Geophys. J. Int. (2003) **152,** 251–265

Fast inversion of large-scale magnetic data using wavelet transforms and a logarithmic barrier method

Yaoguo Li* and Douglas W. Oldenburg

UBC-Geophysical Inversion Facility, Department of Earth and Ocean Sciences, University of British Columbia, Vancouver, Canada, V6T 1Z4

Accepted 2002 May 14. Received 2002 March 25; in original form 1999 October 6

SUMMARY

In this paper wavelet transforms and a logarithmic barrier method are applied to the inversion of large-scale magnetic data to recover a 3-D distribution of magnetic susceptibility. The fast wavelet transform is used, along with thresholding the small wavelet coefficients, to form a sparse representation of the sensitivity matrix. The reduced size of the resultant matrix allows the solution of large problems that are otherwise intractable. The compressed matrix is used to carry out fast forward modelling by performing matrix-vector multiplications in the wavelet domain. The reduction in CPU time is directly proportional to the compression ratio of the matrix. A second important feature of the algorithm used here is the use of an interior-point method of optimization to enforce positivity constraints. In this approach, the positivity is incorporated into the inversion by a sequence of non-linear optimizations approximated by truncated Newton steps. At the heart of the algorithm, a linear system of equations is solved. The conjugate gradient technique has been used as the basic solver to take the advantage of the efficient forward modelling offered by the sparse matrix representation. Overall, the combination of wavelet transforms, interior point optimization and conjugate gradient solutions readily allows us to solve magnetic inverse problems that have a few hundred thousand parameters and tens of thousands of data.

Key words: 3-D, conjugate gradients, interior point method, inversion, magnetic data, positivity, wavelet transform.

INTRODUCTION

In an earlier paper (Li & Oldenburg 1996), a generalized magnetic inversion for constructing 3-D distributions of magnetic susceptibility was developed. That method is capable of dealing with multiple anomalies and arbitrary susceptibility distributions, and it offers an effective means to image magnetic sources in a complex geological environment. The algorithm has been used successfully to interpret magnetic data for mineral exploration problems on scales ranging from ore deposits to mine districts. Inversion of magnetic data at regional scales effectively generates a 3-D image of the regional geology, which can provide valuable information such as that on prospective deposit horizons.

In principle, the above algorithm can be applied to large-scale data. Numerically, however, the computational complexity increases rapidly with the increasing size of the problem and the solution of large-scale inversion of magnetic data faces two major obstacles. The first is the large amount of computer memory required for storing the sensitivity matrix. For example, a moderate-sized problem that has 100 000 unknowns and 5000 data would require 2 Gb to store the sensitivity matrix in single precision. The second obstacle is the large amount of CPU time required for the application of the dense sensitivity matrix to vectors. These two factors directly limit the size of practically solvable problems. To deal with the first obstacle, one can store the matrix outside the core memory or generate it at the time of processing. However, these two options will pay the heavy price of either increased disc access time or increased CPU time for matrix generation. An alternative approach is to carry out the matrix-vector multiplication using the fast Fourier transform (FFT) since the magnetic kernels are translation invariant (Pilkington 1997). The use of FFT's will alleviate the memory limitation and reduce the CPU time dramatically. The restriction of this approach, however, is that the observations must lie above the surface topography and all data must be located over a regular grid on a flat observation surface. Although specialized algorithms utilizing this approach can be generated to solve large problems on regional scales, it cannot address the afore-mentioned difficulties for general applications in which the surface topography is almost always present and the data are located on uneven observation surfaces, over irregular grids, or in boreholes. For a generally applicable algorithm, other strategies need to be considered.

*Presently: Department of Geophysics, Colorado School of Mines, Golden, Colorado.

Recent advances in the theory of wavelet transforms have provided novel means of efficient representation of functions and functional operators, and such representations have in turn led to fast algorithms for various numerical analyses (Beylkin *et al.* 1991; Beylkin 1992, 1993; Harten & Yad-Shalom 1994). In these applications, the operator (e.g. matrix) and function (e.g. vector) are represented in the wavelet domain by only the significant coefficients. Therefore, both the storage requirement and multiplication counts are reduced in proportion to the number of winnowed coefficients. This results in reduced memory and CPU time required for solving a given problem and addresses the two major difficulties in the solution of large-scale magnetic inversions discussed earlier. In this paper, we examine the feasibility of using such a wavelet-based fast matrix-vector multiplication algorithm in inverting large-scale magnetic data sets.

In addition to the above numerical challenges, we must also enforce positivity on the inverted susceptibility since it is usually non-negative for exploration problems. This is an essential component of our algorithm and enables the recovery of depth information about the susceptibility. Our original algorithm (Li & Oldenburg 1996) applies positivity by a nonlinear mapping that is implemented naturally with the iterations of subspace linear inversion method. However, since we now have an implicitly sparse sensitivity matrix from the wavelet compression, it is natural to use the conjugate gradient (CG) technique in solving the linear system of equations. We therefore would like to develop a new approach for enforcing the positivity that will work well with the CG technique. The newly developed interior-point method (IPM) of optimization is ideally suited for this purpose. This is a class of algorithms for solving inequality-constrained optimization problems. We use a special variant called the primal logarithmic barrier method. In this approach, the positivity is implemented using a logarithmic barrier function and the final solution is obtained by solving a sequence of non-linear minimization problems. At the heart of the algorithm is the solution of a linear system of equations using the CG technique based upon the sparse representation of the sensitvity matrix.

The essence of this paper, therefore, is the combined use of the wavelet compression and the logarithmic barrier method of optimization in solving large-scale magnetic inverse problems. We begin with a brief review of the fast wavelet transform based upon orthonormal bases of compactly supported wavelets, and then apply the fast wavelet transform to compress the sensitivity matrix and perform fast forward modelling. Next we discuss the inversion of magnetic data using a logarithmic barrier technique, in which the conjugate gradient method is used as the central solver and the fast forward modelling performs the core computation. We then apply the method to synthetic and field examples and conclude the paper with a brief discussion.

BASICS OF WAVELET TRANSFORMS

Wavelet theory is a vast field where aspects of mathematics, scientific computing, and signal analysis converge under a single framework. For a rigorous mathematical treatment, we refer readers to the classics in this field (e.g. Mallat 1989; Daubechies 1992; Meyer 1993). In this section, we give a general description of wavelet transforms, which provides an intuitive understanding for the use of the fast wavelet transform in our magnetic inverse problems.

The wavelet transform expands a function in the bases formed by the translation and dilation of a single function called the mother wavelet. Let $f(x)$ be the function and $\psi(x)$ be the mother wavelet. Then the wavelet transform $w(a, b)$ is defined by

$$w(a,b) = \int_{-\infty}^{\infty} f(x)\psi_{a,b}(x)\,dx, \tag{1}$$

where

$$\psi_{a,b}(x) = \frac{1}{\sqrt{a}}\,\psi\left(\frac{x-b}{a}\right), \tag{2}$$

and a and b are the dilation and translation variables, respectively. The wavelets are concentrated over a short spatial interval and also have limited bandwidth in the wavenumber domain. The wavelet transform $w(a, b)$ describes the frequency or scale content (measured by a) at different locations (measured by b), that is, it provides resolution in both the spatial and frequency domains. This property makes the wavelet transform well-suited for analyzing non-stationary signals that have transient events. This is in sharp contrast to the Fourier transform, which uses global basis functions and is better suited for stationary signals as it has maximum resolution in the frequency domain and retains no explicit information about the spatial location of an event.

For practical applications, the dilation and translation variables take on a set of discrete values, which are typically dyadic. The wavelet is then expressed as a double-indexed function,

$$\psi_{j,k}(x) = 2^{-j/2}\psi(2^{-j}x - k), \tag{3}$$

where j and k are integers. A class of wavelets are constructed by solving the two-scale difference equation

$$\phi(x) = \sqrt{2}\sum_{k=0}^{L-1} h_k\phi(2x-k),$$
$$\psi(x) = \sqrt{2}\sum_{k=0}^{L-1} g_k\phi(2x-k), \tag{4}$$

where ϕ is the accompanying scaling function. The wavelet $\psi(x)$ is then completely defined by a set of filter coefficients h_k and g_k called the quadrature mirror filter. Daubechies (1988) constructed a class of such wavelets that has several important properties. These wavelets are orthogonal to the dyadic translation and dilation of the original version; they have compact support and are localized both in space and in frequency domain; they are constructed to have M vanishing moments:

$$\int_{-\infty}^{\infty} \psi(x)x^m dx = 0, \quad m = 0, \ldots, M-1, \tag{5}$$

where M is an integer. The last property makes these wavelets orthogonal to low order polynomials. When $M = 1$, we obtain the Haar wavelet; when $M = 2$, we have the Daubechies-4 wavelet defined by a quadrature mirror filter of length 4. As an illustration, both wavelets are shown in Fig. 1.

The orthonormal bases of compactly supported wavelets allow the formulation of multiresolution analysis in which the Hilbert space $\mathbf{L}^2(\mathcal{R})$ is decomposed into a chain of closed subspaces

$$\cdots \subset V_2 \subset V_1 \subset V_0 \subset V_{-1} \subset \cdots \tag{6}$$

Without dwelling on details of the mathematics, it suffices to say that each subspace has a characteristic scale or resolution and the projection of a function onto a subspace V_j gives the representation of the function at that resolution. Let the subspace W_j be the orthogonal complement of V_j in V_{j-1},

$$V_{j-1} = V_j \oplus W_j, \tag{7}$$

then the difference between the representations of the function at two successive resolution scales is in the subspace W_j, which represents

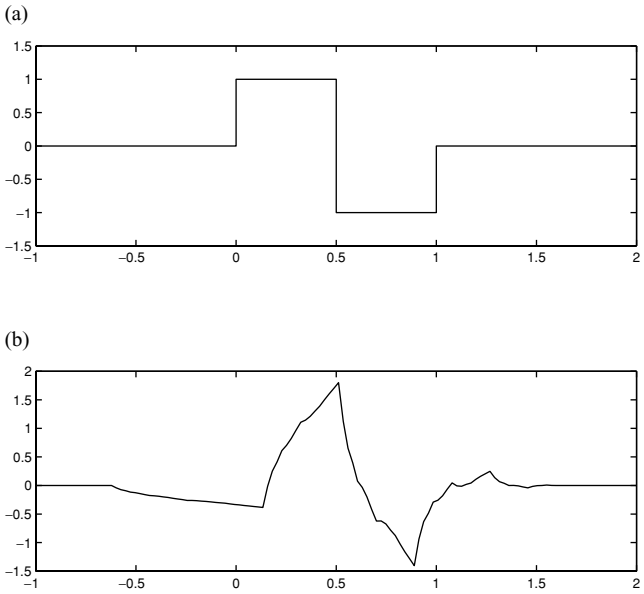

Figure 1. (a) The Haar wavelet, which has 1 vanishing moment. (b) The Daubechies-4 wavelet, which has 2 vanishing moments.

the details of the signal and is given by the wavelet transform on that scale. The subspace V_j is spanned by the scaling functions $\phi_{j,k}$ while W_j is spanned by the wavelets $\psi_{j,k}$. For numerical applications, we have a finite number of scales. Let $j = 0$ be the coarsest scale and $j = -n$ be the finest scale, then

$$V_{-n} = V_0 \oplus \bigoplus_{k=1}^{n} W_{-k}. \tag{8}$$

The discrete wavelet transform using these orthonormal, compactly supported wavelets accomplishes exactly the decomposition expressed in eq. (8) by generating the wavelet coefficients in W_{-j} and the coarsest representation in V_0. An efficient algorithm for computing such an orthonormal wavelet transform is the well-known pyramid algorithm first outlined by Mallat (1989).

