A new iterative regularization method using an equation of Hamilton-Jacobi type

Stefan Kindermann*

Institut für Industriemathematik, Johannes-Kepler-Universität,
A-4040 Linz, Austria

Abstract. Starting from regularization for curve and surface representation we derive an evolution equation of Hamilton-Jacobi type for the solution of ill-posed problems. The discrete version of this equation defines an iteration which incorporates both the Landweber iteration and the level set method of Santosa as a special case. We apply this method to linear integral equations of the first kind in one and two dimensions, and to a nonlinear parameter identification problem. The numerical results indicate that this method is well-suited for ill-posed problems with discontinuous solutions.

Key Words: Nonlinear regularization, ill-posed problems, discontinuous solutions, level-set method, Hamilton-Jacobi equation, regularization for surface and curve representations.

Acknoledgement AMS Subject Classifications: 65J20, 65N21, 70H20

1. Introduction

The level set method [2, 13, 15] has turned out to be an interesting and efficient way of computing the evolution of curves and surfaces. In inverse problems this tool has been utilized to identify the set of discontinuities (jumps) of piecewise constant unknown functions, taking only two values. Adopting the Eulerian point of view on the evolution of surfaces, the level set method can be efficiently implemented by an equation of Hamilton-Jacobi type. The advantage of this method comes from the availability of sophisticated and efficient numerical schemes to solve them [12, 15].

Our motivation for using Hamilton-Jacobi equation arises form a different direction. In [6, 8, 11] the idea of *regularization for curve and surface representation* was developed and applied to ill-posed problems with discontinuous solutions. In this method discontinuous solutions of ill-posed problems are represented by curves and surfaces describing their graphs. By this approach we may work with continuous graph parameterizations instead of discontinuous functions. Contrary to the level set method we do not have to restrict ourselves to an a-priori assumption on the solutions such as being piecewise constant. In this work we will show that the formulation via a Hamilton-Jacobi equation arises quite naturally from a descent method for the error functional using curve and surface parameterization of the graph.

*Supported by the Austrian Science Foundation Funds under grand SFB F013/F1317

2. Level set method

The level set approach has been used to solve inverse problems of the following kind: Find a region D such that

$$F(u) = f \tag{2.1}$$

with an unknown function u of the form

$$u = \begin{cases} 1 & x \in D \\ 0 & x \notin D \end{cases}$$

Here and in the following F denotes an abstract operator modeling the parameter-to-solution map, i.e., F maps the unknown solution u to the measured data f. The main idea is to describe the boundary of D by the zero level-set of an auxiliary function ϕ:

$$\partial D = \{x \in \Omega \,|\, \phi(x) = 0\},$$

where ϕ should be such that

$$\phi(x) > 0 \quad \text{for } x \in \mathring{D} \qquad \phi(x) < 0 \quad \text{for } x \notin \bar{D}.$$

The unknown u is completely determined by ϕ via

$$u = H(\phi),$$

with H denoting the Heaviside function.

In the next step ϕ is considered depending on a time variable t (or similarly on an iteration index for the discretized version) and an evolution equation for $\phi(x,t)$ is defined, with the aim that $\lim_{t\to\infty} H(\phi(x,t))$ approaches the unknown exact solution. The derivation of such an equation is mainly motivated by the objective to find an decrease in the error functional $\frac{1}{2}\|F(u) - f\|^2$.

Level set methods are starting with an evolution equation for ϕ of the form

$$\phi_t = V(x,t)\nabla\phi.$$

The velocity vector $V(x,t)$ is chosen in such a way that the error functional

$$J(u,t) := \frac{1}{2}\|F(u(.,t)) - f\|^2, \qquad u(.,t) = H(\phi(.,t)),$$

is decreasing, i.e., $\frac{d}{dt}J(u,t) < 0$. For instance, Santosa [13] used as velocity

$$V_S(x,t) = -F'(u)^*(F(u) - f)\frac{\nabla\phi}{|\nabla\phi|}$$

($F'(u)^*$ denotes the adjoint of the Fréchet-derivative) yielding the evolution equation

$$\phi_t = -F'(u)^*(F(u) - f)\ |\nabla\phi|. \tag{2.2}$$

Note that there are alternative choices for the velocity field, for instance, Burger [2] used

$$V_B(x,t) = \left(\nabla\Delta^{-1}[\text{sign}(\phi)(F'(u)^*(F(u) - f))]\right).$$

3. Curve and Surface Parameterization

Regularization by curve and surface parameterization was presented in [11] and extended in [6, 8]. The main idea is to represent discontinuous functions by their continuous graphs. Instead of looking for a function as unknown in an equation we consider the parameterizations of its graph as unknown. Since the graph of a (sufficiently regular) discontinuous function in one-dimension can be made continuous by connecting points of discontinuity by a vertical line, we may work with continuous parameterization of curves and surfaces.

In the context of ill-posed equations we consider the parameterization functions $a(s), b(s)$ of the graph of u as unknowns, hence we are using regularization with respect to a, b to solve the corresponding problem formulated in terms of a, b.

If the graph of an unknown - one dimensional - function u is described by some parameterizations $(a(s), b(s))$, then u can be recovered by

$$u(x) = b(a^{-1}(x)). \tag{3.1}$$

Even if the function u defined on an interval I has discontinuities, we may choose continuous representation functions (a, b) if we define the generalized inverse

$$a^{-1}(x) := \inf\{s \in I \mid a(s) = x\}.$$

By this definition the only requirement on the function a is that it is nondecreasing and maps the interval I onto itself. Hence it is a generalized (orientation preserving) diffeomorphism of I. The set of admissible functions a is given in the following definition (using $I = [0, 1]$).

$$\mathcal{P}_1 := \{a \in H^1[0,1] \mid a(0) = 0, \quad a(1) = 1, \quad \dot{a} \geq 0 \text{ a.e. } \}$$

Interestingly, if $a \in \mathcal{P}_1$ and $b \in H^1(I)$ these sets suffice to represent all BV-functions u via (3.1) (see [5]).

As a main advantage in this approach the regularized solutions $u = b(a^{-1})$ can have less regularity ($u \in BV$) than the parameterizations a and b ($\in H^1$). We can use regularization in H^1 for the parameterization and nevertheless find solution which are not in H^1 but in BV.

Remark 3.1. Of course, parameterizations of a curve are in general not unique, but given a function $u \in BV(I)$ it is possible to find a unique parameterization by choosing that one, which minimizes the seminorm

$$|(a,b)|^2_{H^1(I)\times H^1(I)} = \int_I a_s(s)^2 + b_s(s)^2 ds, \tag{3.2}$$

yielding the arclength parameterization of the graph of u (see [11]).

Another important feature is that for sufficiently smoothing operators, we do not need to evaluate a^{-1} directly, but instead use the transformation rule. For integral operators with continuous k we have the following identity (cf. [11])

$$K(b(a^{-1}))(x) := \int_I k(x,y) b(a^{-1}(y))\, dy = \int_I k(x, a(s)) b(s) \dot{a}(s)\, ds.$$

If $k \in C(I \times I)$ then the operator mapping a, b to $K(b(a^{-1}))$ is differentiable (cf., e.g., [4]). Thus the equation

$$K(b(a^{-1})) = f \tag{3.3}$$

can be solved for $a \in \mathcal{P}_1$, $b \in H^1$, by an appropriate nonlinear regularization method.

