Inverse Problems **13** (1997) 63–77. Printed in the UK PII: S0266-5611(97)74070-X

Joint inversion: a structural approach

E Haber and D Oldenburg

UBC-Geophysical Inversion Facility, Department of Geophysics, University of British Columbia, Vancouver, Canada V6T 1Z4

Received 24 April 1996, in final form 16 September 1996

Abstract. We develop a methodology to invert two different data sets with the assumption that the underlying models have a common structure. Structure is defined in terms of absolute value of curvature of the model and two models are said to have common structure if the changes occur at the same physical locations. The joint inversion is solved by defining an objective function which quantifies the difference in structure between two models, and then minimizing this objective function subject to satisfying the data constraints. The problem is nonlinear and is solved iteratively using Krylov space techniques. Testing the algorithm on synthetic data sets shows that the joint inversion is superior to individual inversions. In an application to field data we show that the data sets are consistent with models that are quite similar.

1. Introduction

The need to obtain more detailed information about the earth, the human body, or other physical systems has been the impetus for carrying out different experiments on the same object. In geophysics it is common to collect magnetic, gravity and airborne electromagnetic data in reconnaissance surveys from mineral exploration. In medicine, MRI, PET and CT scans might be performed in attempting the diagnosis. These multiple data sets are important because extra information reduces the ambiguity or non-uniqueness of the interpretation. To be effective, however, we need algorithms which can jointly invert different types of data.

Consider two physical experiments denoted generically by $\mathcal{F}_1[m_1] = d_1$ and $\mathcal{F}_2[m_2] = d_2$. $\mathcal{F}_1$ and $\mathcal{F}_2$ are linear or nonlinear operators representing the physical experiment, m_1, m_2 are physical property models, and d_1, d_2 are data. By 'joint' inversion we mean the ability to simultaneously invert the combined data set (d_1, d_2) and to recover the models (m_1, m_2). This can be carried out using traditional methods if there is a known relationship between the two properties, that is, if $m_2 = g(m_1)$ and g is known. In many environments an explicit functional relationship does not exist but it is noted that the physical properties tend to change at the same location. The relative change, or even the sign of change is not known *a priori* and also the nature of this relative change may vary throughout the model domain. Nevertheless, there is a spatial coincidence of boundaries or transition zones which links the two models. In effect, the two models can be considered to have the same 'structure' and this allows the data sets to be jointly inverted.

Our paper begins by defining a structure operator that is based upon the curvature of the model. A semi-norm that measures the difference in structure is introduced and joint inversion is carried out by minimizing that penalty function subject to adequately fitting the data. The practical solution is formulated in a discrete domain and formally a two-stage algorithm is needed to iteratively solve the problem. We present these algorithms and then

0266-5611/97/010063+15$19.50

introduce a third algorithm which solves the problem using a truncated conjugate gradient solution. The paper concludes with a synthetic example in which cross-well seismic and surface gravity data are jointly inverted, and a field data example in which radio imaging (RIM) data at two different frequencies are inverted.

2. The structure operator

For a bounded function $m \in \mathcal{C}^2$ in one, two or three dimensions, there are numerous definitions for the term 'structure'. Generally, we identify structure by a change of the model with position and hence operators which make use of local gradients or curvature of the model are useful. Let $\mathcal{S}$ denote the structure operator. Desirable properties for $\mathcal{S}$ are that: (1) it should be positive, since positive and negative changes in the model are equal indicators of structure, (2) it should map any model into the range [0, 1] and thus be insensitive to overall scaling of the model, and (3) it should map a 'simple' function into a 'simple' function. This latter property is important because we want to minimize an objective function based upon structure. If the structure function is too rough then the likelihood of becoming trapped in a local minimum is greatly enhanced. We have investigated operators based upon ∇m and $\nabla^2 m$. The fact that ∇m is a vector ultimately leads to greater computational difficulties. The computations are somewhat simplified by using curvature and, as a first stage, we consider the structure operator

$$\mathcal{S}[m] = \begin{cases} 0 & |\nabla^2 m| < \tau \\ 1 & |\nabla^2 m| > \tau. \end{cases} \tag{1}$$

In equation (1) τ is a threshold parameter. If the curvature of the model is less than τ then the structure is defined to be zero, while if it is greater than τ, then the structure is equal to unity. Different choices for τ will yield different structure functions. The choice of τ is therefore crucial and specifying its numerical value requires knowledge of how the physical property might vary and a decision on the part of the interpreter regarding what curvatures of the model are considered to be important.

The definition in equation (1) satisfies many objectives for a structure operator but it suffers from its discontinuous behaviour with respect to τ. Let ϵ denote a small quantity and consider two locations in the model domain where the curvatures are respectively $\tau - \epsilon$ and $\tau + \epsilon$. It is not desirable that the first of these positions is characterized as having no structure, while the second position is said to have as much structure as locations with maximum curvature. Some transitional weighting is desirable. The second reason for needing a smooth operator is technical. We will solve our problem by linearizing an objective function based upon structure and then iterating. Second derivatives of $\mathcal{S}[m]$ need to be calculated and therefore an operator which is twice Frechet differentiable is required.

Smoothness of $\mathcal{S}$, and allowing transitional values between zero and unity can be achieved by introducing two parameters, τ_1 and τ_2, and defining the structure to be:

$$\mathcal{S}[m] = \begin{cases} 0 & |\nabla^2 m| < \tau_1 \\ P_5(|\nabla^2 m|) & \tau_1 < |\nabla^2 m| < \tau_2 \\ 1 & \tau_2 < |\nabla^2 m|. \end{cases} \tag{2}$$

In equation (2) P_5 is a polynomial in one dimension with degree five which makes the structure operator twice Frechet differentiable. The coefficients of P_5 are chosen such that $P_5(\tau_1) = P_5'(\tau_1) = P_5''(\tau_1) = P_5'(\tau_2) = P_5''(\tau_2) = 0$ and $P_5(\tau_2) = 1$. Since P_5 is a simple one-dimensional polynomial the six conditions on the polynomial and its derivatives at

points τ_1 and τ_2 determine its coefficients uniquely. We note that the Frechet differentiable operator should be written as $\mathcal{S}_{\tau_1,\tau_2}$, but for convenience we refer to it only as $\mathcal{S}$.

