A Binary Level Set Model for Elliptic Inverse Problems with Discontinuous Coefficients

Lars Kristian Nielsen*, Xue-Cheng Tai†, Sigurd Ivar Aanonsen‡ and Magne Espedal§.

Abstract

In this paper we propose a variant of a binary level set approach for solving elliptic problems with piecewise constant coefficients. The inverse problem is solved by a variational augmented Lagrangian approach with a total variation regularisation.

In the binary formulation, the seeked interfaces between the domains with different values of the coefficient are represented by discontinuities of the level set functions. The level set functions shall only take two discrete values, i.e. 1 and -1, but the minimisation functional is smooth.

Our formulation can, under moderate amount of noise in the observations, recover rather complicated geometries without requiring any initial curves of the geometries, only a reasonable guess of the constant levels is needed. Numerical results show that our implementation of this formulation has a faster convergence than the traditional level set formulation used on the same problems.

Keywords: Inverse problems, parameter identification, elliptic equation, augmented Lagrangian optimisation, level set methods, total variation regularisation.

1 Introduction

Consider the elliptic partial differential equation with Dirichlet boundary conditions:

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

In this paper we will use observations of the function u to recover the coefficient $q(\mathbf{x})$ by approximating it with a piecewise constant function. The function f is assumed known, and $\partial\Omega$ is the boundary of our domain Ω. This problem is a model problem for many real applications, for example, reservoir simulations [1], medical imaging [2, 3] and underground water investigations [4]. Even as a purely academic problem, this model problem has turned out to be rather difficult to solve numerically.

The problem of recovering the geometry of the coefficient discontinuities has motivated a number of approaches in the literature [5, 6, 7, 8]. A proper regularisation is often applied to control the jumps and the geometry of the discontinuities, see for example [5, 6]. Several approaches have also been used to represent the coefficient implicitly, especially a number of level set methods have been proposed for this purpose; see [9, 10, 11, 12, 13, 14, 15, 16].

For representing $q(\mathbf{x})$ we apply a piecewise constant formulation of the level set method. The original level set method was proposed by Osher and Sethian [17] for tracing interfaces between different phases of fluid flow. It has later been a versatile tool for representing and tracking interfaces separating a domain into subdomains. The method has been applied in a wide range

*Department of Mathematics, University of Bergen and CIPR-Centre for Integrated Petroleum Research, University of Bergen (larskn@mi.uib.no)

†Department of Mathematics, University of Bergen and CIPR-Centre for Integrated Petroleum Research, University of Bergen (tai@mi.uib.no)

‡CIPR-Centre for Integrated Petroleum Research, University of Bergen (Sigurd.Aanonsen@cipr.uib.no)

§Department of Mathematics, University of Bergen and CIPR-Centre for Integrated Petroleum Research, University of Bergen (resme@mi.uib.no)

of applications, i.e. reservoir simulations, inverse problems, image analysis, and optimal shape design problems. For recently surveys of level set methods see [18, 19, 20].

The representation of the interfaces in the standard level set formulation is done implicitly by the zero level set of one or several functions. The corresponding Euler-Lagrange equations give the evolution equations for the level set functions and the different constant values of the coefficient. In these methods the level set functions are forced to be signed distance functions, and are therefore the solution of Hamilton-Jacobi equations.

The *binary level set method*, which we apply in this paper, is classified as a piecewise constant level set approach. For this method, the level set functions are discontinuous functions at convergence, and should only take a fixed number of predefined constant values. Hence, they should not be distance functions as in the continuous formulation. In the binary method, we require the convergence values of the level set functions to be -1 or 1. When solving the optimisation problem with this imposed requirement, we need to minimise a smooth convex functional under a quadratic constraint. The level set method applied in this paper is similar to the method proposed in [21] used for image segmentation. This idea has also appeared in some earlier work [22, 23] for image segmentation. The binary level set idea is in fact very similar to the phase field model applied for many phase transition problems [24, 25, 26]. In this work, we use a novel way to treat the requirement that the level set functions should take the values ± 1. Another related work, utilising piecewise constant level set functions, is presented in [27]. In [27], just one level set function is required to identify an arbitrary number of phases.

In [10], Chan and Tai have performed a study on the elliptic inverse problem which is closely related to the work presented in this paper. They use continuous level set functions in a more standard level set formulation. As in their approach, we will in this work formulate the method in a variational setting, and apply an augmented Lagrangian approach for solving the minimisation problem. The Euler-Lagrange equations give the evolution equations for the level set functions.

Since the minimisation problem is highly ill-posed, we need to regularise the problem. The regularisation applied here is the total variation norm of the recovered coefficient. This will indirectly control both the length of the level set curves and the jumps in the coefficients, see [10, 28].

The contribution in this paper is the use of binary represented level set functions for solving the inverse problem. In our implementation, the relation between the coefficients and the level set functions is constructed such that it to a large extent can reduce the ill-posedness of the inverse problem, and at the same time be able to reconstruct rather complicated geometries.

Both the shape of the regions and the constant values in each region are recovered as part of the formulation. In comparison to most of the related methods using the level set approach, we do not need any initial guess of the contours. A reasonable initial guess of the constant values is though required.

The number of regions with different constant values of the coefficient is not required *a priori*, only an upper bound is needed. If the identified coefficient have n constant values, $\log_2 n$ level set functions are sufficient. If a higher number of functions is applied, the unnecessary functions will disappear (take a constant value) or the superfluous regions will merge with other regions.

This paper is structured as follows: The inverse problem is defined in Section 2, and in Section 3 the general framework of the binary level set method is given. In Section 4 we explain how this framework is utilised to solve the inverse problem. Further the augmented Lagrangian approach and the applied algorithm are given in Section 5. In Section 6 some remarks about the implementation issues are given, and in Section 7 we present the numerical results. Conclusions are given in Section 8.

2 The Inverse Problem

To recover the coefficients $q(\mathbf{x})$ we use observations $u_d \in L^2(\Omega)$ for the solution u of Eq. (1). Let Q be the set of admissible coefficients;

$$Q = \{q | q \in L^\infty(\Omega) \cap TV(\Omega), 0 < q_L(\mathbf{x}) \leq q(\mathbf{x}) \leq q_U(\mathbf{x}) < \infty\},$$

where $q_L(\mathbf{x})$ and $q_U(\mathbf{x})$ is the lower and upper bounds of q, respectively, known *a priori*.

We define $u(q)$ as the solution of Eq. (1) for a given function q. For the optimisation problem we construct the following functional to be minimised;

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

The first term is a measurement of the closeness of u_d and $u(q)$, and the second term is referred to as a regularisation term. The regularisation parameter $\beta > 0$ is a predefined constant weighting the regularisation, and $R(q)$ is a functional used to control the regularity of q. As in [10, 29] we take the total variation norm of q as the regularisation;

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

When q is not differentiable, $|\nabla q|$ is understood as a measure, see [30] p. 221. This regularisation will both control the lengths of the interfaces and the jumps of coefficient values. This differs from standard level set methods where it is common only to control the lengths of the interfaces.

The inverse problem to solve is to find the optimal coefficient q^* which is the solution of the following minimisation problem;

$$q^* = arg\min_{q \in Q} F(q). \tag{4}$$

Note that in Chan and Tai [10], Equation (1) was handled differently. In this work, we are trying to show the performance of the binary level set idea for this ill-posed inverse problem. The advantage of the binary level set method is that it removes the connection between the level set functions and the distance functions. In calculating the distance functions, lower order accuracy schemes may change the sign of the initial function. In addition, sharp corners may be smeared if the distance function is re-initialised too frequently. The second purpose to use the binary level set idea is to remove the Heaviside function and gain (local) convexity and smoothness for the minimisation problems.

3 Binary Level Set Approach

In this section we will present the binary level set formulation. We shall follow the mechanism proposed in [21] where it was applied for segmentation of digital images. The essential ideas for the binary level set method have appeared earlier in [22, 23]. It was in [21], a general framework was shown for the binary level set method and a systematic way was given to treat the constraints. In this work, we use the method to construct a piecewise constant function to approximate the coefficient $q(\mathbf{x})$ in Equation (1).

In standard level set methods, the partition of a domain Ω into a number of subregions $\{\Omega_j\}$ is defined by the sign of the continuous level set functions. For numerical reasons the level set functions are in these cases forced to be signed distance functions, where the *distance* is related to the boundary of the subdomains. In the binary method, we will instead use discontinuous level set functions which at convergence should take the values -1 or 1, inside and outside the subregions. The discontinuities of the functions will represent the boundary of the subdomains.

