Dynamic level set regularization for large distributed parameter estimation problems

K. van den Doel* U. M. Ascher†

March 30, 2007

Abstract

This article considers inverse problems of shape recovery from noisy boundary data, where the forward problem involves the inversion of elliptic PDEs. The piecewise constant solution, a scaling and translation of a characteristic function, is described in terms of a smoother level set function. A fast and simple *dynamic regularization* method has been recently proposed that has a robust stopping criterion and typically terminates after very few iterations. Direct linear algebra methods have been used for the linear systems arising in both forward and inverse problems, which is suitable for problems of moderate size in 2D.

For larger problems, especially in 3D, iterative methods are required. In this article we extend our previous results to large scale problems by proposing and investigating iterative linear system solvers in the present context. Perhaps contrary to one's initial intuition, the iterative methods are particularly useful for the inverse rather than the forward linear systems. Moreover, only very few preconditioned conjugate gradient iterations are applied towards the solution of the linear system for the inverse problem, allowing the regularizing effects of such iterations to take center stage. The efficacy of the obtained method is demonstrated.

Keywords: Inverse problem, Dynamic regularization, Level set, Electrical impedance tomography, DC resistivity.

MSC classifications: 65N21 68U20 65K10

*Department of Computer Science, University of British Columbia, Vancouver, BC, V6T 1Z4, Canada. (kvdoel@cs.ubc.ca). Supported in part under NSERC Discovery Grant 84306.

†Department of Computer Science, University of British Columbia, Vancouver, BC, V6T 1Z4, Canada. (ascher@cs.ubc.ca). Supported in part under NSERC Discovery Grant 84306.

1 Introduction

The recovery of a distributed parameter function with discontinuities from inverse problems with elliptic forward PDEs is fraught with theoretical and practical difficulties [25, 4]. And yet there are many applications that give rise to such problems, including DC resistivity [33], linear potential problems [24, 9], magnetotelluric inversion [30], diffraction tomography [13], electrical impedance tomography (EIT) [6, 11, 7], and Maxwell's equations in low frequencies [26, 27, 19, 20]. In most of these references the parameter function to be recovered, typically related to conductivity, is assumed smooth. This enables a reasonably stable practical reconstruction of regularized solutions. But when discontinuities are allowed – indeed expected in the distributed parameter model – the reconstruction problem, especially in the presence of noise in data measurements, becomes very difficult [3, 4]. Fortunately, it is often reasonable to assume further that the solution may take on at each point only one of two values, thus yielding a shape recovery problem. Better results are then obtained in a much more stable fashion using algorithms that take this a priori information directly into account [1, 8, 9, 12, 10, 37, 14, 16, 17, 32].

To be more specific, consider the following data inversion problem. A number of *forward operators*, $F_k(m)$, $k = 1, \ldots, s$, are given and a *model* $m(\mathbf{x})$ is sought over a discretized domain Ω in 2D or 3D, such that each $F_k(m)$ matches given data b_k up to the noise level in the data measurements. Each forward operator $F_k(m)$ corresponds to a particular experimental setup, and each data vector b_k represents the measurements taken in that experiment. The forward model is further given by

$$F_k(m) = Q_k u_k, \tag{1a}$$

$$u_k = G_k(m), \tag{1b}$$

where Q_k is a matrix which projects the field u_k to data locations (e.g. along the boundary $\partial\Omega$), and $G_k(m)$ is the inverse of an elliptic PDE system discretized on a grid at least as fine as that of the model m using a finite volume or finite element method. For the EIT and DC resistivity problems we write this PDE discretization as

$$A(m)u_k = q_k, \tag{2}$$

where A is a large, sparse matrix, q_k are given and represent the external currents injected into the system, and $G_k(m) = A(m)^{-1}q_k$ [18, 2, 20]. Importantly, we assume in addition that there are two known values, m_I and m_{II}, corresponding say to a homogeneous body and a homogeneous background, such that at each point $\mathbf{x} \in \Omega$

$$m(\mathbf{x}) = m_I \text{ or } m(\mathbf{x}) = m_{II}.$$

A natural approach for incorporating the information that m describes a shape is to use a differentiable level set function $\psi(\mathbf{x})$ such that $m = \chi(\psi)$. The function χ is

a grid-smoothing of the characteristic function based on the values that m may take, and the discontinuities in m occur across the 0-level set of $\psi(\mathbf{x})$ [37]. Note that the elements of the matrix $\chi' = \frac{\partial m}{\partial \psi}$ vanish away from a narrow vicinity of the interface.

A Tikhonov-type regularization [15, 35] now reads

$$\min_{\psi} \phi(\beta) \quad = \quad \frac{1}{2} \sum_{k=1}^{s} \|F_k(m) - b_k\|^2 + \beta R(\psi), \tag{3}$$

where R is a regularization functional and $\beta > 0$ is a parameter. In [37] we have proposed the regularization term

$$R(\psi) \equiv \hat{R}_1 = \frac{1}{2} \left[\int_{\Omega} (1 - |\nabla \psi|^2) d\mathbf{x} \right]^2 \tag{4a}$$

(or, more precisely, a discretization of this on the grid of m) and showed that it performs better for the 2D sample problems considered there than the more obvious choice

$$R(\psi) \equiv \hat{R}_2 = \frac{1}{2} \int_{\Omega} |\nabla \psi|^2 d\mathbf{x}. \tag{4b}$$

Furthermore, we have introduced in [37] a *dynamic regularization* method that performs significantly better than the above Tikhonov-type method. Starting from an initial guess ψ_0, for $n = 0, 1, 2, \ldots$ the iterations read

$$(\sum_{k=1}^{s} \hat{J}_k^T \hat{J}_k + \beta_0 X)\delta\psi \quad = \quad -\sum_{k=1}^{s} \hat{J}_k^T (F_k(m(\psi_n)) - b_k), \tag{5a}$$

$$\psi_{n+1} \quad = \quad \psi_n + \tau \delta\psi, \tag{5b}$$

where $\hat{J}_k = J_k \chi'$, $J_k = \frac{\partial F_k}{\partial m}$, $\beta_0 > 0$ is a small constant and $0 < \tau \leq 1$ is a step size that is determined by line search for the objective function defined by the data fitting term $\phi(0)$ in (3). The matrices in (5a) are all evaluated at ψ_n. We choose $X = R''$, the Hessian of a regularization operator $R(\psi)$. Specifically, let L be a positive definite matrix based on the standard discrete Laplacian. Then $\hat{R}_2'' = L$ and $\hat{R}_1'' = 2(\psi^T L \psi - |\Omega|)L + 4(L\psi)(L\psi)^T$.

One practical advantage of this method is that typically ϕ decreases consistently until some point where the iteration starts stalling, whereby the solution process can be reasonably stopped. Moreover, if the method converges then typically very few iterations are required, see [37].

In general, theory for level set methods for such inverse problems as considered here is currently in an unsatisfactory state. The iteration (5) may therefore admit different interpretations [5]. The one that we have found most useful, especially for extending the method to handle very large problems, is the optimization point of view,

where we set $\beta = 0$ in (3). The Gauss-Newton iteration for the output least squares method then involves a singular matrix, so it is regularized Levenberg-Marquardt fashion, but with the twist that the regularization matrix is $X = R''$, not simply the identity. This matrix can further be viewed as a preconditioner, as elaborated upon below.

On the grids we have experimented with in [37], standard direct sparse solvers are very fast. For the inverse problem system in 2D often this is all that is required [7, 12]. An important advantage of direct methods is that they are relatively hassle-free and easy to implement (especially in MATLAB); for instance, the dependence of the condition number of the matrix in (5) on β_0 is less crucial than it could be if iterative methods were utilized to solve this system *accurately*. However, for larger problems, including forward problems in 3D, we cannot always expect to use direct methods. The present paper is devoted to the development and study of iterative methods for large size problems.

Thus, in Section 2 we develop appropriate iterative methods for the forward problem (2), with an eye towards their invocation within the inverse solution process. This context requires the solution of many linear PDE discretizations with the same matrix, which in turn affects the choice of numerical method.

In Section 3 we then turn to the solution of the linear systems arising in (5a). It is important to realize that these systems arise within the context of an outer, nonlinear iteration which incorporates regularization. The regularization properties of conjugate gradient (CG) and steepest descent (SD) methods have long been recognized [21, 23]. Applying a few CG iterations to the singular system (5a) with $\beta_0 = 0$ yields the desired effect.

In Section 4 we briefly describe a simple method based on thresholding a Tikhonov-type regularization. This is a benchmark approach: our level set methods must beat it in order to be worthwhile.

Numerical experiments of shape recovery in 3D are reported in Section 5. Conclusions are offered in Section 6.

2 Iterative solvers: forward problem

Large, sparse linear systems arise both in (2) and in (5a). Forward problems may well require finer discretization grids, but fast iterative techniques for elliptic PDEs such as (2) are well known. The efficient iterative solution of the inverse linear system (5a) is much more challenging, especially in the absence of χ', and it is needed for larger problem instances, especially in 3D (see e.g. [20, 18, 9]).