To summarize, here are some of the characteristics of the structure operator which we use later.

- The operator maps any function to the interval [0,1].
- The operator has continuous second derivatives.
- The first and second derivatives are zero almost everywhere. They are not zero when $|\nabla^2 m|$ lies in the interval (τ_1, τ_2).
- The decision about what is (and what is not) structure is done by choosing τ_1 and τ_2.
- The structure operator is not invertible.

Applying the structure operator is a two-step process. First the Laplacian, which is a linear operator, is taken. Second, the thresholding and evaluation of $P_5(|\nabla^2 m|)$ is carried out. This is a nonlinear operator. The structure operator is defined only for functions which are in $\mathcal{C}^2$. Discontinuous functions are first approximated using a smooth spline function and we apply our operator to the smooth approximated function.

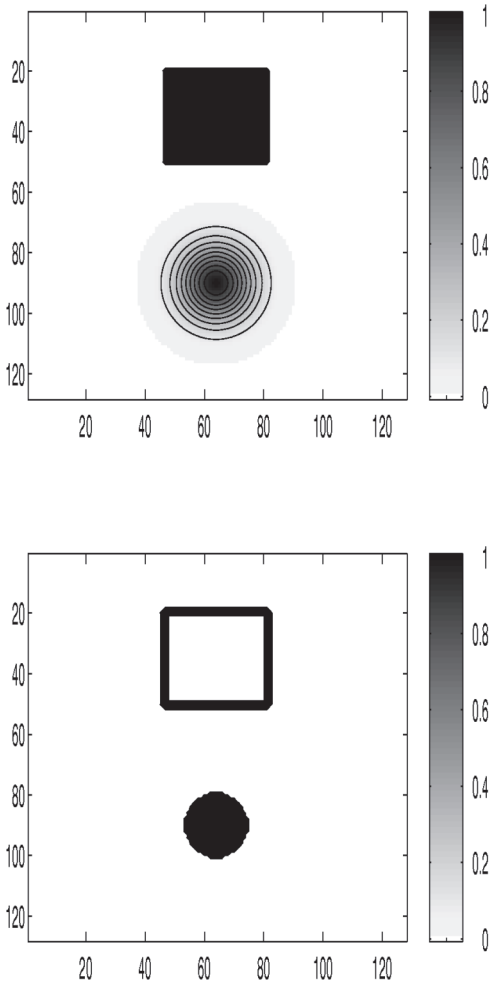

Figure 1. A model made up of a Gaussian and a prism is shown at the top. Application of the structure operator yields the image shown by the bottom panel.

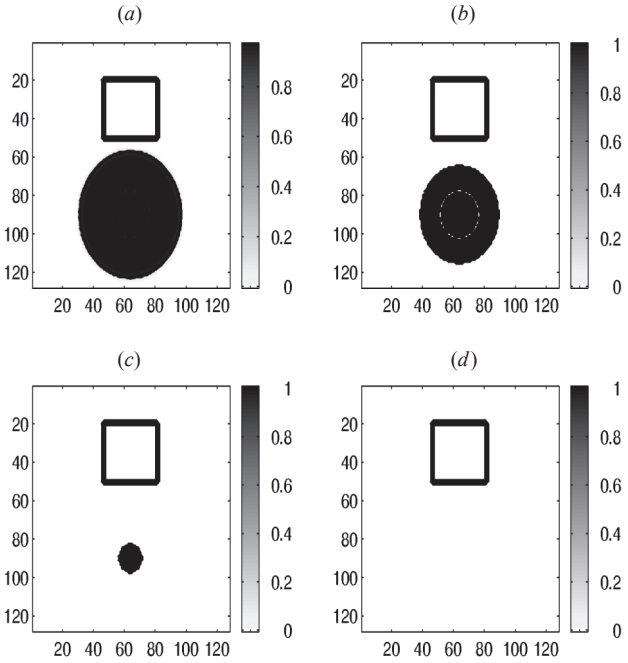

Figure 2. The result of applying the structure operator on the model in figure 1. The value of τ_1 is successively increased for panels (*a*)–(*d*).

The result of applying the structure operator to a model which has smooth and discontinuous components is given in figure 1. Choosing $\tau_1 = \tau_2 = 10^{-5}$ produces the image in figure 1. The Gaussian feature at the bottom is transformed to a circular region of unit amplitude and the prism is manifested only by its edges. The dependence of the structure on the parameters τ_1 and τ_2 is shown in figure 2. In figure 2(*a*), τ_1 is small and the Gaussian appears as two circular features. As τ_1 increases, the outer feature disappears and the diameter of the inner circle is progressively reduced until it disappears in figure 2(*d*).

3. Methodology for joint inversion

In this section we develop the inversion methodology and algorithms for carrying out the joint inversion. We suppose that two experiments have been carried out and the relationships between the models and data are given by

$$\mathcal{F}_i^{(1)}[m_1] = d_i^{(1)} \qquad i = 1 \dots M_1 \qquad \text{and} \qquad \mathcal{F}_i^{(2)}[m_2] = d_i^{(2)} \qquad i = 1 \dots M_2. \tag{3}$$

The data $d^{(1)}$ and $d^{(2)}$ and estimates of their errors, are assumed known. The models m_1 and $m_2 \in \mathcal{D}$ are bounded models in a Hilbert space $\mathcal{D}$ and are to be reconstructed. $\mathcal{F}^{(1)}$ and $\mathcal{F}^{(2)}$ are linear or nonlinear operators which are assumed to be Frechet differentiable.

