Higher Order Variational Methods for Noise Removal in Signals and Images

Diploma Thesis

Stephan Didas

under supervision of

Prof. Joachim Weickert

Department of Mathematics
Saarland University

Saarbrücken, April 2004
– Corrected version, October 2004 –

Preface

Variational methods have become more and more important in image processing during the last years. They offer an intuitive way to understand the noise removal and image restoration process as minimisation of an energy functional. This energy functional provides the opportunity to compare the quality of two filtered versions of the same input image and thus can be seen as a quality measure.

Variational image restoration methods in general yield high quality results. Nonlinear approaches are known to suffer from the effect of turning smooth grey value changes in piecewise constant regions. This effect is called stair-casing.

One can interpret stair-casing as the result of energy functionals that reward piecewise vanishing first derivatives which yields piecewise constant signals and images. A way to circumvent this would be to use higher derivatives in the penaliser terms: Piecewise vanishing second derivatives would lead to piecewise linear results that could better fit to smooth grey value changes. Higher derivatives could analogously yield piecewise polynomials of higher degree.

There is a growing interest in higher order variational methods in the literature. Nielsen, Florack and Deriche (see [18]) have studied linear energy functionals of arbitrary order. They derive basic scale-space properties and give efficient filtering algorithms for the linear case. Greer and Bertozzi further investigate some theoretical properties of special fourth order evolution equations used in image restoration (see [10]). Other authors rather concentrate on the goal of showing the use of higher order methods for practical purposes. Chan, Marquina and Mulet (see [5]) study nonlinear energy functionals of second order and the related diffusion filtering with derivative orders up to four. Total variation methods with the second derivative leading to fourth order partial differential equations are used in [17] by Lysaker, Lundervold and Tai. Besides images they consider one-dimensional signal processing. You and Kaveh [34] also use fourth order PDEs for filtering. Numerical examples are presented which show that higher order methods can in practise be useful for image denoising.

The goal of the present thesis is to investigate some theoretical and practical aspects of higher order variational methods. We use different penaliser functions and compare them in practical applications. To introduce variational methods some facts about first order methods are summarised in Chapter 1.

In Chapter 2 we start with a continuous higher order energy functional and deduce necessary and sufficient conditions for a function to be a minimiser of this functional. The main result are the so-called Euler-Lagrange equations in one and two dimensions. A closer look is also taken at the natural boundary conditions which come with this partial differential equation if we impose no boundary conditions at our set of possible solutions. In the first order context the natural boundary conditions are of Neumann type. This coincides with the usage of Neumann boundary conditions in image processing since they usually yield good visual results. We will see that for higher order functionals the boundary conditions may be a bit more complicated. Signal processing results show that both natural and Neumann boundary conditions may lead to acceptable results, while in image processing Neumann boundary conditions are preferred. With the trace, the Frobenius norm, and the determinant of the Hessian, equivalents for the second derivative in 2D energy functionals are discussed.

Chapter 3 discusses several approaches how to deduce continuous filtering methods from energy functionals or corresponding Euler-Lagrange equations. In the case of linear penalisers one can give direct minimisation methods in the Fourier domain as investigated in [18]. These methods are not applicable in the nonlinear context. Thus one considers the Euler-Lagrange equation and the corresponding diffusion equations. Generalised higher order linear diffusion leads to convolution kernels axiomatically derived in [16]. We try to give an explanation for the behaviour of simple nonlinear filters in terms of forward and backward diffusion.

To use the methods presented in Chapter 3 in practice we need to discretise them. With finite differences and spectral methods, two different ways for derivative approximation are presented in Chapter 4. We are especially interested in matrix representations for discretisations. For finite differences we discuss possibilities to implement both natural and Neumann boundary conditions.

With these discretisation matrices we give discrete versions of the minimisation approaches in Chapter 5. We see that one can either start with a continuous energy functional and discretise the corresponding Euler-Lagrange equations or discretise the energy functional itself. Discretisations for nonlinear diffusion equations obtained from the Euler-Lagrange equations are also investigated. A key point are stability criterions for explicit methods we use to obtain most of the numerical results presented in Chapter 6. These should give a visual impression of the possibilities of nonlinear higher order filtering.

The last Chapter 7 concludes the thesis with a short summary including interesting questions arising in the thesis that invite to be further investigated.

At this place I would like to take the opportunity to thank the people that made this thesis possible. First I like to thank Prof. Joachim Weickert for the interesting topic and for his constant support. Besides many theoretical conceptions he provided me with an implementation of 2D higher order linear filtering with spectral methods. I also thank the members of the Mathematical Image Analysis Group at Saarland University – especially Dr. Bernhard Burgeth – for many hints and discussions on image processing including higher order methods. I thank Natalie Marx – not only for her help on the correction of this thesis. My thank goes to my brother Michael for arousing my interest in mathematical questions. In particular I would like to thank my parents for their permanent support that made my studies possible.

Contents

Chapter 1

First Order Variational Methods

This chapter gives a short overview of first order variational methods in signal and image processing. First we mention how the notions signal and image are used during this thesis: In a continuous framework a signal is usually a real-valued function $f : \Omega \longrightarrow \mathbb{R}$ with $\Omega \subseteq \mathbb{R}$ open. For our purposes it is normally sufficient to assume that Ω is an open interval. Sometimes we use $\Omega = \mathbb{R}$ in considerations in the continuous setting and then usually assume that f has compact support. An image is also a real-valued function where $\Omega \subseteq \mathbb{R}^2$ is an open two-dimensional set. We sometimes use the notion image as generic term including also signals. We often denote the initial image of a filtering or restoration process with f and the filtered version with u.

Variational methods in image processing usually start with an energy functional of the form

$$\mathcal{E}(u) = \int\limits_{\Omega} \left((u-f)^2 + \alpha\, \varphi\left(u_x^2\right) \right) dx \tag{1.1}$$

for one-dimensional examples or

$$\mathcal{E}(u) = \int\limits_{\Omega} \left((u-f)^2 + \alpha\, \varphi\left(|\nabla u|^2\right) \right) dx \tag{1.2}$$

in two dimensions. These methods are also called **regularisation**. The typical energy functionals consist of two terms:

1. The **data term** (also called **similarity term**) $(u-f)^2$: This term rewards similarity to the initial image.

2. The **smoothness term** (also called **regulariser** or **penaliser**) $\varphi\left(u_x^2\right)$: This term rewards some kind of smoothness by involving the first derivative of the image u. Different examples for the penaliser function φ are discussed below.

During this thesis we classify the methods with respect to the derivative order that appears in the smoothness term of the energy functional. Thus the methods considered in this chapter are first order methods. The parameter $\alpha > 0$ is called regularisation parameter and serves as a weight between similarity to the initial data and smoothness.

Variational methods in the above form can be related to (in general nonlinear) diffusion filtering. A detailed investigation of this relation is presented in [22]. With an additional artificial time variable for u and the assumption that the initial image is $u(\cdot, 0) = f$ we get the diffusion equation

$$u_t = \frac{d}{dx}\left(\varphi'\left(u_x^2\right) u_x\right) \tag{1.3}$$

for the functional (1.1). In two dimensions the corresponding diffusion equation reads as

$$u_t = \text{ div}\left(\varphi'\left(|\nabla u|^2\right) \nabla u\right)$$

with initial condition $u(\cdot, 0) = f$. In both cases Neumann boundary conditions

$$\partial_n u = 0 \quad \text{for all } x \in \partial\Omega$$

are assumed. We will further explain how to obtain this relation in Chapter 3. We note that the regulariser $\varphi(x^2)$ corresponds to the diffusivity $\varphi'(x^2)$. We should also mention that first order variational methods lead to diffusion equation with highest derivative order two.

The first example of variational methods in literature can be found in a paper by Tikhonov (see [28]). Here the general regularisation method with linear penaliser

$$\varphi(x) = x$$

is proposed in the context of giving approximative solutions to ill-posed problems. First applications in the field of image processing are presented in [3] and [19]. There are different classical penaliser functions used for image processing.

Besides the Tikhonov regulariser we also consider the function

$$\varphi(x) = 2\lambda^2 \left(\sqrt{1 + \frac{x}{\lambda^2}} - 1\right) \qquad \text{for } \lambda > 0$$

which was first suggested by Charbonnier et al. (see [6]) in 1994.

Rudin, Osher and Fatemi (see [21]) proposed regularisation approaches based on minimisation of the total variation

$$\int_\Omega |\nabla u| dx$$

of u. To stay in the framework sketched above one can approximate total variation based filtering with the penaliser family

$$\varphi(x) = 2\sqrt{\varepsilon^2 + x} - 2\varepsilon \qquad \text{for } \varepsilon > 0.$$

These functions are C^∞ and for $\varepsilon \longrightarrow 0$ they approximate the absolute value of the argument x. These approximations are discussed by Acar and Vogel in [1].

Perona and Malik suggested another penaliser in the context of nonlinear diffusion equations (see [20]) which reads as

$$\varphi(x) = \lambda^2 \ln\left(1 + \frac{x}{\lambda^2}\right) \qquad \text{for } \lambda > 0.$$

Usually several assumptions on the penaliser are made:

1. φ is differentiable and increasing ($\varphi' > 0$).
2. $\varphi(x^2)$ is convex in x. This is used to prove the existence of a minimiser. We note that this assumption is not satisfied for the Perona-Malik penaliser.
3. There are constants $c_1, c_2 > 0$ such that $c_1 x^2 < \varphi(x^2) < c_2 x^2$ for all x^2.

Under these assumptions some scale-space properties for regularisation methods can be shown: Schnörr (see [23]) has deduced well-posedness and regularity of the solution, average grey value invariance and the minimum-maximum principle. Scherzer and Weickert (see [22]) have further shown the Lyapunov property and convergence to the average grey value for $\alpha \longrightarrow \infty$. These scale-space properties are very similar to the properties of nonlinear diffusion filtering as presented in [31].

One should mention that contrast enhancement is only possible for nonconvex regularisers. In the diffusion filter framework one can motivate this with investigations of forward-backward diffusion:

Let us consider the one-dimensional nonlinear diffusion equation

$$u_t = \frac{d}{dx}\left(\varphi'\left(u_x^2\right)u_x\right).$$

We then are interested in the flux function

$$\Phi(u_x) = \varphi'\left(u_x^2\right)u_x$$

which allows us to rewrite the diffusion equation to

$$u_t = \Phi'(u_x)u_{xx}.$$

If $\Phi'(u_x) > 0$ within a region the nonlinear diffusion equation behaves like a scaled version of $u_t = u_{xx}$. In this regions the equation is well-posed, and we call the behaviour **forward diffusion**. In regions where $\Phi'(u_x) < 0$ the equation behaves like a scaled version of the ill-posed equation $u_t = -u_{xx}$. This so-called **backward diffusion** acts edge-enhancing.

We should note that among the penalisers presented above only the Perona-Malik penaliser allows backward diffusion. This can be an advantage in terms of edge-enhancement though it introduces some degree of ill-posedness. More information on this remarks can be found in [31], for example. In Chapter 3 we will carry over these considerations to second order filtering.

Backward diffusion is useful to enhance blurred edges. In regions with smooth grey value changes however it can reduce the quality of an image leading to the stair-casing effect.

We have shortly introduced first order variational methods. In the next chapter we start to carry over some essential parts of this theory to the higher order case.

Chapter 2

Continuous Energy Functionals

Variational methods for noise removal are based on the notion of minimisation of energy functionals. We imagine that unwanted features of an image (such as noise) result in high energy values. In this chapter we start with an energy functional

$$\mathcal{E}(u) = \int\limits_{\Omega} E\left(z, u(z), Du(z), \ldots, D^m u(z)\right) dz \tag{2.1}$$

and search for necessary and sufficient conditions for a function to be optimal with respect to this energy functional. We assume that $\Omega \subset \mathbb{R}^n$ is open and E is an integrable real-valued function.

First we summarise some results of calculus that are needed throughout this chapter and explain the basic procedure. For the treatment of the energy functionals we need some statements about parameter depending integrals and exchanging the order of integration and differentiation. We cite the following lemma from [30]:

Lemma 2.1 (Parameter Depending Integrals) *Let $n, m \in \mathbb{N} \setminus \{0\}$, $B \subset \mathbb{R}^n$ compact with boundary of measure 0, and $C \subset \mathbb{R}^m$. Consider integrals of the form*

$$F(y) = \int\limits_{B} f(x,y)g(x)dx$$

depending on the parameter $y \in C$. If C is open and if f and $\frac{\partial f}{\partial y_j}$ are continuous in $B \times C$ then $\frac{\partial F}{\partial y_j}$ is continuous and one can exchange integration and differentiation:

$$\frac{\partial F}{\partial y_j}(y) = \int\limits_{B} \frac{\partial f}{\partial y_j}(x,y)g(x)dx.$$

Even if we integrate formally over an unbounded set we deal with signals and images which typically have a bounded support. We usually have an interval or a rectangle as integration domain B.

The following lemma will allow us to write the necessary condition for a minimiser as a partial differential equation.

Lemma 2.2 (Fundamental Lemma of the Calculus of Variations) *Let f be a contiuous, real-valued function on some open set $\Omega \subset \mathbb{R}^n$, and suppose that*

$$\int\limits_{\Omega} f(x)\eta(x)dx = 0 \quad \textit{for all } \eta \in C_C^\infty(\Omega)$$

holds. Then we have

$$f(x) = 0 \quad \textit{for all } x \in \Omega.$$

Remark 2.3 Variants of this lemma with different proofs can be found in [9], [2], and [8]. Usually the lemma is proven by contradiction. The proof is primarily based on the fact that for continuous f a single nonzero point leads to an open set where the absolute value of f is greater than some positive bound. Then the space C_C^∞ is rich enough to contain a suitable function η which also has positive values in the open set, and the integral is not zero. The main difference between the different proofs cited above is the actual choice for η.

To reach the minimum of $\mathcal{E}$ we study the behaviour of $\mathcal{E}$ if we disturb the argument function u. Let $\eta \in C^m([a,b])$. Then we consider the function $\Psi : (-\varepsilon_0, \varepsilon_0) \longrightarrow \mathbb{R}$ defined by

$$\Psi(\varepsilon) \quad := \quad \mathcal{E}(u + \varepsilon\eta). \tag{2.2}$$

This function describes the growth of $\mathcal{E}$ in direction η. $\Psi'(0)$ can be considered as directional derivative of $\mathcal{E}$ in direction η: It describes the change of $\mathcal{E}$ at the point u in direction η.

Definition 2.4 (Variation) *We define the* ***first variation of*** $\mathcal{E}$ ***at the point*** u ***in direction*** η *as*

$$\delta\mathcal{E}(u,\eta) \quad := \quad \Psi'(0). \tag{2.3}$$

Higher variations are defined analogously as higher derivatives of Ψ *evaluated at* $\varepsilon = 0$*:*

$$\delta^k\mathcal{E}(u,\eta) \quad := \quad \Psi^{(k)}(0) \textit{ for } k \in \mathbb{N}. \tag{2.4}$$

For our purposes we will only use the first and the second variation. We should also specify the notion of a minimiser:

Definition 2.5 (Minimiser) *A function* $u \in C^m(\overline{\Omega})$ *is called a* ***minimiser*** *of the energy functional* $\mathcal{E}$ *if for all* $\eta \in C^m(\overline{\Omega})$ *there is an* $\varepsilon_0 > 0$ *such that for all* $0 < \varepsilon < \varepsilon_0$ *holds*

$$\mathcal{E}(u + \varepsilon\eta) > \mathcal{E}(u).$$

For fixed disturbance function $\eta \in C^m([a,b])$ the minimisation of $\mathcal{E}$ in direction η can be performed with differential calculus. We obtain from Taylor's formula that

$$\begin{aligned} \Psi(\varepsilon) &= \Psi(0) + \varepsilon\Psi'(0) + \frac{1}{2}\varepsilon^2\Psi''(0) + O(\varepsilon^3) \\ \Longleftrightarrow \quad \mathcal{E}(u+\varepsilon\eta) &= \mathcal{E}(u) + \varepsilon\ \delta\mathcal{E}(u,\eta) + \frac{1}{2}\varepsilon^2\ \delta^2\mathcal{E}(u,\eta) + O(\varepsilon^3). \end{aligned}$$

From differential calculus we immediately get the necessary conditions

$$\Psi'(0) = 0 \quad \text{and} \quad \Psi''(0) \geq 0$$

for Ψ to be minimal at the point 0. Sufficient conditions for a minimum are

$$\Psi'(0) = 0 \quad \text{and} \quad \Psi''(0) > 0.$$

We can also write the necessary and sufficient conditions in terms of the first and second variation as

Lemma 2.6 *For a minimiser u of the energy functional $\mathcal{E}$ the necessary conditions*

$$\delta\mathcal{E}(u,\eta) = 0 \quad \textit{and} \quad \delta^2\mathcal{E}(u,\eta) \geq 0 \quad \textit{for all } \eta \in C^m(\overline{\Omega})$$

hold. If a function u satisfies the conditions

$$\delta\mathcal{E}(u,\eta) = 0 \quad \textit{and} \quad \delta^2\mathcal{E}(u,\eta) > 0 \quad \textit{for all } \eta \in C^m(\overline{\Omega})$$

then it is a minimiser for $\mathcal{E}$.

2.1 The One-Dimensional Case

This section treats the one-dimensional case where u depends on one real variable. The signal processing methods introduced in later chapters will directly rely on the results presented here. Furthermore the general procedure with minimisation of functionals will be explained on the basis of this special case. The next section gives some generalisations on higher dimensions which we need for image processing algorithms.

2.1.1 The Problem

Let $a, b \in \mathbb{R}$, $a \neq b$. We consider a one-dimensional variational problem of order $m \in \mathbb{N} \setminus \{0\}$. That means we are searching a function $u \in C^m([a,b])$ such that the value of the integral

$$\mathcal{E}(u) \quad := \quad \int\limits_a^b E\left(x, u(x), u^{(1)}(x), u^{(2)}(x), \ldots, u^{(m)}(x)\right) dx \qquad (2.5)$$

is minimised. We assume that the integrand $E \in C^m(\mathbb{R}^{m+2})$ depends on $x, u(x)$ and the first m derivatives of the argument u evaluated at the point x. We write $E(x, u_0, u_1, \ldots, u_m)$ to denote the formal arguments of E so that E_{u_k} stands for the partial derivative of E with respect to the variable u_k for $k \in \{0, \ldots, m\}$. The actual arguments of E in the above expression are

$$\left(x, u(x), u^{(1)}(x), \ldots, u^{(m)}(x)\right).$$

We note that these arguments only depend on the evaluation point $x \in \mathbb{R}$, the function $u \in C^m(\mathbb{R})$ and the maximal derivative order $m \in \mathbb{N}$. To simplify the notations we write

$$[x, u, m] \quad := \quad \left(x, u(x), u^{(1)}(x), \ldots, u^{(m)}(x)\right)$$

for the arguments of E.

2.1.2 The Euler-Lagrange Equation

To obtain necessary conditions according to Lemma 2.6 we compute the first derivative of Ψ using Lemma 2.1:

$$\begin{aligned} \Psi'(\varepsilon) \quad &= \quad \frac{d}{d\varepsilon} \int\limits_a^b E\left([x, u + \varepsilon\eta, m]\right) dx \\ &= \quad \int\limits_a^b \frac{d}{d\varepsilon} E\left([x, u + \varepsilon\eta, m]\right) dx \end{aligned}$$

$$= \int\limits_a^b \sum_{k=0}^m E_{u_k}\left([x, u+\varepsilon\eta, m]\right) \frac{d}{d\varepsilon}(u+\varepsilon\eta)^{(k)}(x)dx. \tag{2.6}$$

One can simply compute the inner derivatives in equation (2.6) for all $k \in \{0, \ldots, m\}$ and all $x \in \mathbb{R}$:

$$\begin{aligned} \frac{d}{d\varepsilon}(u+\varepsilon\eta)^{(k)}(x) &= \frac{d}{d\varepsilon}\left(u^{(k)}(x) + \varepsilon\eta^{(k)}(x)\right) \\ &= \frac{d}{d\varepsilon}u^{(k)}(x) + \frac{d}{d\varepsilon}\varepsilon\eta^{(k)}(x) \\ &= \eta^{(k)}(x). \end{aligned} \tag{2.7}$$

We put this in equation (2.6) yielding

$$\Psi'(\varepsilon) = \int\limits_a^b \sum_{k=0}^m E_{u_k}\left([x, u+\varepsilon\eta, m]\right) \eta^{(k)}(x)dx. \tag{2.8}$$

Then we evaluate the derivative of Ψ at the point $\varepsilon = 0$ to obtain the first variation $\delta\mathcal{E}(u, \eta)$ according to definition (2.3):

$$\begin{aligned} \Psi'(0) &= \int\limits_a^b \sum_{k=0}^m E_{u_k}\left([x, u, m]\right) \eta^{(k)}(x)dx \\ &= \sum_{k=0}^m \int\limits_a^b E_{u_k}\left([\,x, u, m\,]\right) \eta^{(k)}(x)dx. \end{aligned} \tag{2.9}$$

At this point we want to use the Fundamental Lemma 2.2 of the calculus of variations. For each of the summands we need to integrate by parts k times so that the factor $\eta(x)$ is present in all of the summands. This yields

$$\begin{aligned} \int\limits_a^b E_{u_k}\left([x, u, m]\right) \eta^{(k)}(x)dx &= (-1)^k \int\limits_a^b \frac{d^k}{dx^k} E_{u_k}\left([x, u, m]\right) \eta(x)dx \\ &\quad + \sum_{l=0}^{k-1} (-1)^l \left[\frac{d^l}{dx^l} E_{u_k}\left([x, u, m]\right) \eta^{(k-1-l)}(x)\right]_a^b. \end{aligned}$$

We should keep in mind that in the case $k = 0$ we consider the sum as empty and evaluate it to zero. Further we use the notation

$$[f(x)]_a^b := f(b) - f(a)$$

for the boundary terms in the partial integrations. Replacing each of the summands in (2.9) according to this partial integration formula leads to

$$\begin{aligned} \delta\mathcal{E}(u, \eta) &= \sum_{k=0}^m \left(\sum_{l=0}^{k-1} (-1)^l \left[\frac{d^l}{dx^l} E_{u_k}\left([x, u, m]\right) \eta^{(k-1-l)}(x)\right]_a^b \right. \\ &\quad \left. + (-1)^k \int\limits_a^b \frac{d^k}{dx^k} E_{u_k}\left([x, u, m]\right) \eta(x)dx \right) \\ &= \sum_{k=0}^m \left(\sum_{l=0}^{k-1} (-1)^l \left[\frac{d^l}{dx^l} E_{u_k}\left([x, u, m]\right) \eta^{(k-1-l)}(x)\right]_a^b \right) \\ &\quad + \int\limits_a^b \left(\sum_{k=0}^m (-1)^k \frac{d^k}{dx^k} E_{u_k}\left([x, u, m]\right) \right) \eta(x)dx. \end{aligned} \tag{2.10}$$

We remember that we search u such that $\delta\mathcal{E}(u,\eta) = 0$ for all $\eta \in C^m([a,b])$. Since $C^m_C(a,b) \subset C^m([a,b])$ it is necessary for a minimiser u that the condition $\delta\mathcal{E}(u,\eta) = 0$ is satisfied for all $\eta \in C^m_C(a,b)$. For these testing functions with compact support in (a,b) all derivatives up to order m vanish at the boundary points a and b. So we may omit the boundary terms in equation (2.10). We see that for a minimiser u of E it holds necessarily for all $\eta \in C^m_C(a,b)$ that

$$0 \;=\; \delta\mathcal{E}(u,\eta) \;=\; \int\limits_a^b \left(\sum_{k=0}^m (-1)^k \frac{d^k}{dx^k} E_{u_k}\left([x,u,m]\right)\right) \eta(x)dx. \qquad (2.11)$$

As an application of the Fundamental Lemma 2.2 of the calculus of variations we note that the integral may only vanish for all testing functions $\eta \in C^m_C(a,b)$ if the equation

$$\sum_{k=0}^m (-1)^k \frac{d^k}{dx^k} E_{u_k}\left(x, u(x), u^{(1)}(x), \ldots, u^{(m)}(x)\right) \quad = \quad 0 \qquad (2.12)$$

holds for all $x \in (a,b)$. Equation (2.12) is called the **Euler-Lagrange equation** belonging to the energy functional (2.5). Since the Euler-Lagrange equation does not involve the testing function η we have found a necessary condition for a minimiser u. We formulate this result as

Proposition 2.7 (Euler-Lagrange Equation) *Each minimiser u of the energy functional (2.5) necessarily satisfies the Euler-Lagrange equation*

$$\sum_{k=0}^m (-1)^k \frac{d^k}{dx^k} E_{u_k}\left(x, u(x), u^{(1)}(x), \ldots, u^{(m)}(x)\right) = 0 \quad \textit{for all } x \in (a,b).$$

2.1.3 Natural Boundary Conditions

We have deduced the Euler-Lagrange equation by using testing functions $\eta \in C^m_C(a,b)$. Now we can also consider some $\eta \in C^m(a,b)$ for which the first m derivatives do not simultaneously vanish at the boundaries. Since the Euler-Lagrange equation (2.12) holds independent of η, the condition $\delta\mathcal{E}(u,\eta) = 0$ breaks down to

$$\sum_{k=0}^m \left(\sum_{l=0}^{k-1} (-1)^l \left[\frac{d^l}{dx^l} E_{u_k}\left([x,u,m]\right) \eta^{(k-1-l)}(x)\right]_a^b\right) \quad = \quad 0.$$

We introduce a new index variable j which stands for the derivative order of η and reorder the sum to get

$$\sum_{j=0}^{m-1} \sum_{k=j+1}^{m-1} (-1)^{k-1-j} \left[\left(\frac{d}{dx}\right)^{k-1-j} E_{u_k}\left([x,u,m]\right) \eta^{(j)}(x)\right]_a^b \quad = \quad 0. \qquad (2.13)$$

With Hermite interpolation we can generate for each $l \in \{0,\ldots,m-1\}$ a testing function η which suffices the conditions $\eta^{(j)}(a) = \delta_{jl}$ and $\eta^{(j)}(b) = 0$ for all $j \in \{0,\ldots,m\}$. The same can be done exchanging the roles of a and b. The sum in equation (2.13) simplifies to

$$\sum_{k=j+1}^m \left(-\frac{d}{dx}\right)^{k-1-j} E_{u_k}\left([x,u,m]\right) = 0 \qquad (2.14)$$

for all $j \in \{0,\ldots,m-1\}$ and for $x \in \{a,b\}$. This gives

Proposition 2.8 (Natural Boundary Conditions) *If no additional boundary conditions are imposed then the minimiser u necessarily satisfies*

$$\sum_{k=j}^{m} \left(-\frac{d}{dx}\right)^{k-j} E_{u_k}\left([x,u,m]\right) = 0 \tag{2.15}$$

for all $j \in \{1,\dots,m\}$ *and for* $x \in \{a,b\}$.

2.1.4 Sufficient Conditions

Until now we have only found necessary conditions for minimisers of energy functionals. We have found these necessary conditions by investigating zero points of the first variation. Let us now in analogy to the differential calculus consider the second variation and the condition

$$\delta^2 \mathcal{E}(u,\eta) \quad := \quad \Psi''(0) \quad > \quad 0 \qquad \text{for all } \eta \in C^m(a,b).$$

We compute the second derivative of Ψ at the point ε using the previous results for the first derivative in equation (2.8) as follows:

$$\begin{aligned}\Psi''(\varepsilon) &= \frac{d}{d\varepsilon}\int_a^b \sum_{k=0}^{m} E_{u_k}\left([x,u+\varepsilon\eta,m]\right)\eta^{(k)}(x)dx \\ &= \int_a^b \sum_{k=0}^{m} \frac{d}{d\varepsilon}E_{u_k}\left([x,u+\varepsilon\eta,m]\right)\eta^{(k)}(x)dx \\ &= \int_a^b \sum_{l,k=0}^{m} E_{u_k u_l}\left([x,u+\varepsilon\eta,m]\right)\eta^{(k)}(x)\eta^{(l)}(x)dx. \end{aligned} \tag{2.16}$$

We evaluate this second derivative at the point $\varepsilon = 0$ and get the second variation $\delta^2\mathcal{E}(u,\eta) = \Psi''(0)$. Searching for a sufficient condition for this to be positive we note that the integrand is a quadratic form. To make this clearer we introduce the matrix $\tilde{H}_E(x,u) \in \mathbb{R}^{(m+1)\times(m+1)}$ of all second derivatives of E to all variables except the space variable x,

$$\begin{aligned}\tilde{H}_E(x,u) &:= \left(E_{u_k u_l}\left([x,u,m]\right)\right)_{\substack{k=0,\dots,m\\ l=0,\dots,m}} \\ &= \begin{pmatrix} E_{u_0 u_0}\left([x,u,m]\right) & \dots & E_{u_0 u_m}\left([x,u,m]\right) \\ \vdots & \ddots & \vdots \\ E_{u_m u_0}\left([x,u,m]\right) & \dots & E_{u_m u_m}\left([x,u,m]\right) \end{pmatrix},\end{aligned}$$

and the vector $\tilde{\eta}(x) \in \mathbb{R}^{m+1}$ of all derivatives of η at the point x:

$$\tilde{\eta}(x) \quad := \quad \left(\eta(x), \eta^{(1)}(x), \dots, \eta^{(m)}(x)\right)^T.$$

We see that $\tilde{H}_E(x,u)$ can be seen as a submatrix of the Hessian of E: It is obtained by neglecting the first row and column in the Hessian which are related to derivatives according to the variable x. With these notations equation (2.16) can be written as

$$\delta^2 E(u,\eta) = \int_a^b \tilde{\eta}^T(x)\tilde{H}_E(x)\tilde{\eta}(x)dx.$$

With this formulation a sufficient condition for the positivity of $\delta^2\mathcal{E}(u,\eta)$ is obviously the pointwise positive definiteness of $\tilde{H}_E(x,u)$ for all $x \in (a,b)$.

Proposition 2.9 (Sufficient Conditions) *If the function u solves the Euler-Lagrange equation (2.12) with natural boundary conditions (2.15) and $\tilde{H}_E(x,u)$ is positive definite for all $x \in (a,b)$ then u is a minimiser for the energy functional $\mathcal{E}$.*

2.1.5 Applications

For our practical purposes we will only take energy functionals into account which can be separated with respect to different derivative orders. We will write a functional of order $m \in \mathbb{N}$. With the functions $\varphi_k \in C^m([a,b])$ for all $k \in \{1,\ldots,m\}$ we write our functional as

$$\mathcal{E}(u) = \int_a^b \left((u-f)^2 + \alpha \sum_{k=1}^m \varphi_k \left(\left(u^{(k)}(x) \right)^2 \right) \right) dx. \tag{2.17}$$

We would like to deduce the Euler-Lagrange equations and sufficient conditions.

