Level Set And Total Variation Regularization For Elliptic Inverse Problems With Discontinuous Coefficients [*]

Tony F. Chan [†]   Xue-Cheng Tai[‡]

March 14, 2003

Abstract

We propose a level set approach for elliptic inverse problems with piecewise constant coefficients. The geometry of the discontinuity of the coefficient is represented implicitly by level set functions. The inverse problem is solved using a variational augmented Lagrangian formulation with total variation regularization of the coefficient. The corresponding Euler Lagrange equation gives the evolution equation for the level set functions and the constant values of the coefficients. We use a multiple level set representation which allows the coefficient to have multiple constant regions. Knowledge of the exact number of regions is not required, only an upper bound is needed. Numerical experiments show that the method can recover coefficients with rather complicated geometries of discontinuities under moderate amount of noise in the observation data. The method is also robust with respect to the initial guess for the geometry of the coefficient discontinuities.

1 Introduction

Consider the partial differential equation:

$$\begin{cases} -\nabla \cdot (q(x)\nabla u) = f & \text{in } \Omega \subset R^2, \\ u = 0 & \text{on } \partial\Omega. \end{cases} \tag{1}$$

[*] We acknowledge support from the Norwegian Research Council (project numbers 135420/431 and 135302/320), from the NSF under contracts NSF ACI-0072112 and NSF INT-0072863, from the ONR under grant N00014-96-1-10277, and from the NIH under grant P20MH65166.

[†] Department of Mathematics, University of California, Los Angeles, 405 Hilgard Avenue, Los Angeles, CA 90095-1555. Email: chan@math.ucla.edu, URL: http://www.math.ucla.edu/˜chan.

[‡] Department of Mathematics, University of Bergen, Johannes Brunsgate 12, N-5007 Bergen, Norway. Email: Tai@mi.uib.no. URL: http://www.mi.uib.no/˜tai.

We want to use observations of the solution u to recover the coefficient $q(x)$. Output-least-squares combined with augmented Lagrangian method will be used. We shall specifically treat the case that $q(x)$ has discontinuities and is piecewise constant. This problem is a model problem for many real industrial applications, for example, reservoir simulations, see [16, 19, 18, 17, 29]; underground water investigations [8], medical imaging [11, 14], and many other applications, see [13, 31, 32, 23]. Even as a purely academic problem, this seemingly simple problem is rather difficult to solve by numerical schemes. In the presence of noise in the observation data, it has been shown theoretically, c.f. [20, 21, 40, 41, 42], that the approximation error increases as the mesh size decreases. Up to now, it seems that there are not many available algorithms that can solve this inverse problem with relative large noise on a sufficient fine mesh.

The desire to recover accurately the geometry of the coefficient discontinuities have motivated a number of approaches in the literature [5, 6, 24, 27, 10, 9, 1]. One approach is to use a regularization of the coefficient which respects the jumps and the geometry of the discontinuities. For example, in our earlier work [5, 6], the Total Variation norm regularization technique is combined with the augmented Lagrangain technique of [24, 27] for this purpose. Other works along this line are [10, 9], etc. An alternative approach is to model the geometry of the discontinuities implicitly in the representation of the coefficient. Specifically, several approaches using level set ideas have been recently proposed for this purpose; see [38, 35, 25, 2, 3, 15, 30] for some poineering work in this direction. The level set method was proposed in Osher and Sethian [33] for tracing interfaces between different phases of fluid flows. Later, it has been used to identify the location of discontinuities for digital image functions [7, 44, 34]. Its application for finding locations of discontinuities for functions for inverse problems and optimal shape design problems is just in its beginning stage. In Osher and Santosa [38, 35], the level set idea was used for some inverse problems associated with eigenvalue distributions. In Ito-Kunisch-Li [25], level set ideas are used for elliptic inverse problems similar to the ones we are considering in this paper. Our approach extends the model in [25] in several directions and also incorporates the level set method in a different way.

We highlight the essential contributions of our paper in the following:

- We formulate the level set method in a variational setting, using a variational augmented Lagrangian formulation. The corresponding Euler Lagrange equation gives the evolution equations for the level set functions and the constant values. One of the advantages of the variational approach is that we can guarantee that the value of the minimization functional is always reducing. The algorithm is rather stable and we could reconstruct rather complicated geometry. In Ito-Kunisch-Li [25] and Burger [25], the shape derivative is used to get the steepest direction with respect to the entire boundary. This derivative is defined on the curve and it is extended to a small neighborhood around the curve. The extended function is then used as the gradient to update the level set function. The approach of [25, 3] cannot guarantee that the minimization functional value is decreasing all the time. The reconstructed curve may converge to the true solution and goes away from the true solution if the iteration continues [3]. On the other hand, it seems that the cost for each iteration for the schemes of [25, 3] is cheaper than the scheme proposed in this work.

- We do not require that the constant values of the coefficient are known *a priori*, as in [25, 3]. These values are recovered as part of our formulation.

- We regularize the recovered coefficient by the total variation norm, which indirectly controls both the jumps in the coefficient and the length of the level sets. We note that in the standard level set methodology, only the length part is regularized, which is insufficient for our purpose because we need to control the jumps as well, due to the ill-posedness of the underlying elliptic inverse problem. Another advantage of controlling the TV-norm instead of just the length of the level sets is that triple junctions are allowed to have arbitrary angles; see the recent work by Vese and Osher [43].

- Through the use of a novel variational multiple level set function approach borrowed from image processing [44], we can recover coefficients with multiple constant values associated with the discontinuous regions. If the identified coefficient have n constant values, then we just need $\log_2(n)$ level set functions. Moreover, the exact number of constant values need not be known *a priori* — an upper bound suffices. If we use more level set functions than are actually needed, the redundant regions will disappear or merge with the other regions during the iterative process.

We note that some preliminary experimental results with the level set method have been reported in our related work [4]. In that work, only the length of the level sets were used in the regularization and a different approach was used to solve for the level sets and the constant values. However, as we pointed out earlier, the need to control the jumps in the coefficient leads naturally to the consideration of using the TV-norm regularization, which we do here. More generally, the variational level set approach we are proposing here is rather general and can be used with any regularization norm. As will be demonstrated later, the use of level set representations of piecewise constant functions can be easily incorporated into standard variational formulations for general optimal shape design problems, in addition to the class of elliptic inverse problems considered in this paper. Finally, even though the presentation of our approach is presented here only for the piecewise constant coefficient case, the formulation can be easily generalized to recover piecewise smooth coefficients, using the formulation presented in [44].

The paper is organized in the following way: In §2, the identification problem is formulated as several minimization problems using the output-least-squares augmented Lagrangian approach as in our previous work [5, 6]. In §3, we give a brief introduction to level set methods and how to compute the total variation of piecewise constant functions represented by level sets. In §4, we present an algorithm that combines the variational level set approach with Uzawa's algorithm for the output-least-squares functionals. In §5, we discuss some numerical implementation issues. Numerical experiments are shown in §6. In all the numerical experiments, the examples are constructed so that the inverse problem is identifiable, see [37, 12, 28] for some conditions to guarantee the identifibility of the inverse problems. In one of numerical examples, we show what may happen if the inverse problem is not identifiable. In the appendix, calculations about the gradient of the minimization functionals which is needed for the variational level set approach are given.

2 Augmented Lagrangian Formulations For The Inverse Problem

Our approach to the inverse problem is based on an augmented Lagrangian formulation used in our earlier work [5, 6] — only the incorporation of the level representation is new. For this reason, we now give a review of the methodology used there. In the next section, we shall combine them with the level set method to solve the identification problems.

In order to recover the coefficient $q(x)$, three different kind of observations are used:

(1) We have an observation $u_d \in L^2(\Omega)$ for the solution u.

(2) We have observations $u_d \in L^2(\Omega),\ \vec{u}_g \in (L^2(\Omega))^2$ for the solution u and its gradient, respectively.

(3) We have observations $u_d \in L^2(\Omega),\ \vec{u}_v \in (L^2(\Omega))^2$ for the solution u and the velocity $q\nabla u$, respectively.

Let K be the set of admissible coefficients:

$$K = \{q| \quad q \in L^\infty(\Omega) \cap TV(\Omega), \quad 0 < \underline{q}(x) \le q(x) \le \bar{q}(x) < \infty\},$$

with $\underline{q}(x)$ and $\bar{q}(x)$ known *a priori*. For a given q and u, we shall define the equation error $e(q,u)$ as the solution of

$$(\nabla e, \nabla v) = (q(x)\nabla u, \nabla v) - (f, v), \quad \forall v \in H_0^1(\Omega). \tag{2}$$

Thus $e(q,u) = 0$ indicates that

$$(q(x)\nabla u, \nabla v) - (f, v) = 0, \quad \forall v \in H_0^1(\Omega),$$

which means that q and u satisfy (1) in the weak sense. Here and also later, the inner product $(\cdot,\cdot)$ is used to denote the $L^2(\Omega)$ inner product. Finally, we use $u(q)$ to denote the solution to $e(q,u) = 0$ for a given q.

Corresponding to each of the three kinds of observations, we solve the following minimization problems:

$$\textbf{(P1)} \qquad \min_{e(q,u)=0,\ q\in K} \frac{1}{2}\|u - u_d\|^2_{L^2(\Omega)} + \beta R(q)\ ,$$

$$(\mathbf{P2}) \qquad \min_{e(q,u)=0,\ q\in K} \frac{1}{2}\|u-u_d\|^2_{L^2(\Omega)} + \frac{1}{2}\gamma\|\nabla u(q) - \vec{u}_g\|^2_{\mathbf{L}^2(\Omega)} + \beta R(q)\ ,$$

$$(\mathbf{P3}) \qquad \min_{e(q,u)=0,\ q\in K} \frac{1}{2}\|u-u_d\|^2_{L^2(\Omega)} + \frac{1}{2}\gamma\|q\nabla u(q) - \vec{u}_v\|^2_{\mathbf{L}^2(\Omega)} + \beta R(q)\ ,$$

where $R(q)$ is a regularization functional used to control the regularity of $q(x)$.

If the coefficient is continuous, $R(q) = \|q\|^2_{H^2(\Omega)}$ or $\|q\|^2_{H^1(\Omega)}$ are commonly used as the regularization term, see [26, 24, 27]. In [27], existence, uniqueness and convergence have been proved for such kinds of regularization. However, if the coefficient has large jumps, the use of H^2 or H^1–regularization is not appropriate due to the discontinuities of the coefficient. In this work, we shall take as regularization functional the following:

$$R(q) = \int_\Omega |\nabla q| dx. \tag{3}$$

In the above, $R(q)$ is in fact the total variation of q; see Ziemer [45] and Giusti [22] for definitions. When q is not differentiable, $|\nabla q|$ is understood as a measure. More precisely,

$$\int_\Omega |\nabla q| dx = \sup\left\{ \int_\Omega q \ \mathrm{div} v dx : v = (v_1, v_2) \in C_0^\infty(\Omega), \right.$$
$$\left. v_1^2(x) + v_2(x) \leq 1 \text{ for } x \in \Omega \right\}. \tag{4}$$

See p. 221 of [45] for some more details.

The augmented Lagrangian method is used to enforce the equation constraint

$$e(q,u) = 0.$$

In the case that only L^2-observations are available, we define the augmented Lagrangian functional for any $r > 0$ as:

$$L_r(q,u,\lambda) = \frac{1}{2}\|u-u_d\|^2_{L^2(\Omega)} + \beta R(q) + \frac{r}{2}\|\nabla e(q,u)\|^2_{L^2(\Omega)} + (\nabla\lambda, \nabla e(q,u))_{L^2(\Omega)}.$$

In the above, we are trying to enforce the equation constraint $-\nabla \cdot (q\nabla u) = f$ in the H^{-1}-norm. For problem $(\mathbf{P2})$, the augmented Lagrangian functional is defined as:

$$\begin{aligned} L_r(q,u,\lambda) &= \frac{1}{2}\|u-u_d\|^2_{L^2(\Omega)} + \frac{1}{2}\gamma\|\nabla u - \vec{u}_g\|^2_{L^2(\Omega)} + \beta R(q) \\ &+ \frac{r}{2}\|\nabla e\|^2_{L^2(\Omega)} + (\nabla\lambda, \nabla e)_{L^2(\Omega)}, \\ &\text{for } q \in K, \quad u \in H_0^1(\Omega),\ \lambda \in H_0^1(\Omega). \end{aligned}$$

For problem (**P3**), the augmented Lagrangian functional is then:

$$\begin{aligned} L_r(q,u,\lambda) &= \frac{1}{2}\|u-u_d\|^2_{L^2(\Omega)} + \frac{1}{2}\gamma\|q\nabla u - \vec{u}_v\|^2_{L^2(\Omega)} + \beta R(q) \\ &+ \frac{r}{2}\|\nabla e\|^2_{L^2(\Omega)} + (\nabla\lambda, \nabla e)_{L^2(\Omega)}, \\ &\qquad \text{for } q \in K, u \in H^1_0(\Omega), \lambda \in H^1_0(\Omega). \end{aligned}$$

To find a saddle point for the augmented Lagrangian functional for the three different cases, the following general algorithm is used:

Algorithm 1 *Choose* $u_0 \in H^1_0(\Omega), \lambda_0 \in H^1_0(\Omega)$ *and* $r > 0$.

