A Piecewise Constant Level Set Method for Elliptic Inverse Problems

Xue-Cheng Tai
Department of Mathematics,
University of Bergen, Norway
Email: tai@mi.uib.no

Hongwei Li
Center for Integrated Petroleum Research,
University of Bergen, Norway
Email: Hongwei.li@cipr.uib.no

March 26, 2006

Abstract

We apply a piecewise constant level set method to elliptic inverse problems. The discontinuity of the coefficients is represented implicitly by a piecewise constant level set function, which allows to use one level set function to represent multiple phases. The inverse problem is solved using a variational penalization method with the total variation regularization of the coefficients. An operator splitting scheme is used to get efficient and robust numerical schemes for solving the obtained problem. Numerical experiments show that the method can recover coefficients with rather complicated geometry of discontinuities under a moderate amount of noise in the observation data.

keywords: inverse problem, level set method, piecewise constant, operator splitting

1 Introduction

Consider the partial differential equation:

$$\begin{aligned} -\nabla \cdot (q(x)\nabla u) &= f \quad x \in \Omega \subset R^2, \\ u(x) &= 0 \qquad x \in \partial\Omega. \end{aligned} \tag{1.1}$$

The problem is to use some observations of the solution u to recover the coefficient $q(x)$. For many applications, the coefficient $q(x)$ is discontinuous. Sometimes, it is important to identify the location of the discontinuities as well as the values for the coefficient. In this work, we assume that $q(x)$ is discontinuous over some subdomain boundaries, and inside the subdomains, the function values have very little variations. Therefore, $q(x)$ can be approximated by piecewise constant functions.

Identification of discontinuous functions has been an interesting problem for sometimes. One traditional approach is to use proper regularization strategy to allow discontinuities in the recovered functions, see [7, 6, 9, 10]. Recently, the level set method, firstly proposed by Osher and Sethian in [25], has been extended and used for this kind of problems in [4, 8, 12, 17, 26, 14, 1, 2, 3]. In the recent surveys [29, 5], some more detailed information about these approaches and references can be found. In [17], the authors have used the level set idea to solve an inverse conductivity problem. Assuming known conductivity values, the unknown interface (i.e. the shape) can be identified by using the observed solutions in a thin layer along the boundary of a domain. In [12], the level set method has been applied to electrical impedance tomography problem. Given a set of current density, as well as the corresponding values of electrical potential on $\partial\Omega$, the method can determine the electrical conductivity inside Ω. In this paper, we assume that an observation for the solution is available at every mesh point in Ω. A new technique used in this work is to replace the traditional level set methods of [25] by some recent piecewise constant level set methods (PCLSM). The PCLSM was proposed by Tai et al. in [18, 31, 20, 19], and has been applied to image segmentation problems in [31, 30]. We shall also refer to [16, 28, 27, 15, 13, 21] for some related and similar ideas.

The remainder of this paper is organized as following. We will first formulate the model problem by a output-least-squares approach in Section 2, then the PCLSM will be introduced and incorporated into the formulation in Section 3. The algorithm will be described in Section 4. We will address some implementation issues in Section 5, and present our numerical experiments in Section 6. The last section goes for the conclusion.

2 Formulations for the elliptic inverse problem

Due to the ill-posedness of the problem, output-least-squares method is often used for recovering $q(x)$. Assume that $u_d \in L^2(\Omega)$ is an observation for u, and let K be the set of admissible coefficients

$$K = \{q | q \in L^\infty(\Omega) \cap TV(\Omega), \quad 0 < \underline{q}(x) \leqslant q(x) \leqslant \bar{q}(x) < \infty\}, \tag{2.1}$$

where $\underline{q}(x)$ and $\bar{q}(x)$ are known a priori, and TV denotes the total variation norm. The minimization functional for the output-least-squares method is

$$F(q) = \int_\Omega \frac{1}{2}|u(q) - u_d|^2 dx + \beta R(q), \tag{2.2}$$

where $R(q) = \int_\Omega |\nabla q| dx$ is the total variation norm of q, $u(q)$ is the solution of (1.1) for a given q and β is the regularization parameter which controls the regularity of the identified coefficient. We solve the following minimization problem to find $q(x)$:

$$\min_{q \in K} F. \tag{2.3}$$

We restrict the solution for $q(x)$ to the space of piecewise constant functions, i.e., we want to partition Ω into several subdomains, and in each subdomain $q(x)$ is a constant. We want to know the subdomains as well as the corresponding values inside the subdomains. This problem is very similar in nature to the "image segmentation" as studied in [31, 20]. We use the PCLSM as a mechanism to solve such a kind of problems involving interfaces.

3 Piecewise constant level set formulation

Suppose a function q, defined in the domain $\Omega \subset R^2$, has different properties in different subdomains of Ω. In many industrial applications, one needs to identify the subdomains of Ω. In other words, one needs to partition Ω into a union of subdomains, $\Omega = \bigcup_{i=1}^n \Omega_i$, so that each subdomain has a distinctive property of q. Usually we apply some variational models and try to find the subdomains and the properties by minimizing an energy functional. So we need some techniques to represent the subdomains and the properties so that they can be incorporated into the minimization process. The level set method proposed by Osher and Sethian [25], is a convenient and successful technique for this kind of interface problems, which has the advantage of dealing with topological changes of the implicitly represented curves. The curve that divides Ω into its subdomains, can be represented implicitly as the zero level set curve $\Gamma = \{x \in \Omega, \phi(x) = 0\}$ of a higher dimensional level set function ϕ. Rather than evolving the curve itself, the level set approach evolves the level set function ϕ. In the standard level set method, ϕ is represented as the signed distance function:

$$\phi(x) = \begin{cases} \text{distance}(x, \Gamma), & x \in \text{interior of } \Gamma, \\ -\text{distance}(x, \Gamma), & x \in \text{exterior of } \Gamma. \end{cases} \tag{3.1}$$

For multiple subdomains, one can use multiple level set functions [8]. In general, one can use n level set functions to represent 2^n subdomains.

The piecewise constant level set methods, proposed in [18, 20, 19], are alternatives to the level set idea of [25]. The PCLSM has been used in [31, 30] for image segmentation problems. Here, in this paper, we use them to solve the inverse problem.