The first partial derivative of the integrand E with respect to the variable u_k is

$$E_{u_k} \quad = \quad 2\varphi_k' \left(u_k^2\right) u_k \qquad \text{for all } k \in \{1,\ldots,m\}.$$

Together with $E_u = 2(u-f)$ we can put this in equation (2.12) to get the Euler-Lagrange equation

$$0 = u(x) - f(x) + \alpha \sum_{k=1}^m (-1)^k \frac{d^k}{dx^k} \left[\varphi_k' \left(\left(u^{(k)}(x) \right)^2 \right) u^{(k)}(x) \right] \tag{2.18}$$

for all $x \in (a,b)$. The corresponding natural boundary conditions according to Proposition 2.8 read as

$$\sum_{k=j}^m (-1)^{k-j} \frac{d^{k-j}}{dx^{k-j}} \left[\varphi_k' \left(\left(u^{(k)}(x) \right)^2 \right) u^{(k)}(x) \right] = 0$$

for $x \in \{a,b\}$ and all $j \in \{1,\ldots,m\}$. We also compute the second derivatives of E to check the conditions of Proposition 2.9; they are

$$\begin{aligned} E_{u_k u_k} &= 4\varphi_k'' \left(u_k^2\right) u_k^2 + 2\varphi_k' \left(u_k^2\right) \\ E_{u_k u_j} &= 0 \end{aligned}$$

for all $j,k \in \{1,\ldots,m\}$, $j \neq k$. The matrix $\tilde{H}_E(x)$ is therefore a diagonal matrix which is positive definite if all diagonal entries are positive. This is equivalent to the conditions

$$2\varphi_k'' \left(\left(u^{(k)}(x) \right)^2 \right) u^{(k)}(x) + \varphi_k' \left(u^{(k)}(x) \right) \quad > \quad 0$$

for all $k \in \{1,\ldots,m\}$. If all penaliser functions φ_k satisfy the conditions

$$2\varphi_k'' \left(x^2\right) x^2 + \varphi_k' \left(x^2\right) \quad > \quad 0$$

for all $x \in \mathbb{R}$ each solution of the Euler-Lagrange equation is a minimiser of the energy functional $\mathcal{E}$.

To conclude the section on one-dimensional variational problems we would like to give two examples that are used for signal processing in later chapters. These examples are special cases of the functional (2.17).

Example 2.10 Let us choose a derivative order $m \in \mathbb{N} \setminus \{0\}$ and consider the nonlinear energy functional which only depends on the mth derivative of u:

$$\mathcal{E}(u) = \int\limits_a^b \left((u-f)^2 + \alpha\varphi\left(\left(u^{(m)}\right)^2\right)\right) dx. \tag{2.19}$$

If u is a minimiser of $\mathcal{E}$ it satisfies the Euler-Lagrange equation

$$0 = u - f + \alpha(-1)^m \frac{d^m}{dx^m}\left(\varphi'\left(\left(u^{(m)}\right)^2\right)u^{(m)}\right) \tag{2.20}$$

with the natural boundary conditions

$$\frac{d^k}{dx^k}\left(\varphi'\left(\left(u^{(m)}\right)^2\right)u^{(m)}\right) = 0 \quad \text{for } k \in \{0,\dots,m-1\}.$$

For $m > 1$ we have $E_{u_1} = 0$ and thus $\tilde{H}_E$ is not positive definite (since it has one row and column filled with zero entries). In this case Proposition 2.9 is to weak to obtain sufficient conditions.

Example 2.11 Let us consider a nonlinear energy functional which only depends on the first two derivatives:

$$\mathcal{E}(u) = \int\limits_a^b \left((u-f)^2 + \alpha\left[\varphi_1\left(\left(u^{(1)}\right)^2\right) + \varphi_2\left(\left(u^{(2)}\right)^2\right)\right]\right) dx.$$

The Euler-Lagrange equation for this functional is

$$0 = u - f + \alpha\left[-\frac{d}{dx}\left(\varphi_1'\left(\left(u^{(1)}\right)^2\right)u^{(1)}\right) + \frac{d^2}{dx^2}\left(\varphi_2'\left(\left(u^{(2)}\right)^2\right)u^{(2)}\right)\right]$$

for $x \in (a,b)$ with the natural boundary conditions

$$\begin{aligned} \varphi_1'\left(\left(u^{(1)}\right)^2 u^{(1)}\right) - \frac{d}{dx}\varphi_2'\left(\left(u^{(2)}\right)^2\right) &= 0, \\ \varphi_2'\left(\left(u^{(2)}\right)^2 u^{(2)}\right) &= 0 \end{aligned}$$

for $x \in \{a,b\}$. In this case we have the sufficient conditions

$$\begin{aligned} \varphi_1''(x^2)x^2 + \varphi_1'(x^2) &> 0, \\ \varphi_2''(x^2)x^2 + \varphi_2'(x^2) &> 0. \end{aligned}$$

2.2 The Two-Dimensional Case

2.2.1 The Problem

Let $\Omega \subseteq \mathbb{R}^2$ be an open subset of $\mathbb{R}^2$ with piecewise smooth boundary $\partial\Omega$ and $m \in \mathbb{N}$. We consider the minimisation of the functional

$$\mathcal{E}(u) = \int\limits_\Omega E\left(z, u(z), Du(z), D^2u(z), \dots, D^m u(z)\right) dz$$

where $D^k u(z)$ denotes the k-th differential of the function u at the point x. Our energy functional may depend on all partial derivatives of u up to order m. Again we simplify the notations by introducing the abbreviation

$$[z,u,m]_p \quad := \quad \big(z,u(z),Du(z),D^2u(z),\dots,D^m u(z)\big)$$

to express that E may depend on all partial derivatives up to order m of u evaluated at the point z. We should add that we restrict ourselves to the case $\Omega \subset \mathbb{R}^2$ mainly for simplicity reasons with treating the natural boundary conditions. The computations in general also work in higher dimensions. Numerical tests will be performed up to dimension 2.

2.2.2 The Euler-Lagrange Equation

Again we start with the condition $\delta\mathcal{E}(u,\eta) = 0$ given in Lemma 2.6 and compute the first variation $\delta\mathcal{E}(u,\eta) = \Psi'(0)$ using Lemma 2.1:

$$\begin{aligned}\Psi'(\varepsilon) &= \int\limits_\Omega \frac{d}{d\varepsilon} E\left([z,u+\varepsilon\eta,m]_p\right) dz \\ &= \int\limits_\Omega \sum_{k=0}^m \sum_{|\alpha|=k} \frac{k!}{\alpha!} E_{u_\alpha}\left([z,u+\varepsilon\eta,m]_p\right)\, D^\alpha\eta(z)dz \\ &= \sum_{k=0}^m \sum_{|\alpha|=k} \frac{k!}{\alpha!} \int\limits_\Omega E_{u_\alpha}\left([z,u+\varepsilon\eta,m]_p\right)\, D^\alpha\eta(z)dz. \end{aligned} \tag{2.21}$$

We evaluate the derivative Ψ' at the point $\varepsilon = 0$ and we write down the first variation as

$$\delta\mathcal{E}(u,\eta) \quad = \quad \sum_{k=0}^m \sum_{|\alpha|=k} \frac{k!}{\alpha!} \int\limits_\Omega E_{u_\alpha}\left([z,u,m]_p\right)\, D^\alpha\eta(z)dz. \tag{2.22}$$

To derive the Euler-Lagrange equation we assume that $\eta \in C_C(\Omega)$. In this case we can integrate by parts each summand of equation (2.22) without introducing integrals over the boundary of Ω. For one summand this integration yields

$$\begin{aligned}&\int\limits_\Omega E_{u_\alpha}\left([z,u,m]_p\right) D^\alpha\eta(z)dz \\ &\quad = \quad (-1)^{|\alpha|} \int\limits_\Omega D^\alpha E_{u_\alpha}\left([z,u,m]_p\right)\eta(z)dz.\end{aligned} \tag{2.23}$$

We rewrite equation (2.22) using (2.23) for each summand and exchange the order of sums and integration to get the formula

$$\delta\mathcal{E}(u,\eta) \quad = \quad \int\limits_\Omega \left(\sum_{k=0}^m (-1)^k \sum_{|\alpha|=k} \frac{k!}{\alpha!} D^\alpha E_{u_\alpha}\left([z,u,m]_p\right)\right)\eta(z)dz. \tag{2.24}$$

The Fundamental Lemma 2.2 of the calculus of variations assures the following proposition:

Proposition 2.12 (Euler-Lagrange Equation) *For a minimiser u of the energy functional $\mathcal{E}$ the Euler-Lagrange equation*

$$\sum_{k=0}^m (-1)^k \sum_{|\alpha|=k} \frac{k!}{\alpha!} D^\alpha E_{u_\alpha}\left(z,u(z),Du(z),\dots,D^m u(z)\right) \quad = \quad 0 \tag{2.25}$$

holds for all $z \in \Omega$.

2.2.3 Natural Boundary Conditions

As it is the most important case for our applications we consider the open set $\Omega \subset \mathbb{R}^2$ and the highest derivative order $m = 2$. In the 2D-case we will see in the numerical tests that using Neumann boundary conditions may lead to better results than the natural boundary conditions. We will deduce natural boundary conditions only for the special case given above since it is the most relevant case for our numerical examples. With these assumptions our energy functional has the form

$$\mathcal{E}(u) = \int\limits_{\Omega} E(z, u(z), \nabla u(z), H_u(z))dz \tag{2.26}$$

where z denotes the point $z = (x, y) \in \Omega$ and

$$\nabla u(z) := (u_x(z), u_y(z))^T \quad \text{and} \quad H_u(z) := \begin{pmatrix} u_{xx}(z) & u_{xy}(z) \\ u_{yx}(z) & u_{yy}(z) \end{pmatrix}.$$

In this special case we can rewrite equation (2.22) with respect to the particular partial derivatives as

$$\begin{aligned} \delta\mathcal{E}(u,\eta) &= \sum_{k=0}^{2} \sum_{|\alpha|=k} \frac{k!}{\alpha!} \int\limits_{\Omega} E_{u_\alpha}\left(z, u(z), \nabla u(z), H_u(z)\right) D^\alpha \eta(z) dz \\ &= \int\limits_{\Omega} E_u \big(\eta + E_{u_x}\eta_x + E_{u_y}\eta_y + E_{u_{xx}}\eta_{xx} + E_{u_{xy}}\eta_{xy} \\ &\qquad + E_{u_{yx}}\eta_{yx} + E_{u_{yy}}\eta_{yy}\big)\, dz \end{aligned} \tag{2.27}$$

We have omitted the arguments $(z, u(z), u_x(z), \ldots, u_{yy}(z))$ from all derivatives of E and (z) from all derivatives of η to simplify the notation. In opposite to the last section boundary integrals appear when we integrate formula (2.27) by parts:

$$\begin{aligned} \delta\mathcal{E}(u,\eta) = & \int\limits_{\Omega} E_u \eta dz \\ & - \int\limits_{\Omega} \partial_x E_{u_x} \eta dz + \int\limits_{\partial\Omega} E_{u_x} \eta \nu_x dz \\ & - \int\limits_{\Omega} \partial_y E_{u_y} \eta dz + \int\limits_{\partial\Omega} E_{u_y} \eta \nu_y dz \\ & - \int\limits_{\Omega} \partial_x E_{u_{xx}} \eta_x dz + \int\limits_{\partial\Omega} E_{u_{xx}} \eta_x \nu_x dz \\ & - \int\limits_{\Omega} \partial_y E_{u_{xy}} \eta_x dz + \int\limits_{\partial\Omega} E_{u_{xy}} \eta_x \nu_y dz \\ & - \int\limits_{\Omega} \partial_x E_{u_{yx}} \eta_x dz + \int\limits_{\partial\Omega} E_{u_{yx}} \eta_y \nu_x dz \\ & - \int\limits_{\Omega} \partial_y E_{u_{yy}} \eta_y dz + \int\limits_{\partial\Omega} E_{u_{yy}} \eta_y \nu_y dz. \end{aligned} \tag{2.28}$$

For $z \in \partial\Omega$ the outer normal is denoted with $\nu(z)$. A second partial integration removes all derivatives of η in integrals over the whole set Ω:

$$\begin{aligned}
\delta\mathcal{E}(u,\eta) \quad = \quad & \int\limits_{\Omega} \big(E_u - \partial_x E_{u_x} - \partial_y E_{u_y} + \partial_{xx} E_{u_{xx}} \\
& \qquad + \partial_{xy} E_{u_{yx}} + \partial_{yx} E_{u_{xy}} + \partial_{yy} E_{u_{yy}}\big)\, \eta dz \\
& + \int\limits_{\partial\Omega} \big(E_{u_x} - \partial_x E_{u_{xx}} - \partial_y E_{u_{xy}}\big)\, \nu_x \eta dz \\
& + \int\limits_{\partial\Omega} \big(E_{u_y} - \partial_x E_{u_{yx}} - \partial_y E_{u_{yy}}\big)\, \nu_y \eta dz \\
& + \int\limits_{\partial\Omega} \big(E_{u_{xx}} \nu_x + E_{u_{xy}} \nu_y\big)\, \eta_x dz \\
& + \int\limits_{\partial\Omega} \big(E_{u_{yx}} \nu_x + E_{u_{yy}} \nu_y\big)\, \eta_y dz.
\end{aligned} \tag{2.29}$$

As we have already seen in the last section, together with the Fundamental Lemma 2.2, equation (2.29) can be used to deduce the Euler-Lagrange equation which in the general 2D second order setting reads as

$$0 = E_u - \partial_x E_{u_x} - \partial_y E_{u_y} + \partial_{xx} E_{u_{xx}} + \partial_{xy} E_{u_{yx}} + \partial_{yx} E_{u_{xy}} + \partial_{yy} E_{u_{yy}}. \tag{2.30}$$

This result coincides with the formula derived in the last section.

One can surely choose test functions $\eta \in C^2(\overline{\Omega})$ with the properties $\eta_{|\partial\Omega} \neq 0$ and $\nabla\eta_{|\partial\Omega} = 0$ since each nonzero constant function satisfies this. We claim that $\delta\mathcal{E}(u,\eta) = 0$ for such η. We start with equation (2.29) and remind that equation (2.30) holds. Together with $\nabla\eta_{|\partial\Omega} = 0$ we see that in this case

$$\begin{aligned}
\delta\mathcal{E}(u,\eta) \quad = \quad & \int\limits_{\partial\Omega} \big(E_{u_x} - \partial_x E_{u_{xx}} - \partial_y E_{u_{xy}}\big)\, \nu_x \eta dz \\
& + \int\limits_{\partial\Omega} \big(E_{u_y} - \partial_x E_{u_{yx}} - \partial_y E_{u_{yy}}\big)\, \nu_y \eta dz.
\end{aligned} \tag{2.31}$$

The Fundamental Lemma 2.2 of the calculus of variations tells us that

$$\big(E_{u_x} - \partial_x E_{u_{xx}} - \partial_y E_{u_{xy}}\big)\, \nu_x + \big(E_{u_y} - \partial_x E_{u_{yx}} - \partial_y E_{u_{yy}}\big)\, \nu_y = 0 \tag{2.32}$$

holds for all $z \in \partial\Omega$.

Now we consider the special case when $\Omega = (a,b) \times (c,d)$ is a rectangular subset of $\mathbb{R}^2$. This seems to be a severe restriction at first sight. For our main purpose, the analysis of 2D images this assumption is realistic. For rectangular Ω we can surely find a function $\eta \in C^2(\overline{\Omega})$ with $\eta_{|\partial\Omega} = \eta_{y_{|\partial\Omega}} = 0$ and $\eta_{x_{|\partial\Omega}} \neq 0$. The same reasoning as above yields

$$E_{u_{xx}} \nu_x + E_{u_{xy}} \nu_y = 0 \quad \text{ for all } z \in \partial\Omega. \tag{2.33}$$

Again we can exchange the roles of x and y in our assumptions about the function η: If we are choosing $\eta \in C^2(\overline{\Omega})$ with $\eta_{|\partial\Omega} = \eta_{x_{|\partial\Omega}} = 0$ and $\eta_{y_{|\partial\Omega}} \neq 0$ we see that

$$E_{u_{yx}} \nu_x + E_{u_{yy}} \nu_y = 0 \quad \text{ for all } z \in \partial\Omega. \tag{2.34}$$

We summarise these conditions in the following

Proposition 2.13 (Natural Boundary Conditions) *Let $\Omega = (a,b) \times (c,d)$, and assume that u is a minimiser of the functional (2.26). Then u satisfies the equation*

(2.30) with the natural boundary conditions

$$\begin{aligned}\left(E_{u_x} - \partial_x E_{u_{xx}} - \partial_y E_{u_{xy}}\right)\nu_x + \left(E_{u_y} - \partial_x E_{u_{yx}} - \partial_y E_{u_{yy}}\right)\nu_y &= 0\\ E_{u_{xx}}\nu_x + E_{u_{xy}}\nu_y &= 0\\ E_{u_{yx}}\nu_x + E_{u_{yy}}\nu_y &= 0\end{aligned}$$

for all $z \in \partial\Omega$.

2.2.4 Sufficient Conditions

For the sufficient conditions we again allow derivatives up to order m. Starting with equation (2.21) for $\Psi'(\varepsilon)$ we can simply compute

$$\begin{aligned}\Psi''(\varepsilon) &= \frac{d}{d\varepsilon}\sum_{k=0}^{m}\sum_{|\alpha|=k}\frac{k!}{\alpha!}\int\limits_{\Omega} E_{u_\alpha}\left(\left[z, u+\varepsilon\eta, m\right]_p\right) D^\alpha\eta(z)dz\\ &= \sum_{k,l=0}^{m}\sum_{|\alpha|=k}\sum_{|\beta|=l}\frac{k!}{\alpha!}\frac{l!}{\beta!}\int\limits_{\Omega} E_{u_\alpha u_\beta}\left(\left[z, u+\varepsilon\eta, m\right]_p\right) D^\alpha\eta(z)D^\beta\eta(z)dz\\ &= \int\limits_{\Omega}\sum_{k,l=0}^{m}\sum_{|\alpha|=k}\sum_{|\beta|=l}\frac{k!}{\alpha!}\frac{l!}{\beta!}E_{u_\alpha u_\beta}\left(\left[z, u+\varepsilon\eta, m\right]_p\right) D^\alpha\eta(z)D^\beta\eta(z)dz.\end{aligned}$$

Again we observe that the integrand can be written as a quadratic form. We define the matrix $\tilde{E}$ analogue to the one-dimensional case. In the higher dimensional case the matrix entries are weighted second derivatives of E due to the multiplicity of the multiindices. We have

$$\tilde{E}(z) := \left(\frac{k!}{\alpha!}\frac{l!}{\beta!}E_{u_\alpha u_\beta}\left(\left[z, u, m\right]_p\right)\right)_{\substack{k,l=1,\ldots,m\\ |\alpha|=k,|\beta|=l}}.$$

The pointwise positive definiteness of $\tilde{E}$ is a sufficient condition for u satisfying the Euler-Lagrange equation to be a minimiser of $\mathcal{E}$. In the next section we will see that our functionals usually do not depend on all partial derivatives up to the given order. The conditions derived here are normally to weak to obtain a statement in this case.

2.2.5 Applications

In the one-dimensional case we could simply use the square of the second derivative of u as argument of the regulariser to write an energy functional of the form

$$\mathcal{E}(u) = \int\limits_a^b \left((u-f)^2 + \alpha\varphi\left(u_{xx}^2\right)\right)dx.$$

In a higher dimensional setting it is not clear what to choose as an appropriate expression for the square of the second derivative. Since the equivalent to the second derivative is the Hessian matrix in this case there are several terms which can be taken into consideration.

If we require rotational invariance we can reduce the number of reasonable choices. Interesting rotational invariant expressions are for example the trace of the Hessian, which is the Laplacian of u, the Frobenius norm or the determinant of the Hessian. We are going to consider general nonlinear energy functionals depending on these three expressions and deduce the corresponding Euler-Lagrange equations.

Example 2.14 (Laplacian) At first let us consider a regulariser depending on

$$\begin{aligned}(\Delta u)^2 &= (u_{xx}+u_{yy})^2\\ &= u_{xx}^2+2u_{xx}u_{yy}+u_{yy}^2\end{aligned}$$

given by

$$\mathcal{E}(u)=\int\limits_\Omega \left((u-f)^2+\alpha\,\varphi\left((\Delta u)^2\right)\right)dz. \tag{2.35}$$

The integrand E does not depend on u_x, u_y and the mixed second derivatives of u, that means $E_{u_x}=E_{u_y}=E_{u_{xy}}=E_{u_{yx}}=0$. The remaining partial derivatives of E are

$$\begin{aligned}E_u &= 2(u-f)\\ E_{u_{xx}} &= 2\alpha\,\varphi'\left((\Delta u)^2\right)\Delta u\\ E_{u_{yy}} &= 2\alpha\,\varphi'\left((\Delta u)^2\right)\Delta u.\end{aligned}$$

With these we can write down the Euler-Lagrange equation

$$\begin{aligned} 0 &= 2(u-f)\ +\alpha\,\partial_{xx}\left(2\varphi'\left((\Delta u)^2\right)\Delta u\right)\\ &\qquad\qquad +\alpha\,\partial_{yy}\left(2\varphi'\left((\Delta u)^2\right)\Delta u\right)\\ \Longleftrightarrow\qquad 0 &= u-f+\alpha\,\Delta\left(\varphi'\left((\Delta u)^2\right)\Delta u\right). \end{aligned}\tag{2.36}$$

We note that in the linear case $\varphi(x)=cx$ this equation reads as

$$0=u-f+\alpha c\,\Delta^2 u.$$

We would like to derive the boundary conditions also for the linear case. Similar to Section 2.2.3 we assume that $\Omega=(0,1)^2\subset\mathbb{R}^2$. Equation (2.32) yields in this case for all $z\in\partial\Omega$

$$\begin{aligned} 0 &= -\alpha c\left((\partial_x\Delta u)\nu_x+(\partial_y\Delta u)\nu_y\right)\\ \Longleftrightarrow\qquad 0 &= (u_{xxx}+u_{xyy})\,\nu_x+(u_{xxy}+u_{yyy})\,\nu_y. \end{aligned}$$

With our special choice of Ω we get the conditions

$$\begin{aligned} u_{xxx}+u_{xyy}=0 &\quad \text{for all } z\in\{(0,y)|y\in(0,1)\}\cup\{(1,y)|y\in(0,1)\}\\ u_{xxy}+u_{yyy}=0 &\quad \text{for all } z\in\{(x,0)|x\in(0,1)\}\cup\{(x,1)|x\in(0,1)\}. \end{aligned}$$

Equation (2.33) is $\Delta u\nu_x=0$ for all $z\in\partial\Omega$. This can be formulated as the condition

$$\Delta u=0 \text{ for all } z\in\{(x,0)|x\in(0,1)\}\cup\{(x,1)|x\in(0,1)\}.$$

Similarly we can derive

$$\Delta u=0 \text{ for all } z\in\{(0,y)|y\in(0,1)\}\cup\{(1,y)|y\in(0,1)\}$$

from equation (2.34). We conclude that

$$\Delta u=0 \text{ in the set } \partial\Omega\setminus\{(0,0),(0,1),(1,0),(1,1)\}.$$

Example 2.15 (Frobenius Norm of the Hessian) We do the same computations as above for the squared Frobenius norm of the Hessian

$$\|H_u\|_F^2=u_{xx}^2+u_{xy}^2+u_{yx}^2+u_{yy}^2.$$

The energy functional is then

$$\mathcal{E}(u)=\int\limits_\Omega \left((u-f)^2+\alpha\varphi\left(\|H_u\|_F^2\right)\right)dz. \tag{2.37}$$

In this case the integrand E does not depend on the first-order partial derivatives of u, that means $E_{u_x} = E_{u_y} = 0$. We compute the other partial derivatives of the integrand

$$\begin{aligned} E_u &= 2(u-f) \\ E_{u_{xx}} &= \varphi'\left(\|H_u\|_F^2\right) 2u_{xx} \\ E_{u_{xy}} &= \varphi'\left(\|H_u\|_F^2\right) 2u_{xy} \\ E_{u_{yx}} &= \varphi'\left(\|H_u\|_F^2\right) 2u_{yx} \\ E_{u_{yy}} &= \varphi'\left(\|H_u\|_F^2\right) 2u_{yy} \end{aligned}$$

and write down the Euler-Lagrange equation (in the case $\alpha \neq 0$) as

$$\begin{aligned} \frac{u-f}{\alpha} &= -\partial_{xx}\left(\varphi'\left(\|H_u\|_F^2\right) u_{xx}\right) - \partial_{xy}\left(\varphi'\left(\|H_u\|_F^2\right) u_{xy}\right) \\ &\quad -\partial_{yx}\left(\varphi'\left(\|H_u\|_F^2\right) u_{yx}\right) - \partial_{yy}\left(\varphi'\left(\|H_u\|_F^2\right) u_{yy}\right). \end{aligned}$$

Again we consider the linear case $\varphi(x) = cx$ and the according Euler-Lagrange equation

$$\begin{aligned} 0 &= u - f + \alpha c\left(\partial_{xx}u_{xx} + \partial_{xy}u_{xy} + \partial_{yx}u_{yx} + \partial_{yy}u_{yy}\right) \\ &= u - f + \alpha c\left(u_{xxxx} + 2u_{xxyy} + u_{yyyy}\right) \\ &= u - f + \alpha c\Delta^2 u. \end{aligned}$$

We note that in the linear case the Euler-Lagrange equations of Example 2.14 and Example 2.15 coincide. We derive the boundary conditions for $\Omega = (0,1)^2$. Equation (2.32) yields

$$0 = -\alpha c\left((u_{xxx} + u_{xyy})\nu_x + (u_{xxy} + u_{yyy})\nu_y\right)$$

which in our case leads to the conditions

$$\begin{aligned} u_{xxx} + u_{xyy} = 0 &\quad \text{for all } z \in \{(0,y)|y \in (0,1)\} \cup \{(1,y)|y \in (0,1)\} \\ u_{xxy} + u_{yyy} = 0 &\quad \text{for all } z \in \{(x,0)|x \in (0,1)\} \cup \{(x,1)|x \in (0,1)\}. \end{aligned}$$

Equation (2.33) yields

$$\begin{aligned} u_{xx} = 0 &\quad \text{for all } z \in \{(0,y)|y \in (0,1)\} \cup \{(1,y)|y \in (0,1)\} \\ u_{xy} = 0 &\quad \text{for all } z \in \{(x,0)|x \in (0,1)\} \cup \{(x,1)|x \in (0,1)\}. \end{aligned}$$

Equation (2.34) yields

$$\begin{aligned} u_{xy} = 0 &\quad \text{for all } z \in \{(0,y)|y \in (0,1)\} \cup \{(1,y)|y \in (0,1)\} \\ u_{yy} = 0 &\quad \text{for all } z \in \{(x,0)|x \in (0,1)\} \cup \{(x,1)|x \in (0,1)\}. \end{aligned}$$

We see that the boundary conditions are not the same as in the last example.

For linear φ and if we only consider functions vanishing at the boundary we get the same minimisers. This result is also mentioned in [18] where a simple proof in the Fourier domain is given. They also note that partial integration leads to the same result.