Our goal is to jointly invert the data $[d_1, d_2]$ and to reconstruct the models $[m_1, m_2]$ under the assumption that they have similar structure. A measure of the difference in structure between the two models can be defined as

$$\phi_c = \int_{\mathcal{D}} (\mathcal{S}[m_1] - \mathcal{S}[m_2])^2 \, \mathrm{d}V. \tag{4}$$

We need to find m_1 and m_2 that make ϕ_c as small as possible and also satisfy equations (3) to a degree that is justified by the errors in the data. The inverse problem becomes a minimization problem which is formulated as:

Minimize ϕ_c subject to:

$$\phi_d = \|\mathcal{F}^{(1)}[m_1] - d_1\|^2 + \|\mathcal{F}^{(2)}[m_2] - d_2\|^2 = \phi_d^*. \tag{5}$$

In equation (5) ϕ_d^* is a target misfit and its value depends upon the errors ascribed to the data. If the data are accurate then ϕ_d^*=0. In this paper we generally assume that errors are Gaussian, independent and have unit standard deviation. With this assumption ϕ_d is a χ^2 variable with M degrees of freedom and its expected value is equal to M. Setting $\phi_d^* = M$ provides a reasonable target.

To solve the above minimization problem numerically we divide the model domain into N cells. The data equations become:

$$\mathcal{F}_i[m] = d_i \qquad i = 1 \ldots M, \;\; M = M_1 + M_2 \tag{6}$$

where $m = [m_1, m_2] \in R^{2N}$ is a vector of length $2N$. The misfit is written as

$$\phi_d = \sum_{i=1}^{M} \|\mathcal{F}_i[m] - d_i\|^2. \tag{7}$$

Finding the structure of a discrete model is done by first applying the discrete Laplacian, and then applying the threshold operator. Each cell is assigned a structural value and the structural measure in equation (4) is rewritten as

$$\phi_c = \sum_{i=1}^{N} (\mathcal{S}[m_1^{(i)}] - \mathcal{S}[m_2^{(i)}])^2. \tag{8}$$

Our inverse problem is solved by finding $m = [m_1, m_2]$ such that equation (8) is minimized subject to the constraint $\phi_d = \phi_d^*$.

3.1. Solving the problem

In order to solve the problem we define a penalty parameter μ and combine the data in equations (7) and the structural measure (8) into one objective function ϕ:

$$\phi = \phi_c + \mu\phi_d. \tag{9}$$

In general, the parameter μ controls the trade-off between ϕ_c and ϕ_d. For any given value of μ we want to minimize ϕ. As μ increases, ϕ_c increases while ϕ_d decreases. We adjust μ so that when minimization is complete we will also have satisfied the constraint $\phi_d = \phi_d^*$.

The structural measure in equation (8), and possibly the data equations (7), are nonlinear. We solve the problem iteratively. At every iteration we linearize the data constraints and the structure operator, and solve a linear set of equations. Let m denote the current model and let δm be a model perturbation. The perturbed data functionals (when keeping only the first-order terms) are

$$\mathcal{F}_i[m + \delta m] = \mathcal{F}_i[m] + \sum_j \frac{\partial \mathcal{F}_i}{\partial m_j} \delta m_j \tag{10}$$

where the sensitivity $\frac{\partial \mathcal{F}_i}{\partial m_j}$ is calculated at m. Setting the left-hand side of equation (10) equal to the observations, and letting $G_{ij}(m) = \frac{\partial \mathcal{F}_i}{\partial m_j}$, yields

$$\mathcal{F}_i[m] + \sum_j G_{ij}(m)\delta m_j = d_i \tag{11}$$

or

$$\sum_j G_{ij}(m)\delta m_j = b_i$$

where $b_i = d_i - \mathcal{F}_i[m]$. The linearized misfit corresponding to equation (7) becomes

$$\phi_d^{\text{lin}} = \|G\delta m - b\|^2. \tag{12}$$

Because of the neglect of higher-order terms in equation (10), we note that this linearized misfit is not, in general, equal to the true misfit ϕ_d evaluated by equation (7).

Linearizing ϕ_c at the model m yields

$$\phi_c^{\text{lin}}(m + \delta m) = \phi_c(m) + (\nabla_m \phi_c)^T \delta m + \tfrac{1}{2}\delta m^T H \delta m \tag{13}$$

where

$$H = \frac{\partial^2 \phi_c}{\partial m_i \partial m_j}$$

is the Hessian of the model objective function.

We proceed by substituting the linearized functions into (9) and minimizing the linearized quadratic functional

$$\min \phi^{\text{lin}}(\delta m, \mu) = \phi_c^{\text{lin}} + \frac{\mu}{2}\|G\delta m - b\|^2 = \phi_c^{\text{lin}} + \frac{\mu}{2}\phi_d{}^{\text{lin}}. \tag{14}$$

A major goal in the computations is to find the correct value of μ so that $\phi_d = \phi_d^*$. Because we are working with an approximation of the data equations and the structure operator, we do not expect that ϕ_d equals ϕ_d^{lin} or that ϕ_c equals ϕ_c^{lin} as δm becomes large. This difference between the linearized problem, which we can solve, and the nonlinear problem, which we want to solve, leads us to restrict the size of δm at each iteration. We appeal to the trust region formulation (Dennis and Schnabel 1983) which limits δm to lie in a region where the quadratic approximation holds for approximating the nonlinear function. This trust region can be evaluated by checking the linear approximation against the nonlinear function. We evaluate:

$$T = \frac{|\phi - \phi^{\text{lin}}|}{\phi}$$

where ϕ^{lin} defined above is the linearized objective function and ϕ is its nonlinear counterpart. If T is sufficiently small then we are inside the trust region.