We potentially have *three* levels of iteration. The *outer iteration* of the dynamic regularization method is described by (5). The linear system (5a) which has to be solved at every outer iteration may itself be solved by an iterative method. We call this the *inner iteration*. The preconditioned CG method (PCG) we advocate in Section 3 is such a scheme. Finally, at each PCG iteration we have to solve (2) and

a preconditioner. If this is done iteratively we shall refer to it as the *inner-inner iteration*.

An obvious modification of the all-direct methods used in [37] would be to solve (2) iteratively, using a PCG, while applying a direct method for (5a) which, at least in 2D, is not a prohibitively large system; see Section 4.1 of [37]. However, the problem with such an approach is that when forming the matrix $\hat{J}_k$, several forward problems (2) and their adjoints for different right hand sides must be solved, making the formation of the matrices in system (5a) rather expensive.

Specifically, writing $\hat{J}_k = -Q_k A^{-1} B_k \chi'$, where $B_k = \frac{\partial A(m)u_k}{\partial m}$ (cf. [18]), we can form $\hat{J}_k$ either column-wise or row-wise through $\hat{J}_k^T$. The number of non-zero columns in $\hat{J}_k$ is proportional to the number of grid points p_1 "near" the interface boundary, where nearness is defined by appropriately truncating χ', which in turn is a grid smoothing of a Dirac delta function. On the other hand, using the adjoint approach there are inversions of A for each column of Q_k^T. Let p_2 be the number of rows in Q_k and $p = \min(p_1, p_2)$. It then follows that, including the adjoints, a total of at least $2sp$ forward problems must be solved at each outer iteration of (5a).

Because all these forward problems only differ in their right hand sides, this can be done effectively using a standard direct method that decomposes A only once, followed by a forward-backward substitution for each right hand side. In contrast, using iterative solution methods for the forward problems (the inner-inner iteration) in a way that shares information among the $2sp$ solves is difficult if not impossible here.

On the other hand, there are advantages to solving (5a) by an iterative method, rather than a direct method. At the nth outer iteration in (5a), for each inner iteration of an iterative solver we only need to compute the products of $\hat{J}_k$ and $\hat{J}_k^T$ with a vector once, which involves solving (2) and its adjoint once for a total of $2s$ forward solves. In practice, to be discussed more fully below, the number of outer iterations r required to solve (5a) is quite small, so the number of forward solves required now is only $2sr$, which is much less than when solving (5a) directly. Note further that, in many practical applications, the number of experiments s is still large (40 or more is not uncommon). A direct method for the forward problem (2) that decomposes A only once per outer iteration of (5a) will outperform any (inner-inner) iterative solution method for (2) in the large s limit, provided there is enough fast access computer memory available.

For sufficiently large problems that easily occur in 3D, an iterative solution for the forward problem (2) becomes necessary because of memory considerations. For the EIT / DC resistivity problem in 3D with which we experiment here, we use BICGSTAB [1] combined with an incomplete LU preconditioner with a drop tolerance

[1] In our implementation, we use a version of the code of [31] which employs a formulation where the forward problem matrix A is not symmetric on a non-uniform grid. It is not difficult to symmetrize A, which would in turn inspire a switch from BICGSTAB to CG; however, such a modification is not within our present focus of attention.

δ. The ILU preconditioner can be shared over the $2sr$ solutions with different right hand sides. In light of the above discussion, the drop tolerance δ should be chosen to be as small as available computer memory allows, at least for applications with a large number s of data sets.

3 Iterative solvers: inverse problem

The linear system (5a) describes a stabilized nonlinear Gauss-Newton iteration. If it is solved by an iterative method then an inner iteration is defined. An obvious question is then, how accurate does this inner iteration process need to be?

In the current context, in particular, the linear system (5a) contains the regularization term $\beta_0 X$. But it is well known that applying only a few preconditioned conjugate gradient (PCG) inner iterations towards the solution of the linear system (5a) has a regularizing effect [23, 21, 15, 22]. We therefore set $\beta_0 = 0$ and apply PCG inner iterations directly to the singular system.

The approach is reminiscent of Steihaug's trust region method [28, 34]. Note, however, that in the context of a singular system a textbook trust region approach does not apply. Moreover, experiments indicate that we must precondition this CG iteration to obtain a fast procedure.

An effective preconditioner is the discrete Laplacian, $X = L$. This is directly related to the Hessians of both $\hat{R}_1$ and $\hat{R}_2$, so the difference between the regularization functionals of (4), so pronounced when the system (5a) is solved accurately [37], is now diminished.

Note that L^{-1} is a smoother. Starting the PCG iteration for $\delta\psi$ from a zero guess, the quantity $\|L^{1/2}\delta\psi\|$ increases monotonically [28]. However, attempting to use this fact to control the iteration has not proved very effective in the current context of an inner iteration within an outer, nonlinear procedure. Indeed, our experiments consistently suggest that applying three PCG iterations for each n in (5a) is a happy compromise. In Section 5, Experiment 2, we demonstrate that many more inner iterations may require reinstating $\beta_0 X$, while fewer inner iterations than 3 result in the necessity for more outer iterations.

To compute the effect of the preconditioner (i.e., solve $Lv = q$ for any given vector q), we follow [31] and use a CG method with an SSOR preconditioner, which is an inner-inner iteration. This can be improved, especially if there is enough memory, noting that L is a constant matrix independent of the outer as well as the inner iterations. However, the calculation of the preconditioner takes less than 1% of the computation time for the problems we have considered here, which is why we have not pursued this any further.

4 Thresholding Tikhonov

If we are not allowed to make the shape reconstruction assumption that at each point $\mathbf{x}$ the model m may only take on one of two values, then level set methods do not apply. More conventional direct reconstructions of $m(\mathbf{x})$ may then be obtained using a Tikhonov regularization of the form

$$\min_m \phi(\beta) \quad = \quad \frac{1}{2}\sum_{k=1}^{s} \|F_k(m) - b_k\|^2 + \beta R(m), \tag{6a}$$

with a regularization term

$$R(m) = \frac{1}{2}\int_\Omega |\nabla m|^2 d\mathbf{x}. \tag{6b}$$

Next, if we reinstate the additional a priori information leading to shape reconstruction then the solution of (6) may be used for two purposes. The first is to provide a simple alternative method to our level set procedure. Specifically, we estimate the shape boundary by thresholding the continuous function $m(\mathbf{x})$, using the harmonic average of the maximum and minimum values of $m(\mathbf{x})$ in order to assign either m_I or m_{II} at each point $\mathbf{x}$. In practice, we have found the harmonic average to give much better results than an arithmetic average. Below we refer to this procedure as the L_2-method.

The second purpose is to use the L_2-method in order to provide an initial guess for the level set method.

5 Numerical experiments

In two dimensions, although all-direct methods can be applied, we can speed up the iteration (5) somewhat for larger grids. For the EIT problem considered in [37], solving only (5a) iteratively while solving the forward problems with a direct method is about twice faster than a direct solve on a 128^2 grid; on a 64^2 grid the direct and iterative methods are about equally fast.[2]

To experiment with larger 3D problems, we have implemented a level set "plug-in" to RESINVM3D [31], a MATLAB package for inverting DC resistivity and electrical resistivity tomography data in 3D. Our level set function implementation is as described in [37], except that we allow for non-uniform grids. At a grid cell we take the sharpening function χ to be

$$\chi(\xi) = \frac{m_I - m_{II}}{2}\tanh(\xi/h_{\text{av}}) + \frac{m_I + m_{II}}{2}, \tag{7}$$

[2]Note that this is not an entirely fair comparison, as we use the highly optimized MATLAB backslash function for the direct solve. However, the point is that in this range of problem sizes, a direct method is entirely satisfactory and automatic, and this allows us to swiftly move on to other challenges.

where $h_{\text{av}} = dV^{\frac{1}{3}}$ with dV the cell volume and ξ a dummy variable.

For our numerical experiments we have constructed artificial models and data. Gaussian noise was added to each individual artificial datum b according to

$$b \to b(1 + \eta * \epsilon),$$

where ϵ is a standard normal random variable and η is the noise level. As it is not desirable to reconstruct beyond the accuracy of the data noise it is useful to define a *misfit* for a given reconstruction ψ as

$$misfit = \sqrt{\frac{\sum_{k=1}^{s} \|F_k(m(\psi)) - b_k\|^2}{\sum_{k=1}^{s} \|F_k(m(\psi))\|^2}}.$$

Any reconstruction with a misfit smaller than η will model noise rather than data.

We have performed five different experiments to test the methods described above. The first three experiments are based on realistic geophysical surveys, adapted from [31]. The data was computed on a finer grid (64^3 or 128^3) than the grid used for the reconstruction, in order to avoid so-called "inverse crimes".

