have been omitted for reasons of clarity, and that these could be incorporated in a straightforward manner.

More recently, Bunks *et al.* (1995) have advocated a 'multiscale' approach to seismic-waveform inversion, in which the large-scale parameters of the problem are solved early in an iterative scheme, in order to attempt to obtain a 'reasonable degree of convergence to the neighbourhood of the global minimum [of the misfit function]'. They advocate decomposing the data by (temporal) scale (i.e. frequency) and decomposing the model by (spatial) scale (i.e. wavenumber). The long-scale, low-frequency components of the data are used in the initial iterations to locate the long-scale, large-wavenumber components of the model. Proceeding from low-frequency components of the data to the higher frequencies has also been advocated by Pratt & Goulty (1991) and Song *et al.* (1995). The strategy of proceeding from the large wavenumbers of the model to the small wavelengths is comparable to the approach taken by Williamson (1990).

The methods we have developed in this appendix are exactly what are required for a formal multiscale approach. The decomposition of the data by frequency is inherent to the FDI approach—it is entirely natural that we would attempt to use the low frequencies early in the iterative solution. A decomposition by wavenumber could be implemented by using a small number of basis vectors in the description of the inversion parameters, representing only the long wavenumbers of the model. In the language of the multigrid methods described by Bunks *et al.* (1995), we require a restriction operator that restricts a given model to its large-scale components, and an injection operator that interpolates a large-scale solution to smaller scale lengths. Eq. (A9) can be interpreted as the following set of operations:

(1) obtain the gradient for the point collocation scheme, $\nabla_m E$;

(2) use the projection matrix $\mathbf{A}^{\mathrm{t}}$ as a restriction operator to restrict the gradient $\nabla_m E$ to its large-scale components;

(3) use the restricted gradient and the approximate inverse Hessian as a 'relaxation operator' to find a model update at these large scale lengths;

(4) finally, project the large-scale solution onto the (small-scale) model space $\mathbf{m}$ using the projection matrix $\mathbf{A}$ as the injection operator.

APPENDIX B: COMPARISON WITH SNIEDER (1990)

In this appendix we compare the formula given in this paper for the non-linear term in the inverse Hessian with the methods developed by Snieder (1990) for incorporating non-linear effects into iterative non-linear inverse methods. For the purposes of this comparison we follow the steps used by Snieder, but we use the notation, and the matrix methods, developed in this paper, to obtain an explicit formula for the first-order multiple correction term.

Forward modelling is non-linear in the parameters $\mathbf{p}$. Thus, if we perturb the parameters by an amount $\varepsilon\delta\mathbf{p}$, the forward problem constitutes a non-linear mapping from the parameter perturbations to the data perturbation

$$\begin{aligned}\delta\mathbf{d}(\varepsilon) &= F(\varepsilon\delta\mathbf{p}) = \varepsilon\delta\mathbf{d}^{(1)} + \varepsilon^2\delta\mathbf{d}^{(2)} + \dots \\ &= \varepsilon F^{(1)}\delta\mathbf{p} + \varepsilon^2\,\delta\mathbf{p}^{\mathrm{t}}F^{(2)}\delta\mathbf{p} + \dots \end{aligned} \tag{B1}$$

(note that the vector $\delta\mathbf{d}$ used here is not defined in exactly the same manner as the same vector in the body of the paper, although the two quantities are analogous). In eq. (B1),

$$\delta\mathbf{d}^{(1)} = F^{(1)}\delta\mathbf{p} = \mathbf{J}\delta\mathbf{p} \tag{B2}$$

and

$$\delta\mathbf{d}^{(2)} = \delta\mathbf{p}^{\mathrm{t}}F^{(2)}\delta\mathbf{p}\,. \tag{B3}$$

The second data perturbation vector, $\delta\mathbf{d}^{(2)}$, is defined in terms of its components

$$\delta d_i^{(2)} = \delta\mathbf{p}^{\mathrm{t}}\mathbf{J}_i^{(2)}\delta\mathbf{p}\,, \tag{B4}$$

with

$$\mathbf{J}_i^{(2)} = \frac{\partial^2 u_i}{\partial\mathbf{p}^{\mathrm{t}}\partial\mathbf{p}}\,. \tag{B5}$$

In these equations, $\mathbf{J}$ is the Frechét derivative matrix introduced in eq. (11) and $\mathbf{J}_i^{(2)}$ is the matrix of second partial derivatives of the ith data component. The Taylor expansion of the forward problem above is commonly referred to as the Born series; the first term is the first Born approximation, the second term the second Born approximation and so on.

The inverse problem is a non-linear mapping from the data to the estimated parameters, $\delta\hat{\mathbf{p}}(\varepsilon)$:

$$\begin{aligned}\delta\hat{\mathbf{p}}(\varepsilon) &= I(\delta\mathbf{d}(\varepsilon)) = \delta\hat{\mathbf{p}}^{(1)} + \delta\hat{\mathbf{p}}^{(2)} + \dots \\ &= I^{(1)}\delta\mathbf{d} + \delta\mathbf{d}^{\mathrm{t}}I^{(2)}\delta\mathbf{d}^* + \dots\,, \end{aligned} \tag{B6}$$

in which the terms in the Taylor expansion involve the operators $I^{(1)}$, $I^{(2)}$, ..., yet to be determined.