4. Regularization by a Hamilton-Jacobi Equation

For the linear and one-dimensional problem (3.3) we can *formally* derive an equation similar to (2.2) by the approach via curve representation: we start from Tikhonov Regularization for curve representation, i.e., we search for (using (3.2))

$$\begin{aligned} (a_\alpha, b_\alpha) &:= \operatorname{argmin}_{a\in\mathcal{P}_1, b\in H^1} J(a,b) \\ J(a,b) &:= \tfrac{1}{2}\left(\|K(b(a^{-1})) - f\|^2 + \alpha |a,b|^2_{H^1\times H^1}\right). \end{aligned} \tag{4.1}$$

The theory of nonlinear Tikhonov Regularization yields sufficient conditions for convergence (as $\alpha \to 0$) of the regularized parameterizations (a_α, b_α) to parameterizations of the graph of the exact solution, $(a^\dagger, b^\dagger)$, and also for convergence of $u_\alpha := b_\alpha(a_\alpha^{-1}) \to u^\dagger$ (cf. [4]). An implementation of the regularization method (4.1) and numerical results can be found in [6].

It is shown in [4] that minimizing $J(a,b)$ is equivalent to minimizing

$$J_1(a,b) := \tfrac{1}{2}\left(\|K(b(a^{-1})) - f\|^2 + \alpha\left(\int_0^1 L(a,b)(s)ds\right)^2\right) \tag{4.2}$$

with

$$L(a,b)(s) := \sqrt{\dot{a}^2(s) + \dot{b}^2(s)},$$

in the sense that for any $(a,b) \in \mathcal{P}_1 \times H^1$ we can find a $\phi \in \mathcal{P}_1$ such that $J(a(\phi), b(\phi)) = J_1(a,b) = J_1(a(\phi), b(\phi))$, hence up to a change of the parameterization these functionals are equivalent. And since $J(a,b) \geq J_1(a,b)$ both functionals have the same minima. Note that the functional $J_1(a,b)$ is invariant under parameterization, i.e., $J_1(a(\phi), b(\phi)) = J_1(a,b)$ for a smooth diffeomorphism ϕ.

Now let us describe a descent flow for the functional $J_1(a,b)$. We consider a, b depending on an additional variable t, i.e., $(a = a(t,s), b = b(t,s))$ with t indicating an artificial time variable. The evolution equation of steepest descent looks as follows:

$$\begin{pmatrix} a_t(t,s) \\ b_t(t,s) \end{pmatrix} = -\lambda(s,t) \begin{pmatrix} \frac{\partial}{\partial a} J_1(a,b) \\ \frac{\partial}{\partial b} J_1(a,b) \end{pmatrix} \tag{4.3}$$

$$\begin{aligned} \begin{pmatrix} \frac{\partial}{\partial a} J_1(a,b) \\ \frac{\partial}{\partial b} J_1(a,b) \end{pmatrix} &= \begin{pmatrix} -b_s \\ a_s \end{pmatrix} [K^*(K(b(a^{-1})) - f)](a(s)) - \\ &\qquad \alpha\, C_1(t)\, \frac{d}{ds}\left[\frac{1}{L(a,b)(s)} \begin{pmatrix} a_s \\ b_s \end{pmatrix}\right] \end{aligned} \tag{4.4}$$

with $C_1(t) = \int_0^1 L(a,b)(s)ds$ and K^* the L^2- adjoint of K. If $\lambda(s,t)$ is a chosen positive function then (a_t, b_t) is a descent direction.

The right hand side of Equation (4.3) is not in $H^1 \times H^1$, but it is well defined in $(H^1 \times H^1)'$. Hence the correct function space to look for a solution of (4.3) would be

$$(a,b) \in W(0,T) \times W(0,T), \qquad W(0,T) := \{f \in L^2([0,T],H^1) \mid \tfrac{d}{dt}f \in L^2([0,T],(H^1)')\},$$

together with appropriate Dirichlet boundary conditions on a, b.

There is one detail in Equation (4.3): Note that $J_1(a,b)$ is invariant under reparameterization, i.e., it does not change when (a,b) is replaced by $(a(\phi), b(\phi))$ with a diffeomorphism ϕ of I. However, the corresponding parts on the right hand side of (4.3) are not invariant in general. To obtain this, we may normalize the right hand of (4.3) by taking

$$\lambda(s,t) := \frac{1}{L(a,b)(s,t)},$$

hence we consider the evolution equation

$$\begin{aligned} \begin{pmatrix} a_t \\ b_t \end{pmatrix} &= \frac{-1}{L(a,b)(s,t)} \Big[\begin{pmatrix} -b_s \\ a_s \end{pmatrix} [K^*(K(b(a^{-1})) - f)](a(s)) - \\ &\qquad \alpha\, C_1(t)\, \frac{d}{ds}\left[\frac{1}{L(a,b)(s)} \begin{pmatrix} a_s \\ b_s \end{pmatrix} \right] \Big] \end{aligned} \tag{4.5}$$

Solving these equations would yield a similar regularization method as in [6]. This approach means to work in *Lagrangian* variables, i.e., the calculations are done with respect to a reference domain, that is the domain where a, b live. Instead, one can as well look for a description in *Eulerian* coordinates, which is a formulation in the variables x of the unknown function $u(x) = b(a^{-1}(x))$ itself.

We will formally derive an evolution equation for $u(x)$ from (4.5), assuming that a, b are sufficiently smooth and that $a_s > 0$ for all t. In fact, the chain rule yields

$$u_t(x,t) = b_t(a^{-1}(x,t),t) + b_s(a^{-1}(x,t),t)\frac{\partial}{\partial t}a^{-1}(x,t).$$

Together with

$$x = a(a^{-1}(x,t),t) \Rightarrow 0 = a_s(a^{-1}(x,t),t)\frac{\partial}{\partial t}a^{-1}(x,t) + a_t(a^{-1}(x,t),t)$$

we find

$$u_t(x,t) = b_t(a^{-1}(x,t),t) - \frac{b_s(a^{-1}(x,t),t)}{a_s(a^{-1}(x,t),t)}a_t(a^{-1}(x,t),t)$$

Using (4.3) and our assumptions we find that

$$\begin{aligned} u_t = -\lambda(a^{-1}(x,t),t)\Big[&\left(K^*(K(b(a^{-1})) - f)\right)(x) \left(a_s(a^{-1}(x,t),t) + \frac{b_s(a^{-1}(x,t),t)^2}{a_s(a^{-1}(x,t),t)} \right) \\ &+\alpha C_1(t) \left(-\frac{d}{ds}\left[\frac{1}{L(a,b)(s)}a_s\right] b_s + \left(\frac{d}{ds}\left[\frac{1}{L(a,b)(s)}b_s\right]\right) \right)\Big|_{s=a^{-1}(x,t)} \Big] \end{aligned}$$

Replacing b with $b(s,t) = u(a(s,t),t)$ we get

$$\begin{aligned} u_t = -\lambda(a^{-1}(x,t),t)\Big[&(K^*(K(u)-f)(x)) \left(\frac{1+u_x^2(x,t)}{(a^{-1})_x(x,t)} \right) \\ &+\alpha C_1(t) \left(\frac{u_{xx}(x,t)}{L(a,b)(a^{-1})} a_s(a^{-1}(x,t),t) \right) \Big]. \end{aligned}$$