The forward problem (2) for all our experiments is defined through

$$\begin{aligned} \nabla \cdot (e^m \nabla u_k) &= q_k, \quad k = 1, \ldots, s, \\ \frac{\partial u_k}{\partial \nu}|_{\partial\Omega} &= 0, \end{aligned} \tag{8}$$

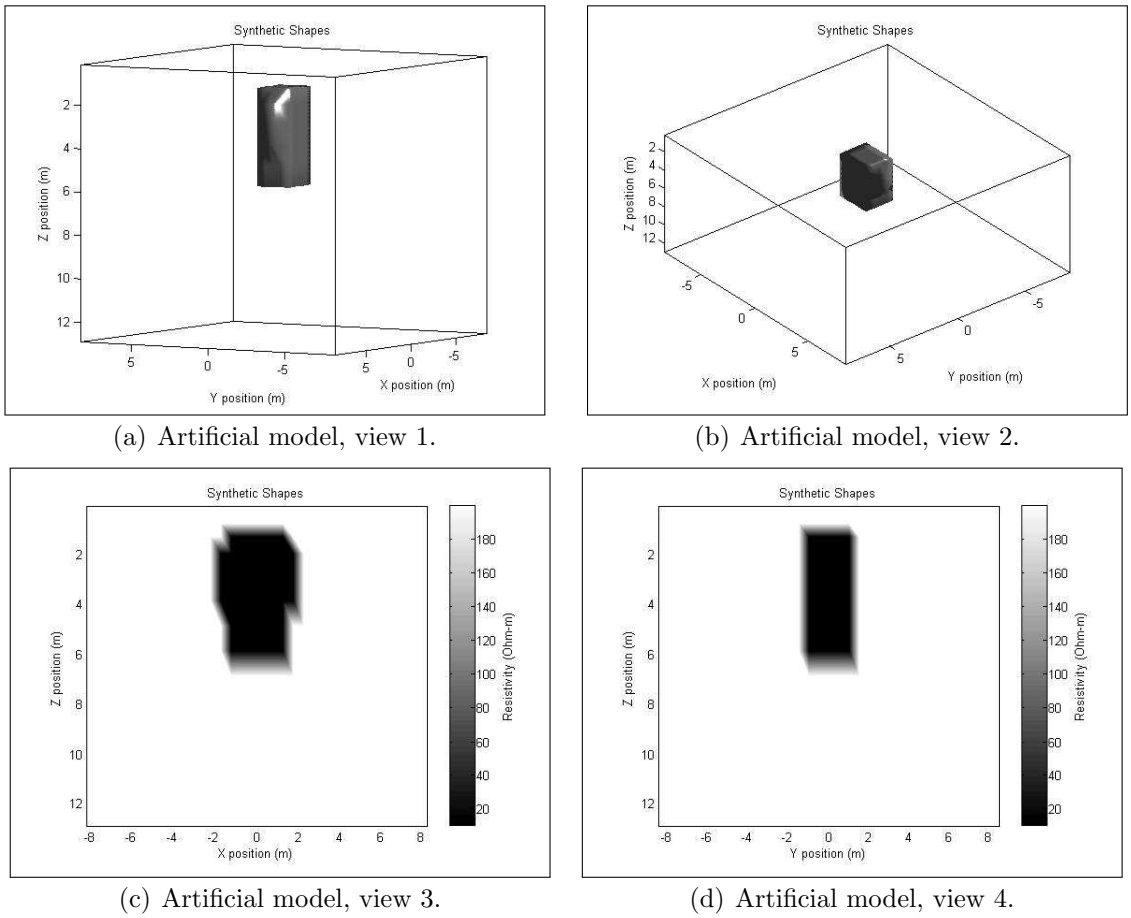
which is discretized on a staggered grid as described in [2], with the constant null-space removed in a standard way. The model $m = m(\mathbf{x})$ is related to the resistivity ρ (in $\Omega \cdot m$) through $\rho = e^{-m}$.

5.1 Experiment 1

This experiment employs surface electrodes to image the injection of a saline tracer into the subsurface. Injection of tracers is often performed by hydrologists in an effort to obtain information about subsurface hydrologic properties. In this experiment, we simulate the tracer-saturated soil as a 10 $\Omega \cdot m$ material and the soil as a 200 $\Omega \cdot m$ material. This corresponds to $m_I = -2.30$ and $m_{II} = -5.30$. On the surface we deploy 96 electrodes; 38 of these electrodes function as potential electrodes only, while the remaining 58 serve as both potential and current electrodes. The experiment involves $s = 41$ independent current pairs. For each of these pairs, data are recorded at the remaining 94 electrodes. Data are recorded using dipoles, so for each current pair we record 93 independent data. This results in a total of 3813 data for the entire survey. Artificial data is computed on a 64^3 grid, polluted with 3% noise, and the reconstructions are done on a 16^3 grid. The rectangular tracer region is depicted in Figure 1.

In Figure 2 we show reconstructions using the default L_2-method of RESINVM3D, with $\beta = 10^{-3}$. The misfit of .04 is close to the ideal value of .03.

In Figure 3 we show reconstruction using the level set method, with 3 PCG iterations per inner iteration. The initial guess was obtained by visual matching of the solution obtained with the conventional method. If a different, more remote, configuration is chosen as the initial guess then more outer iterations are required, as expected. Observe that considerable detail has been added in the reconstruction.



(a) Artificial model, view 1.

(b) Artificial model, view 2.

(c) Artificial model, view 3.

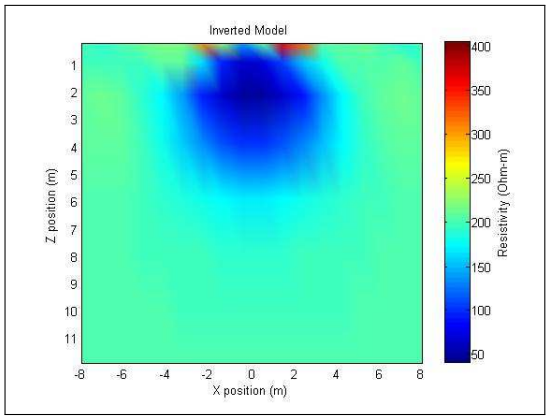
(d) Artificial model, view 4.

Figure 1: The artificial model for Experiment 1.

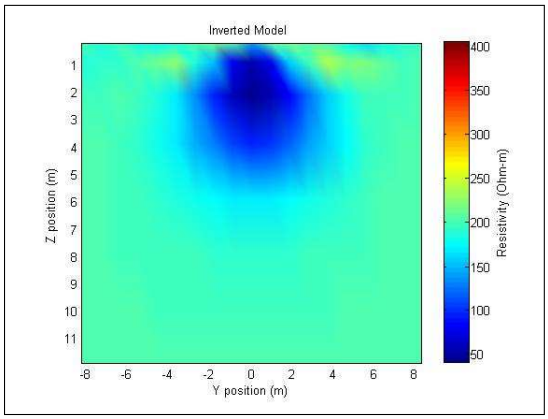
5.2 Experiment 2

In this experiment the setup of our current sources and data electrodes is the same as in Experiment 1, but the synthetic data now consists of two disjoint shapes.

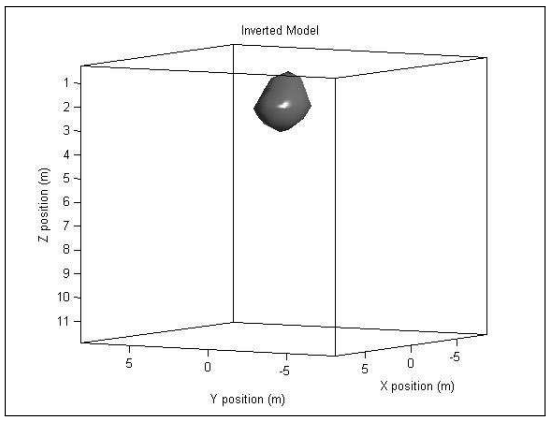
For this data the L2-method described in Section 4 is not able to resolve the distinct shapes, but the level set based method succeeds. In Figures 4 and 5 we show the synthetic model as well as reconstructions on 16^3 and 32^3 grids.



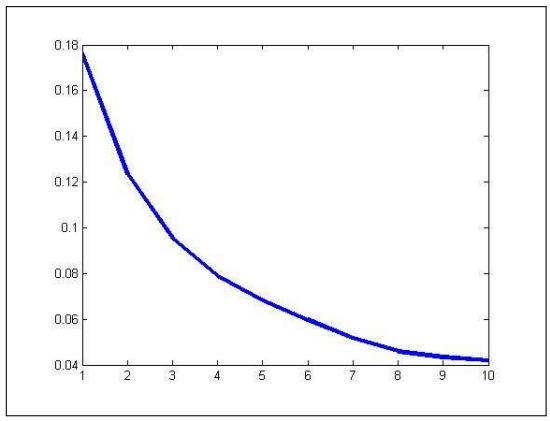
(a) Reconstructed model, view 1.