- *Find* $q^{k+1} \in K$ *such that:*

$$q^{k+1} = \arg\min_{q\in K} L_r(q, u^k, \lambda^k). \tag{5}$$

- *Find* $u^{k+1} \in H^1_0(\Omega)$ *such that:*

$$u^{k+1} = \arg \min_{u\in H^1_0(\Omega)} L_r(q^{k+1}, u, \lambda^k). \tag{6}$$

- *Update the multiplier as:*

$$\lambda^{k+1} = \lambda_k + re(q^{k+1}, u^{k+1}).$$

In the above, we have used the conventional notation $\arg\min F(v)$ to denote the minimizer of $\min F(v)$. The convergence analysis for this kind of Uzawa algorithms for the inverse problem we consider here can be found in [27, 10].

The observation data u_d and $\vec{u}_g$ often contain noise. Some denoising algorithms are proposed in [6]. Numerical evidence shows that these denoising techniques are rather important to get stable numerical performances.

3 Variational Problems Involving Level Set Represented Functions

Here, we state some of the details of the level set idea, following Osher and Sethian [33], Osher and Fedkiw [34] and Vese and Chan[44]. Let Γ be a closed curve in Ω. Associated with Γ, we define a ϕ as a signed distance function by:

$$\phi(x) = \begin{cases} \text{distance}(x,\Gamma), & x \in \text{interior of } \Gamma \\ -\text{distance}(x,\Gamma), & x \in \text{exterior of } \Gamma. \end{cases}$$

It is clear that Γ is the zero level set of the function ϕ. In case that Γ is not closed, but divide the domain into two parts, then the function can be defined to be positive on one side of the curve and negative on the other side of the curve. The function ϕ is called a level set function for Γ. It is clear that ϕ satisfies the partial differential equation:

$$|\nabla\phi| = 1, \quad \text{in } \Omega. \tag{7}$$

However, ϕ is not the only function that satisfies equation (7) in the distribution sense. In order to define a unique solution for the equation, we need to introduce the concept of viscosity solution. The existence and uniqueness of viscosity solutions for linear and nonlinear partial differential equations is an active research field with rich literature results. There are different ways to introduce viscosity solutions. One way to introduce the viscosity function is to add an extra time variables t. Let $\tilde{\phi}$ be any function such that Γ is the zero level set curve of $\tilde{\phi}$ and $\tilde{\phi}$ is positive inside Γ and negative outside Γ. Then the distance function ϕ is the steady state of the following time dependent equation (c.f. [34, 36]):

$$\frac{\partial d}{\partial t} + \mathrm{sign}(d)(|\nabla d| - 1) = 0 \qquad d(x,0) = d_0 = \tilde{\phi}, \tag{8}$$

i.e. $d(x,t;\tilde{\phi}) \to \phi(x)$ as $t \to \infty$. Moreover the steady state is unique. Later, we shall see that we only need the value of d in a band of width ϵ around Γ. Correspondingly, we only need to solve equation (8) for $t \le O(\epsilon)$.

Equation (8) is a hyperbolic first order equation which can be solved by finite element or finite difference method combined with the characteristic method for hyperbolic equations. In order to introduce a finite difference approximation, let us assume that (x_i, y_j) are the nodes for the finite difference method and $\Delta x, \Delta y, \tau$ are the mesh sizes for the x, y and t variables respectively. Different schemes have been proposed in the literature for solving (8). It is desirable that the scheme has the following properties:

- It can reach a steady state as quickly as possible.
- The steady state solution should have the same sign as the initial solution $\tilde{\phi}$. This guarantees that the zero level set curve is not changed.

We use the algorithm of Peng et al [36], which has good performances with respect to these properties. In [36, p.427], the sign function $\mathrm{sign}(d)$ is replaced by:

$$s(d) = \frac{d}{\sqrt{d^2 + |\nabla d|^2 \Delta x^2}}. \tag{9}$$

This will add some viscosity to the solution and helps to drive the solution to a steady state. Among different upwind schemes used to solve (8), the following Godunov scheme is rather stable and accurate to first order:

$$\begin{aligned} d_{ij}^{k+1} &= d_{ij}^{k} - \frac{\tau}{\Delta x} s_{ij}^{+} \left(\sqrt{\max[(a^+)^2, (b^-)^2] + \min[(c^+)^2, (d^-)^2]} - 1 \right) \\ &\quad - \frac{\tau}{\Delta x} s_{ij}^{-} \left(\sqrt{\max[(a^-)^2, (b^+)^2] + \min[(c^-)^2, (d^+)^2]} - 1 \right). \end{aligned} \tag{10}$$

In the above, s_{ij} is the approximation to $s(d_{ij})$ with (9) and a, b, c, d are defined as:

$$a = D_x^- d_{ij}^k, \qquad b = D_x^+ d_{ij}^k,$$
$$c = D_y^- d_{ij}^k, \qquad d = D_y^+ d_{ij}^k.$$

The forward difference operators D_x^+, D_y^+ and the backward difference operators D_x^-, D_y^- are defined in the standard way. On the boundary, the forward or backward difference operators may not be defined. If so, their values are set to be zero. For a given number μ, μ^+ and μ^- denotes the actions $\mu^+ = \max(0, \mu)$ $\mu^- = \min(0, \mu)$. The scheme (10) is stable under the condition

$$\frac{\tau}{\Delta x} |s_{ij}| \leq \frac{1}{2}.$$

Once the level set function is defined, we can use it to represent general piecewise constant functions as follows. For example, assuming that $q(x)$ equals q_1 inside Γ and equals q_2 outside Γ, it is easy to see that q can be represented as:

$$q = q_1 H(\phi) + q_2 \left(1 - H(\phi)\right), \tag{11}$$

where the Heaviside function $H(\phi)$ is defined by:

$$H(\phi) = \begin{cases} 1, & \phi > 0 \\ 0, & \phi \leq 0. \end{cases}$$

In order to identify the coefficient q, we just need to identify the level set function ϕ and the piecewise constant values q_i's.

If the function $q(x)$ has many pieces, then we need to use multiple level set functions. We shall follow the ideas of Vese and Chan [44]. Assume that we have two closed curves Γ_1 and Γ_2, and we associate the two level set functions $\phi_j, j = 1, 2$ with these curves. Then the domain Ω is divided into four parts:

$$\begin{aligned} \Omega_1 &= \{x \in \Omega, \quad \phi_1 > 0, \quad \phi_2 > 0\}, \\ \Omega_2 &= \{x \in \Omega, \quad \phi_1 > 0, \quad \phi_2 < 0\}, \\ \Omega_3 &= \{x \in \Omega, \quad \phi_1 < 0, \quad \phi_2 > 0\}, \\ \Omega_4 &= \{x \in \Omega, \quad \phi_1 < 0, \quad \phi_2 < 0\}. \end{aligned} \tag{12}$$

Using the Heaviside function again, we can express q with possibly up to four pieces of constant values as:

$$\begin{aligned} q \;=\; & q_1 H(\phi_1) H(\phi_2) + q_2 H(\phi_1)(1 - H(\phi_2)) + \\ & + q_3 (1 - H(\phi_1)) H(\phi_2) + q_4 (1 - H(\phi_1))(1 - H(\phi_2)). \end{aligned} \tag{13}$$

By generalizing, we see that n level set functions give the possibility of 2^n regions. For $i = 1, 2, \cdots, 2^n$, let $\text{bin}(i-1) = (b_1^i, b_2^i, \cdots, b_n^i)$ be the binary representation of $i-1$, where $b_j^i = 0$ or 1. The representation of q would look like:

$$q = \sum_{i=1}^{2^n} q_i \prod_{j=1}^{n} R_i(\phi_j), \tag{14}$$

where

$$R_i(\phi_j) = \begin{cases} \quad H(\phi_j), & \text{if } b_j^i = 0; \\ 1 - H(\phi_j), & \text{if } b_j^i = 1. \end{cases}$$

Even if we the true q needs less than 2^n distinct regions, we can still use n level set functions since some subdomains are allowed to be empty. In using such a representation, we only need to determine the maximum number of level set functions we want to use before we start.

Gradient type methods will be used to find the minimizers with respect to q_i and ϕ_i. To illustrate our approach, let us consider a very general minimization problem:

$$\min_{q \in V} F(q), \tag{15}$$

where V is a space or set containing piecewise constant functions and possibly with some other extra constraints. Such kinds of minimization problems arise from many inverse problems and optimal shape design problems.

For any given $q \in V$, it can be represented as in (14). Using the chain rule, it is easy to see that the following relations hold:

$$\frac{\partial F}{\partial q_i} = \int_\Omega \frac{\partial F}{\partial q}\frac{\partial q}{\partial q_i} dx, \qquad \frac{\partial F}{\partial \phi_i} = \frac{\partial F}{\partial q}\frac{\partial q}{\partial \phi_i}. \tag{16}$$

For many inverse problems and optimal shape design problems, $\frac{\partial F}{\partial q}$ is known and there are ready softwares to compute them. In order to use the level set method, we just need to compute the derivatives $\frac{\partial q}{\partial q_i}$ and $\frac{\partial q}{\partial \phi_i}$.

Let us first consider a simple case where we only have one level set function and the piecewise constant function $q(x)$ is represented as in (11). Then it is easy to see that:

$$\frac{\partial F}{\partial q_1} = \int_\Omega \frac{\partial F}{\partial q} H(\phi) dx, \qquad \frac{\partial F}{\partial q_2} = \int_\Omega \frac{\partial F}{\partial q}(1 - H(\phi)) dx,$$

$$\frac{\partial F}{\partial \phi} = (q_1 - q_2)\delta(\phi)\frac{\partial F}{\partial q}. \tag{17}$$

In the above, δ denotes the Dirac function, i.e. $\delta(0) = 1$ and $\delta(x) = 0, \forall x \neq 0$. If we define $\Omega_1 = \{x | x \in \Omega,\ \phi > 0\}$, $\Omega_2 = \{x | x \in \Omega,\ \phi \leq 0\}$, then it is easy to see that:

$$\frac{\partial F}{\partial q_1} = \int_{\Omega_1} \frac{\partial F}{\partial q} dx, \qquad \frac{\partial F}{\partial q_2} = \int_{\Omega_2} \frac{\partial F}{\partial q} dx.$$

Now we consider the more general case of n level set functions as given in (14). From (14), we see that:

$$\frac{\partial q}{\partial q_i} = \prod_{j=1}^{n} R_i(\phi_j), \qquad \frac{\partial q}{\partial \phi_i} = \sum_{i=1}^{2^n} q_i \Big(\prod_{j=1, j \neq i}^{n} R_i(\phi_j) \Big) D(\phi_i), \tag{18}$$

where

$$D(\phi_i) = \begin{cases} \delta(\phi_i), & \text{if } b_j^i = 0; \\ -\delta(\phi_i), & \text{if } b_j^i = 1. \end{cases}$$

It can be seen that $\frac{\partial q}{\partial q_i}$ is nonzero only in the region corresponding to $q = q_i$. Inside this region, $\frac{\partial q}{\partial q_i} = 1$.

To be precise mathematically, the delta function $\delta(\phi)$ needs to be understood as the limit of some very smooth functions in a proper norm. In the numerical implementations, we sometimes replace the Heaviside function H by a C^∞ function H_ϵ. Correspondingly, we replace δ by δ_ϵ. The smooth function H_ϵ is chosen in a way such that the derivatives $\frac{\partial F}{\partial q_i}$ and $\frac{\partial F}{\partial \phi}$ involving H_ϵ and δ_ϵ converges to the corresponding derivatives of H and δ. The convergence of $\frac{\partial F}{\partial \phi}$ should be understood in the sense of pointwise convergence. In fact, it is better not to interpret H and δ functions as distributions.