By our choice of λ and with

$$L(a,b)(a^{-1}) = a_s(a^{-1})\sqrt{1+u_x^2} = \frac{\sqrt{1+u_x^2}}{(a^{-1})_x(x,t)}$$

we finally obtain

$$u_t = -\left(K^*(K(u)-f)\right)\left(\sqrt{1+u_x^2(x,t)}\right) + \alpha C_1(t)\left(\frac{u_{xx}(x,t)}{1+u_x^2}\right) \tag{4.6}$$

The terms involving only first order derivatives of u resemble to (2.2). The similarity is even stronger if we choose a weighted seminorm of (a,b) and give the descent directions for a and b weights. For instance, we might use

$$\|(a,b)\|_\sigma^2 := (1-\sigma)\|a_s\|_{L^2}^2 + \sigma\|b_s\|_{L^2}^2 \quad \sigma \in (0,1]$$

and

$$\begin{pmatrix} a_t \\ b_t \end{pmatrix} = -\lambda(s,t)\begin{pmatrix} (1-\sigma) \\ \sigma \end{pmatrix}\begin{pmatrix} \frac{d}{da}J_1(a,b) \\ \frac{d}{db}J_1(a,b) \end{pmatrix} \tag{4.7}$$

Here σ close to 0 means that we emphasize the inner variations (i.e., minimization with respect to the parameterization of the interval I), whereas $\sigma = 1$ would yield an ordinary Tikhonov regularization with the H^1-norm. With this weighted norm, (4.7) together with the corresponding choice of $\lambda = (\sigma a_s^2 + (1-\sigma)b_s^2)^{-\frac{1}{2}}$, the evolution equation for u becomes (skipping the second order terms)

$$u_t = -\left(K^*(K(u)-f)\right)\left(\sqrt{\sigma + (1-\sigma)u_x^2}\right). \tag{4.8}$$

Thus, for the extremal case $\sigma = 0$ this coincides almost with the equation for ϕ in (2.2). Hence in our interpretation, the level set method is a descent method for the least squares functional where the descent only uses inner variations (i.e., only use $\frac{d}{da}J_1$). In our framework, the level set method corresponds to starting with a piecewise constant function u and minimizing the error functional by finding a diffeomorphism $\phi(x,t)$ such that $u(\phi^{-1}(x,t))$ yields a decrease of the least squares functional.

The difference between (4.8) and (2.2) comes from the fact that the corresponding term of $K^*(K(u)-f)$ (u is the unknown) has the form $K^*(K(H(\phi))-f)$ (ϕ is unknown) in (2.2). Since in the application for the latter always binary valued unknown functions are occurring it is clear that this a-priori information should have to be incorporated (via the Heaviside function H) into the evolution equation. However we are interested in solving equation (2.1) for general, not necessarily two valued functions, thus we propose to use (4.8) for this purpose.

There are several points to discuss. In the derivation of (4.8) we dropped the terms involving the regularization parameter α and the second order derivative terms of u. In the context of regularization theory this means that (4.8) is the continuous analogue of a Landweber iteration (using the L^2-adjoint). From the theory one might expect instabilities of the procedure as $t \to \infty$ if the data are contaminated with noise. Therefore the Landweber iteration needs a correct choice of the stopping time dependent on the noise level (cf. [3]).

On the other hand, for any positive α, minimization of (4.2) is stable with respect to data noise. Although the correspondence of (4.8) with a Landweber type iteration suggests, that instabilities occur, this might not happen for the discretized scheme. Because by an appropriate approximation of the derivative u_x (e.g. by upwinding) we may implicitly add an artificial diffusion term and hence the iteration corresponding to (4.8) is in fact a descent method for a regularized functional with a small regularization parameter. Hence at least for a sufficiently coarse discretization no stopping time is required, which is also confirmed by the numerical results.

Let us now consider the multidimensional and nonlinear case. Using the same ideas as for curve regularization we might consider the following functional to minimize

$$J(c,\phi) = \frac{1}{2}\|F(c(\phi^{-1})) - f\|^2 \tag{4.9}$$

where c is defined on a domain Ω, and ϕ is an appropriate diffeomorphism of Ω.

We would like to define an evolution equation for $u(.,t) = c(\phi^{-1}(.,t),t)$ where c_t, ϕ_t are defined as steepest descent flow for (4.9). In the following we denote by $D\phi$ the Jacobian matrix of ϕ and by $|D\phi|$ the Jacobian determinant.

Taking the Gateaux-derivative with respect to c in direction h we obtain (assuming differentiability)

$$\begin{aligned}\frac{d}{dt}J(c+th_1,\phi)_{t=0} &= \left\langle F'(c(\phi^{-1}))h_1(\phi^{-1}), F(c(\phi^{-1}) - f\right\rangle_{L^2}\\ &= \left\langle h_1, \left[F'(c(\phi^{-1}))^* \left(F(c(\phi^{-1})) - f\right)\right](\phi)|D\phi|\right\rangle_{L^2}\end{aligned}$$

Thus the steepest descent is obtained by

$$h_1 = -\left[F'(c(\phi^{-1}))^* \left(F(c(\phi^{-1})) - f\right)\right](\phi)|D\phi|.$$

On the other hand if we keep c fixed, and try to minimize the functional over ϕ we obtain for t sufficiently small:

$$\frac{d}{dt}J(c,\phi+th_2)\Big|_{t=0} = \lim_{t=0}\frac{1}{t}\left\langle F'(c(\phi^{-1}))\left(c((\phi+th_2)^{-1}) - c(\phi^{-1})\right), F(c(\phi^{-1})) - f\right\rangle_{L^2}$$

With $r := F'(c(\phi^{-1}))^* \left(F(c(\phi^{-1})) - f\right)$ we find

$$\frac{d}{dt}J(c,\phi+th_2)_{t=0} = \left\langle r, \nabla c^T(\phi^{-1})\frac{d}{dt}(\phi+th_2)^{-1})\Big|_{t=0}\right\rangle_{L^2}$$

Now since

$$\begin{aligned}0 = \frac{d}{dt}\left[(\phi+th_2)((\phi+th_2)^{-1})\right]\Big|_{t=0} &= h_2(\phi^{-1}) + [(D\phi)(\phi^{-1})]\frac{d}{dt}(\phi+th_2)^{-1})\Big|_{t=0}\\ &= h_2(\phi^{-1}) + [D(\phi^{-1})]^{-1}\frac{d}{dt}(\phi+th_2)^{-1})\Big|_{t=0}\end{aligned}$$

we obtain

$$\frac{d}{dt}(\phi+th_2)^{-1}\Big|_{t=0} = -D(\phi^{-1})h_2(\phi^{-1}). \tag{4.10}$$

Hence,

$$\frac{d}{dt}J(c,\phi+th_2)\Big|_{t=0} = -\left\langle r(\phi)|D\phi|, \nabla c^T D(\phi^{-1})(\phi)h_2\right\rangle_{L^2}$$