Lemma 2.16 *In the linear case $\varphi(x) = cx$ and if $u \in C^3(\Omega)$ is assumed the energy functionals (2.35) and (2.37) yield the same minimisers.*

Proof: First we note that for this proof we need further smoothness assumptions: It should be $u \in C^3(\overline{\Omega})$ since we need a third derivative during the computation. We compute by partial integration

$$\int\limits_\Omega u_{xy}^2 dz = \int\limits_\Omega u_{xy}u_{yx} dz$$

$$\begin{aligned} &= -\int\limits_{\Omega} u_{xxy}u_y dz + \int\limits_{\partial\Omega} u_{xy}u_y\nu_x dz \\ &= \int\limits_{\Omega} u_{xx}u_{yy}dz - \int\limits_{\partial\Omega} u_{xx}u_y\nu_y dz + \int\limits_{\partial\Omega} u_{xy}u_y\nu_x dz. \end{aligned}$$

For $u \in C_C^3(\Omega)$ it holds that $u_{xx} = u_{xy} = u_y = u = 0$ on the boundary $\partial\Omega$. We can leave the boundary integrals out and get

$$\int\limits_{\Omega} u_{xy}^2 dz \quad = \quad \int\limits_{\Omega} u_{xx}u_{yy}dz$$

which leads us to the equation

$$\begin{aligned} \int\limits_{\Omega} \|H_u(z)\|_F^2 dz &= \int\limits_{\Omega} \left(u_{xx}^2 + 2u_{xy}^2 + u_{yy}^2\right) dz \\ &= \int\limits_{\Omega} \left(u_{xx}^2 + 2u_{xx}u_{yy} + u_{yy}^2\right) dz \\ &= \int\limits_{\Omega} (\Delta u)^2 dz. \end{aligned}$$

This shows that the functionals have the same value and thus have the same minimisers. □

Example 2.17 (Determinant of the Hessian) As a last example of second order in 2D we will consider energy functionals depending on the squared determinant of the Hessian:

$$\begin{aligned} \det{}^2(H_u) &= (u_{xx}u_{yy} - u_{xy}u_{yx})^2 \\ &= u_{xx}^2u_{yy}^2 - 2u_{xx}u_{yy}u_{xy}u_{yx} + u_{xy}^2u_{yx}^2. \end{aligned}$$

These can be written in the form

$$\mathcal{E}(u) = \int\limits_{\Omega} \left((u-f)^2 + \alpha\varphi\left(\det{}^2(H_u(z))\right)\right) dz.$$

We get the partial derivatives of the integrand as

$$\begin{aligned} E_u &= 2(u-f) \\ E_{u_x} &= 0 \\ E_{u_y} &= 0 \\ E_{u_{xx}} &= 2\alpha\,\varphi'\left(\det{}^2(H_u(z))\right)\det(H_u(z))u_{yy} \\ E_{u_{xy}} &= -2\alpha\,\varphi'\left(\det{}^2(H_u(z))\right)\det(H_u(z))u_{yx} \\ E_{u_{yx}} &= -2\alpha\,\varphi'\left(\det{}^2(H_u(z))\right)\det(H_u(z))u_{xy} \\ E_{u_{yy}} &= 2\alpha\,\varphi'\left(\det{}^2(H_u(z))\right)\det(H_u(z))u_{xx}. \end{aligned}$$

With these we can write down the Euler-Lagrange equation

$$\begin{aligned} 0 \quad = \quad u - f + \alpha\ [\quad & \partial_{xx}\left(\varphi'\left(\det{}^2(H_u(z))\right)\det(H_u(z))u_{yy}\right) \\ - \quad & \partial_{xy}\left(\varphi'\left(\det{}^2(H_u(z))\right)\det(H_u(z))u_{yx}\right) \\ - \quad & \partial_{yx}\left(\varphi'\left(\det{}^2(H_u(z))\right)\det(H_u(z))u_{xy}\right) \\ + \quad & \partial_{yy}\left(\varphi'\left(\det{}^2(H_u(z))\right)\det(H_u(z))u_{xx}\right) \quad]. \end{aligned}$$

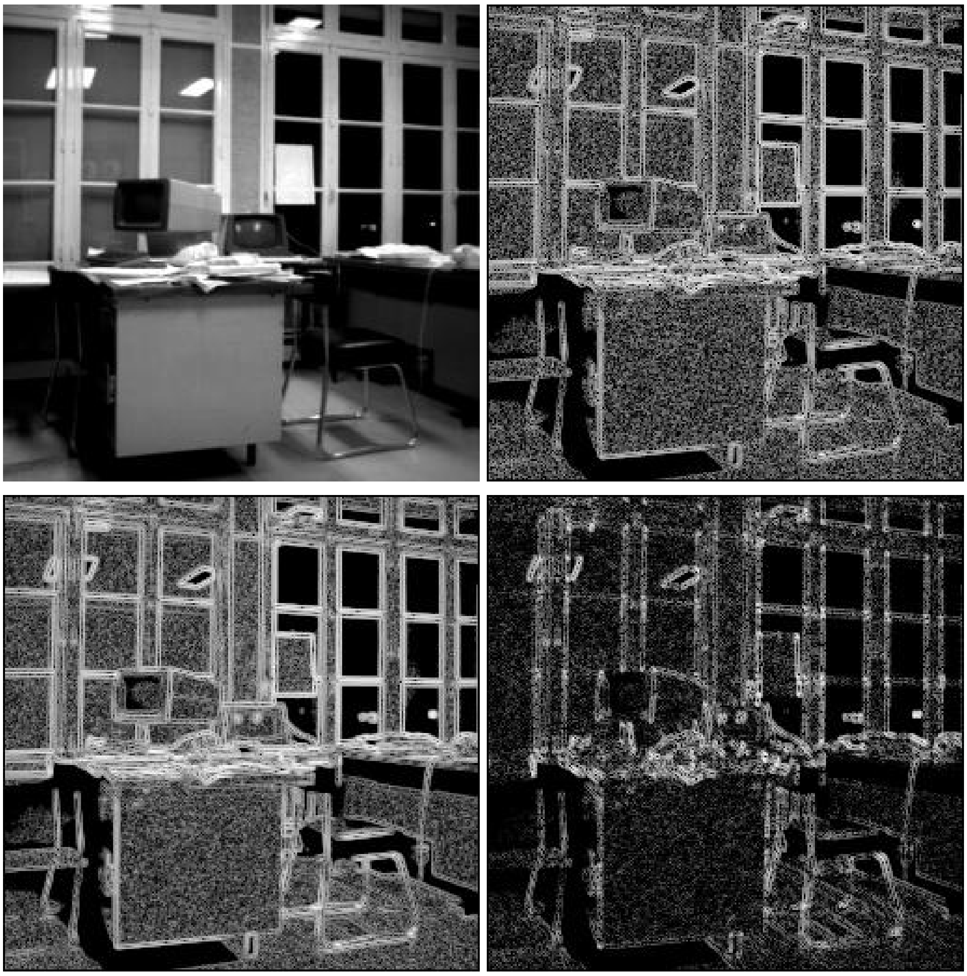

Figure 2.1: Top left: Original image u, Top right: $(\Delta u)^2$, Bottom left: $\|H_u\|_F^2$, Bottom right: $\det^2(H_u)$. The derivatives were approximated with finite differences. The logarithm of the grey values was taken and rescaled to the range 0 to 255 for better visualisation.

We have seen three general types of nonlinear energy functionals in two dimensions which can be used as starting point to derive image processing methods. To conclude this chapter figure 2.1 gives a visual impression of the behaviour of this three expressions in a real image. It seems that Laplacian and Frobenius norm of the Hessian yield similar results penalising mainly near edges. The determinant of the Hessian seems to emphasise rather corners and junctions. Thus for most of the applications the Laplacian and the Hessian will be more interesting. We will prefer to use the Laplacian for our image restoration methods for simplicity reasons. Nevertheless it would be interesting to investigate how filtering methods based on the other two rotationally invariant expressions behave in practice. It should also be mentioned that the Frobenius norm of the Hessian could be easily generalised to higher derivative orders as the sum of the squares of all partial derivatives with a certain order.

Chapter 3

Continuous Filtering

After giving conditions for a minimiser this section will introduce methods that use these conditions to find one. In our applications a minimiser of an energy functional is a signal or an image which should be in some way better than the initial one. Thus we consider the resulting image as a filtered version of the initial one and we will use the notion filtering for the process of minimising the energy functional. We are interested in the continuous framework in this chapter. Discretisations of these methods and the application to concrete problem instances will be described in the next two chapters.

3.1 Filtering Approaches

As a starting point we discuss different approaches at the example of an energy functional that depends on the weighted squares of the derivatives of u. We consider a one-dimensional linear energy functional of higher order of the form

$$E(u) \quad = \quad \int\limits_{\mathbb{R}} \left((u-f)^2 + \alpha \sum_{k=1}^{m} \lambda_k \left(\frac{d^k}{dx^k} u \right)^2 \right) dx \tag{3.1}$$

with $m \in \mathbb{N}$, $\lambda_k \in \mathbb{R}$ for all $k \in \{1, \ldots, m\}$ and $\alpha > 0$. We integrate over the whole set $\mathbb{R}$ in this case to avoid the influence of boundary conditions.

3.1.1 Direct Minimisation of the Energy Functional

The first approach reformulates the energy functional in the Fourier domain. This strategy is discussed in [18]. Since we use the Fourier transform to simplify the energy functional this is only applicable for linear penalisers. We can rewrite the functional (3.1) using the norm in $L_2(\mathbb{R})$ and the linearity of the integral and get

$$E(u) \quad = \quad \|u-f\|_2^2 + \alpha \sum_{k=1}^{m} \lambda_k \left\| \frac{d^k}{dx^k} u \right\|_2^2 . \tag{3.2}$$

We then apply the Fourier transform to the energy functional. In the following $\hat{f}$ denotes the Fourier-Plancherel transform of the function f. For details on this transform we refer to [33]. With the Plancherel equation the functional can be written as

$$E(\hat{u}) \quad = \quad \left\| \hat{u} - \hat{f} \right\|_2^2 + \alpha \sum_{k=1}^{m} \lambda_k \left\| (i\xi)^k \hat{u} \right\|_2^2$$

$$\begin{aligned} &= \left\|\hat{u}-\hat{f}\right\|_2^2 + \alpha\sum_{k=1}^{m}\lambda_k \left\|\xi^k\hat{u}\right\|_2^2 \\ &= \int\limits_{\mathbb{R}} \left(\left|\hat{u}-\hat{f}\right|^2 + \alpha\sum_{k=1}^{m}\lambda_k \left|\xi^k\hat{u}\right|^2\right) dx. \end{aligned}$$

We note that the energy functional in the Fourier domain only depends on $|\hat{u}|^2$ since the derivative operation is transformed into multiplication via the Fourier transform. From Chapter 2 we see with a decomposition in the real and imaginary part that the corresponding Euler-Lagrange equation to this functional is given by

$$0 \quad = \quad E_{\hat{u}} \quad = \quad 2\left(\hat{u}-\hat{f}\right) + 2\alpha\sum_{k=1}^{m}\lambda_k\xi^{2k}\hat{u}.$$

It immediately follows that

$$\hat{f} = \hat{u} + \alpha\sum_{k=1}^{m}\lambda_k\xi^{2k}\hat{u}$$

which gives us the analytical solution in the Fourier domain

$$\hat{u} = \left(1+\alpha\sum_{k=1}^{m}\lambda_k\xi^{2k}\right)^{-1}\hat{f}.$$

Remark 3.1 It is also possible to compute the Euler-Lagrange equation and then solve it in the Fourier domain. As a special case of (2.18) we see that the corresponding Euler-Lagrange equation to the functional (3.1) is given by

$$\begin{aligned} 0 &= u - f + \alpha\sum_{k=1}^{m}\lambda_k(-1)^k\frac{d^k}{dx^k}u^{(k)} \\ &= u - f + \alpha\sum_{k=1}^{m}(-1)^k\lambda_k u^{(2k)}. \end{aligned} \tag{3.3}$$

Applying the Fourier transform then yields:

$$\begin{aligned} 0 &= \hat{u} - \hat{f} + \alpha\sum_{k=1}^{m}\lambda_k(-1)^k\xi^{2k}\hat{u} \\ &= \hat{u} - \hat{f} + \alpha\sum_{k=1}^{m}\lambda_k\xi^{2k}\hat{u}. \end{aligned}$$

It immediately follows that

$$\hat{f} = \hat{u} + \alpha\sum_{k=1}^{m}\lambda_k\xi^{2k}\hat{u}$$

which gives us the analytical solution in the Fourier domain

$$\hat{u} = \left(1+\alpha\sum_{k=1}^{m}\lambda_k\xi^{2k}\right)^{-1}\hat{f}.$$

In the case of linear filtering we can give analytical solutions with the help of the Fourier transform.

3.1.2 Parabolic Differential Equations

Instead of solving the Euler-Lagrange equation directly as in the last section one can also use iterative methods to approximate a solution. This may seem to be unnecessary for the linear case since analytical solutions can easily be found. Nevertheless we will draw our attention on this point to get results which can be generalised to nonlinear filtering.

Diffusion Equations

We start with the Euler-Lagrange equation (3.3) and exclude the trivial case $\alpha = 0$ in which the only solution is $u = f$. With positive α we can write the equation as

$$\frac{u-f}{\alpha} = \sum_{k=1}^{m} (-1)^{k+1} \lambda_k u^{(2k)}. \tag{3.4}$$

In the energy functional the parameter α determines the weight between the similarity and the smoothness term. Higher values of α lead to smoother solutions with respect to the actual penalty function. One could also think of filtering as an evolution process and interpret α as a time parameter. Let $u(x,t)$ denote the minimiser of the energy functional with $t = \alpha$. In this framework the initial signal f is the signal $u(\cdot, 0)$ at time $t = 0$. This corresponds with the observation that the left-hand side of equation (3.4) can be interpreted as implicit discretisation of the time derivative.

We can consider equation (3.4) as fully implicit discretisation of the parabolic differential equation

$$u_t = \sum_{k=1}^{m} (-1)^{k+1} \lambda_k u^{(2k)}$$

with initial condition $u(x,0) = f(x)$ for all $x \in \mathbb{R}$ and stopping time $t = \alpha$. We note that these interpretations are also possible in a nonlinear setting. So the Euler-Lagrange equation from Example 2.10 leads to the nonlinear diffusion equation

$$u_t = (-1)^{m+1} \frac{d^m}{dx^m} \left(\varphi' \left(\left(u^{(m)} \right)^2 \right) u^{(m)} \right). \tag{3.5}$$

Since the quadratic data term $(u-f)^2$ appears in all energy functionals we consider for signal and image processing, the corresponding Euler-Lagrange equations always contain the linear term $u - f$. So for every energy functional with a quadratic data term we can give a corresponding diffusion equation where the right-hand side comes from the smoothness term. This can be immediately carried over to higher dimensional examples.

Diffusion-Reaction Equations

One can also interpret the solution of the Euler-Lagrange equation (3.3) as the steady state of the diffusion-reaction equation

$$u_t = u - f + \alpha \sum_{k=1}^{m} (-1)^k \lambda_k u^{(2k)}$$

with the artificial time variable t and initial condition $u(\cdot, 0) = f$.

In this case one would get the equation

$$u_t = u - f + (-1)^m \alpha \frac{d^m}{dx^m} \left(\varphi' \left(\left(u^{(m)} \right)^2 \right) u^{(m)} \right). \tag{3.6}$$

belonging to the Euler-Lagrange equation from example (2.10). To solve the problem with the diffusion-reaction approach one has to reach the steady state efficiently. This can be seen as the main disadvantage compared with the pure diffusion equation: In the diffusion setting the scale parameter α is interpreted as stopping time. In the diffusion-reaction case one has to try to compute the limit for $t \longrightarrow \infty$ independent of the value of α.

3.1.3 Generalised Linear Diffusion

In this section we turn our attention to higher order diffusion equations. The proceeding is similar to the treatment of the diffusion equation in [27].

We have seen in the last section how the functional (3.1) leads to the parabolic differential equation

$$u_t = \sum_{k=1}^{m} (-1)^{k+1} \lambda_k \frac{d^{2k}}{dx^{2k}} u \tag{3.7}$$

with initial condition $u(x,0) = f(x)$ for all $x \in \mathbb{R}$ and stopping time $t = \alpha$. Applying the Fourier transform with respect to the x-variable to both sides of this equation yields

$$\begin{aligned} \hat{u}_t &= \sum_{k=1}^{m} (-1)^{k+1} \lambda_k (-1)^k \xi^{2k} \hat{u} \\ &= \sum_{k=1}^{m} (-1)^{2k+1} \lambda_k \xi^{2k} \hat{u} \\ &= -\sum_{k=1}^{m} \lambda_k \xi^{2k} \hat{u} \end{aligned} \tag{3.8}$$

with initial condition $\hat{u}(\xi, 0) = \hat{f}(\xi)$ for all $\xi \in \mathbb{R}$.

We have used the Fourier transform to turn the x-derivatives into algebraic multiplications. Only the derivative with respect to t is preserved and we obtain an ordinary differential equation with one parameter ξ. The unique solution for equation (3.8) is

$$\hat{u}(\xi, t) = \exp\left(-t \sum_{k=1}^{m} \lambda_k \xi^{2k}\right) \hat{f}(\xi) \tag{3.9}$$

for all $t \geq 0$ and all $\xi \in \mathbb{R}$.

We rewrite equation (3.9) to get some more properties of the solution of equation (3.8):

$$\begin{aligned} \hat{u}(\xi, t) &= \exp\left(-t \sum_{k=1}^{m} \lambda_k \xi^{2k}\right) \hat{f}(\xi) \\ &= \left(\prod_{k=1}^{m} \exp\left(-t\lambda_k \xi^{2k}\right)\right) \hat{f}(\xi) \\ &= \exp\left(-t\lambda_m \xi^{2m}\right) \cdot \ldots \cdot \exp\left(-t\lambda_1 \xi^2\right) \hat{f}(\xi). \end{aligned} \tag{3.10}$$

We see that the solution of equation (3.8) can be represented by multiplying the initial values with exponential functions in the Fourier domain.

Definition 3.2 (Multiplicators) *We define the multiplicator functions*

$$G_k^{\lambda}(\xi, t) := \exp\left(-t\lambda\xi^{2k}\right) \qquad \textit{with } k \in \mathbb{N}, \lambda \in \mathbb{R}$$

that appear in the Fourier domain for generalised linear filtering.

If we restrict λ to nonnegative values these functions are bounded:

$$0 \quad \leq \quad G_k^\lambda(\xi,t) = \exp\left(-t\lambda\xi^{2k}\right) \quad \leq \quad 1$$

for all $k \in \mathbb{N}$, $t \in \mathbb{R}_0^+$ and $\xi \in \mathbb{R}$.

Remark 3.3 (Stability) Together with equation (3.9) we get an upper bound for the L_2-norm of $\hat{u}(\cdot,t)$:

$$\begin{aligned}
\|\hat{u}(\cdot,t)\|_2^2 &= \int\limits_{\mathbb{R}} |\hat{u}(\xi,t)|^2 d\xi \\
&= \int\limits_{\mathbb{R}} \left|\exp\left(-t\lambda_m\xi^{2m}\right) \cdot \ldots \cdot \exp\left(-t\lambda_1\xi^2\right) \hat{f}(\xi)\right|^2 d\xi \\
&= \int\limits_{\mathbb{R}} \left|\exp\left(-t\lambda_m\xi^{2m}\right)\right|^2 \cdot \ldots \cdot \left|\exp\left(-t\lambda_1\xi^2\right)\right|^2 |\hat{f}(\xi)|^2 d\xi \\
&\leq \int\limits_{\mathbb{R}} |\hat{f}(\xi)|^2 d\xi \\
&= \|\hat{f}\|_2^2. \qquad (3.11)
\end{aligned}$$

This shows not only that $\hat{u}(\cdot,t)$ is in L_2 again for initial data in L_2: Since the mapping $\hat{T}_t : L_2 \longrightarrow L_2$, $\hat{f} \longmapsto \hat{u}(\cdot,t)$ is linear, we have also shown the continuity of $\hat{T}_t$.

Together with the Fourier-Plancherel transform $\mathcal{F} : L_2 \longrightarrow L_2$ (see [33, Section V.2], linear filtering with fixed nonnegative coefficients $\lambda_1, \ldots, \lambda_m$ and stopping time t can be written as linear continuous operator $T_t : L_2 \longrightarrow L_2$, $f \longmapsto \mathcal{F}^{-1}\hat{T}_t\mathcal{F}f$. The Plancherel identity assures that the equivalent inequation to (3.11) holds for the norm of $T_t f = u(\cdot,t)$ for all $t \in \mathbb{R}_0^+$:

$$\|u(\cdot,t)\|_2 \quad = \quad \|\hat{u}(\cdot,t)\|_2 \quad \leq \quad \|\hat{f}\|_2 \quad = \quad \|f\|_2.$$

We conclude that linear diffusion filtering of higher order is L_2-continuous with norm not greater than 1. That means stability with respect to the L_2-norm.

Remark 3.4 (Negative Values for λ) We want to note that these properties only hold for nonnegative coefficients $\lambda_1, \ldots, \lambda_m$. The functions

$$G_k^\lambda(\xi,t) = \exp\left(-t\lambda\xi^{2k}\right)$$

are unbounded for negative λ. Especially the high frequency components of our signal f are amplified by multiplying them with exponentially growing functions. Filtering with negative $\lambda_1, \ldots, \lambda_m$ leads to a rapidly growing L_2-norm of the result $u(\cdot,t)$. Thus the problem is ill-posed for negative filter parameters. For the rest of this section we assume the $\lambda_1, \ldots, \lambda_m$ to be nonnegative.

For $t \geq 0$, the functions $G_k^\lambda(\cdot,t)$ are in the Schwartz space (see [33]). So their Fourier backtransform $p_k^\lambda(\cdot,t)$ exists and is in the Schwartz space, too (see [33, Lemma V.2.5]). Together with the convolution theorem the solution of equation (3.8) can also be considered as convolution in the spatial domain

$$u(x,t) = \left(p_m^{\lambda_m}(\cdot,t) * p_{m-1}^{\lambda_{m-1}}(\cdot,t) * \ldots * p_1^{\lambda_1}(\cdot,t) * f\right)(x). \qquad (3.12)$$

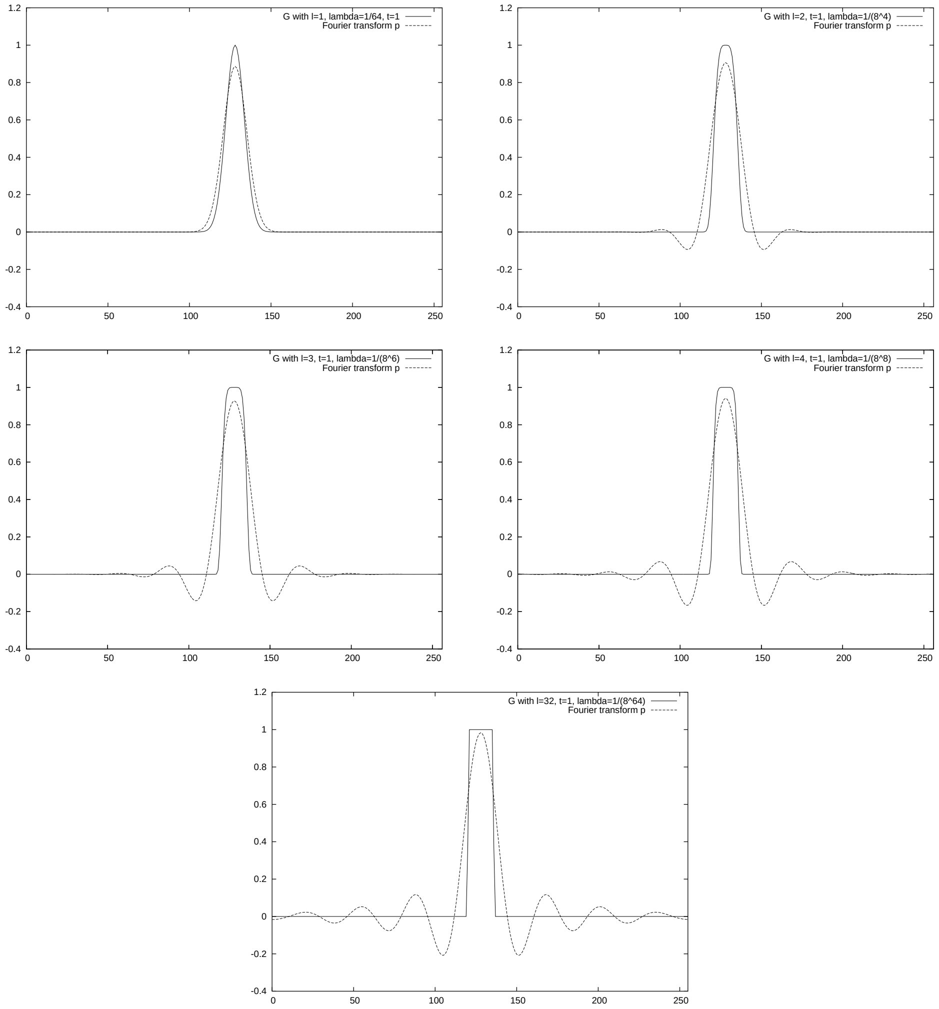

Figure 3.1: $G_k^\lambda(\cdot,t)$ and corresponding convolution kernels $p_k^\lambda(\cdot,t)$ for $k \in \{1,2,3,4,32\}$. The values for λ and t are chosen to visualise the main characteristics of the functions.

Definition 3.5 (Convolution Kernels) *We define the convolution kernels appearing in generalised linear filtering as*

$$\begin{aligned} p_k^\lambda(\cdot,t) &:= \mathcal{F}^{-1} G_k^\lambda(\cdot,t) \quad \textit{for } k \in \mathbb{N}, \lambda > 0 \\ &= \int\limits_{\mathbb{R}} \exp\left(-t\lambda\xi^{2k}\right) \exp\left(ix\xi\right) d\xi. \end{aligned}$$

Figure 3.1 shows some discrete approximations of the functions $G_k^\lambda(\cdot,t)$ and the resulting convolution kernels $p_k^\lambda(\cdot,t)$ visualised via discrete Fourier transform. We have used a signal size of 256 pixels. In the case $k = 1$, we have the well-known Gaussian kernel which has a Gaussian as Fourier transform again. We note that the nonnegativity of the Gaussian corresponds with the minimum-maximum property of the corresponding diffusion process. Since higher order kernels $p_k^\lambda(\cdot,t)$ always reach negative values, the corresponding higher order diffusion will in general violate a minimum-maximum property.

Remark 3.6 From the signals shown in Figure 3.1 one could suppose that the number of intervals where p_k^λ has negative values is $2(k-1)$. It could also be possible to choose the λ_k such that the convolution of multiple kernels is positive. This would be the counterpart in the diffusion framework to the positivity results from [18] for direct minimisation of linear energy functionals. One can also see that the discrete versions of the functions G_k^λ get similar to a box function for $k \longrightarrow \infty$ and the functions p_k^λ get similar to a sinc function. Perhaps it is possible to prove convergence in the sense of the L_2-norm in the continuous case, too (with the L_2-continuity of the Fourier transform, it would suffice to show the convergence for either G_k^λ or p_k^λ).

We note some scale-space properties of generalised linear filtering. The first one we have already used in the above remarks:

Proposition 3.7 (Linearity) *Generalised linear filtering is a linear operator.*

From the convolution representation (3.12) with kernels $p_k^\lambda \in C^\infty$ we can derive that the filtering result is C^∞ for L_2 initial data:

Proposition 3.8 (Smoothness of the Solution) *We consider generalised linear filtering with initial data $f \in L_2(\mathbb{R})$. Then the solution u is in $C^\infty(\mathbb{R})$.*

Proof: We start with (3.12) and compute the convolution kernel

$$p = p_m^{\lambda_m}(\cdot,t) * p_{m-1}^{\lambda_{m-1}}(\cdot,t) * \ldots * p_1^{\lambda_1}(\cdot,t).$$

Then p is in $C^\infty(\mathbb{R})$, and we can write

$$\frac{d}{dx}u(x,t) = \int\limits_{\mathbb{R}} \left(\frac{d}{dx}p(y-x)\right) f(y)dy$$

which can be iterated and shows the existence of the derivatives of u. □

From equation (3.9), the properties of the exponential function and the linearity of the Fourier-Plancherel transform $\mathcal{F}$ we can also derive:

Proposition 3.9 (Semigroup-Property of Linear Filtering) *The set of generalised linear filtering operators T_t satisfies the semigroup-property*

$$\begin{aligned} T_{s+t}f &= T_s T_t f \\ T_0 f &= f \end{aligned}$$

for all $f \in L_2$.

Proposition 3.10 (Invariance of the Average Grey Value) *For all $t > 0$ it is*

$$\int\limits_{\mathbb{R}} u(x,t)dx = \int\limits_{\mathbb{R}} f(x)dx.$$

Proof: We keep in mind that the average grey value can be expressed as Fourier coefficient

$$\int\limits_{\mathbb{R}} f(x)dx = \int\limits_{\mathbb{R}} f(x)\exp(0ix)dx = \hat{f}(0).$$

Thus we have for all $t > 0$

$$\begin{aligned}\int_{\mathbb{R}} u(x,t)dx &= \hat{u}(0,t)\\ &= \exp(-t\lambda_m 0)\cdot\ldots\cdot\exp(-t\lambda_1 0)\hat{f}(0)\\ &= \hat{f}(0)\\ &= \int_{\mathbb{R}} f(x)dx\end{aligned}$$

which completes the proof. □

Proposition 3.11 (Translational Invariance) *Generalised linear diffusion filtering is translational invariant.*

Proof: This statement follows directly from the fact that generalised linear diffusion can be written as spatial convolution with a kernel $p(x,t)$. Then a substitution shows for all $a \in \mathbb{R}$

$$\begin{aligned}u(x+a,t) &= \int_{\mathbb{R}} p\left(x+a-y,t\right)f(y)dy\\ &= \int_{\mathbb{R}} p\left(x-(y-a),t\right)f(y)dy\\ &= \int_{\mathbb{R}} p\left(x-\tilde{y}\right)f\left(\tilde{y}+a\right)d\tilde{y},\end{aligned}$$

which is the claimed translational invariance. □

If we restrict ourselves to only one derivative order we have also scale invariance.

Proposition 3.12 (Scale Invariance) *If the function f is scaled with $\sigma > 0$ then there is a $\tilde{t} > 0$ such that*

$$\left(T_t f\left(\frac{\cdot}{\sigma}\right)\right)(x) = \left(T_{\tilde{t}} f(\cdot)\right)\left(\frac{x}{\sigma}\right)$$

holds.

Proof: First we note that with the substitution $\tilde{y} = \frac{y}{\sigma}$

$$\begin{aligned}\left(T_t f\left(\frac{\cdot}{\sigma}\right)\right)(x) &= \int_{\mathbb{R}} p_k^\lambda(x-y,t)f\left(\frac{y}{\sigma}\right)dy\\ &= \sigma\int_{\mathbb{R}} p_k^\lambda(x-\sigma\tilde{y},t)f(\tilde{y})d\tilde{y}.\end{aligned} \tag{3.13}$$

We observe that with Definition 3.5 and a second substitution

$$\begin{aligned}p_k^\lambda(x-\sigma\tilde{y},t) &= \int_{\mathbb{R}} \exp\left(-t\lambda\xi^{2k}\right)\exp\left(i(x-\sigma\tilde{y})\xi\right)d\xi\\ &= \frac{1}{\sigma}\int_{\mathbb{R}} \exp\left(-\frac{t}{\sigma^{2k}}\lambda\tilde{\xi}^{2k}\right)\exp\left(i\left(\frac{x}{\sigma}-\tilde{y}\right)\tilde{\xi}\right)d\tilde{\xi}\\ &= \frac{1}{\sigma}p_k^\lambda\left(\frac{x}{\sigma}-\tilde{y},\frac{t}{\sigma^{2k}}\right).\end{aligned}$$

Using this in equation (3.13) yields

$$\begin{aligned}\left(T_t f\left(\frac{\cdot}{\sigma}\right)\right)(x) &= \int\limits_{\mathbb{R}} p_k^\lambda\left(\frac{x}{\sigma}-y, \frac{t}{\sigma^{2k}}\right) f(y)dy \\ &= \left(T_{\frac{t}{\sigma^{2k}}} f\right)\left(\frac{x}{\sigma}\right)\end{aligned}$$

which proves the claimed scale-invariance. □

We note that the convolution with $p_k^\lambda(\cdot, t)$ has been axiomatically derived by Iijima in [16]. He starts with demanding linearity, translational and scale invariance and the semigroup property and obtains the convolution with $p_k^\lambda(\cdot, t)$. More on these scale-space axiomatics can be found in [32].