4 Uzawa's Algorithm For Variational Level Set Methods

The Uzawa algorithm outlined in Algorithm 1 can be easily modified to incorporate the level set representation of the coefficient. Our approach is to replace the step corresponding to minimization with respect to q by a first step of minimizing with respect to q_i followed by a step of minimizing with respect to ϕ_j. The minimization steps are performed using a gradient based method with a line search. We outline the algorithm below.

Algorithm 2 *Choose initial q_i^0, ϕ_j^0, u^0 and λ^0. Set $k = 0$.*

- **(Update q_i's)** *Let $p^k = \left\{ -\frac{\partial L_r(q_i^k, \phi_j^k, u^k, \lambda^k)}{\partial q_i} \right\}_{i=1}^{2^n}$. Find α^k such that*

$$\alpha^k = \arg \min_{q_i^k + \alpha p_i^k \in [a_i, b_i], i=1,\cdots,2^n} L_r(q_i^k + \alpha p_i^k, \phi_j^k, u^k, \lambda^k). \qquad (19)$$

Set

$$\{q_i^{k+1}\}_{i=1}^{2^n} = \{q_i^k\}_{i=1}^{2^n} + \alpha^k p^k. \qquad (20)$$

- **(Update ϕ_j's)** *For $j = 1, 2, \cdots n$, define $\psi_j^k = -\frac{\partial L_r(q_i^{k+1}, \phi_j^k, u^k, \lambda^k)}{\partial \phi_j}$ and find σ_j^k such that*

$$\sigma_j^k = \arg \min_{\sigma_j \in R} L_r(q_i^{k+1}, \phi_j^k + \sigma_j \psi_j^k, u^k, \lambda^k). \qquad (21)$$

Set

$$\tilde{\phi}_j^k = \phi_j^k + \sigma_j^k \psi_j^k. \qquad (22)$$

- **(Reinitialization of ϕ_j's)** *If "necessary", reinitialize the level set functions $\{\tilde{\phi}_j^k\}_{j=1}^n$, i.e. set $d_0 = \tilde{\phi}_j^k$. Choose an appropriate τ_0 and solve equation (8) to $t = \tau_0$. The reinitialized ϕ_j^{k+1} is then taken as*

  $$\phi_j^{k+1} = d(x, \tau_0; \tilde{\phi}_j^k). \tag{23}$$

  Otherwise set $\phi_j^{k+1} = \tilde{\phi}_j^k$.

- **(Update u)** *Calculate $v^k = -\frac{\partial L_r(q_i^{k+1}, \phi_j^{k+1}, u^k, \lambda^k)}{\partial u}$. Find ξ_k such that*

  $$\xi^k = \arg\min_{\xi \in R} L_r(q_i^{k+1}, \phi_j^{k+1}, u^k + \xi v^k, \lambda^k). \tag{24}$$

  Set

  $$u^{k+1} = u^k + \xi^k v^k. \tag{25}$$

- **(Update λ)** *After each fixed number of iterations, update the Lagrangian multiplier as*

  $$\lambda^{k+1} = \lambda^k + re\left(q_i^{k+1}, \phi_j^{k+1}, u^{k+1}\right),$$

  Otherwise do not update λ, i.e. $\lambda^{k+1} = \lambda^k$. In the above, e is the equation error corresponds to q_i^{k+1}, ϕ_j^{k+1} and u^{k+1}.

- *Go to the next iteration for k.*

The reason we need the reinitialization step is because after the updating of ϕ_j in (22), the level set function may not be a distance function any more. See §5.3 for explanations about when it is "necessary" to reinitialize the level set functions.

For Algorithm 2, we need to calculate the derivatives of L_r with respect to $\{q_i\}_{i=1}^{2^n}$, $\{\phi_j\}_{j=1}^n$ and u, i.e. we need to compute p^k, ψ_j^k and v^k for each iteration. In addition to the calculations of the derivatives, we also need to do three line-searches as in (19), (21) and (24). See §5.2 about some cautions that shall be taken for the line-searches. In the Appendix, we shall give the details about the calculation of $\frac{\partial L_r}{\partial q}$. To get p^k and ψ_j^k at each iteration, we need to calculate $\frac{\partial L_r}{\partial q_i}$ and $\frac{\partial L_r}{\partial \phi_j}$ and this can be easily done using (16).

5 Implementation Issues

We discuss here several numerical issues that arise in the implementation of Algorithm 2.

5.1 Smooth Approximations to H and δ Functions

In numerical implementations, it is desirable to replace the Heaviside function H and the delta function δ by some smoothed counterparts. In our simulations, the following smoothed functions for the Heaviside-function H and delta-function δ have been used (c.f. [44]):

$$H_\epsilon(\phi) = \frac{1}{\pi}\tan^{-1}\frac{\phi}{\epsilon} + \frac{1}{2}, \tag{26}$$

$$\delta_\epsilon(\phi) = \frac{\epsilon}{\pi\left(\phi^2 + \epsilon^2\right)}. \tag{27}$$

In order to have good accuracy, we need to choose ϵ sufficiently small. For small ϵ, δ_ϵ is a smooth function, but with very sharp singular layers. This makes it difficult to represent the δ_ϵ function by finite element functions which is needed for $\frac{\partial L_r}{\partial \phi_j}$ and perform the numerical integrations needed to get the derivatives $\frac{\partial L_r}{\partial q_i}$. From our numerical experience, it was found that it is not good to use too small ϵ for H_ϵ and δ_ϵ.

If there is no observation error or there is very little observation error, we shall be able to recover a rather accurate coefficient. For such cases, we replace the δ function by δ_ϵ, but do not replace H by H_ϵ, as doing so will introduce an unnecessary error in the determination of the location of the discontinuities. The value of ϵ is always taken to be $\epsilon = h$ and h is the mesh size used for the spacial x and y variables.

If the observation error is large, we replace both H and δ by H_ϵ and δ_ϵ with $\epsilon = h$.

5.2 Precautions About The Line-searches

The equation error function e depends on ϕ_j, q_i and u. For fixed ϕ_j and u, e is linear with respect to q_i. For fixed ϕ_j and q_i, e is linear with respect to u. The regularization functional $R(q)$ defined in (3) is nonlinear with respect to ϕ_j, but linear with respect to q_i. Thus the Lagrangian functional L_r is quadratic with respect to q_i if the other variables are fixed and also quadratic with respect to u for fixed other variables. Correspondingly, we see that the minimization functions for linear searches (19) and (24) are quadratic functions. We can therefore derive explicit formulas for getting the minimizers α^k and ξ^k. However, cautions must be taken for the line-search (21).

In Figure (1), we plot the line-search functions

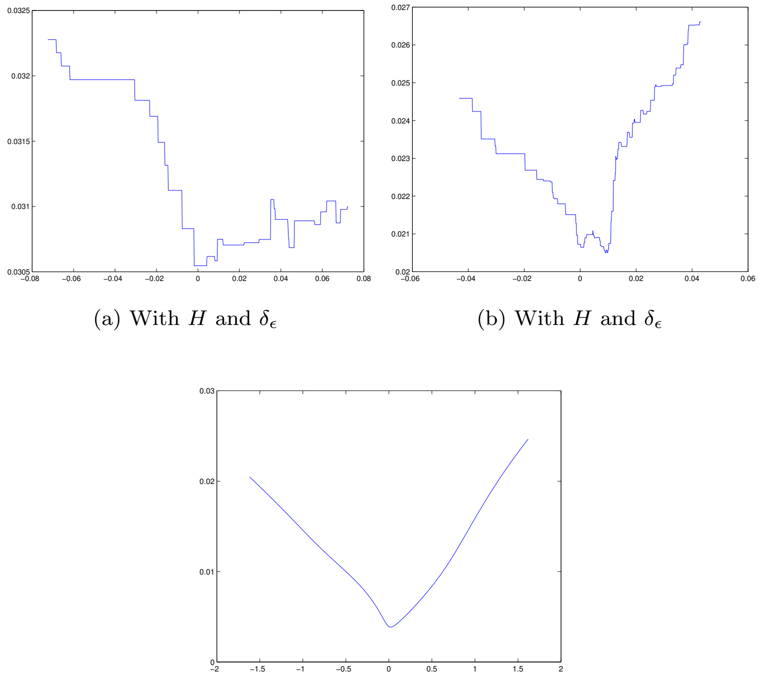
$$g(\sigma) = L_r(q_i^{k+1}, \phi_j^k + \sigma\psi_j^k, u^k, \lambda^k)$$

for some given k and j and these plots show some of the typical situations during the iterations. In Figure 1.c, the Heaviside function H is replaced by H_ϵ. In plots 1.a and 1.b, H is not replaced. With H, $g(\sigma)$ is a stair-case function. With H_ϵ, $g(\sigma)$ is a "smooth" function. In Figures 1.a and 1.c, $\sigma = 0$ is the minimizer for the line-search. In Figure 1.b, $\sigma = 0$ is a local minimizer, but there is another global line-search minimizer at $\sigma = 0.012$. Very often, the line search can miss the global minimizer in situations like in Figure 1.b.

We see from the above discussion that it happens very often in our simulations that the line-search (21) returns a value $\sigma_j^k = 0$, i.e. we are not able to decrease the function value, indicating that we are at a local minimizer with the ϕ_j functions. If this happens, we always take:

$$\sigma_j^k = \sigma_{fixed}. \tag{28}$$

This corresponds to a standard gradient descent step with step size σ_{fixed}. The fixed step size σ_{fixed} should be chosen properly in order to avoid stability problems. The step sizes obtained by the line-searches could give us a hint about what range the fixed step sizes should be. Another good strategy is to use the line-searches in the beginning stage of the iterations and then switch to gradient descent with fixed step sizes for all subsequent steps. We did not use this strategy in our numerical experiments.



(a) With H and δ_ϵ

(b) With H and δ_ϵ

(c) With H_ϵ and δ_ϵ

Figure 1: The plot for the line-search functions for (21) with H or H_ϵ.

Methods of the type of Golden-section are often used for line-searches. This kind of methods often can only find local minimizers. If the line-search function is a stair-case function, then every point is a local minimizer. If we use Golden-section type method, some extra checking needs to be added to avoid the case that the line-search is trapped in an interval where the line-search function is constant. In our simulations, we check that the values at the Golden-section points are not constant for a fixed number of updatings. If so, we just enlarge the search interval by a given factor.

5.3 When to Reinitialize The Level Set Functions

The cost to reinitialize a level set function to a distance function is rather cheap and we could do the reinitialization at each time step. However, there are two disadvantages if we reinitialize too often. First, the scheme (10) is a first order explicit scheme. It is known that the first order scheme is adding a certain amount of diffusion to the obtained solution. If we reinitialize too often, the added diffusion is too much and we may not be able to recover geometries with sharp corners. Second, our inverse problem is ill-posed. Even when the computed ϕ_j is still far from the true function, the gradient ψ_j^k can be already very small. Very often, the correction value $\sigma_j^k \psi_j^k$ is so small that it takes many iterations to change the sign of a nodal value. If we reinitialize too often, it is possible that none of the nodal values has changed sign and we are not able to move the level set functions.

In our simulations, we reinitialize a level set function when the level set function undergoes a sufficient amount of change. The criteria we have used is that the L^2-norm of a level set function has changed more than a given percentage. Other criteria can surely be used also. However, if the level set function is not reinitialized for an intensive long period, for example, a fixed number of iterations, we reinitialize it any way.