Let us first assume that Ω need to be divided into two subregions, Ω_1 and Ω_2, such that $\Omega = \bar{\Omega}_1 \cup \bar{\Omega}_2$ and $\Omega_1 \cap \Omega_2 = \emptyset$. A representation of this domain can be given by

$$\phi(\mathbf{x}) = \begin{cases} 1 & \forall \mathbf{x} \in \Omega_1 \\ -1 & \forall \mathbf{x} \in \Omega_2, \end{cases} \tag{5}$$

and the curve separating Ω_1 and Ω_2 is implicitly given as the discontinuity of ϕ, see Figure 1. The properties of ϕ can be used to construct a scalar function $q(\mathbf{x})$ with distinct constant values inside these two different subdomains. If we assume that the value of $q(\mathbf{x})$ is equal to c_1 in Ω_1 and equal to c_2 in Ω_2, then q can be written as

$$q = \frac{1}{2}\left[\, c_1(\phi + 1) - c_2(\phi - 1) \,\right]. \tag{6}$$

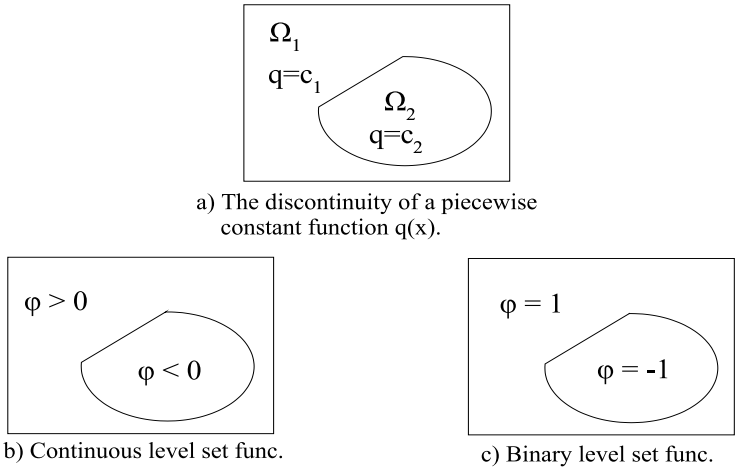


Figure 1: Level set representations of a piecewise constant function $q(\mathbf{x})$. In this example q has two regions with different constant values, c_1 and c_2. By continuous level set functions the discontinuity of q can be represented as in Figure b), and by binary level set functions ϕ is forced to take the values -1 and 1, as in Figure c).

As in the continuous level set formulation, multiple level set functions can be used to represent more than two regions. Following the terminology applied in [21], a function having four constant regions can be represented by two level set functions, and expressed as

$$q = \frac{1}{4}\left[\, c_1(\phi_1+1)(\phi_2+1) - c_2(\phi_1+1)(\phi_2-1) - c_3(\phi_1-1)(\phi_2+1) + c_4(\phi_1-1)(\phi_2-1)\,\right]. \quad (7)$$

To construct the basis functions for a general number of level set functions, N, we introduce the following notation: For $j \in \{1, 2, \ldots, 2^N\}$ let $(b_1^{j-1}, b_2^{j-1}, \ldots, b_N^{j-1})$ be the binary representation of $j-1$, that is b_i^{j-1}=-1 or 1. Further we define $s(j) = \sum_{i=1}^{N} b_i^{j-1}$, and ψ_j as the product

$$\psi_j = \frac{1}{N}(-1)^{s(j)} \prod_{i=1}^{N} (\phi_i + 1 - 2b_i^{j-1}). \quad (8)$$

For a given $\mathbf{c} = (c_1, c_2, \ldots, c_{2^N})$ and level set functions $\boldsymbol{\phi} = \{\phi_i\}_{i=1}^{N}$, the following formula gives a general representation for piecewise constant functions;

$$q(\boldsymbol{\phi}, \mathbf{c}) = \sum_{j=1}^{2^N} c_j \psi_j(\boldsymbol{\phi}). \quad (9)$$

Eq. (6) and Eq. (7) are special cases of this formula, but with each ψ_j expressed in terms of $\boldsymbol{\phi}$. In the first case, we have $\psi_1 = \frac{1}{2}(\phi+1)$ and $\psi_2 = -\frac{1}{2}(\phi-1)$ in Eq. (6). With two level set functions, we get $\psi_1 = \frac{1}{4}(\phi_1+1)(\phi_2+1)$, $\psi_2 = -\frac{1}{4}(\phi_1+1)(\phi_2-1), \ldots$ in Eq. (7).

In the following, we let $K(x) = x^2 - 1$. The level set functions are required to satisfy

$$K(\phi_i) = \phi_i^2 - 1 = 0 \quad \forall\, i. \quad (10)$$

This requirement will force the level set functions to take the values -1 or 1 at convergence. With this constraint fulfilled, the basis functions will be characteristic functions for the corresponding subdomains, i.e. $\psi_j = 1$ in Ω_j and zero elsewhere. That is, the support of the different basis functions are non-overlapping, supp $\psi_i \cap$ supp $\psi_j = \emptyset \;\; \forall\, i \neq j$, and the total support of all the basisfunctions covers the complete domain, i.e. $\Omega = \cup_{j=1}^{2^N}$supp ψ_j.

4 The Binary Level Set Method for the Inverse Problem

From the last section, we see that every piecewise constant function can be represented as in (9) under the requirement that the level set functions satisfy (10). In order to find a piecewise constant function, we just need to find the corresponding $\mathbf{c}$ values and the level set functions ϕ_i. If we define the vector $\mathbf{K}(\boldsymbol{\phi}) = \{K(\phi_i)\}_{i=1}^N$, we can thus reformulate problem (4) as

$$(\boldsymbol{\phi}^*, \mathbf{c}^*) = \arg\,\{\min_{\boldsymbol{\phi},\mathbf{c}} F(q(\boldsymbol{\phi},\mathbf{c})) \quad \text{subject to} \quad \mathbf{K}(\boldsymbol{\phi}) = \mathbf{0}\}, \tag{11}$$

where the optimal coefficient can be calculated by $q^* = q(\boldsymbol{\phi}^*, \mathbf{c}^*)$. The constraint $\mathbf{K} = \mathbf{0}$ is applied to control the structure of the level set functions, and will therefore depend on the choice of basis functions.

Define $\tilde{F}(\boldsymbol{\phi}, \mathbf{c}) = F(q(\boldsymbol{\phi}, \mathbf{c}))$. To evolve the level set functions and update the constant values such that $q(\mathbf{x})$ will converge to the optimal solution, we need to calculate the derivatives of $\tilde{F}$ with respect to $\boldsymbol{\phi}$ and $\mathbf{c}$. By the chain rule we have, c.f. [10],

$$\frac{\partial \tilde{F}}{\partial \phi_i} = \frac{\partial F}{\partial q}\frac{\partial q}{\partial \phi_i} \quad \forall\, i = 1, 2, \ldots, N \tag{12}$$

and

$$\frac{\partial \tilde{F}}{\partial c_j} = \int_\Omega \frac{\partial F}{\partial q}\frac{\partial q}{\partial c_j}\, d\mathbf{x} \quad \forall\, j = 1, 2, \ldots, 2^N. \tag{13}$$

For a wide number of problems $\frac{\partial F}{\partial q}$ is known, and we only need to compute $\frac{\partial q}{\partial \phi_i}$ and $\frac{\partial q}{\partial c_j}$. In this work, $\frac{\partial F}{\partial q}$ is calculated by the adjoint method;

$$\frac{\partial F}{\partial q} = -\nabla u \cdot \nabla z - \beta \nabla \cdot \left(\frac{\nabla q}{|\nabla q|}\right), \tag{14}$$

where $z \in H_0^1(\Omega)$ is the solution of

$$\begin{aligned} -\nabla \cdot (q(\mathbf{x})\nabla z) &= u - u_d \quad \text{in } \Omega \subset \mathbb{R}^2 \\ z &= 0 \quad \text{on } \partial\Omega. \end{aligned} \tag{15}$$

To compute $\frac{\partial \tilde{F}}{\partial q}$ once, we need to solve both (1) and (15) once.

5 Augmented Lagrangian Formulation

We apply an augmented Lagrangian method to solve problem (11). The Lagrangian functional involves both $\tilde{F}$ and the constraint K;

$$L(\boldsymbol{\phi}, \mathbf{c}, \boldsymbol{\lambda}) = \tilde{F}(\boldsymbol{\phi}, \mathbf{c}) + \textstyle\sum_{i=1}^N \int_\Omega \lambda_i K(\phi_i)\, dx + \mu \sum_{i=1}^N \int_\Omega |K(\phi_i)|^2\, dx. \tag{16}$$

Here $\mu > 0$ is a penalisation parameter which usually is a fixed parameter chosen *a priori*, or it can in some cases be increased carefully through the iterations to improve the convergence. $\boldsymbol{\lambda} = \{\lambda_i\}_{i=1}^N$ is the Lagrangian multipliers where λ_i is a function defined in the same domain as ϕ_i.

We search a saddle point of L and therefore require

$$\frac{\partial L}{\partial \phi_i} = 0, \quad \frac{\partial L}{\partial \lambda_i} = 0 \quad \forall\, i \in \{1, \ldots, N\} \quad \text{and} \quad \frac{\partial L}{\partial c_j} = 0 \quad \forall\, j \in \{1, \ldots, 2^N\}. \tag{17}$$

From the definition of L we have that

$$\frac{\partial L}{\partial \phi_i} = \frac{\partial \tilde{F}}{\partial \phi_i} + \lambda_i K(\phi_i) + 2\mu K'(\phi_i), \qquad \frac{\partial L}{\partial \lambda_i} = K(\phi_i) \quad \text{and} \quad \frac{\partial L}{\partial c_j} = \frac{\partial \tilde{F}}{\partial c_j}. \tag{18}$$

To find a saddle point of L, we apply an iterative algorithm. Starting with initial guesses $\boldsymbol{\phi}^0$, $\mathbf{c}^0$ and $\boldsymbol{\lambda}^0$, we iterate towards the better approximations denoted by $\boldsymbol{\phi}^k$, $\mathbf{c}^k$ and $\boldsymbol{\lambda}^k$ where $k = \{1, 2, \dots\}$. When the change of these variables approach zero, the iterations can be stopped.