The discrete wavelet transform is a linear transformation, and it can be symbolically represented by a matrix that acts upon a vector representing the discretized function. Let $\mathbf{W}$ be the matrix representing the fast wavelet transform and $\mathbf{v}$ be the vector, then the wavelet transform $\tilde{v}$ is given by,

$$\tilde{v} = \mathbf{W}\mathbf{v}. \tag{9}$$

The transform $\mathbf{W}$ is orthonormal and its inverse is given by its transpose, $\mathbf{W}^{-1} = \mathbf{W}^T$. Thus we have the following identity,

$$\mathbf{W}^T\mathbf{W} = \mathbf{W}\mathbf{W}^T = \mathbf{I}, \tag{10}$$

where $\mathbf{I}$ is the identity matrix. This relation allows the reconstruction of a function from its wavelet coefficients,

$$\mathbf{v} = \mathbf{W}^T\tilde{v}. \tag{11}$$

The wavelet transform of a function based on the orthonormal, compactly supported wavelets usually has a large number of coefficients that are either zero or very close to zero. Winnowing the coefficients whose magnitudes are below a certain level still allows reconstruction of the original function with a high degree of accuracy. Thus a function can have a sparse representation in the wavelet domain. This results first from the localization property of the wavelet as localized events do not produce significant coefficients at a distant location, and second, the orthogonality of the wavelets with low order polynomials means that a smooth piece of the function that is well-approximated by a polynomial will be represented by using only a few large coefficients.

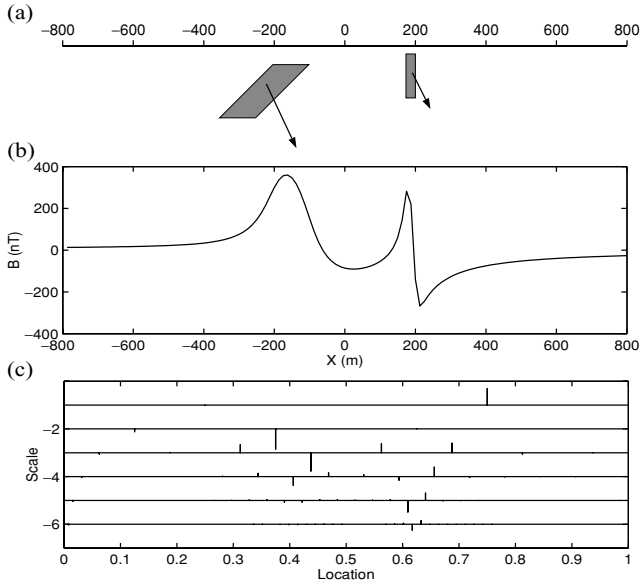

Figure 2. (a) Fast wavelet transform of a magnetic–anomaly profile generated by a 2-D distribution of susceptibilities. (b) The magnetic data. (c) The wavelet transform. The coefficients are shown as vertical lines with the length proportional to the amplitude. They are plotted as a function of location and scale.

As an example of wavelet transform applied to data arising from magnetic problems, Fig. 2 shows the wavelet transform (using Daubechies-4 wavelet) of a magnetic-anomaly profile. There are 128 data points in the profile. The wavelet coefficients are shown as vertical lines with the length proportional to the amplitude at their locations and scales. Notice that most of the wavelet coefficients are very small. Discarding the coefficients that are below three different thresholds relative to the largest coefficient produces the reconstructed profiles shown in Fig. 3. They can be compared with the true data. The reconstructions have used only 53, 43, 23 coefficients, respectively, out of the total of 128. At the low threshold level, the reconstruction is virtually identical to the original profile, and as the threshold level increases, the small scale distortions begin to appear but the long wavelength features remain. This example demonstrates the efficient sparse representation of a function in the wavelet domain. It is this property that we shall use to construct a fast magnetic inversion algorithm: we will compress the 3-D sensitivity so that the required memory and CPU time during the inversion are reduced.

In our application, the susceptibility distribution is a 3-D image, as is the sensitivity function that corresponds to a given observation. In order to apply the wavelet transform to our problems, we need to have the multi-dimensional form of the transform. One way of accomplishing the wavelet transform in multi-dimensions is to apply the 1-D wavelet transform independently to each dimension (e.g. Daubechies 1992). This is analogous to the fast Fourier transform in multi-dimensions. Such a multi-dimensional wavelet transform amounts to using 3-D wavelets formed by the tensor products of 1-D wavelets. We use this form in our work for its simplicity.

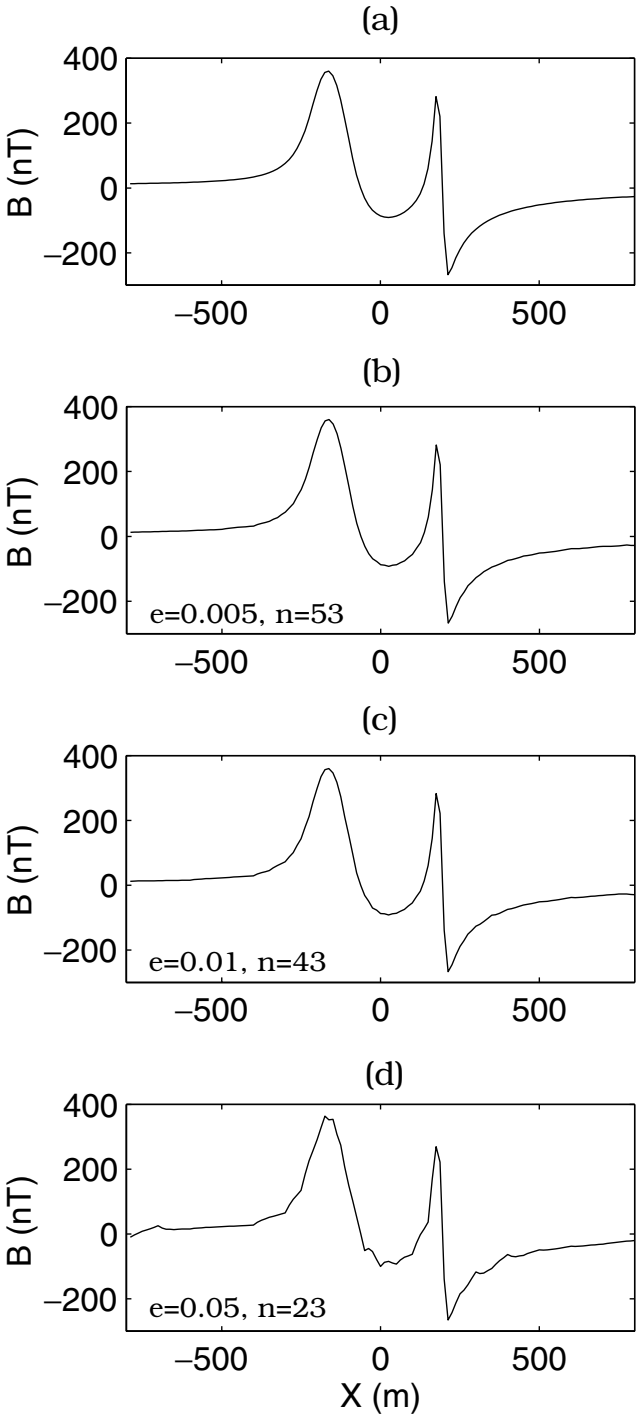

Figure 3. Comparison of the true profile (a) with three reconstructions from wavelet coefficients that are thresholded by different levels (b, c, d). The value of e in these panels indicates the relative threshold. All coefficients that are smaller than the largest wavelet multiplied by e are set to zero. The value of n in each panel indicates the number of coefficients that are kept.

FAST FORWARD MAPPING FOR MAGNETIC DATA

We now proceed to develop fast forward mapping of magnetic data using the wavelet transform based upon orthonormal bases of compactly supported wavelets. Let the 3-D distribution of the susceptibility be represented by a set of cuboidal cells of constant values. The collection of the susceptibility values in the cells forms the model vector $\boldsymbol{\kappa}$. The magnetic data observed above, and in, this 3-D susceptibility distribution are collected into the data vector $\mathbf{d}$. Then, under the commonly adopted assumption that there is no remanent magnetization and that the self-demagnetization effect is negligible, the observed data are linearly related to the susceptibility model by the sensitivity matrix:

$$\mathbf{G}\kappa = \mathbf{d}, \tag{12}$$

where $\mathbf{G}$ is the sensitivity matrix and it is full. In the direct approach, the entire matrix is stored during the inversion and it is applied to vectors by direct multiplication.

Our goal is to construct an alternative approach such that less memory is needed for storing $\mathbf{G}$ and fewer operations are needed to apply it to a vector. In the following, we first present the wavelet domain equivalent of two matrix-vector multiplications that are necessary for carrying out an inversion. We then discuss the issues concerning the practical applications.

Forward mapping in the wavelet domain

The rows of the sensitivity matrix are defined over the 3-D domain of the model κ and can be treated as 3-D images. Application of the 3-D wavelet transform can therefore produce a sparse representation of them. Let $\mathbf{W}$ be the 3-D wavelet transform, and $\tilde{\mathbf{G}}$ be the matrix whose rows are the wavelet transforms of corresponding rows of $\mathbf{G}$. Then $\tilde{\mathbf{G}}$ is given by

$$\tilde{\mathbf{G}} = \mathbf{G}\mathbf{W}^T. \tag{13}$$

We call $\tilde{\mathbf{G}}$ the transformed sensitivity matrix. We also apply the identical wavelet transform to the model vector κ to produce the transformed model $\tilde{\kappa}$. Let

$$\tilde{\kappa} = \mathbf{W}\kappa. \tag{14}$$

Multiplying eqs (13) and (14), applying eqs (10) and (12) yields,

$$\tilde{\mathbf{G}}\tilde{\kappa} = \mathbf{d}. \tag{15}$$

Therefore the forward modelling is accomplished by the multiplication of the transformed sensitivity and the transformed model in the wavelet domain.