(b) Reconstructed model, view 2.



(c) Artificial model, surface reconstruction with harmonic averages.



(d) Misfits at each outer iteration.

Figure 2: Results with the L_2-method for Experiment 1, reconstructed on a 16^3 grid using $\beta = 10^{-3}$.

In Figures 6 we show the level set reconstructions on a 32^3 grid at various iterations using 3 PCG iterations, and in Figures 7 we see the result of using 10 PCG iterations. The reconstruction is clearly much better with 3 PCG iterations, indicating that there is not enough regularization with 10 PCG iterations. In Figure 8 we re-introduce a non-zero $\beta_0 = 0.01$. This provides the additional regularization needed and the result is good again. But of course the reconstruction is more than three times slower than with 3 PCG iterations per outer iteration. The results from an experiment with just 1 PCG iteration, equivalent to preconditioned steepest descent with line search, are shown in Figure 9. About three times as many outer iterations are required compared with using 3 PCG iterations. However, since the computational cost of the solution of (5a) is now reduced by about a factor of three, the total computational cost is roughly the same.

5.3 Experiment 3

Experiment 3 is adapted from the second example in [31] and employs subsurface electrodes. The acquisition technique involves first placing a small number of permanent current electrodes, then using an electrode mounted on a cone penetrometer to make potential measurements at various locations in the subsurface. In this synthetic example, a factory is suspected of leaking contaminants into the ground beneath its footprint. The presence of the factory makes it impossible to perform a surface-based geophysical survey over the region of interest. As we wish to image the volume beneath the factory, we need to surround the volume with electrodes. For this example we have eight source electrodes and 168 potential electrodes.

The conductivity model used for this experiment has a homogeneous 100 $\Omega \cdot m$ background resistivity, corresponding to a fresh water sandy aquifer. The contaminated soil is modeled as a 5 $\Omega \cdot m$ material, the shape of which is depicted in Figure 10. This corresponds to $m_I = -1.61$ and $m_{II} = -4.61$. The reconstruction was done on a 24^3 grid and the synthetic data was computed on a 48^3 grid. The 8 source electrodes yield 28 independent current pairs, and there are 168 potential electrode positions. With this acquisition technique we have a potential electrode located at infinity, thus we acquire pole-dipole data. The entire experiment yields 4704 synthetic data points. As with the previous experiments, the data were corrupted with 3% Gaussian noise.

The results recorded in Figure 10 show that our level set method performs very well. Some details of the true model are recovered, and the algorithm stalls smoothly at the right place. The reconstruction by the L_2-method is clearly inferior, although this does not show in the misfit level.

5.4 Experiment 4

In this experiment we employ electrodes and receivers inside the medium. We scale our domain to $[0\ 1]^3$. A positive electrode and a negative electrode are placed at opposing corners of the cube, giving 4 different source configurations. Data is taken inside a cube of sides 0.4 in the center on a uniform 7^3 grid of electrodes. Artificial data polluted with 1% noise is computed on a 128^3 grid and reconstructions are done on coarser grids.

In Figure 11 we show reconstructions. Not much seems to be gained from using the finer, 64^3 grid.

5.5 Experiment 5

In this experiment the source and receiver configuration is as in Experiment 4, but we use a different synthetic model resembling a buried alien artifact, depicted in Figure 12. We have computed data on a 64^3 grid, polluted it with 1% Gaussian noise, and performed reconstructions on 32^3 and 16^3 grids.

The level set reconstructions are again better than using just Tikhonov. The finer grid distinctly allows more articulation of the model.

6 Conclusions and further thoughts

Shape optimization of the type considered in this article, where the forward operator involves the inversion of an elliptic PDE, is notoriously difficult. These problems are highly ill-posed, and limitations of any relevant numerical method in certain situations are well documented. Moreover, often methods proposed in the literature for the same task appear rather different from one another, leaving much room for interpretation. In this article, abandoning the comfort of direct linear algebra methods in order to solve larger problems iteratively, we have found ourselves connecting between different approaches and making several fundamental adjustments.

The forward problem is solved many times in order to construct one sensitivity matrix. This suggests that an iterative solver for the linear system arising in the dynamic regularization iteration (5) is important, whereas for the forward a direct method would be particularly advantageous if we only have the memory for it. This, even though fast iterative methods such as multigrid for (2) are well developed [36]. For the forward problem an incomplete LU decomposition proves useful because it can be shared for many right hand sides.

Furthermore, the inverse linear system is no longer solved accurately, and this significantly alters the method itself, not just the means for solving its linear systems. Specifically, a few PCG iterations take over the regularizing task and efficiently yield a descent direction for the output least squares objective function. The Hessian of the regularization operators (4) now appears only in the preconditioning, and there is no effective difference between them, in contrast to [37]. The present method can also be derived from a Tikhonov-type formulation of a level set method as in [37], letting $\beta \to 0$ there.

In usual Tikhonov-type regularizations one incorporates a priori information into the process through the regularization operator. This potentially includes not only information about the expected smoothness but also a prior, or a reference function $m_{\text{ref}}(\mathbf{x})$. In contrast, here the smoothness is explicitly expressed through the level set function, and (barring statistical considerations) a prior may only be used as an initial guess for the nonlinear iteration.

Let us stress again that our usage of level set functions is strictly and directly related to iterating with a smoother function: whereas our unknown model m is only in $W_0(\Omega) = \mathcal{L}_2(\Omega)$, the function ψ used to describe it is in $W_1(\Omega)$, i.e. both it and its gradient are in $\mathcal{L}_2(\Omega)$. We do *not* evolve the level set in a way that involves the Hamilton-Jacobi equations [29, 9].

We have not included specific run times because they are significantly affected by memory availability and because we have made heavy use of a particular version of someone else's code [31]. In general, however, a complete run on a 16^3 grid requires

a few minutes on a laptop, a 32^3 grid requires a few hours and a 64^3 grid requires about 24 hours. We have indicated in several places how these times can be improved. The resolution is often insufficient on a 16^3 grid. Non-uniform grids therefore suggest themselves, and this is a subject for our future investigation.

References

[1] A. Alessandrini and S. Vessella. Lipschitz stability for the inverse conductivity problem. *Adv. Appl. Math.*, 35:207–241, 2005.

[2] U. Ascher and E. Haber. A multigrid method for distributed parameter estimation problems. *J. ETNA*, 18:1–18, 2003.

[3] U. Ascher and E. Haber. Computational methods for large distributed parameter estimation problems with possible discontinuities. In *Proc. Symp. Inverse Problems, Design and Optimization*, pages 201–208, 2004. M. Colaco, H. Orlande and G. Dulikravich (eds.).

[4] U. Ascher, E. Haber, and H. Huang. On effective methods for implicit piecewise smooth surface recovery. *SIAM J. Scient. Comput.*, 28:339–358, 2006.

[5] U. Ascher, H. Huang, and K. van den Doel. Artificial time integration. *BIT Numerical Mathematics*, 47:3–25, 2007.

[6] L. Borcea, J. G. Berryman, and G. C. Papanicolaou. High-contrast impedance tomography. *Inverse Problems*, 12:835–858, 1996.

[7] L. Borcea, G. Gray, and Y. Zhang. Variationally constrained numerical solution of electrical impedance tomography. *Inverse Problems*, 19:1159–1184, 2003.

[8] M. Burger. A level set method for inverse problems. *Inverse Problems*, 17:1327–1355, 2001.

[9] M. Burger. Levenberg-marquardt level set methods for inverse obstacle problems. *Inverse problems*, 20:259–282, 2004.

[10] T. Chan and X. Tai. Level set and total variation regularization for elliptic inverse problems with discontinuous coefficients. *J. Comp. Phys.*, 193:40–66, 2003.

[11] M. Cheney, D. Isaacson, and J.C. Newell. Electrical impedance tomography. *SIAM Review*, 41:85–101, 1999.

[12] E. Chung, T. Chan, and X. Tai. Electrical impedance tomography using level set representations and total variation regularization. *J. Comp. Phys.*, 205:357–372, 2005.

[13] A. J. Devaney. The limited-view problem in diffraction tomography. *Inverse Problems*, 5:510–523, 1989.

[14] O. Dorn and D. Lesselier. Level set methods for inverse scattering. *Inverse Problems*, 22:R67–R131, 2006. Topical Review.

[15] H.W. Engl, M. Hanke, and A. Neubauer. *Regularization of Inverse Problems*. Kluwer, 1996.

[16] F Fruhauf, O. Scherzer, and A. Leitao. Analysis of regularization methods for the solution of ill-posed problems involving unbounded operators and a relation to constraint optimization. *SIAM J. Numer. Anal.*, 43:767–786, 2005.