This suggests to choose as steepest descent

$$h_2 = r(\phi)|D\phi| \left[D(\phi^{-1})(\phi)\right]^T \nabla c.$$

The analogue of Equation (4.5) is now

$$\begin{pmatrix} \phi_t \\ c_t \end{pmatrix} = \lambda(s_1, s_2) r(\phi)|D\phi| \begin{pmatrix} [D(\phi^{-1})(\phi)]^T \nabla c \\ -1 \end{pmatrix} \tag{4.11}$$

again with an appropriate scaling λ. Following the one-dimensional case we include a normalization such that the right hand side of Equation (4.11) becomes invariant under reparameterization. Similar as in the one-dimensional case we multiply the right hand side with

$$\lambda(s_1, s_2, t) = \left\| \begin{pmatrix} \partial_{s_1}\phi \\ \partial_{s_1} c \end{pmatrix} \times \begin{pmatrix} \partial_{s_2}\phi \\ \partial_{s_2} c \end{pmatrix} \right\|_{\mathbb{R}^3}^{-1}.$$

As in the one-dimensional case we obtain an evolution equation for u by the chain rule:

$$u_t = c_t(\phi^{-1}) + \nabla c^T(\phi^{-1}) \frac{d}{dt}\phi^{-1}.$$

Analogue to (4.10) we get

$$u_t = c_t(\phi^{-1}) - \nabla c^T(\phi^{-1}) D(\phi^{-1}) \phi_t(\phi)^{-1}.$$

With (4.11) we finally arrive at the following formula:

$$u_t = -\lambda(\phi^{-1}, t) r |(D\phi)(\phi^{-1})| \left(1 + (\nabla c^T)(\phi^{-1}) D(\phi^{-1}) D(\phi^{-1})^T (\nabla c)(\phi^{-1})\right).$$

Expressing c via u using

$$\nabla u = (D(\phi^{-1}))^T (\nabla c)(\phi^{-1})$$

and by our choice of $\lambda = |D\phi^{-1}|(\phi)$ we end up with

$$u_t = -\lambda(\phi^{-1}, t) r \frac{1 + |\nabla u|^2}{|D\phi^{-1}|} = -r\sqrt{1 + |\nabla u|^2}.$$

Just as above we may give the flow direction with respect to c and ϕ different weights. The resulting evolution equation is similar to the one-dimensional case:

$$u_t = -F'(u)^*(F(u) - f)\sqrt{\sigma + (1-\sigma)|\nabla u|^2}, \quad \sigma \in (0, 1]. \tag{4.12}$$

5. Numerical implementation and results

We consider the evolution equation (4.12) (resp. (4.8)) with weight $\sigma \in (0, 1]$. By an explicit time discretization $u_t \sim \frac{u_{n+1} - u_n}{\Delta t}$ we obtain an iteration procedure of the form

$$u_{n+1} - u_n = -\Delta t C_n F'(u_n)^*(F(u_n) - f)\sqrt{\sigma + (1-\sigma)|\nabla u_n|^2}. \tag{5.1}$$

Here C_n denotes a number which controls the length of the chosen time-step. For the computations we also have to discretize the function u_n. In the one-dimensional case we use piecewise linear functions on a uniform grid. For u_n being defined on the unit interval we use the discretization points $x_i = \frac{i-1}{N}$, $i = 1 \dots N+1$ and the corresponding values $u_{n,i} = u_n(x_i)$ and find an approximation $u_{n,h}$ by linear interpolation on each interval $[x_i, x_{i+1}]$. We use a discretization scheme by Sethian [14]: Define the forward and backward differences

$$\Delta^+ u_{n,i} := \frac{u_{n,i+1} - u_{n,i}}{\Delta x} \quad \Delta^- u_{n,i} := \frac{u_{n,i} - u_{n,i-1}}{\Delta x} \quad i = 1 \dots N+1 \quad \Delta x = x_{i+1} - x_i$$

and set $u_{n,0} = u_{n,N+2} = 0$. Then the iteration for solving an time-discretized Hamilton-Jacobi equation of the form

$$u_{n+1}(x) - u_n(x) = \Delta t v_n(x) \sqrt{\sigma + (1-\sigma)|\nabla u_n(x)|^2}$$

with an arbitrary function v_n is defined as

$$\begin{aligned} u_{n+1}(x_i) = u_n(x_i) + & \\ \Delta t C_n \Big(\min(v_n(x_i,), 0) & \sqrt{\sigma + (1-\sigma)\left(\max(\Delta^- u_{n,i}, 0)^2 + \min(\Delta^+ u_{n,i}, 0)^2\right)} + \\ & \max(v_n(x_i), 0) \sqrt{\sigma + (1-\sigma)\left(\min(\Delta^- u_{n,i}, 0)^2 + \max(\Delta^+ u_{n,i}, 0)^2\right)} \Big) \end{aligned} \tag{5.2}$$

where C_n is chosen as

$$C_n = \eta \frac{\Delta x}{\Delta t \max_{i=1\dots N+1}(|v_n(x_i)|)} \qquad \eta \in (0,1).$$

In the simplest case of our algorithm we use

$$v_n = F'(u_n)^*(F(u_n) - f). \tag{5.3}$$

For the two dimensional case we follow a similar discretization scheme. Let u_n be defined on the unit square. Then we choose the discretization points $x_{i,j} = (x_i, x_j)$, $i, j = 1 \dots N+1$, and the variables $u_{n,i,j} = u_n(x_{i,j})$. We define the differences analogue to the one-dimensional case

$$\begin{aligned} \Delta_x^+ u_{n,i,j} &:= \frac{u_{i+1,j} - u_{i,j}}{x_{i+1,j} - x_{i,j}} \quad \Delta_x^- u_{n,i,j} := \frac{u_{i,j} - u_{i-1,j}}{x_{i,j} - x_{i-1,j}} \quad i,j = 1 \dots N+1, \\ \Delta_y^+ u_{n,i,j} &:= \frac{u_{i,j+1} - u_{i,j}}{x_{i,j+1} - x_{i,j}} \quad \Delta_y^- u_{n,i,j} := \frac{u_{i,j} - u_{i,j-1}}{x_{i,j} - x_{i,j-1}} \quad i,j = 1 \dots N+1. \end{aligned}$$

Again we extend u_n by zero for the values $x_{i,j}$ which are not in $[0,1]^2$. The difference scheme for (5.1) looks as follows:

$$\begin{aligned} u_{n+1}(x_{i,j}) = u_n(x_{i,j}) & \\ \Delta t C_n \Big\{ \min(v_n(x_{i,j}), 0) & \Big[\sigma + (1-\sigma)\Big(\max(\Delta_x^- u_{n,i,j}, 0)^2 + \min(\Delta_x^+ u_{n,i,j}, 0)^2 + \\ & \qquad \max(\Delta_y^- u_{n,i,j}, 0)^2 + \min(\Delta_y^+ u_{n,i,j}, 0)^2 \Big)\Big]^{\frac{1}{2}} \\ + \quad \max(v_n(x_{i,j}), 0) & \Big[\sigma + (1-\sigma)\Big(\min(\Delta_x^- u_{n,i,j}, 0)^2 + \max(\Delta_x^+ u_{n,i,j}, 0)^2 + \\ & \qquad \min(\Delta_y^- u_{n,i,j}, 0)^2 + \max(\Delta_y^+ u_{n,i,j}, 0)^2 \Big)\Big]^{\frac{1}{2}} \Big\} \end{aligned}$$

with

$$C_n = \eta \frac{\Delta x}{\Delta t \max_{i,j=1\dots N+1}(|v_n(x_{i,j})|)} \qquad \eta \in (0,1) \quad \Delta x = x_{i+1,j} - x_{i,j} = x_{i,j+1} - x_{i,j}.$$