Eq. (15) allows the application of the sensitivity matrix to a general model vector. Most inversion algorithms also require the application of $\mathbf{G}^T$ to a data vector $\mathbf{v}_d$ to produce a 'model' vector $\mathbf{v}_\kappa$, i.e.,

$$\mathbf{G}^T\mathbf{v_d} = \mathbf{v}_\kappa. \tag{16}$$

By multiplying the two sides of eq. (16) by $\mathbf{W}^T\mathbf{W} = \mathbf{I}$ and regrouping the terms, we obtain,

$$\mathbf{W}^T\left(\tilde{\mathbf{G}}^T\mathbf{v}_d\right) = \mathbf{v}_\kappa. \tag{17}$$

Thus the application of $\mathbf{G}^T$ to a vector is accomplished by the multiplication of the transformed sensitivity with the data and an inverse wavelet transform.

We now turn our attention to the goal of generating an efficient forward modelling algorithm. The transformation to the wavelet domain itself does not decrease the required storage nor does it increase the speed of calculation. It is the properties of functions in the wavelet domain, and the processing based upon these properties,

that make the fast algorithm possible. As discussed in the preceding section, the wavelet transform of a generally smooth function has many coefficients that are close to zero. Setting to zero those coefficients that are below a certain threshold still allows the reconstruction of the original function with high degree of accuracy. That is, only the significant coefficients that are above the threshold are needed in the wavelet domain to represent the function. Thus, we can apply thresholding to the transformed sensitivity matrix $\tilde{\mathbf{G}}$ to generate a sparse matrix $\tilde{\mathbf{G}}_s$. We have adopted a threshold that is applied to each row of $\tilde{\mathbf{G}}$,

$$\tilde{\mathbf{g}}_{ij}^s = \begin{cases} \tilde{\mathbf{g}}_{ij}, & |\tilde{\mathbf{g}}_{ij}| \geq \delta_i \\ 0, & |\tilde{\mathbf{g}}_{ij}| < \delta_i \end{cases} \quad i = 1, \ldots, N \tag{18}$$

where $\tilde{\mathbf{g}}_{ij}$ are the elements of $\tilde{\mathbf{G}}$ and δ_i is the threshold level for the ith row. The value of δ_i is discussed in the next section. $\tilde{\mathbf{g}}_{ij}^s$ then form the sparse representation $\tilde{\mathbf{G}}^s$.

During the course of an inversion, we need only to store the sparse matrix $\tilde{\mathbf{G}}_s$, which yields the desired reduction in the memory requirement. In addition, the matrix $\tilde{\mathbf{G}}_s$ is substituted for $\tilde{\mathbf{G}}$ in eqs (15) and (17) so that the application of $\mathbf{G}$ or $\mathbf{G}^T$ is carried out by sparse multiplication in the wavelet domain and the number of operations is reduced in proportion to the sparseness of $\tilde{\mathbf{G}}_s$. This yields the desired reduction in CPU time.

The reduction in memory and CPU time therefore depends directly on how sparse the wavelet representation is. To measure the sparseness, we define a compression ratio that is given by the ratio of the number of entries in matrix $\mathbf{G}$ to the number coefficients that are kept in $\tilde{\mathbf{G}}_s$. Typical values of the compression ratio ranges from 10 to 50. It follows that we can achieve a reduction in CPU by a factor of 20 to 50. This is a significant saving for large-scale applications.

When such a fast forward modelling is used in an inversion, it requires repeated application of forward and inverse wavelet transform. Specifically, one application of $\mathbf{G}$ requires one forward wavelet transform and one application of $\mathbf{G}^T$ requires an inverse wavelet transform. This will incur some computational overhead. However, forward and inverse wavelet transforms are carried out using the pyramid algorithm (Mallat 1989), which is extremely efficient. The computational overhead is negligible compared to the savings achieved through sparse representation of the sensitivity matrix.

To summarize, the fast forward modelling used in an inversion takes the following steps:

(1) Generate the matrix $\mathbf{G}$ row by row. As each row is computed, apply the wavelet transform to it and then threshold the wavelet coefficients according to (18) to produce a sparse representation. This step only requires the memory to hold one row of the sensitivity matrix.

(2) Store the sparse matrix $\tilde{\mathbf{G}}_s$ as its rows are generated. Only the memory to store the sparse matrix in Yale (row major) format is needed.

(3) Apply $\tilde{\mathbf{G}}_s$ to vectors using sparse multiplication during the inversion.

Practical aspects

In order to apply the above method to realistic magnetic inverse problems, several practical issues need to be treated. These include the choice of the wavelet, method of thresholding, analysis of error bounds and effective treatment of surface topography in the model. These are discussed here.

There are many choices of wavelet that can be used to compress the sensitivity matrix. The desired wavelet should produce the greatest compression ratio and the least distortion to the sensitivity. This suggests the use of wavelets with higher vanishing moments, such as Daubechies wavelets, or symmlets. Since the inversion is carried out in the spatial domain, while the matrix-vector multiplications are done in the wavelet domain, there is an added cost of transforming a model vector between the two domains, which is directly proportional to the length of the quadrature mirror filter defining the wavelet. This suggests the use of shorter wavelets. Numerical test have shown that the Haar wavelet and Daubechies wavelet with 2 or 3 vanishing moments perform well and longer wavelets do not seem to improve the overall performance. In general, the Haar wavelet yields a better compression ratio when the required accuracy of reconstruction is low and therefore produces the most savings in memory and CPU time. As the accuracy requirement increases, Daubechies wavelets out-perform Haar wavelets. As a result, our algorithm uses mostly Daubechies wavelets.

We threshold the wavelet coefficients of the sensitivity matrix based upon their amplitudes according to eq. (18). All coefficients that are below a threshold, δ_i, are set to zero. This introduces certain errors in the forward mapping. We therefore choose the value of δ_i to be small enough so the mapping error is acceptable. A direct measure is given by the relative data error, $r_d = ||\delta\mathbf{d}||/||\mathbf{d}||$, where $\mathbf{d}$ is the data vector and $\delta\mathbf{d}$ is the vector of errors resulted from thresholding the transformed sensitivity, and $||\cdot||$ denotes the L_2 norm of a vector. However, the error is model dependent and cannot be easily evaluated. It can be used to verify the accuracy of the sensitivity after the inversion is completed. For determining the threshold level, δ_i, we use an alternative measure. We require that each row of the sensitivity matrix has an acceptable reconstruction error, which is given by $r_i(\delta_i) = ||\mathbf{G}_i - \mathbf{G}_{ri}||/||\mathbf{G}_i||$, $i = 1, \ldots, N$, where $\mathbf{G}_i$ is the ith row of the sensitivity matrix, $\mathbf{G}_{ri}$ is the ith row of the reconstructed sensitivity matrix. Since the wavelet transform is orthonormal and $||\mathbf{G}_i|| = ||\tilde{\mathbf{G}}_i||$, the above equation can be expressed as the ratio of the norm of winnowed coefficients over the norm of the transformed sensitivity,

$$r_i(\delta_i) = \sqrt{\frac{\sum\limits_{|\tilde{\mathbf{g}}_{ij}|<\delta_i} \tilde{\mathbf{g}}_{ij}^2}{\sum\limits_j \tilde{\mathbf{g}}_{ij}^2}}, \quad i = 1, \ldots, N. \tag{19}$$

The quantity $r_i(\delta_i)$ increases monotonically with δ_i. Thus the value of δ_i can be determined for a prescribed accuracy r^* by a simple line search carried out directly in the wavelet domain. It is expensive to calculate an individual threshold for each row so we have therefore implemented a relative thresholding method. Assume the i_0th row is a representative row, and δ_{i_0} is the corresponding absolute threshold. A relative threshold can be defined as $\epsilon = \delta_{i_0} / \max_j(|\tilde{\mathbf{g}}_{i_0 j}|)$ such that a coefficient is set to zero when the ratio of its amplitude over the largest coefficient is less than ϵ. This relative threshold is then applied to every row of the matrix by defining the individual threshold level as

$$\delta_i = \epsilon \max_j(|\tilde{\mathbf{g}}_{ij}|), \quad i = 1, \ldots, N. \tag{20}$$

When more than one distinct group of data, such as surface and borehole data, are present, a representative row is chosen and the relative threshold is calculated for each group. It is observed that a value for r_i close to 0.05 provides a sufficiently good reconstruction of the sensitivities and the resulting relative error for testing models is usually smaller.

The sensitivity of magnetic data decays rapidly with the distance between the source and the observation location. Given the

geometry of a typical magnetic inversion problem, this decay can easily range over several orders of magnitude. This disparity in the elements of the sensitivity matrix will translate to the disparity in the wavelet coefficients and thresholding based upon relative amplitude is therefore likely to produce greater distortions in the region far away from the observation location. Fortunately, the magnetic inversion requires the use of a depth weighting function (Li & Oldenburg 1996, 2000). That weighting counteracts the decay of the sensitivities so that the inversion constructs a model based primarily on the non-decaying portion of the sensitivities, and this produces a susceptibility distribution that has better depth characteristics. Numerically, the result is achieved by treating the product of the depth weighting function and susceptibility as the new model. The amplitude and structural complexity of the new model are controlled by a model objective function. The forward modelling, required as part of the inversion process, is achieved by multiplying a weighted sensitivity matrix with the new model as shown below. We then apply wavelet compression to the weighted sensitivity matrix so that both the compression ratio and reconstruction accuracy are improved.

Let $\mathbf{Z}$ be the diagonal matrix representing the depth weighting. To evaluate $\mathbf{G}\kappa$ we write $\mathbf{G}\kappa = \mathbf{G}\mathbf{Z}^{-1}\mathbf{Z}\kappa$. Let

$$\mathbf{G}_\zeta = \mathbf{G}\mathbf{Z}^{-1}, \quad \mathbf{m} = \mathbf{Z}\kappa. \tag{21}$$

Then solving the magnetic inversion with the depth weighting is equivalent to solving an inverse problem whose forward mapping is given by

$$\mathbf{G}_\zeta \mathbf{m} = \mathbf{d}. \tag{22}$$

We can now simply apply the wavelet transform to $\mathbf{G}_\zeta$ and carry out the thresholding as presented above. The matrix-vector operation then involves the weighted model vector $\mathbf{m}$, and the rest of the analysis follows in the similar manner.

The last issue is the surface topography, which causes the sensitivity to be defined over a non-rectangular region. A related issue is the discretization of the model domain with a finite difference mesh whose number of cells in each dimension is not dyadic (equal to an integer power of 2). The fast wavelet transform requires that the dimension of the data be dyadic. We circumvent both difficulties by padding the sensitivity 'image' with zeroes so that the number of points in each direction is dyadic. The model is also padded with zeros in the same manner. These added elements usually increase the number of significant coefficients slightly, and they do not greatly affect the over-all compression ratio.