[17] E. Haber. A multilevel, level-set method for optimizing eigenvalues in shape design problems. *J. Comp. Phys.*, 198:518–534, 2004.

[18] E. Haber and U. Ascher. Preconditioned all-at-one methods for large, sparse parameter estimation problems. *Inverse Problems*, 17:1847–1864, 2001.

[19] E. Haber, U. Ascher, D. Aruliah, and D. Oldenburg. Fast simulation of 3D electromagnetic using potentials. *J. Comput. Phys.*, 163:150–171, 2000.

[20] E. Haber, U. Ascher, and D. Oldenburg. Inversion of 3D electromagnetic data in frequency and time domain using an inexact all-at-once approach. *Geophysics*, 69:1216–1228, 2004.

[21] M. Hanke. Regularizing properties of a truncated newton-cg algorithm for non-linear inverse problems. *Numer. Funct. Anal. Optim.*, 18:971–993, 1997.

[22] M. Hanke and P. C. Hansen. Regularization methods for large scale problems. *Survey on Mathematics for Industry*, 3:253–315, 1993.

[23] P. C. Hansen. *Rank Deficient and Ill-Posed Problems*. SIAM, Philadelphia, 1998.

[24] F. Hettlich and W. Rundell. Iterative methods for the reconstruction of an inverse potential problem. *Inverse Problems*, 12:251–266, 1996.

[25] V. Isakov. *Inverse Problems for Partial Differential Equations*. Springer, 2006.

[26] G. Newman and D. Alumbaugh. Three-dimensional massively parallel electromagnetic inversion–I. theory. *Geophysical journal international*, 128:345–354, 1997.

[27] G. Newman and D. Alumbaugh. Three-dimensional massively parallel electromagnetic inversion–II, analysis of a crosswell electromagnetic experiment. *Geophysical journal international*, 128:355–367, 1997.

[28] J. Nocedal and S. Wright. *Numerical Optimization*. New York: Springer, 1999.

[29] S. Osher and J. Sethian. Fronts propagating with curvature dependent speed: algorithms based on Hamilton-Jacobi formulations. *J. Comp. Phys.*, 79:12–49, 1988.

[30] R. L. Parker. *Geophysical Inverse Theory*. Princeton University Press, Princeton NJ, 1994.

[31] A. Pidlisecky, E. Haber, and R. Knight. RESINVM3D: A MATLAB 3D Resistivity Inversion Package. *Geophysics*, 72(2):H1–H10, 2007.

[32] F. Santosa. A level-set approach for inverse problems involving obstacles. *ESAIM Controle Optim. Calc. Var.*, 1:17–33, 1996.

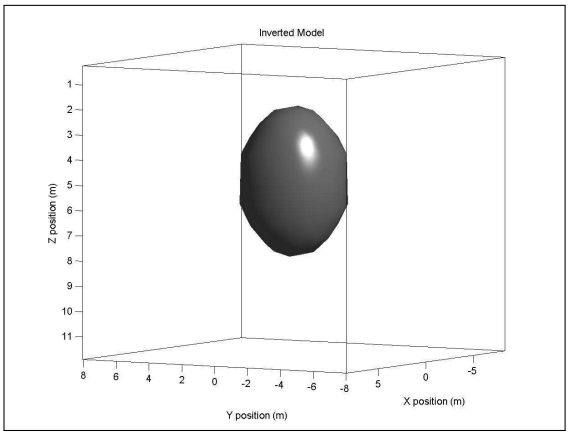
[33] N.C. Smith and K. Vozoff. Two dimensional DC resistivity inversion for dipole dipole data. *IEEE Trans. on geoscience and remote sensing*, GE 22:21–28, 1984.

[34] T. Steihaug. The conjugate gradient method and trust regions in large scale optimization. *SIAM J. Num. Anal.*, 20:626–637, 1983.

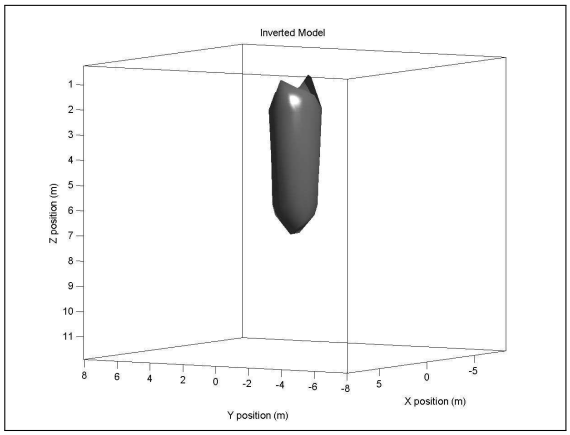
[35] A.N. Tikhonov and V.Ya. Arsenin. *Methods for Solving Ill-posed Problems*. John Wiley and Sons, Inc., 1977.

[36] U. Trottenberg, C. Oosterlee, and A. Schuller. *Multigrid*. Academic Press, 2001.

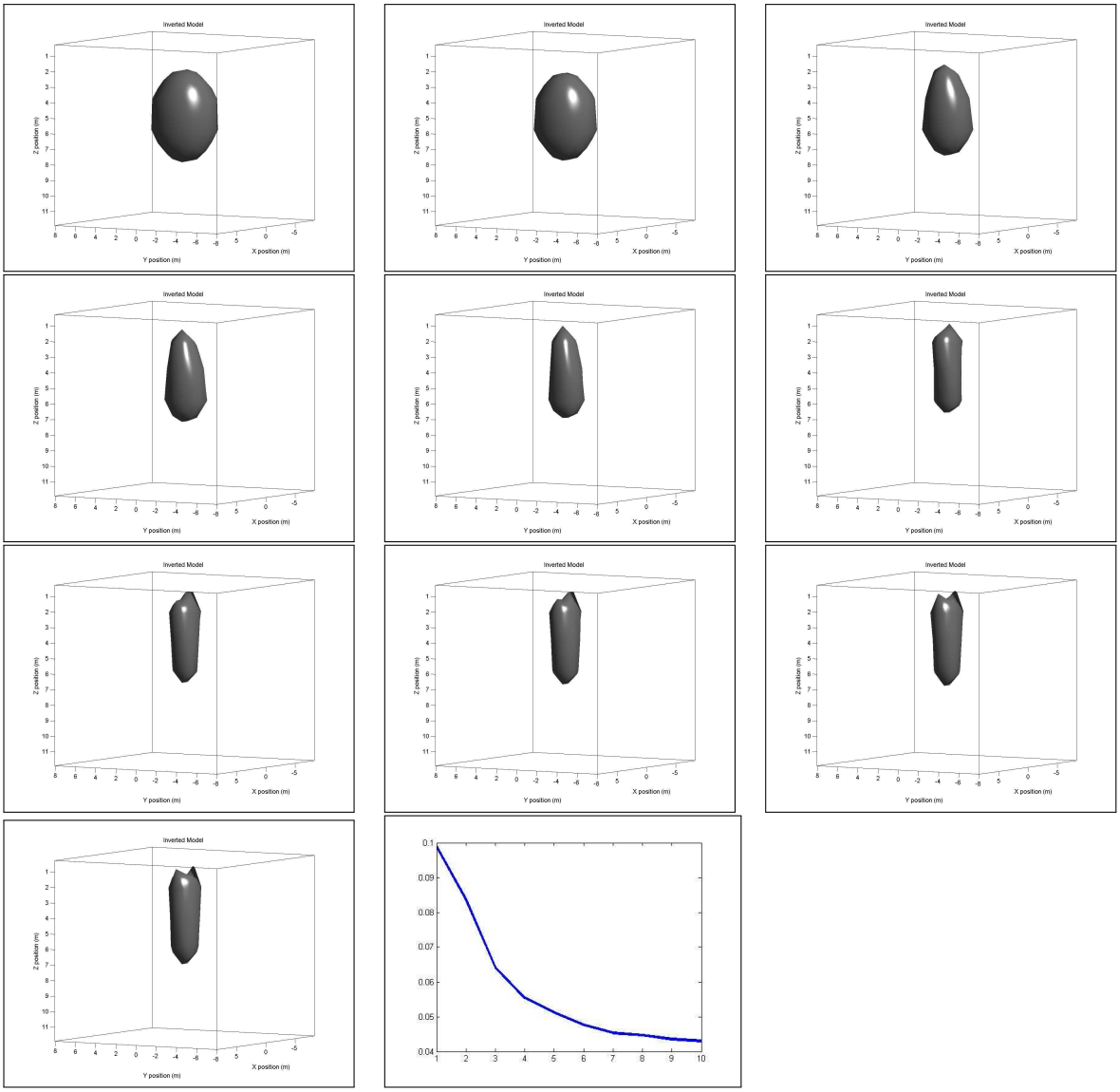
[37] K. van den Doel and U. Ascher. On level set regularization for highly ill-posed distributed parameter estimation problems. *J. Comp. Phys.*, 216:707–723, 2006.