The minimisation with respect to $\boldsymbol{\phi}$ is done by introducing an artificial time variable t, and then solve the PDE

$$\frac{\partial \boldsymbol{\phi}}{\partial t} = -\frac{\partial L}{\partial \boldsymbol{\phi}}. \tag{19}$$

When this reaches a steady solution we have $\frac{\partial \boldsymbol{\phi}}{\partial t} = 0$ which implies $\frac{\partial L}{\partial \phi_i} = 0 \ \ \forall\, i$. Numerically we discretise Eq. (19) by a forward Euler scheme and get the following updating scheme for $\boldsymbol{\phi}$;

$$\boldsymbol{\phi}^{k+1} = \boldsymbol{\phi}^k - \Delta t_\phi \, \frac{\partial L}{\partial \boldsymbol{\phi}}(\boldsymbol{\phi}^k, \mathbf{c}^k, \boldsymbol{\lambda}^k), \tag{20}$$

where $\Delta t_\phi > 0$ is a timestep. The timestep can be chosen as a small fixed number, or a line search can be applied to find an optimal steplength for each iteration.

A similar approach is applied to update $\mathbf{c}$, but in difference from the updating of $\boldsymbol{\phi}$, we introduce separate artificial time variables for each of the constants c_j. This results in the following updating scheme for each constant;

$$c_j^{k+1} = c_j^k - \Delta t_{c_j} \frac{\partial L}{\partial c_j}(\boldsymbol{\phi}^{k+1}, \mathbf{c}^k, \boldsymbol{\lambda}^k), \tag{21}$$

where the most recent values of $\boldsymbol{\phi}$ are used in the calculation of the gradients. The minimisation approach described here, for updating $\boldsymbol{\phi}$ and $\mathbf{c}$, is known as the *steepest decent method*. In the calculations, we also assume that the constants c_j have lower and upper bounds a_j and b_j which are known *a priori*, i.e. $c_j \in [a_j, b_j]$.

To update the Lagrangian multipliers we follow the approach in [31, 21], i.e.

$$\boldsymbol{\lambda}^{k+1} = \boldsymbol{\lambda}^k + \mu \mathbf{K}(\boldsymbol{\phi}^{k+1}). \tag{22}$$

We incorporate the updating in Eq. (20), (21) and (22) into the following algorithm where line searches are included:

Algorithm A (Uzawas Algorithm for Variational Level Set Methods)

Determine how many level set functions, N, to use.

Initialise: $\boldsymbol{\phi}^0$, $\mathbf{c}^o$ and $\boldsymbol{\lambda}^0$ and set $k = 0$.

1. Update $\boldsymbol{\phi}$;
   (a) Compute q, u and z by Eq. (9), (1) and (15), respectively.
   (b) Define: $\boldsymbol{\alpha}_\phi^k = \frac{\partial L}{\partial \boldsymbol{\phi}}(\boldsymbol{\phi}^k, \mathbf{c}^k, \boldsymbol{\lambda}^k)$.
   (c) Find the optimal time step: $\Delta t_\phi = arg \min_{\Delta t} L\left(\boldsymbol{\phi}^k - \Delta t \, \boldsymbol{\alpha}_\phi^k, \mathbf{c}^k, \boldsymbol{\lambda}^k\right)$.
   (d) Evolve the level set functions: $\boldsymbol{\phi}^{k+1} = \boldsymbol{\phi}^k - \Delta t_\phi \boldsymbol{\alpha}_\phi^k$.
2. Update $\mathbf{c}$ (after a fixed number of iterations);
   For each c_j, $j = 1, 2, \dots, 2^N$:
   (a) Compute q, u and z by Eq. (9), (1) and (15), respectively.
   (b) Define: $\alpha_{c_j}^k = \frac{\partial L}{\partial c_j}(\boldsymbol{\phi}^{k+1}, \mathbf{c}^k, \boldsymbol{\lambda}^k)$.
   (c) Define the search interval: Let $M \in \mathbb{R}$ be all values of Δt such that $c_j^k - \Delta t \, \alpha_{c_j}^k \in [a_j, b_j]$.
   (d) Find the optimal timestep: $\Delta t_{c_j} = arg \min_{\Delta t \in M} L\left(\boldsymbol{\phi}^{k+1}, \mathbf{c}^k - \Delta t \, \alpha_{c_j}^k \mathbf{e}_j, \boldsymbol{\lambda}^k\right)$, where $\mathbf{e}_j$ is the j'th unit vector.
   (e) Update this constant: $c_j^{k+1} = c_j^k - \Delta t_{c_j} \alpha_{c_j}^k$.

3. Update $\boldsymbol{\lambda}$ (after a fixed number of iterations);

$$\boldsymbol{\lambda}^k = \boldsymbol{\lambda}^k + \mu \mathbf{K}(\boldsymbol{\phi}^{k+1}).$$

4. Iterate again if necessary;

$$k = k + 1.$$

Notice that q is updated implicitly using the most recently calculated values of $\boldsymbol{\phi}$ and $\mathbf{c}$. In this algorithm we do not use step 2 and 3 in every iteration. This is because the algorithm becomes unstable if $\mathbf{c}$ and $\boldsymbol{\lambda}$ are updated too often. In principle we could have run step 1 to convergence before doing the other steps. Numerically this is not strictly necessary and it would have been computationally heavy. We have therefore updated $\mathbf{c}$ and $\boldsymbol{\lambda}$ after a fixed numbers of iterations.

6 Implementation Issues

As is typical for Augmented Lagrangian algorithms, the convergence is fast in the beginning and it slows down when the solution is getting closer to the true minimiser. A natural solution to this is to apply larger timesteps when evolving $\boldsymbol{\phi}$, but this will make the algorithm unstable. Another problem, when solving inverse problems, is that the sensitivities related to changes in u with respect to q may be very small in some regions. This will further slow down the speed of convergence.

To speed up the algorithm we have introduced a modification of the level set function when computing $q = q(\boldsymbol{\phi}, \mathbf{c})$. Instead of applying the level set functions $\boldsymbol{\phi}$ directly as in (6) and (7), we replace each ϕ_i by the function

$$\tilde{\phi}_i = \operatorname{sgn}(\phi_i) = \begin{cases} \frac{\phi_i}{|\phi_i|} & \text{for} \quad \phi_i \neq 0, \\ 0 & \text{else.} \end{cases} \tag{23}$$

For notational convenience we define the vector $\tilde{\boldsymbol{\phi}} = \{\tilde{\phi}_i\}_{i=1}^N$. Inserting this into Eq. (6) will give the following construction of q:

$$q = \begin{cases} \frac{1}{2}\left[c_1\left(\frac{\phi}{|\phi|}+1\right) - c_2\left(\frac{\phi}{|\phi|}-1\right)\right] & \text{for} \quad \phi_i \neq 0, \\ \frac{1}{2}(c_1 + c_2) & \text{else.} \end{cases} \tag{24}$$

Eq. (7) can be modified in exactly the same way, and for general cases we replace ϕ_i by $\tilde{\phi}_i$ when calculating ψ_j in (8). By the chain rule we have

$$\frac{\partial q}{\partial \phi_i} = \frac{\partial q}{\partial \tilde{\phi}_i}\frac{\partial \tilde{\phi}_i}{\partial \phi_i} = \frac{\partial q}{\partial \tilde{\phi}_i}\,\delta(\phi_i), \tag{25}$$

where δ denotes the delta Dirac function, i.e. $\delta(0) = 1$ and $\delta(\phi_i) = 0 \ \forall\, \phi_i \neq 0$.

In numerical implementations, it is desirable to replace $\tilde{\phi}_i$ by a smoothed approximation. The chosen approximation is

$$\tilde{\phi}_i \approx \frac{\phi_i}{\sqrt{\phi_i^2 + \epsilon}}, \tag{26}$$

where ϵ is a small positive number which has to be chosen. As ϕ_i is replaced by $\tilde{\phi}_i$, the gradient calculation in (12) and (13) also needs to be changed using (25). However, we have observed from our numerical experiments that good results are obtained if we just replace $\delta(\phi_i)$ in (25) by 1. Thus, the codes used for calculating the gradients do not need to have any change. The only change we need to have for the codes, is to replace ϕ_i by $\tilde{\phi}_i$ when calculating ψ_j.

As described earlier, there will in inverse problems typically be some low sensitive areas where $\frac{\partial u}{\partial q}$ is very small. When updating $\boldsymbol{\phi}$, the contribution from $\frac{\partial F}{\partial q}$ will be small in these regions. This produces difficulties in finding the minimum of $\tilde{F}$, and in our approach the constraint will often be too dominating in the low sensitive regions compared to the other regions. In order to

overcome this difficulty, we have to weight the constraint very low, which in turn will give a slow convergence.