Tests on synthetic examples suggest that the achievable compression ratio for the sensitivity matrix is between 10 and 50 when a reconstruction error better than 5 per cent is required. The compression ratio decreases to below 10 for borehole data while it can approach 100 for aeromagnetic data with large terrain clearance. This range of ratios gives an indication about the size of problems that can be tackled with the wavelet transform approach. For example, a workstation with only 128 Mb of memory can potentially invert aeromagnetic data that requires a sensitivity matrix of up to 2 Gb in a direct approach. This represents a significant increase in the size of solvable problems.

Numerical tests

We now illustrate the fast forward modelling algorithm using a synthetic problem. The size of the problem is chosen to be small enough so that all matrices can fit into the core memory. This allows the CPU time for accessing the sensitivity matrix from the disc and the CPU time for matrix-vector multiplication to be measured separately. The model domain is defined over a 1000 m by 1000 m by 500 m volume and the mesh has 30 cells in each horizontal direction and 20 cells in the vertical direction. The observations are located 2 m above the surface and on a grid of 30 by 30 points. We assume an inducing field direction of $I = 65^\circ$ and $D = 25^\circ$, and carry out the modelling for the total field observations. (Note that the regular grid for the data is chosen for convenience and it does not affect the result to be presented in the following.) For the illustration, we have also chosen a susceptibility model that consists of six rectangular blocks of different sizes. These blocks are placed at various locations within the mesh to ensure that there is a contribution to the observations from different regions of the model.

Table 1 shows the comparison of CPU time for accessing the sensitivity matrix from disc and for applying the sensitivity to a model vector. The CPU time is based on the performance on a Sparc20 workstation. Both the Haar wavelet and Daubechies-4 wavelet are tested for two different thresholding levels. Under these different combinations of a wavelet with the threshold, the achieved compression ratio under acceptable data accuracy (e.g. 5 per cent) is greater than 10. The CPU time needed for computing the matrix-vector multiplication in the wavelet domain ranges from one-tenth to a quarter of the CPU time for the same task in the spatial domain. These are significant savings. In addition, we also compare the CPU time needed to access the matrix from the disc. This time becomes relevant to large-scale problems in practical applications when the entire sensitivity matrix cannot be stored in the core memory and has to be accessed from disc for each matrix-vector multiplication. The relative difference between the forward modelled data and the true data are compared in the last column of Table 1. It is clear that, given the same threshold for the wavelet coefficients, the Daubechies wavelet achieves a lower compression ratio but much higher accuracy in the forward modelled data.

Fig. 4 compares the true sensitivity with the approximations reconstructed from the thresholded wavelet transforms for an observation location at the centre of the data grid. The sensitivity displayed in

Table 1. The sensitivity matrix for the test problem is compressed using Haar and Daubechies-4 wavelets and different thresholding levels. The resultant compression ratio and the CPU time for accessing the compressed matrix and for performing the forward modelling are compared in this table. The relative data error for the synthetic susceptibility model is also shown.

	Relative threshold	Compression ratio	Multiplication time (s)	Relative data error (per cent)
Wavelet domain (Haar)	0.005	34.0	0.2	8.70
	0.001	13.3	0.4	2.00
Wavelet domain (Daubechies-4)	0.005	18.6	0.4	3.64
	0.001	10.0	0.6	0.67
Spatial Domain	N/A	1.0	2.8	0.0

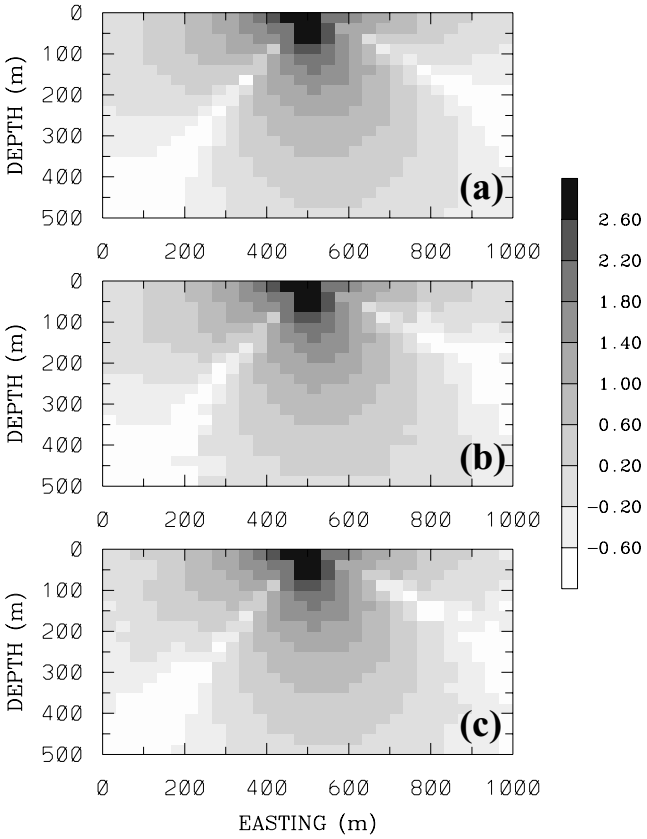

Figure 4. Comparison of the true sensitivity with the approximations reconstructed from thresholded wavelet transforms. The sensitivity shown is that of an observation located 2 m above the surface at the centre of the grid. Panel (a) is the true sensitivity. Panels (b) and (c) are reconstructed from wavelet transforms using Haar and Daubechies-4 wavelets respectively. The relative threshold level is 0.001. The gray scale indicates the logarithm of the absolute value of the sensitivity.

the figure is a cross-section in east–west direction directly below the observation location. The true sensitivity is well approximated by the reconstructed versions from Haar wavelet (Fig. 4b) and Daubechies-4 wavelet (Fig. 4c). Only minor discrepancies are noticeable near the zero-crossing of the sensitivity. The corresponding distortion in the calculated surface field is negligible and, therefore, is not a concern for the inversion. Fig. 5 shows the comparison between the true total field data on the surface and those computed from compressed sensitivities using Haar and Daubechies-4 wavelets, respectively, with a relative threshold of 0.001. Only minor distortions occur near the edge of the data map.

INVERSION BY LOGARITHMIC BARRIER METHOD USING CONJUGATE GRADIENTS

Formulation of the inverse problem

We now formulate the inversion using the wavelet transformed sensitivity matrix. As discussed in the preceding section, we represent the model by a large number of cells having constant susceptibilities. The data and susceptibility model are linearly related by the sensitivity matrix as expressed in eq. (12) or eq. (22). As in our earlier work, we obtain the inverse solution by minimizing an objective of the weighted model subject to fitting the observations to

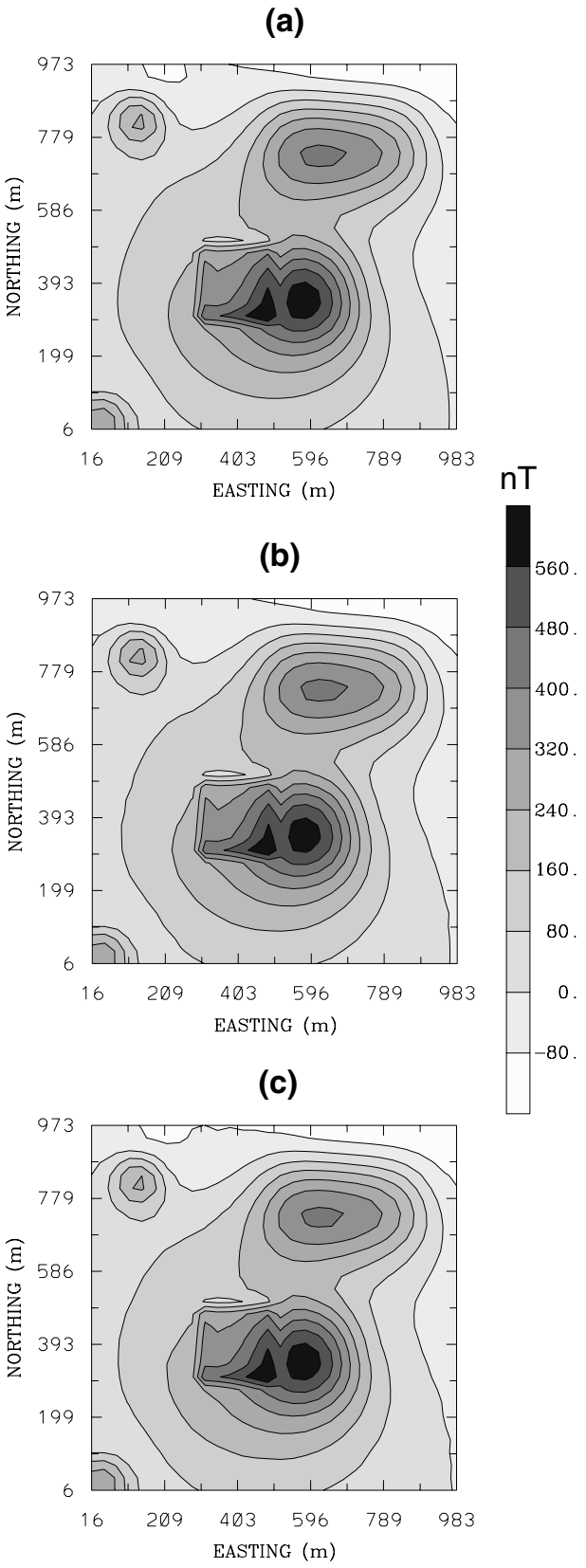

Figure 5. Comparison of true total field magnetic data on the surface with those computed from the compressed sensitivity matrices. Panel (a) is the true data. Panels (b) and (c) are the computed data when the sensitivity matrix is compressed using Haar and Daubechies-4 wavelet, respectively, and a relative threshold of 0.001.

the degree determined by the errors. The model objective function is given by

$$\phi_m = \alpha_s \int_V \{\zeta(\mathbf{r})[\kappa(\mathbf{r}) - \kappa_0]\}^2 \, dv + \alpha_x \int_V \left\{ \frac{\partial \zeta(\mathbf{r})[\kappa(\mathbf{r}) - \kappa_0]}{\partial x} \right\}^2 dv + \alpha_y \int_V \left\{ \frac{\partial \zeta(\mathbf{r})[\kappa(\mathbf{r}) - \kappa_0]}{\partial y} \right\}^2 dv + \alpha_z \int_V \left\{ \frac{\partial \zeta(\mathbf{r})[\kappa(\mathbf{r}) - \kappa_0]}{\partial z} \right\}^2 dv, \quad (23)$$

where $\kappa(\mathbf{r})$ is the susceptibility model, κ_0 is a reference model, and $\zeta(\mathbf{r})$ is the generalized depth-weighting function, which can be a simple depth weighting or a more general distance weighting. The constants α_s, α_x, α_y, and α_z are real, positive numbers controlling the importance of each term, and we usually set $\alpha_s \ll 1$ and $\alpha_x = \alpha_y = \alpha_z = 1$. Readers are referred to Li & Oldenburg (1996, 2000), for more details about the model objective function. We discretize this objective function using a finite difference approximation to produce

$$\phi_m = \kappa^T \mathbf{Z}^T \mathbf{W}_m^T \mathbf{W}_m \mathbf{Z} \kappa \equiv \mathbf{m} \mathbf{W}_m^T \mathbf{W}_m \mathbf{m}, \quad (24)$$

where $\mathbf{W}_m^T \mathbf{W}_m$ is the usual weighting matrix representing the model objective function and $\mathbf{Z}$ is the diagonal matrix containing the discretized depth weighting function. Here we have used eq. (21).