3.2 Nonlinear Diffusion of Second Order

Let us now turn our attention to nonlinear energy functionals of derivative order 2 in one dimension. Such functionals can be written in the following form:

$$E(u) = \int\limits_{\mathbb{R}} \left((u-f)^2 + \alpha\varphi\left(u_{xx}^2\right)\right) dx. \tag{3.14}$$

We assume that $\varphi \in C^3(\mathbb{R})$, that means it is three times continuous differentiable. Further we demand that it satisfies the condition $\varphi(0) = 0$ to ensure that the integral converges at least for constant or linear argument functions u. The corresponding Euler-Lagrange equation to (3.14) is the elliptic equation

$$\frac{u-f}{\alpha} = -\frac{d^2}{dx^2}\left(\varphi'\left(u_{xx}^2\right) u_{xx}\right).$$

As we have discussed in Section 3.1.2 this can be interpreted as a fully implicit discretisation of the parabolic partial differential equation

$$u_t = -\frac{d^2}{dx^2}\left(\varphi'\left(u_{xx}^2\right) u_{xx}\right) \tag{3.15}$$

with initial condition $u(0, x) = f(x)$ and stopping time $t = \alpha$. We now expand the right-hand side of this equation. Using

$$\begin{aligned}\frac{d}{dx}\left(\varphi'\left(u_{xx}^2\right) u_{xx}\right) &= \left[\frac{d}{dx}\varphi'\left(u_{xx}^2\right)\right] u_{xx} + \varphi'\left(u_{xx}^2\right)\left[\frac{d}{dx}u_{xx}\right] \\ &= 2\varphi''\left(u_{xx}^2\right) u_{xx}^2 u_{xxx} + \varphi'\left(u_{xx}^2\right) u_{xxx} \\ &= \left(2\varphi''\left(u_{xx}^2\right) u_{xx}^2 + \varphi'\left(u_{xx}^2\right)\right) u_{xxx}\end{aligned}$$

we obtain

$$\begin{aligned}\frac{d^2}{dx^2}\left(\varphi'\left(u_{xx}^2\right) u_{xx}\right) &= \left[\frac{d}{dx}\left(2\varphi''\left(u_{xx}^2\right) u_{xx}^2 + \varphi'\left(u_{xx}^2\right)\right)\right] u_{xxx} \\ &\quad + \left[2\varphi''\left(u_{xx}^2\right) u_{xx}^2 + \varphi'\left(u_{xx}^2\right)\right] u_{xxxx} \\ &= 2\left[\left(\frac{d}{dx}\varphi''\left(u_{xx}^2\right)\right) u_{xx}^2 + \varphi''\left(u_{xx}^2\right)\left(\frac{d}{dx}u_{xx}^2\right)\right. \\ &\quad \left. + \varphi''\left(u_{xx}^2\right) u_{xx} u_{xxx}\right] u_{xxx} \\ &\quad + \left[2\varphi''\left(u_{xx}^2\right) u_{xx}^2 + \varphi'\left(u_{xx}^2\right)\right] u_{xxxx}\end{aligned}$$

$$
\begin{aligned}
&= \left[\varphi'''\left(u_{xx}^2\right) 2u_{xx}^3 u_{xxx} + \varphi''\left(u_{xx}^2\right) 2u_{xx}u_{xxx} \right.\\
&\qquad \left. +\varphi''\left(u_{xx}^2\right) u_{xx}u_{xxx}\right] 2u_{xxx}\\
&\quad + \left[2\varphi''\left(u_{xx}^2\right) u_{xx}^2 + \varphi'\left(u_{xx}^2\right)\right] u_{xxxx}\\
&= \left[2u_{xxx}^2\left(2\varphi'''\left(u_{xx}^2\right) u_{xx}^2 + 3\varphi''\left(u_{xx}^2\right)\right)\right] u_{xx}\\
&\quad + \left[2\varphi''\left(u_{xx}^2\right) u_{xx}^2 + \varphi'\left(u_{xx}^2\right)\right] u_{xxxx}.
\end{aligned}
$$

Introducing the abbreviations

$$
\begin{aligned}
\Phi_1(x^2) &:= 2\varphi'''(x^2)x^2 + 3\varphi''(x^2)\\
\Phi_2(x^2) &:= 2\varphi''(x^2)x^2 + \varphi'(x^2) \qquad (3.16)
\end{aligned}
$$

allows us to rewrite equation (3.15) as follows:

$$
\begin{aligned}
u_t &= -\frac{d^2}{dx^2}\left(\varphi'\left(u_{xx}^2\right) u_{xx}\right)\\
&= -\left(2u_{xxx}^2 \Phi_1\left(u_{xx}^2\right)\right) u_{xx} - \Phi_2\left(u_{xx}^2\right) u_{xxxx}. \qquad (3.17)
\end{aligned}
$$

As we have seen in Section 3.1.3 the well-posedness of diffusion processes depends on the signs of the factors in front of u_{xx} and u_{xxxx}. We now consider the nonlinear terms $-2u_{xxx}^2\Phi_1\left(u_{xx}^2\right)$ and $\Phi_2\left(u_{xx}^2\right)$ as coefficients of u_{xx} and u_{xxxx} and we are therefore interested in the signs of these terms.

In regions where $u_{xxx}^2 \neq 0$ the signs only depend on the functions Φ_1 and Φ_2 which involve the first three derivatives of the penalty function φ. Note that the expression $\Phi_2(x^2)$ as defined in (3.16) also appears in the consideration of first order filtering to distinguish between forward and backward diffusion. So the highest diffusion order will behave like the first order diffusion in a standard first order setting. The third derivative u_{xxx} appears only quadratic and therefore cannot change the sign. It can be seen as a weight parameter on the first order diffusion process. We keep in mind that we only obtain necessary conditions for first order forward or backward diffusion.

The conditions on Φ_1 and Φ_2 can be summarised as follows:

$\Phi_1\left(u_{xx}^2\right) < 0$	first order forward diffusion (well-posed)
$\Phi_1\left(u_{xx}^2\right) > 0$	first order backward diffusion (ill-posed)
$\Phi_2\left(u_{xx}^2\right) > 0$	second order forward diffusion (well-posed)
$\Phi_2\left(u_{xx}^2\right) < 0$	second order backward diffusion (ill-posed).

3.3 Application to Penalty Functions

Let us now investigate some commonly used penalisers φ and study their effect on corresponding second order diffusions. The investigated penalisers are already mentioned in the first chapter. More references can be found there.

3.3.1 Linear Filtering

A linear filter is obtained with $\varphi(x) := cx$ for some $c \in \mathbb{R}$ and all $x \in \mathbb{R}$.
With $\varphi'(x) = c$ and $\varphi''(x) = \varphi'''(x) = 0$ it immediately follows that $\Phi_1(x) = 0$ and $\Phi_2(x) = c$ hold for all $x \in \mathbb{R}$. Only the highest diffusion order is kept in this case. The corresponding parabolic differential equation is

$$u_t = -c\, u_{xxxx}.$$

For $c > 0$, linear filtering leads to a well-posed second-order forward diffusion.

3.3.2 Charbonnier

The Charbonnier filter with

$$\varphi(x) := 2\lambda^2 \left(\sqrt{1+\frac{x}{\lambda^2}} - 1 \right) \text{ with } \lambda \in \mathbb{R}^+, x \in \mathbb{R}_0^+$$

leads to forward diffusion in the first order filtering framework. Note that the argument x will be always nonnegative in energy functionals of the form (3.14) since the penalty depends on the square of the second derivative of our data u. We will now derive that its behaviour is exactly the same in a second order filter.

The first three derivatives of φ are:

$$\begin{aligned} \varphi'(x) &= \left(1+\frac{x}{\lambda^2}\right)^{-\frac{1}{2}} \\ \varphi''(x) &= \frac{-1}{2\lambda^2}\left(1+\frac{x}{\lambda^2}\right)^{-\frac{3}{2}} \\ \varphi'''(x) &= \frac{3}{4\lambda^4}\left(1+\frac{x}{\lambda^2}\right)^{-\frac{5}{2}}. \end{aligned}$$

We then derive our sign functions as defined in (3.16):

$$\begin{aligned} \Phi_1(x^2) &= \frac{3x^2}{2\lambda^4}\left(1+\frac{x^2}{\lambda^2}\right)^{-\frac{5}{2}} + \frac{-3}{2\lambda^2}\left(1+\frac{x^2}{\lambda^2}\right)^{-\frac{3}{2}} \\ &= \frac{-3}{2\lambda^2}\left(1+\frac{x^2}{\lambda^2}\right)^{-\frac{5}{2}} \quad < 0 \quad \text{for all } x \in \mathbb{R} \text{ and} \end{aligned}$$

$$\begin{aligned} \Phi_2(x^2) &= \frac{-2x^2}{2\lambda^2}\left(1+\frac{x^2}{\lambda^2}\right)^{-\frac{3}{2}} + \left(1+\frac{x^2}{\lambda^2}\right)^{-\frac{1}{2}} \\ &= \left(1+\frac{x^2}{\lambda^2}\right)^{-\frac{3}{2}} \quad > 0 \quad \text{for all } x \in \mathbb{R}. \end{aligned}$$

The Charbonnier filter of second order always performs forward diffusion.

3.3.3 Perona-Malik

Let us consider the classical Perona-Malik penalty function

$$\varphi(x) := \lambda^2 \ln\left(1+\frac{x}{\lambda^2}\right) \qquad \text{with } x \in \mathbb{R}_0^+, \lambda \in \mathbb{R}^+.$$

The Perona-Malik function has the property to lead to forward-backward diffusion in first order filtering methods. The parameter λ is the limit to distinguish between forward diffusion (for $|u_x| < \lambda$) and backward diffusion (for $|u_x| > \lambda$). Thus λ plays the role of a contrast parameter. Choosing a suitable value for λ allows to smooth in the interior of a region (where $|u_x| < \lambda$), while enhancing edges (with $|u_x| < \lambda$) by means of backward diffusion. We will see that the behaviour in the second-order case is not that easy to describe.

Computation of the first three derivatives of our penalty function yields

$$\begin{aligned} \varphi'(x) &= \left(1+\frac{x}{\lambda^2}\right)^{-1} \\ \varphi''(x) &= \frac{-1}{\lambda^2}\left(1+\frac{x}{\lambda^2}\right)^{-2} \\ \varphi'''(x) &= \frac{2}{\lambda^4}\left(1+\frac{x}{\lambda^2}\right)^{-3}. \end{aligned}$$

Then we consider the sign determining functions Φ_1 and Φ_2.

$$\begin{aligned}
\Phi_1(x^2) &= 2\varphi'''(x^2)x^2 + 3\varphi''(x^2) \\
&= \frac{4x^2}{\lambda^4}\left(1+\frac{x^2}{\lambda^2}\right)^{-3} + \frac{-3}{\lambda^2}\left(1+\frac{x^2}{\lambda^2}\right)^{-2} \\
&= \frac{x^2-3\lambda^2}{\lambda^4\left(1+\frac{x^2}{\lambda^2}\right)^3} \\
\Phi_2(x^2) &= 2\varphi''(x^2)x^2 + \varphi'(x^2) \\
&= \frac{-2x^2}{\lambda^2}\left(1+\frac{x^2}{\lambda^2}\right)^{-2} + \left(1+\frac{x^2}{\lambda^2}\right)^{-1} \\
&= \frac{\lambda^2-x^2}{\lambda^2}\left(1+\frac{x^2}{\lambda^2}\right)^{-2}
\end{aligned}$$

We see that the term Φ_2 for the fourth derivative order leads to the well-known condition from first order filtering (with the difference that this function depends on the square of the second and not of the first derivative.)

$$\Phi_2(x^2) > 0 \quad\Longleftrightarrow\quad x^2 < \lambda^2 \quad\Longleftrightarrow\quad |x| < \lambda$$

We have second order forward diffusion in regions where $|u_{xx}| < \lambda$ and second order backward diffusion if $|u_{xx}^2| > \lambda$.

The first order diffusion depends on Φ_1:

$$\Phi_1(x^2) < 0 \quad\Longleftrightarrow\quad x^2 - 3\lambda^2 < 0 \quad\Longleftrightarrow\quad x^2 < 3\lambda^2 \quad\Longleftrightarrow\quad |x| < \sqrt{3}\lambda.$$

So this term may lead to forward first order diffusion in the case that $|u_{xx}^2| < \sqrt{3}\lambda$ and to backward first order diffusion if it is greater. We conclude that the behaviour of second-order Perona-Malik filtering can be divided into three cases:

$\lvert u_{xx}\rvert < \lambda$	forward diffusion of first and second order (well-posed)
$\lambda < \lvert u_{xx}\rvert < \sqrt{3}\lambda$	forward diffusion of first order and backward diffusion of second order
$\lvert u_{xx}\rvert > \sqrt{3}\lambda$	backward diffusion of first and second order (ill-posed).

3.3.4 Total Variation Approximations

As next example we consider approximations of the total variation function. We fix $\varepsilon > 0$ and consider the regularised total variation penaliser

$$\varphi(x) := 2\sqrt{(\varepsilon^2 + x)} - 2\varepsilon \qquad \text{for } x \in \mathbb{R}_0^+.$$

We need to subtract the constant term 2ε to get $\varphi(0) = 0$. As in the above examples we need the derivatives of φ at first; these are

$$\begin{aligned}
\varphi'(x) &= \left(\varepsilon^2+x\right)^{-\frac{1}{2}} \\
\varphi''(x) &= \frac{-1}{2}\left(\varepsilon^2+x\right)^{-\frac{3}{2}} \\
\varphi'''(x) &= \frac{3}{4}\left(\varepsilon^2+x\right)^{-\frac{5}{2}}.
\end{aligned}$$

According to formula (3.16) we compute the coefficient functions

$$\begin{aligned}\Phi_1(x^2) &= \frac{3x^2}{2}\left(\varepsilon^2+x^2\right)^{-\frac{5}{2}} + \frac{-3}{2}\left(\varepsilon^2+x^2\right)^{-\frac{3}{2}}\\ &= \frac{-3\varepsilon^2}{2}\left(\varepsilon^2+x^2\right)^{-\frac{5}{2}} \quad < 0 \quad \text{for all } x\in\mathbb{R} \text{ and}\\ \Phi_2(x^2) &= \frac{-2x^2}{2}\left(\varepsilon^2+x^2\right)^{-\frac{3}{2}} + \left(\varepsilon^2+x^2\right)^{-\frac{1}{2}}\\ &= \varepsilon^2\left(\varepsilon^2+x^2\right)^{-\frac{3}{2}} \quad > 0 \quad \text{for all } x\in\mathbb{R}.\end{aligned}$$

We deduce that regularised total variation approximations always perform forward diffusion. In the case the second derivative $|u_{xx}|$ is small the value of ε plays an important role: The limits of our two functions are

$$\lim_{x\longrightarrow 0}\Phi_1(x^2) \quad = \quad \lim_{x\longrightarrow 0} -\frac{3}{2}\cdot\frac{\varepsilon^2}{\sqrt{\varepsilon^2+x^2}^5} \quad = \quad -\frac{3}{2}\varepsilon^{-3}$$

and

$$\lim_{x\longrightarrow 0}\Phi_2(x^2) \quad = \quad \lim_{x\longrightarrow 0} \frac{\varepsilon^2}{\sqrt{\varepsilon^2+x^2}^3} \quad = \quad \frac{1}{\varepsilon}.$$

We see that the value of ε for the TV-approximation is very important for the speed of the diffusion in regions with small second derivative.

3.3.5 Total Variation

Instead of the approximation we now consider the TV functional itself. We use the penaliser $\varphi(x) = 2\sqrt{x}$ for $x \in \mathbb{R}_0^+$. The derivatives of φ

$$\begin{aligned}\varphi'(x) &= x^{-\frac{1}{2}}\\ \varphi''(x) &= -\frac{1}{2}x^{-\frac{3}{2}}\\ \varphi'''(x) &= \frac{3}{4}x^{-\frac{5}{2}}\end{aligned}$$

are only defined for $x \in \mathbb{R}^+$. Our coefficient functions then are

$$\begin{aligned}\Phi_1(x^2) &= \frac{3}{2}(x^2)^{-\frac{5}{2}}x^2 - \frac{3}{2}(x^2)^{-\frac{3}{2}}\\ &= \frac{3}{2}x^{-3} - \frac{3}{2}x^{-3}\\ &= 0 \quad \text{for all } x\in\mathbb{R}^+ \text{ and}\\ \Phi_2(x^2) &= -(x^2)^{-\frac{3}{2}}x^2 + (x^2)^{-\frac{1}{2}}\\ &= -x^{-1} + x^{-1}\\ &= 0 \quad \text{for all } x\in\mathbb{R}^+.\end{aligned}$$

We see that the total variation method is exactly the border case between forward and backward diffusion. Since the derivatives are not defined for $x = 0$ the method is usually approximated as mentioned above for practical implementations.

Chapter 4

Discretisation

After studying methods for minimisation in the continuous setting we want to carry over these to the discrete domain. At first we will take a look at two ways to approximate the derivative only involving the function's values at discrete points: the finite differences and spectral methods. Both of them will be used in the noise removal methods described in this thesis.

4.1 General Remarks and Notations

First we like to introduce some general notations that will be used throughout this chapter.

In the one-dimensional continuous case we usually consider energy functionals with an interval $\Omega = (a, b)$ with $a, b \in \mathbb{R}$, $b > a$ as integration domain. Our data and results are functions $f : [a, b] \longrightarrow \mathbb{R}$. For discrete methods we choose a signal length $n \in \mathbb{N}$ and replace the interval by equidistant grid points

$$\begin{aligned} \Omega_h &:= \{x_i \mid i = 0, \ldots, n-1\} \quad \text{with} \\ x_i &:= a + ih \quad \text{for} \quad i \in \{0, \ldots, n-1\}, \end{aligned}$$

where $h := \frac{b-a}{n-1}$ is our spatial step size. We substitute the function f by a vector $\mathbf{f} \in \mathbb{R}^n$ containing the values of f at these grid points

$$\mathbf{f} := (f(x_0), \ldots, f(x_{n-1}))^T.$$

This vector will always be denoted with the letter of the corresponding function in bold print. Extending this vector at both sides with zeros, one can represent it also as an infinite sequence. It is clear that this sequence belongs to the spaces

$$\ell^p(\mathbb{Z}) := \left\{ (\xi_k)_{k \in \mathbb{Z}} \subset \mathbb{R}; \sum_{k \in \mathbb{Z}} |\xi_k|^p < \infty \right\}$$

for all $p \in \mathbb{N}$ since it is bounded and only a finite number of entries is nonzero. This extension can be useful to write finite differences as linear operators or as convolution. We are especially interested in the space $\ell^2(\mathbb{Z})$ since is is a Hilbert space with the scalar product

$$\langle \mathbf{f}, \mathbf{g} \rangle = \sum_{k \in \mathbb{Z}} f_k g_k.$$

We introduce some common operators on this space that will be useful to express finite difference methods.

Definition 4.1 (Left and Right Shift) *We define the* ***left shift operator***

$$L : \ell^2(\mathbb{Z}) \longrightarrow \ell^2(\mathbb{Z}), \quad (L\mathbf{f})_k = \mathbf{f}_{k+1} \quad \textit{for all } k \in \mathbb{Z},$$

and the ***right shift operator***

$$R : \ell^2(\mathbb{Z}) \longrightarrow \ell^2(\mathbb{Z}), \quad (R\mathbf{f})_k = \mathbf{f}_{k-1} \quad \textit{for all } k \in \mathbb{Z}.$$

The following lemma summarises some useful and simple properties of the shifts that we will use later on:

Lemma 4.2 (Properties of Shifts) *For L and R defined as above the following properties hold:*

1. *The left and the right shift are adjoint operators, i. e. for all $\mathbf{f}, \mathbf{g} \in \ell^2(\mathbb{Z})$ the identity $\langle L\mathbf{f}, \mathbf{g}\rangle = \langle \mathbf{f}, R\mathbf{g}\rangle$ holds.*
2. *The left and the right shift are inverse to each other, i. e. $LR = RL = I$. In particular the operators L and R commute.*

We also define a subspace of $\ell^2(\mathbb{Z})$ isomorphic to $\mathbb{R}^n$ such that we can use the theory in our practical examples with finite Ω_h.

Definition 4.3 (M^n and Orthogonal Projections) *Let $n \in \mathbb{N}$. We define*

$$M^n := \left\{(\xi_k)_{k\in\mathbb{Z}} \in \ell^2(\mathbb{Z}) \mid \xi_k = 0 \textit{ for all } k \notin \{0, \dots, n-1\}\right\}$$

as the subspace of $\ell^2(\mathbb{Z})$ of all sequences for which only the entries with indices $0, \dots, n-1$ may be nonzero. With $P_{M^n} : \ell^2(\mathbb{Z}) \longrightarrow M^n$ we denote the orthogonal projection onto M^n.

It is clear that M^n is isomorphic to $\mathbb{R}^n$. We would also note that as orthogonal projection the operator P_{M^n} is self-adjoint. These notions will be useful to obtain matrix representations for the results in $\ell^2(\mathbb{Z})$.

With $\ell^\infty(\mathbb{Z})$ we denote the space of all bounded real sequences over $\mathbb{Z}$. The extension of $\mathbf{f}$ can also be regarded as a sequence in ℓ^∞. For details on this sequence spaces we refer to introductions on functional analysis like [13] or [33]. We will see that in the case of spectral methods a periodic extension of the vector to a sequence in $\ell^\infty(\mathbb{Z})$ is the natural proceeding.

In this work we also consider two-dimensional examples with integration domain $\Omega = (a,b) \times (c,d) \subset \mathbb{R}^2$ with $a, b, c, d \in \mathbb{R}$, $b > a$, $d > c$. This domain will be replaced by an equidistant rectangular grid of the form

$$\begin{aligned} \Omega_h &:= \{(x_i, y_j) \mid i = 0, \dots, n_x - 1 \text{ and } j = 0, \dots, n_y - 1\} \qquad \text{with} \\ x_i &:= a + ih_x \quad \text{for } i \in \{0, \dots, n_x - 1\} \\ y_j &:= b + jh_y \quad \text{for } j \in \{0, \dots, n_y - 1\}, \end{aligned}$$

where $h_x := \frac{b-a}{n_x-1}$ and $h_y := \frac{d-c}{n_y-1}$ are the spatial step sizes in the coordinate axis directions. The values of a function $f : \Omega \longrightarrow \mathbb{R}$ on the set Ω_h could be written as a matrix $\tilde{f} \in \mathbb{R}^{n_x \times n_y}$ with entries $\tilde{f}_{ij} = f(x_i, y_j)$. Instead we prefer to embed the values in the vector $\mathbf{f} \in \mathbb{R}^{n_x n_y}$ with

$$\mathbf{f}_{i+jn_x} = f(x_i, y_j), \quad i = 0, \dots, n_x - 1, \ j = 0, \dots, n_y - 1.$$

So we are able to write noise removal methods as matrix-vector multiplications.

Before we are able to talk about discrete noise removal methods we need some background on the approximation of derivatives. We will summarise the main results concerning two different ways to approximate a derivative.

4.2 Finite Differences

Finite differences are the commonly used way to approximate derivatives. The main idea is to approximate differential quotients $f'(x) = \lim\limits_{h \longrightarrow \infty} \frac{f(x+h)-f(x)}{h}$ by suitable difference quotients.

Definition 4.4 (Forward, Backward and Central Difference) *Let $h > 0$ be an arbitrary step size, $x \in \mathbb{R}$, and $f : [x-h, x+h] \longrightarrow \mathbb{R}$. We define the* ***forward difference***

$$\left(D_h^+ f\right)(x) := \frac{f(x+h)-f(x)}{h},$$

the ***backward difference***

$$\left(D_h^- f\right)(x) := \frac{f(x)-f(x-h)}{h}$$

and the ***symmetric*** *or* ***central difference***

$$\left(D_h^0 f\right)(x) := \frac{f(x+h)-f(x-h)}{2h}.$$

If f is sufficient smooth (in this case C^2) the Taylor formula

$$f(x+h) = f(x) + hf'(x) + \frac{1}{2}h^2 f''(\xi) \text{ for } \xi \in (x, x+h)$$

yields the error estimate

$$\left(D_h^+ f\right)(x) = f'(x) + O(h)$$

expressed with the Landau symbol. The error is linear in h. The backward difference has the same accuracy order as the forward difference. The central difference even yields a quadratic convergence behaviour for $h \longrightarrow 0$. We note that

$$\left(D_h^+ f\right)(x) = \frac{f(x+h)-f(x)}{h} = \left(D_{h/2}^0 f\right)\left(x + \frac{h}{2}\right)$$

and so the forward difference of f at the point x with step size h can also be considered as an approximation for $f'\left(x + \frac{h}{2}\right)$ of second order.

Iterating the forward difference yields approximations for higher derivatives:

$$\begin{aligned} \left(D_h^+ D_h^+ f\right)(x) &= \frac{\left(D_h^+ f\right)(x+h) - \left(D_h^+\right)(x)}{h} \\ &= \frac{f(x+2h) - 2f(x+h) + f(x)}{h^2} \\ &\approx f''(x+h). \end{aligned}$$

It is useful to write the forward difference as linear operator to generalise the iteration and to get some statements concerning boundary conditions. The idea of this can be found in [4] and [14], for example, where calculations with operator symbols for differences are performed. We use the vector notation for the values of f at the grid points $\Omega_h = \{x, x+h, \ldots, x+mh\}$ as indicated in the preceeding section and set $\mathbf{f} = (f(x), \ldots, f(x+mh))$. Extending $\mathbf{f}$ to both sides with zero entries makes a sequence in $\ell^2(\mathbb{Z})$ out of it. The pointwise forward and backward differences now can be written as linear operator on ℓ^2:

Definition 4.5 (Difference Operators) *We consider the pointwise forward difference operator $D_h^+ : \ell^2(\mathbb{Z}) \longrightarrow \ell^2(\mathbb{Z})$ defined by*

$$\left(D_h^+ \mathbf{f}\right)_k = \frac{\mathbf{f}_{k+1} - \mathbf{f}_k}{h} \qquad \textit{for all } k \in \mathbb{Z}.$$

The pointwise backward difference operator $D_h^- : \ell^2(\mathbb{Z}) \longrightarrow \ell^2(\mathbb{Z})$ is analogously defined by

$$\left(D_h^- \mathbf{f}\right)_k = \frac{\mathbf{f}_k - \mathbf{f}_{k-1}}{h} \qquad \textit{for all } k \in \mathbb{Z}.$$

Alternatively we can write the pointwise forward and backward difference in terms of shift operators and the identity operator:

Lemma 4.6 (Differences with Shift Operators) *The forward difference operator can be written as the difference of a left shift operator and the identity I scaled with $\frac{1}{h}$:*

$$D_h^+ = \frac{1}{h}(L - I). \tag{4.1}$$

The backward difference operator is

$$D_h^- = \frac{1}{h}(I - R) = R D_h^+. \tag{4.2}$$

We now calculate powers of D_h^+ and D_h^- to get higher differences and approximations of higher derivatives.

Lemma 4.7 (Higher Differences) *The higher differences can be calculated as*

$$\begin{aligned} \left(D_h^+\right)^m &= \frac{1}{h^m} \sum_{l=0}^{m} (-1)^{m-l} \binom{m}{l} L^l \quad \textit{and} \\ \left(D_h^-\right)^m &= R^m \left(D_h^+\right)^m . \end{aligned}$$

Proof: With the observation that L and I commute, we get by taking powers of equation (4.1) with the binomial formula

$$\begin{aligned} \left(D_h^+\right)^m &= \frac{1}{h^m}(L - I)^m \\ &= \frac{1}{h^m} \sum_{l=0}^{m} \binom{m}{l} L^l (-I)^{m-l} \\ &= \frac{1}{h^m} \sum_{l=0}^{m} (-1)^{m-l} \binom{m}{l} L^l. \end{aligned}$$

The second statement follows from equation (4.2). With Lemma 4.2 it is clear that the shifts and the difference operators commute and therefore

$$\left(D_h^-\right)^m = \left(R D_h^+\right)^m = R^m \left(D_h^+\right)^m$$

which is our second statement. □

From Lemma 4.7 we obtain the coefficients for the values of $\mathbf{f}$ that are needed to compute an approximation of the mth derivative at the point x:

$$\left(\left(D_h^+\right)^m \mathbf{f}\right)_0 = \sum_{l=0}^{m} \frac{(-1)^{m-l}}{h^m} \binom{m}{l} \mathbf{f}_l .$$

We see that the mth finite difference is a linear combination of the values of f at the $m+1$ points $x, x+h, \ldots, x+mh$. The coefficients of these values are the binomial coefficients divided by h^m and with alternating signs such that the last summand $\frac{1}{h^m} f(x+mh)$ has positive sign.

Remark 4.8 (Approximation of Higher Derivatives) The property that this expression $\left(\left(D_h^+\right)^m f\right)(x)$ computed of the values of f at the $m+1$ equidistant points $x, \ldots, x+mh$ yields an approximation for $f^{(m)}$ at the central point $\left(x + \frac{mh}{2}\right)$ is mentioned in [14] and [24], for example. A proof based on polynomial interpolation can be found in [24].

The next remark yields an interpretation of finite differences in terms of local polynomial interpolation. It should point out the approach behind finite differences in comparison to the spectral methods presented in the next section.

Remark 4.9 (Differences and Local Interpolation) The first forward difference can also be seen as the exact derivative of the local interpolation of f with a polynomial of degree 1 at an arbitrary point between x and $x+h$. Analogously to get the mth forward difference a local interpolation polynomial p of degree m with the $m+1$ equidistant grid points $x, x+h, \ldots, x+mh$ is build. The mth derivative of p at the point $x + \frac{mh}{2}$ is the mth difference. We refer to [24] for details.

Lemma 4.10 (Adjoint Operator) *For the mth forward difference $\left(D_h^+\right)^m$ we get the adjoint operator $\left(-D_h^-\right)^m$.*

Proof: For all $\mathbf{f}, \mathbf{g} \in \ell^2$ Lemma 4.2 and Lemma 4.6 show that

$$\begin{aligned} \langle \left(D_h^+\right)^m \mathbf{f}, \mathbf{g} \rangle &= \frac{1}{h^m} \langle (L-I)^m \mathbf{f}, \mathbf{g} \rangle \\ &= \frac{1}{h^m} \langle \mathbf{f}, (R-I)^m \mathbf{g} \rangle \\ &= \langle \mathbf{f}, \left(-D_h^-\right)^m \mathbf{g} \rangle, \end{aligned}$$

and so we get the adjoint operator as given above. □

Remark 4.11 (Finite Differences as Convolution) The iterated finite difference operator $\left(D_h^+\right)^m : \ell^2(\mathbb{Z}) \longrightarrow \ell^2(\mathbb{Z})$ from the proof of Lemma 4.7 is linear and shift invariant. Thus can also be expressed as a discrete convolution

$$(\mathbf{d}^m * \mathbf{f})_k = \sum_{l \in \mathbb{Z}} \mathbf{d}_{k-l}^m f_l$$

with the convolution kernel

$$\mathbf{d}_l^m = \begin{cases} \frac{(-1)^{m-l}}{h^m} \binom{l}{m} & \text{if } -l \in \{-m, \ldots, 0\} \\ 0 & \text{else.} \end{cases}$$

This notation can also be used to deduce implementations of different boundary conditions with discrete methods.