(a) Initial guess.

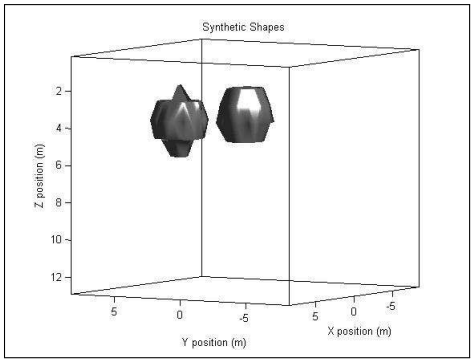


(b) Result after 10 iterations.

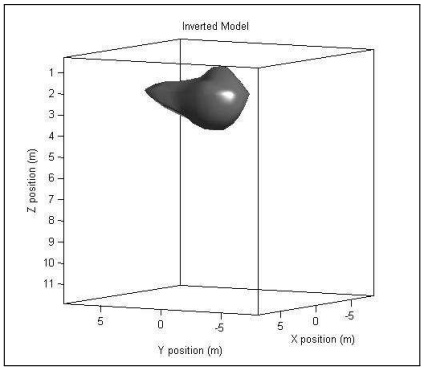


(c) Misfit at each iteration.

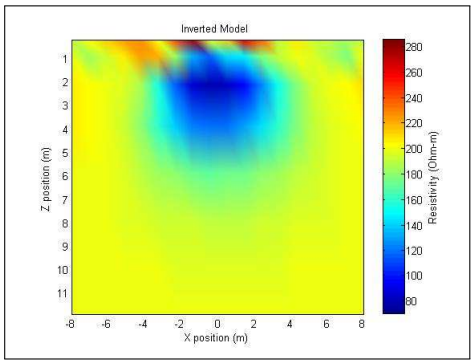
Figure 3: Experiment 1: Reconstructions on a 16^3 grid using the level set method. Depicted are the initial state, the result after 10 iterations, and the individual iterations.



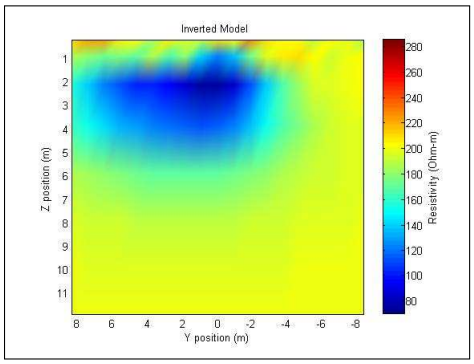
(a) Synthetic model.



(b) L_2-method, 16^3 grid.

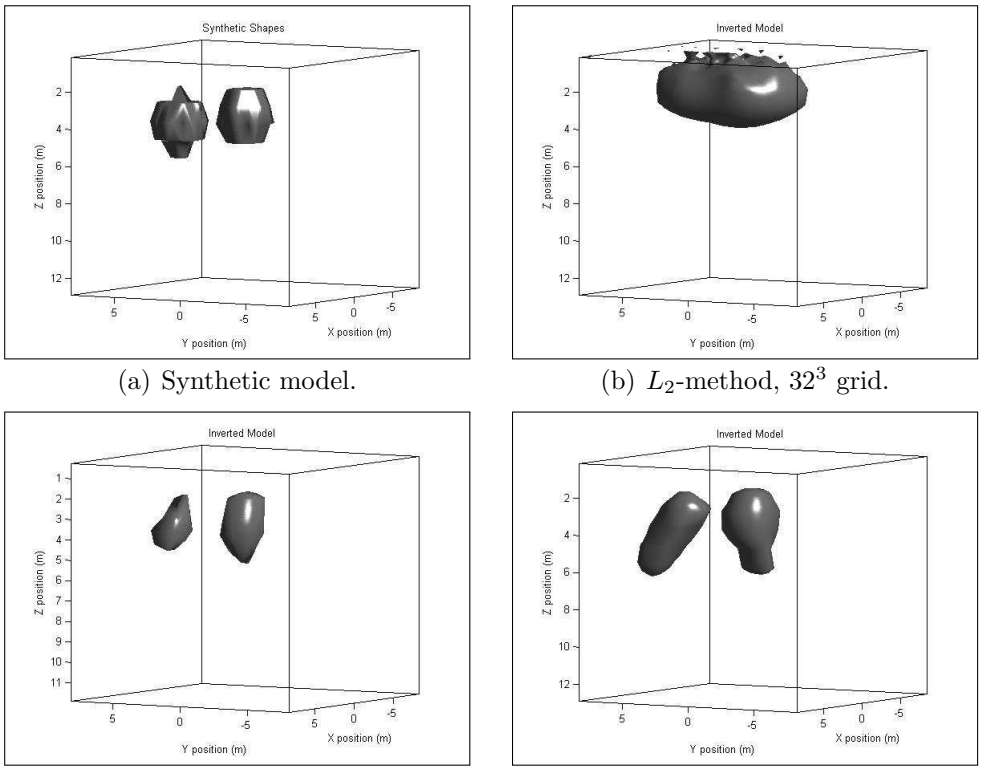


(c) X-view.



(d) Y-view.

Figure 4: Results for Experiment 2. The misfit is .04 after 10 iterations. For the L_2-method reconstructions, $\beta = 10^{-3}$.



(a) Synthetic model. (b) L_2-method, 32^3 grid.

(c) Level set, 16^3 grid. (d) Level set, 32^3 grid.

Figure 5: Results for Experiment 2. The misfit is .04 after 10 iterations. For the L_2-method reconstructions, $\beta = 10^{-3}$. Contrast the lack of separation for the L_2-method with the relative success of the level set approach.

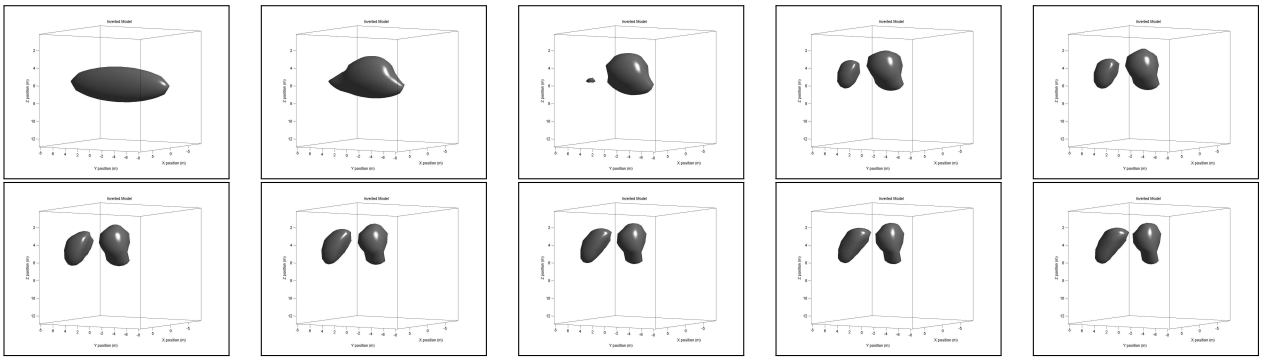


Figure 6: Experiment 2: Reconstructions using 3 PCG iterations per outer iteration. Shown are the initial guess and 9 successive iterations.

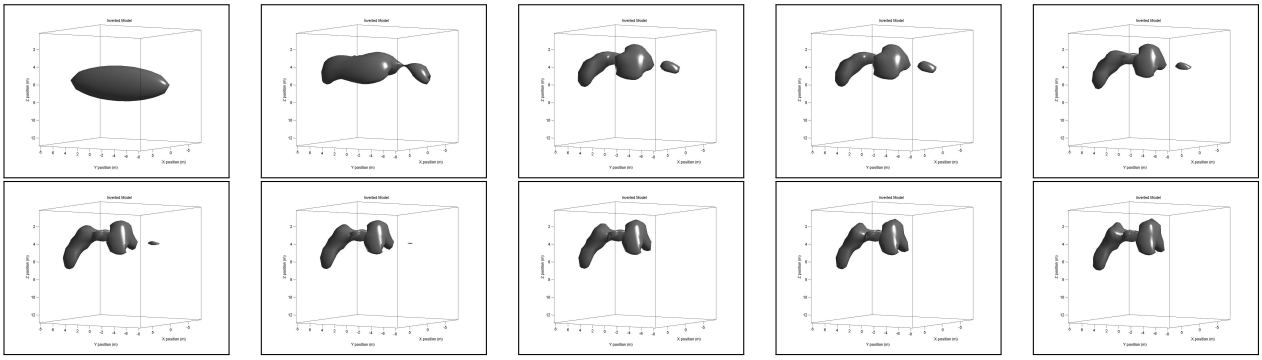


Figure 7: Experiment 2: Reconstructions using 10 PCG iterations per outer iteration. Shown are the initial guess and 9 successive iterations.