If we apply a small value of ϵ in (26), a small change in ϕ_i will give large changes in q when ϕ_i is close to zero. This will reduce the problem of a too dominating constraint in the low sensitive regions, but only as long as $|\phi_i|$ has a relative low value. Another and may be more important fact, is that q will converge faster in the regions which have larger sensitivity. It seems like this will make the calculated derivatives more reliable in the rest of the domain. The optimal shape of the level set functions will then easier be found, even with a higher weight of the constraint. This will in turn make the method faster.

By numerical experiments we have found it desirable to start with a rather large value of ϵ, and then decrease ϵ during the iterations. In this setting it is also natural to increase μ during the iterations. This differs from other related works [21, 27, 10], where a fixed μ has been used to reduce the ill-conditioness of the problem.

The minimisation with respect to $\mathbf{c}$ is a highly ill-conditioned process. It should therefore not be done too early or too frequently during the iterations. To further stabilise this process, we have applied a predefined search interval $[a_j, b_j]$ for each constant such that there will be no risk of producing values completely out of range.

When updating both $\boldsymbol{\phi}$ and $\mathbf{c}$, we have to be aware that these quantities are dependent of each other. In this work $\boldsymbol{\phi}$ and $\mathbf{c}$ are updated in an alternating way (see Alg. A). Usually we start to update $\mathbf{c}$ before $|\boldsymbol{\phi}|$ has the unit value for all points in the domain. A situation which then may occur, is that a combination of $|\phi_i| \neq 1$ (for some i) and c_j-values not corresponding to the constant values of the exact q, can in some cases still produce the correct value of q. Equilibrium at this point should theoretically be avoided by the constraint $\mathbf{K}(\boldsymbol{\phi}) = 0$, but numerically it may cause trouble. In the suggested implementation, where $\boldsymbol{\phi}$ is replaced by $\tilde{\boldsymbol{\phi}}$, this problem can be considerably reduced by applying a slightly lower ϵ-value in (26) when updating $\mathbf{c}$ (Step 2 of Alg. 1) than the ϵ-value used when updating $\boldsymbol{\phi}$ (Step 1 of Alg. 1). An equilibrium where q takes the correct value and $|\tilde{\phi}_i|$ is not close to 1 will then be impossible.

7 Numerical Results

We will test the proposed algorithm on several two dimensional problems. Our domain $\Omega = (0,1)\times(0,1)$ is divided into a rectangular mesh with uniform mesh size $h = 1/64$ in both directions. The functions f, q, $\boldsymbol{\phi}$ and the required derivatives are approximated by piecewise constants over this mesh. The force function $f(x_1, x_2) = 20\pi^2 \sin(\pi x_1)\cos(\pi x_2)$.

For a given function $q(\mathbf{x})$, approximations of u and z can be found by solving correspondingly Eq. (1) and Eq. (15) by a finite element or a finite difference method. In our case we have chosen to apply some already developed software solving this by a finite element method.

To construct the synthetic data we add random noise to the measurements corresponding to the true coefficient. Let u_{ex} be the numerical solution (with no noise added) for the true $q(\mathbf{x})$, and let σ be the noise level. We get $\tilde{u}_d$ by calculating $\tilde{u}_d = u_{\text{ex}} + \sigma R_d ||u_{\text{ex}}||_{L^2(\Omega)}/||R_d||_{L^2(\Omega)}$, where R_d is a finite element function with nodal values being uniform random numbers between -1 and 1, and with zero mean. Thereafter we apply the total variation denoising technique of Chan and Tai [6] to smooth $\tilde{u}_d$. The smoothed version of $\tilde{u}_d$ is used as observations u_d.

We start with the initial $\boldsymbol{\phi}$ equal to zero, that is equally far from the two searched values, -1 and 1. This initialisation means that we do not assume anything about the shape of $\boldsymbol{\phi}$ *a priori*. In all the examples the constant values of $q(\mathbf{x})$ are assumed unknown, but we need to specify an interval which defines the lower and upper bounds for the constants. The initial c_j-values are chosen equal to the lower bounds of the corresponding intervals.

The penalty parameter μ is increased slowly by a fixed factor through the iterations. For the cases with one level set function this factor can be higher than for cases involving more level set functions. Letting μ_k be the value of μ for iteration k, we have used $\mu_k{=}\mu_0 \cdot 1.01^k$ for one level set function, and $\mu_k{=}\mu_0 \cdot 1.002^k$ for two level set functions. We keep μ_k fixed when it reaches an upper bound. For the different examples we choose a suitable initial value μ_0.

The Lagrangian multiplier $\boldsymbol{\lambda}$ is initially equal to zero and kept at this value in the beginning of the optimisation. After a fixed number of iterations we start to update $\boldsymbol{\lambda}$ each 10^{th} iteration.

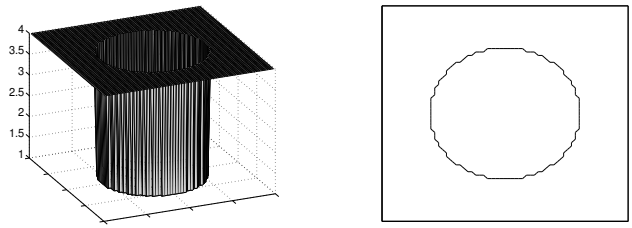
The frequency for updating the constant values, $\mathbf{c}$, will influence the speed of convergence. A seldom updating of $\mathbf{c}$ will give a slow convergence, while a too frequent updating may cause unstable behaviour. We have chosen to update $\mathbf{c}$ each 5^{th} iteration in the examples involving a circle shaped discontinuity, and each 20^{th} iteration in the examples involving more complicated geometries. In the last example with three different regions and with noise equal to or larger than 1% we observed unstable behaviour when applying an updating frequency of 20. In these cases we therefore increased the interval between each updating to 50, which stabilised the method.

In the cases where we substitute ϕ_i by $\tilde{\phi}_i$ from Equation (26), we have chosen 0.1 as the initial ϵ-value, and decreased it by a factor of 0.98 in each iteration until the value reaches a lower bound equal to 10^{-7}.

When plotting the level set functions we have in some of the examples plotted both the curve indicating the sign change of ϕ and the discontinuity curve of the true $q(\mathbf{x})$. We have used a dotted line for the discontinuity of the true $q(\mathbf{x})$ and a solid curve to indicate the sign of each ϕ_i. As before the index k will in the figures denote the estimate at iteration k, and k will be given on the x-axis on the convergence plots. No superscript on q and c_i will indicate the corresponding true values.

Example 1: Convergence, regularisation and noise
In this example, we will look into a simple case in order to study some of the basic properties related to the solution strategy. The true function to recover, $q(\mathbf{x})$, has two subregions with different constant values. In Figure 2a) the true $q(\mathbf{x})$ is plotted, and in Figure 2b) the corresponding curve of discontinuity is shown.

We will first test the original binary formulation explained in Section 3, and compare this to the suggested implementation proposed in Section 6. In the further text we will refer to the ordinary and the novel implementation as the *Ordinary Binary Level Set Method (OBLSM)* and the *Signed Binary Level Set Method (SBLSM)*, respectively. The example will be tested with 5% noise added to the observation data, and the lower and upper bounds for the constants are 0.5 below and above the true values.



a) True $q(\mathbf{x})$. b) True discontinuity curve.

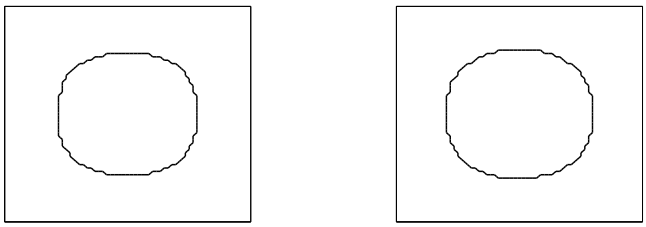
Figure 2: True $q(\mathbf{x})$ and the corresponding discontinuity for example 1.

Independent of the method of choice, the coefficient in this example is relatively easy to recover with a rather good accuracy. The choice of the control parameters β and μ_0 will though slightly influent the results and the error in the solution. An optimal choice of the parameters for one of the methods is not necessarily the optimal choice for the other method. Because of this, we will compare the convergence of the two methods with three different sets of parameters.

The applied parameters for the three tests are given in Table 1. In Test 1, the set of parameters is equal for the two methods, while in Test 2 and 3, β and μ_0 are tuned such that the error of q is approximately equal for the two different approaches. In Figure 3, the discontinuity of the converged level set functions are plotted for both OBLSM and SBLSM. The discontinuity curve is nearly the same for both methods. This is also the case for the two latter tests, and we therefore omit the corresponding plots for these tests. The convergence can though be different for the two methods. Comparisons of the convergence are shown in Figure 4, 5 and 6.