The data misfit is given by the χ^2 misfit measure. Let

$$\phi_d = (\mathbf{G}\kappa - \mathbf{d}_{obs})^T (\mathbf{G}\kappa - \mathbf{d}_{obs}) = (\mathbf{G}_\zeta \mathbf{m} - \mathbf{d}_{obs})^T (\mathbf{G}_\zeta \mathbf{m} - \mathbf{d}_{obs}), \quad (25)$$

where $\mathbf{d}_{obs}$ is the vector containing the observed data. We assume that the data and corresponding rows of the sensitivity matrix have been normalized by the standard deviations of the data errors.

The inverse solution is obtained by minimizing the model objective function subject to constraints that the data misfit achieves an acceptable value and that the susceptibility model be positive. For generality, we introduce a regularization parameter μ and solve the following equivalent minimization,

$$\begin{aligned} &\text{minimize :} \quad \phi = \phi_d + \mu \phi_m, \\ &\text{subject to :} \quad \mathbf{m} > \mathbf{0}. \end{aligned} \quad (26)$$

Here μ controls the relative importance of the model norm and data misfit. When the standard deviations of data errors are known, the acceptable misfit is given by the expected value ϕ_d^* and we will search for the value of μ that produces the expected misfit. Otherwise, an estimated value of μ will be prescribed. The details of various aspects of choosing a tradeoff parameter will be discussed in a following section. Once the vector $\mathbf{m}$ is obtained through the minimization problem in eq. (26), the susceptibility model κ is obtained by $\kappa = \mathrm{Z}^{-1}\, \mathbf{m}$.

Minimization by logarithmic barrier method

Having spent much effort on obtaining an efficient forward modelling algorithm that reduces both the memory requirement and CPU time, it is then logical to choose a minimization technique in which the efficiency of the forward modelling can be directly translated to the inverse solution. To achieve this goal, we use a primal logarithmic barrier method with the conjugate gradient technique as the central linear solver. The logarithmic barrier method was originally developed for solving linear and quadratic programming problems with inequality constraints (e.g. Gill *et al.* 1991; Wright 1997). We adapt that method to incorporate a positivity constraint.

In the logarithmic barrier method, the positivity constraint is implemented as a logarithmic term. The new objective function is given by (Gill *et al.* 1991; Saunders 1995)

$$\phi(\lambda) = \phi_d + \mu \phi_m - 2\lambda \sum_{j=1}^{M} \ln(m_j), \quad (27)$$

where $-2\lambda \sum_{j=1}^{M} \ln(m_j)$ is called the barrier function, λ is the barrier parameter, and the tradeoff parameter μ is fixed during the minimization. As the name suggests, the logarithmic barrier function forms a barrier along the boundary of the feasible domain (zero bound in our problem) and prevents the minimization from crossing over to the infeasible region. The method solves a sequence of nonlinear minimizations with decreasing λ and, as λ approaches zero, the sequence of solutions approaches the solution of eq. (26).

In our implementation, the parameters are constrained only from below by zero, and it is then necessary to scale the model $\mathbf{m}$ so that $m_i < 1$. This guarantees that the barrier function is positive. Otherwise, a model value greater than unity will cause the barrier function to be negative and the logarithmic barrier solution will not converge. This is not a concern with magnetic inversion since the susceptibility is generally far less than unity. If upper bounds on the parameters are available, they can be imposed in the same manner. It is then unnecessary to scale the parameters.

The minimization process starts with a large value of λ and an initial model $\mathbf{m}$ whose elements are all positive. It then finds the solution iteratively with the barrier parameter λ being decreased at each iteration. However, the minimization of the nonlinear functional in eq. (27) at each barrier iteration is an expensive process. Instead of carrying out the full minimization at each iteration, it is common to take a Newton step for each value of λ and adjust the step length so that the updated model remains positive (e.g. Gill *et al.* 1991). The step length is also used to determine the decreased value of the barrier parameter λ for the next iteration.

At the nth iteration, we apply one step of Newton method to minimize eq. (27) to yield

$$\left(\mathbf{G}_\zeta^T \mathbf{G}_\zeta + \mu \mathbf{W}_m^T \mathbf{W}_m + \lambda^{(n)} \mathbf{X}^{-2}\right) \Delta \mathbf{m} = -\mathbf{G}_\zeta^T \delta \mathbf{d} - \mu \mathbf{W}_m^T \mathbf{W}_m \delta \mathbf{m} + \lambda^{(n)} \mathbf{X}^{-1} \mathbf{e}, \quad (28)$$

where $\mathbf{X} = \mathrm{diag}\{\mathbf{m}_1, \ldots, \mathbf{m}_\mathrm{M}\}$, $\mathbf{e} = \{1, \ldots, 1\}$, $\delta \mathbf{d} = \mathbf{G}_\zeta \mathbf{m}^{(n-1)} - \mathbf{d}_{obs}$, and $\boldsymbol{\delta}\, \mathbf{m} = \mathbf{m}^{(n-1)} - \mathbf{m}_0$. The solution of the above equation yields the search direction $\boldsymbol{\Delta}\mathbf{m}$. It is then used to update the model by a reduced step length so that the new model remains positive. Thus

$$\mathbf{m}^{(n)} = \mathbf{m}^{(n-1)} + \gamma \beta \Delta \mathbf{m}, \quad (29)$$

where β is the maximum permissible step length and is given by

$$\beta = \begin{cases} 1, & \text{if } \Delta \mathbf{m} > 0 \\ \min\limits_{\Delta \mathbf{m}_j < 0} \frac{m_j^{(n-1)}}{|\Delta \mathbf{m}_j|}, & \text{otherwise} \end{cases}. \quad (30)$$

The parameter γ is prescribed to be within (0, 1), and a value very close to unity, for instance 0.99, has been used in the literature. However, we have found empirically that a slightly smaller value of γ seems to perform better for the current problem and our algorithm uses $\gamma = 0.925$. The barrier parameter is then updated by

$$\lambda^{(n+1)} = [1 - \min(\beta, \gamma)] \lambda^{(n)}. \quad (31)$$

The barrier iterations are continued until the value of λ is sufficiently small such that the barrier term has a negligible contribution to the

total objective function in eq. (27). By this stage, the model objective function and data misfit are not expected to change very much so we also use an additional termination criterion that the objective function is changing less than 1 per cent.

The remaining issues for a practical algorithm are the specification of the initial value of λ and the solution of the central equation given in eq. (28). Once an initial model $\mathbf{m}^{(0)}$ is chosen, the starting value of λ can be chosen so as to balance the barrier term with the sum of the remaining terms in eq. (27),

$$\lambda^{(0)} = \frac{\phi_d^{(0)} + \mu\phi_m^{(0)}}{2\sum_{j=1}^{M} \ln\left(m_j^{(0)}\right)}. \tag{32}$$

This choice has worked well in all our applications.

For the solution of the central eq. (28), we use a conjugate gradient (CG) technique. This is for the following two reasons. First, it is prohibitively expensive to form explicitly the matrix

$$\mathbf{A} = \mathbf{G}_\zeta^T \mathbf{G}_\zeta + \mu \mathbf{W}_m^T \mathbf{W}_m + \lambda^{(n)} \mathbf{X}^{-2} \tag{33}$$

in eq. (28), but it is relatively easy to apply $\mathbf{A}$ to a vector implicitly. The operation of multiplying $\mathbf{A}$ to a vector is dominated by the first term $\mathbf{G}_\zeta^T \mathbf{G}_\zeta$, since $\mathbf{W}_m^T \mathbf{W}_m$ and $\mathbf{X}^{-2}$ are extremely sparse. We have the sparse wavelet representation of the matrix $\mathbf{G}_\zeta$, and the computation of $\mathbf{G}_\zeta^T \mathbf{G}_\zeta \mathbf{v}_\kappa$ is accomplished by two matrix-vector multiplications followed by an inverse wavelet transform. The multiplication is carried out in the wavelet domain using eqs (15) and (17). Numerically, we have an equivalent sparse system to solve and the conjugate gradient method is a natural choice. Second, the

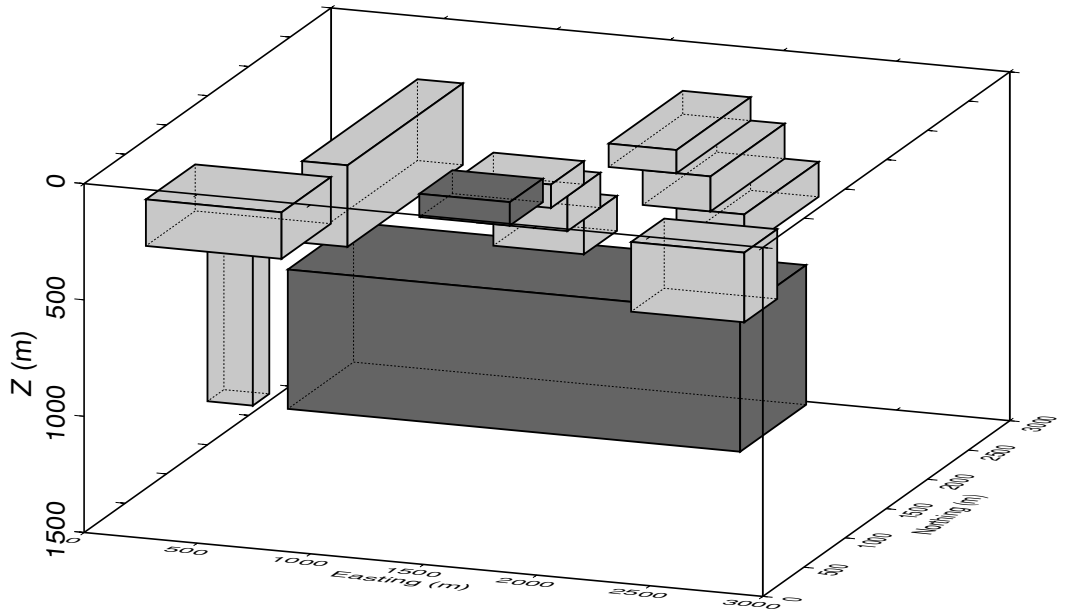

Figure 6. The perspective view of the test susceptibility model. Six smaller blocks are buried at shallow depth and one large block is buried at a greater depth. The lighter blocks have a susceptibility of 0.05 SI and the darker ones have a susceptibility of 0.08 SI.