5.1. Linear Problem

We applied the algorithm to a one-dimensional integral equation of the first kind, i.e., we want to solve

$$F(u) = \int_0^1 k_1(x,y)u(y)dy = f.$$

As kernel function we choose the smoothing kernel

$$k_1(x,y) = \begin{cases} (1 - \frac{(x-y)^2}{0.1})^6 & \text{if } (x-y)^2 \leq 0.1 \\ 0 & \text{else} \end{cases}$$

We used three test examples of discontinuous piecewise polynomial solutions:

Example 1:

$$u_1(x) = \begin{cases} \frac{x-0.1}{0.6} & \text{if } x \in [0.1, 0.7] \\ 0 & \text{else} \end{cases}$$

Example 2:

$$u_2(x) = \chi_{[0.3,0.5]}(x)$$

Example 3:

$$u_3(x) = 0.5\chi_{[0.3,0.5]}(x) + \chi_{[0.65,0.75]}(x)$$

In our computations, a spacial discretization with $N = 100$ was used. For the noise free case we stopped the iteration when $\frac{\|u_n - u_{n-1}\|}{\|u_n\|} \leq 10^{-4}$. As a starting value we took $u_0 = 0$.

As a two-dimensional linear test example we consider again an integral equation with the corresponding two-dimensional kernel

$$k_2(x,y,s,t) = k_1(\|(x,y)^T - (s,t)^T\|).$$

The exact solution was taken to be of the following kind:

Example 1 (Circle)

$$u_1(x,y) = \chi_{B_{0.55,0.45}(0.3)}$$

Example 2 (Ramp)

$$u_2(x,y) = \frac{x-0.25}{0.35}\chi_{\{(x,y)|0.25 \leq x \leq 0.6, 0.2 \leq y \leq 0.8\}}$$

Example 3 (Moon)

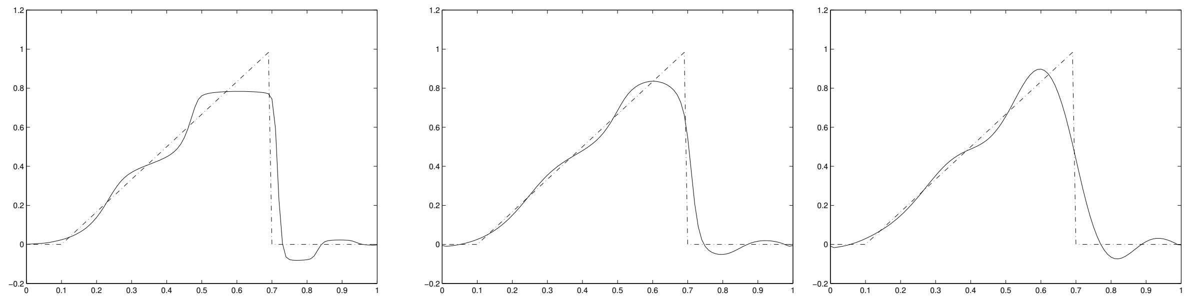
$$u_3(x,y) = \chi_{B_{0.55,0.5}(0.3)}(1 - \chi_{B_{0.45,0.5}(0.2)})$$

($B_{x_0,y_0}(r)$ denotes the circle with midpoint at (x_0, y_0) and radius r, χ is the characteristic function). For the spacial discretization we used $N = 25$ in each coordinate direction.

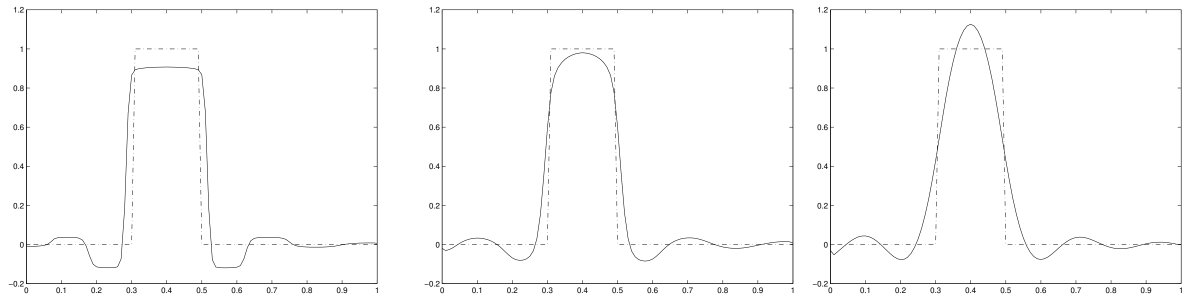
Figure 5.1 shows the result for the one-dimensional test examples, for different choices of σ. The exact solution is shown dash-dotted in the pictures.

Figure 5.2 shows the computed solution for the case of the two-dimensional integral equation using exact data and again for different choices of σ.

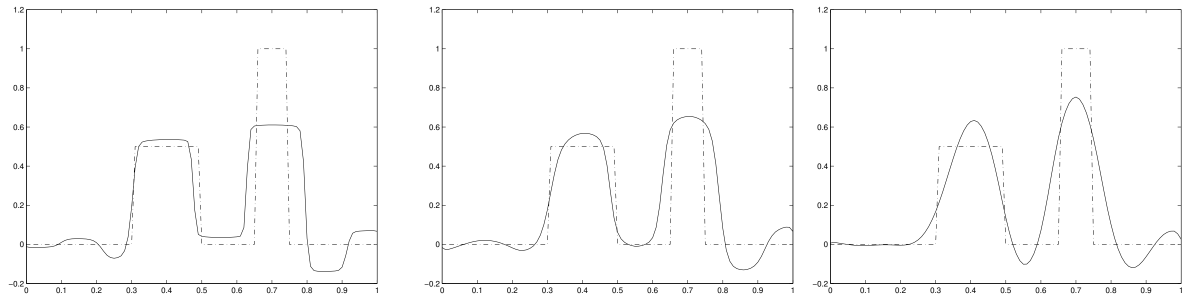
Both from the one-dimensional (Example 1) and two-dimensional (Example 2) results we see that for small σ the solution suffers from the so-called staircase effect: This means, that the results have the tendency to become piecewise constant, even if the



Example 1, $\sigma = 0.01$ Example 1, $\sigma = 0.5$ Example 1, $\sigma = 0.99$



Example 2, $\sigma = 0.01$ Example 2, $\sigma = 0.5$ Example 2, $\sigma = 0.99$



Example 3, $\sigma = 0.01$ Example 3, $\sigma = 0.5$ Example 3, $\sigma = 0.99$

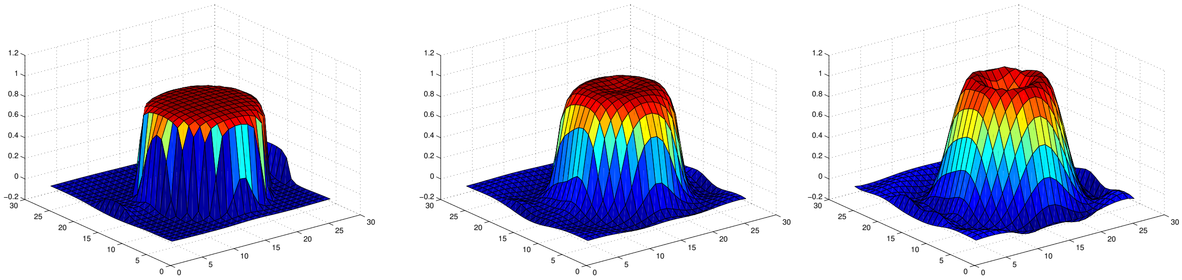
Figure 5.1: Results for exact data

exact solution is linear. This effect is well-known to happen for many BV-like regularizations and filter. For $\sigma \sim 1$ we observe the well-known Gibbs-phenomenon, that the computed solution shows an 'overshoot' where the exact solution has jumps.