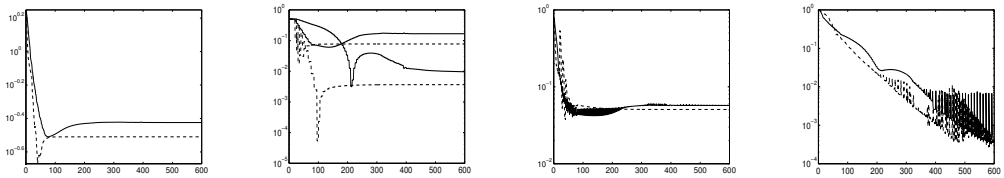
	Ex. 1, Test 1.	Ex. 1, Test 2.	Ex. 1, Test 3.
OBLSM	$\beta = 1 \cdot 10^{-4}$ $\mu_0 = 0.05$	$\beta = 1 \cdot 10^{-3}$ $\mu_0 = 0.025$	$\beta = 1 \cdot 10^{-4}$ $\mu_0 = 0.025$
SBLSM $\phi_i \to \tilde{\phi}_i$	$\beta = 1 \cdot 10^{-4}$ $\mu_0 = 0.05$	$\beta = 1 \cdot 10^{-4}$ $\mu_0 = 0.025$	$\beta = 1 \cdot 10^{-4}$ $\mu_0 = 0.05$

Table 1: Parameters for Example 1 for the Ordinary Binary Level Set Method (OBLSM) and the Signed Binary Level Set Method (SBLSM).



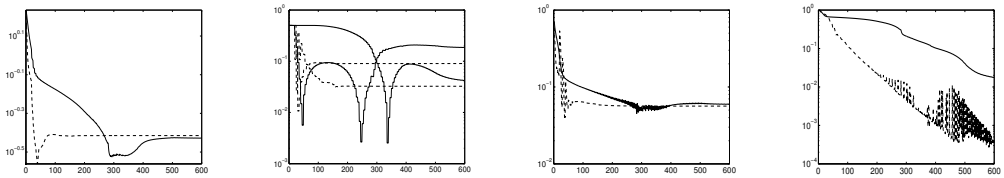
a) Final discontinuity OBLSM. b) Final discontinuity SBLSM.

Figure 3: Example 1, Test 1: The discontinuity curve of the level set function for the two methods at convergence. The final results from the two different approaches are very similar. The parameters are $\beta = 10^{-4}$ and $\mu_0 = 0.05$ for both methods.



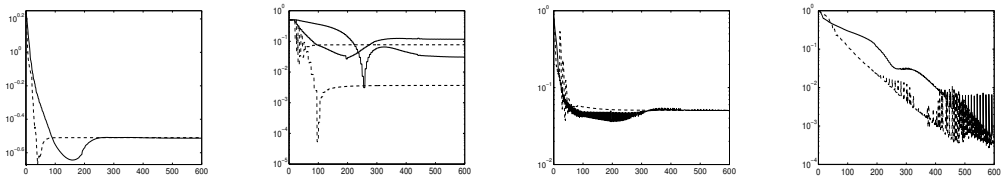
a) $||q(\mathbf{x}) - q^k||_{L^2(\Omega)}$. b) $|c_j - c_j^k|$. c) $||u(\mathbf{x}) - u^k||_{L^2(\Omega)}$. d) $||K(\phi^k)||_{L^2(\Omega)}$.

Figure 4: Example 1, Test 1: Convergence of the two methods. The solid lines are the results for OBLSM and the dashed lines are the results for SBLSM. The parameters are $\beta = 10^{-4}$ and $\mu_0 = 0.05$ for both methods. We observe that the convergence of SBLSM is slightly quicker than for OBLSM. The final errors (Fig. a), b) and c)) are also lower for SBLSM.



a) $||q(\mathbf{x}) - q^k||_{L^2(\Omega)}$. b) $|c_j - c_j^k|$. c) $||u(\mathbf{x}) - u^k||_{L^2(\Omega)}$. d) $||K(\phi^k)||_{L^2(\Omega)}$.

Figure 5: Example 1, Test 2: Convergence of the two methods. The solid lines are the results for OBLSM and the dashed lines are the results for SBLSM. For this test, β is tuned such that the error of the recovered q is approximately equal for the two methods. The parameters for OBLSM are $\beta = 10^{-3}$ and $\mu_0 = 0.025$, and the parameters for SBLSM are $\beta = 10^{-4}$ and $\mu_0 = 0.025$. According to the plotted measures, we observe a considerably faster convergence for SBLSM than for OBLSM.



a) $||q(\mathbf{x}) - q^k||_{L^2(\Omega)}$. b) $|c_j - c_j^k|$ c) $||u(\mathbf{x}) - u^k||_{L^2(\Omega)}$. d) $||K(\phi^k)||_{L^2(\Omega)}$.

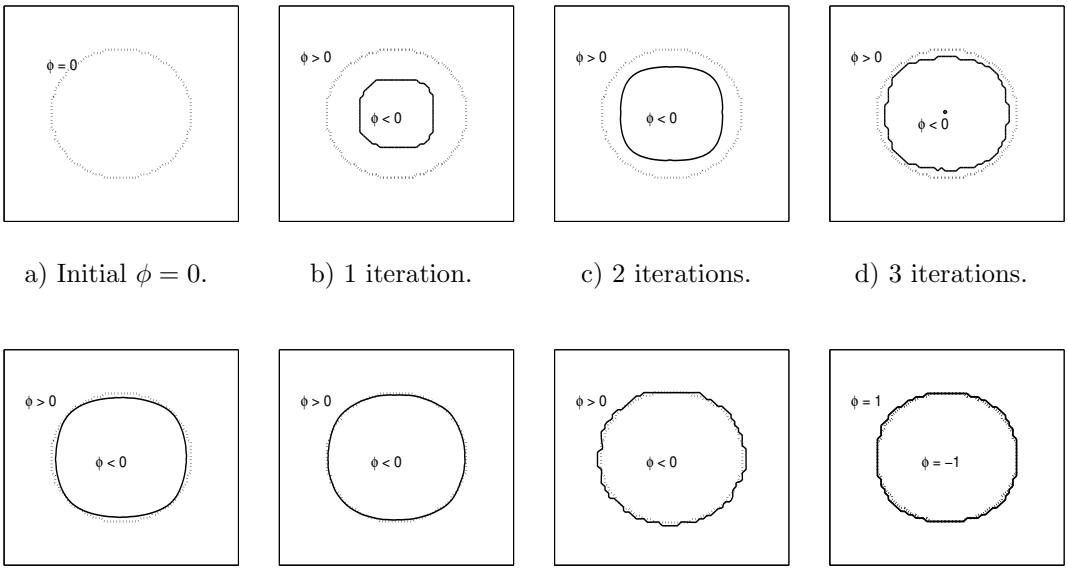
Figure 6: Example 1, Test 3: Convergence of the two methods. The solid lines are the results for OBLSM and the dashed lines are the results for SBLSM. In this case, μ_0 is tuned such that the error of the recovered coefficient is equal for the two methods. The parameters for OBLSM are $\beta = 10^{-4}$ and $\mu_0 = 0.025$, and the parameters for SBLSM are $\beta = 10^{-4}$ and $\mu_0 = 0.05$. The measures plotted in Fig. a), b) and d) give a considerably faster decrease for SBLSM than for OBLSM. The error of the computed observations, u^k, (Fig. c)) is for OBLSM decreasing as quickly as for SBLSM, though, a stable value for this measure is reached much quicker for SBLSM (after about 100 iterations) than for OBLSM (after about 300 iterations).

All the three tests show a faster convergence of SBLSM than OBLSM. Especially the convergence of the c_j-values is much faster. Also notice that for SBLSM $||K(\phi^k)||_{L^2(\Omega)}$ is approaching zero almost linearly on the semilogarithmic scale before it starts oscillating at a low value, while OBLSM gives a slower decrease. Because of the faster convergence, we apply SBLSM in the rest of the examples.

In [10] there are done studies of the same cases as tested in the rest of this paper. According to the number of requiered iterations, we observe a considerably faster convergence of the SBLSM implementation than what is the case for the continious level set formulation applied in [10]. For the remaining numerical studies not connected to the discussed three tests, we will skip the notation SBLSM as this always is the applied method.

In Figure 7 the evolvement of the level set function for Test 1 and 3 with SBLSM is shown. As in all studied cases, the value of ϕ is initially zero on the entire domain. That means, we do not assume anything about the shape of the discontinuity of ϕ (and hence of q). The initial $\phi = 0$ will produce a function $q(\mathbf{x})$ which is constant with a value equal to the arithmetic mean of the initial c_i-values.

Because of the relative simple contour in this example, the curve indicating the sign of the level set function (the solid curves in Figure 7) will approximate the discontinuity of $q(\mathbf{x})$ very quickly. It though takes longer time to get the final piecewise constant solution where ϕ is equal to 1 or -1. The error plots of $q(\mathbf{x})$ and $\mathbf{c}$ (see dotted lines in Fig. 4a) and 4b)) indicate convergence of the solution before $||K(\phi^k)||_{L^2(\Omega)}$ is close to zero. For practical applications, it may therefore be reasonable to stop before $|\phi^k|$ is exactly equal to 1.



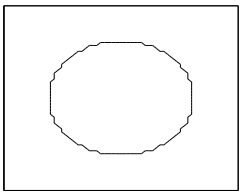
a) Initial $\phi = 0$. b) 1 iteration. c) 2 iterations. d) 3 iterations.

e) 5 iterations. f) 30 iterations. g) 200 iterations. h) 600 iterations.