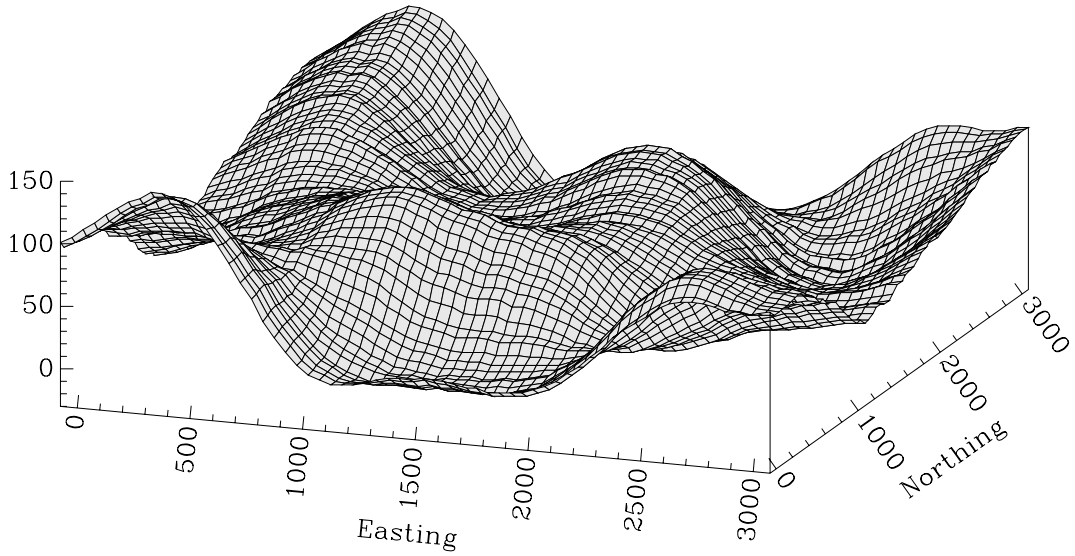

Figure 7. The 3-D perspective view of the surface topography over the 3-D model shown in Fig. 6. The total relief is 150 m. The top of the inversion mesh is placed at the elevation of 125 m as only a very few points on the surface are actually above 125 m. The depth indicated in the plotted model is referenced from that elevation.

solution of eq. (28) only gives an approximate search direction and it is unnecessary to solve it precisely. The CG method allows us to obtain a partial solution and thereby reduce the computational time. This results in a truncated Newton step for the barrier iteration. We generally terminate the CG solution when the relative residual is below 10^{-2} and this has worked satisfactorily. Increased CG accuracy does not seem to improve the solution very much but significantly increases the CPU time.

As barrier iterations progress, many model elements approach the zero bound. Thus the barrier component, $\lambda \mathbf{X}^{-2}$, in eq. (28) can cause the matrix $\mathbf{A}$ to be poorly conditioned and this makes the CG solver converge very slowly. We treat this difficulty by applying a Jacobi-like pre-conditioner to the CG solver. The diagonal pre-conditioner $\mathbf{C}$ consists of the square root of the diagonal elements of the matrix $\mathbf{A}$,

$$\mathbf{C} = \text{diag}\,\{\sqrt{A_{ii}}\}. \tag{34}$$

The pre-conditioner is updated at each barrier iteration. Numerical examples have shown that the pre-conditioning is necessary to have the CG converge in a small number of iterations. Without it, the number of required CG iterations increases rapidly as the final solution is approached.

Determination of regularization parameter

The choice of the regularization parameter μ ultimately depends upon the magnitude of the error associated with the data. The inversion of noisier data requires heavier regularization, thus a greater value of μ is needed. When the standard deviation associated with each datum is known, the data misfit defined by eq. (25) has a known expected value ϕ_d^*, which is equal to the number of data if the errors are assumed to be independent Gaussian noise with zero mean. The value of μ should be such that the expected misfit is achieved. This entails a line search based on the misfit curve as a function of μ.

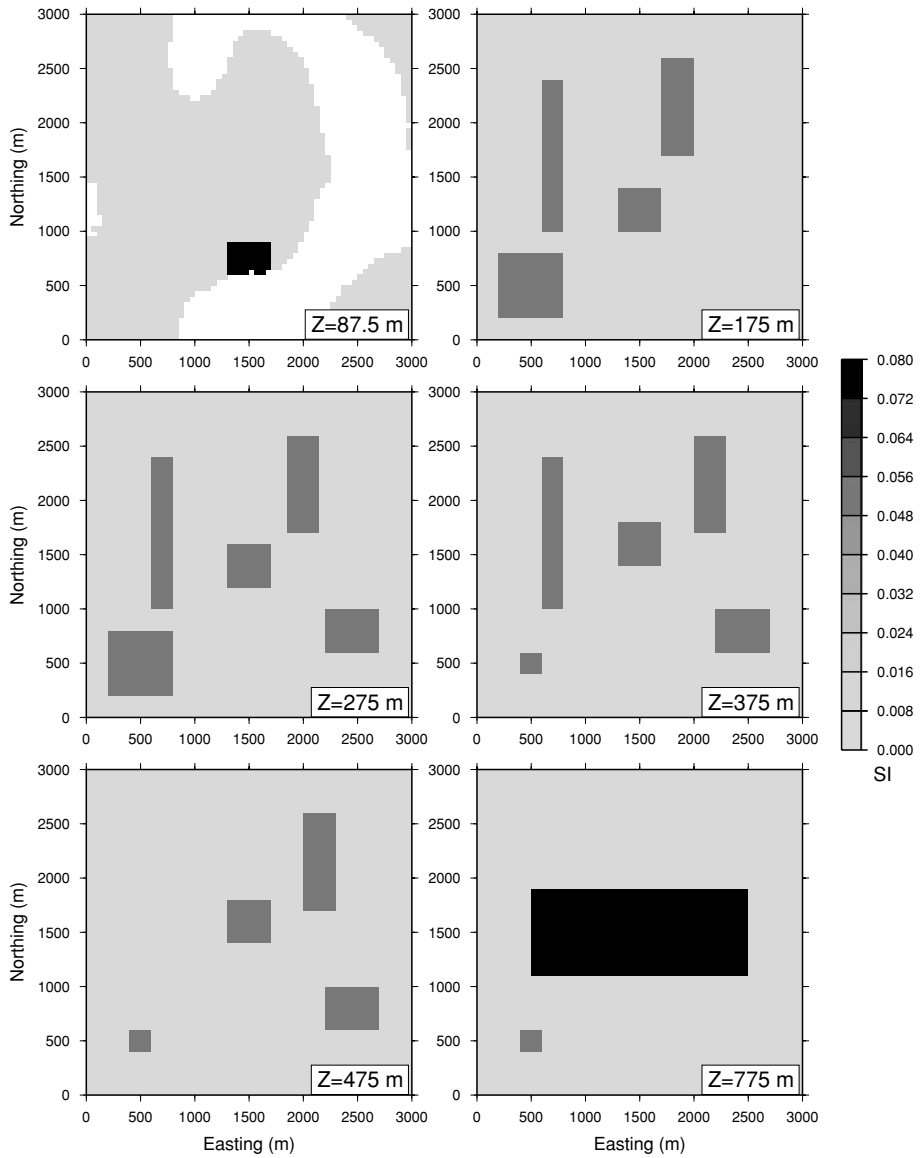

Figure 8. The true susceptibility model is displayed in six plan-sections. The depth of each section is indicated by the labels. The gray scale shows the susceptibility in SI units. A part of the surface topography can be seen in the section at $z = 87.5$ m .

Because of the positivity constraint, our problem is nonlinear. Thus for each μ a nonlinear solution using a logarithmic barrier method must be obtained. This is computationally demanding and we therefore have developed the following strategy to reduce the cost.

It is observed that, when plotted on a log-log scale, the misfit curves, $\phi_d(\mu)$, for 3-D magnetic inversion with, and without, positivity often parallel each other in the vicinity of the expected misfit. The curve with positivity lies above the curve without positivity. We proceed by first performing a line search without positivity to find a μ_0 that gives rise to ϕ_d^*. This search also generates the slope, s_0, of the misfit curve at μ_0. This process is very efficient and the required CPU time is much smaller compared to the time required for the solution with positivity. A rigorous line search incorporating positivity then starts with an initial guess of $\mu = 0.5\mu_0$. This usually yields a misfit that is very close to the target value. However, if the misfit is not sufficiently close to ϕ_d^*, a new guess for μ is obtained which makes use of the approximate slope s_0. The inversion with updated μ can be solved efficiently if the logarithmic barrier algorithm is started with an initial model that is close to the final solution. Here we use the current model but first perturb it so all elements are well away from the zero bound. Such 'warm starts' of the logarithmic barrier method also require an appropriate starting value for λ. We invoke eq. (32) with $m^{(0)}$ replaced by the perturbed model. If further iterations of a line search are demanded, then the same procedure is used except that an estimate of the slope of the misfit curve is obtained directly from optimizations, already carried out, that incorporate positivity. The line search using this strategy is often successful in reaching the target misfit ϕ_d^* after testing two to four values of μ.

In practical applications, the estimate of data error is often not available. The degree of regularization, hence the value of μ, needs to be determined based on other criteria. A commonly used method in linear inverse problems is the generalized cross-validation (GCV) technique. The use of GCV in inverse problems with inequality constraints such as positivity requires the solution of a number of auxiliary optimization problems, which requires a large amount of computation. However, we have observed that applying GCV to the 3-D magnetic inversion without positivity can produce a reasonable estimate of the data error and corresponding value of μ. That error can serve as a starting point for further adjustment by the user based on his or her judgement. If no other information is available, the value of μ obtained in this manner can be used directly in the final inversion that has the positivity imposed. In this case, only one logarithmic barrier solution is needed. Numerical tests have indicated that this simplistic use of GCV is in fact surprisingly effective unless the data have a large negative bias or are distributed too sparsely. In the following, we outline the implementation of an efficient method for evaluating the GCV function.