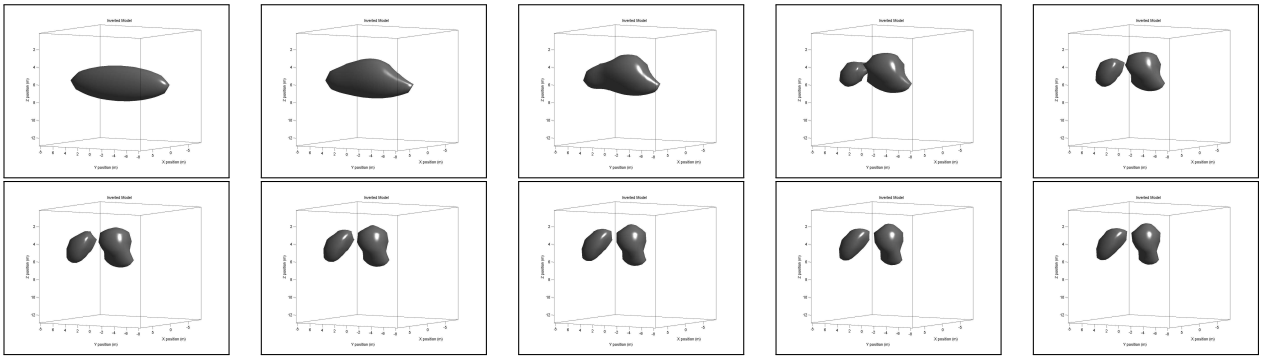


Figure 8: Experiment 2: Reconstructions using 10 PCG iterations with a non-zero $\beta_0 = 10^{-2}$. Shown are the initial guess and 9 successive iterations.

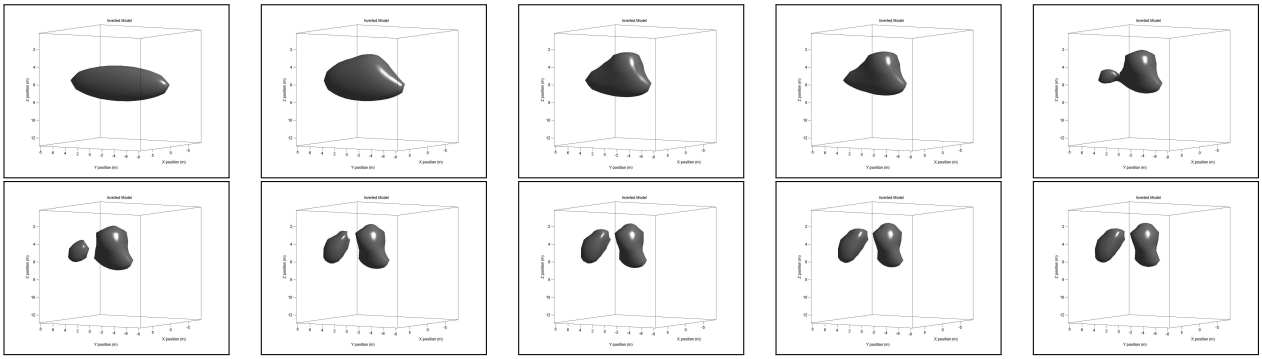
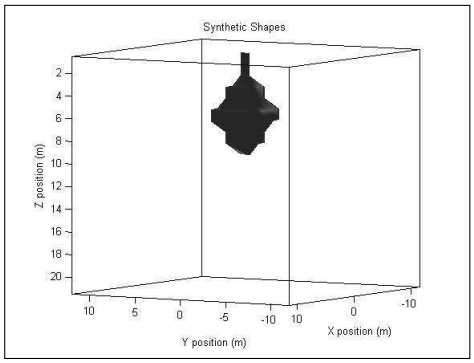
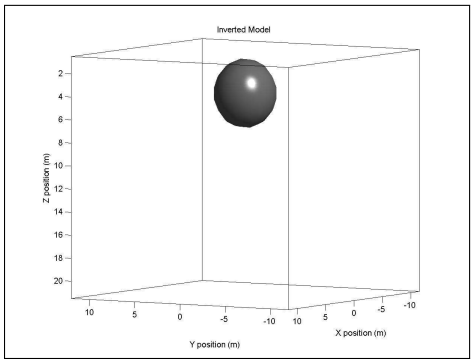


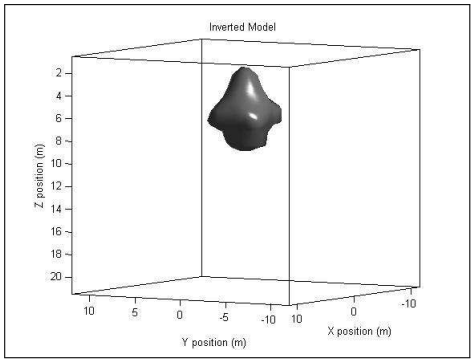
Figure 9: Experiment 2: Reconstructions using 1 PCG iteration. Shown are the initial guess and iterations $3, 6, \ldots, 24, 27$.



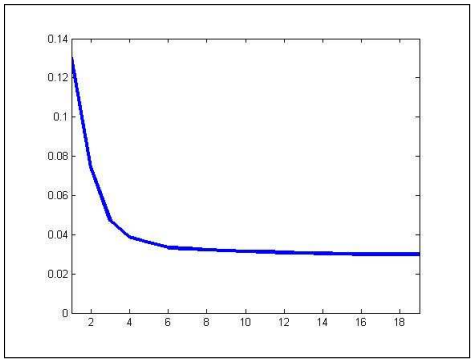
(a) Synthetic model.



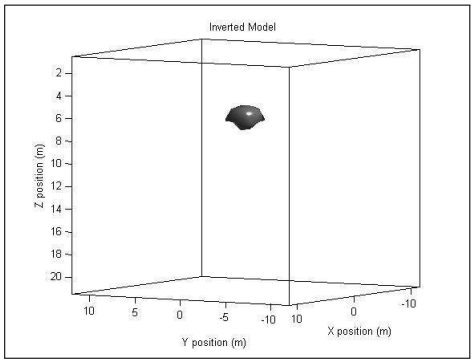
(b) Initial guess.



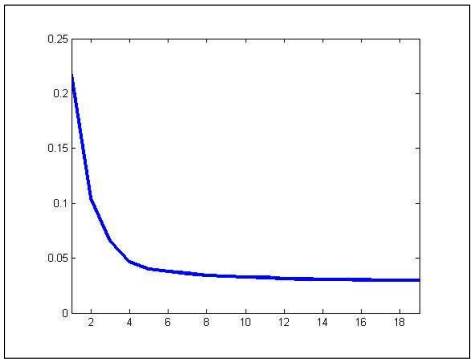
(c) Level set reconstruction after 19 iterations.



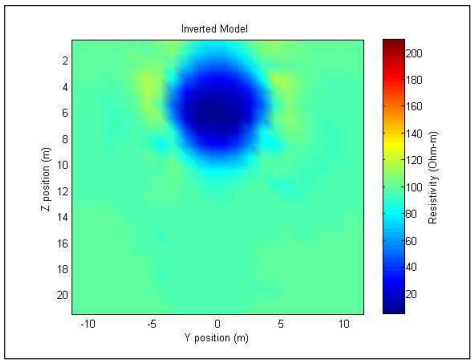
(d) Level set misfit.



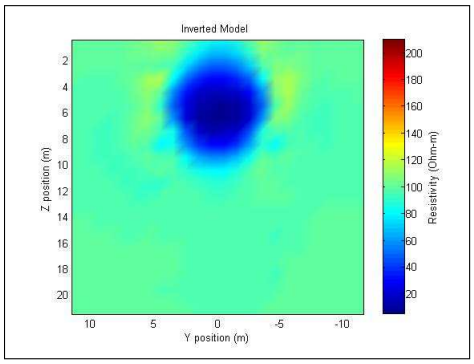
(e) L_2-method reconstruction after 19 iterations.



(f) L_2-method misfit

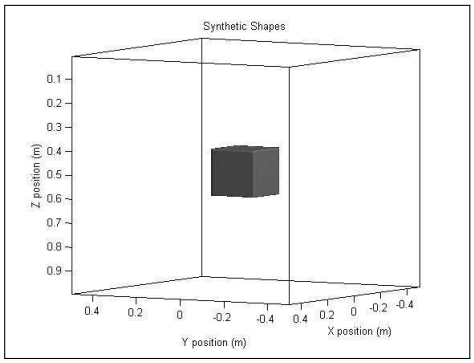


(g) L_2 reconstruction, X-view.