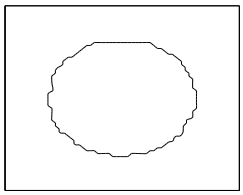
Figure 7: Example 1: The sign of the level set function is shown for different iterations (with SBLSM, Test 1 and 3). The dotted lines are the discontinuity of the true $q(\mathbf{x})$ and the solid curves indicate the sign of ϕ. Inside this curve $\phi < 0$, and outside the curve $\phi > 0$. The discontinuity of $q(\mathbf{x})$ is approximately matched after about 30 iterations, though it takes much more effort to get a stable solution where $|\phi|$=1 and both ϕ and $\mathbf{c}$ have stopped changing. The applied parameters are $\beta = 10^{-4}$ and $\mu_0 = 0.05$.

In Figure 8 we show results corresponding to different levels of noise. We have applied the same bounds for $\mathbf{c}$ as before. The shape of the discontinuity is recovered very well for low noise, and with slightly less accuracy when the noise increases. The errors in the recovered constants are increasing more rapidly for larger amount of noise.

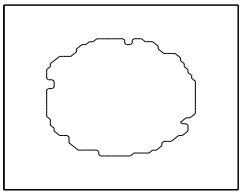
In the next study we investigate the effect of the regularisation parameter. We have used $\mu_0 = 0.025$ and added 5% noise. Except from the β parameter, the rest of set up for this study is as before. In Figure 9 we have shown the discontinuity curves for different values of β. As expected, a too low β will give an oscillating curve, and the larger the value of β is, the smoother will the curve of the discontinuity be. It seems like a value of β between 10^{-3} and 10^{-4} is a good choice for this example.



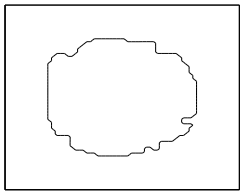
a) $\sigma = 0$, $\beta = 10^{-7}$,
$|c_1 - c_1^*| = 0.04$, $|c_2 - c_2^*| = 0.01$,
$||q(\mathbf{x}) - q^*||_{L^2(\Omega)} = 0.23$.



b) $\sigma = 0.05$, $\beta = 10^{-4}$,
$|c_1 - c_1^*| = 0.04$, $|c_2 - c_2^*| = 0.09$,
$||q(\mathbf{x}) - q^*||_{L^2(\Omega)} = 0.40$.

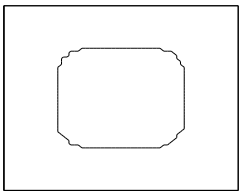


c) $\sigma = 0.20$, $\beta = 10^{-4}$,
$|c_1 - c_1^*| = 0.04$, $|c_2 - c_2^*| = 0.22$,
$||q(\mathbf{x}) - q^*||_{L^2(\Omega)} = 0.54$.

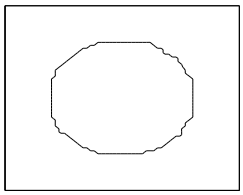


d) $\sigma = 0.40$, $\beta = 10^{-4}$,
$|c_1 - c_1^*| = 0.02$, $|c_2 - c_2^*| = 0.33$,
$||q(\mathbf{x}) - q^*||_{L^2(\Omega)} = 0.57$.

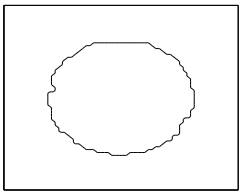
Figure 8: Example 1: The identified discontinuity of the level set function for different amount of noise. The accuracy of the discontinuity is slightly decreasing when we add more noise, while the error in the recovered constants is increasing more rapidly for larger amount of noise. $\mu_0 = 0.025$ for all levels of noise.



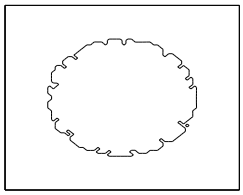
a) $\beta = 10^{-2}$,
$|c_1 - c_1^*| = 0.50$, $|c_2 - c_2^*| = 0.08$,
$||q(\mathbf{x}) - q^*||_{L^2(\Omega)} = 0.644$.



b) $\beta = 10^{-3}$,
$|c_1 - c_1^*| = 0.07$, $|c_2 - c_2^*| = 0.09$,
$||q(\mathbf{x}) - q^*||_{L^2(\Omega)} = 0.298$.



c) $\beta = 10^{-4}$,
$|c_1 - c_1^*| = 0.04$, $|c_2 - c_2^*| = 0.09$,
$||q(\mathbf{x}) - q^*||_{L^2(\Omega)} = 0.401$.

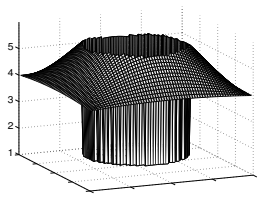


d) $\beta = 10^{-5}$,
$|c_1 - c_1^*| = 0.03$, $|c_2 - c_2^*| = 0.09$,
$||q(\mathbf{x}) - q^*||_{L^2(\Omega)} = 0.491$.

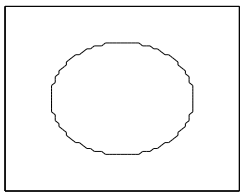
Figure 9: Example 1: The discontinuity of the level set function for different values of β. A too low value of β gives an oscillating curve of discontinuity, and as expected, an increase in β will increase the smoothness of this curve. In all cases $\sigma = 0.05$ and $\mu_0 = 0.025$.

Example 2: Approximation of a piecewise smooth function
We have in this case considered the problem of approximating a function $q(\mathbf{x})$ which is a piecewise smooth function, but not a piecewise constant function. The exact coefficient is constructed by $q(\mathbf{x}) = c(\mathbf{x}) \exp(8x_1(1-x_1)(1-x_2)$, where $c(\mathbf{x}) = 1$ inside the circle shown in Figure 10 b) and $c(\mathbf{x}) = 4$ in the rest of the domain. The noise level is 5%, and the bounds of the c_j-values are chosen to be $4 \leq c_1 \leq 5$ and $1 \leq c_2 \leq 2$.

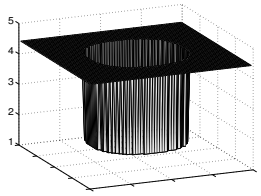
We have restricted the approximation to have two constant levels. A higher number of allowed levels will produce a more accurate approximation. From the plots in Figure 10 we can see that the location of the discontinuity is recovered rather well, and the identified function q^* is a good approximation of the true coefficient. The error $||q(x) - q^k||_{L^2(\Omega)}$ is reduced from 2 to below 0.6.



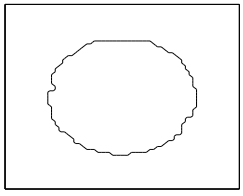
a) True $q(\mathbf{x})$.



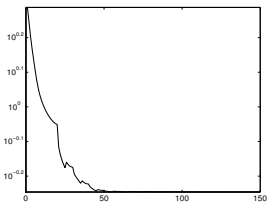
b) True discontinuity curve.



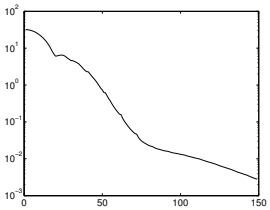
c) Identified q^*.



d) Discontinuity of ϕ^*.



e) $||q(\mathbf{x}) - q^k||_{L^2(\Omega)}$.



f) $||K(\phi^k)||_{L^2(\Omega)}$.

Figure 10: Example 2: Identification of a piecewise smooth function with $\sigma = 0.05$, $\beta = 10^{-4}$ and $\mu_0 = 0.025$. The approximation is restricted to have two levels, and the error $||q(\mathbf{x}) - q^k||_{L^2(\Omega)}$ is reduced from 2 till below 0.6.

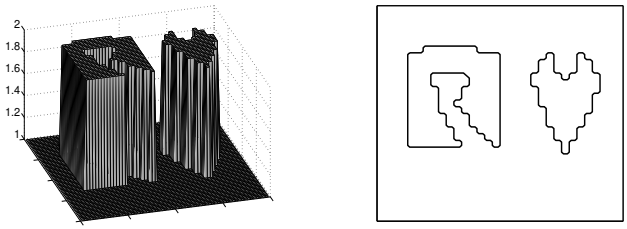
Example 3: Two regions with complicated geometry
In this example we try to recover a function with a more complicated geometry of the locations of the discontinuities. The true function $q(\mathbf{x})$ and the curve indicating the discontinuity are shown in Figure 11. This function consists of three distinct regions, but since two of the regions have identical constant values, only one level set function is needed to represent $q(\mathbf{x})$. For the lower and upper bounds in the line searches for the constant values we have applied values which are 0.2 below and above the corresponding true values. We will recover this coefficient with three different levels of noise; $\sigma = 0.001$ (Fig. 12 and 13), $\sigma = 0.01$ (Fig. 14) and $\sigma = 0.05$ (Fig. 15). The applied parameters are given in Table 2.

	$\sigma = 0.001$	$\sigma = 0.01$	$\sigma = 0.05$
β	$5 \cdot 10^{-5}$	$1 \cdot 10^{-4}$	$1 \cdot 10^{-4}$
μ_0	0.005	0.01	0.01

Table 2: Parameters for Example 3. (μ is increased during the iterations such that $\mu_k = \mu_0 \cdot 1.01^k$ up to a value of 4.)