At each iteration we attempt to find that δm and μ which brings ϕ_d as close to ϕ_d^* as possible while still restricting the perturbation by the trust region constraint. This is accomplished by including an additional penalty term in the objective function. Let W be a nonsingular weighting matrix. The modified objective function becomes

$$\phi(\delta m, \mu, \xi) = \phi_c + \frac{1}{2}\xi\|W\delta m\|^2 + \frac{\mu}{2}\phi_d \tag{15}$$

and its linear counterpart:

$$\phi^{\text{lin}}(\delta m, \mu, \xi) = \phi_c^{\text{lin}} + \frac{1}{2}\xi\|W\delta m\|^2 + \frac{\mu}{2}\phi_d^{\text{lin}}.$$

The parameter ξ is adjusted to keep the step size sufficiently small. Taking the derivative of equation (15) with respect to δm and setting the resultant equal to zero gives

$$(H + \xi W^T W + \mu G^T G)\delta m = -\nabla_m(\phi_c) + \mu G^T b. \tag{16}$$

This is a linear system which we write as

$$J_{\xi,\mu}\delta m = d. \tag{17}$$

The matrix $J_{\xi,\mu}$ is positive definite and symmetric. This system is solved and the model is updated. The iterations continue until the misfit equals the desired target ϕ_d^*. The practical difficulties centre around finding the sequence of ξ, μ such that good convergence is obtained. This problem is addressed next.

3.2. Algorithms for solving the minimization problem

At every iteration one needs to choose a parameter ξ which controls the step size, and a regression parameter μ. The choice of these two parameters is not simple and correspondingly we present a two-step solution. We first design an algorithm to choose ξ when μ is specified. We rely heavily on trust region theory as presented by Dennis and Schnabel (1983). Next, we present an algorithm for selecting μ. If computing power were not an issue, solution of the minimization problem could be obtained by using algorithms 1 and 2. However, the computational overhead using algorithm 1 is high. Therefore, we have developed a modified and more efficient approach, using a Krylov space technique that leads to algorithm 3.

3.2.1. Choice of ξ. Assume for this section that μ is fixed. The primary role of ξ is to limit the step size so that the quadratic approximation is valid. However, we do not want to choose a step which is too small because that reduces the convergence rate. We want to choose ξ to 'stretch' the quadratic approximation to its full capacity. This is done by the following.

Algorithm 1.

1. Choose $a_1, a_2 \in (0, 1)$.
2. At iteration n calculate: $\phi(m_n) = \phi_c(m_n) + \frac{\mu}{2}\phi_d(m_n)$.
3. Choose ξ.
4. Calculate δm by solving equation (16).
5. Evaluate $m_{n+1}^{\text{trial}} = m_n + \delta m$.
6. $$\text{test}_1 = \phi(m_{n+1}^{\text{trial}})/\phi(m_n)$$ $$\text{test}_2 = \frac{|\phi(m_{n+1}^{\text{trial}} - \phi^{\text{lin}}|}{\phi(m_{n+1}^{\text{trial}})}.$$
7. • If $\text{test}_1 > 1$ narrow the trust region. Set $\xi \to 2\xi$. Go to 4.
   • If $\text{test}_1 < 1$ and $\text{test}_2 > a_2$ narrow the trust region. Set $\xi \to 2\xi$. Go to 4.
   • If $\text{test}_1 < 1$ and $\text{test}_2 < a_1$ extend the trust region, set $\xi \to \xi/2$. Go to 4.
   • If $\text{test}_1 < 1$ and $a_1 < \text{test}_2 < a_2$, accept δm.
8. $m_{n+1} = m_n + \delta m$; save ξ as an initial guess for the next iteration.

In our algorithm we chose $a_1 = 0.1$; $a_2 = 0.9$. The algorithm decreases the value of ϕ in every iteration for a constant μ. It stops when ϕ no longer decreases.

3.2.2. Choice of μ. While optimization literature extensively treats well-posed nonlinear problems it lacks treatment of ill-posed nonlinear problems. The most common approach to

solving ill-posed nonlinear problems is to employ Tichonov's regularization. The nonlinear optimization problem: minimize $\phi_m = \|m\|^2$ subject to: $\mathcal{F}[m] = d$ is solved by first forming an objective function: $\phi_T = \phi_m + \mu\phi_d$. For fixed μ this objective function is minimized with respect to m. The value of μ is altered until a desired misfit is achieved. The way to choose μ, even when $\mathcal{F}$ is linear, is complicated and according to Hanke and Hansen (1993) there is no black box to find it. For nonlinear problems finding μ is even more complicated, and although this problem has been approached by some, there is no specific algorithm which works for general problems. The difficulty is exacerbated in our problem because our structure operator is highly nonlinear.

We first assume that a minimum of ϕ exists for every μ. Second, as in Tichonov regularization, it is clear that as $\mu \to 0$ we minimize mainly ϕ_c, and thus ϕ_d is large. As $\mu \to \infty$ we minimize mainly ϕ_d and thus ϕ_c becomes large. We cannot prove that ϕ_d is a monotonic increasing function of μ, as it is in the Tichonov case for linear problems, but we assume that this is the case here. We can then use a Newton-type algorithm to iteratively find the right μ. The algorithm goes as follows:

Algorithm 2.

1. Choose μ.
2. Apply algorithm 1 to obtain the minimum of $\phi = \phi_c + \mu\phi_d$.
3. When a solution is found
- if $\phi_d > \phi_d^*$ decrease μ and go to 2;
- if $\phi_d < \phi_d^*$ increase μ and go to 4.
4. Interpolate to find μ.

For point (3) in the above algorithm, an estimate for the next value of μ can be facilitated by keeping track of previous values of μ and ϕ.