We will give two examples for these notation. The component with index 0 is bold printed.

$$d_h^+ = \frac{1}{h} (\ldots, 0, 1, \mathbf{-1}, 0, \ldots).$$

We see that the convolution with d_h^+ yields a sequence consisting of the forward differences of f at the points x_l for all $l \in \mathbb{Z}$:

$$\left(d_h^+ * f\right)_l = \sum_{k\in\mathbb{Z}} \left(d_h^+\right)_{l-k} f_k = \frac{f_{l+1} - f_l}{h} = \left(D_h^+ f\right)(x_l).$$

The approximation for the second derivative written as convolution stencil looks like this:

$$d_h^2 = \frac{1}{h^2}\left(\ldots, 0, 1, -2, \mathbf{1}, 0, \ldots\right).$$

When we consider finite differences as exact derivatives of polynomials it is clear how they should behave if the considered function f is a polynomial. The following proposition and remark will confirm this assumption.

Proposition 4.12 (First Difference of Polynomials) *Let p be a real polynomial of degree $m \in \mathbb{N}$, i. e. $p(x) = \sum\limits_{k=0}^{m} a_k x^k$ with $a_k \in \mathbb{R}$ for all $k \in \{0, \ldots, m\}$ and $a_m \neq 0$. Let $h > 0$ be an arbitrary step size and $x \in \mathbb{R}$. The forward difference of p with step size h is a polynomial of degree $m - 1$.*

Proof: We compute the forward difference with step size h of p at point x

$$\begin{aligned}
\left(D_h^+ p\right)(x) &= \frac{p(x+h) - p(x)}{h} \\
&= \frac{1}{h}\sum_{k=0}^{m} a_k \left((x+h)^k - x^k\right) \\
&= \frac{1}{h}\sum_{k=0}^{m} a_k \left(\left(\sum_{l=0}^{k} \binom{k}{l} x^{k-l} h^l\right) - x^k\right) \\
&= \frac{1}{h}\sum_{k=0}^{m} a_k \left(\sum_{l=1}^{k} \binom{k}{l} x^{k-l} h^l + \binom{k}{0} x^k h^0 - x^k\right) \\
&= \frac{1}{h}\sum_{k=0}^{m}\sum_{l=0}^{k-1} a_k \binom{k}{l+1} x^{k-1-l} h^l
\end{aligned}$$

The exponent of x gets maximal for $l = 0$ and $k = m$. The summand with the highest exponent of x is

$$a_m \binom{m}{1} x^{m-1} h^0 \quad = \quad m\, a_m\, x^{m-1}$$

and does not vanish since we assumed that $a_m \neq 0$. This computation does not make any use of the particular point x. We conclude that the forward difference of a polynomial of degree m is a polynomial of degree $m - 1$. □

Remark 4.13 (Higher Differences of Polynomials) One can also get a statement about higher differences of polynomials: Iterating the argumentation from the preceeding lemma shows that the mth forward difference of a polynomial of degree m is constant (or a polynomial of degree 0). The above computations even allow us to give the exact value of this difference which is $m!\ a_m$. It is clear that the $(m+1)$th difference will vanish.

4.2.1 Matrix Notation

As already indicated above we also introduce a matrix notation for finite difference derivative approximation. By multiplying the matrix $D^F_{1,n} \in \mathbb{R}^{(n-1)\times n}$ defined by

$$D^F_{1,n} := \frac{1}{h}\begin{pmatrix} -1 & 1 & 0 & \dots & 0 \\ 0 & -1 & 1 & \ddots & \vdots \\ \vdots & \ddots & \ddots & \ddots & 0 \\ 0 & \dots & 0 & -1 & 1 \end{pmatrix}$$

with a vector $\mathbf{f}$, the product

$$D^F_{1,n}\mathbf{f} = \left(\frac{f(x_2)-f(x_1)}{h}, \dots, \frac{f(x_n)-f(x_{n-1})}{h}\right)$$

yields an approximation of the derivative f' at the equidistant grid points

$$\frac{x_1+x_2}{2}, \dots, \frac{x_{n-1}+x_n}{2}.$$

The F in $D^F_{1,n}$ stands for derivative approximation with **F**inite differences. For a second derivative approximation we iterate the first derivative taking care of the right matrix dimensions and get

$$D^F_{1,n-1}D^F_{1,n}\mathbf{f} = \left(\frac{f(x_3)-2f(x_2)+f(x_1)}{h^2}, \dots, \frac{f(x_n)-2f(x_{n-1})+f(x_{n-2})}{h^2}\right).$$

This approximates the second derivative f'' at the grid points $x_2, \dots, x_{n-1}$. We define the matrix $D^F_{2,n} \in \mathbb{R}^{(n-2)\times n}$ as

$$D^F_{2,n} = D^F_{1,n-1}D^F_{1,n} = \frac{1}{h^2}\begin{pmatrix} 1 & -2 & 1 & 0 & \dots & 0 \\ 0 & 1 & -2 & 1 & \ddots & \vdots \\ \vdots & \ddots & \ddots & \ddots & \ddots & 0 \\ 0 & \dots & 0 & 1 & -2 & 1 \end{pmatrix}.$$

This can be extended to higher orders with the

Definition 4.14 (Finite Differences as Matrix) *For $n \in \mathbb{N}\setminus\{0\}$ we define the finite difference derivative approximation of order $m \in \{1, \dots, n-1\}$ as the $(n-m)\times n$ - matrix*

$$D^F_{m,n} = D^F_{1,n-m+1} \cdot \dots \cdot D^F_{1,n}.$$

Remembering Remark 4.8 we note that multiplying the matrix $D^F_{m,n}$ with a vector $\mathbf{f} \in \mathbb{R}^n$

$$D^F_{m,n}\begin{pmatrix} \mathbf{f}_0 \\ \vdots \\ \mathbf{f}_{n-1} \end{pmatrix} = \begin{pmatrix} \left(\left(D^+_h\right)^m \mathbf{f}\right)_0 \\ \vdots \\ \left(\left(D^+_h\right)^m \mathbf{f}\right)_{n-1-m} \end{pmatrix} \approx \begin{pmatrix} f^{(m)}\left(x_0 + \frac{mh}{2}\right) \\ \vdots \\ f^{(m)}\left(x_{n-1-m} + \frac{mh}{2}\right) \end{pmatrix}$$

yields an approximation of the mth derivative of f at the points

$$x_0 + \frac{mh}{2}, x_1 + \frac{mh}{2}, \dots, x_{n-1-m} + \frac{mh}{2}.$$

To deduce an upper bound for the spectral norm we use the theorem of Gershgorin (see [26], theorem 6.9.4).

Theorem 4.15 (Gershgorin) *The union of all discs*

$$K_i := \left\{ \mu \in \mathbb{C};\ |\mu - a_{ii}| \leq \sum_{\substack{k=1\\k\neq i}}^{n} |a_{ik}| \right\} \quad \textit{for } i = 1, 2, \ldots, n$$

contains all eigenvalues of the $n \times n$ matrix $A = (a_{ik})$.

We use this to prove the following

Lemma 4.16 (Spectral Norm Estimate) *For the spectral norm of the matrix $D^F_{m,n}$ the inequality*

$$\left\| D^F_{m,n} \right\|_2 \leq \frac{2^m}{h^m}$$

holds.

Proof: First we consider the spectral norm of $D^F_{1,n}$. We see that

$$\left(D^F_{1,n}\right)^T D^F_{1,n} = \frac{1}{h^2} \begin{pmatrix} 1 & -1 & 0 & \ldots & 0 \\ -1 & 2 & -1 & \ddots & \vdots \\ 0 & \ddots & \ddots & \ddots & 0 \\ \vdots & \ddots & -1 & 2 & -1 \\ 0 & \ldots & 0 & -1 & 1 \end{pmatrix},$$

and with the Theorem 4.15 of Gershgorin we state that

$$\sigma\left(\left(D^F_{1,n}\right)^T D^F_{1,n} \right) \subseteq \left\{ \mu \in \mathbb{C};\ \left|\mu - \frac{2}{h^2}\right| \leq \frac{2}{h^2} \quad \vee \quad \left|\mu - \frac{1}{h^2}\right| \leq \frac{1}{h^2} \right\}$$

where σ denotes the set of all eigenvalues (the spectrum). As a product of the form $A^T A$ the matrix is also symmetric and therefore all eigenvalues are real, and we get

$$\sigma\left(\left(D^F_{1,n}\right)^T D^F_{1,n} \right) \subseteq \left[0, \frac{4}{h^2}\right].$$

For the spectral radius of $D^F_{1,n}$ this means

$$\left\| D^F_{1,n} \right\|_2 \leq \frac{2}{h}.$$

We state that this estimate is independent of the dimension n. Since the spectral norm is submultiplicative we state that

$$\begin{aligned} \left\| D^F_{m,n} \right\|_2 &\leq \left\| D^F_{1,n-m+1} \right\|_2 \cdot \ldots \cdot \left\| D^F_{1,n} \right\|_2 \\ &\leq \frac{2^m}{h^m}, \end{aligned}$$

and the estimate is proven. □

Remark 4.17 (Restriction of the Operator) We can also understand the matrix $D^F_{m,n}$ as a restriction of the pointwise forward difference operator $\left(D^+_h\right)^m$ onto the space M^{n-m} from Definition 4.3. Identifying $\mathbb{R}^n$ with M^n we can write the matrix as the operator

$$P_{M^{n-m}} \left(D^+_h\right)^m.$$

Remark 4.18 (Transposed Matrix and Boundary Conditions) We will use the matrix $D^F_{m,n}$ in discrete energy functionals in Section 5.1. In the gradient of such a functional the transposed matrix $\left(D^F_{m,n}\right)^T$ occurs in a natural way. We would like to understand some more properties of this matrix. Since it is a matrix with real entries it can be expressed as adjoint operator of the matrix $D^F_{m,n}$. Thus we consider with Lemma 4.10 and Lemma 4.7 the adjoint operator

$$\begin{aligned}\left(P_{M^{n-m}}\left(D_h^+\right)^m\right)^* &= (-1)^m\left(D_h^-\right)^m P_{M^{n-m}}\\ &= (-1)^m R^m\left(D_h^+\right)^m P_{M^{n-m}}.\end{aligned}$$

Basically this is a shifted version of the mth difference multiplied with $(-1)^m$. We would like to investigate in detail how this operator acts on a sequence $\mathbf{f}$: The projection $P_{M^{n-m}}$ maps all entries $\mathbf{f}_k$ of $\mathbf{f}$ with indices $k \notin \{0,\dots,n-m-1\}$ to zero. Using the formula from Lemma 4.7 it follows that

$$\begin{aligned}\left(\left(D_h^+\right)^m P_{M^{n-m}}\mathbf{f}\right)_k &= \sum_{l=0}^{m}\frac{(-1)^{m-l}}{h^m}\binom{m}{l}\left(P_{M^{n-m}}\mathbf{f}\right)_{k+l}\\ &= \sum_{l=0}^{m}\frac{(-1)^{m-l}}{h^m}\binom{m}{l}\mathbf{f}_{k+l}\\ &= \left(\left(D_h^+\right)^m\mathbf{f}\right)_k \qquad \text{for all } k\in\{0,\dots,n-2m-1\}.\end{aligned}$$

For $k \le -m-1$ or $k \ge n-m$ we can also state that

$$\left(\left(D_h^+\right)^m P_{M^{n-m}}\mathbf{f}\right)_k = 0,$$

since only zero entries of $P_{M^{n-m}}\mathbf{f}$ are taken into consideration to compute the mth difference. The interesting regions are now $-m \le k \le -1$ and $n-m+1 \le k \le n$. Here nonzero entries of $\mathbf{f}$ may occur, and also zeros generated by the projection are involved. One could consider the values calculated there as approximation of the mth derivative of f under the assumption of further conditions. For example it is

$$\left(\left(D_h^+\right)^m P_{M^{n-m}}\mathbf{f}\right)_{-1} = \left(\left(D_h^+\right)^m\mathbf{f}\right)_{-1} - \frac{(-1)^m}{h^m}\mathbf{f}_{-1}.$$

Since $\frac{(-1)^m}{h^m}$ is nonzero, the value $\left(\left(D_h^+\right)^m P_{M^{n-m}}\mathbf{f}\right)_{-1}$ can be considered as the approximation of $f^{(m)}\left(x_0+\frac{mh}{2}\right)$ under the assumption that $f(x_{-1})=0$.

More generally for $k\in\{1,\dots,m\}$ it is

$$\left(\left(D_h^+\right)^m P_{M^{n-m}}\mathbf{f}\right)_{-k} = \left(\left(D_h^+\right)^m\mathbf{f}\right)_{-k} - \sum_{l=0}^{k-1}\frac{(-1)^{m-l}}{h^m}\binom{m}{l}\mathbf{f}_{l-k}.$$

We can state that $\left(\left(D_h^+\right)^m P_{M^{n-m}}\mathbf{f}\right)_{-k}$ can be considered as the approximation of $f^{(m)}\left(x_{-k}+\frac{mh}{2}\right)$ under the assumption that

$$\sum_{l=0}^{k-1}\frac{(-1)^{m-1}}{h^m}\binom{m}{l}\mathbf{f}_{l-k} = 0.$$

We interpret this assumption as a discretisation of the condition

$$f^{(k-1)}(\xi) = 0 \qquad \text{for } \xi\in\left(x_0, x_0-\frac{mh}{2}\right)$$

what can be understood as the discrete version of the boundary condition

$$f^{(k-1)}(a) = 0$$

for the left boundary a of our interval. These considerations can be made for the right boundary in the same way.

4.2.2 Finite Differences and Polynomial Data Fitting

We would like to further investigate the connection between polynomials and finite differences. Assume that the coefficients $a_0, \dots a_{m-1}$ of a polynomial

$$p(x) = \sum_{i=0}^{m-1} a_i x^i$$

of degree $m-1$ are given. The evaluation of p at a given point $x \in \mathbb{R}$ can be written as the scalar product of the two vectors $\mathbf{a}$, $\mathbf{x} \in \mathbb{R}^m$ defined as

$$\begin{array}{rcl} \mathbf{a} & := & (a_0, a_1, \dots, a_{m-1}) \\ \mathbf{x} & := & \left(x^0, x^1, \dots, x^{m-1}\right). \end{array}$$

Taking the scalar product of a with different x means evaluating the polynomial at different points. For $n \geq m$ we define the submatrices $V_n^m \in \mathbb{R}^{n \times m}$ of the Vandermonde matrix as

$$V_n^m := \begin{pmatrix} 1 & x_0 & x_0^2 & \dots & x_0^{m-1} \\ 1 & x_1 & x_1^2 & \dots & x_1^{m-1} \\ \vdots & \vdots & \vdots & & \vdots \\ 1 & x_{n-1} & x_{n-1}^2 & \dots & x_{n-1}^{m-1} \end{pmatrix}.$$

The matrix-vector product $V_n^m \mathbf{a}$ yields the evaluation of the polynomial p represented by its coefficients in $\mathbf{a}$ evaluated at the points $x_0, \dots, x_{n-1}$.

Conversely, one can start with grid points $x_0, \dots, x_{n-1}$ and given values $\mathbf{y} := (y_0, \dots, y_{n-1})^T$. If there exists a polynomial p of degree less than m with $p(x_i) = y_i$ for all $i \in \{0, \dots, n-1\}$ it can be found as the solution of the linear equation system

$$V_n^m \mathbf{a} = \mathbf{y}.$$

If the points $x_0, \dots, x_{n-1}$ are pairwise different the matrix V_n^k has full rank m and if a solution exists it is unique. For details on matrices of Vandermonde type refer to [15] and [35]. If no solution exists we can solve the according least square problem

$$\min_{\mathbf{a} \in \mathbb{R}^n} \|V_n^m \mathbf{a} - \mathbf{y}\|_2$$

to get the best possible solution in the sense of least squares. It is well-known that this least-square approximation can be expressed as the orthogonal projection of $\mathbf{y}$ onto the range of V_n^k, i. e.

$$\mathbf{a}^* = P_{\mathrm{ran}(V_n^m)} \mathbf{y}.$$

We refer to [26] for more information on the topic least-square data fitting and generalised inverse.

The following proposition builds up the link between finite differences and least square polynomial data fitting:

Proposition 4.19 (Kernel of $D_{n,m}^F$) *The kernel of the finite difference approximation matrix $D_{m,n}^F$ is the subspace of $\mathbb{R}^n$ consisting of all equidistant polynomial evaluations for polynomials with degree less than m, i. e.*

$$\ker\left(D_{m,n}^F\right) = \mathrm{ran}\ \left(V_n^m\right).$$

Proof: First we show that $\mathrm{ran}\ (V_n^m) \subseteq \ker\left(D_{m,n}^F\right)$. We start with $\mathbf{y} \in \mathrm{ran}\ (V_n^m)$. There exists a polynomial p of degree less than m with $p(x_i) = y_i$.

It follows from Proposition 4.12 that applying the forward difference k times yields zero, and $\mathbf{y}$ is in the kernel of $D^F_{m,n}$.

Now we show the opposite direction $\ker\left(D^F_{m,n}\right) \subseteq \text{ ran } (V^m_n)$. Assume $\mathbf{y} \in \ker\left(D^F_{m,n}\right)$ and y is a function with the interpolation property

$$y(x_i) = \mathbf{y}_i \quad \text{for all } i \in \{0, \dots, n-1\}.$$

From the definition of $D^F_{m,n}$ we know that this is equivalent to

$$\left(\left(D^+_h\right)^m y\right)(x) = 0 \quad \text{for all } x \in \{x_1, \dots, x_{n-m}\}$$

and for any function y with the interpolation property

$$y(x_i) = \mathbf{y}_i \quad \text{for all } i \in \{0, \dots, n-1\}.$$

Since the higher differences $\left(\left(D^+_h\right)^k y\right)(x_0)$ with $k \in \{m+1, \dots, n-1\}$ can be expressed as linear combinations of the differences with order m, we see that

$$\left(\left(D^+_h\right)^k y\right)(x_0) = 0 \quad \text{for all} \quad k \in \{m+1, \dots, n-1\}.$$

It is well-known that there exists a polynomial $p(x) = \sum\limits_{k=0}^{n-1} a_k x^k$ of degree less than n with the interpolation property

$$p(x_i) = \mathbf{y}_i \quad \text{for all } i \in \{1, \dots, n\}.$$

This polynomial is uniquely determined. Applying the forward difference $n-1$ times yields a constant

$$\left(D^+_h\right)^{n-1} p = (n-1)!\, a_{n-1}.$$

Since the difference is zero, the leading coefficient also vanishes, and the degree of p is less than $n-1$. We can iterate this argumentation as long as the difference $\left(D^+_h\right)^l$ is zero. It follows that p has degree less than or equal $m-1$. This is equivalent to our claim that $\mathbf{y} \in \text{ ran } (V^m_n)$. □

4.2.3 Neumann Boundary Conditions

In image processing problems often Neumann boundary conditions are assumed. We have already seen a derivative approximation of second order with Neumann boundary conditions in the proof of Lemma 4.16. The matrix

$$D^{FN}_{2,n} := -\left(D^F_{1,n}\right)^T D^F_{1,n} = \frac{1}{h^2}\begin{pmatrix} -1 & 1 & 0 & \dots & 0 \\ 1 & -2 & 1 & \ddots & \vdots \\ 0 & \ddots & \ddots & \ddots & 0 \\ \vdots & \ddots & 1 & -2 & 1 \\ 0 & \dots & 0 & 1 & -1 \end{pmatrix}$$

gives a pointwise approximation of the second derivative. At the boundary pixels the second derivative is approximated under the assumption that $\mathbf{f}_{-1} = \mathbf{f}_0$ and $\mathbf{f}_{n-1} = \mathbf{f}_n$ respectively. This is a finite difference discretisation for the conditions $f'(a) = f'(b) = 0$. Instead of the matrices considered above, here we really have an $n \times n$ matrix, that means we obtain as much derivative approximations as we have function values. This attempt does only work for the approximation of second derivatives, in general we use the

Definition 4.20 (Matrix Notation) *For $n \in \mathbb{N} \setminus \{0\}$ we define the matrix for derivative approximation of order $m \in \{1, \ldots, n-1\}$ with Neumann boundary conditions as*

$$D^{FN}_{m,n} := \begin{cases} \left(D^{FN}_{2,n}\right)^l & \text{if } m = 2l \\ D^F_{1,n} \left(D^{FN}_{2,n}\right)^l & \text{if } m = 2l+1. \end{cases}$$

We see that $D^{FN}_{m,n}$ is an $n \times n$-matrix for even derivative order m and an $(n-1) \times n$-matrix for odd m.

Remark 4.21 (Transposed Matrix) For even m the matrix $D^{FN}_{m,n}$ is symmetric, and therefore the transposed matrix also approximates the mth derivative. For odd m we get an approximation of the negated mth derivative. Altogether one can write that multiplication with the transposed matrix $\left(D^{FN}_{m,n}\right)$ approximates $(-1)^m$ times the mth derivative similar to the matrix for natural boundary conditions.

An estimate for the spectral norm can be obtained directly from the proof of lemma (4.16):

Lemma 4.22 (Spectral Norm) *For the matrices defined above we have the spectral norm estimate*

$$\left\|D^{FN}_{m,n}\right\|_2 \leq \frac{2^m}{h^m}.$$

From linear algebra we know that for a real matrix A the identity

$$\text{ran } A = \left(\ker A^T\right)^\perp$$

holds. It follows that $\ker A^T A = \ker A$ and this proves the following

Lemma 4.23 (Kernel of $D^{FN}_{m,n}$) *The kernel of the finite difference approximation with Neumann boundary conditions $D^{FN}_{m,n}$ consists of all constant vectors.*

Proof: $D^{FN}_{m,n}$ is defined as product of $D^F_{1,n}$ followed by the transposed matrix and vice versa. We see that

$$\ker\left(D^{FN}_{m,n}\right) = \ker\left(D^F_{1,n}\right) = \langle(1, 1, \ldots, 1)\rangle.$$

Here $\langle x\rangle$ denotes the subspace spanned from the vector $x \in \mathbb{R}^n$. In this case the subspace consists of all constant vectors. □

4.2.4 The Two-Dimensional Case

So far we have only considered the one-dimensional case. For image restoration algorithms we also need approximations for partial derivatives of functions depending on two spatial variables. In Example 2.14 we have considered a class of energy functionals depending on the Laplacian of our image f. We will shortly present the methods we use to approximate the Laplacian with finite differences. Details can be found in [11] and [25], for example. In most of the cases we use the classical and simplest way to approximate a Laplacian with finite differences which is given in stencil notation by

$$\Delta f(x_i, y_j) \approx \frac{1}{h^2} \begin{array}{|c|c|c|} \hline & 1 & \\ \hline 1 & \mathbf{-4} & 1 \\ \hline & 1 & \\ \hline \end{array} \cdot \mathbf{f}_{i,j}.$$

At all inner points we can approximate the Laplacian with this stencil. This can also be written in matrix form. For natural boundary conditions we use the transposed matrix as indicated in the one-dimensional case.

For Neumann boundary conditions we have to distinguish between inner pixels, boundary pixels and corner pixels. The boundary pixels use a different stencil which is noted for an upper boundary pixel:

$$\Delta f(x_i, y_0) \approx \frac{1}{h^2}\begin{array}{|c|c|c|}\hline 1 & \mathbf{-3} & 1 \\ \hline & 1 & \\ \hline\end{array} \cdot \mathbf{f}_{i,0}$$

under the assumption that $\frac{\partial f}{\partial y}(y_0) = 0$. We also note a corner pixel stencil

$$\Delta f(x_0, y_0) \approx \frac{1}{h^2}\begin{array}{|c|c|}\hline \mathbf{-2} & 1 \\ \hline 1 & \\ \hline\end{array} \cdot \mathbf{f}_{0,0}$$

so we have given an example for each of the three pixel classes. If we write down the corresponding matrix L for Neumann boundary conditions we see that the values in a stencil appear in the row that corresponds to the actual pixel. The bold printed entry stands in the diagonal of the matrix. With Gershgorin's Theorem 4.15 we deduce that the eigenvalues of L are contained in the set

$$\left\{\mu \in \mathbb{C};\ \left|\mu + \frac{4}{h^2}\right| \le \frac{4}{h^2} \quad \vee \quad \left|\mu + \frac{3}{h^2}\right| \le \frac{3}{h^2} \quad \vee \quad \left|\mu + \frac{2}{h^2}\right| \le \frac{2}{h^2}\right\}.$$

From the stencils one can also deduce that L is symmetric, and thus all eigenvalues are real. We see that

$$\sigma(L) \subseteq \left[-\frac{8}{h^2}, 0\right]. \tag{4.3}$$

We note that the entries of each row of L have sum 0, that means the constant image is in the kernel of L.

We also consider a second commonly used approximation of the Laplacian for which the leading error term is rotationally invariant:

$$\Delta f(x_i, y_j) \approx \frac{1}{6h^2}\begin{array}{|c|c|c|}\hline 1 & 4 & 1 \\ \hline 4 & \mathbf{-20} & 4 \\ \hline 1 & 4 & 1 \\ \hline\end{array} \cdot \mathbf{f}_{i,j}.$$

Again we give an example how to treat an upper boundary pixel for Neumann boundary conditions:

$$\Delta f(x_i, y_0) \approx \frac{1}{6h^2}\begin{array}{|c|c|c|}\hline 5 & \mathbf{-16} & 5 \\ \hline 1 & 4 & 1 \\ \hline\end{array} \cdot \mathbf{f}_{i,0}$$

under the assumption that $\frac{\partial f}{\partial y}(y_0) = 0$. For the upper left corner the following stencil is used:

$$\Delta f(x_0, y_0) \approx \frac{1}{6h^2}\begin{array}{|c|c|}\hline \mathbf{-11} & 5 \\ \hline 5 & 1 \\ \hline\end{array} \cdot \mathbf{f}_{0,0}.$$

Again we consider the corresponding matrix $\tilde{L}$. Using Gershgorin's Theorem 4.15 we obtain that

$$\begin{aligned}\sigma\left(\tilde{L}\right) \quad \subseteq \quad & \left\{\mu \in \mathbb{C}\,;\ \left|\mu + \frac{20}{6h^2}\right| \le \frac{20}{6h^2} \quad \vee \quad \left|\mu + \frac{16}{6h^2}\right| \le \frac{16}{6h^2}\right. \\ & \left. \vee \quad \left|\mu + \frac{11}{6h^2}\right| \le \frac{11}{6h^2}\right\}.\end{aligned}$$

Again the matrix $\tilde{L}$ is symmetric and has real eigenvalues contained in the interval

$$\sigma\left(\tilde{L}\right) \subseteq \left[-\frac{20}{3h^2}, 0\right]. \tag{4.4}$$

We also have row sum 0 for $\tilde{L}$, and thus the constant image is in the kernel of $\tilde{L}$, too.

4.3 Spectral Methods

Finite differences are a local method to approximate the derivative of a function. Spectral methods can be seen as the global counterpart of them. The basic idea behind spectral methods is to find a global interpolating function for the function f and take the derivative of this approximation as an estimate for the derivative of f. This approach can be made with approximations in different function spaces.

Here we focus on the so-called Fourier spectral methods which use a trigonometric rational function

$$p(x) = \sum_{k=-n/2}^{n/2-1} \hat{f}_k \exp\left(\frac{2\pi i k}{n} x\right) \tag{4.5}$$

with coefficients $\hat{f}_k$ as approximation. We assume that our pixel number n is even in this case. The exact derivative of this function is a trigonometric rational function

$$p'(x) = \sum_{k=-n/2}^{n/2-1} \hat{f}_k \frac{2\pi i k}{n} \exp\left(\frac{2\pi i k}{n} x\right).$$

with coefficients $\hat{f}_k \frac{2\pi i k}{n}$. We see that the model algorithm for calculating a spectral derivative approximation consists of three steps:

1. Calculate the Fourier coefficients $\hat{\mathbf{f}}_k$ of p out of the given values $\mathbf{f}$ at the grid points.

2. Multiply these coefficients with $\frac{2\pi i k}{n}$ to obtain the coefficients of the derivative p'.

3. Evaluate p' at the grid points.

It should also be mentioned that this approach also works for higher derivatives with powers of the coefficients in step (2.) since

$$p^{(m)}(x) = \sum_{k=-n/2}^{n/2-1} \hat{f}_k \left(\frac{2\pi i k}{n}\right)^m \exp\left(\frac{2\pi i k}{n} x\right).$$

A description of these method can be found in [29]. We would like to explain this general approach, point out some shortcomings and further investigate a commonly used way to circumvent them.

It is well-known that trigonometric interpolation and evaluation of phase polynomials can be performed with the discrete Fourier transform. Usually the discrete Fourier transform is written as

$$\hat{\mathbf{f}}_j := \frac{1}{\sqrt{n}} \sum_{k=0}^{n-1} \mathbf{f}_k \exp\left(-\frac{2\pi i j k}{n}\right) \qquad \text{for } j \in \{0, \ldots, n-1\}. \tag{4.6}$$

Then the $\hat{\mathbf{f}}_j$ are the coefficients of the unique phase polynomial p that satisfies the interpolation property

$$p(k) = \frac{1}{\sqrt{n}} \sum_{j=0}^{n-1} \hat{\mathbf{f}}_j \exp\left(\frac{2\pi i j k}{n}\right) = \mathbf{f}_k \qquad \text{for } k \in \{0, \ldots, n-1\}.$$

Remark 4.24 (Periodicity) We note that the approximation with trigonometric polynomials inherently introduces periodicity:

1. The trigonometric monomials $\exp\left(\frac{2\pi ij}{n}x\right)$ with $j \in \mathbb{Z}$ are n-periodic in x since
$$\exp\left(\frac{2\pi ij}{n}(x+n)\right) = \exp\left(\frac{2\pi ij}{n}x\right)\exp(2\pi ij) = \exp\left(\frac{2\pi ij}{n}x\right).$$
Therefore the trigonometric interpolation polynomial is n-periodic. Any function given at n discrete points will be periodically extended by trigonometric interpolation. Thus the periodic extension of the function values given by the vector $\mathbf{f} \in \mathbb{R}^n$ to a series $\mathbf{f} \in \ell^\infty$ which was already noted in Section 4.1 makes sense in this framework.

2. The expressions $\exp\left(\frac{2\pi ijk}{n}\right)$ for $j, k \in \mathbb{Z}$ are n-periodic in j and k. If we consider the periodic extension of $\mathbf{f}$ we also get n-periodic Fourier coefficients $\hat{f}_{j+n} = \hat{f}_j$ for all $j \in \mathbb{Z}$. We note that the coefficients of the polynomial (4.5) can be obtained by a shift from the coefficients calculated in (4.6) with a standard algorithm for discrete Fourier transform such as the Fast Fourier Transform.