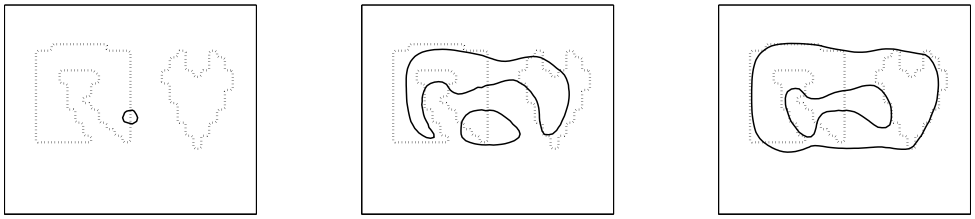
The recovered coefficient is very accurate for the case with $\sigma = 0.001$. Even the sharp corners are matched with high accuracy. Concerning the noise we can tolerate about 1% noise. With larger noise than this, both the discontinuity and the constant values are inaccurate (see Fig. 15).

The applied level set method is not moving the curve in the same way as the continuous level set formulation, but changing the value of ϕ at every grid point according to the gradient information. As in other level set methods it is no problem to split one region into two separate regions, or in the opposite case, let two separate regions merge into one region. This property is necessary to let *one* level set function represent a function with several distinct regions with equal constant levels.

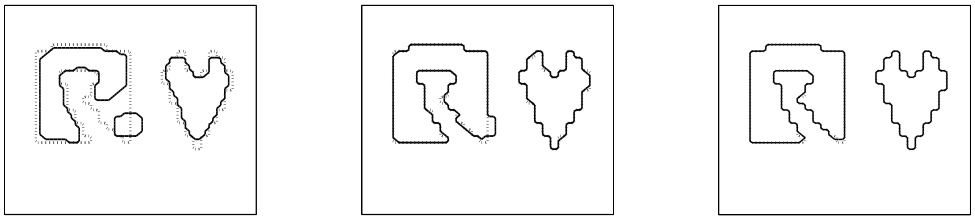


a) True $q(\mathbf{x})$. b) True discontinuity curve.

Figure 11: Example 3: True $q(\mathbf{x})$ and the corresponding discontinuity.

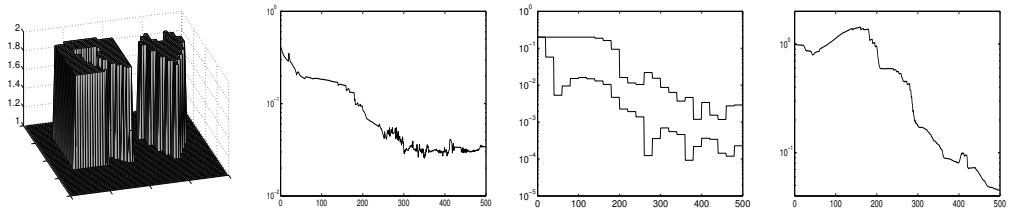


a) 3 iterations. b) 5 iterations. c) 10 iterations.



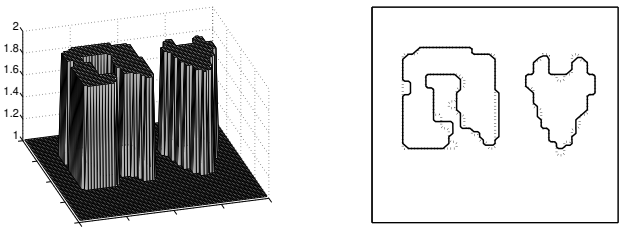
d) 50 iterations. e) 200 iterations. f) 500 iterations.

Figure 12: Example 3: The curve where the level set function changes sign (solid lines) at different iterations with $\sigma = 0.001$. The discontinuity of the true coefficient is shown with the dotted lines.



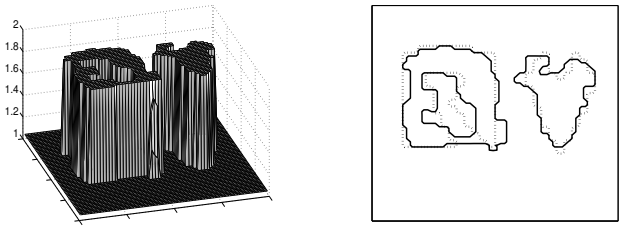
a) Computed q^*. b) $||q(\mathbf{x}) - q^k||_{L^2(\Omega)}$. c) $|c_i - c_i^k|$. d) $||K(\phi^k)||_{L^2(\Omega)}$.

Figure 13: Example 3: The computed q^* and the convergence of q^k, $\mathbf{c}^k$ and $\boldsymbol{\phi}^k$ with $\sigma = 0.001$. At convergence we had $||q(\mathbf{x}) - q^*||_{L^2(\Omega)} = 0.03$, $|c_1 - c_1^*| = 0.0005$ and $|c_2 - c_2^*| = 0.005$.



a) Computed q^*. b) Discontinuity of ϕ^*.

Figure 14: Example 3: The computed q^* and the discontinuity of ϕ^* with $\sigma = 0.01$. At convergence we had $||q(\mathbf{x}) - q^*||_{L^2(\Omega)} = 0.15$, $|c_1 - c_1^*| = 0.01$ and $|c_2 - c_2^*| = 0.05$.



a) Computed q^*. b) Discontinuity of ϕ^*.

Figure 15: Example 3: The computed q^* and the discontinuity of ϕ^* with $\sigma = 0.05$. At convergence we had $||q(\mathbf{x}) - q^*||_{L^2(\Omega)} = 0.25$, $|c_1 - c_1^*| = 0.05$ and $|c_2 - c_2^*| = 0.1$.

Example 4: Multiple level sets

In this example the same geometry as in the previous example is used, but the true $q(\mathbf{x})$ has now three regions with *different* constant values. $q(\mathbf{x})$ is equal to 1 outside the two curves, and equal to 3 and 2 inside the left and right curve, respectively (see Fig. 16). Because of more than two different constant levels, we need (at least) two level set functions to represent this coefficient function.

The lower and upper bounds of the constants are chosen to be 0.2 below and above the true values. Also in this case we do not assume anything about the geometries, but start with a zero value of both level set functions.

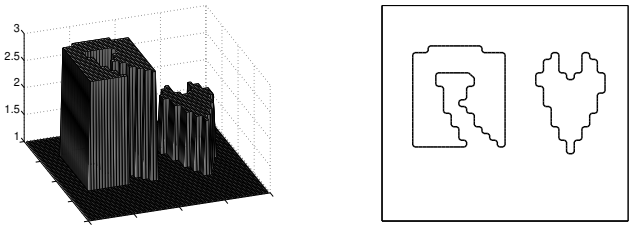
We have tested to recover this coefficient with three different levels of noise. With $\sigma = 0.001$ we show the evolvement of $\boldsymbol{\phi}$ in Figure 17 and the convergence in Figure 18. In Figure 19 and 20 the final results for the cases with $\sigma = 0.01$ and $\sigma = 0.05$ are plotted. The applied parameters for these cases are given in Table 3.

In the plots of the level set functions, ϕ_1 gives the discontinuity between all three regions, while ϕ_2 only gives the discontinuity between the highest and lowest region. This is one way to represent this function, other combinations of the different ϕ_i and c_j can produce the same $q(\mathbf{x})$. Two level set functions can represent up to four regions, but in this case we only need three regions. In the used representation, the fourth and last combination of the signs of ϕ_1 and ϕ_2 is empty. That is the case for all the tests.

Compared to the previous examples where one level set function is applied, we have to increase μ more slowly with two level set functions. Because of this we often need more iterations to achieve convergence. The result from this study is quite similar to the results from Example 3. Also in this case we can tolerate nose up to about $\sigma = 0.01$. The errors become higher in both the recovered shape and the constant values when the noise is larger than this level, see Figure 20.

	$\sigma = 0.001$	$\sigma = 0.01$	$\sigma = 0.05$
β	$5 \cdot 10^{-6}$	$5 \cdot 10^{-5}$	$1 \cdot 10^{-4}$
μ_0	$5 \cdot 10^{-4}$	0.01	0.05

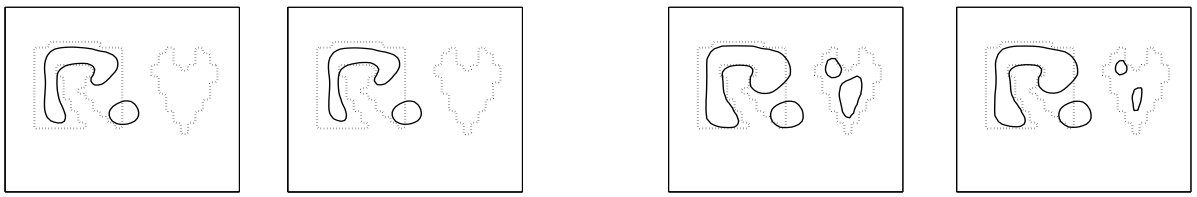
Table 3: Parameters for Example 4. (μ is increased during the iterations such that $\mu_k = \mu_0 \cdot 1.002^k$ up to a value of 4.)