We present the essential ideas of the PCLSM in the following. Assume that we need to partition the domain Ω into a set of subdomains Ω_i, $i = 1, 2, \ldots, n$, where the number of subdomains n is a priori known. In order to identify the subdomains, we try to identify a piecewise constant level set function ϕ such that

$$\phi = i, \quad \text{in } \Omega_i, \quad i = 1, 2, \ldots, n. \tag{3.2}$$

Associated with such a level set function ϕ, the characteristic functions of the subdomains are given as

$$\psi_i = \frac{1}{\alpha_i} \prod_{j=1, j\neq i}^{n} (\phi - j), \quad \alpha_i = \prod_{k=1, k\neq i}^{n} (i - k). \tag{3.3}$$

If ϕ is given as in (3.2), then we have $\psi_i(x) = 1$ for $x \in \Omega_i$, and $\psi_i(x) = 0$ elsewhere. Under the condition (3.2), any function given by

$$q(x) = \sum_{i=1}^{n} c_i \psi_i(\phi(x)) \tag{3.4}$$

is a piecewise constant function, with $q = c_i$ in Ω_i. In order to satisfy (3.2), we need to impose some constraints on ϕ. Define

$$K(\phi) = (\phi - 1)(\phi - 2) \cdots (\phi - n) = \prod_{i=1}^{n} (\phi - i). \tag{3.5}$$

The level set function ϕ should satisfy

$$K(\phi) = 0 \tag{3.6}$$

at every point in Ω. Thus, we can use (3.4) to represent q, and transform the minimization problem with respect to q into a minimization problem with respect to c_i and ϕ under the constraint (3.6). More precisely, we need to solve

$$\min_{\substack{c_i, \phi \\ K(\phi)=0}} F, \quad F = \int_\Omega \frac{1}{2}|u - u_d|^2 + \beta R(q), \tag{3.7}$$

to find the constant values c_i and the piecewise constant level set function ϕ. Above, q is a function of c_i and ϕ as in (3.4), and by the equation (1.1), u is related to q, thus u depends on c_i and ϕ.

One can transform the above constrained minimization problem to an unconstrained one by some kind of penalization or Lagrangian techniques. In this work, we use the penalization method, i.e. we choose a small constant $\mu > 0$ and solve the following minimization problem

$$\min_{c_i, \phi} L, \quad L = F + \frac{1}{2\mu} W = \int_\Omega \frac{1}{2}|u - u_d|^2 + \beta R(q) + \frac{1}{2\mu} \int_\Omega K^2(\phi) dx. \tag{3.8}$$

Here, $W(\phi) = \int_\Omega K^2(\phi) dx$. In order to solve (3.8), the derivatives $\frac{\partial L}{\partial c_i}$ and $\frac{\partial L}{\partial \phi}$ are needed. It is known that

$$\frac{\partial F}{\partial q} = -\nabla u \cdot \nabla z - \beta \nabla \cdot \left(\frac{\nabla q}{|\nabla q|} \right), \tag{3.9}$$

where u is the forward solution of (1.1) with a given $q(x)$, and $z(x) \in H_0^1(\Omega)$ is the solution of the adjoint problem

$$-\nabla u \cdot (q(x) \nabla z) = u - u_d \text{ in } \Omega, \quad z = 0 \text{ on } \partial\Omega. \tag{3.10}$$

According to (3.4) and (3.3), the computations of $\frac{\partial q}{\partial \phi}$ and $\frac{\partial q}{\partial c_i}$ are straightforward, i.e.,

$$\frac{\partial q}{\partial \phi} = \sum_{i=1}^{n} c_i \psi_i'(\phi), \quad \frac{\partial q}{\partial c_i} = \psi_i.$$

By using the chain rule, which was verified in the Appendix of [29], we have

$$\frac{\partial F}{\partial \phi} = \frac{\partial F}{\partial q}\frac{\partial q}{\partial \phi}, \qquad \frac{\partial F}{\partial c_i} = \int_\Omega \frac{\partial F}{\partial q}\frac{\partial q}{\partial c_i}, \qquad W'(\phi) = 2K(\phi)K'(\phi). \tag{3.11}$$

As a result of these relations, it is easy to get

$$\frac{\partial L}{\partial \phi} = \frac{\partial F}{\partial \phi} + \frac{1}{2\mu}W'(\phi) = \frac{\partial F}{\partial q}\frac{\partial q}{\partial \phi} + \frac{1}{\mu}K(\phi)K'(\phi), \tag{3.12}$$

$$\frac{\partial L}{\partial c_i} = \frac{\partial F}{\partial c_i} = \int_\Omega \frac{\partial F}{\partial q}\frac{\partial q}{\partial c_i}. \tag{3.13}$$

4 Algorithms

To find a minimizer of (3.8), we use the following general sequential algorithm.

Algorithm 1. Choose initial values for ϕ^0 and $c_i^0,\ i = 1, 2, ..., n$. For $k = 1, 2, \ldots$, do

1. Find $\bar{c}^{k+1} = \{c_i^{k+1}, i = 1, 2, \ldots, n\}$, such that

$$\bar{c}^{k+1} = \arg\min_{\bar{c}} L(\bar{c}, \phi^k). \tag{4.1}$$

2. Find ϕ^{k+1} such that

$$\phi^{k+1} = \arg\min_{\phi} L(\bar{c}^{k+1}, \phi). \tag{4.2}$$

3. Check the convergence, if converged, stop; else goto 1.

Here and after, we will use $\arg\min L$ to denote the minimizer of L. We use a gradient based method with a line search to find the minimizers for (4.1) and (4.2). In the computation, we assume that a good guess for the constants are known a priori, and that the minimization variables $\{c_i,\ i = 1, 2, ..., n\}$ are confined to intervals around the true values. The most time consuming part of the above algorithm is the minimization problem (4.2). The minimizer of (4.2) satisfies

$$\frac{\partial L}{\partial \phi} = \frac{\partial F}{\partial \phi} + \frac{1}{2\mu}W'(\phi) = 0. \tag{4.3}$$

The gradient descent method is often used to solve the above equation, i.e., we solve the following ordinary differential equation to the steady state

$$\phi_t + \frac{\partial L}{\partial \phi} = 0. \tag{4.4}$$

Normally, explicit schemes are used to solve this equation. In order to get better efficiency and stability, we use the operator splitting scheme of [22, 23] to solve the above equation. Given ϕ^0, we find $\phi^{k+1/2}$ and ϕ^{k+1} from the following equations for $k = 1, 2, \ldots$

$$\frac{\phi^{k+1/2} - \phi^k}{\tau} + \frac{\partial F}{\partial \phi}(\bar{c}^k, \phi^{k+1/2}) = 0, \tag{4.5}$$

$$\frac{\phi^{k+1} - \phi^{k+1/2}}{\tau} + \frac{1}{2\mu}W'(\phi^{k+1}) = 0. \tag{4.6}$$

The pseudo time-step τ needs to be chosen properly. Note that (4.6) can be re-written as:

$$\phi^{k+1} - \phi^{k+1/2} + \alpha_2 K(\phi^{k+1})K'(\phi^{k+1}) = 0, \tag{4.7}$$

where $\alpha_2 = \tau/\mu$. To solve this equation, we need to find the roots for a polynomial of order $2n-1$. By choosing τ and μ properly, we can guarantee that this equation only has one real root, and the Newton method can be used to find the root very efficiently, see [11]. The equation (4.7) will be referred to as the *constraint equation* later on.