Remark 4.25 (Choice of Indices) The periodicity obtained in the last remark imposes the question why a special reminder system modulo n is preferred in our considerations. With respect to the interpolation property it would not make any difference to consider a trigonometric polynomial
$$p(x) = \sum_{j=0}^{n-1} \hat{f}_k \exp\left(\frac{2\pi ij}{n}x\right)$$
instead of (4.5). One could even use the same Fourier coefficients for this polynomial. The reason is that the index sets
$$\left\{-\frac{n}{2}, \dots, \frac{n}{2}-1\right\} \quad \text{or} \quad \left\{-\frac{n}{2}+1, \dots, \frac{n}{2}\right\}$$
lead to the interpolations with least oscillations. The interpolation property is not violated by oscillations in between the considered points, but the derivative. This can be seen in terms of the total variation of the participating monomials
$$\int_0^n \left|\frac{2\pi ik}{n}\exp\left(\frac{2\pi ik}{n}x\right)\right| dx = \frac{2\pi}{n}|k|.$$
These values are minimal for the index sets given above.

4.3.1 Matrix Notation

First we write the discrete Fourier transform as matrix-vector multiplication. Let us consider the vector space $\mathbb{C}^n$ with the canonical scalar product
$$\langle \mathbf{f}, \mathbf{g} \rangle := \sum_{k=0}^{n-1} \mathbf{f}_k \overline{\mathbf{g}}_k.$$
The vectors
$$v_k := \frac{1}{\sqrt{n}}\left(\exp\left(\frac{2\pi i0k}{n}\right), \exp\left(\frac{2\pi i1k}{n}\right), \dots, \exp\left(\frac{2\pi i(n-1)k}{n}\right)\right)^T$$

for $k \in \{0, \ldots, n-1\}$ then build an orthonormal basis. The matrix F with the v_k as row vectors

$$F := \begin{pmatrix} v_0^T \\ v_1^T \\ \vdots \\ v_{n-1}^T \end{pmatrix}$$

is therefore unitary and can be considered as a change of the basis. The discrete Fourier transform of a vector $\mathbf{f}$ can be expressed as matrix-vector product

$$F\mathbf{f} = \left(\hat{\mathbf{f}}_0, \ldots, \hat{\mathbf{f}}_{n-1}\right)^T.$$

Since F is unitary the inverse of F is $\overline{F}^T$, and it is

$$\|F\|_2 = \left\|\overline{F}^T\right\|_2 = 1.$$

After calculating the coefficients we shift them as indicated above: We use a periodic shift

$$\begin{aligned} S : \left(\hat{\mathbf{f}}_0, \ldots, \hat{\mathbf{f}}_{n-1}\right) \quad &\longmapsto \quad \left(\hat{\mathbf{f}}_{\frac{n}{2}}, \ldots, \hat{\mathbf{f}}_{n-1}, \hat{\mathbf{f}}_0, \ldots, \hat{\mathbf{f}}_{\frac{n}{2}-1}\right) \\ &= \quad \left(\hat{\mathbf{f}}_{-\frac{n}{2}}, \ldots, \hat{\mathbf{f}}_{\frac{n}{2}-1}\right) \end{aligned}$$

to get the coefficients from our trigonometric interpolation function (4.5). This shift can be expressed with a permutation matrix S with $\|S\|_2 = 1$. This completes the first step from our model algorithm.

The second step is to obtain the coefficients of the derivative. This can be written as a diagonal matrix $D \in \mathbb{C}^{n \times n}$ defined as

$$D := \frac{2\pi i}{n} \operatorname{diag}\left(-\frac{n}{2}, \ldots, \frac{n}{2} - 1\right).$$

We note that the spectral norm of D is the maximum of the absolute values of its eigenvalues which are simply the diagonal entries:

$$\|D\|_2 = \left|\frac{2\pi i}{n} \frac{n}{2}\right| \cdot \left|-\frac{n}{2}\right| = \pi.$$

We should also note that the kernel of D has dimension 1 since exactly one diagonal entry is zero.

The third step then consists of the periodic shift S and the Fourier backtransform which implements the evaluation of the trigonometric polynomial.

We summarise this procedure to the first attempt for a spectral derivative approximation matrix:

Definition 4.26 (Spectral Derivative Approximation) *For $m, n \in \mathbb{N} \setminus \{0\}$ one can write the spectral approximation of the mth derivative as matrix $D^S_{m,n} \in \mathbb{C}^{n \times n}$ defined with*

$$D^S_{m,n} = \overline{F}^T S D^m S F.$$

We summarise the properties of these matrices.

Lemma 4.27 (Properties of $D^S_{m,n}$) *For all $m, n \in \mathbb{N} \setminus \{0\}$ the following properties hold:*

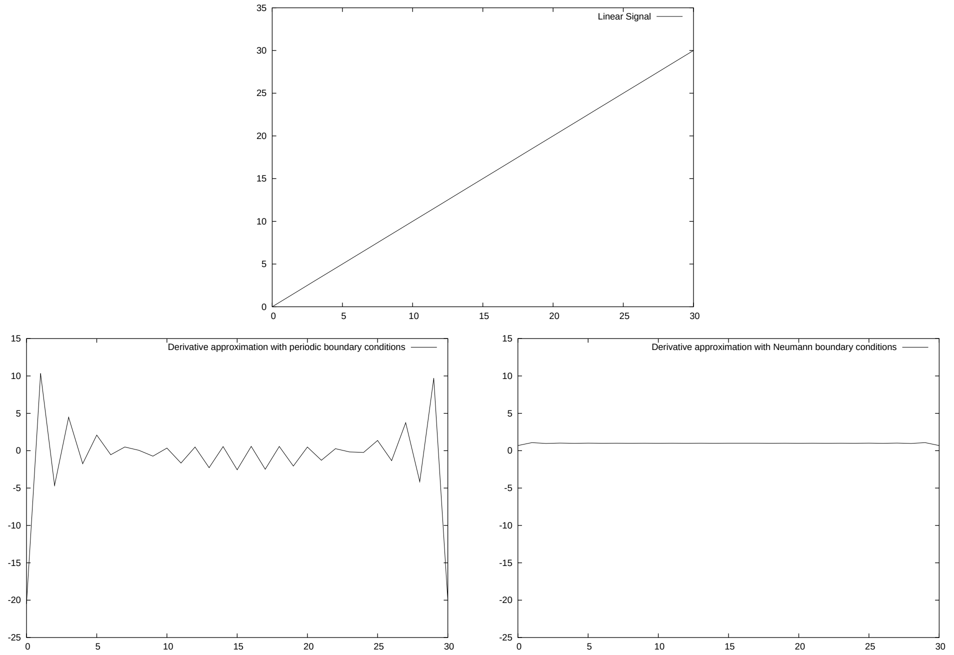

Figure 4.1: Spectral methods for derivative approximation and boundary conditions.

1. *The spectral norm is the same as for D^m:*

$$\left\| D^S_{m,n} \right\|_2 = \pi^m.$$

2. *The kernel consists of all constant vectors*

$$\ker D^S_{m,n} = \left\langle (1, 1, \ldots, 1)^T \right\rangle.$$

Proof: The first statement follows from the fact that F and S are unitary matrices. For the second statement we note that the entries in D are the eigenvalues of $D^S_{m,n}$, and the kernel is the eigenspace belonging to the eigenvalue zero. Only one entry in D (the one with index 0) is zero. The corresponding eigenvector to this eigenvalue is

$$(\exp(0), \ldots, \exp(0))^T = (1, \ldots, 1)^T$$

and so the second statement has been shown. □

4.3.2 Problems

There are two problems for the derivative approximation defined above:

Boundary Conditions

As already mentioned in Remark 4.24 the approximation of our signal introduces periodic boundary conditions. These are usually not adequate for our signal and image processing problems. Figure 4.1 shows an example to visualise the difference between periodic and Neumann boundary conditions.

The Highest Frequency and Imaginary Derivatives

The indices in the sum (4.5) are not symmetric around zero. The summand with index $-n/2$ has no counterpart with positive index. The corresponding vector in our matrix F is

$$\begin{aligned} v_{-n/2} &= \left(\exp\left(\frac{-2\pi i}{n}\left(-\frac{n}{2}\right)\left(-\frac{n}{2}\right)\right), \ldots, \exp\left(\frac{-2\pi i}{n}\left(-\frac{n}{2}\right)\left(\frac{n}{2}-1\right)\right)\right)^T \\ &= \left(\exp\left(\pi i\left(\frac{-n}{2}\right)\right), \ldots, \exp\left(\pi i\left(\frac{n}{2}-1\right)\right)\right)^T \\ &= (-1)^{\frac{n}{2}} \cdot (1, -1, \ldots, 1, -1)^T . \end{aligned}$$

This corresponds to the values of the monomial $p(x) = \exp(\pi i x)$ at the evaluation points $-n/2, \ldots, n/2-1$. We see that this monomial has the derivative $p'(x) = \pi i \exp(\pi i x)$ which is imaginary at these evaluation points. Since there is no counterpart with positive index $n/2$ the derivative approximation would be imaginary. One could simply avoid this problem by setting the Fourier coefficient to zero, but one should be aware of the fact that an alternating signal gets a zero derivative approximation this way.

We circumvent both of these problems by extending our signal to the double size, calculate the derivative approximation as defined above and restrict the signal to its original size again. This technique is commonly used in image processing to circumvent unwanted boundary artifacts caused by filters in the Fourier domain. We investigate how this works for derivative approximation.

Definition 4.28 (Extension and Restriction) *Let*

$$E := \begin{pmatrix} 1 & 0 & \ldots & 0 \\ 0 & \ddots & \ddots & \vdots \\ \vdots & \ddots & \ddots & 0 \\ \vdots & & 0 & 1 \\ \vdots & & 0 & 1 \\ \vdots & \iddots & \iddots & 0 \\ 0 & \iddots & \iddots & \vdots \\ 1 & 0 & \ldots & 0 \end{pmatrix} \quad \text{and } R := \begin{pmatrix} 1 & 0 & \ldots & 0 & 0 & \ldots & 0 \\ 0 & \ddots & \ddots & & \vdots & & \vdots \\ \vdots & \ddots & \ddots & 0 & \vdots & & \vdots \\ 0 & \ldots & 0 & 1 & 0 & \ldots & 0 \end{pmatrix}$$

be the extension of a signal and the corresponding restriction (or the projection of the first n components).

Lemma 4.29 (Spectral Norms) *Let $T \subset \mathbb{C}^{2n}$ be the subspace of all vectors $\mathbf{f} \in \mathbb{C}^{2n}$ with the property*

$$\|g_{2n-1-k}\| = \|g_k\| \quad \text{for all } k \in \{0, \ldots, n-1\}.$$

We note that ran $E \subset T$. For $E : \mathbb{C}^n \longrightarrow \mathbb{C}^{2n}$ and $R : T \longrightarrow \mathbb{C}^n$ we obtain the spectral norms

$$\|E\|_2 = \sqrt{2} \text{ and } \|R\|_2 = \frac{1}{\sqrt{2}}.$$

Proof: Since the spectral norm is compatible with the Euclidean vector norm we calculate that

$$\|E\mathbf{f}\| = \left(\sum_{k=0}^{2n-1} |(E\mathbf{f})_k|^2\right)^{\frac{1}{2}} = \left(2\sum_{k=0}^{n-1} |\mathbf{f}_k|^2\right)^{\frac{1}{2}} = \sqrt{2}\|\mathbf{f}\|$$

for all $\mathbf{f} \in \mathbb{C}^n$. For all $\mathbf{f} \in T$ we have

$$\|R\mathbf{f}\| = \left(\sum_{k=0}^{n-1} |R\mathbf{f}_k|^2\right)^{\frac{1}{2}} = \left(\frac{1}{2}\sum_{k=0}^{2n-1} |\mathbf{f}_k|^2\right)^{\frac{1}{2}} = \frac{1}{\sqrt{2}}\|\mathbf{f}\|.$$

So we have proven the matrix norms as given above. □

Now we need to show that ran $\left(D^S_{m,n}E\right) \subset T$. With this result we can conclude that the spectral norm for derivative approximation with extension of the signal is not greater than without extension:

$$\|RD^S_{m,n}E\|_2 \leq \frac{1}{\sqrt{2}}\|D^S_{m,n}\|_2\sqrt{2} = \|D^S_{m,n}\|.$$

We now show the claim from above by first computing the Fourier coefficients of the extended signal:

$$\begin{aligned}
\hat{f}_k &= \frac{1}{\sqrt{2n}}\sum_{j=0}^{2n-1} f_j \exp\left(\frac{-2\pi ijk}{2n}\right) \\
&= \frac{1}{\sqrt{2n}}\sum_{j=0}^{n-1} f_j \exp\left(\frac{-\pi ijk}{n}\right) + \frac{1}{\sqrt{2n}}\sum_{j=n}^{2n-1} f_{2n-1-j} \exp\left(\frac{-\pi ijk}{n}\right).
\end{aligned}$$

We perform an index transformation in the second sum introducing the new index $l = 2n - j - 1$ yielding

$$\begin{aligned}
\sum_{j=n}^{2n-1} f_{2n-1-j} \exp\left(\frac{-\pi ijk}{n}\right) &= \sum_{l=0}^{n-1} f_l \exp\left(\frac{-\pi i(2n-1-l)k}{n}\right) \\
&= \sum_{l=0}^{n-1} f_l \exp\left(\frac{\pi i(l+1)k}{n}\right) \exp\left(-2\pi ik\right).
\end{aligned}$$

We remember that $k \in \mathbb{Z}$ and thus $\exp(-2\pi ik) = 1$ and use this to recombine both sums in the above computation:

$$\begin{aligned}
\hat{f}_k &= \frac{1}{\sqrt{2n}}\sum_{j=0}^{n-1} f_j \exp\left(\frac{-\pi ijk}{n}\right) + \frac{1}{\sqrt{2n}}\sum_{l=0}^{n-1} f_l \exp\left(\frac{\pi ilk}{n}\right)\exp\left(\frac{\pi ik}{n}\right) \\
&= \frac{1}{\sqrt{2n}}\sum_{j=0}^{n-1} f_j \left[\exp\left(\frac{-\pi ijk}{n}\right) + \exp\left(\frac{\pi ijk}{n}\right)\exp\left(\frac{\pi ik}{n}\right)\right] \\
&= \frac{1}{\sqrt{2n}}\exp\left(\frac{\pi ik}{2n}\right)\sum_{j=0}^{n-1} f_j \left[\exp\left(\frac{-\pi i\left(j+\frac{1}{2}\right)k}{n}\right) + \exp\left(\frac{\pi i\left(j+\frac{1}{2}\right)k}{n}\right)\right] \\
&= \frac{1}{\sqrt{2n}}\exp\left(\frac{\pi ik}{2n}\right)\sum_{j=0}^{n-1} f_j 2\cos\left(\frac{\pi\left(j+\frac{1}{2}\right)k}{n}\right).
\end{aligned}$$

Now we consider the associated interpolating function

$$\begin{aligned}
p(x) &= \frac{1}{\sqrt{2n}}\sum_{k=-n}^{n-1} \hat{f}_k \exp\left(\frac{2\pi ik}{2n}x\right) \\
&= \frac{1}{\sqrt{2n}}\sum_{k=-n}^{n-1} \hat{f}_k \exp\left(\frac{\pi ik}{n}x\right)
\end{aligned}$$

$$
\begin{aligned}
&= \frac{1}{2n}\sum_{k=-n}^{n-1}\left(\exp\left(\frac{\pi ik}{2n}\right)\sum_{j=0}^{n-1} f_j 2\cos\left(\frac{\pi\left(j+\frac{1}{2}\right)k}{n}\right)\right)\exp\left(\frac{\pi ik}{n}x\right)\\
&= \frac{1}{n}\sum_{k=-n}^{n-1}\sum_{j=0}^{n-1} f_j\cos\left(\frac{\pi\left(j+\frac{1}{2}\right)k}{n}\right)\exp\left(\frac{\pi ik}{n}\left(x+\frac{1}{2}\right)\right).
\end{aligned}
$$

We would like to show that this interpolating function is symmetric to the axis $x = n - \frac{1}{2}$. We compute $p\left(n-\frac{1}{2}+x\right)$ and $p\left(n-\frac{1}{2}-x\right)$ to show equality up to complex conjugation.

$$
\begin{aligned}
&p\left(n-\frac{1}{2}+x\right)\\
&= \frac{1}{n}\sum_{k=-n}^{n-1}\sum_{j=0}^{n-1} f_j\cos\left(\frac{\pi\left(j+\frac{1}{2}\right)k}{n}\right)\exp\left(\frac{\pi ik}{n}\left(n-\frac{1}{2}+x+\frac{1}{2}\right)\right)\\
&= \frac{1}{n}\sum_{k=-n}^{n-1}\sum_{j=0}^{n-1} f_j\cos\left(\frac{\pi\left(j+\frac{1}{2}\right)k}{n}\right)(-1)^k\exp\left(\frac{\pi ik}{n}x\right)\\
&= \frac{1}{n}\sum_{k=-n}^{n-1}\sum_{j=0}^{n-1} f_j\cos\left(\frac{\pi\left(j+\frac{1}{2}\right)k}{n}\right)\overline{(-1)^k\exp\left(-\frac{\pi ik}{n}x\right)}\\
&= \frac{1}{n}\sum_{k=-n}^{n-1}\sum_{j=0}^{n-1} f_j\cos\left(\frac{\pi\left(j-\frac{1}{2}\right)k}{n}\right)\overline{\exp\left(\frac{\pi ik}{n}\left(n-\frac{1}{2}-x+\frac{1}{2}\right)\right)}\\
&= \overline{p\left(n-\frac{1}{2}-x\right)}.
\end{aligned}
$$

It follows that

$$
p^{(k)}\left(n-\frac{1}{2}+x\right) = (-1)^k\overline{p^{(k)}\left(n-\frac{1}{2}-x\right)}
$$

which implies the important fact that

$$
\left|p^{(k)}\left(n-\frac{1}{2}+x\right)\right| = \left|p^{(k)}\left(n-\frac{1}{2}-x\right)\right|.
$$

We note that the derivatives have the property needed to apply Lemma 4.29, and we can make our spectral norm estimate from above. Since the vector

$$
(-1,1,\ldots,-1,1)\in\mathbb{C}^{2n}
$$

is orthogonal to the range of E, we see that the according Fourier coefficient is zero. Thus we do not have the problem with the unsymmetric treatment of the highest frequency mentioned above.

4.3.3 Higher Dimensions

Spectral approximation of derivatives works analogously in higher dimensions, too. Here we use multivariate trigonometric polynomials to interpolate our data. In 2D such a polynomial looks like this:

$$
p(x,y) = \sum_{j=-n_x/2}^{n_x/2-1}\sum_{k=-n_y/2}^{n_y/2-1}\hat{f}_{j,k}\exp\left(\frac{2\pi ij}{n_x}x\right)\exp\left(\frac{2\pi ik}{n_y}y\right).
$$

The Laplacian of such a polynomial then is

$$\begin{aligned}\Delta p(x,y) &= \sum_{j=-n_x/2}^{n_x/2-1} \sum_{k=-n_y/2}^{n_y/2-1} \hat{f}_{j,k} \left(-\left(\frac{2\pi j}{n_x}\right)^2 - \left(\frac{2\pi k}{n_y}\right)^2 \right) \\ &\qquad \exp\left(\frac{2\pi i j}{n_x} x\right) \exp\left(\frac{2\pi i k}{n_y} y\right). \qquad (4.7)\end{aligned}$$

It is well-known that the discrete Fourier transform can be generalised to higher dimensions and the main properties remain true. Therefore one can also write the derivative approximation as a matrix of the form

$$D^S = \overline{F}^T SDSF$$

with the higher dimensional analogues of the matrices defined above. We note that the shift operators S in this case shift along each space direction. We should note that the matrix D has different entries in this case which cause a higher spectral norm. For the Laplacian in 2D we get this entries out of equation (4.7) as

$$-\left(\frac{2\pi j}{n_x}\right)^2 - \left(\frac{2\pi k}{n_y}\right)^2$$

with $j = -\frac{n_x}{2}, \ldots, \frac{n_x}{2} - 1$ and $k = -\frac{n_y}{2}, \ldots, \frac{n_y}{2} - 1$. Since F and S are unitary matrices these values are already the eigenvalues of D^S. We see that

$$\begin{aligned}\left\|D^S\right\|_2 &\leq \left| -\left(\frac{2\pi\left(-\frac{n_x}{2}\right)}{n_x}\right)^2 - \left(\frac{2\pi\left(-\frac{n_y}{2}\right)}{n_y}\right)^2 \right| \\ &\leq \frac{4\pi^2 n_x^2}{4n_x^2} + \frac{4\pi^2 n_y^2}{4n_y^2} \\ &\leq 2\pi^2.\end{aligned}$$

Analogously one could approximate the Laplacian in $\mathbb{R}^n$ which would lead to the upper bound $n\pi^2$ for the spectral norm of our derivative approximation matrix. As described for the one-dimensional case it is also possible to avoid artifacts caused by periodic boundary conditions by mirroring the image in all space directions before calculating the derivative approximation. With separability of the higher dimensional discrete Fourier transform one can obtain the same result as above: The product of the spectral norms of the extension and the restriction operation is 1.

Chapter 5

Discrete Filtering

In the last chapter we have discussed different ways to approximate derivatives including different discrete versions for the operators $\frac{d^m}{dx^m}$ and Δ. We now apply these to obtain discrete methods for signal and image restoration.

For one-dimensional signal restoration we start with the energy functional (2.19)

$$\mathcal{E}(u) = \int\limits_a^b \left((u-f)^2 + \alpha\,\varphi\left(\left(u^{(m)}\right)^2 \right) \right) dx$$

which was presented in Example 2.10. The corresponding Euler-Lagrange equation (2.20) is

$$0 = u - f + \alpha(-1)^m \frac{d^m}{dx^m} \left(\varphi'\left(\left(u^{(m)}\right)^2 \right) u^{(m)} \right).$$

In the two-dimensional case we will use the functional (2.35)

$$\mathcal{E}(u) = \int\limits_\Omega \left((u-f)^2 + \alpha\,\varphi\left((\Delta u)^2 \right) \right) dz$$

already discussed in Example 2.14. In this case we have the corresponding Euler-Lagrange equation (2.36)

$$0 = u - f + \alpha\,\Delta\left(\varphi'\left((\Delta u)^2 \right) \Delta u \right).$$

First one can directly discretise the energy functional. We will have a closer look at this approach in Section 5.1. With the Euler-Lagrange equations we can also treat the minimisation problem as solving a partial differential equation. Analogously to Section 3.1 we can discretise the Euler-Lagrange equation. This is especially useful for linear penalisers and will be carried out in detail in Section 5.2. We have already seen in Section 3.1 that it is also possible to consider parabolic equations from diffusion and diffusion-reaction type corresponding to the Euler-Lagrange equations. Discretisations for the diffusion equations will be the topic of Section 5.3.

5.1 Discrete Energy Functionals

Let $n, p \in \mathbb{N}\backslash\{0\}$ be positive numbers, $\mathbf{u}, \mathbf{f} \in \mathbb{R}^n$ and $A \in \mathbb{R}^{p\times n}$ a real $(p\times n)$-matrix with entries $A = (a_{ik})_{\substack{i=1,\dots,p\\k=1,\dots,n}}$. Let us denote the ith component of the matrix-vector product $A\mathbf{u}$ with

$$(A\mathbf{u})_i := \sum_{k=1}^{n} a_{ik}\mathbf{u}_k \quad (i = 1, \dots, p).$$

Consider the function $\mathcal{E}_d : \mathbb{R}^n \longrightarrow \mathbb{R}$ defined by

$$\mathcal{E}_d(\mathbf{u}) \quad := \quad \sum_{k=1}^{n}(\mathbf{u}_k - \mathbf{f}_k)^2 + \alpha \sum_{k=1}^{p} \varphi\left((A\mathbf{u})_k^2\right) \tag{5.1}$$

with $\alpha \geq 0$ and $\varphi \in C^2(\mathbb{R}_0^+)$. We are interested in minimising the value of $\mathcal{E}_d$ which is considered as a discrete energy functional.

Remark 5.1 (Choices of A) For the matrix A we can insert the derivative approximations from Chapter 4, for example. For one-dimensional purposes we could use $A = D_{m,n}^F$ from Definition 4.14 or $A = D_{m,n}^{FN}$ (see Definition 4.20). Embedding an image in a vector as discussed in Section 4.1 the function (5.1) can also be a discrete version for the two-dimensional functional (2.35). In this case we can use the matrices L or $\tilde{L}$ given in Section 4.2.4 instead of A.

To minimise the function $\mathcal{E}_d$ we have to compute the gradient of $\mathcal{E}_d$ at first. We compute the partial derivative of $\mathcal{E}_d$ with respect to every real variable $\mathbf{u}_i$:

$$\begin{aligned}
\frac{\partial \mathcal{E}_d}{\partial \mathbf{u}_i} &= \left(\frac{\partial}{\partial \mathbf{u}_i}\sum_{k=1}^{n}(\mathbf{u}_k - \mathbf{f}_k)^2\right) + \left(\alpha \frac{\partial}{\partial \mathbf{u}_i}\sum_{k=1}^{p}\varphi\left((A\mathbf{u})_k^2\right)\right) \\
&= 2(\mathbf{u}_i - \mathbf{f}_i) + \alpha\left(\frac{\partial}{\partial \mathbf{u}_i}\sum_{k=1}^{p}\varphi\left(\left(\sum_{l=1}^{n} a_{kl}\mathbf{u}_l\right)^2\right)\right) \\
&= 2(\mathbf{u}_i - \mathbf{f}_i) + \alpha\sum_{k=1}^{p}\varphi'\left((A\mathbf{u})_k^2\right) 2\left(\sum_{l=1}^{n} a_{kl}\mathbf{u}_l\right) a_{ki} \\
&= 2\left(\mathbf{u}_i - \mathbf{f}_i + \alpha\sum_{k=1}^{p} a_{ki}\varphi'\left((A\mathbf{u})_k^2\right)(A\mathbf{u})_k\right).
\end{aligned}$$

We define a diagonal matrix that contains the nonlinear terms so that we are able to write the gradient as matrix product.

Definition 5.2 (Nonlinearity Matrix) *Let $\Phi_A(\mathbf{u})$ denote the $(p \times p)$-diagonal matrix*

$$\Phi_A(\mathbf{u}) := \operatorname{diag}\left(\varphi'\left((A\mathbf{u})_k^2\right)\right)_{k=1,\dots,p} \in \mathbb{R}^{p\times p}. \tag{5.2}$$

With this definition the gradient $\nabla\mathcal{E}_d$ can be written as matrix product

$$\nabla\mathcal{E}_d \quad = \quad 2\left(\mathbf{u} - \mathbf{f} + \alpha A^T \Phi_A(\mathbf{u}) A\mathbf{u}\right). \tag{5.3}$$

A necessary condition for a minimum of $\mathcal{E}_d$ then is $\nabla\mathcal{E}_d = 0$ or equivalently

$$\left(I + \alpha A^T \Phi_A(\mathbf{u}) A\right)\mathbf{u} = \mathbf{f} \tag{5.4}$$

which can be seen as a discrete analogon to the Euler-Lagrange equations (2.20) or (2.36), respectively. In the next section we will discuss different discretisations of the Euler-Lagrange equations which in general lead to the same result. Here we see the natural appearance of the transposed matrix that was already mentioned in Remark 4.18. This can be seen as a motivation to consider the transposed matrices to all derivative approximations in the last chapter and to use it in discretisations in the next sections.

We also compute the entries of the Hessian of $\mathcal{E}_d$. Here we use the Kronecker symbol

$$\delta_{ij} = \begin{cases} 1 & \text{if } i = j \\ 0 & \text{else.} \end{cases}$$

The entries of the Hessian then are

$$\begin{aligned}
\frac{\partial^2 \mathcal{E}_d}{\partial \mathbf{u}_j \partial \mathbf{u}_i} &= 2\left(\delta_{ij} + \alpha \frac{\partial}{\partial \mathbf{u}_j}\left(\sum_{k=1}^{p} a_{ki}\varphi'\left((A\mathbf{u})_k^2\right)(A\mathbf{u})_k\right)\right) \\
&= 2\left(\delta_{ij} + \alpha \sum_{k=1}^{p} a_{ki}\frac{\partial}{\partial \mathbf{u}_j}\left(\varphi'\left((A\mathbf{u})_k^2\right)(A\mathbf{u})_k\right)\right) \\
&= 2\left(\delta_{ij} + \alpha \sum_{k=1}^{p} a_{ki}\left[\left(\frac{\partial}{\partial \mathbf{u}_j}\varphi'\left((A\mathbf{u})_k^2\right)\right)(A\mathbf{u})_k\right.\right. \\
&\qquad\qquad \left.\left. + \varphi'\left((A\mathbf{u})_k^2\right)\left(\frac{\partial}{\partial \mathbf{u}_j}(A\mathbf{u})_k\right)\right]\right) \\
&= 2\left(\delta_{ij} + \alpha \sum_{k=1}^{p} a_{ki}\left(2\varphi''\left((A\mathbf{u})_k^2\right)(A\mathbf{u})_k^2 a_{kj} + \varphi'\left((A\mathbf{u})_k^2\right)a_{kj}\right)\right) \\
&= 2\left(\delta_{ij} + \alpha \sum_{k=1}^{p} a_{ki}a_{kj}\left(2\varphi''\left((A\mathbf{u})_k^2\right)(A\mathbf{u})_k^2 + \varphi'\left((A\mathbf{u})_k^2\right)\right)\right).
\end{aligned}$$

To write the Hessian as a matrix product we define the $(p \times p)$-diagonal matrix

$$G_A(\mathbf{u}) \quad := \quad \operatorname{diag}\left(2\varphi''\left((A\mathbf{u})_k^2\right)(A\mathbf{u})_k^2 + \varphi'\left((A\mathbf{u})_k^2\right)\right)_{k=1,\dots,p}. \tag{5.5}$$

With this definition the Hessian can be written as

$$H_{\mathcal{E}_d}(\mathbf{u}) = 2\left(I + \alpha A^T G_A(\mathbf{u})A\right). \tag{5.6}$$

Since we try to minimise the function $\mathcal{E}_d$ we are interested in conditions for the Hessian $H_{\mathcal{E}_d}$ to be positive definite. Equation (5.6) leads us directly to necessary conditions for positive definiteness of $H_{\mathcal{E}_d}$: Let us assume that all entries of the diagonal matrix $G_A(\mathbf{u})$ are nonnegative, i. e. $G_A(\mathbf{u})$ is positive semidefinite. It follows immediately that $A^T G_A(\mathbf{u})A$ is positive semidefinite, too. Since $\alpha > 0$ it is clear that the Hessian is positive definite in this case. A sufficient condition for global positive definiteness of the Hessian is therefore given by

$$2\varphi''(x^2)x^2 + \varphi'(x^2) \;>\; 0 \quad \text{for all } x \in \mathbb{R}. \tag{5.7}$$

In this case the energy functional $\mathcal{E}_d$ is globally convex, and all points where the gradient vanishes are local minima.