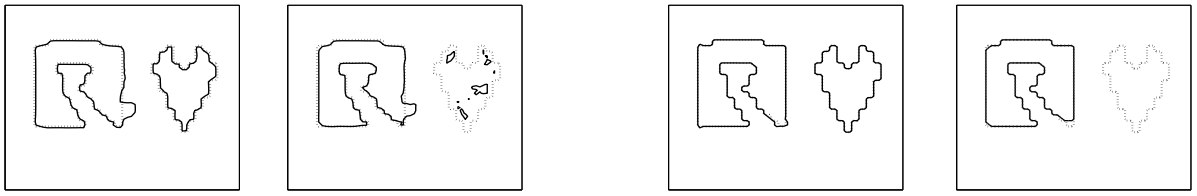
a) True $q(\mathbf{x})$. b) True discontinuity curves.

Figure 16: Example 4: True $q(\mathbf{x})$ and the corresponding discontinuities.

a) ϕ_1, 10 iterations. b) ϕ_2, 10 iterations. c) ϕ_1, 20 iterations. d) ϕ_2, 20 iterations.

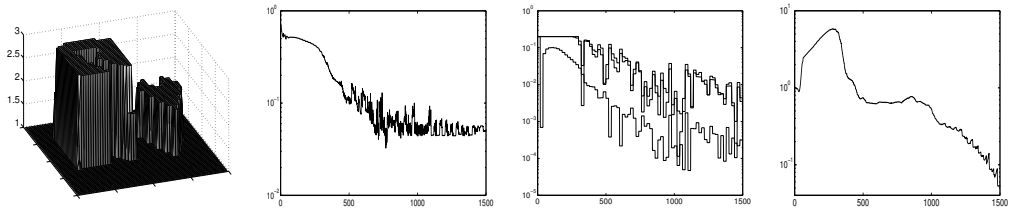


e) ϕ_1, 50 iterations. f) ϕ_2, 50 iterations. g) ϕ_1, 200 iterations. g) ϕ_2, 200 iterations.



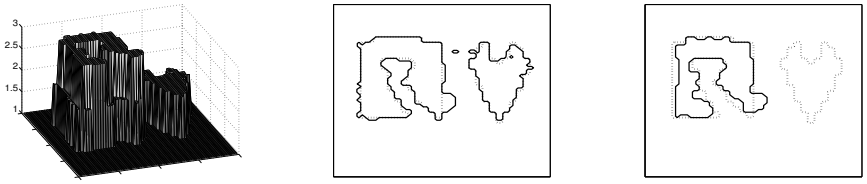
i) ϕ_1, 500 iterations. j) ϕ_2, 500 iterations. k) ϕ_1, 1500 iterations. l) ϕ_2, 1500 iterations.

Figure 17: Example 4: The curves where the level set functions change sign (solid lines) for different iterations with $\sigma = 0.001$. The discontinuities of the true $q(\mathbf{x})$ are shown with the dotted lines. The function q^* corresponding to ϕ_1 and ϕ_2 plotted in Figure k) and l) is shown in Figure 18 a).



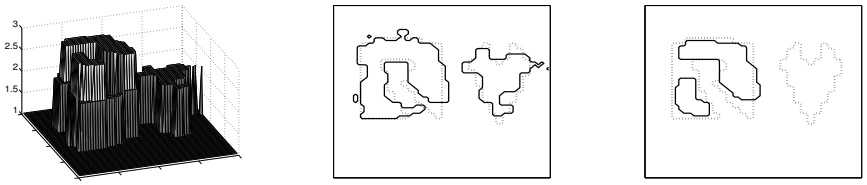
a) Computed q^*. b) $||q(\mathbf{x}) - q^k||_{L^2(\Omega)}$. c) $|c_j - c_j^k|$. d) $||\mathbf{K}(\boldsymbol{\phi}^k)||_{L^2(\Omega)}$.

Figure 18: Example 4: The computed q^* and the convergence with $\sigma = 0.001$. The errors of the computed solution are $||q(\mathbf{x}) - q^*||_{L^2(\Omega)} = 0.25$, $|c_1 - c_1^*| = 0.001$, $|c_2 - c_2^*| = 0.01$ and $|c_3 - c_3^*| = 0.01$.



a) Computed q^*. b) ϕ_1^*. c) Final ϕ_2^*.

Figure 19: Example 4: The computed q^* and the discontinuities of $\boldsymbol{\phi}$ for $\sigma = 0.01$. The errors for the computed solution are $||q(\mathbf{x}) - q^*||_{L^2(\Omega)} = 0.27$, $|c_1 - c_1^*| = 0.01$, $|c_2 - c_2^*| = 0.06$ and $|c_3 - c_3^*| = 0.07$.



a) Computed q^*. b) ϕ_1^*. c) ϕ_2^*.

Figure 20: Example 4: The computed $q(\mathbf{x})$ and the discontinuities of $\boldsymbol{\phi}$ for $\sigma = 0.05$. The errors for the computed solution are $||q(\mathbf{x}) - q^*||_{L^2(\Omega)} = 0.42$, $|c_1 - c_1^*| = 0.04$, $|c_2 - c_2^*| = 0.14$ and $|c_3 - c_3^*| = 0.03$.

8 Conclusions

In this work we have introduced a binary level set approach for solving elliptic inverse problems. The multi-level set representation is regularised by a total variational norm.

This method is not moving the interfaces during the iterative process, but moving the level set functions towards -1 or 1 at every grid point. This gives some advantages when matching special geometries; sharp corners can be recovered very accurately, and problems with merging or separating regions will not be an issue. The reinitialisation of the level set functions used in the continuous formulation is not needed for the binary level set method.

Numerical results show that rather complicated geometries can be recovered under moderate amount of noise. An initial guess of the geometries is not needed, only a reasonable guess of

the constant levels is required. When applying the suggested implementation of this method, the number of iterations to achieve convergence is considerably reduced compared with corresponding results from the continuous level set method, c.f.[10].

References

[1] R.E. Ewing, editor. *The mathematics of reservoir simulation*. Frontiers in applied mathematics. SIAM, Philadelphia, 1983.

[2] M. Cheney, D. Isaacson, and J. Newell. Electrical impedance tomography. *SIAM Review*, 41(1):85–101, 1999.

[3] D.C. Dobson and F. Santosa. An image-enhancement technique for electrical impedance tomography. *Inverse Problems*, 10:317–334, 1994.

[4] G. Chavent, J. Jaffré, and S. Jan-Jégou. Estimation of relative permeabilities in three-phase flow in porous media. *Inverse Problems*, 15:33–39, 1999.

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

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

[7] G. Chavent and K. Kunisch. Regularization of linear least squares problems by total bounded variation. *ESAIM; COCV*, 2:359–376, 1997.

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

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

[10] T. 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.

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

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

[13] 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.

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

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

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

[17] S. Osher and J.A. Sethian. Fronts propagating with curvature-dependent speed: algorithms based on hamilton-jacobi formulations. *J. Comput. Phys*, 79(1):12–49, 1988.

[18] X.-C. Tai and T. Chan. A survey on multiple level set methods with applications for identifying piecewise constant functions. *Int. J. of numerical analysis and modeling*, 1(1):25–47, 2004.

[19] M. Burger and S. Osher. A survey on level set methods for inverse problems and optimal design. CAM-Report 04-02, 2004.

[20] S. Osher and R.P. Fedkiw. *Level Set Methods and Dynamic Implicit Surfaces*. Springer, 2002.

[21] J. Lie, M. Lysaker, and X.-C. Tai. A binary level set model and some applications to mumford-shah image segmentation. *UCLA, CAM 04-31*, 2004.

[22] F. Gibou and R. Fedkiw. Fast hybrid k-means level set algorithm for segmentation. *Stanford Technical Report*, November 2002.

[23] B. Song and T. Chan. A fast algorithm for level set based optimization. *UCLA, CAM*, 68, 2002.

[24] B. Bourdin and A. Chambolle. Design-dependent loads in topology optimization. *ESAIM: Contr. Optim. Calc. Var.*, 9:19–48, 2003.

[25] C. Samson, L. Blanc-Feraud, G. Aubert, and J. Zerubia. A variational model for image classification and restoration. *TPAMI*, 22(5):460–472, May 2000.

[26] G. Aubert and P. Kornprobst. *Mathematical problems in image processing : partial differential equations and the calculus of variations*, volume 147 of *Applied mathematical sciences*. Springer, New York, 2002.

[27] J. Lie, M. Lysaker, and X.-C. Tai. A piecewise constant level set level set framework. In *European Congress on Computational Methods in Applied Sciences and Engineering*, Jyväskylä, July 2004.

[28] L.A. Vese and S. Osher. The level set method links active contours, mumford-shah segmentation, and total variation restoration. Technical report, UCLA, Math. Dept., Cam 02-05, 2002.

[29] E. Chung, T. Chan, and X.-C. Tai. Electrical impedance tomography using level set representation and total variational regularization. *Journal of Computational Physics*, 205:357–372, 2005.

[30] W.P. Ziemer, editor. *Weakly Differentiable Functions: Sobolev spaces and functions of bounded variation*, volume 120 of *Graduate Texts in Mathematics*. Springer, New York, 1989.

[31] J. Nocedal and S.J. Wright. *Numerical Optimization*. Springer Series in Optimization Research. Springer-Verlag, New York, 1999.