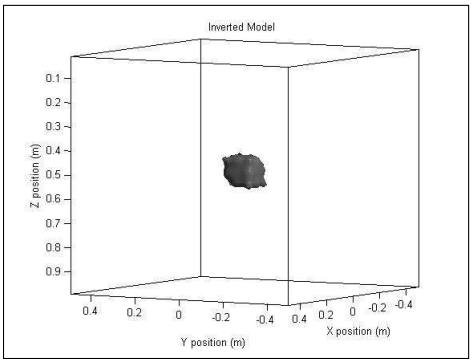


(h) L_2 reconstruction, Y-view.

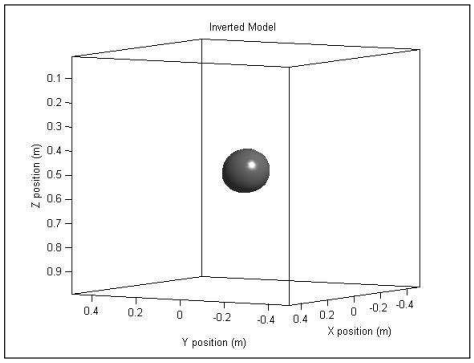
Figure 10: Experiment 3: The calculated misfit for all reconstructions was .03, which is ideal. For the L_2-method reconstructions, $\beta = 10^{-3}$.



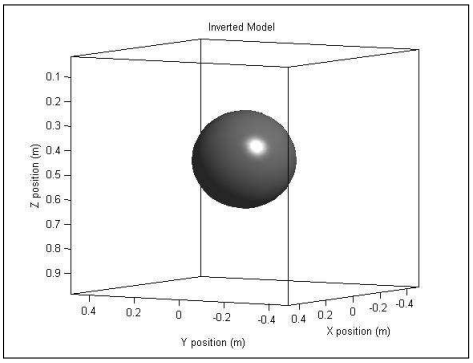
(a) True model.



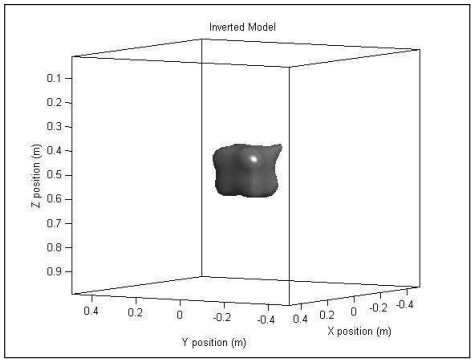
(b) L_2 reconstruction, $\beta = 10^{-9}$, misfit $= 10^{-2}$ (ideal), 34 iterations on 64^3 grid.



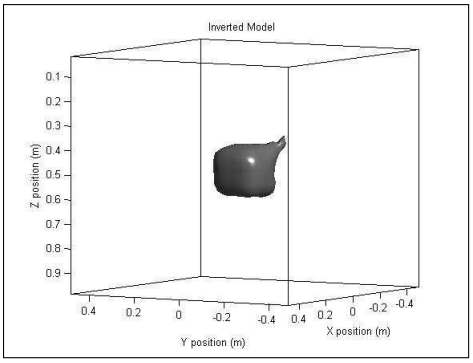
(c) L_2 reconstruction, $\beta = 10^{-7}$, misfit $= .02$, 34 iterations on 64^3 grid.



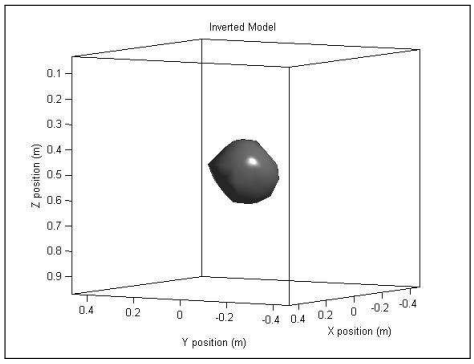
(d) Initial guess for level set method.



(e) Level set, misfit $= .01$, 33 iterations on 64^3 grid.

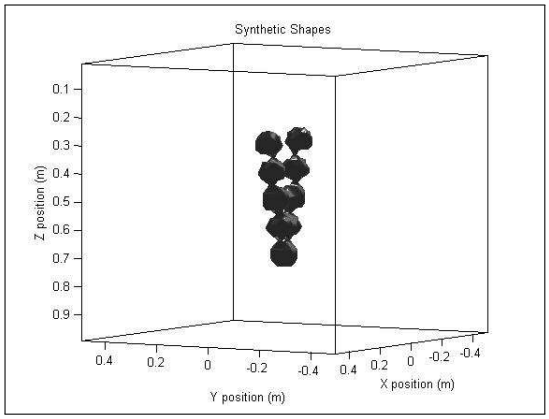


(f) Level set, misfit $= .018$, 20 iterations on 32^3 grid.

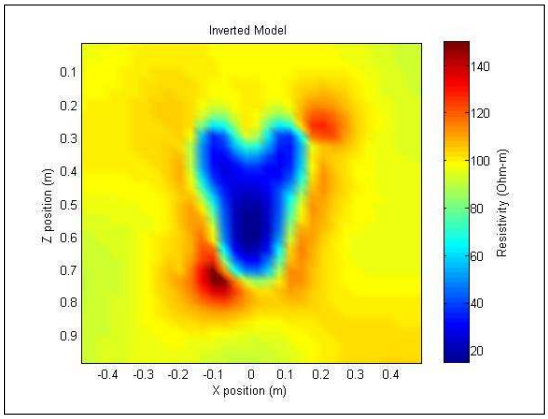


(g) Level set, misfit $= .02$, 33 iterations on 16^3 grid.

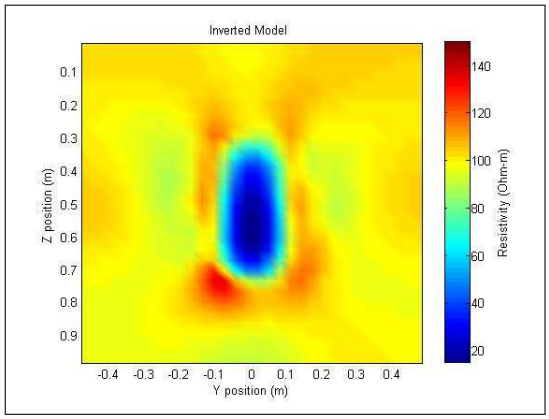
Figure 11: Experiment 4: Cube reconstructions on 64^3 and 32^3 grids.



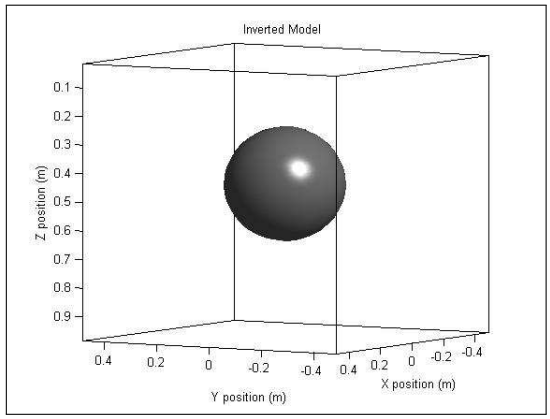
(a) True model.



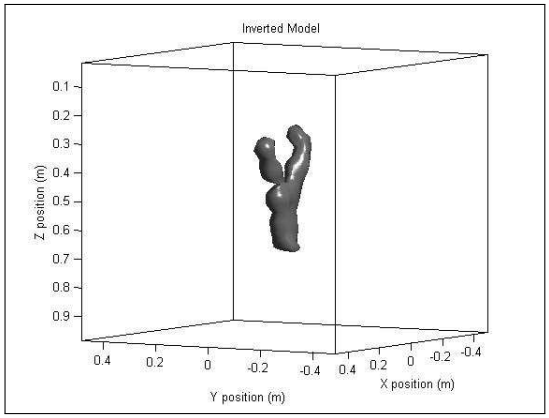
(b) L_2 reconstruction, $\beta = 10^{-9}$, misfit = .01 (ideal), 40 iterations on 32^3 grid.



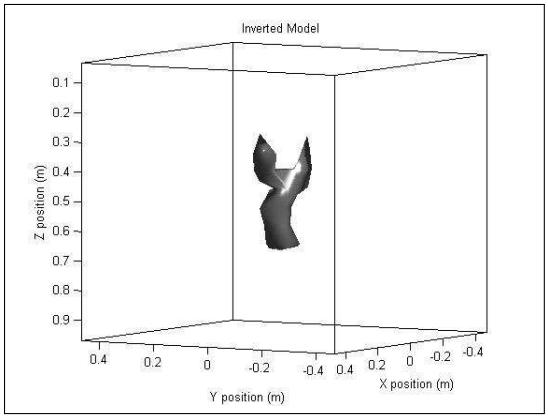
(c) L_2 reconstruction, Y-view.



(d) Initial guess for level set method.



(e) Level set, misfit = .01, 40 iterations on 32^3 grid.



(f) Level set, misfit = .02, 40 iterations on 16^3 grid.

Figure 12: Experiment 5: Reconstructions on 32^3 and 16^3 grids.