If we wish the estimated parameters $\delta\hat{\mathbf{p}}(\varepsilon)$ to reproduce the ideal model, $\varepsilon\delta\mathbf{p}$, as accurately as possible, then we may assume the equality $\delta\hat{\mathbf{p}}(\varepsilon)\rightarrow\varepsilon\delta\mathbf{p}$ (see Snieder 1990 for a more rigorous discussion of why this approach is valid). Following this, we insert eq. (B1) into eq. (B6) and equate equal powers of ε to obtain

$$\delta\mathbf{p} = I^{(1)}\mathbf{J}\delta\mathbf{p} \tag{B7}$$

$$0 = I^{(1)}\,\delta\mathbf{p}^{\mathrm{t}}F^{(2)}\delta\mathbf{p} + \delta\mathbf{p}^{\mathrm{t}}\mathbf{J}^{\mathrm{t}}I^{(2)}\mathbf{J}^*\delta\mathbf{p}^* \tag{B8}$$

$$\vdots$$

Eq. (B7) is satisfied when $I^{(1)}$ is the standard (Gauss–Newton) least-squares estimator for the linearized problem, that is when

$$\delta\hat{\mathbf{p}}^{(1)} = I^{(1)}\delta\mathbf{d} = \mathbf{H}_a^{-1}\,\mathcal{R}e\{\mathbf{J}^{\mathrm{t}}\delta\mathbf{d}^*\}\,, \tag{B9}$$

in which

$$\mathbf{H}_a = \mathcal{R}e\{\mathbf{J}^{\mathrm{t}}\mathbf{J}^*\} \tag{B10}$$

is the approximate Hessian (i.e. the first term in eq. 32).

Eq. (B8), obtained by equating terms in ε^2, can be rewritten using (1) the substitution from the first Born approximation, $\varepsilon\mathbf{J}\delta\mathbf{p}\approx\delta\mathbf{d}$, and (2) the first term in the inverse problem expansion, $\varepsilon\delta\mathbf{p}\approx\delta\hat{\mathbf{p}}^{(1)}$:

$$\delta\hat{\mathbf{p}}^{(2)}=\delta\mathbf{d}^{\mathrm{t}}\,I^{(2)}\delta\mathbf{d}=-I^{(1)}\,\delta\hat{\mathbf{p}}^{(1)^{\mathrm{t}}}F^{(2)}\delta\hat{\mathbf{p}}^{(1)}\,, \tag{B11}$$

in which the left-hand side is the next term in the inverse series, which we can now compute, given the operator $F^{(2)}$ defined in eqs (B3) and (B4).

Substituting eq. (B11) into the inverse problem expansion (B6), we obtain an explicit expression for the first two terms of the non-linear inverse mapping:

$$\begin{aligned}\delta\hat{\mathbf{p}}&=\delta\hat{\mathbf{p}}^{(1)}-\mathbf{H}_a^{-1}\,\delta\hat{\mathbf{p}}^{(1)^{\mathrm{t}}}F^{(2)}\delta\hat{\mathbf{p}}^{(1)}\\&=\delta\hat{\mathbf{p}}^{(1)}-\mathbf{H}_a^{-1}\,\mathscr{Re}\{\mathbf{J}^{\mathrm{t}}\delta\hat{\mathbf{d}}^{*(2)}\}\\&=\delta\hat{\mathbf{p}}^{(1)}-\mathbf{H}_a^{-1}\,\mathscr{Re}\{\mathbf{J}^{\mathrm{T}}\delta\hat{\mathbf{d}}^{(2)}\}\end{aligned} \tag{B12}$$

[the superscript T represents the Hermitian (conjugate) transpose, and we are thus conjugating both terms inside the brackets], where the ith component of the vector $\delta\hat{\mathbf{d}}^{(2)}$ is given by

$$\delta\hat{d}_i^{(2)}=\delta\hat{\mathbf{p}}^{(1)^{\mathrm{t}}}\mathbf{J}_i^{(2)}\delta\hat{\mathbf{p}}^{(1)}\,. \tag{B13}$$

The linearized inverse $\delta\hat{\mathbf{p}}^{(1)}$ is defined in eq. (B9). The second term in eq. (B12) corrects the linearized inverse by subtracting the estimated non-linear components from the inverse. The second-order algorithm for non-linear FDI thus proceeds as follows:

(1) form the first-order (linearized) inverse $\delta\hat{\mathbf{p}}^{(1)}$ using eq. (B9);

(2) compute the second (forward) Born approximation from the linearized inverse to estimate the first-order multiples using eq. (B13);

(3) using eq. (B12), apply the first-order inverse operator to the estimated multiples, and finally subtract the result from $\delta\hat{\mathbf{p}}^{(1)}$.

At no time need the forward model be updated—all quantities are computed in the original background model.