6 Numerical Experiments

We tested Algorithm 2 on several two dimensional problems. Let $\Omega = (0,1) \times (0,1), f = 20\pi^2 \sin(\pi x)\sin(\pi y)$. Let u^* be the exact solution for the exact q and σ be the noise level. We get $\tilde{u}_d$ by $\tilde{u}_d = u^* + \sigma \|u^*\|_{L^2} / \|R_d\|_{L^2} R_d$. Here R_d is a finite element function with nodal values being uniform random numbers between $[-1,1]$ with zero mean. We then apply the denoising technique of Chan and Tai [6] to smooth $\tilde{u}_d$ and the smoothed function from $\tilde{u}_d$ is used as the observation data u_d for our identification problem. In using the techniques of [6], we are not eliminating the noise. Instead, the noise is diffused and the obtained u_d satisfies the following equalities:

$$\|u_d - \tilde{u}_d\|_{L^2(\Omega)} = \|u^* - \tilde{u}_d\|_{L^2(\Omega)} = \sigma \|u^*\|_{L^2(\Omega)}.$$

The domain Ω is first divided into a rectangular mesh with uniform mesh size h for both the x and the y variables, i.e. $\Delta x = \Delta y = h$. The coefficient q and the level set functions are approximated by piecewise constants over this rectangular mesh and we denote P_h the piecewise contant functions space over this mesh. Thus, the derivatives $\frac{\partial F}{\partial q}, \frac{\partial q}{\partial \phi_i}, \frac{\partial q}{\partial q_i}$ are piecewise constants over the rectangular mesh. Finite difference method is used to reinitialize the level set functions as in (10). The nodal value d_{ij}^k is in fact the piecewise constant value of d^k over the (i,j) rectangular element.

The functions e, u and λ can be approximated by finite difference or finite element methods. In order to re-use some softwares that we already have, finite element approximations are used for e, u and λ. The finite element mesh is produced from the rectangular mesh by dividing each square into two triangular elements using the diagonal of positive slope. Piecewise continuous linear functions are used for the finite element space and we denote this piecewise linear finite element space by S_h.

As line search is used in our simulations, the numerical integration used for $L_r(q, u, \lambda)$ must be consistent with the discretized approximations we use to get $\partial L_r(q, u, \lambda)/\partial q$ and $\partial L_r(q, u, \lambda)/\partial u$. To explain the details, let us use A_q to denote the matrix A_q corresponding to the following standard discrete elliptic operator

$$(A_q u, v) = (q\nabla u, \nabla v), \quad \forall u,\ v \in S_h \subset H_0^1(\Omega). \tag{29}$$

Assume that $\{\psi_i\}$ are the basis functions for S_h and $\{\phi_i\}$ are the basis functions for P_h. For a given q, the matrix $A_q = [a_{ij}]_{n_1 \times n_1}$ has the entries $a_{ij} = \int_\Omega q\nabla\psi_i \cdot \nabla\psi_j$. We also use B_u to denote the matrix corresponding to

$$(B_u q, v) = (\nabla u \cdot \nabla v, q), \quad \forall q \in P_h, \quad \forall v \in S_h, \tag{30}$$

For a given u, the matrix $B_u = [b_{ij}]_{n_1 \times n_2}$ has the entries $b_{ij} = \int_\Omega \phi_i \nabla u \cdot \nabla \psi_j$. It is easy to see that B_u is generally neither square nor symmetric. Let $C = (A_q)|_{q=1}$. For any given function v, we use $\mathbf{v}$ to denote the vector containing the nodal values of v. From (2), it is easy to see that the nodal values $\mathbf{e}$ of e with a given q and u can be obtained by

$$\mathbf{e} = C^{-1}(A_q\mathbf{u} - \mathbf{f}) = C^{-1}(B_u\mathbf{q} - \mathbf{f}).$$

As q is piecewise contant and u, λ are piecewise linear, all the integrations needed to get A_q and B_u can be done exactly. Similarly, all the integrations needed to get $L_r(q, u, \lambda)$ can be done exactly. When we calculate $\partial L_r(q, u, \lambda)/\partial q$ and $\partial L_r(q, u, \lambda)/\partial u$, we need to use the matrices corresponding to these exact integrations. In our calculations, the matrices are never assembled. Instead, we just have some subroutines to calculate the product of the matrices on some given vectors. In practical implementations, we need to invert the matrix C to get C^{-1}. The matrix C^{-1} can be replaced by domain decomposition or multigrid preconditioners for the Laplacian operator, see [39].

In all the examples, the mesh size is $h = 1/64$. The time step for solving (10) is taken as $\tau = 0.01h$. We choose $\tau_0 = h$ when the level set functions are reinitialized as in (23). The level set functions are reinitialized when the L^2 norm has changed more than 10% or it has not been updated for 200 iterations (for some of the tests, we use 500 iterations).

If the observation contains no noise, then we know that $u = u_d$ is the minimizer and therefore we can keep the Lagrangian multiplier λ to be zero all the time and we also do not need to update u. The delta function δ is replaced by δ_ϵ with $\epsilon = h$, but the Heaviside function H is not replaced by H_ϵ. In case that the observation contains noise, we replace both H and δ by H_ϵ and $\delta_\epsilon, \epsilon = h$.

In theory, the Augmented Lagrangian method should converge for any $r > 0$. In practice, the value of r must be properly chosen in order to have a good convergence. Even with a properly chosen r, the convergence is normally faster in the beginning and then slows down. One typical thing is that it is very difficult to force the equation error function e to the machine accuracy.

If the Lagrangian multiplier λ is updated too often, the algorithm is not stable. In our simulations, we choose to update it after a fixed number of iterations. How often do we need to update depends on the problem. For example 1, we update after each 20 or 50 iterations. For the other examples, we update after 200 or 500 iterations. If we can not find a proper interval to update, we simply keep λ to be zero all the time. This reduces the Augmented Lagrangain method to the penalization method.

In some of the figures, the dotted lines in the background show the true zero level curves and the dashed lines are the computed zero level curves. We omit the numerical experiments we have done for problems (**P2**) and (**P3**). When the noise level is very low, Algorithm 2 could produce rather accurate solutions for all the three formulations (**P1**) – (**P3**). We can get a rather accurate solution for formulation (**P2**) with noise level as high as 50 times the noise level that we can handle with formulations (**P1**) and (**P3**).

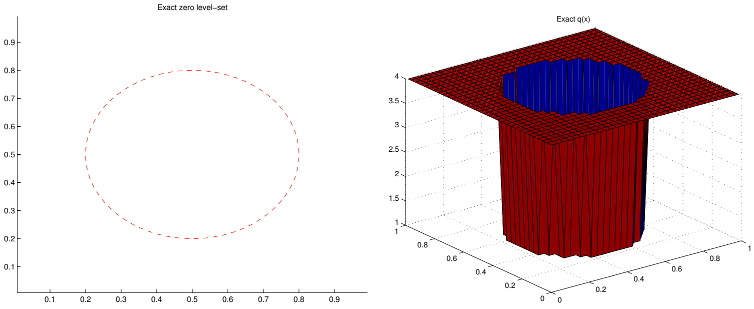
6.1 Example 1

We first test a simple problem. The exact coefficient $q(x)$ is given in Figure 2, i.e. $q(x) = 1$ inside a circle and $q(x) = 4$ outside the circle. In Figure 3, we add 5% noise, i.e. $\sigma = 5\%$ and the zero level set for the computed solution at different iterations are shown. Notice that we start with an initial guess for q with the location of the discontinuities being a small circle and the constant values specified to be the lower bounds a_i's. We see that, even with such a poor initial guess for q, Algorithm 2 is able to recover both the constant values and the location of the discontinuities quite well. In fact, after only about 100 steps, the recovered coefficient is already quite accurate.

Next, we show results corresponding to different noise levels. In Figure 4, we show the identified q and the corresponding zero level set curve for different amount of noise. For this example, we see that the coefficient can be recovered reasonably well even in the presence of up to 20% noise.

We also studied the effect of the regularization parameter β. In Figure 5, we show the results for a range of β values with $\sigma = 5\%$ noise. As expected, with larger β value, the identified $q(x)$ and zero level set curve become smoother. The recovered constant values remains good. This is true for all the examples presented in this section. We can clearly see that $\beta = 10^{-9}$ is too small and the identified zero level set curve is oscillatory. Comparing the results in Figure 5, it seems that $\beta = 10^{-5}$ is a good choice.

In Figure 6, we consider a case where the true $q(x)$ is a piecewise smooth, but not constant, function. In this case, Algorithm 2 tries to approximate q by a piecewise constant function. The true coefficient we used is given by $q(x) = c(x)e^{8x(1-x)y(1-y)}$ with $c(x) = 4$ outside the circle and $c(x) = 1$ inside



(a) Exact zero levet set

(b) Exact $q(x)$

Figure 2: The exact $q(x)$ and the location of the discontinuity

the circle. The noise level is set at 5%. The recovered q is shown in Figure 6.d. We see that the location of the discontinuities is recovered very accurately. Moreover, the recovered q is a pretty good piecewise constant approximation to the true q. In Figure 6.e, the error $\|q - q^k\|_{L^2(\Omega)}$ is plotted. The plots in Figure 6.f show the $\|e^k\|_{L^2(\Omega)}$ and $\|r^k\|_{L^2(\Omega)}$ and e^k is the equation error at iteration k and r^k is the projection of the residual $-\nabla \cdot (q^k \nabla u^k) - f$ into the finite element space. It is typical that the augmented Lagrangian method cannot reduce the equation error to machine accuracy. The L^2 error $\|q - q^k\|_{L^2(\Omega)}$ is reduced from 2.5 to 0.6 and stays at 0.6. The error connot be reduced further.

Another observation we can make is that the computed solutions converge to the true solution faster in the beginning. After this initial phase, the convergence slows down and it takes a lot of iterations to move the computed solution to true solution. This is true for the computational results of Figure 3 and also for all the other examples. This is quite typical of augmented Lagrangian type methods. Also, due to the ill-posedness of the inverse problem, the derivatives $\frac{\partial L_r}{\partial q}$ goes to zero much faster than the error of the computed solutions. It can happen that when the computed solution is still far from the true solution, the derivatives have already approached zero. It would be natural to use large step sizes when this happens. However, the algorithm is unstable if the step sizes are increased. This is the reason for the slow convergence near the solution.

For all the tests for this example, the lower and upper bound used for the line-search (19) are taken as: $a_1 = 0.5, b_1 = 2, a_2 = 2, b_2 = 8$.

6.2 Example 2

In this example, we try to identify a more complicated geometry for the location of discontinuities. The exact coefficient $q(x)$ is given in Figure 7, i.e. $q(x) = 2$ inside the two closed curves and $q(x) = 1$ outside the curves. Even though there appears to be 3 distinct piecewise constant regions, since 2 of the constant values are identical, only one level set function is needed. The zero level set for the computed solution with $\sigma = 0$ (i.e. noiseless) at different iterations are shown in Figure 9. We start with an initial guess for q with the location of discontinuities specified as a circle as shown, and with constant values given by the lower bounds a_i. We see that starting with one initial curve, Algorithm 2 is able to automatically split it into 2 pieces and capture the two separate regions successfully. This is the intrinsic advantage of the level set approach. Moreover, the recovered constant values are quite accurate. The errors for the computed quantities are shown in Figure 8. The errors are reduced from 1 to 10^{-5} and then oscillates around 10^{-5}. This oscillatory behavior is caused by the approximation error from the line-searches and the approximations we have done in the numerical integrations and differentiations.

In Figure 10, we show the results using a different initial curve. We see that the recovered coefficient is the same as using the previous initial curve, even though the intermediate stages are quite different. This lends confidence that Algorithm 2 is robust to the choice of the initial guess for q. The error $|q - q_i|$ and $\|q - q^k\|_{L^2(\Omega)}$ are reduced to 10^{-7} and then oscillates around it. We omit the plot for the errors.

In Figure 11, we show the results for two different noise levels: 1% and 5%. We see that we can only tolerate about 1% noise. With 5% of noise, the identified zero level set is rather poor. For the tests in Figure 11, we start with $|q_1^0 - q_1| = 0.4, |q_2^0 - q_2| = 0.8$ which gives a corresponding $\|q^0 - q\|_{L^2(\Omega)} = 0.6$. We use the same initial level set curve as in Figure 9. At convergence, we have $|q_1^k - q_1| = 0.01, |q_2^k - q_2| = 0.1$ and $\|q^k - q\|_{L^2(\Omega)} = 0.1$ for $\sigma = 1\%$. For $\sigma = 5\%$, we have $|q_1^k - q_1| = 0.008, |q_2^k - q_2| = 0.1$ and $\|q^k - q\|_{L^2(\Omega)} = 0.2$. It is interesting to note that even for the large noise (i.e. $\sigma = 5\%$) case, the constant values are recovered rather accurately, despite the fact that the recovered zero level curve is not so good. Note that the error in L^2 norm for the identified q for $\sigma = 1\%$ and $\sigma = 5\%$ is not so much different from each other, but the identified zero levelet curve is rather different.

For all the tests for this example, the lower and upper bound used for the line-search (19) are taken as: $a_1 = 0.5, b_1 = 2, a_2 = 1, b_2 = 4$.