Set $\alpha_1 = \frac{1}{\tau}$. Then the minimizer of the following problem is a solution of (4.5):

$$\min(\frac{1}{2\tau}||\phi - \phi^k||^2 + F) = \min(\frac{1}{2}\alpha_1||\phi - \phi^k||^2 + F). \tag{4.8}$$

So, we can apply a gradient like method with line search to determine the optimal step length. Incorporating these operator splitting schemes into Algorithm 1, we get the following algorithm:

Algorithm 2. Choose initial values for ϕ^0 and $c_i^0,\ i = 1, 2, ..., n$. For $k = 1, 2, \ldots$, do

1. Find $\bar{c}^{k+1}$ such that
$$\bar{c}^{k+1} = arg \min_{\bar{c}} L(\bar{c}, \phi^k).$$
2. Compute $\phi^{k+1/2}$ by
$$\phi^{k+1/2} = \phi^k - \gamma_k \frac{\partial F}{\partial \phi}(\bar{c}^{k+1}, \phi^k),$$
where γ_k is determined by a line search method.
3. Compute ϕ^{k+1} from (4.7) by a Newton method.
4. Check the convergence, if converged, stop; else goto 1.

5 Implementation issues

We would like to address some important issues in the implementation of Algorithm 2.

5.1 Regularization parameter

The regularization term in the first part F plays a very important role. Because of the ill-posedness of the inverse problem, the regularization term is ncecessary. The parameter β, which is used to control the influence of the regularization term, is vital not only for the convergence rate of the algorithm, but also for the computed solution. In this work, we use the Total-Variation (TV) regularization. The TV regularization controls both the length of the level set curves as well as the jumps of q over the solution domain. Usually, β should be chosen according to the noise level in the observation data.

From our numerical experiences, we find that it is better to neglect the regularization term at the beginning stage of the iteration. At this stage, we should let the output-least-squares term to drag ϕ into the right direction without thinking about the regularity of q. We adjust the value of β at a later stage. We set β to be large or small for each iteration according to some criteria. In our applications, when some conditions are met, we set $\beta = \beta_{max}$, or else set $\beta = \beta_{min}$. The constants β_{min} and β_{max} may have a certain dependence on the problem, and to a large extent, on the noise level. In our computations, we have used the following criterion:

Criterion for triggering regularization

a. The level set curves do not change any more.

b. F is decreasing, which means that $F(\phi^{k+1}) < F(\phi^k)$.

To testify **Criterion a.**, we define

$$\widetilde{\phi} = round(\phi),$$

where $round(\cdot)$ denotes the function that rounds a variable to its nearest integers. So we can check the norm $||\widetilde{\phi}^{k+1} - \widetilde{\phi}^k||$ to testify if the level set curves are reluctant to change.

We can write the Criterion as a function:

$$\beta = TriggerRegu(\phi^k, \phi^{k+1}). \tag{5.1}$$

During each iteration, we call this function to set β. If the criterion is met, then the function would return $\beta = \beta_{max}$, or else return $\beta = \beta_{min}$.

It should be pointed out that the above criterion is just based on our experiences, and we have only heuristic reasons rather than rigorous mathematical justifications to do so.

5.2 Updating c_i

We assume that good guesses for the values of c_i are known. In our numerical experiments, we update c_i after each 5 to 10 iterations. When updating c_i, we confine the c_i value to an interval which allows a 20% perturbation of the true value. We use the line search to determine the step length to update c_i by gradient method as in [8].

5.3 Constraint term

We also need to choose a proper value for α_2 for the constraint equation. In order to guarantee that (4.7) has only one real root, the value of α_2 needs to be chosen according to the phase number n. We need to have $\alpha_2 \leq 2$ for $n = 2$ and $\alpha_2 \leq 0.71$ for $n = 3$, c.f. [11]. We also implement the constraint equation (4.7) in a dynamic manner. That means that we don't solve (4.7) in every iteration, and we just solve it when the following conditions are met:

a. The level set curves do not change any more.

b. F does not decrease, which means that $F(\phi^{k+1}) \geq F(\phi^k)$.

The above criterion means that we don't need to solve the constraint equation at the beginning stage, and we just need it when the level set curve almost converges. Usually, we can also skip the solving of (4.7) for the first k_0 iterations, and then switch it on, where k_0 is a number which may depend on the problem. Integrating all these ingredients into Algorithm 2, the following algorithm is the one we have used for our numerical experiments:

Algorithm 3. Choose initial values for ϕ^0, β and $c_i^0,\ i = 1, 2, ..., n$. For $k = 1, 2, \ldots$, do

1. Update $\bar{c}^{k+1}$ by
$$c_i^{k+1} = c_i^k - \theta_i^k \frac{\partial F}{\partial c_i}(\bar{c}^k, \phi^k).$$

2. Compute $\phi^{k+1/2}$ by
$$\phi^{k+1/2} = \phi^k - \gamma_k \frac{\partial F}{\partial \phi}(\bar{c}^{k+1}, \phi^k),$$

3. Solve (4.7) to get ϕ^{k+1} if needed, or else set $\phi^{k+1} = \phi^{k+1/2}$.

4. Compute
$$\beta = TriggerRegu(\phi^k, \phi^{k+1}).$$

5. Check the convergence, if converged, stop; else goto 1.

Here, the step sizes $\bar{\theta}^k = \{\theta_i^k,\ i = 1, 2, ..., n\}$ as well as γ_k are determined by a line search method.