3.2.3. Krylov space techniques. The major computational effort in implementing algorithm 1 involves solving the $M \times M$ system of equations (15). Although μ is known, ξ is not and the matrix system needs to be solved many times. An alternate, albeit approximate, approach is to set $\xi = 0$ so that the equations are

$$(H + \mu G^T G)\delta m = -\nabla_m(\phi_c) + \mu G^T b. \tag{18}$$

Equation (17) is then solved with an iterative solver which is terminated after a few iterations. This approach is not new. Hanke and Hansen (1993) used Krylov space techniques (conjugate gradient, and least squares conjugate gradient) and effectively demonstrated that the number of iterations can be used as an effective regularizer. Brown and Saad (1990, 1994) have also used a similar Krylov space method for the solution of a nonlinear problem. The implementation of this approach is immediate if $W = I$, the identity matrix. However, if $W \neq I$ then the problem needs to be transferred to a standard form. Since W is sparse this is efficiently done using a QR transformation (Hanke and Hansen 1993). In our work, the matrix is symmetric and positive semi-definite, and we use the conjugate gradient solver. The replacement for algorithm 1 is:

Algorithm 3.

1. Choose $a_1, a_2 \in (0, 1)$.
2. At iteration n calculate $\phi(m_n) = \phi_c(m_n) + \mu\phi_d(m_n)$.
3. Start the CG iteration to solve equation (16) with $\xi = 0$. For each iteration:
4. Calculate δm_k (the δm that is achieved by stopping the CG solution at iteration k) and store δm_k.

5. Evaluate $m_{n+1}^{\text{trial}} = m_n + \delta m_k$.
6.

$$\text{test}_1 = \phi(m_{n+1}^{\text{trial}})/\phi(m_n)$$
$$\text{test}_2 = \frac{|\phi(m_{n+1}^{\text{trial}}) - \phi^{\text{lin}}(m_{n+1}^{\text{trial}})|}{\phi(m_{n+1}^{\text{trial}})}.$$

7.

• If $\text{test}_1 > 1$ narrow the trust region (iterate less in CG). Choose a solution from the previous iteration.

• If $\text{test}_1 < 1$ and $\text{test}_2 > a_2$ narrow the trust region (iterate less in CG). Choose a solution from the previous iteration.

• If $\text{test}_1 < 1$ and $\text{test}_2 < a_1$ extend the trust region. Continue the CG iterations. Go to 4.

• If $\text{test}_1 < 1$ and $a_1 < \text{test}_2 < a_2$, accept δm_k, ($\delta m = \delta m_k$).

8. $m_{n+1} = m_n + \delta m$.

This algorithm achieves approximately the same solution as algorithm 1 but it does so with far fewer computations. The main reason is that in algorithm 1 we need to invert a matrix for every ξ, while using the Krylov space technique we do not invert the matrix even once.

3.3. Numerical implementation of joint inversion

Although we have provided algorithms for carrying joint inversion, there are still some numerical difficulties when trying to apply them to a new problem. In this section we try to give some general guidelines for the numerical implementation of the joint inversion.

There are three key aspects to the practical implementation. First, the structure operator is highly nonlinear. We have found that it is difficult to decrease the value of ϕ_c. Therefore, it is recommended to start the minimization from two models which have a similar structure, preferably models for which $\phi_c = 0$. Half spaces are good starting models. Then allow the algorithm to reduce the data misfit slowly while the structural similarity increases. Starting the algorithm with models obtained by inverting each data individually will generally fail since the models are not similar but yet $\phi_d = \phi_d^*$.

The second aspect is the fact that the derivatives of the structure operator are zero over portions of the model domain. Non-zero values occur only in the transition zones where $\tau_1 < |\nabla^2 m| < \tau_2$. The Hessian will be zero if $|\nabla^2 m|$ lies outside these bounds and, at those locations, the structure operator has no effect on the inversion. It is, therefore, extremely important to estimate $[\tau_1, \tau_2]$ correctly. If τ_1 is too large, the structure operator will not be effective in the inversion until considerable structure has been built up. It may then be difficult to reduce this structure in such a way that the two models evolve toward the same structure. Conversely, if τ_1 is very small and τ_2 is large then the entire model is regarded as having structure. This will allow models to have substantial curvature everywhere on the model domain and it is likely that the final models would not be similar.

The third aspect of the joint inversion is the fact that it is easier to carry out the numerical computations by working with the Laplacian of the model. This requires the introduction of an invertible transformation between the original and new model. We use the operator $\nabla^2 + \epsilon I$ where ϵ is a small quantity. The advantages of working with transformed variables is that the structure operator is now defined only by the analytic expression which determines the thresholding. The gradient and Hessian of ϕ_c can be evaluated directly for any given

model. The transformed variables are also an advantage when working with Krylov space techniques since the CG algorithm is defined with weighting function $W = I$. When $W \neq I$, the problem first needs to be transformed into standard form (Hanke and Hansen 1993). If we work with the Laplacian then this transformation is not needed.

4. Examples

To demonstrate the joint inversion methodology we give two examples. The first is a synthetic example in which we invert data from seismic tomography and gravity surveys. The second is a joint inversion of RIM tomography data acquired at two different frequencies.

4.1. Joint inversion of gravity and seismic tomography data

Seismic velocities depend upon density and elastic constants. Although velocity usually increases with density, the relationship is dependent upon the geologic environment and hence is generally unknown. Nevertheless, we expect that velocity and density would be good candidates for joint inversion.

As a synthetic example we consider a two-dimensional earth which has the anomalous velocity structure given in figure 3. The two positive anomalies have the same maximum amplitude and they are Gaussian shaped. The density anomalies, shown in figure 4, are also Gaussian shaped but the near surface anomaly is negative while the deeper anomaly is positive. The locations of the extrema of velocity coincide with those of density.

We first carry out a seismic tomography experiment. Twenty sources are equally spaced in one well and first arrivals from each source are measured at 20 receivers in the second well. The velocity of the medium is $c(x, z)$ and the travel time is

$$t_i = \int_{l_i} \frac{\mathrm{d}l}{c(x, z)} = \int_{l_i} s(x, z)\, \mathrm{d}l \tag{19}$$

where l_i is the ray path i, and $s(x, z) = 1/c(x, z)$ is the slowness.