In the absence of positivity, the solution to the inverse problem is obtained by solving the following equation,

$$\left(\mathbf{G}_\zeta^T\mathbf{G}_\zeta + \mu\mathbf{W}_m^T\mathbf{W}_m\right)\Delta\mathbf{m} = -\mathbf{G}_\zeta^T\delta\mathbf{d}, \tag{35}$$

where $\delta\mathbf{d} = \mathbf{G}_\zeta\mathbf{m}_0 - \mathbf{d}_{obs}$. The corresponding expression for GCV is (Golub *et al.* 1979; Wahba 1990)

$$V(\mu) = \frac{\left\|\left[\mathbf{I} - \mathbf{G}_\zeta\left(\mathbf{G}_\zeta^T\mathbf{G}_\zeta + \mu\mathbf{W}_m^T\mathbf{W}_m\right)^{-1}\mathbf{G}_\zeta^T\right]\mathbf{d}\right\|^2}{\left\{N - \text{trace}\left[\mathbf{G}_\zeta\left(\mathbf{G}_\zeta^T\mathbf{G}_\zeta + \mu\mathbf{W}_m^T\mathbf{W}_m\right)^{-1}\mathbf{G}_\zeta^T\right]\right\}^2}, \tag{36}$$

where N is the number of data. The numerator is the data misfit obtained when solving the eq. (35). Since the inverse $(\mathbf{G}_\zeta^T\mathbf{G} + \mu\mathbf{W}_m^T\mathbf{W}_m)^{-1}$ can be obtained by using a CG solver and fast matrix multiplication, the numerator can be evaluated very quickly. The more difficult task is to evaluate the trace of the term in the brackets to obtain the value of the denominator. This is carried out by using the stochastic trace estimator of Hutchinson (1990), which states that an unbiased estimate of the trace of a matrix $\mathbf{A}$ is given by

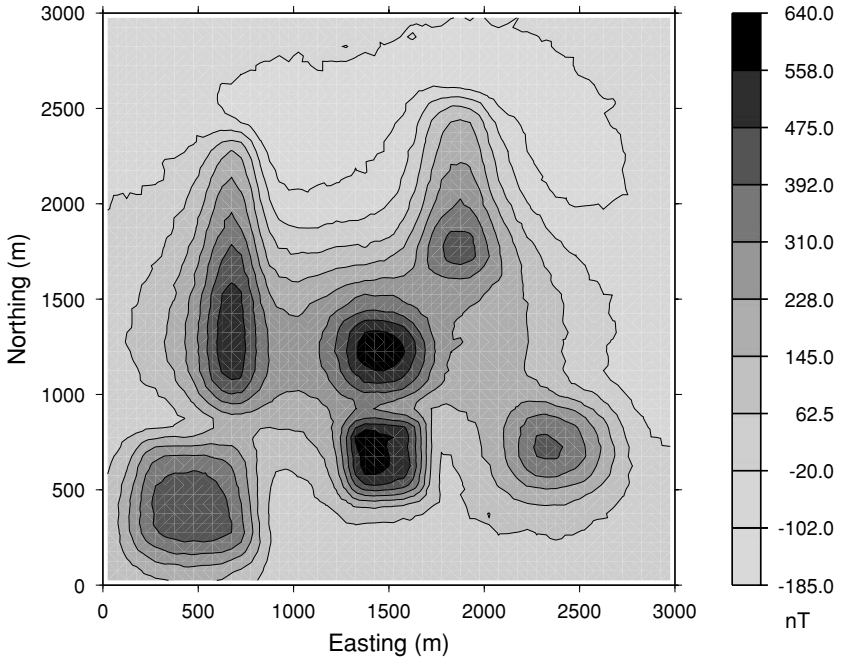

Figure 9. The total field anomaly produced by the susceptibility model shown in Fig. 8. The inducing field direction is $I = 65^\circ$ and $D = 25^\circ$. The data are simulated over an undulating surface that parallels the topography with a constant terrain clearance of 75 m. Uncorrelated Gaussian noise with a standard deviation of 5 nT is added to the data. The gray scale shows the field in nT.

$$\text{trace}\,(\mathbf{A}) = \mathbf{u}^T\mathbf{A}\mathbf{u}, \tag{37}$$

where $\mathbf{u}$ is a random vector of -1 and 1 each having a probability of 0.5. The GCV function is then approximated by

$$V(\mu) \simeq \frac{\left\|\left[\mathbf{I} - \mathbf{G}_\zeta\left(\mathbf{G}_\zeta^T\mathbf{G}_\zeta + \mu\mathbf{W}_m^T\mathbf{W}_m\right)^{-1}\mathbf{G}_\zeta^T\right]\mathbf{d}\right\|^2}{\left\{N - \mathbf{u}^T\mathbf{G}_\zeta\left(\mathbf{G}_\zeta^T\mathbf{G}_\zeta + \mu\mathbf{W}_m^T\mathbf{W}_m\right)^{-1}\mathbf{G}_\zeta^T\mathbf{u}\right\}^2}. \tag{38}$$

Therefore, the evaluation of $V(\mu)$ for each value of μ is equivalent to inverting eq. (35) two times with different right-hand sides. The first inversion is applied to the data vector $\mathbf{d}$ and the second to the random vector $\mathbf{u}$. Finding the μ that minimizes the GCV function entails a line search, and the process can be sped up by starting the CG solver using, as the initial model, the respective models from the solutions for the previous value of μ.

NUMERICAL EXAMPLE

We now apply our fast algorithm to a synthetic example. The model consists of seven different magnetic bodies embedded in a non-susceptible background beneath a topographic surface. Among them, six are relatively small bodies having different sizes and shapes. They are buried at different depths to simulate small scale anomalies. One large body is buried at a greater depth below the small bodies to generate a broad anomaly over which the smaller anomalies are superimposed. The total relief of surface topography is 150 m over a 3 km by 3 km area. Fig. 6 shows a perspective view of the model geometry. Fig. 7 is a perspective view of the surface topography. The elevation varies mostly between 0 and 125 m, with a few points reaching 150 m. Fig. 8 shows six plan sections of the model. We have generated total-field anomaly data above the

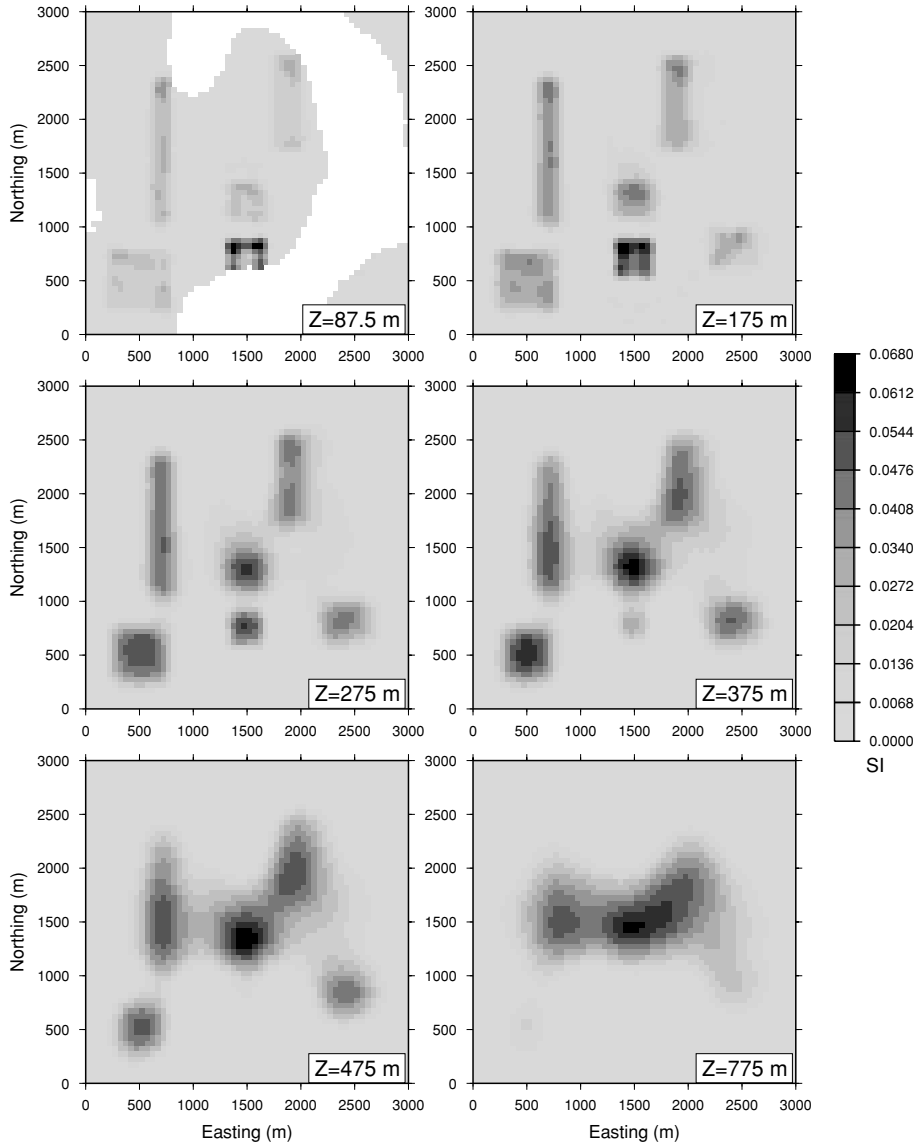

Figure 10. Recovered susceptibility model is displayed in six plan-sections. The depth of each section is shown by the label. This figure is to be compared with the true model in Fig. 8. The six shallow prisms and one large prism at depth are all imaged.

surface assuming an inducing field direction of $I = 65^\circ$ and $D = 25^\circ$. The data are located at a constant terrain clearance of 75 m on a 60 by 60 grid of 50-m spacings, which gives rise to a total of 3600 data. Fig. 9 shows the noise-contaminated data. The standard deviation is 5 nT for all data. The observed data indicate the presence of six shallow anomalies, but there is little indication about the deep source.

For the inversion, we use a mesh occupying a volume of 3.2 km by 3.2 km by 1.5 km. The top of the mesh is placed at the elevation of 125 m. The cell width is 50 m in both horizontal directions and the thickness varies from 25 m near surface to 100 m at the bottom. After the surface topography is discretized onto the mesh, the resulting model contains a total of 110 000 cells. Storage of the entire sensitivity matrix, requires more than 1500 Mb. Clearly, solving a problem of this size by the direct approach is beyond the capability of many workstations currently available.

To perform the inversion using the fast algorithm, we first generate the sparse representation of the sensitivity matrix using the Daubechies-4 wavelet. The depth weighting function $\zeta(\mathbf{r})$ is used and the required relative accuracy for the reconstructed sensitivity is 5 per cent. With these choices, a compression ratio of 76 is achieved and the transformed sensitivity matrix is stored with less than 45 Mb. The high compression ratio is achieved because the magnetic kernels become smoother as observation height increases, so fewer wavelet coefficients are needed for a good reconstruction. The model objective function is specified by choosing $\alpha_s = 0.0001$ and $\alpha_x = \alpha_y = \alpha_z = 1.0$, and a zero reference model. We start the inversion using a constant initial model of 0.001 and set the target misfit to the expected value of 3600. The recovered susceptibility model is displayed in Fig. 10 in six plan-sections. Comparison with Fig. 8 shows that the different anomalies in the true model, including the deep prism, are well-imaged. However, the more interesting aspect of this inversion is its small demand on computing resources. The inversion uses 60 Mb of memory, and is completed in 150 minutes on a SUN Sparc20 workstation.