6 Numerical experiments

We take the examples in [8] to testify the efficiency of our Algorithm 3. Let $\Omega = (0,1) \times (0,1)$, $f = 20\pi^2 \sin(\pi x)\sin(\pi y)$. Let $u^\star$ be the exact finite element solution for the exact q, and σ be the noise level. We construct the observed solution by $u_d = u^\star + \sigma ||u^\star||_{L^2}/||R_d||_{L^2} R_d$, where R_d is a finite element function with nodal values being uniform random numbers between $[-1,1]$ with zero mean. Here we assume that the observation data is available at every node of Ω. When the noise parameter $\sigma > 5e-3$, then we apply the denoising technique of Chan and Tai [7] to smooth u_d and use the smoothed u_d as the observed solution. The domain Ω is divided into a rectangular mesh with uniform mesh size h for both x and y directions. In all our numerical experiments, we take $h = 1/64$. In the plots for the examples, we will always use solid lines to represent the numerical computed interface while use dash lines to represent the true interface. The interface for the computed solution at different iterations is obtained by rounding the value of function ϕ to the nearest integer value.

To alleviate the concern of inverse crime, the exact solution $u^\star$ is produced on a much finer mesh using the true q. For the tests given later, $u^\star$ is obtained on a mesh with size $h/4$. It would be more appropriate to get $u^\star$ by using a different simulator on a completely different mesh. In some of the experiments, we shall test on how much noise the algorithm can tolerate. For some of the tested examples, rather large amount of noise can be tolerated and we believe that approximation errors produced from a different simulator for producing the observation data shall not prevent our algorithm from working. In [24, p.451], tests haven been done for another model problem and it was shown that discontinuous coefficients can be recovered even the true discontinuity does not align with the mesh.

6.1 Example 1

We first consider a simple problem in this example. The exact coefficient $q(x)$ is given in Figure 1, i.e., $q(x) = 1$ inside a circle and $q(x) = 2$ outside the circle. We take $\sigma = 5\%$. The initial values for c_i and ϕ are taken as $\bar{c} = (0.6, 1.6)$ and $\phi = 1.5$. The level set curves for the computed solution at different iterations and the exact level set curve are shown in Figure 2. Note that the initial guess for the level set curve is rather poor. Algorithm 3 can recover the location of the discontinuities very well in less than 40 iterations. We also show the results with different noise levels in Figure 3. We see that q can be recovered well even with 40% noise. Note that we use $\beta_{min} = 10^{-6}$ for 40% noise, and $\beta_{min} = 10^{-9}$ for other cases.

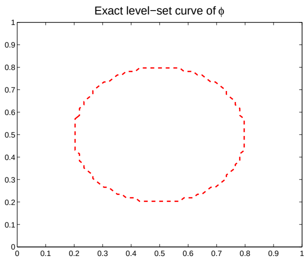
In Figure 4, we try to show a case where the true $q(x)$ is not piecewise constant, but a perturbation of a piecewise constant function. More precisely, we assume that the true coefficient is

$$q(x) + 0.4 * rand(q(x) - 0.5),$$

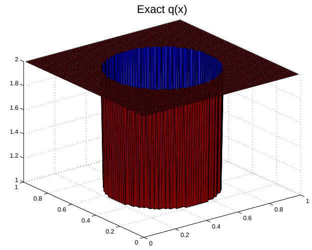
where $q(x)$ is as shown in Figure 1, and $rand(\cdot)$ produces the random numbers between [-1, 1]. For the observation, we take $\sigma = 5\%$. The level set curves for the computed solution at different iterations, and the exact level set curve are shown in Figure 7. We see that with poor initial guess, $q(x)$ can still be recovered quite accurately in about 90 iterations.

6.2 Example 2

The exact coefficient $q(x)$ is given in Figure 5, i.e., $q(x) = 2$ inside the two closed curves and $q(x) = 1$ outside the curves. The level set curves for the computed solution with different noise levels at different iterations are shown in Figure 8, Figure 9 and Figure 10. With different noise levels, Algorithm 3 needs about 300-400 iterations to recover $q(x)$. Compared to Example 1, more iterations are needed here due to the fact that this problem is more complicated. For this example, it seems that we can only tolerate up to 5% noise which is much less than in Example 1.

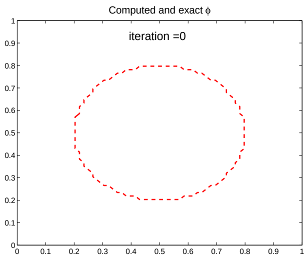


(a) exact level set curve

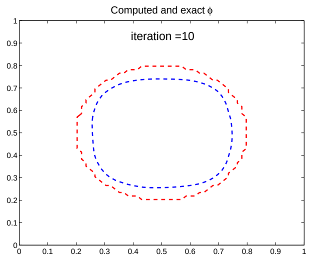


(b) exact q(x)

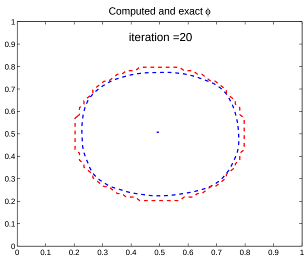
Figure 1: The exact $q(x)$ and the location of the discontinuity.