6.3 Example 3

In this last example, we use the same geometry as in the last example, but the coefficient $q(x)$ takes two different constant values inside the two curves. The exact coefficient $q(x)$ is as given in Figure 12, i.e. $q(x) = 1$ outside the two curves, $q(x) = 2$ inside one of the curves and $q(x) = 3$ inside the another curve. Thus we have 3 distinct constant regions for q and therefore we need two level set functions to represent $q(x)$. The zero level set for the computed solution with $\sigma = 0.1\%$ at different iterations are shown in Figure 13. In this run, we start with an initial q with 3 distinct constant regions represented by the two circles and the background. The results show that not only are the location of the discontinuities recovered accurately, but also the constant values (the precise numerical values are not shown in the figure). After about 50 iterations, the approximate shape of the geometry of the discontinuities is already rather good, but it take many more subsequent iterations to further refine it. Note that in this example, the topologies of the discontinuities (i.e. the number of distinct constant regions) of the initial guess for q and the exact solution are the same and Algorithm 2 automatically determines that the region Ω_1 is empty.

In Figure 14, we show the results using an initial guess for q with a different topology than the true q — the initial q has 4 distinct constant regions. At first glance, it may appear that the recovered coefficient has 4 constant regions, which is one more than that for the true q. However, notice that a disconnected part of the interior of one level set function (corresponding to the heart-shaped region) is completely embedded in the interior of the other level set function. It turns out that the recovered constant value in this embedded region is the same as the recovered constant value for the surrounding region and therefore the recovered q is a good approximation to the true q. During the iteration, the shape of the embedded region continued to evolve as long as these two constant values are not the same. However, it may also happen that the embedded region can disappear before the two constant values become close enough, leaving only 3 regions. To summarize, starting with 4 constant regions, we may end up with either 3 or 4 constant regions for the recovered coefficient but in either case the accuracy is equally good. The fundamental reason for this phenomenon is due to the non-uniqueness of the multi-level set representation of piecewise constant functions with fewer constant regions than the maximum allowable.

In order to test how much noise we can tolerate, the computed zero level curves with noise $\sigma = 1\%$ and $\sigma = 5\%$ are shown in Figure 15. It seems that we cannot tolerate more than 1% of noise. For both tests, we start with $|q_1^0 - q_1| = 0.4, |q_2^0 - q_2| = 0.8, |q_3^0 - q_3| = 1.2$ with a corresponding $\|q^0 - q\|_{L^2(\Omega)} = 0.8$. The same initial level set curve as in Figure 13 is used. At convergence, we have $|q_1^0 - q_1| = 0.06, |q_2^0 - q_2| = 0.1, |q_3^0 - q_3| = 0.1$ and $\|q^k - q\|_{L^2(\Omega)} = 0.1$ for $\sigma = 1\%$. For $\sigma = 5\%$, we have $|q_1^0 - q_1| = 0.03, |q_2^0 - q_2| = 0.12, |q_3^0 - q_3| = 0.06$ and $\|q^k - q\|_{L^2(\Omega)} = 0.4$. We see that for large noise (i.e. $\sigma = 5\%$), the recovered zero level curve is not very good, but the constants are recovered rather accurately.

For all the tests for this example, the lower and upper bound used for the line-search (19) are taken as: $a_1 = 0.5, b_1 = 2, a_2 = 1, b_2 = 4, a_3 = 1, b_3 = 6$.

6.4 Example 4

Identifiibility for the inverse problems is rather important to get a corrected recovered coefficient, see [37, 12, 28] for some analsyis about the identifibility for the inverse problems. In this example, we try to show what may happen if the coefficient is not identifiable in part of the domain Ω. The true state $u \in H_0^1(\Omega)$ is chosen to be constant in a sub-domain $\Omega_c \subset \Omega$, i.e. $u =$ constant in Ω_c. The true coefficient q has two constant values, i.e. $q = 1$ or $q = 4$, see Figure 16. The coefficient has jumps on two curves. One of the curves is inside Ω_c and another one is outside. The f function for equation (1) is obtained from the known u and q. It is clear that q is not identifiable in the sub-domain Ω_c. In the test for Figure 16, we use the true u as the observation without adding noise. The recovered coefficient is shown in Figure 16.c. It is amazing to see that the discontinuity which is outside of Ω_c is recovered rather accurately. However, the algorithm is not able to identify the discontinuity inside Ω_c. The recovered contant values are $q_1 = 4.25$ and $q_2 = 0.79$ and it seems that it is difficult to get the contant values to a much better accuracy.

In practical applications, it is possible that the gradient $|\nabla u|$ is vary large in some regions and very small in some other regions. For such cases, we may also have problems to recover q inside the region where $|\nabla u|$ is very small and the recovered value may have very poor accuracy.

7 Conclusions

We have introduced a variational level set approach for elliptic inverse problems with piecewise constant coefficients. The key ingredients are a multi-level set representation of the coefficient and a way to regularize it in this representation using the total variation norm. A nice feature of our approach is that we do not require knowledge of the exact number of constant regions, only an upper bound is needed. Results from a range of numerical experiments show that the method can recover coefficients with rather complicated geometries of discontinuities under moderate amount of noise in the observation data. The method is also robust with respect to the initial guess for the geometry of the coefficient discontinuities, at least for low noise levels. Our formulation can be used with other forms of regularization and for more general optimal shape design problems.

8 Appendix: Calculations Of The Gradient Of The Minimization Functionals

We recall from (16) that we have:

$$\frac{\partial L_r}{\partial q_i} = \int_\Omega \frac{\partial L_r}{\partial q}\frac{\partial q}{\partial q_i}dx, \qquad \frac{\partial L_r}{\partial \phi_i} = \frac{\partial L_r}{\partial q}\frac{\partial q}{\partial \phi_i}. \tag{31}$$

Thus, we need to calculate $\frac{\partial L_r}{\partial q}$ for the functional L_r defined for (**P1**) - (**P3**). First, let us note that the derivative of the regularization functional $R(q)$ can be given as in the following:

$$\frac{\partial R}{\partial q} = -\nabla \cdot \left(\frac{\nabla q}{|\nabla q|}\right). \tag{32}$$

In order to get (32), we need to assume that

$$\beta|\nabla q|^{-1}\frac{\partial q}{\partial n} = 0, \qquad \text{on } \partial\Omega.$$

As the regularization parameter β is always chosen to be small, we neglect the contribution from the boundary term even in case that the above condition is not satisfied on the boundary of Ω.

The equation error function e defined in (2) depends on q and u. From the definition, it is easy to see that the Gateaux differentials of e with respect to q or u in a direction μ satisfies

$$(\nabla\left(\frac{\partial e}{\partial q}\cdot\mu\right),\nabla v)=(\mu\nabla u,\nabla v),\quad \forall v\in H_0^1(\Omega). \tag{33}$$

$$(\nabla\left(\frac{\partial e}{\partial u}\cdot\mu\right),\nabla v)=(q(x)\nabla\mu,\nabla v),\quad \forall v\in H_0^1(\Omega). \tag{34}$$

For problem (**P1**), we have

$$\left(\frac{\partial L_r}{\partial q}\cdot\mu\right)(q,u,\lambda)=\beta\left(-\nabla\cdot\left(\frac{\nabla q}{|\nabla q|}\right),\mu\right)+r\left(\frac{\partial e}{\partial q}\cdot\mu,e\right)+\left(\lambda,\frac{\partial e}{\partial q}\cdot\mu\right).$$

With the help of (33) , it is easy to see that

$$\left(\frac{\partial L_r}{\partial q}\cdot\mu\right)(q,u,\lambda)=\beta\left(-\nabla\cdot\left(\frac{\nabla q}{|\nabla q|}\right),\mu\right)+r(\mu\nabla u,\nabla e)+(\nabla\lambda,\mu\nabla e).$$

From the above formula, we get that

$$\frac{\partial L_r(q,u,\lambda)}{\partial q}=-\beta\nabla\cdot\left(\frac{\nabla q}{|\nabla q|}\right)+\nabla u\cdot\nabla(re+\lambda). \tag{35}$$

Using similar techniques and (34), one could get

$$\frac{\partial L_r(q,u,\lambda)}{\partial u}=u-u_d-\nabla\cdot(q\nabla(re+\lambda)). \tag{36}$$

For problem (**P2**), the calculation of $\frac{\partial L_r}{\partial q}$ is the same, and we can also easily get that

$$\frac{\partial L_r(q,u,\lambda)}{\partial u}=u-u_d-\gamma\nabla\cdot(\nabla u-\vec{u}_g)-\nabla\cdot(q\nabla(re+\lambda)).$$

We omit the details for problem (**P3**).

As explained in §6, piecewise constant functions P_h over a rectangular mesh are used to approximate q and the level set functions. Continuous piecewise linear functions S_h over a triangular mesh are used to approximate functions e,u and λ. From the definitions of the matrices A_q and B_u, we can see that discretized values of the differentials $\partial L_r(q,u,\lambda)/\partial q$ and $\partial L_r(q,u,\lambda)/\partial u$ given in (35) and (36) can be obtained by

$$\partial L_r(q,u,\lambda)/\partial\mathbf{q}=\beta R'(\mathbf{q})+B_u^*(r\mathbf{e}+\lambda); \tag{37}$$

$$\partial L_r(q,u,\lambda)/\partial\mathbf{u}=M(\mathbf{u}-\mathbf{u_d})+A_q(r\mathbf{e}+\lambda). \tag{38}$$

In the above, M is the stardard mass matrix which would be the identity matrix if we use reduced integrations only for the term $\|u - u_d\|^2_{L^2}$, B_u^* is the transpose of B_u and $\mathbf{q}, \mathbf{e}, \lambda, \mathbf{u}, \mathbf{u_d}$ are the vectors of nodal values for the corresponding functions.

The q function is approximated by piecewise constants over the rectangular mesh. In implementations, we approximate the regularization functional $R(q)$ by

$$R(q) = \sum_i \sum_j \sqrt{|q_{i,j} - q_{i-1,j}|^2 + |q_{i,j} - q_{i,j-1}|^2 + \epsilon h^2}. \tag{39}$$

In the above, $q_{i.j}$ is the value of q over the (i,j) rectangular element. It is known that $R(q)$ reduces to the TV-norm of q if we take $\epsilon = 0$. In order to get $R'(q)$, we just need to calculate $\partial R(q)/\partial q_{i,j}$ for all i and j. Using (39), it is easy to get the formula for calculating $\partial R(q)/\partial q_{i,j}$, which is essentially a finite difference approximation for $-\nabla \cdot \left(\frac{\nabla q}{\sqrt{|\nabla q|^2 + \epsilon}} \right)$. We always take ϵ small (for example $\epsilon = 10^{-10}$) to avoid dividing zero numbers.

References

[1] Kari Brusdal and Trond Mannseth. Basis norm rescaling for nonlinear parameter estimation. *SIAM J. Sci. Comput.*, 21(6):2114–2125 (electronic), 2000.

[2] M. Burger. A level set method for inverse problems. *Inverse problems*, 17:1327–1356, 2001.

[3] M. Burger. A framework for the construction of level set methods for shape optimization and reconstruction. preprint, 2002.

[4] T. F. Chan, B. Heimsund, T. K. Nilssen, and X.-C. Tai. Level set methods and augmented lagrangian for a parameter identification problem. *UCLA, Math. Depart., CAM-report 02-45*, 2002.

[5] T. F. Chan and X.-C. Tai. Augmented lagrangian and total variation methods for recovering discontinuous coefficients from elliptic equations. In M. Bristeau, G. Etgen, W. Fitzgibbon, J. L. Lions, J. Periaux, and M. F. Wheeler, editors, *Computational Science for the 21st Century*, pages 597–607. John Wiley & Sons, 1997.

[6] T F. Chan and X.-C. Tai. Identification of discontinuous coefficients in elliptic problems using total variation regularization. Technical report, CAM-report 97-35, 1997. Available online athttp://www.mi.uib.no/~tai/.

[7] Tony F. Chan and Luminita A. Vese. Image segmentation using level sets and the piecewise constant mumford-shah model. Technical report, CAM Report 00-14, UCLA, Math. Depart., April 2000. revised December 2000.

[8] G. Chavent, J. Jaffré, and S Jan-Jégou. Estimation of relative permeabilities in three-phase flow in porous media. conference on inverse problems, control and shape optimization (carthage, 1998). *Inverse Problems*, 15:33–39, 1999.