Rather than using travel times directly, we first remove the effects of a uniform background velocity c_0 and thereby work with travel-time perturbations Δt_i. These are linearly related to the slowness perturbation $\Delta s(x, z)$ according to

$$\Delta t_i = \int_{l_i} \Delta s(x, z)\, \mathrm{d}l. \tag{20}$$

In reality, the ray path joining any source and receiver is curved because of refraction. However, for purposes of illustrating joint inversion there is no loss of generality in assuming straight ray paths, and we do so here. To carry out numerical computations we discretize the model into 600 cells. Equation (20) becomes a linear system of the form $\Delta t = G_1 \Delta s$ where G_1 is a sparse matrix. The data, contaminated by 5% random Gaussian noise, are shown in figure 4(*b*).

A surface gravity survey is carried out to measure the anomalous gravitational acceleration in the vertical direction. Letting $\Delta\rho(x, z)$ denote the anomalous density per unit length, then the gravitational data are given by

$$\Delta g_i = \Delta g(x_i, 0) = \int_V 2\gamma \frac{\Delta\rho(x, z) z}{((x - x_i)^2 + z^2)}\, \mathrm{d}x\, \mathrm{d}z \tag{21}$$

where γ is the gravitational constant. Data are acquired at 400 equispaced points. The density model is discretized with the same mesh used in the tomography problem and

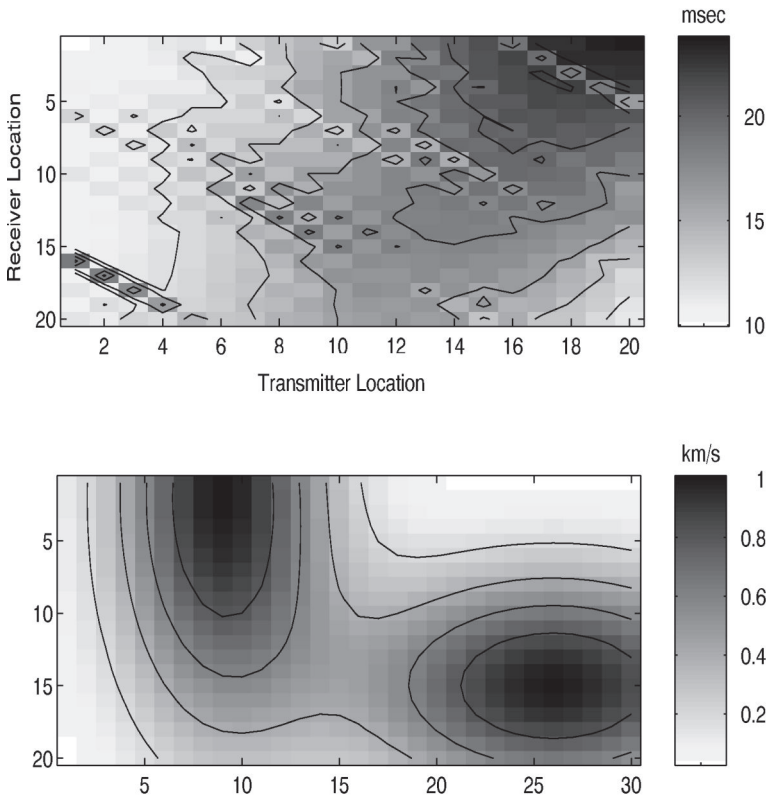

Figure 3. The anomalous slowness model used in the cross-well seismic survey is shown at the bottom. The anomalous travel-time data, without the addition of noise, are plotted at the top.

hence the gravity data take the form $\Delta g = G_2 \Delta\rho$ where G_2 is a 400×600 matrix but is full. The data, contaminated with 5% random Gaussian noise are shown in figure 4.

Before proceeding with the joint inversion we invert each data set separately. The data equations are first divided by their standard deviations. We then minimize $\|Lm\|^2$ subject to requiring that $\phi_d^* = 400$, that is, the final misfit is equal to the expected χ^2 value. Here L is the stabilized Laplacian operator $\nabla^2 + \epsilon I$. The results are shown in figure 5. The seismic inversion shows two regions of increased slowness. These regions correspond well with the location of the true anomaly, however they do not have the correct amplitude.

In the gravity inversion, the recovered density, shows the desired negative and positive anomalies in their correct lateral position but depth information is incorrect. The anomalous density extends to the surface. This is a typical result when inverting gravity data without incorporating any vertical weighting to force the density to be more distributed with depth.

For the joint inversion we again first normalize the equations by their standard deviations. We now have a total of 800 data and correspondingly set the desired misfit $\phi_d^* = 800$. The two linear systems are put together as described earlier. The structure operator is the same stabilized Laplacian as used in the individual inversions and the threshold values are $\tau_1 = 0.5$ and $\tau_2 = 0.7$ for both the seismic and gravity models. The starting slowness and density models were the zero models. The misfit convergence curve and a plot of ϕ_c as a function of iteration are given in figure 6. The computed slowness and density models are shown in figures 5(e) and (f) respectively. The improvement over the individual inversions