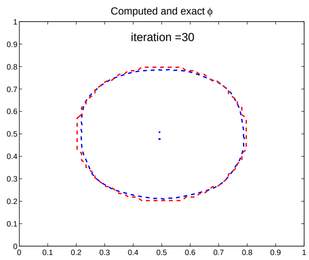
(a) Initial



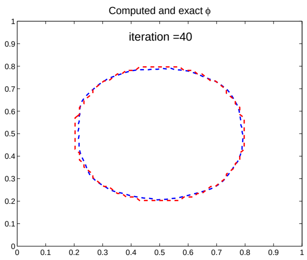
(b) 10 iterations



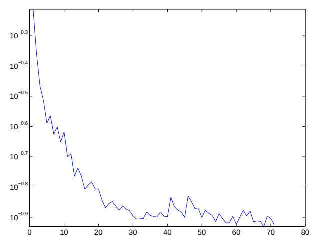
(c) 20 iterations



(d) 30 iterations



(e) 40 iterations

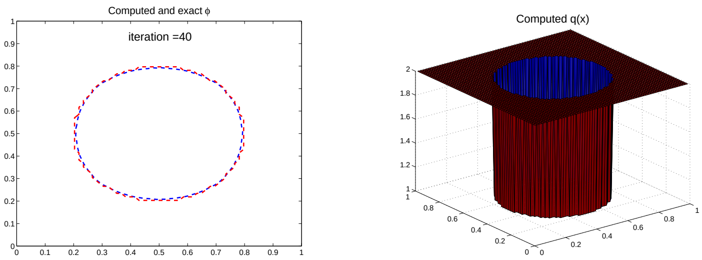


(f) The error $||q - q^k||_{L^2(\Omega)}$ v.s. iteration number

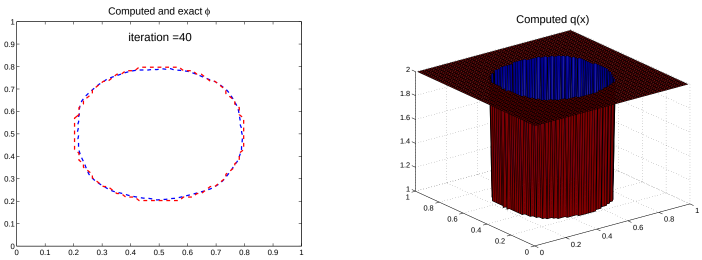
Figure 2: The computed solution at different iterations and the computational error for Example 1 with $\sigma = 5\%$, $\beta_{min} = 10^{-9}$, $\beta_{max} = 0.015$, $\alpha_2 = 0.2$, initial $\bar{c} = [0.6,\ 1.6]$ and initial level set function $\phi = 1.5$.

6.3 Example 3

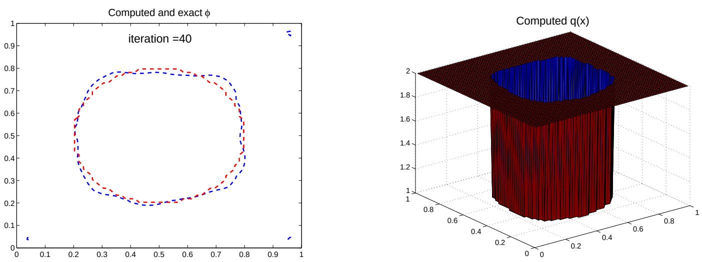
The exact coefficient $q(x)$ takes two different constant values inside the two curves, see Figure 6, i.e., $q(x) = 2$ inside one of the curves, $q(x) = 3$ inside the other curve and $q(x) = 1$ otherwise. So we have three distinct constant subdomains for $q(x)$. The level set curves for the computed solution with



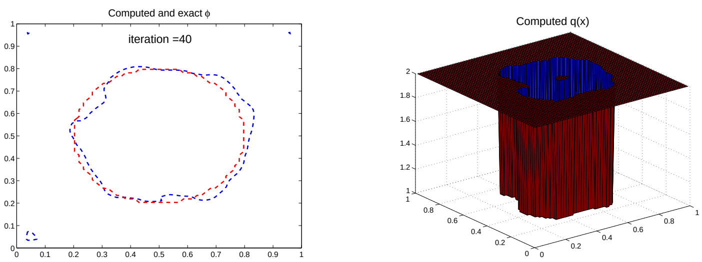
(a) With $\sigma = 0\%$



(b) With $\sigma = 5\%$



(c) With $\sigma = 20\%$



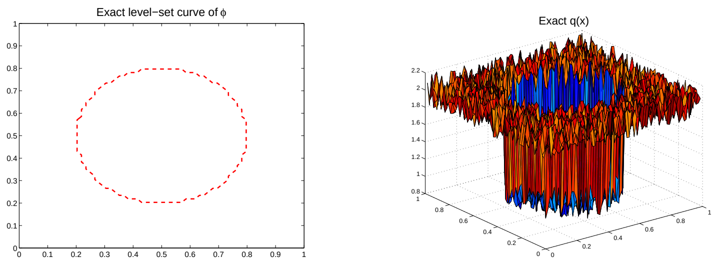
(d) With $\sigma = 40\%$

Figure 3: The computed level set curves and $q(x)$ with noise level $\sigma = 0\%,\ 5\%\ ,20\%,\ 40\%$. We have used β_{min} and β_{max} as before, except for $\sigma = 40\%$, we set $\beta_{min} = 10^{-6}$.

different noise levels at different iterations are shown in Figure 11 and Figure 12. For this example, it takes about 1000 iterations for the algorithm to get converge, and it seems that we can not tolerate noise that is more than 1%.

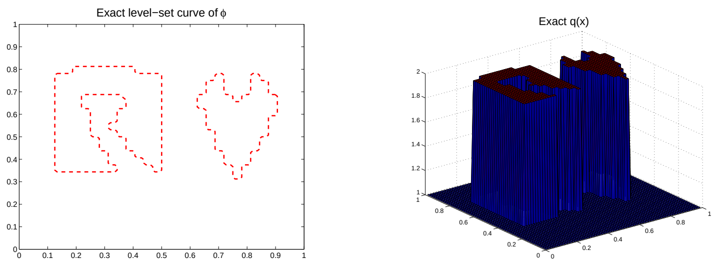
7 Conclusions

In this paper, the piecewise constant level set method has been used to formulate the elliptic inverse problem. We divide the energy functional into two parts, and we incorporate in them, both the con-



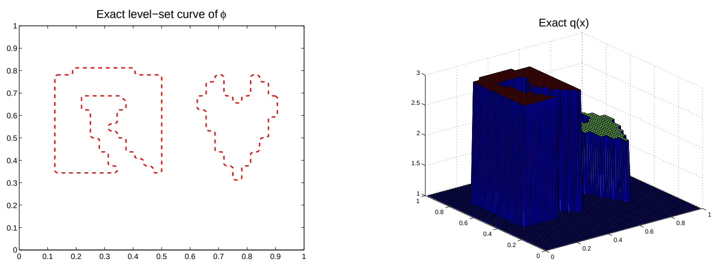
(a) exact level set curve　　(b) exact q(x)

Figure 4: The exact $q(x)$ and the location of the discontinuity.