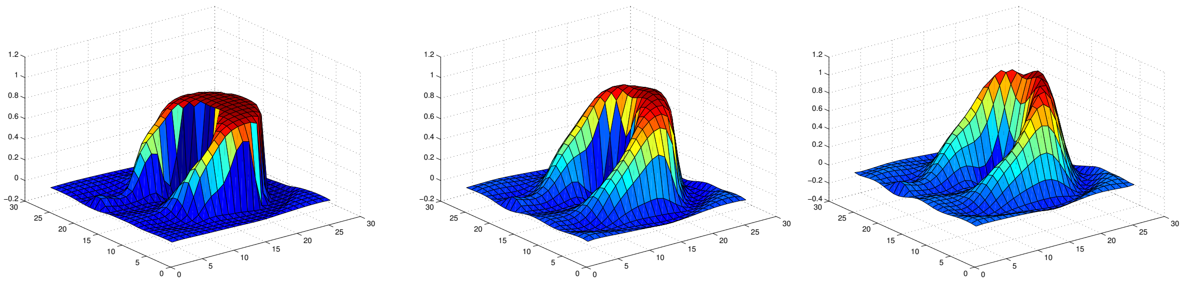
On the other hand, for the choice $\sigma = 0.5$ both effects can be mildered, especially for the piecewise linear solution this yields best results. (Note that the staircase effect is irrelevant in the context of the level set method, since there the exact solution is always taken as piecewise constant.) If the solution is known to be piecewise constant a-priori, then a choice of a small σ seems to be best.

When $\sigma \sim 0$ we noticed that the convergence is rather slow. This is clear since we choose as initial guess $u_0 = 0$ and hence $\nabla u_0 = 0$. Thus, the right hand side in (5.1) is multiplied in the first iterations by a factor of the order $\sqrt{(\sigma)} \sim 0$.

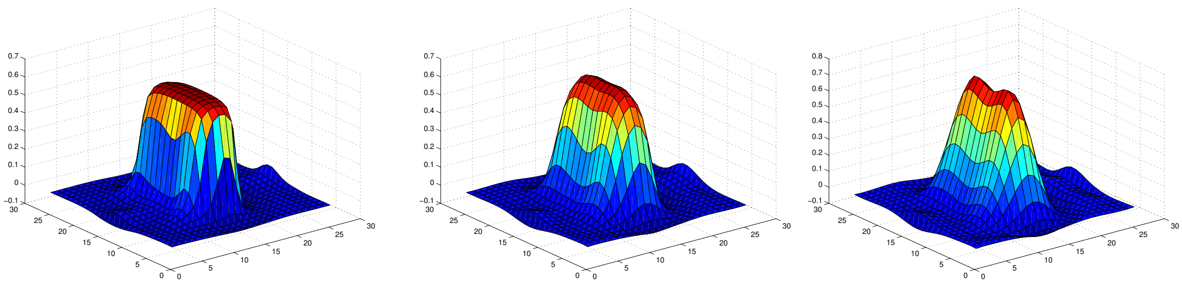
We also observed that with this choice of σ the iteration is quite sensitive with respect to the initial guess, which can be explained from our interpretation that for small σ we are only using inner variations for minimizing the least squares functional, i.e. the initial guess is only deformed by diffeomorphisms.



Example 1, $\sigma = 0.01$ Example 1, $\sigma = 0.5$ Example 1, $\sigma = 0.99$



Example 2, $\sigma = 0.01$ Example 2, $\sigma = 0.5$ Example 2, $\sigma = 0.99$



Example 3, $\sigma = 0.01$ Example 3, $\sigma = 0.5$ Example 3, $\sigma = 0.99$

Figure 5.2: Results Two-Dimensional Integral Equation

As a remedy we propose to choose σ depending on the iteration index, in such a way that for the first iterations σ is close to 1 and it decreases with the iteration. For instance we used

$$\sigma_n = \max(0.995^n, \sigma_\infty) \tag{5.4}$$

where σ_∞ is the final value of σ. Moreover we used as a starting value an initial guess which is obtained by Tikhonov regularization:

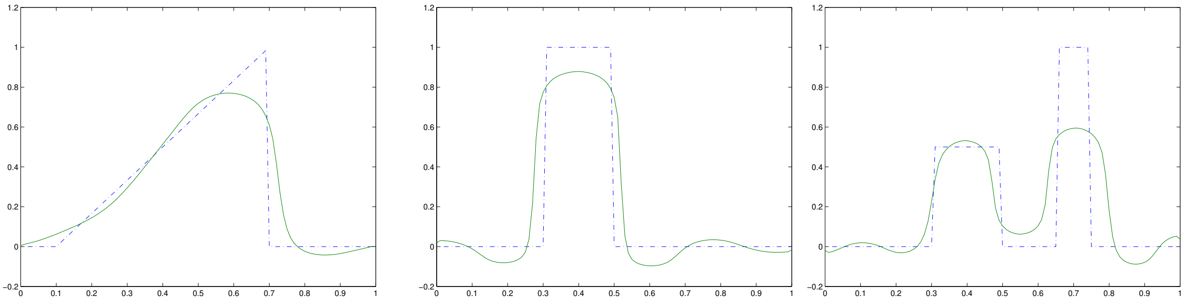
$$u_0 = (K^*K) + \alpha_0 G)^{-1} f$$

α_0 should be not to small, since otherwise the residual is already close to 0 and the iteration will almost saturate then. We choose $\alpha_0 = 1$. By this procedure we could significantly improve the speed of the algorithm with similar results as before.

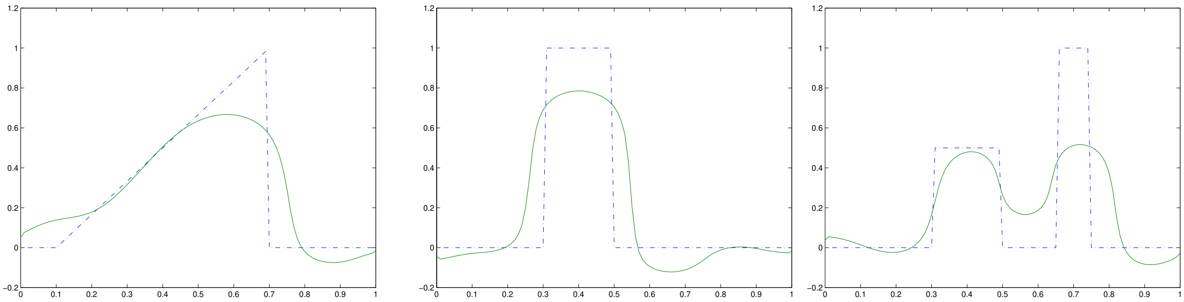
We now turn to the case of noisy data f_δ. Here we perturbed the exact data by adding uniformly distributed random noise. The iteration is stopped according to Morozov's discrepancy principle, i.e. for the first iteration index, which satisfies

$$\|F(u_n) - f_\delta\| \leq \tau\delta, \quad \tau > 1,$$

where $\delta = \|f - f_\delta\|$ denotes the noise level. The parameter τ was chosen to $\tau = 1.5$. Figure 5.3 show the result of the iteration with the choice of σ as in (5.4) and the velocity (5.3), for the noise level δ of 10 % and 50%. The final value of σ was in this case $\sigma \sim 0.3$.