Remark 5.3 (Linear Penaliser) In the case of a linear penaliser $\varphi(x) = cx$ with $c \geq 0$ we have $\Phi_A(\mathbf{u}) = cI$ for all $\mathbf{u} \in \mathbb{R}^n$, and the system of equations (5.4) is linear:

$$\left(I + \alpha c A^T A\right)\mathbf{u} = \mathbf{f}.$$

Since $A^T A$ is positive semidefinite and symmetric and $c \geq 0$, the whole system matrix is positive definite and symmetric. That means we have a unique solution which can be efficiently found with the Conjugate-Gradient method, for example.

For a nonlinear penaliser we have to apply nonlinear optimisation strategies. As an example we consider a gradient descent approach.

Gradient Descent Approach

We formulate a gradient descent minimisation approach for discrete energy functionals of the type (5.1). In each step of the iterative algorithm we start with the vector $\mathbf{u}^k$ and go a distance τ_k in direction p^k so that the energy gets lower. Since $-\nabla\mathcal{E}_d(x)$ points to the direction of the steepest descent of $\mathcal{E}_d$ in the point x, one can choose $p_k := -\nabla\mathcal{E}_d(\mathbf{u}^k)$. For this choice the special gradient descent method is called **steepest descent method**:

$$\begin{aligned} \mathbf{u}^0 &:= \mathbf{f} \\ \mathbf{u}^{k+1} &:= \mathbf{u}^k - \tau_k \nabla\mathcal{E}_d(\mathbf{u}^k) \quad k \in \mathbb{N}. \end{aligned}$$

In our special case the method to the function $\mathcal{E}_d$ is given by

$$\begin{aligned} \mathbf{u}^0 &:= \mathbf{f} \\ \mathbf{u}^{k+1} &:= \mathbf{u}^k - \tau_k \left(\mathbf{u}^k - \mathbf{f} + \alpha A^T \Phi_A(\mathbf{u}^k) A\mathbf{u}^k\right). \end{aligned} \tag{5.8}$$

5.2 Discretisation of the Euler-Lagrange Equation

In Chapter 2 we deduced that the linear energy functional of the type

$$\mathcal{E}(u) = \int\limits_\Omega (u-f)^2 + \alpha\left(u^{(m)}\right)^2 dx \tag{5.9}$$

for $m \in \mathbb{N} \setminus \{0\}$ and $\alpha > 0$ leads to the corresponding Euler-Lagrange equation

$$0 = u - f + (-1)^m \alpha u^{(2m)}. \tag{5.10}$$

We now discretise our interval Ω as indicated in Section 4.1 and choose a matrix A to approximate the $2m$th derivative. We then get a system of linear equations

$$0 = \mathbf{u} - \mathbf{f} + (-1)^m \alpha A\mathbf{u}$$

which can also be written as

$$\left(I + (-1)^m \alpha A\right)\mathbf{u} = \mathbf{f}. \tag{5.11}$$

The invertibility of $I + (-1)^m\alpha A$ is equivalent to the existence of a unique solution of this problem. This depends on how we choose the matrix A. From the last chapter we have obtained three different ways to choose it:

Finite Differences with Natural Boundary Conditions

With the matrix $D^F_{m,n}$ from Definition 4.14 we have seen that

$$A = (-1)^m \left(D^F_{m,n}\right)^T D^F_{m,n}$$

yields an approximation for $u^{(2m)}$ assuming natural boundary conditions. Our system of linear equations then reads as

$$\left(I + \alpha\left(D^F_{m,n}\right)^T D^F_{m,n}\right)\mathbf{u} = \mathbf{f}.$$

Since $\left(D^F_{m,n}\right)^T D^F_{m,n}$ is positive semidefinite, it is clear that our system matrix is invertible and there is a unique solution. With Proposition 4.19 we know that

$$\ker\left(D^F_{m,n}\right)^T D^F_{m,n} = \ker D^F_{m,n} = \text{ ran } V^m_n.$$

So with this discretisation linear filtering only operates on the orthogonal space of ran V^m_n. That means for all $\alpha > 0$ the orthogonal projection of our initial signal $\mathbf{f}$ onto the space of all polynomial evaluations with degree less than m is preserved.

Finite Differences with Neumann Boundary Conditions

Instead of $D^F_{m,n}$ for natural boundary conditions one can also use $D^{FN}_{m,n}$ (see Definition 4.20) to implement Neumann boundary conditions. We then use the linear system of equations

$$\left(I + \alpha \left(D^{FN}_{m,n}\right)^T D^{FN}_{m,n}\right) \mathbf{u} = \mathbf{f}.$$

The existence of a solution follows analogously to the preceeding paragraph from the positive definiteness of our system matrix. The main difference to natural boundary conditions is that in this case only the mean grey value is preserved. This follows from the statement of Lemma 4.23 which is

$$\ker D^{FN}_{m,n} = \langle (1, \dots, 1) \rangle .$$

Spectral Methods

In the case of spectral methods the linear system of equations can be solved in the Fourier domain. If we use the matrix $A = D^S_{2m,n}$ from Definition 4.26 we can write the linear system as

$$\left(I + (-1)^m \alpha D^S_{2m,n}\right) \mathbf{u} = \left(I + (-1)^m \alpha \overline{F}^T D F\right) \mathbf{u} = \mathbf{f}$$

We use the fact that F is unitary to solve this system:

$$\begin{aligned} & \overline{F}^T \left(I + (-1)^m \alpha D\right) F \mathbf{u} &=& \ \mathbf{f} \\ \iff & \mathbf{u} &=& \ \left(\overline{F}^T \left(I + (-1)^m \alpha D\right) F\right)^{-1} \mathbf{f} \\ \iff & \mathbf{u} &=& \ \overline{F}^T \left(I + (-1)^m \alpha D\right)^{-1} F \mathbf{f}. \end{aligned}$$

We also note that $I + (-1)^m \alpha D$ is a diagonal matrix which can be simply inverted by taking the reciprocal diagonal entries. We can also apply the extension and restriction matrices described in the last chapter in this framework.

5.3 Methods with Parabolic Equations

Besides the energy functional and the corresponding Euler-Lagrange equation we can also take the equations of diffusion or diffusion-reaction type presented in Section 3.1.2 to obtain image processing methods. In these equations a temporal and one or more spatial partial derivatives are involved. First we consider the semi-discrete case if only the spatial derivatives are discretised. Here we obtain some convergence results for linear filtering. Then we will take a look at fully discrete methods also suitable for nonlinear filtering. These are used for filtering examples in the next chapter.

5.3.1 The Semi-Discrete Case

As we have discussed in Section 3.1.2 the Euler-Lagrange equation (5.10) can be transferred into the parabolic partial differential equation

$$u_t = (-1)^{m+1} \frac{d^{2m}}{dx^{2m}} u \qquad (5.12)$$

with initial condition $u(t, x) = f(x)$ for all $x \in \Omega$. Spatial discretisation of (5.12) yields a system of ordinary differential equations given in matrix notation as

$$\mathbf{u}_t = (-1)^{m+1} A \mathbf{u}. \qquad (5.13)$$

Here $\mathbf{u}_t$ stands for the componentwise differentiation of $\mathbf{u}$. This system can be solved exactly in time $t > 0$ with

$$\mathbf{u}(t) = \exp\left((-1)^{m+1} tA\right)\mathbf{u}(0) = \exp\left((-1)^{m+1} tA\right)\mathbf{f} \tag{5.14}$$

as it is shown in [12], for example. We remember that the matrix exponential function for quadratic matrices $A \in \mathbb{C}^{n\times n}$ is defined via the power series

$$\exp(A) = \sum_{k=0}^{\infty} \frac{A^k}{k!}.$$

If A can be written in diagonal form

$$A = \overline{U}^T DU \qquad \text{with } D = \operatorname{diag}(d_1, \ldots, d_n)$$

with the eigenvalues $d_1, \ldots, d_n$ and a unitary matrix U, the exponential of A is

$$\exp(A) = \overline{U}^T \exp(D) U \text{ with } \exp(D) = \operatorname{diag}(e^{d_1}, \ldots, e^{d_n}).$$

The matrix exponential function also satisfies the functional equation

$$\exp(A)\exp(B) = \exp(A + B)$$

known from the real and complex exponential functions (see [12] or [7] for details). Thus we can immediately deduce a scale-space property for semi-discrete filtering independent of the discretisation matrix A.

Lemma 5.4 (Semigroup Property) *The set of semi-discrete filtering operators T_t defined by*

$$T_t\mathbf{f} = \mathbf{u}(f) = \exp\left((-1)^m tA\right)\mathbf{f}$$

satisfies the semigroup property

$$\begin{aligned} T_0\mathbf{f} &= \mathbf{f} \\ T_{(s+t)}\mathbf{f} &= T_s\left(T_t\mathbf{f}\right) \end{aligned}$$

for all $\mathbf{f} \in \mathbb{R}^n$ and $s, t \geq 0$.

We now consider three different cases for the approximation of derivatives:

Finite Difference Approximations

For finite difference approximations of the derivative we will get a statement for $t \longrightarrow \infty$.

Natural Boundary Conditions As already described in the last section we can choose $A = \left(D^F_{m,n}\right)^T D^F_{m,n}$ as a discrete equivalent to $(-1)^m \frac{d^{2m}}{dx^{2m}}$ with natural boundary conditions. The system of ordinary differential equations then is given by

$$\begin{aligned} \mathbf{u}_t &= -\left(D^F_{m,n}\right)^T D^F_{m,n}\mathbf{u} \\ \mathbf{u}(0) &= \mathbf{f}. \end{aligned}$$

It has the solution

$$\mathbf{u}(t) = \exp\left(-t\left(D^F_{m,n}\right)^T D^F_{m,n}\right)\mathbf{f}.$$

Since $\left(D^F_{m,n}\right)^T D^F_{m,n}$ is real and symmetric, it is positive semidefinite and can be diagonalised:

$$\left(D^F_{m,n}\right)^T D^F_{m,n} = U^T D^F U$$

where U is a unitary matrix and

$$D^F = \operatorname{diag}\left(\lambda_1, \ldots, \lambda_n\right)$$

is a diagonal matrix containing the eigenvalues. The exact solution then is

$$\mathbf{u}(t) = U^T \operatorname{diag}\left(\exp(-t\lambda_1), \ldots, \exp(-t\lambda_n)\right) U\mathbf{f}.$$

From Lemma 4.19 we know that $\ker\left(D^F_{m,n}\right) = \operatorname{ran} V^m_n$. In particular the kernel has dimension m, and together with the positive semi-definiteness we have

$$\lambda_1 = \ldots = \lambda_m < \lambda_{m+1} \leq \lambda_n.$$

Since $\lambda_j > 0$ for $j \in \{m+1, \ldots, n\}$, it follows that $\lim\limits_{t \longrightarrow \infty} \exp(-t\lambda_j) = 0$ and thus

$$\lim_{t \longrightarrow \infty} \mathbf{u}(t) = U^T \operatorname{diag}\left(1, \ldots, 1, 0, \ldots, 0\right) U\mathbf{f}.$$

The eigenvectors belonging to the nonzero eigenvalues span the kernel of $D^F_{m,n}$, and thus the limit for $t \longrightarrow \infty$ is the projection onto the kernel. From Section 4.2.2 we know that this projection yields the least-square polynomial data fitting of degree less than m.

Neumann Boundary Conditions In the case of Neumann boundary conditions we can use the matrix

$$A = \left(D^{FN}_{m,n}\right)^T D^{FN}_{m,n}$$

as discrete version of $(-1)^m \frac{d^{2m}}{dx^{2m}}$. The above argumentation works in exactly the same way. The only difference is that the kernel is only one-dimensional. It follows from Lemma 4.23 that

$$\ker\left(D^{FN}_{m,n}\right)^T D^{FN}_{m,n} = \left\langle (1, \ldots, 1)^T \right\rangle.$$

Thus the limit for $t \longrightarrow \infty$ of our solution is the mean value of our initial data:

$$\lim_{t \longrightarrow \infty} \mathbf{u}(t) = \frac{1}{n} \sum_{k=0}^{n-1} \mathbf{f}_k.$$

This can be considered as the least-square polynomial data fitting of degree 0.

Spectral Methods

For spectral methods we have already a matrix in diagonal form so that we can compute the exponential of this matrix exactly. We can use $D^S_{m,n}$ from Definition 4.26 as a discretisation for $\frac{d^m}{dx^m}$. So we get the solution

$$\begin{aligned} \mathbf{u}(t) &= \exp\left((-1)^{m+1} t D^S_{2m,n}\right) \mathbf{f} \\ &= \overline{F}^T \exp\left((-1)^{m+1} t \operatorname{diag}\left(\lambda_1, \ldots, \lambda_n\right)\right) F\mathbf{f}. \end{aligned}$$

From section 4.3 we know that

$$\begin{aligned}\lambda_k &= \left(\frac{2\pi i\left(k-1-\frac{n}{2}\right)}{n}\right)^{2m} \\ &= (-1)^m \left(\frac{2\pi\left(k-1-\frac{n}{2}\right)}{n}\right)^{2m} \qquad \text{for } k \in \{1,\ldots,n\}.\end{aligned}$$

The exact solution is

$$\mathbf{u}(t) = \overline{F}^T \operatorname{diag}\left(\mu_1,\ldots,\mu_n\right) F\mathbf{f}$$

with diagonal entries

$$\mu_k = -t\exp\left(\left(\frac{2\pi\left(k-1-\frac{n}{2}\right)}{n}\right)^{2m}\right) \qquad \text{for } k \in \{1,\ldots,n\}.$$

We note that with spectral methods it is possible to calculate the exact solution for $t > 0$ in $O(n\log n)$ with the Fast Fourier transform. Since the eigenvalue 0 only appears one time and the corresponding eigenvector is the constant vector, we have preservation of the average grey value. For $t \longrightarrow \infty$ we obtain convergence towards the average grey value. To avoid artifacts due to boundary conditions we may use the extension and restriction (see Definition 4.28), and the filtering process reads as

$$\mathbf{u}(t) = R\overline{F}^T \operatorname{diag}(\mu_1,\ldots,\mu_{2n}) FE\mathbf{f}$$

with analogously defined μ_k with respect to double signal size. With the notes of Section 4.3.3 these results immediately carry over to higher dimensions and the Laplacian.

5.3.2 The Discrete Case

We note that the semi-discrete results discussed in the last section are only suitable for linear filtering. The next step is to discretise also the temporal derivative $\mathbf{u}_t$. We discuss an implicit or an explicit discretisation taking the forward or the backward difference instead of $\mathbf{u}_t$. Linear combinations of both methods which are called θ-schemes are also commonly used; they will not be considered here.

We start with the nonlinear diffusion equation (3.5):

$$u_t = (-1)^{m+1}\frac{d^m}{dx^m}\left(\varphi'\left(\left(u^{(m)}\right)^2\right)u^{(m)}\right).$$

We discretise the equation in the space variable using an approximation matrix A for the mth derivative and the matrix Φ_A from definition 5.2. (We note that the dimensions of the quadratic matrix Φ_A is equal to the row number of A which in general must not be quadratic. The finite difference approximation with natural boundary conditions $D^F_{m,n}$ is an example of a non-quadratic matrix A.) We use the transposed matrix A^T also as approximation of the mth derivative since it appeared in Section 5.1. This yields the system of ordinary differential equations

$$\mathbf{u}_t = (-1)^{m+1}A^T\Phi_A(\mathbf{u})A\mathbf{u}. \tag{5.15}$$

Later on we insert matrices from the last chapter for A. The next step will be the discretisation in time. Here we have several possibilities which will only be sketched here. Most principles from first order filtering will also be useful in higher order filters.

The two border cases for discretisation in time are fully implicit and fully explicit schemes. Let us choose a temporal step size $\tau > 0$. With $\mathbf{u}^k$ we denote the vector $\mathbf{u}(k\tau)$ for $k \in \mathbb{N}$. Our process always starts with the initial image $\mathbf{u}(0) = \mathbf{f}$.

Implicit Discretisation

A fully implicit scheme evaluates the right-hand side at the time $(k+1)\tau$ yielding the system of nonlinear equations

$$\begin{aligned} \frac{\mathbf{u}^{k+1}-\mathbf{u}^k}{\tau} &= (-1)^{m+1} A^T \Phi_A\left(\mathbf{u}^{k+1}\right) A \mathbf{u}^{k+1} \\ \Longleftrightarrow \qquad \mathbf{u}^k &= \mathbf{u}^{k+1} - \tau\left((-1)^{m+1} A^T \Phi_A\left(\mathbf{u}^{k+1}\right) A\right) \mathbf{u}^{k+1} \\ \Longleftrightarrow \qquad \mathbf{u}^k &= \left(I + \tau(-1)^m A^T \Phi_A\left(\mathbf{u}^{k+1}\right) A\right) \mathbf{u}^{k+1}. \end{aligned} \tag{5.16}$$

To obtain u^{k+1} and to perform one step of this scheme we have to solve a nonlinear system of equations.

Remark 5.5 (Implicit Discretisation and Euler-Lagrange equation) An implicit scheme for linear filtering reads as

$$\left(I + \tau(-1)^m A^T A\right) \mathbf{u}^{k+1} = \mathbf{u}^k.$$

If we start with $k = 0$ and perform only one step with step size $\tau = \alpha$ we see the similarity to equation (5.11) which we obtained from the discretisation of the elliptic Euler-Lagrange equation. (We must take care on the different meanings of the matrices A in both equations.) This works analogously for the nonlinear case.

For our implementations we use explicit discretisation. This will be further explained in the next section.

Explicit Discretisation

In contrast to the implicit discretisation given above, explicit methods evaluate the right-hand side at the time $k\tau$. We get the iteration scheme

$$\begin{aligned} \frac{\mathbf{u}^{k+1}-\mathbf{u}^k}{\tau} &= (-1)^{m+1} A^T \Phi_A\left(\mathbf{u}^{k}\right) A \mathbf{u}^{k} \\ \Longleftrightarrow \qquad \mathbf{u}^{k+1} &= \left(I + \tau(-1)^{m+1} A^T \Phi_A\left(\mathbf{u}^{k}\right) A\right) \mathbf{u}^{k}. \end{aligned} \tag{5.17}$$

This explicit method has the advantage that the entries of $\mathbf{u}^{k+1}$ can be computed from the known vector $\mathbf{u}^k$ as evaluation of the nonlinearity and matrix-vector multiplication. In contrast to methods with an implicit component no linear or nonlinear system of equations has to be solved. Thus these methods are relatively easy to implement.

Remark 5.6 (Gradient Descent Methods) The explicit scheme (5.17) may remind us at the gradient descent approach (5.8). One can understand the scheme (5.17) as a gradient descent method with fixed step size $\tau_k = \tau$ for a discrete energy functional

$$\mathcal{E}_d\left(\mathbf{u}\right) = \alpha \sum_{k=1}^{p} \varphi\left((Au)_k^2\right).$$

Vice versa the scheme (5.8) can be considered as an explicit discretisation for the corresponding diffusion-reaction equation mentioned in Section 3.1.2.

All matrices we have seen for higher order derivative approximation have nonnegative off-diagonal entries, and therefore a maximum-minimum property will not hold in general for such methods. Instead of this useful property we claim that our method should be stable with respect to the Euclidean norm, that means

$$\|\mathbf{u}^k\|_2 \le \|\mathbf{f}\|_2 \quad \text{for all } k \in \mathbb{N}.$$

Since the spectral norm is a compatible matrix norm to the Euclidean vector norm we get an upper bound for $\|\mathbf{u}^{k+1}\|_2$ from the scheme (5.17):

$$\|\mathbf{u}^{k+1}\|_2 \leq \left\|I + \tau(-1)^{m+1}A^T\Phi_A(\mathbf{u}^k)A\right\|_2 \|\mathbf{u^k}\|_2.$$

Since Φ_A is a diagonal matrix, $I + \tau(-1)^{m+1}A^T\Phi_A\left(\mathbf{u}^k\right)A$ is symmetric for all $k \in \mathbb{N}$. It follows that all eigenvalues are real. Thus the method is 2-stable under the condition that

$$\sigma\left(I + \tau(-1)^m A^T\Phi_A\left(\mathbf{u}^k\right)A\right) \subseteq [-1,1]. \tag{5.18}$$

Remark 5.7 (Nonlinearity Matrix) First we remember that $\Phi_A(\mathbf{u})$ (cf. Definition 5.2) is defined as

$$\Phi_A(\mathbf{u}) = \operatorname{diag}\left(\varphi'\left((A\mathbf{u})_k^2\right)\right)_{k=1,\ldots,n-m}.$$

The eigenvalues of Φ_A therefore are in the range of the first derivative φ' of the penaliser. For linear filtering, Perona-Malik and Charbonnier filtering we have

$$0 \leq \varphi'(x) \leq 1 \quad \text{for all } x \in \mathbb{R}.$$

The total variation approximation satisfies

$$0 \leq \varphi'(x) \leq \frac{1}{\varepsilon} \quad \text{for all } x \in \mathbb{R}.$$

Since these values are all nonnegative it follows immediately that $\Phi_A\left(\mathbf{u}^k\right)$ is always positive semi-definite. We also obtain an upper bound for the spectral norm of $\Phi_A\left(\mathbf{u}^k\right)$ independent of the derivative approximation A, the current vector $\mathbf{u}^k$ and the size of Φ_A. That means we have $\|\Phi_A(\mathbf{u})\|_2 \leq 1$ for linear filtering, Perona-Malik and Charbonnier penalisers and $\|\Phi_A(\mathbf{u})\|_2 \leq \frac{1}{\varepsilon}$ for total variation approximation.

We will now have a closer look at the stability condition (5.18) depending on the derivative approximation we choose.

Finite Differences

We consider a one-dimensional method with natural boundary conditions and use the finite difference matrix $D^F_{m,n}$ from Definition 4.14. In Remark 4.18 we have seen that $\left(D^F_{m,n}\right)^T$ then approximates $(-1)^m$ times the mth derivative. Therefore we have to modify our scheme from above to

$$\mathbf{u}^{k+1} = \left(I - \tau\left(D^F_{m,n}\right)^T \Phi_{D^F_{m,n}}\left(\mathbf{u}^k\right)D^F_{m,n}\right)\mathbf{u}^k$$

in this case. Here we have $D^F_{m,n} \in \mathbb{R}^{n-m\times n}$ and therefore $\Phi_{D^F_{m,n}}\left(\mathbf{u}^k\right) \in \mathbb{R}^{n-m\times n-m}$ for all $k \in \mathbb{N}$. Concerning Remark 5.7 we assume that there is an upper bound $c \in \mathbb{R}$ for the derivative of the penaliser such that

$$0 \leq \varphi'(x) \leq c \quad \text{for all } x \in \mathbb{R}$$

holds. We know that $\Phi_{D^F_{m,n}}\left(\mathbf{u}^k\right)$ is then positive semidefinite, and it follows that the matrix $\left(D^F_{m,n}\right)^T \Phi_{D^F_{m,n}}\left(\mathbf{u}^k\right)D^F_{m,n}$ is also positive semidefinite. For the spectral norm we have

$$\left\|\Phi_{D^F_{m,n}}\left(\mathbf{u}^k\right)\right\|_2 \leq c.$$

Together with Lemma 4.16 we conclude that

$$\begin{aligned}\sigma\left(\left(D^F_{m,n}\right)^T \Phi_{D^F_{m,n}}\left(\mathbf{u}^k\right) D^F_{m,n}\right) &\subseteq \left[0, \left\|D^F_{m,n}\right\|_2^2 \left\|\Phi_{D^F_{m,n}}(\mathbf{u}^k)\right\|_2\right] \\ &\subseteq \left[0, \left(\frac{2}{h}\right)^{2m} c\right].\end{aligned}$$

We now are able to consider the spectrum of our system matrix:

$$\sigma\left(I - \tau\left(D^F_{m,n}\right)^T \Phi_{D^F_{m,n}}\left(\mathbf{u}^k\right) D^F_{m,n}\right) \subseteq \left[1 - \tau\left(\frac{2}{h}\right)^{2m} c, 1\right].$$

Stability in the Euclidean norm is satisfied if

$$1 - \tau\left(\frac{2}{h}\right)^{2m} c \geq -1$$

which can be formulated as a restriction on the time step size

$$\tau \leq \frac{h^{2m}}{2^{2m-1}c}.$$

We formulate this result as

Proposition 5.8 (Stability Limits in 1D) *Assume that the first derivative of the penaliser function φ is bounded such that*

$$0 \leq \varphi'(x) \leq c \quad \text{for all } x \in \mathbb{R}.$$

The one-dimensional explicit scheme for mth order nonlinear filtering with natural boundary conditions

$$\mathbf{u}^{k+1} = \left(I - \tau\left(D^F_{m,n}\right)^T \Phi_{D^F_{m,n}}\left(\mathbf{u}^k\right) D^F_{m,n}\right)\mathbf{u}^k \tag{5.19}$$

is stable with respect to the Euclidean norm for time step size τ satisfying

$$\tau \leq \frac{h^{2m}}{2^{2m-1}c}.$$

The same stability limit holds for the one-dimensional explicit scheme with Neumann boundary conditions

$$\mathbf{u}^{k+1} = \left(I - \tau\left(D^{FN}_{m,n}\right)^T \Phi_{D^{FN}_{m,n}}\left(\mathbf{u}^k\right) D^{FN}_{m,n}\right)\mathbf{u}^k. \tag{5.20}$$

Proof: The scheme for natural boundary conditions has been considered above. The only facts from the matrix $D^F_{m,n}$ that are important for these considerations are that it is real-valued, the transposed matrix approximates $(-1)^m$ times the mth derivative, and the spectral norm bound from Lemma 4.16. With Remark 4.21 and Lemma 4.22 we can use the same reasoning also for Neumann boundary conditions with the matrix $D^{FN}_{m,n}$. □

For two-dimensional schemes derived from equation (2.36) we use the matrices L and $\tilde{L}$, respectively, which were described in Section 4.2.4. We get the explicit scheme

$$\mathbf{u}^{k+1} = \left(I - \alpha\, L\Phi_L\left(\mathbf{u}^k\right) L\right)\mathbf{u}^k, \tag{5.21}$$

and the same formula with $\tilde{L}$ instead of L. Our reasoning of the one-dimensional case immediately carries over to two dimensions, and we get

Proposition 5.9 (Stability Limits in 2D) *Again assume that $c > 0$ is an upper bound for the first derivative of the penaliser φ such that*

$$0 \leq \varphi'(x) \leq c \quad \textit{for all } x \in \mathbb{R}.$$

The explicit scheme for second order nonlinear filtering in 2D given in formula (5.21) is stable with respect to the Euclidean norm for time step sizes τ satisfying

$$\tau \leq \frac{h^2}{32c}.$$

For the alternative approximation of the Laplacian with matrix $\tilde{L}$ instead of L in formula (5.21) we get the stability condition

$$\tau \leq \frac{9h^2}{200c}.$$

We should note that the discrete Laplacian with better rotation invariance even allows us to take larger time step sizes τ.

Spectral Methods

For spectral methods we first consider the one-dimensional case. We use the matrix $D^S_{m,n}$ (see Definition 4.26) to approximate the mth derivative. This yields the explicit scheme

$$\mathbf{u}^{k+1} = \Big(I + \tau(-1)^{m+1} \left(D^S_{m,n}\right) \Phi_{D^S_{m,n}} \left(\mathbf{u}^k\right) D^S_{m,n}\Big) \mathbf{u}^k.$$

We rewrite this scheme with $(-1)^m = i^{2m}$ and Definition 4.26 to

$$\begin{aligned} \mathbf{u}^{k+1} &= \Big(I - \tau \left(i^m D^S_{m,n}\right) \Phi_{D^S_{m,n}} \left(\mathbf{u}^k\right) \left(i^m D^S_{m,n}\right)\Big) \mathbf{u}^k \\ &= \Big(I - \tau \overline{F}^T (iD)^m F \Phi_{D^S_{m,n}} \left(\mathbf{u}^k\right) \overline{F}^T (iD)^m F\Big) \mathbf{u}^k. \end{aligned} \tag{5.22}$$

From Section 4.3 we know that $\overline{iD}^T = iD$ is Hermitean and thus

$$\overline{\overline{F}^T (iD) F}^T = \overline{F}^T (iD) F.$$

With this property and since $\Phi_{D^S_{m,n}}$ is a real positive semidefinite diagonal matrix, we know that the scheme matrix

$$I + \tau(-1)^m \left(D^S_{m,n}\right)^T \Phi_{D^S_{m,n}} \left(\mathbf{u}^k\right) D^S_{m,n} = I - \tau \overline{F}^T (iD)^m F \Phi_{D^S_{m,n}} \left(\mathbf{u}^k\right) \overline{F}^T (iD)^m F$$

is positive semidefinite, too. We conclude that the eigenvalues of this matrix are in the interval

$$\left[0, \tau \left\|D^S_{m,n}\right\|_2^2 \left\|\Phi_{D^S_{m,n}}\right\|_2\right]$$

and with Lemma 4.27 we have proven

Proposition 5.10 (Stability Limits in 1D) *With the assumptions from Proposition 5.8 at the penaliser φ, the explicit scheme for mth order one-dimensional nonlinear diffusion with spectral methods*

$$\mathbf{u}^{k+1} = \Big(I + \tau(-1)^{m+1} \left(D^S_{m,n}\right) \Phi_{D^S_{m,n}} \left(\mathbf{u}^k\right) D^S_{m,n}\Big) \mathbf{u}^k$$

is stable with respect to the Euclidean norm for time step sizes

$$\tau \leq \frac{2}{\pi^{2m} c}.$$

For two-dimensional image processing we use the matrix D^S defined in Section 4.3.3 to approximate the Laplace operator. Analogously to the above considerations we get

Proposition 5.11 (Stability limits in 2D) *Under the assumptions from Proposition 5.8 at the penaliser φ, the explicit scheme for nonlinear filtering derived from equation (2.36)*

$$\mathbf{u}^{k+1} = \left(I - \tau D^S \Phi_{D^S}\left(\mathbf{u}^k\right) D^S\right) \mathbf{u}^k$$

is stable with respect to the Euclidean norm for time step sizes

$$\tau \leq \frac{1}{2\pi^4 c}.$$

We may also add the extension and restriction operators from Definition 4.28 in 1D or their 2D equivalents. Lemma 4.29 and the considerations in Section 4.3.3 assure that the same stability limits stay true even in this case.