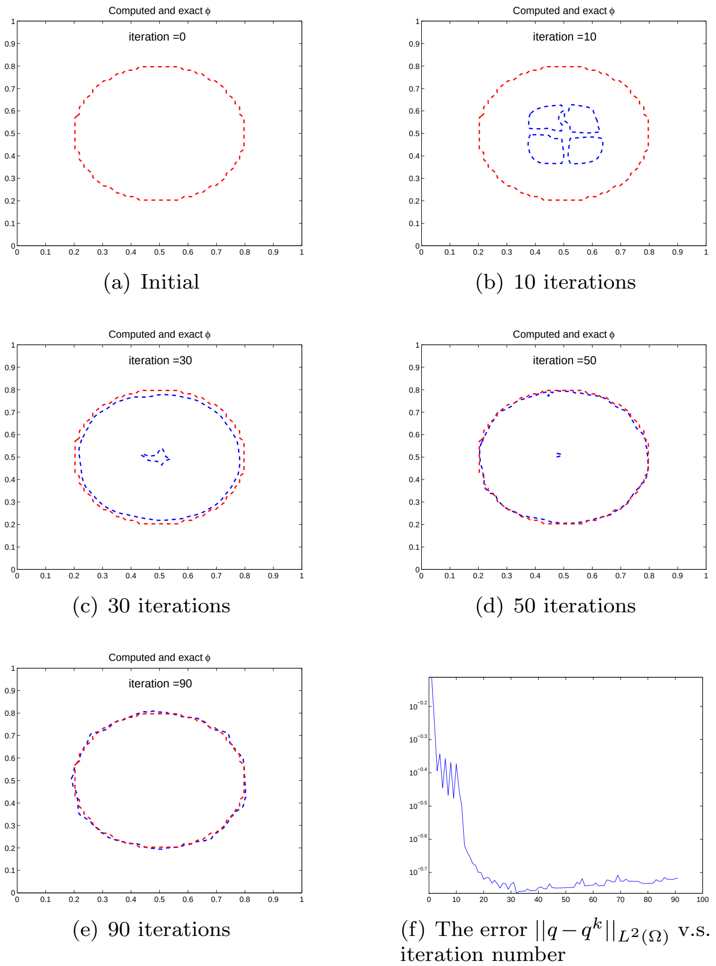
(a) exact level set curve　　(b) exact q(x)

Figure 5: The exact $q(x)$ and the location of the discontinuity .



(a) exact level set curve　　(b) exact q(x)

Figure 6: The exact $q(x)$ and the location of the discontinuity .



(a) Initial (b) 10 iterations

(c) 30 iterations (d) 50 iterations

(e) 90 iterations (f) The error $||q - q^k||_{L^2(\Omega)}$ v.s. iteration number

Figure 7: The computed solution at different iterations and the computational error for Example 1 with a perturbed $q(x)$. We have used $\sigma = 5\%$, $\beta_{min} = 10^{-9}$, $\beta_{max} = 0.015$, $\alpha_2 = 0.2$, initial $\bar{c} = [0.6,\ 1.6]$ and initial level set function $\phi = 1.5$.

straint and the total variation regularization in a dynamic manner. Numerical experiments show that our approach is very efficient and robust with respect to the geometry of the coefficient discontinuities and the initial guess of the level set function. Algorithm 3 can recover $q(x)$ quite accurately with moderate amount of noise in the observation data.

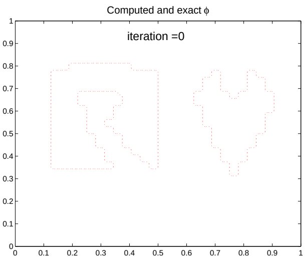
References

[1] U. Ascher and E. Haber. Grid refinement and scaling for distributed parameter estimation problems. *Inverse Problems*, 17:571–590, 2001.

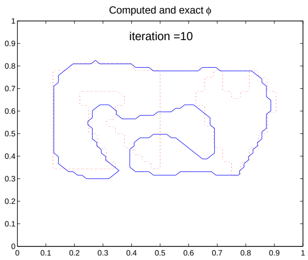
[2] U. Ascher and E. Haber. Computational methods for large distributed parameter estimation problems with possible discontinuities. *Symp. Inverse Problems, Design and Optimization*, 2004.

[3] U. Ascher, E. Haber, and H. Huang. On effective methods for implicit piecewise smooth surface recovery. Submitted 2004.

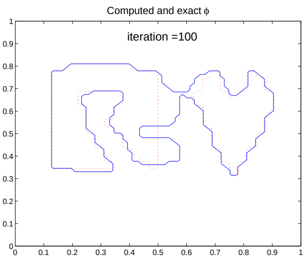
[4] M. Burger. A level set method for inverse problems. *Inverse problems*, 17:1327–1355, 2001.



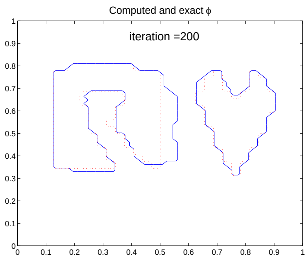
(a) Initial



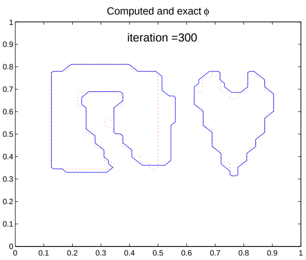
(b) 10 iterations



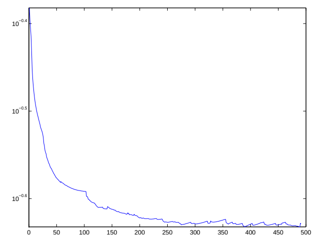
(c) 100 iterations



(d) 200 iterations



(e) 300 iterations



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

Figure 8: The computed solution at different iterations and the computational error for Example 2 with $\sigma = 0\%$. We have used $\beta_{min} = 10^{-6}$, $\beta_{max} = 0.015$, $\alpha_2 = 0.02$, initial $\bar{c} = [0.8,\ 1.8]$ and initial level set function $\phi = 1.5$.

[5] Martin Burger and Stanley J. Osher. A survey on level set methods for inverse problems and optimal design. *European J. Appl. Math.*, 16(2):263–301, 2005.

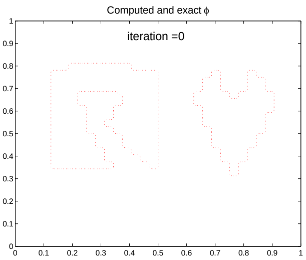
[6] T.F. Chan and X.-C. Tai. Augmented lagrangian and total variation methods for recovering discontinuous coefficients from elliptic equations. In M. Bristeau, G.Etgen, W. Fitzgibbon, J.L. Lions, J. Periaux, and M.F. Wheeler, editors, *Computational Science for the 21st Century*, pages 597–607, New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

[7] T.F. Chan and X.-C. Tai. Identification of discontinuous coefficients from elliptic problems using total variation regularization. *SIAM J. Sci. Comput.*, 25(3):881–904, 2003.