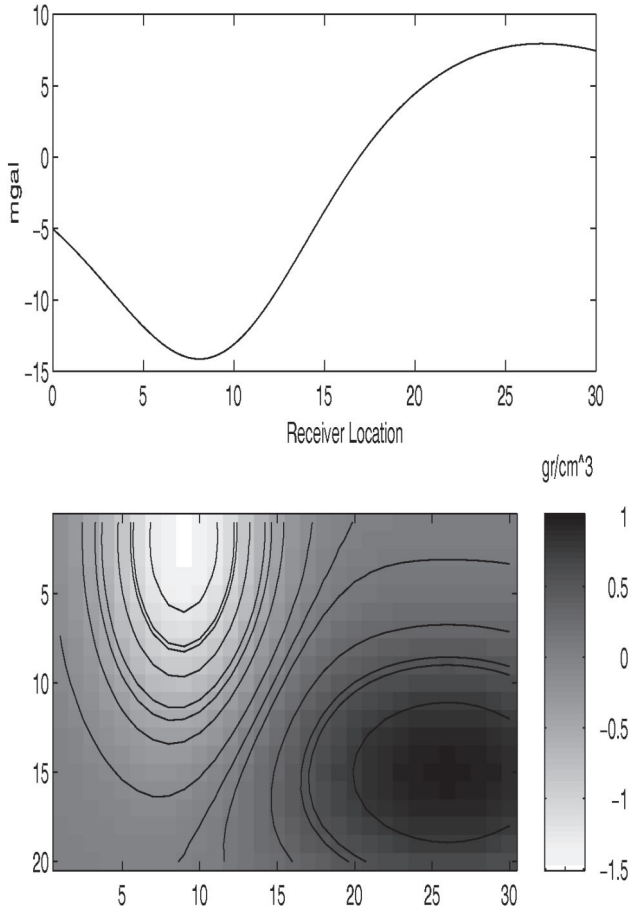

Figure 4. The anomalous density model is shown at the bottom. Surface gravity data, uncontaminated with error, are shown at the top.

is substantial. The slowness anomaly on the left has the correct amplitude. The anomalous densities have moved downward and now coincide well with the true locations. There are still discrepancies between the true and recovered models but overall the agreement is quite good. This illustrates that the gravity data have provided complementary information for the inversion of seismic data and vice versa.

4.2. Joint inversion of different frequencies of RIM data

RIM is a high frequency EM survey which is used to obtain information about electrical conductivity. A transmitter, in this case a vertical magnetic dipole, is deployed in one borehole and a receiver coil is in another borehole. Ray theory is adopted and the source signal is assumed to travel along a straight line connecting the source and receiver. The amplitude of the EM wave at the receiver is given by

$$A_r = \frac{A_0}{r} \mathrm{e}^{-\int_{l_i} s(x,z,\omega)\,\mathrm{d}l} \tag{22}$$

where A_0 is the amplitude at the source, r is the distance between the transmitter and the receiver, l_i is the ray path and $s(x, z, \omega)$ is an attenuation coefficient which depends upon conductivity of the medium and frequency. The frequency dependence arises because

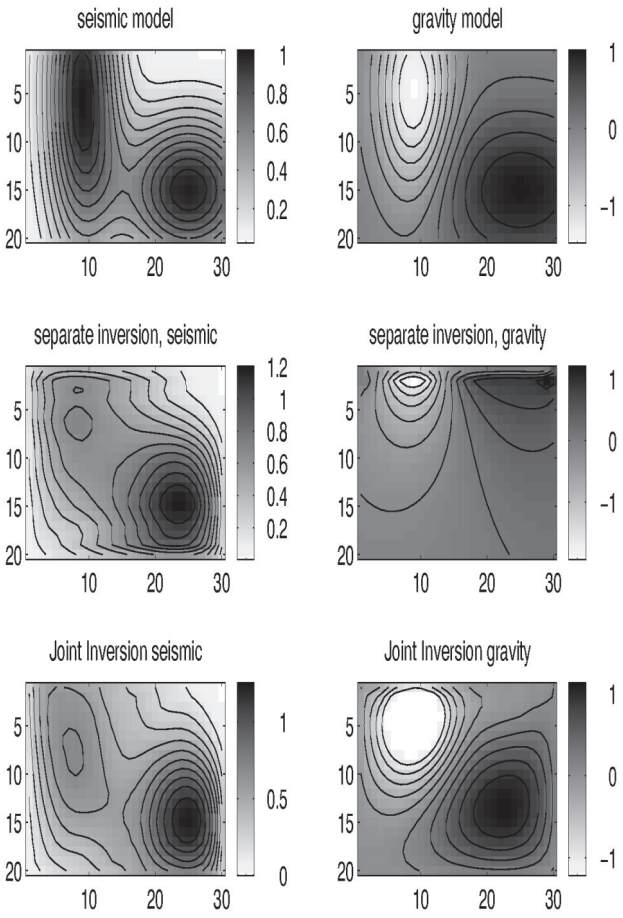

Figure 5. The true slowness and density models are shown in the top row. Inversion of the individual data sets produces the models in the middle row. The results of the joint inversion are shown at the bottom and they compare well with the true models.

the electrical conductivity is frequency dependent at these high frequencies (Knight and Nur 1987). Taking the logarithm of equation (22) yields a linear relationship between the attenuation coefficient and the data:

$$\log(A_r)_i - \log(A_0) + \log(r_i) = -\int_{l_i} s(x, z, \omega)\, \mathrm{d}l. \tag{23}$$

RIM data are generally collected over a range of frequencies and data from each frequency are inverted to estimate s. Although s is frequency dependent we expect that models at different frequencies would have a similar structure. Correspondingly, this makes RIM data a good candidate for the joint inversion.

Field data were acquired in two boreholes separated by 70 m. There were 31 source positions and 32 receiver positions and 856 data were collected at each frequency. The boreholes intersected mineralization and the goal of the RIM survey was to see if extensions of the mineralization, which are thought to appear as regions of high attenuation, could be traced outward from the holes.