[9] G. Chavent and K. Kunisch. Regularization of linear least squares problems by total bounded variation. *ESAIM Control Optim. Calc. Var.*, 2:359–376 (electronic), 1997.

[10] Zhiming Chen and Jun Zou. An augmented Lagrangian method for identifying discontinuous parameters in elliptic systems. *SIAM J. Control Optim.*, 37(3):892–910 (electronic), 1999.

[11] Margaret Cheney, David Isaacson, and Jonathan C. Newell. Electrical impedance tomography. *SIAM Rev.*, 41(1):85–101 (electronic), 1999.

[12] C. Chicone and J. Gerlach. A note on the identifibility of distributed parameters in elliptic equations. *SIAM J. Math. Anal.*, 18:1378–1384, 1987.

[13] David Colton, Heinz W. Engl, Alfred K. Louis, Joyce R. McLaughlin, and William Rundell, editors. *Surveys on solution methods for inverse problems.* Springer-Verlag, Vienna, 2000.

[14] D. C. Dobson and F. Santosa. An image enhancement technique for eletrical impedance tomography. *Inverse problems*, 10:317–334, 1994.

[15] O. Dorn, E. M. Miller, and C. M. Rappaport. A shape reconstruction method for eletronomagnetic tomography using adjoint and level sets. *Inverse problems*, 16:1119–1156, 2000.

[16] R. Ewing, editor. *The mathematics of reservoir simulation.* SIAM, Philadelphia, 1983.

[17] Richard Ewing, Tao Lin, and Richard Falk. Inverse and ill-posed problems in reservoir simulation. In *Inverse and ill-posed problems (Sankt Wolfgang, 1986)*, pages 483–497. Academic Press, Boston, MA, 1987.

[18] Richard E. Ewing and Tao Lin. Parameter identification problems in single-phase and two-phase flow. In *Control and estimation of distributed parameter systems (Vorau, 1988)*, pages 85–108. Birkhäuser, Basel, 1989.

[19] Richard E. Ewing and Tao Lin. A class of parameter estimation techniques for fluid flow in porous media. *Adv. in Water Res.*, 14(2):89–97, 1991. Parameter identification in ground water flow, transport, and related processes, Part I.

[20] Richard E. Ewing, Tao Lin, and Yanping Lin. A mixed least-squares method for an inverse problem of a nonlinear beam equation. *Inverse Problems*, 15(1):19–32, 1999. Conference on Inverse Problems, Control and Shape Optimization (Carthage, 1998).

[21] Richard S. Falk. Error estimates for the numerical identification of a variable coefficient. *Math. Comp.*, 40(162):537–546, 1983.

[22] E. Giusti. *Minimal surfaces and functions of bounded variations.* Birkhäuser, 1984.

[23] Frank Hettlich and William Rundell. The determination of a discontinuity in a conductivity from a single boundary measurement. *Inverse Problems*, 14(1):67–82, 1998.

[24] K. Ito and K. Kunisch. The augmented lagrangian method for parameter estimation in elliptic systems. *SIAM J. Control Optim.*, 28:113–136, 1990.

[25] K.Ito, K. Kunisch, and Z. Li. Level-set function approach to an inverse interface problem. *Inverse Problems*, 17:1225–1242, 2001.

[26] C. Kravaris and H. Seinfeld. Identification of parameter in distributed systems by. *SIAM J. Contr. Optim.*, 23:217–241, 1985.

[27] K. Kunisch and X.-C. Tai. Sequential and parallel splitting methods for bilinear control problems in Hilbert spaces. *SIAM J. Numer. Anal.*, 34:91–118, 1997.

[28] K. Kunisch and L. W. White. Identifiability under approximation for an elliptic boundary value problem. *SIAM J. Contr. Optim.*, 25:279, 1987.

[29] Tao Lin and Richard Ewing. Parameter estimation for distributed systems arising in fluid flow problems via time series methods. In *Inverse problems (Oberwolfach, 1986)*, pages 117–126. Birkhäuser, Basel, 1986.

[30] A. Litman, D. Lesselier, and F. Santosa. Reconstruction of a two dimensional binary obstacle by control evolution of a level set. *inverse problems*, 14:685–706, 1998.

[31] Bruce D. Lowe and William Rundell. Unique recovery of a coefficient in an elliptic equation from input sources. *Inverse Problems*, 11(1):211–215, 1995.

[32] Bruce D. Lowe and William Rundell. The determination of a coefficient in an elliptic equation from average flux data. *J. Comput. Appl. Math.*, 70(1):173–187, 1996.

[33] S. Osher and J. A. Sethian. Fronts propagating with curvature dependent speed: Algorithms based on hamilton-jacobi formulations. *J. Comput. Phys.*, 79:12–49, 1988.

[34] Stanley Osher and Ronald R. Fedkiw. Level set methods. Technical report, CAM Report 00-08, UCLA, Math. Depart., February 2000.

[35] Stanley J. Osher and Fadil Santosa. Level set methods for optimization problems involving geometry and constraints. I. Frequencies of a two-density inhomogeneous drum. *J. Comput. Phys.*, 171(1):272–288, 2001.

[36] Danping Peng, Barry Merriman, Stanley Osher, Hongkai Zhao, and Myungjoo Kang. A PDE-based fast local level set method. *J. Comput. Phys.*, 155(2):410–438, 1999.

[37] G. Richter. An inverse problem for steady state diffusion equation. *SIAM J. Appl. Math*, 41:210–221, 1981.

[38] Fadil Santosa. A level-set approach for inverse problems involving obstacles. *ESAIM Contrôle Optim. Calc. Var.*, 1:17–33 (electronic), 1995/96.

[39] X.-C. Tai, J. Froyen, M. E. Espedal, and T. F. Chan. Overlapping domain decomposition and multigrid methods for inverse problems. In *Domain Decomposition methods 10, the tenth international conference on domain decomposition methods*, Contemporary Mathematics, vol. 218, pages 523–529. American Mathematical Society, 1998. See also URL: http://www.mi.uib.no/˜tai/.

[40] X.-C. Tai and T. Karkainen. Identification of a nonlinear parameter in a parabolic equation from a linear equation. *Comp. Appl. Mat.*, 14:157–184, 1995.

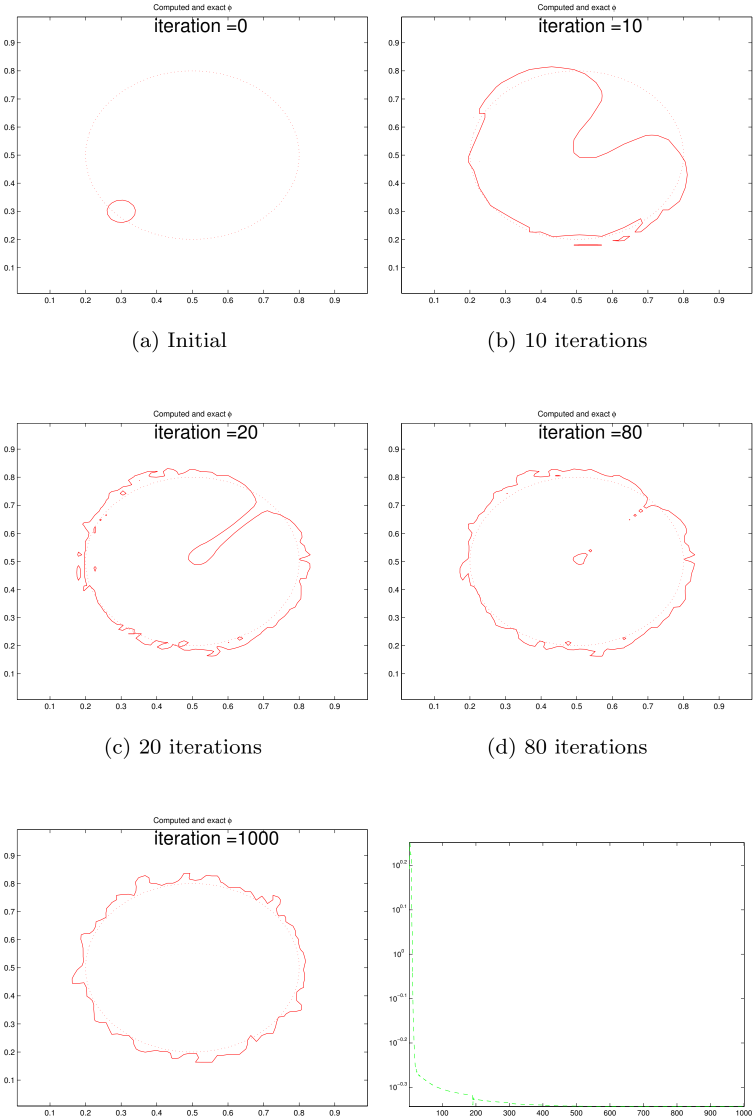
[41] Xue-Cheng Tai and Pekka Neittaanmaki. Error estimates for numerical identification of distributed parameters. *J. Comp. Math.*, Supplementary issue:66–78, 1992.

[42] Xue-Cheng Tai and Pekka Neittaanmaki. Pointwise error estimates for distributed parameter identification. In D. Bainov, editor, *Proceedings of the seventh international colloqium on differential equations*, pages 455–468, The Netherlands, 1997. International Science Publishers.

[43] L. Vese and S. Osher. The level set method links active contours, mumford-shah segmentation, and total variation restoration. *UCLA, Math. Depart., CAM-report 02-05*, 2002.

[44] Luminita A. Vese and Tony F. Chan. A new multiphase level set framework for image segmentation via the Mumford and Shah model. *International Journal of Computer Vision*, 50:271–293, 2002.

[45] W. P. Ziemer. *Weakly differentiable functions*. Springer-Verlag, 1989.



(a) Initial

(b) 10 iterations

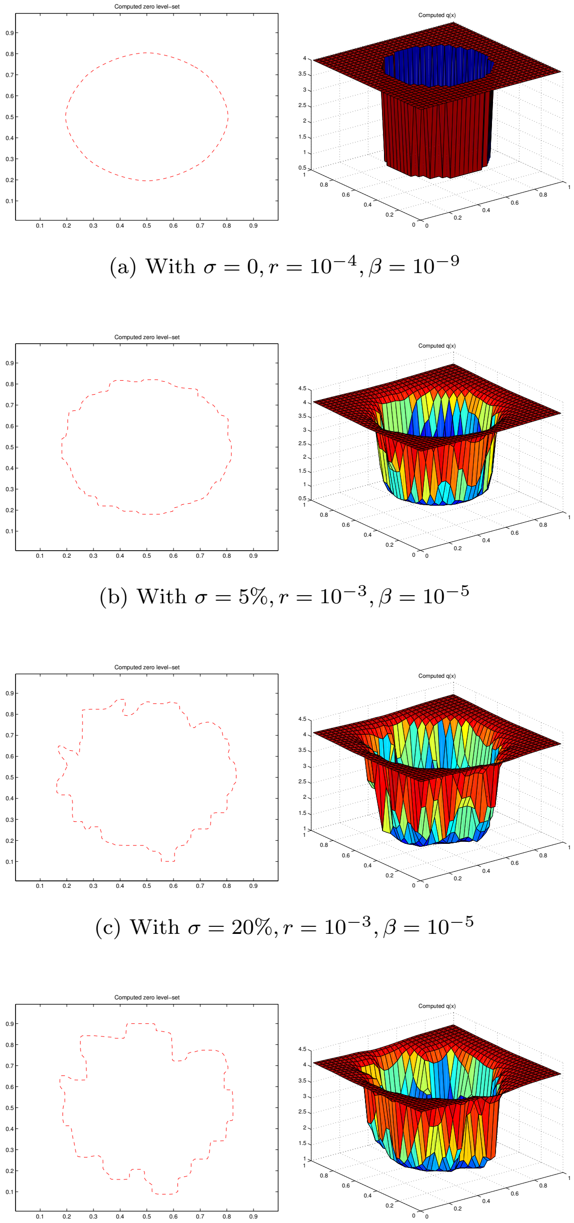
(c) 20 iterations

(d) 80 iterations

(e) 1000 iterations

(f) The error $\|q - q^k\|_{L^2(\Omega)}$ v.s. iteration number

Figure 3: The computed solution at different iterations and the computational error for Example 1 with $\sigma = 5\%, r = 10^{-3}, \beta = 10^{-6}$. Even with a rather poor initial guess, Algorithm 2 is able to recover both the constant values and the location of the discontinuities quite well. In fact, after only about 100 steps, the recovered coefficient is already quite accurate.