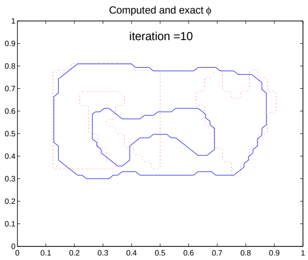
[8] T.F. Chan and X.-C. Tai. Level set and total variation regularization for elliptic inverse problems with discontinuous coefficients. *Journal of Computational Physics*, 193:40–66, 2003.

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

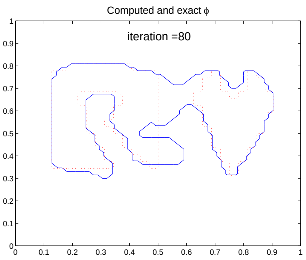
[10] Z. Chen and J. Zou. An augmented lagrangian method for identifying discontinuous parameters in elliptic systems. *SIAM J. Control Optim.*, 37(3):892–910, 1999(electronic).



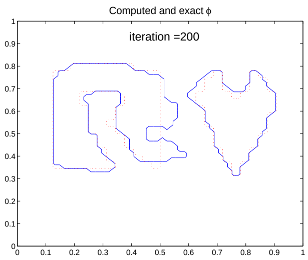
(a) Initial



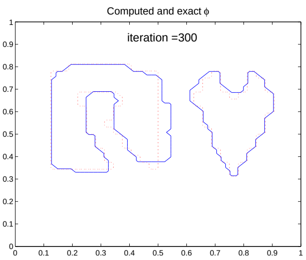
(b) 10 iterations



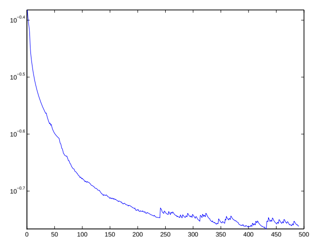
(c) 80 iterations



(d) 200 iterations



(e) 300 iterations



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

Figure 9: The computed solution at different iterations and the computational error for Example 2 with $\sigma = 1\%$. The values for the other parameters are $\beta_{min} = 10^{-6}$, $\beta_{max} = 0.015$, $\alpha_2 = 0.02$, initial $\bar{c} = [0.8,\ 1.8]$ and initial level set function $\phi = 1.5$.

[11] Oddvar Christiansen and Xue-Cheng Tai. Fast implementation of piecewise constant level set methods. Cam-report-06, UCLA, Applied Mathematics, 2005.

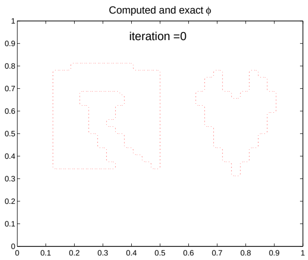
[12] Eric T. Chung, Tony F. Chan, and Xue-Cheng Tai. Electrical impedance tomography using level set representation and total variational regularization. *J. Comput. Phy.*, 205:357–372, 2005.

[13] Jason T. Chung and Luminita A. Vese. Energy minimization based segmentation and denoising using a multilayer level set approach. *Lecture Notes in Computer Science (Springer-Verlag)*, 3757:439–455, 2005.

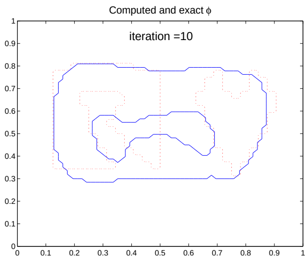
[14] O. Dorn, E. Miller, and C. Rappaport. A shape reconstruction method for electromagnetic tomography using adjoint fields and level sets. *Inverse Problems*, 16:1119–1156, 2000. Special issue on Electromagnetic Imaging and Inversion of the Earth's Subsurface.

[15] Selim Esedoglu and Yen-Hsi Richard Tsai. Threshold dynamics for the piecewise constant mumford-shah functional. Tech. Rep. CAM-report-04-63, UCLA Dep. Math, 2004.

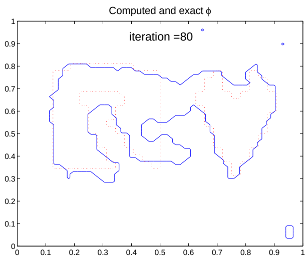
[16] Frédéric Gibou and Ronald Fedkiw. A fast hybrid k-means level set algorithm for segmentation. *Stanford Technical Report*, 2002.



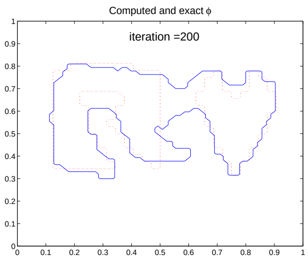
(a) Initial



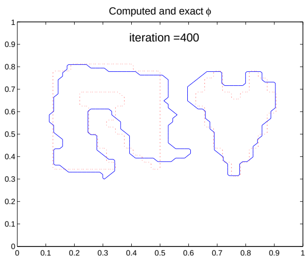
(b) 10 iterations



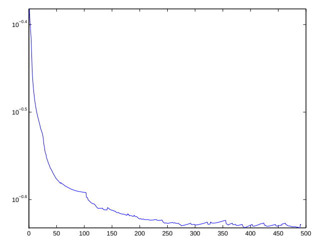
(c) 80 iterations



(d) 200 iterations



(e) 400 iterations



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

Figure 10: The computed solution at different iterations and the computational error for Example 2 with $\sigma = 5\%$, $\beta_{min} = 10^{-6}$, $\beta_{max} = 0.015$, $\alpha_2 = 0.02$, initial $\bar{c} = [0.8,\ 1.8]$ and initial level set function $\phi = 1.5$.

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

[18] J. Lie, M. Lysaker, and X.-C. Tai. Piecewise constant level set methods and image segmentation. In Ron Kimmel, Nir Sochen, and Joachim Weickert, editors, *Scale Space and PDE Methods in Computer Vision: 5th International Conference, Scale-Space 2005*, volume 3459, pages 573–584. Springer-Verlag, Heidelberg, April 2005.

[19] J. Lie, M. Lysaker, and X.-C. Tai. A binary level set model and some applications to image processing. *IEEE Trans. Image Process.*, to appear. Also as UCLA, Applied Math., CAM-report-04-31, 2004.

[20] J. Lie, M. Lysaker, and X.-C. Tai. A variant of the levelset method and applications to image segmentation. *Math. Comp.*, to appear. Also as UCLA, Applied Math. CAM-report-03-50, 2003.