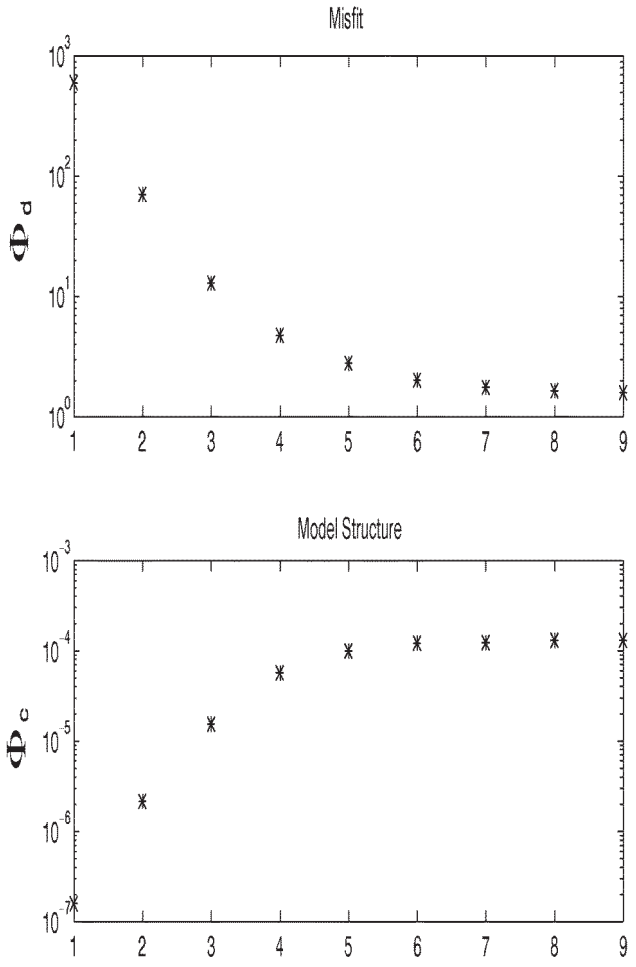

Figure 6. The data misfit ϕ_d as a function of iteration is plotted at the top. The measure of structural misfit ϕ_c is given below. The value of μ was kept fixed for all iterations.

Unfortunately good estimates of the errors were not available. The procedure for dealing with this for the individual inversions was to use the L-curve technique (Hansen 1994) and stop the inversion when the model norm began to increase sharply without much decrease in the misfit. The model domain was divided into 43×37 cells and the stabilized Laplacian was minimized. The results of the separate inversions for frequencies 3.1 and 11.5 MHz are shown in the top two panels of figure 7. There is some correspondence between the two images but there are also major differences. With the L-curve technique the rms misfit relative to the data was 0.33.

In carrying out the joint inversion we set the target misfit to be equal to the sum of the misfits from the two individual inversions. Again the stabilized Laplacian was used for the structure operator and τ_1 and τ_2 were, respectively, 10^{-4} and 10^{-2}. The models, recovered after 10 iterations, are shown in figures 7(c) and (d). The two models from the joint inversion are quite similar and they differ substantially from those obtained in the individual inversions.

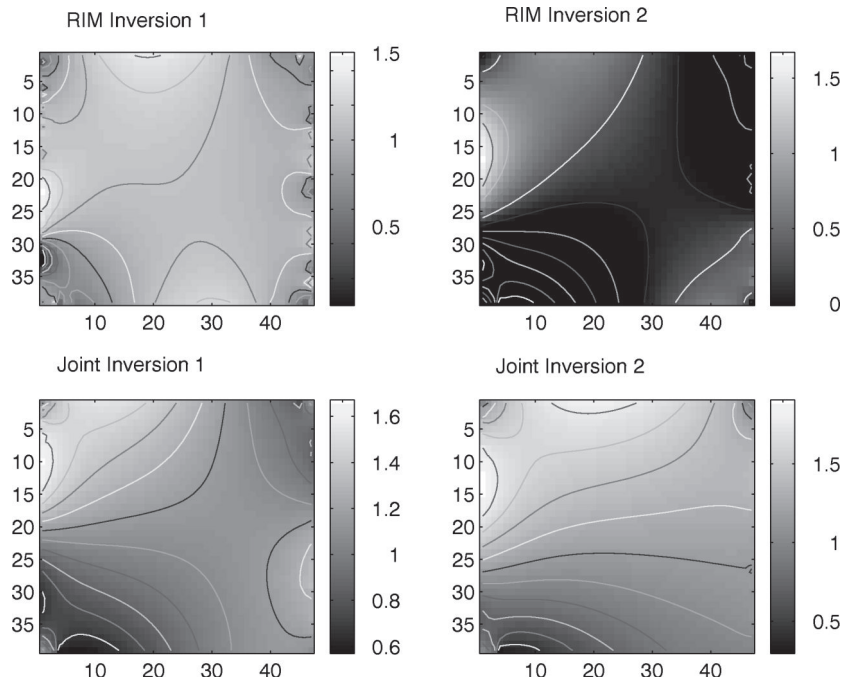

Figure 7. Inversion of two frequencies of RIM data. Models obtained by inverting the individual data sets are provided at the top. The models obtained with a joint inversion are at the bottom.

5. Summary and conclusion

We have developed a generic approach to invert two data sets when the underlying models are linked by having structural similarity. The importance of this approach is most evident in surveys which are sensitive to different physical properties. Without a way to link the two properties we generally proceed by inverting the data sets separately. A joint interpretation of the inversion results is then carried out looking for similarities or differences between the recovered models or by carrying out additional processing on the images. However, non-uniqueness inherent in the individual inversions may mean that incorrect conclusions are obtained. The joint inversion approach provided here reduces the non-uniqueness and hopefully improves the quality of interpretation.

References

Brown P and Saad Y 1990 Hybrid Krylov methods for nonlinear systems of equations *SIAM J. Sci. Stat. Comput.* **11** 450–81

——1994 Convergence theory of nonlinear Newton–Krylov algorithms *SIAM J. Opt.* **4** 297–330

Dennis J E and Schnabel R B 1983 *Numerical Methods for Unconstrained Optimization and Nonlinear Equations* (Englewood Clifts, NJ: Prentice-Hall)

Hanke M and Hansen P C 1993 Regularization methods for large scale problems *Surv. Math. Indust.* **3** 253–315

Hansen P C 1994 *PhD Thesis* Technical University of Denmark

Knight R and Nur A 1987 The dielectric constant of sandstones, 60 kHz to 4 Mhz *Geophys.* **52** 644–54