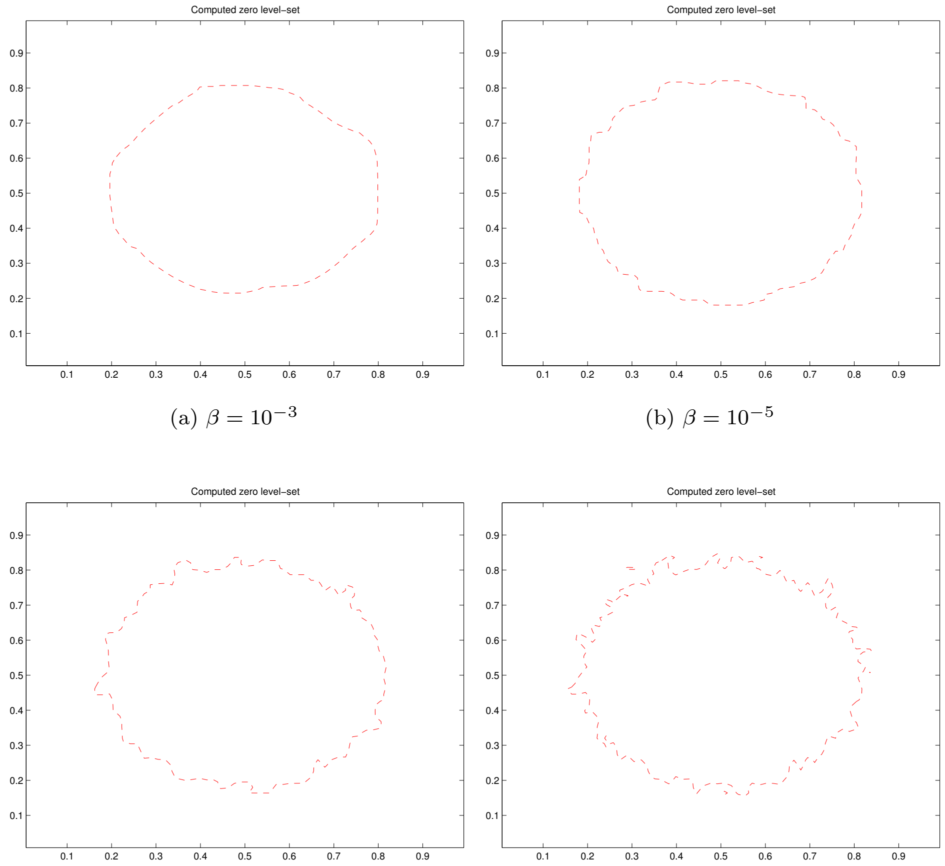
(a) With $\sigma = 0, r = 10^{-4}, \beta = 10^{-9}$

(b) With $\sigma = 5\%, r = 10^{-3}, \beta = 10^{-5}$

(c) With $\sigma = 20\%, r = 10^{-3}, \beta = 10^{-5}$

(d) With $\sigma = 40\%, r = 10^{-3}, \beta = 5 \times 10^{-5}$

Figure 4: The identified zero level set curve and the identified $q(x)$ with noise level $\sigma = 0\%, 5\%, 20\%, 40\%$. For noise level less than 20%, both the discontinuity location and the constant values are recovered accurately.



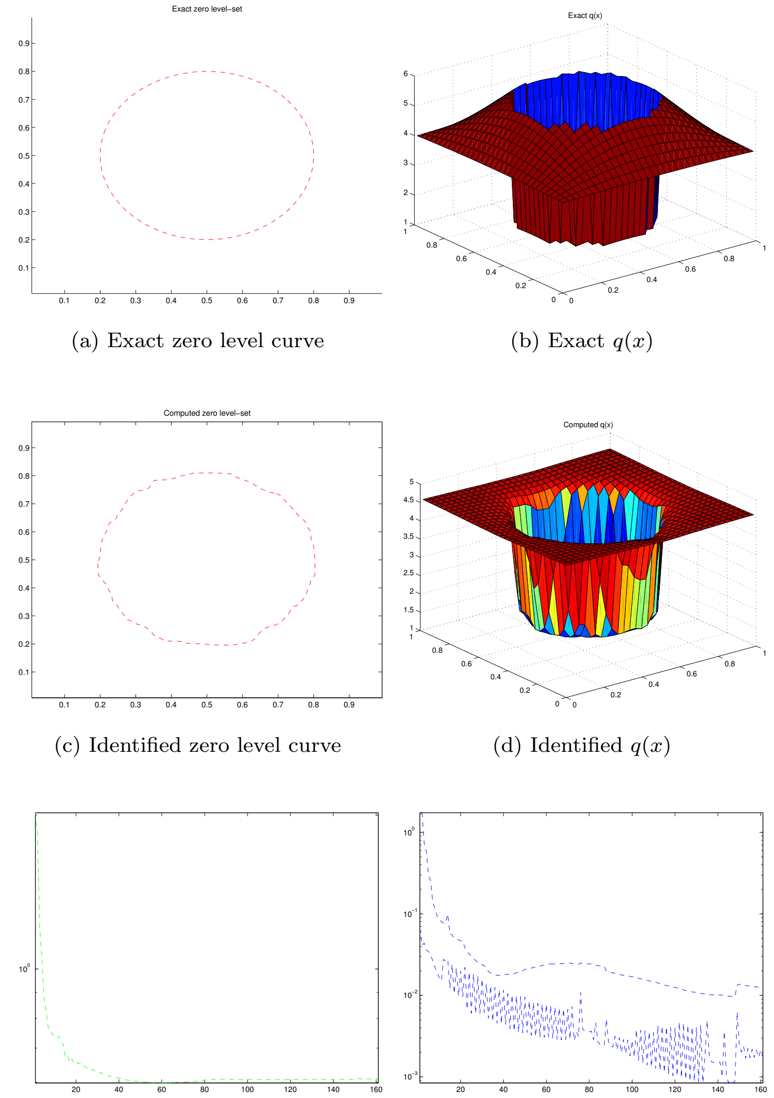
(a) $\beta = 10^{-3}$

(b) $\beta = 10^{-5}$

(c) $\beta = 10^{-6}$

(d) $\beta = 10^{-9}$

Figure 5: The identified zero level set curve with different β for $\sigma = 5\%$. With larger β, we get a smoother curve. The recovered constant values remains good. For this problem, it seems that $\beta = 10^{-5}$ is a good choice.



(a) Exact zero level curve

(b) Exact $q(x)$

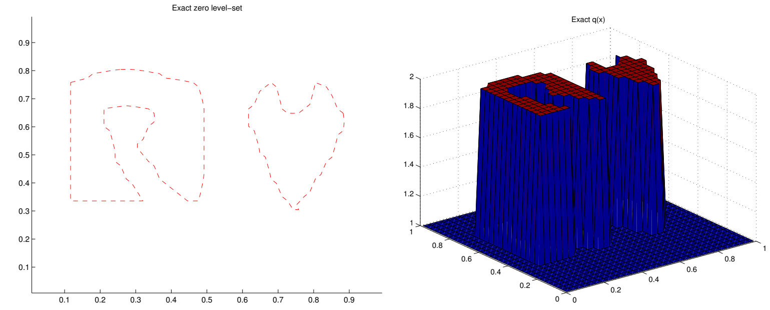
(c) Identified zero level curve

(d) Identified $q(x)$

(e) Error $\|q - q^k\|_{L(\Omega)}$ v.s iteration number

(f) Error $\|e^k\|_{L^2(\Omega)}$ and $\|r^k\|_{L^2(\Omega)}$v.s iteration number

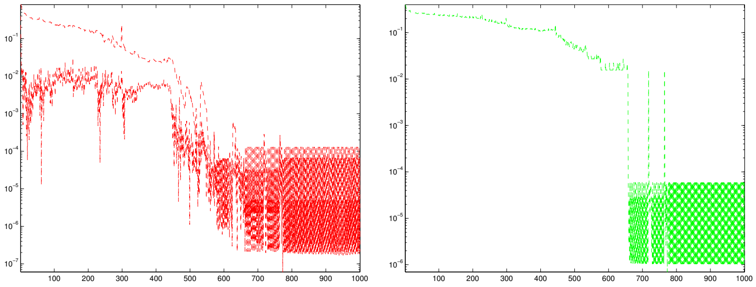
Figure 6: Identifying a piecewise smooth function by a piecewise constant function. We have $\sigma = 5\%, r = 10^{-3}, \beta = 10^{-5}$. The location of the discontinuities is recovered rather accurately. Moreover, the recovered q is a pretty good piecewise constant approximation to the true q. The L^2 error $\|q - q^k\|_{L^2(\Omega)}$ is reduced from 2.5 to 0.6 and stays at 0.6. The error connot be reduced further.



(a) Exact zero levet sets

(b) Exact $q(x)$

Figure 7: The exact zero level set curve and $q(x)$ for Example 2.



(a) Error $|q_i - q_i^k|$

(b) Error $\|q - q^k\|_{L^2(\Omega)}$

Figure 8: The convergence of the identified functions with $\sigma = 0$. The errors are reduced from 1 to 10^{-5} and then oscillates around 10^{-5}. This oscillatory behaviour is caused by numerical approximation errors.

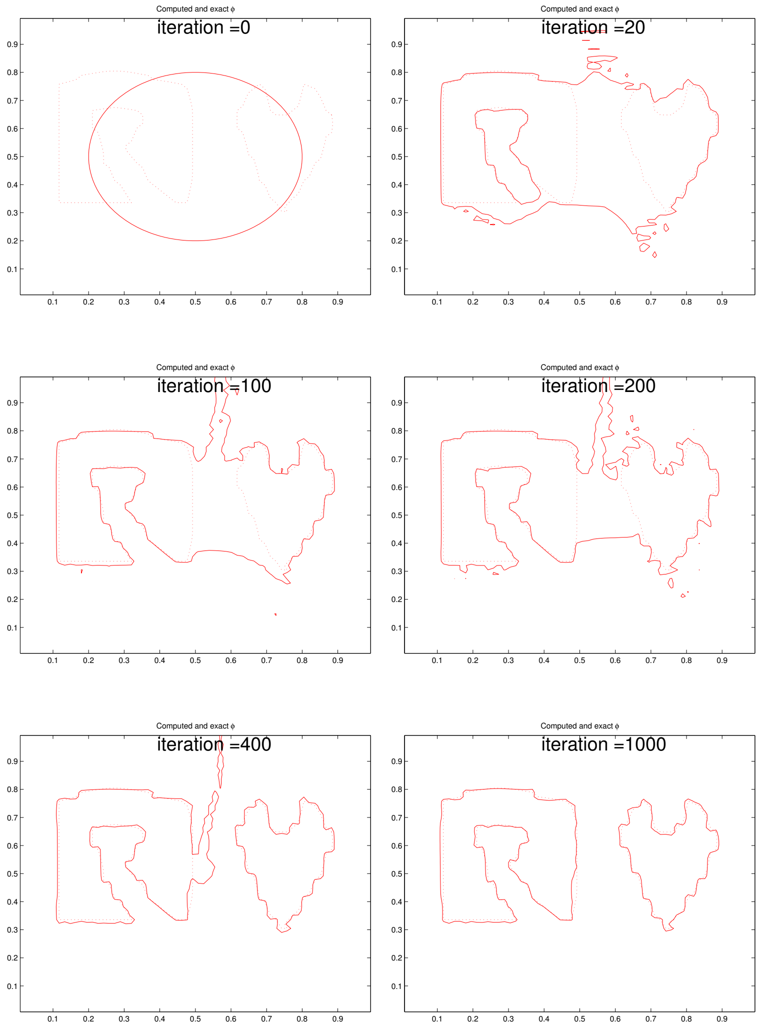


Figure 9: The identified zero level curve at differnet iterations with $\sigma = 0, r = 10^{-4}, \beta = 10^{-9}$. We see that starting with one initial curve, Algorithm 2 is able to automatically split it into 2 pieces and capture the two separate regions successfully. The approximate shape of the two objects is identified rather well after about 300 iterations.