[21] A. Litman. Reconstruction by level sets of n-ary scattering obstacles. *Inverse Problems*, 21(6):S131–S152, 2005.

[22] T. Lu, P. Neittaanmäki, and X.-C. Tai. A parallel splitting up method and its application to Navier-Stokes equations. *Appl. Math. Lett.*, 4(2):25–29, 1991.

[23] T. Lu, P. Neittaanmäki, and X.-C. Tai. A parallel splitting-up method for partial differential equations and its applications to Navier-Stokes equations. *RAIRO Modél. Math. Anal. Numér.*, 26(6):673–708, 1992.

[24] T. K. Nilssen and X. C. Tai. Parameter estimation with the augmented Lagrangian method for a parabolic equation. *J. Optim. Theory Appl.*, 124(2):435–453, 2005.

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

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

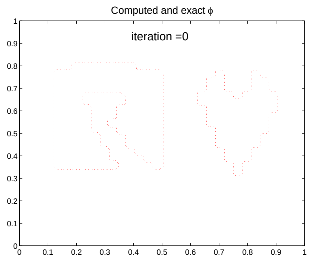
[27] J. Shen. Gamma-convergence approximation to piecewise constant Mumford-Shah segmentation. Tech. Rep. CAM-report-05-16, UCLA Dep. Math, 2005.

[28] B. Song and T. Chan. A fast algorithm for level set based optimization. *CAM-UCLA*, 68, 2002. Under revision for publication in SIAM J. Sci. Comput.

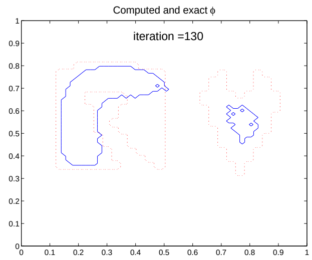
[29] X.-C. Tai and T. F. Chan. A survey on multiple level set methods with some applications for identifying piecewise constant functions. *International J. Numer. Anal. Modelling*, 1, 2004.

[30] X.-C. Tai and C. Yao. Fast piecewise constant level set method with newton updating. Technical report, UCLA, Applied Math. CAM-report-05-52, 2005.

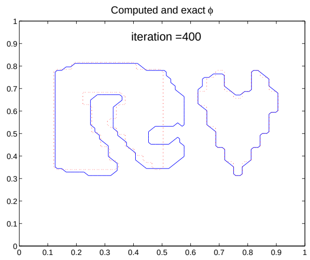
[31] Xue-Cheng Tai, Oddvar Christiansen, Ping Lin, and Inge Skjaelaaen. A remark on the mbo scheme and some piecewise constant level set methods. Technical report, UCLA, Applied Mathematics, CAM-report-05-24., 2005.



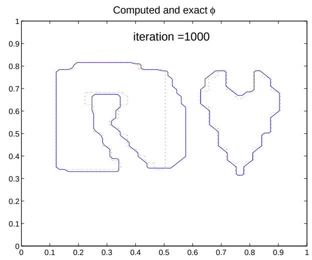
(a) Initial



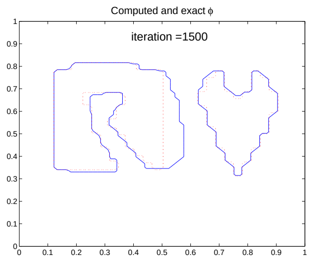
(b) 130 iterations



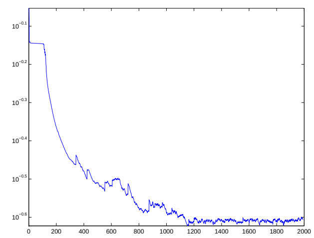
(c) 400 iterations



(d) 1000 iterations

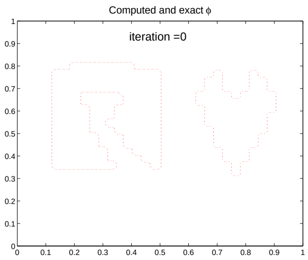


(e) 1500 iterations

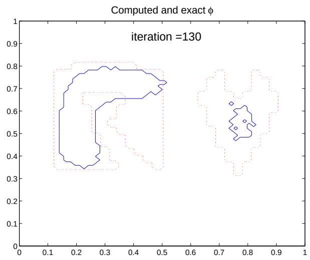


(f) The error $||q - q^k||_{L^2(\Omega)}$ v.s. iteration number

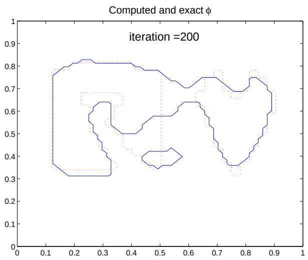
Figure 11: The computed solution at different iterations and the computational error for Example 3 with $\sigma = 0.1\%$, $\beta_{min} = 10^{-9}$, $\beta_{max} = 0.015$, $\alpha_2 = 0.02$, initial $\bar{c} = [0.8,\ 1.8,\ 2.8]$, and initial level set function $\phi = 1.0$.



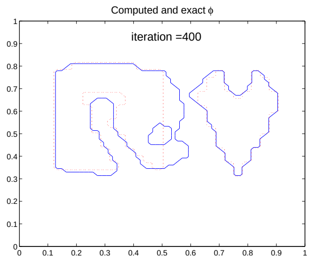
(a) Initial



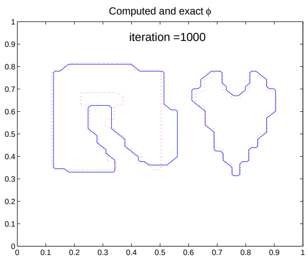
(b) 130 iterations



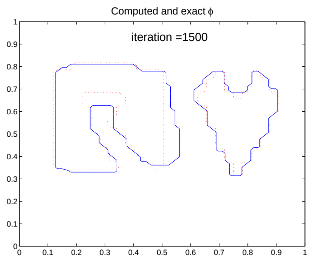
(c) 200 iterations



(d) 400 iterations



(e) 1000 iterations



(f) 1500 iterations

Figure 12: The computed solution at different iterations and the computational error for Example 3 with $\sigma = 1\%$, $\beta_{min} = 10^{-6}$, $\beta_{max} = 0.015$, $\alpha_2 = 0.02$, initial $\bar{c} = [0.8,\ 1.8,\ 2.8]$ and initial level set function $\phi = 1.0$.