It is interesting to try to manipulate the full Newton inversion formula (see eq. 41 in the body of the paper) to obtain a form similar to eq. (B12):

$$\begin{aligned}&[\mathbf{H}_a+\mathbf{R}]\delta\mathbf{p}=\mathscr{Re}\{\mathbf{J}^{\mathrm{t}}\delta\mathbf{d}^*\}\\&[\mathbf{I}+\mathbf{H}_a^{-1}\mathbf{R}]\delta\mathbf{p}=\mathbf{H}_a^{-1}\,\mathscr{Re}\{\mathbf{J}^{\mathrm{t}}\delta\mathbf{d}^*\}=\delta\hat{\mathbf{p}}^{(1)}\\&\delta\mathbf{p}=[\mathbf{I}+\mathbf{H}_a^{-1}\mathbf{R}]^{-1}\,\delta\hat{\mathbf{p}}^{(1)}\\&\delta\mathbf{p}=[\mathbf{I}-\mathbf{H}_a^{-1}\mathbf{R}+\{\mathbf{H}_a^{-1}\mathbf{R}\}^2-\ldots]\delta\hat{\mathbf{p}}^{(1)}\,.\end{aligned} \tag{B14}$$

If we retain only the first two terms in the series expansion and substitute the expression for $\mathbf{R}$ given in eq. (34) in the body of the paper, we obtain an approximate version of the full Newton algorithm:

$$\delta\hat{\mathbf{p}}=\delta\hat{\mathbf{p}}_1-\mathbf{H}_a^{-1}\,\mathscr{Re}\left\{\left(\frac{\partial}{\partial\mathbf{p}^{\mathrm{t}}}\,\mathbf{J}^{\mathrm{t}}\right)(\delta\mathbf{d}^*\quad\delta\mathbf{d}^*\quad\ldots\quad\delta\mathbf{d}^*)\,\delta\hat{\mathbf{p}}^{(1)}\right\}. \tag{B15}$$

If eq. (B15) is expressed in terms of the components of the resulting parameter estimates, $\delta\hat{p}_i$, we obtain the following expression for the second-order corrections to the parameter estimates (making use of the implied summation convention):

$$\delta\hat{p}_i^{(2)}=[-\mathbf{H}_a^{-1}]_{ik}\,\mathscr{Re}\left\{J_{lm}^*\delta\hat{p}_m^{(1)}\,\frac{\partial^2u_l}{\partial p_k\partial p_j}\,\delta\hat{p}_j^{(1)}\right\}\quad\text{(Newton)} \tag{B16}$$

(where we have substituted the predicted data from the linearized inversion, $\delta\mathbf{d}\approx\mathbf{J}\,\delta\hat{\mathbf{p}}^{(1)}$, into eq. (B15). This form is remarkably similar to the expression we obtain from eq. (B12), Snieder's algorithm:

$$\delta\hat{p}_i^{(2)}=[-\mathbf{H}_a^{-1}]_{ik}\,\mathscr{Re}\left\{J_{kl}^*\delta\hat{p}_m^{(1)}\,\frac{\partial^2u_l}{\partial p_m\partial p_j}\,\delta\hat{p}_j^{(1)}\right\}\quad\text{(Snieder)} \tag{B17}$$

(again making use of the implied summation convention). The two approaches, eqs (B16) and (B17), contain exactly the same terms, but differ in the order of summation applied.

The essential similarities and differences between the two approaches are summarized as follows.

(1) Both equations use the approximate inverse Hessian $\mathbf{H}_a^{-1}$ to refocus the correction term.

(2) Eq. (B16) predicts the correction by correlating the second-order partial derivatives with the (true) data residuals (one correlation is obtained from each possible pair of nodes in the model). For each node in the image, the correlations for this node with all others are multiplied with the linearized parameter estimates and summed over all nodes.

(3) Eq. (B17) seeks the same corrections by first predicting only the non-linear effects in the data, and then using the linearized inverse to estimate the effects on the model estimates.

(4) Eq. (B16) contains only the first two terms in the inverse series for the Hessian matrix. We thus expect the full inverse, as described in the body of the paper to be a more effective operator than the one embodied in eq. (B16). In contrast, in developing eq. (B17) the higher-order terms were explicitly neglected.

(5) Computing the second-order partial derivatives in eq. (B17) explicitly would require m^2 computations. However, we can use a simple forward-propagation algorithm to compute the action of this operator so that the vector $\delta\hat{\mathbf{d}}^{(2)}$ in eq. (B13) can be computed using

$$\delta\hat{\mathbf{d}}^{(2)}=-\mathbf{S}^{-1}\,\mathbf{f}^{(ij)}\delta p_i\delta p_j \tag{B18}$$

(summation is implied over the indices i and j), where $\mathbf{f}^{(ij)}$ are the second-order virtual sources defined in eq. (47). As in the computation of $\mathbf{R}$ by backpropagation, this computation also requires only m additional forward problems to be solved (still in the same model) to obtain the second-order virtual sources.

GJI000 21/4/98 09:18:10 3B2 version 5.20
The Charlesworth Group, Huddersfield 01484 517077