Next, we apply the algorithm to a set of field data. Since the inversion methodology for constructing 3-D susceptibility has been published in previous publications, and the emphasis of the current work is on numerical efficiency, we use as an example the ground magnetic data from Mt Milligan copper-gold porphyry deposit in central British Columbia. A subset of this data set was used in Li & Oldenburg (1996), and we re-invert it here to provide a comparison with the previously published result so the adequacy of the present inversion strategy can be established.

The host rocks for the deposit are early Mesozoic volcanic and sedimentary rocks and contain intrusive monzonitic rocks that have accessory magnetite. The copper and gold are known to be concentrated in the potassic alteration assemblage, which is mainly around the contact of the monzonite intrusions and may extend outward and into fractured volcanic rocks. Among other minerals, magnetite is one of the strong indicators of the potassic alteration. Thus the magnetic inversion is expected to recover high susceptibility in the monzonite stock and in the regions of intense alteration. Readers are referred to Oldenburg *et al.* (1997) for more detail. The study was concentrated in an area of 1.2 km by 1 km, which covers a large monzonite body known as the MBX stock and contains a reasonably isolated set of magnetic anomalies. The reduced magnetic anomaly is shown in a contour map in Fig. 11. There are 2009 data points located at a 25-m intervals in both horizontal directions. The direction of the inducing field is $I = 75^\circ$ and $D = 25.73^\circ$. Each datum is assumed to have an error whose standard deviation is equal to 5 percent of its magnitude plus 10 nT.

To invert these data, we use a model mesh that is horizontally larger than the data area and coincides at the top with the highest point on the topographic surface, and extends to 450 m depth. The mesh has a cell width of 25 m beneath the area of data, and the cell thickness varies from 12.5 m near the surface to 25 m at depth. This results in a mesh with $52 \times 44 \times 22$ cells. Once the mesh is defined, the topography is discretized onto it. The 43 428 cells below this surface define the susceptibility model. For the purpose of comparison, we have carried out the inversion with, and without, the wavelet compression of the sensitivity matrix. For wavelet compression, we use Daubechies-4 wavelet, and a reconstruction accuracy of 5 per cent. The achieved compression ratio is 17. The dense sensitivity

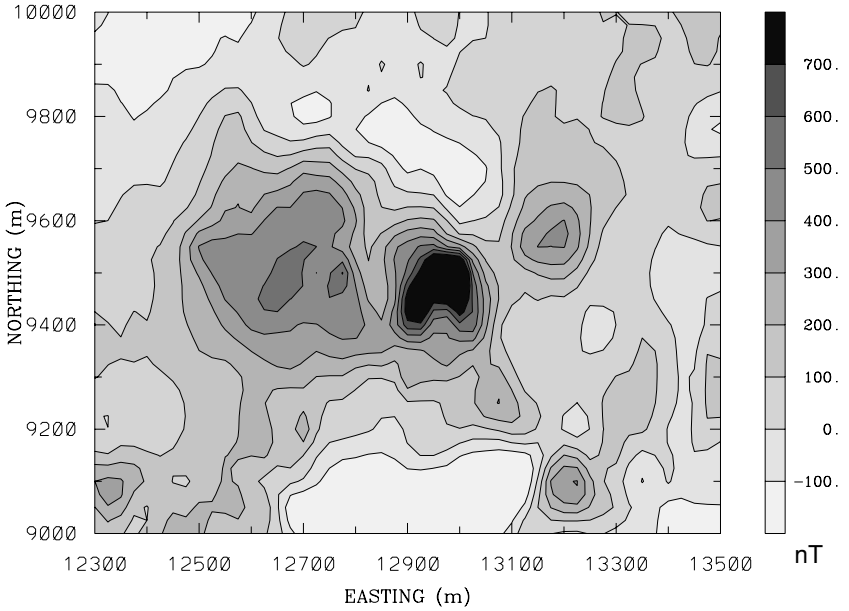

Figure 11. The anomalous magnetic data at Mt Milligan copper–gold porphyry deposit. The data are on a 25 m by 25 m grid and at a terrain clearance of 20 m. The inducing field direction is at $I = 75^\circ$ and $D = 25.73^\circ$.

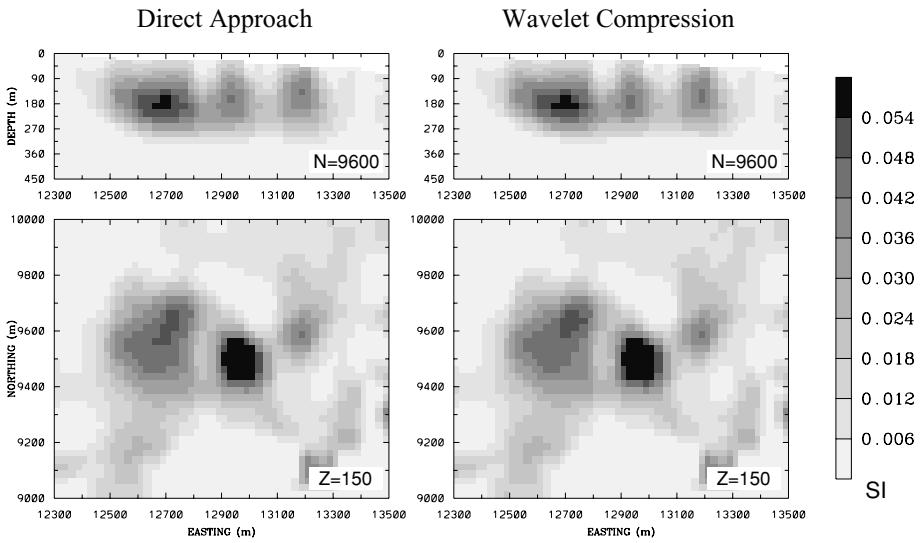

Figure 12. Comparison of recovered susceptibility models from inversions with, and without, wavelet compression of the sensitivity matrix. Both models are shown in one cross-section and one plan-section. The model on the left is produced by the inversion without wavelet compression, and the model on the right is recovered by using the compressed sensitivity matrix. There is little difference between the two results.

matrix requires 335 Mb to store while the compressed matrix requires only 39 Mb. The CPU time required for inversion, when the dense matrix is used, is 20 times that required when the compressed matrix is used. This represents significant savings in computing resources. Fig. 12 compares the two models in cross-section at $N =$ 9600 m and in the plan-section at $Z = 150$ m. The two models are virtually identical. There are only minor differences between the two models and those occur in regions of low susceptibility away from the main anomalies. It is clear that the model recovered using the compressed matrix has not lost any information that might affect the final interpretation, yet the savings in the computing resources are significant.

CONCLUSION

We have developed a fast algorithm for inverting large-scale magnetic data based on wavelet compression and logarithmic barrier method of minimization. The large, dense sensitivity matrix is compressed by applying a 3-D wavelet transform to each row and thresholding the resultant transformed matrix. This yields a sparse representation of the sensitivity which, in turn, allows reduced memory requirements for storage and reduced operation counts for matrix-vector multiplications. This produces an efficient forward mapping. We have also adapted the logarithmic barrier method to solve the inverse problem with positivity constraints. The fast matrix-vector multiplication makes it possible to use conjugate gradient techniques as the central solver for the barrier method. This combination of solution strategy translates the savings in the forward mapping directly to the speed of inverse solution. As a result, the size of magnetic inverse problems that can be solved is greatly increased and the required CPU time for solution is reduced significantly. This has had a significant and positive impact upon the interpretation of field data sets.

ACKNOWLEDGMENTS

This work was supported by an NSERC IOR grant and an industry consortium 'Joint and Cooperative Inversion of Geophysical and Geological Data'. Participating companies are Placer Dome, BHP Minerals, Noranda Exploration, Cominco Exploration, Falconbridge, INCO Exploration & Technical Services, Hudson Bay Exploration and Development, Kennecott Exploration Company, Newmont Gold Company, WMC Ltd, and CRA Exploration Pty.

REFERENCES

Beylkin, G., Coifman, R. & Rokhlin, V., 1991. Fast wavelet transforms and numerical algorithms I, *Comm. Pure and Appl. Math.,* **44,** 141–183.

Beylkin, G., 1992. On the fast algorithm for multiplication of functions in the wavelet bases, International Conference on Wavelets and applications, Toulouse. Proceedings, eds Meyer Y. & Roques S., Editions Frontieres.

Beylkin, G., 1993. Wavelets and fast numerical algorithms, *Proceedings of Symposia in Applied Mathematics,* **47,** 89–117.

Daubechies, I., 1988. Orthonormal bases of compactly supported wavelets., *Comm. Pure and Appl. Math.,* **41,** 909–996.

Daubechies, I., 1992. *Ten lectures on wavelets,* SIAM, Philadelphia, PA.

Gill, P.E., Murray, W., Ponceleon, D.B. & Saunders, M., 1991. Solving reduced KKT systems in barrier methods for linear and quadratic programming. Technical Report SOL 91–7, Stanford University, Stanford, CA.

Golub, G.H., Heath, M. & Wahba, G., 1979. Generalized cross-validation as a method for choosing a good ridge regression parameter, *Technometrics,* **21,** 215–223.

Harten, A. & Yad-Shalom, I., 1994. Fast multiresolution algorithms for matrix-vector multiplication, *SIAM J. Numer. Anal.,* **31,** 1191–1218.

Hutchinson, M.F., 1990. A stochastic estimator of the trace of the influence matrix for Laplacian smoothing splines, Communications in Statistics, Simulation and Computation, **19,** 443–450.

Li, Y. & Oldenburg, D.W., 1996. 3D inversion of magnetic data, *Geophysics,* **61,** 394–408.

Li, Y. & Oldenburg, D.W., 2000. Joint inversion of surface and three-component borehole magnetic data, *Geophysics,* **65,** 540–552.

Mallat, S., 1989. A theory for multiresolution signal decomposition: the wavelet representation, *IEEE Trans. PAMI,* **11,** 674–693.

Meyer, Y., 1993. *Wavelets: Algorithms and applications,* SIAM, Philadelphia, PA.

Oldenburg, D.W., Li, Y. & Ellis, R.G., 1997. Inversion of geophysical data over a copper–gold porphyry deposit: A case history for Mt Milligan, *Geophysics,* **62,** 1419–1431.

Pilkington, M., 1997. 3-D magnetic imaging using conjugate gradients, *Geophysics,* **62,** 1132–1142.

Saunders, M., 1995. Cholesky-based methods for sparse least squares: The benefits of regularization. Technical Report SOL 95–1, Stanford University, Stanford, CA.

Wahba, G., 1990. Spline models for observational data, CBMS-NSF, regional Conference Series in Applied mathematics, **59,** SIAM, Philadelphia, PA.

Wright, S.J., 1997. Primal-dual interior-point methods, SIAM, Philadelphia, PA.