Example 1, 10% noise | Example 2, 10% noise | Example 3, 10% noise



Example 1, 50% noise | Example 2, 50% noise | Example 3, 50% noise

Figure 5.3: Results for noisy data

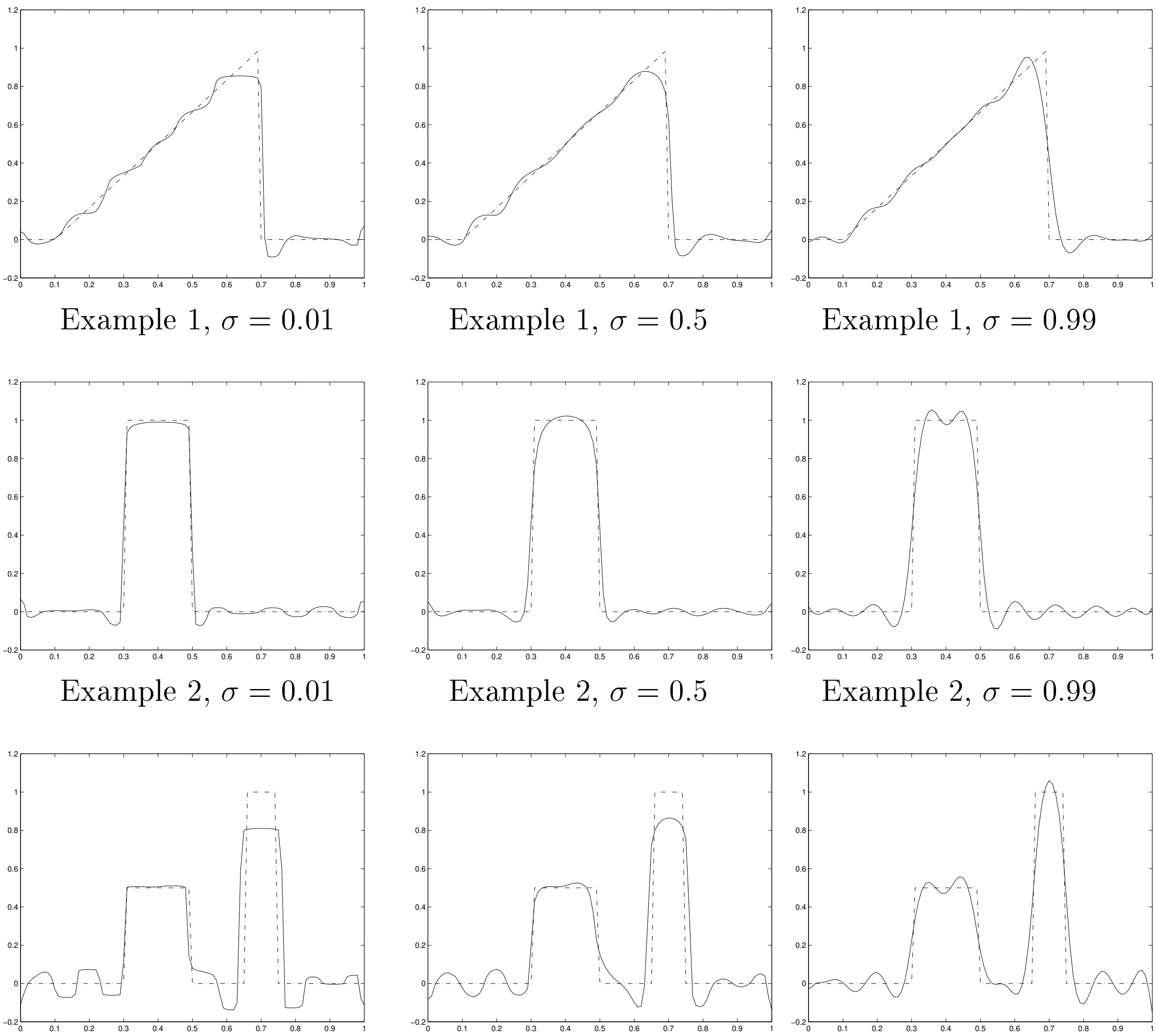
From the figure it can be observed that the iteration is stable with respect to data noise.

We also tried several modification for the iteration (5.1): We can improve the performance of the algorithm by choosing different type of velocity v_n. We already pointed out that in the extremal case $\sigma = 1$ the method (5.1) becomes the Landweber iteration, which is know to be rather slow. Hence we expect to find an improvement of the method for a different choice of the velocity. Note that the iteratively regularized Gauss-Newton method (IRGN) [1] is in general faster than the Landweber iteration. This motivates us to use a modified velocity, corresponding to the IRGN method: For the results in figure 5.4 we used

$$v_n = (K^*K) + \alpha G)^{-1} K^*(Ku_{n-1} - f), \tag{5.5}$$

where G is the mass matrix $G_{i,j} = \int_0^1 \phi_i(x)\phi_j(x)dx$, and ϕ_i are the continuous piecewise linear H^1-finite-element ansatz functions on a uniform rectangular grid (Courant elements). In our examples we set $\alpha = 10^{-5}$.

The results are quite satisfactory for Example 1 and Example 2. For Examples 3 oscillations in the computed solution indicate that an additional regularization term in the iteration might be useful.



Example 1, $\sigma = 0.01$ — Example 1, $\sigma = 0.5$ — Example 1, $\sigma = 0.99$

Example 2, $\sigma = 0.01$ — Example 2, $\sigma = 0.5$ — Example 2, $\sigma = 0.99$

Example 3, $\sigma = 0.01$ — Example 3, $\sigma = 0.5$ — Example 3, $\sigma = 0.99$

Figure 5.4: Results IRGN-iteration

5.2. Nonlinear Problem

We finally applied the iteration (5.1) to a nonlinear parameter identification problem. As a benchmark we choose the problem of identification of the γ in the elliptic PDE

$$\begin{aligned} \nabla.((1+\gamma)\nabla w) &= \rho \quad \text{in } \Omega = [0,1]^2 && (5.6)\\ w\Big|_{\partial\Omega} &= 0 && (5.7) \end{aligned}$$