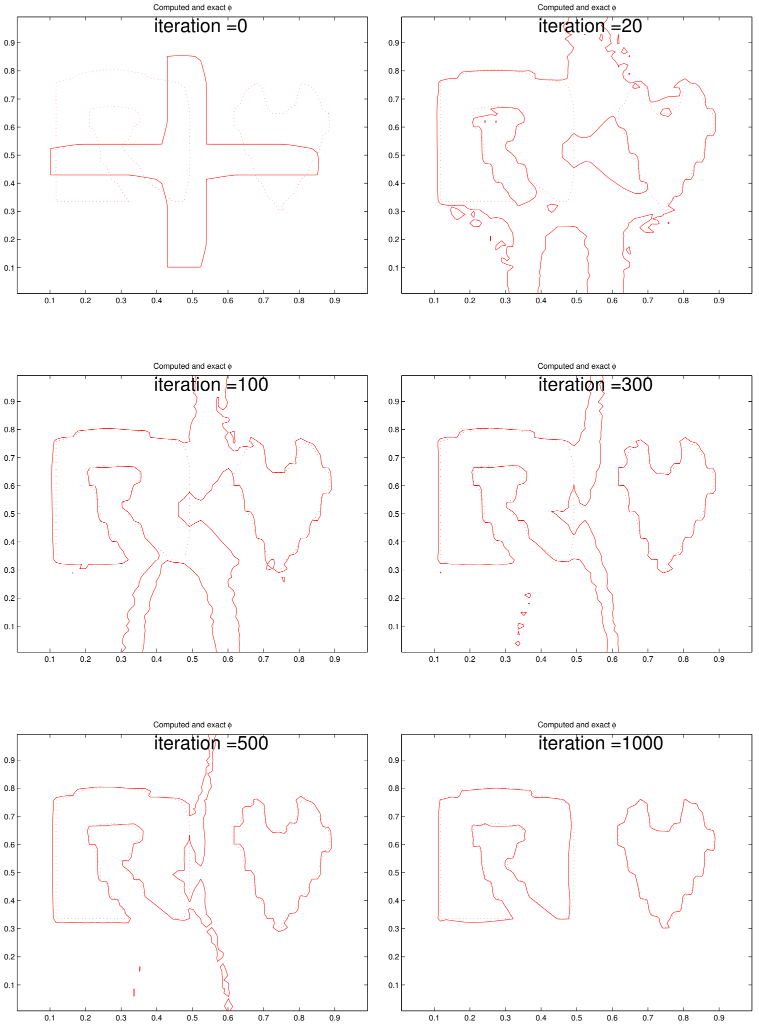
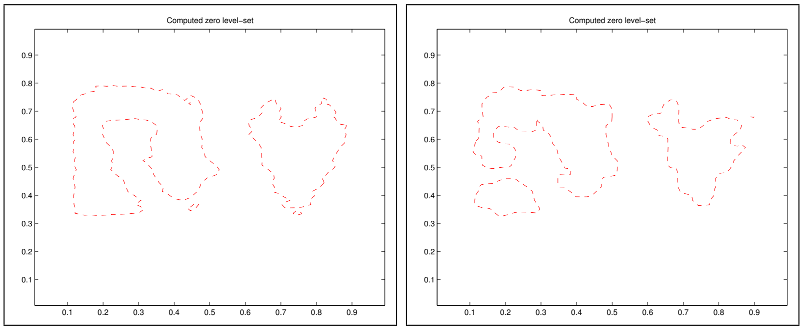


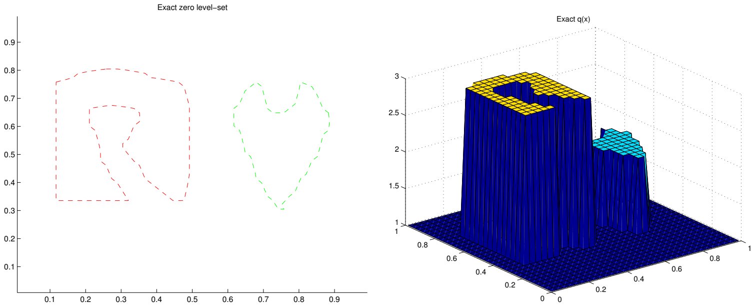
Figure 10: The identified zero level curve at differnet iterations with another initial guess and with $\sigma = 0, r = 10^{-4}, \beta = 10^{-9}$. We see that the recovered coefficient is the same as using the previous initial curve, even though the intermediate stages are quite different. The approximate shape of the two objects is identified rather well after about 500 iterations.



(a) With $\sigma = 1\%, r = 5\times10^{-4}, \beta = 5 \times 10^{-7}$

(b) With $\sigma = 5\%, r = 5 \times 10^{-4}, \beta = 5 \times 10^{-5}$

Figure 11: The identified zero level curve for different noise levels for Example 2. It seems that we can only tolerate about 1% noise. For the large noise (i.e. $\sigma = 5\%$) case, the recovered zero level curve is not so good, but the constant values are recovered rather accurately.



(a) Exact zero levet sets

(b) Exact $q(x)$

Figure 12: The exact zero level set curve and $q(x)$ for Example 3.

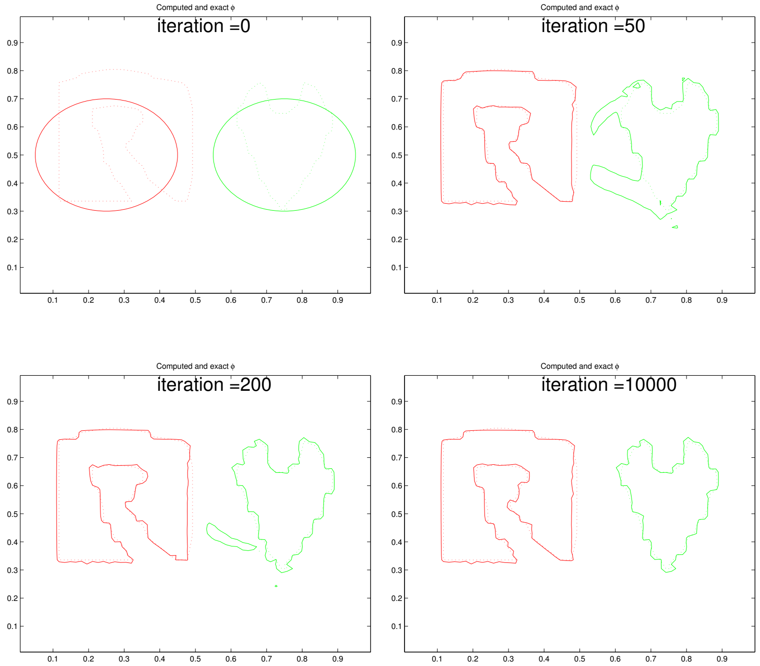


Figure 13: Identified zero level curve at different iterations with an initial guess that only gives three regions. After about 50 iterations, the approximate shape of the geometry of the discontinuities is already rather good, but it take many more subsequent iterations to further refine it. The topologies of the discontinuities (i.e. the number of distinct constant regions) of the initial guess for q and the exact solution are the same and Algorithm 2 automatically determines that the region Ω_1 is empty.

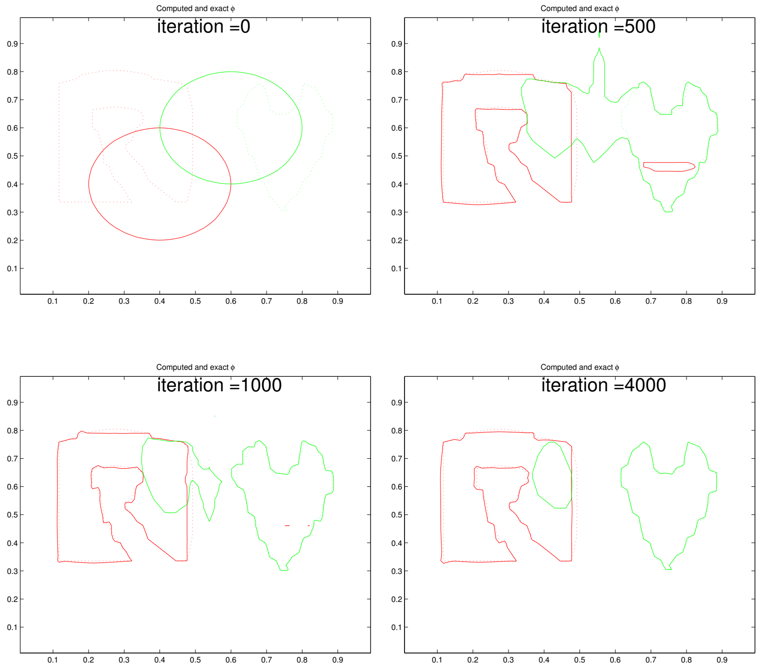
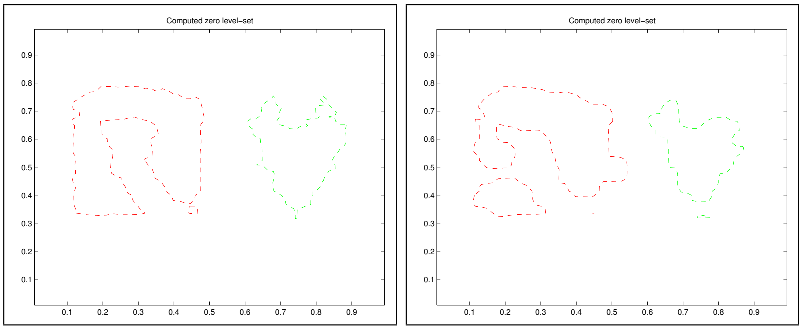


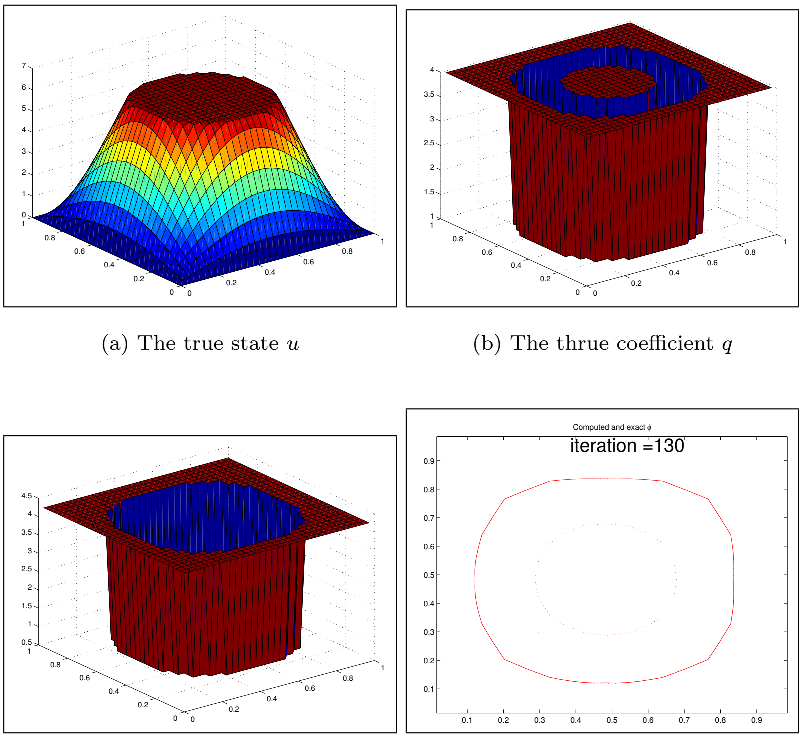
Figure 14: Identified zero level curve at different iterations with an initial guess that gives four regions. The extra region does not disappear and the constant for this region is equal to to the constant for the region surrounding it. This region stops to change as soon as the two constants are nearly the same. The final identified q is the same as in Figure 13. Both the discontinuity location and the constants are accurately recovered.



(a) With $\sigma = 1\%, r = 5\times10^{-4}, \beta = 5 \times 10^{-6}$

(b) With $\sigma = 5\%, r = 5 \times 10^{-4}, \beta = 5 \times 10^{-5}$

Figure 15: The identified zero level curve for different noise levels for Example 3. It seems that we cannot tolerate more than 1% of noise.



(a) The true state u

(b) The thrue coefficient q

(c) The recovered q

(d) The location of the exact and recovered discontinuities for q

Figure 16: The true state u is constant in a subdmain Ω_c and q is not identifiable in this subdomain. The coefficient has two discontinuities. One is inside Ω_c and another one is outside. The one which is outside Ω_c is recovered accurately. The algorithm is not able to identify the discontinuity inside Ω_c.