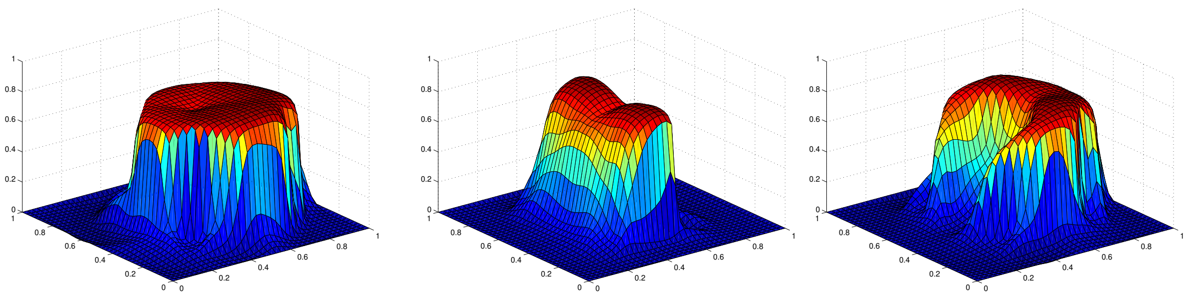
from measurements of $w(x)$, $x \in \Omega$, and with given right hand side ρ. The forward operator mapping the parameter to the data is

$$F(\gamma) = w(x) \quad x \in \Omega,\ w \text{ solution of } (5.6), (5.7).$$

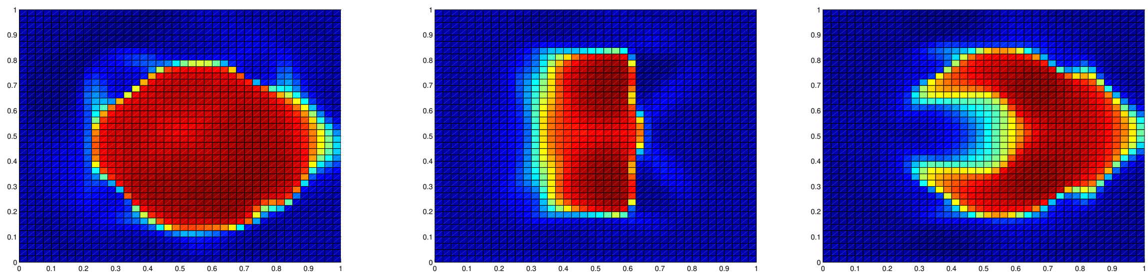
For a more detailed description of this problem and an approach via the moving grid method we refer to [7, 9]. Due to the nonlinearity of the problem we observed that the choice of v in (5.3) yields satisfying results, but shows rather slow convergence. Hence we used the Gauss-Newton-type velocity (5.5), where the inversion is replaced by a fixed number of Conjugate-Gradient iterations. This regularization procedure, i.e IRGN with Conjugate-Gradient iteration, has also successfully been used in [9].

As right hand side f we took $\rho(x, y) = \sin(2\pi x)\cos(\pi y)$. The discretization was done on a 40×40 grid. We applied the choice of σ according to (5.4), with initial guess $\gamma_0 = 0$, and $\sigma_\infty = 0.1$. In order to avoid inverse crimes we computed the data on a much finer grid (120×120) and used linear interpolation to obtain data on the computation grid.

Figure 5.5 shows the result for exact data for the same choice of test example as in the linear case before (Circle, Ramp and Moon).



Example 1 Example 2 Example 3



Example 1 Example 2 Example 3

Figure 5.5: Computed γ

From the picture it can be seen that the discontinuities are found reasonable well. Again we observed difficulties to detect the shape of the ramp, due to the staircasing.

Remark 5.1. We can compare the iteration (5.1) with the iterations in the limiting case $\sigma = 0$ and $\sigma = 1$, i.e. the level set method and the Landweber iteration.

If we consider the convergence theory for the later, it might be expected that the iteration becomes instable as $n \to \infty$ in the presence of noise. In fact, in the noisy case the Landweber iteration has to be stopped according to a stopping rule, for instance the discrepancy principle. However this instability does not occur in our numerical experiment. Even if the data are contaminated with noise, the iteration saturates. This significant difference to Landweber iteration is due to the numerical discretization scheme, which adds artificial diffusion to the process. This diffusion can be seen as part of a regularization effect, where the regularization parameter depends on the discretization size and the type of the discretization scheme.

Of course, by this reasoning the regularization by the diffusion of the numerical scheme will depend on the discretization itself. The theory of Tikhonov regularization [3] requires an regularization parameter which has to be correlated to the data noise. Hence if the grid size will tend to zero the discretized solution may not be bounded. The remedy to this would be to include the regularization term with an appropriate regularization parameter α if the discretization becomes small. However, in our numerical experiments we observed that this was not necessary.

If we look at the case $\sigma = 0$ we immediately see that the iteration need not converge to minimum norm solution of (2.1), for instance if u_n is constant except at the points where v vanishes, then the iteration saturates at u_{n+1} which need not be a solution to the original problem. This is related to the fact that minimization of the least squares functional with respect to inner variation is nonlinear, even if the underlying operator is linear, and thus the corresponding functional can have many local minima. Note that this saturation effect does not happen when $\sigma > 0$.

Comparing the approach in Eulerian variables via the iteration (5.1) with those in Lagrangian variables [9, 10] it can be said that a fixed grid of the Eulerian variables is obviously more simple to implement. Moreover, the iteration (5.1) can easily be embedded into existing code for standard iterative regularization methods. On the other hand the adaptive grid of the Lagrangian approach provides more flexibility in the approximation of the unknown solution. Hence for the same gridsize an adaptive method will in general yield better results.

Acknowledgment

The author would like to thank A. Neubauer for the very constructive and helpful discussions.

References

[1] B. Blaschke, A. Neubauer, and O. Scherzer, *On convergence rates for the iteratively regularized Gauss-Newton method*, IMA Journal of Numer. Anal. 17 (1997), 421–436.

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

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

[4] S. Kindermann, *Regularization of Ill-posed Problems with discontinuous solutions by Curve and Surface Representation*, PhD-thesis, University of Linz, October 2001.

[5] S. Kindermann and A. Neubauer, *Each $\mathcal{BV}$-function is representable by an $\mathcal{H}^1$-curve*, Technical Report 1/1999, Industrial Mathematics Institute, University of Linz, 1999.

[6] ——, *Identification of discontinuous parameters by regularization for curve representations*, Inverse Problems 15 (1999), 1559–1572.

[7] ——, *Estimation of discontinuous parameters of elliptic partial differential equations by regularization for surface representations*, Inverse Problems 17 (2001), 789–803.

[8] ——, *Regularization for surface representations of discontinuous solutions of linear ill-posed problems*, Numer. Funct. Anal. Optim. 22,1&2 (2001), 79–105.

[9] ——, *Parameter identification by surface representation via the moving grid approach*, Technical Report, SFB-Report, University Linz, 2002, submitted.

[10] A. Neubauer, *Estimation of discontinuous solutions of ill-posed problems by regularization for surface representations: numerical realization via moving grids*, in: Y.C. Hon, M. Yamamoto, J. Cheng, and J.Y. Lee, eds., Recent Development in Theories and Numerics, International Conference on Inverse Problems, World Scientific Publisher, Singapore, 2003, 67–83.

[11] A. Neubauer and O. Scherzer, *Regularization for curve representations: Uniform convergence for discontinuous solutions of ill-posed problem*, SIAM J. Appl. Math 58 (1998), 1891–1900.

[12] S. Osher and J. A. Sethian, *Fronts propagating with curvature depending speed: algorithms based on Hamilton-Jacobi formulations*, J. Comp. Physics 79 (1988), 12–49.

[13] F. Santosa, *A level-set approach for inverse problems involving obstacles*, ESAIM, Control Optim. Calc. Var. 1 (1996), 17–33.

[14] J. A. Sethian, *A fast marching level set method for monotonically advancing fronts*, Proc. Natl. Acad. Sci. USA 93 4 (1996), 1591–1595.

[15] ——, *Level Set Methods and Fast Marching Methods*, Cambridge Monographs on Applied and Computational Mathematics, Cambridge University Press, Cambridge, 1999.