Now we consider the effect of this stability limits in practice. We assume that $c = 1$ (for example for a Charbonnier or Perona-Malik penaliser) and use the spatial step size $h = 1$. The following table then shows the maximal step sizes for different derivative orders and discretisations:

Dimension	Order	Discretisation	$\tau_{\max}$
1	1	FD	0.5
		Spectral	0.2026424
	2	FD	0.125
		Spectral	0.0205320
	3	FD	0.03125
		Spectral	0.0020803
	4	FD	0.0078125
		Spectral	0.0002108
2	2 (Laplacian)	FD matrix (L)	0.03125
		FD matrix ($\tilde{L}$)	0.045
		Spectral	0.0051330

We see that finite difference approximations usually allow larger time step sizes. Additionally the complexity of one explicit step is at least $O(n \log n)$ for spectral methods since the Fourier transform has to be performed. For one step with finite differences one has complexity $O(n)$. This shows that nonlinear finite difference explicit methods are faster than spectral methods.

In the next chapter we present some filtering results obtained with the filters described here.

Chapter 6

Experimental Results

After investigating theoretical results for continuous and discrete filtering this chapter deals with numerical tests of higher order filters in practice. In the first section we will present some filtering results which were obtained with finite difference methods. We always use a spatial step size $h = 1.0$ in this case. The second section compares finite differences and spectral methods on the basis of concrete examples. Finally we present some results for combined filters of first and second order in Section 6.3.

As a measure for the quality of a filtering result we use the Signal-Noise-Ratio and the Peak-Signal-Noise-Ratio.

Definition 6.1 (Signal-Noise-Ratio) *Let* $\mathbf{g} \in \mathbb{R}^n$ *be the original signal and* $\mathbf{f} \in \mathbb{R}^n$ *a noisy version of* $\mathbf{g}$*. Then the signal-noise-ratio is defined as*

$$\mathrm{SNR}(\mathbf{f}, \mathbf{g}) := 10 \log_{10} \left(\frac{\sum\limits_{i=1}^{n} \mathbf{g}_i^2}{\sum\limits_{i=1}^{n} (\mathbf{g}_i - \mathbf{f}_i)^2} \right).$$

The peak-signal-noise-ratio is defined as

$$\mathrm{PSNR}(\mathbf{f}, \mathbf{g}) := 10 \log_{10} \left(\frac{n \max\limits_{i=1,\dots,n} \mathbf{g}_i^2}{\sum\limits_{i=1}^{n} (\mathbf{g}_i - \mathbf{f}_i)^2} \right).$$

For images we define SNR and PSNR analogously. The sums and the maximum are also taken over all pixels in this case. We should note that these measures are not always ideal: For 2D images they do not specifically reward sharp edges that are optically more pleasant then blurred ones.

6.1 Filtering Results

We start with one-dimensional filtering examples and first compare different penalisers. Then we will present more detailed test results for higher order total variation approximations. Filtering examples in 2D conclude the section. For all examples shown in this section we use explicit discretisations of the diffusion equations corresponding to the Euler-Lagrange equations. These methods are described in Section 5.3.2.

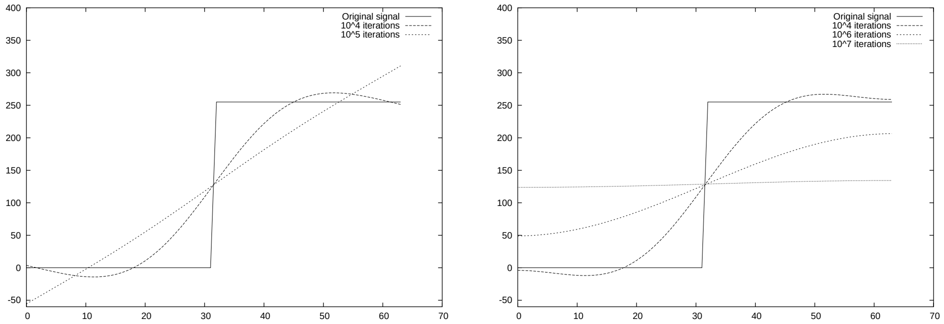

Figure 6.1: Second order linear filtering in 1D. Left: Natural boundary conditions, Right: Neumann boundary conditions.

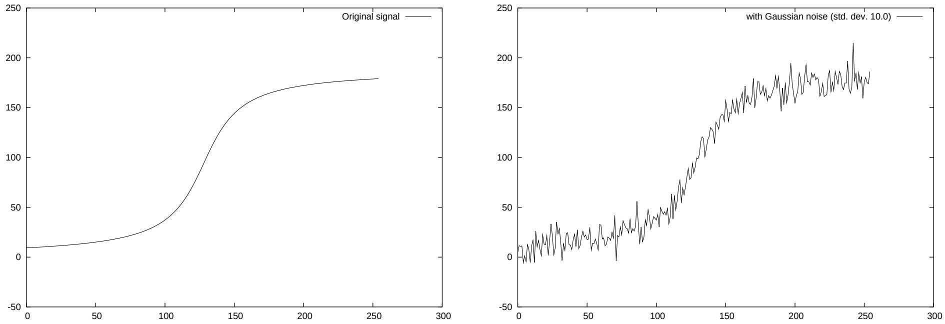

Figure 6.2: Original arc tangent signal and noisy version.

6.1.1 Filtering in 1D

First we perform linear filtering in 1D to show the behaviour of different boundary conditions. We use finite difference discretisation with Neumann and natural boundary conditions. Figure 6.1 visualises some results of Section 5.3. For natural boundary conditions, the least square polynomial data fitting is preserved. (In this case we use second order filtering and get a linear least square polynomial fitting.) For Neumann boundary conditions, only the mean value is preserved.

We then compare different penalisers in 1D. For this purpose we try to achieve the highest SNR/PSNR value that is reached for each filter type. For linear filtering the only parameter is the evolution time. Perona-Malik and Charbonnier filtering also depend on a second parameter λ. In this case we perform optimisation over both parameters. For the total variation approximation we fix our value for ε and only take the maximum over the iteration number.

Different Penalisers in 1D

As test signal for different nonlinear penalisers in 1D we use an arc tangent signal. The starting point for our denoising tests is a noisy version of this signal with additive Gaussian noise of standard deviation 10.0. This noisy version has SNR 22.02 and PSNR 25.72. Both signals are shown in Figure 6.2.

Figure 6.3 shows the filtering results for first order filtering and second order filtering with natural boundary conditions. The examples with Neumann boundary conditions can be found in Figure 6.4.

Linear Filtering For linear filtering we have searched the iteration number for which the SNR and PSNR values got maximal. Corresponding iteration numbers are listed behind the SNR and PSNR values. We see from Table 6.1 that second order linear filtering yields better results than first order filtering.

Perona-Malik For Perona-Malik filtering we have searched both the optimal iteration number and the optimal parameter λ to obtain high SNR values. We have considered values between 0.1 and 10.0 in steps of 0.1 and taken the maximal SNR value for each of this λ for this purpose to obtain the values in Table 6.2.

Charbonnier For Charbonnier filtering also the maximisation considers both the iteration number and the parameter λ. Here smaller values for λ are needed, we have used 0.01 up to 1.0 in steps of 0.01. The results can be found in Table 6.3.

Total Variation Approximation In this example we use the parameter $\varepsilon = 10^{-2}$ in our approximation of the total variation penaliser. So the iteration number is the only parameter we have to consider for the maximal SNR/PSNR in Table 6.4. To get an impression about the typical behaviour of the SNR and PSNR values during the filtering process some examples can be found in Figure 6.5. These curves come from filtering the arc tangent signal with total variation approximation, $\varepsilon = 0.01$. We see that the SNR and PSNR are increasing until they reach a maximum in both cases. This fits to the imagination that the filters should first remove and then smooth the information contained in the signal.

We summarise a few observations from these results: At first we see that the parameters for corresponding maximal SNR and PSNR pairs are relatively close together. For Perona-Malik and Charbonnier penalisers we reach the maximal SNR always with the same λ as the maximal PSNR. The differences in the iteration numbers are also rather small. Thus in the next tests we will only give one parameter set where both SNR and PSNR are approximately maximal.

6.1.2 Special Case: TV Approximation

From Figure 6.3 one could get the impression that total variation approximation filtering yields piecewise polynomial results. We would like to further investigate especially this property and therefore choose a test signal that consists of polynomials with degree from zero to three. Figure 6.6 shows the original signal and a noisy version with SNR 6.94 and PSNR 10.72.

Numerical tests were done with both values $\varepsilon = 10^{-2}$ and $\varepsilon = 10^{-4}$. The maximal SNR and PSNR values for the approximation with $\varepsilon = 10^{-2}$ are listed in Table 6.5 The same tests were done with $\varepsilon = 10^{-4}$. The corresponding results and parameters can be found in Table 6.6. We should add that the tests with fourth order filtering with natural boundary conditions were stopped after 2.7 ·

Order	Bound. Cond.	τ	Result		Iterations
1	Neumann	0.5	SNR:	32.06	70
			PSNR:	35.75	70
2	Neumann	0.125	SNR:	35.92	28131
			PSNR:	39.61	28140
2	Natural	0.125	SNR:	35.45	21690
			PSNR:	39.14	21754

Table 6.1: SNR/PSNR maximisation (linear penaliser) of the arc tangent signal.

Order	Bound. Cond.	τ	Result		Iterations	λ
1	Neumann	0.1	SNR:	34.62	344	4.6
			PSNR:	38.31	343	4.6
2	Neumann	0.1	SNR:	36.12	97948	0.2
			PSNR:	39.81	97949	0.2
2	Natural	0.1	SNR:	35.45	27409	9.9
			PSNR:	39.14	27412	9.9

Table 6.2: SNR/PSNR maximisation (Perona-Malik penaliser) of the arc tangent signal.

Order	Bound. Cond.	τ	Result		Iterations	λ
1	Neumann	0.1	SNR:	35.07	782	0.77
			PSNR:	38.76	782	0.77
2	Neumann	0.1	SNR:	36.29	108779	0.03
			PSNR:	39.97	109067	0.03
2	Natural	0.1	SNR:	35.71	137336	0.02
			PSNR:	39.40	137333	0.02

Table 6.3: SNR/PSNR maximisation (Charbonnier penaliser) of the arc tangent signal.

Order	Bound. Cond.	τ	Result		Iterations
1	Neumann	10^{-4}	SNR:	33.16	595902
			PSNR:	36.85	595749
2	Neumann	10^{-4}	SNR:	34.91	4000789
			PSNR:	38.60	4000779
2	Natural	10^{-4}	SNR:	34.37	4564197
			PSNR:	38.06	4564162

Table 6.4: SNR/PSNR maximisation (Total variation approximation penaliser) of the arc tangent signal.

Order	Bound. Cond.	τ	SNR	PSNR	Iterations
1	Neumann	$5 \cdot 10^{-3}$	16.00	19.78	$4.04 \cdot 10^{3}$
2	Neumann	$1.25 \cdot 10^{-3}$	18.30	22.08	$5.55 \cdot 10^{4}$
2	Natural	$1.25 \cdot 10^{-3}$	18.41	22.18	$4.62 \cdot 10^{4}$
3	Neumann	$3.125 \cdot 10^{-4}$	17.91	21.69	$5.44 \cdot 10^{5}$
3	Natural	$3.125 \cdot 10^{-4}$	17.97	21.74	$8.80 \cdot 10^{5}$
4	Neumann	$7.812 \cdot 10^{-5}$	17.91	21.68	$1.53 \cdot 10^{7}$
4	Natural	$7.812 \cdot 10^{-5}$	17.22	20.99	$1.86 \cdot 10^{7}$

Table 6.5: SNR/PSNR values for total variation approximation with $\varepsilon = 10^{-2}$ and the piecewise polynomial signal.

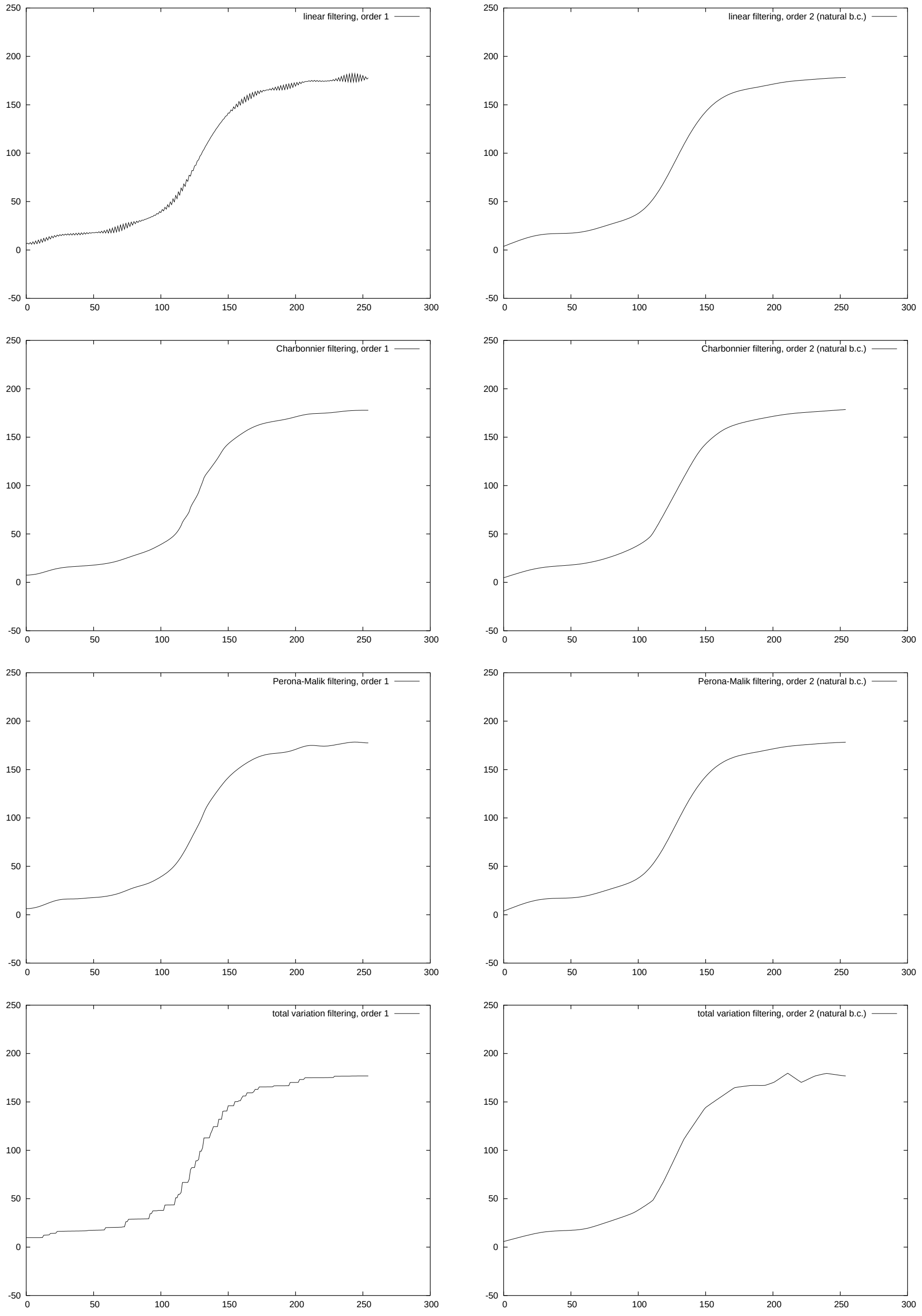

Figure 6.3: Results of SNR/PSNR maximisation in 1D. Left column: First order filtering, Right column: Second order filtering with natural boundary conditions. Top row: Linear filtering, Second row: Charbonnier penaliser, Third row: Perona-Malik penaliser, Bottom row: Total variation approximation with $\varepsilon = 0.01$.

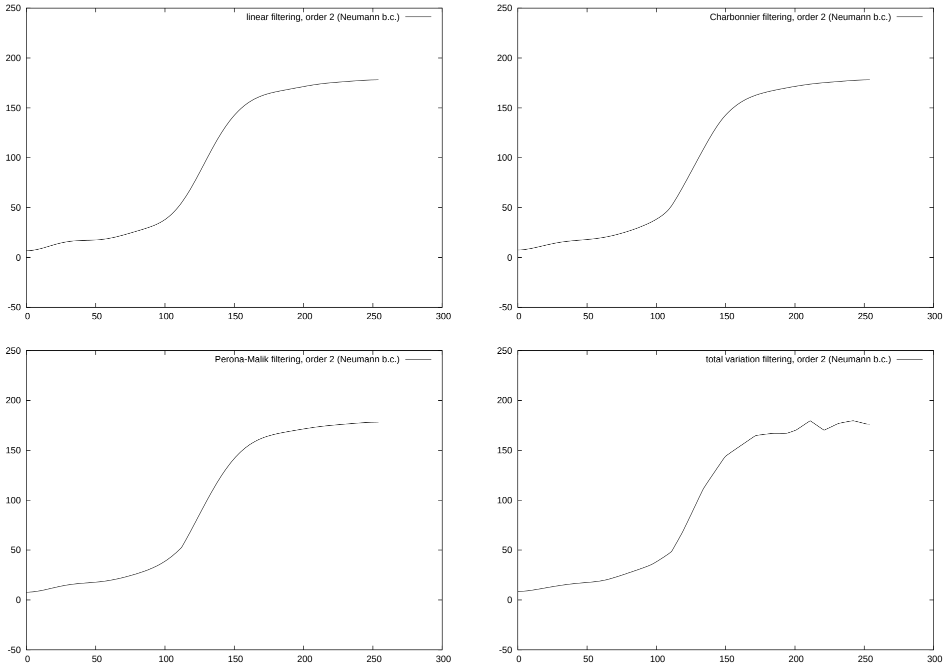

Figure 6.4: Results of SNR/PSNR maximisation in 1D. Second order filtering with Neumann boundary conditions and different penalisers. Top left: Linear penaliser, Top right: Charbonnier, Bottom left: Perona-Malik, Bottom right: TV approximation with $\varepsilon = 0.01$.

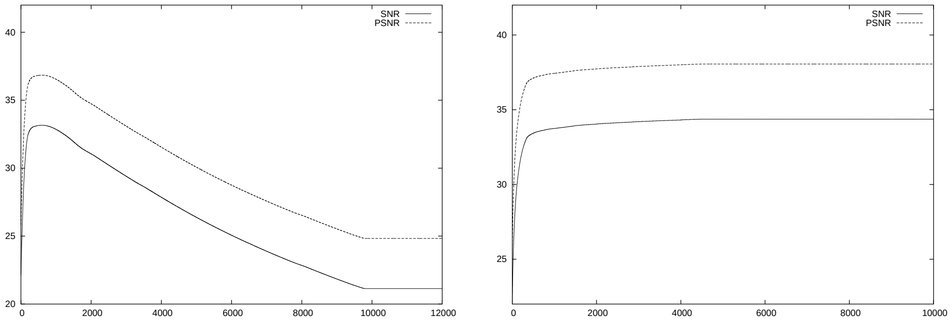

Figure 6.5: SNR and PSNR values during total variation approximation filtering. Left: First order filtering, Right: Second order filtering. The resolution of the x-axis is 1000 iterations.

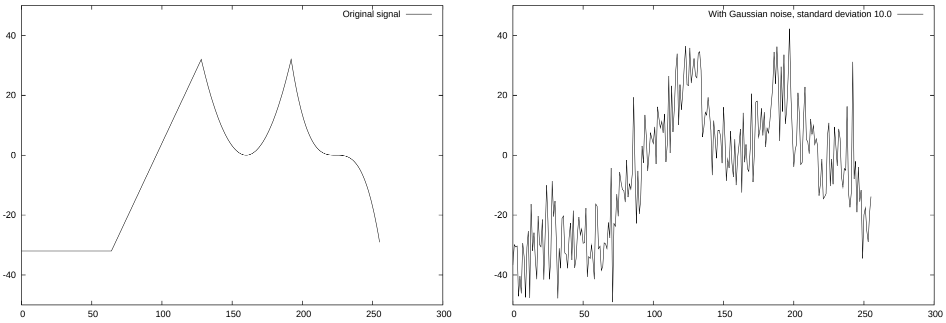

Figure 6.6: Piecewise polynomial test signal. Left: Original signal. Right: Signal with additive Gaussian noise, standard deviation 10.0.

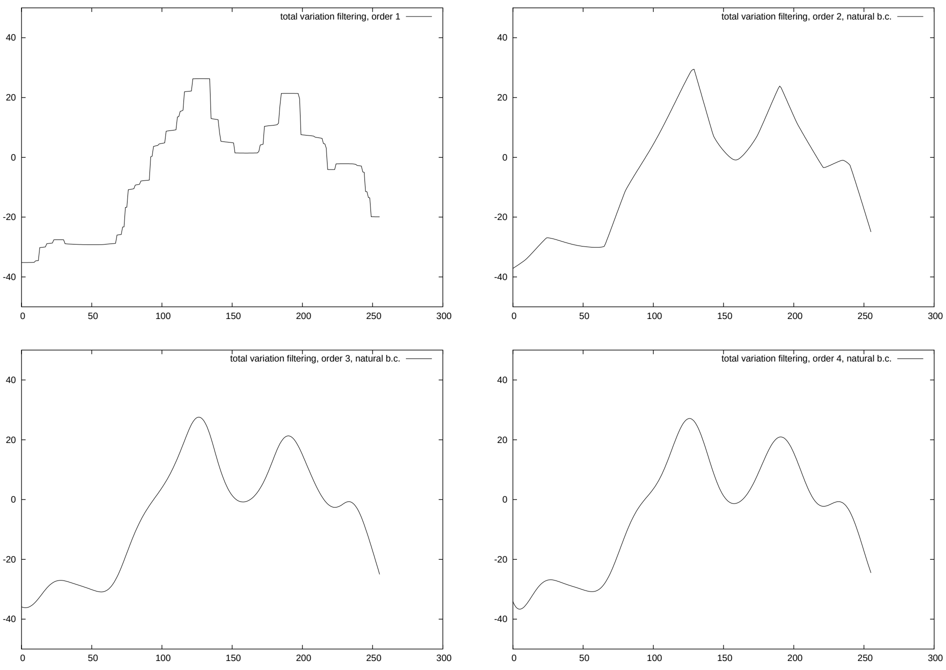

Figure 6.7: Results of SNR/PSNR maximisation with total variation approximations, $\varepsilon = 10^{-2}$, and natural boundary conditions. Top left: Order 1, Top right: Order 2, Bottom left: Order 3, Bottom right: Order 4.

Order	Bound. Cond.	τ	SNR	PSNR	Iterations
1	Neumann	$5 \cdot 10^{-5}$	15.92	19.70	$3.93 \cdot 10^{5}$
2	Neumann	$1.25 \cdot 10^{-5}$	17.94	21.72	$8.30 \cdot 10^{6}$
2	Natural	$1.25 \cdot 10^{-5}$	17.98	21.76	$1.32 \cdot 10^{7}$
3	Neumann	$3.125 \cdot 10^{-6}$	17.97	21.75	$6.59 \cdot 10^{7}$
3	Natural	$3.125 \cdot 10^{-6}$	18.11	21.89	$1.08 \cdot 10^{8}$
4	Neumann	$7.8125 \cdot 10^{-7}$	17.79	21.57	$7.62 \cdot 10^{8}$
4	Natural	$7.8125 \cdot 10^{-7}$	> 17.56	> 21.33	$> 2.7 \cdot 10^{9}$

Table 6.6: SNR/PSNR values for total variation approximation with $\varepsilon = 10^{-4}$ and the piecewise polynomial signal.

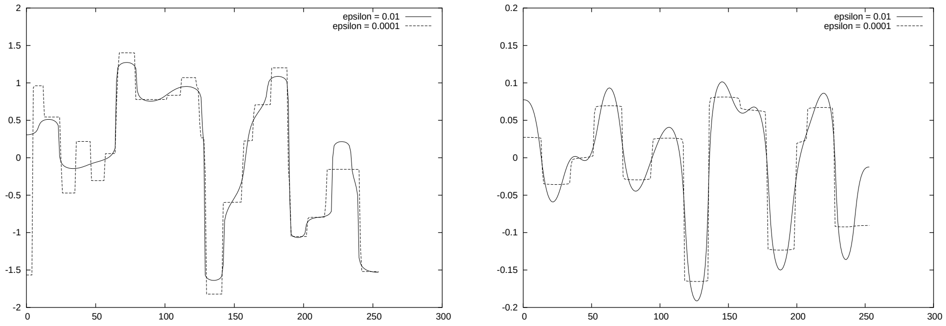

Figure 6.8: Derivative approximations for filtering results. Left: First derivative of second order filtering result. Right: Second derivative of third order filtering result.

10^9 iterations due to the high running time (the SNR and PSNR values were still increasing at that moment).

We see that in terms of SNR and PSNR, the differences between both filtering methods for ε are not very big. The filtering results also look very similar, thus we have only shown the results for $\varepsilon = 10^{-2}$ in Figure 6.7. As a technical remark we should add that the small time step sizes for filtering of third and fourth order (especially with $\varepsilon = 10^{-4}$) demand relatively high calculation precision. For the implementations in C we had to use the floating point data type **double** instead of **float** for the higher order examples.

For a further understanding of the differences between the two cases we have a look at the derivative approximations of the filtering results. The conjecture that mth order total variation approximation filtering yields results that piecewise consist of polynomials of degree $m - 1$ has already been mentioned. The filtering results in Figure 6.7 seem to confirm this hypothesis.

To obtain further indications we consider the finite difference approximation of the $(m - 1)$th derivative for the results of mth order filtering. To corroborate our belief this $(m - 1)$th derivative should be piecewise constant. Figure 6.8 shows that the behaviour of the derivative strongly depends on the value of ε. For $\varepsilon = 10^{-4}$ the derivative approximations really seem to be piecewise constant. Though they are not shown here similar results can also be found for fourth order filtering and the third derivative.

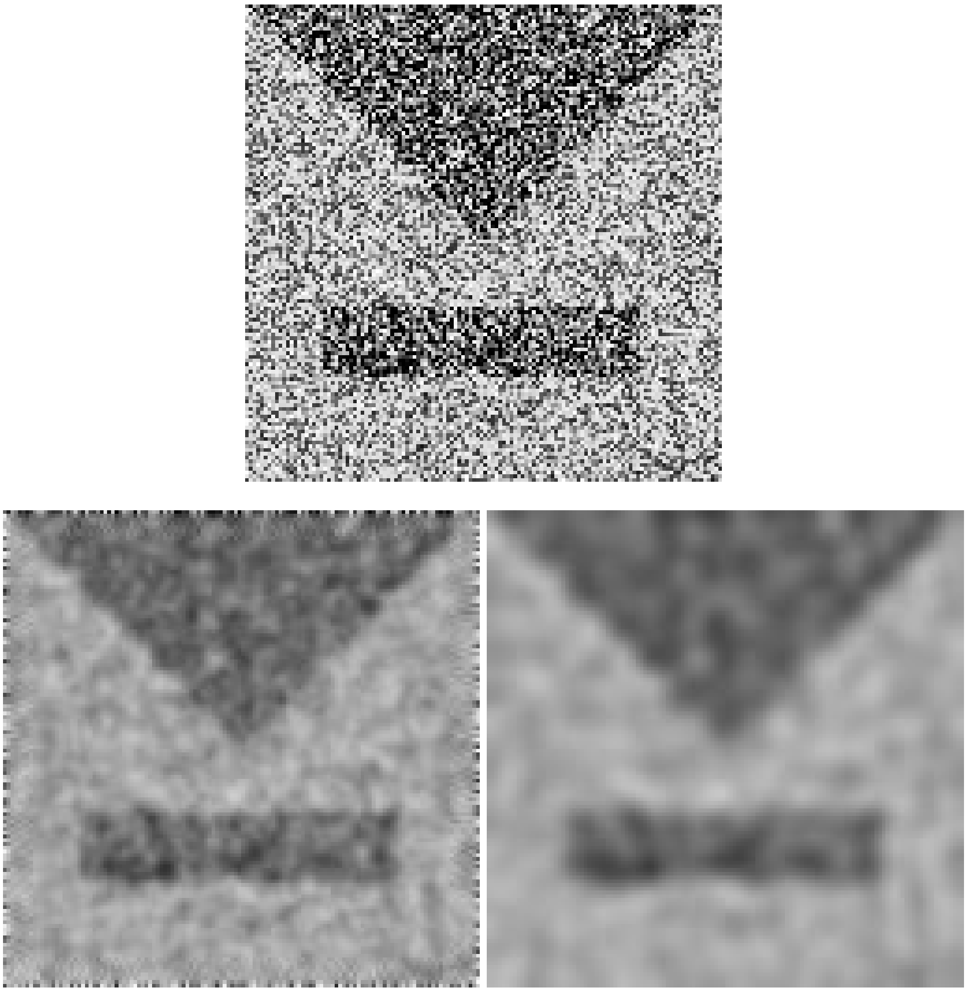

Figure 6.9: Linear filtering with natural and Neumann boundary conditions in 2D. Top: Original image, Bottom left: Natural boundary conditions, Bottom right: Neumann boundary conditions.

6.1.3 Different Penalisers in 2D

First we show an example of 2D linear filtering with natural boundary conditions. We start with the energy functional shown in Example 2.14 with a linear penaliser and use the natural boundary conditions. An example for such a linear filtering is shown in Figure 6.9 compared with Neumann boundary conditions. Since the treatment of the boundary pixels seems not to be appropriate we use Neumann boundary conditions in the following 2D examples.

For a comparison of different penalisers in 2D we use the test image shown in Figure 6.10. Though the original image has the size 256x256 pixels we only show a section of the size 128x128 pixels for a better visualisation of the filtering effects. Figure 6.11 contains the filtering results for linear, Charbonnier and Perona-Malik penalisers. Images filtered with total variation approximations with $\varepsilon = 10^{-2}$ can be seen in Figure 6.12. The corresponding parameters which yield the maximal SNR and PSNR values are listed in Table 6.7. In opposite to the one-dimensional case here we have stopped the computations when SNR and PSNR were decreasing due to computational time. Again we have optimised over the number of iterations for all filters and in addition over the parameter λ for Charbonnier and Perona-Malik

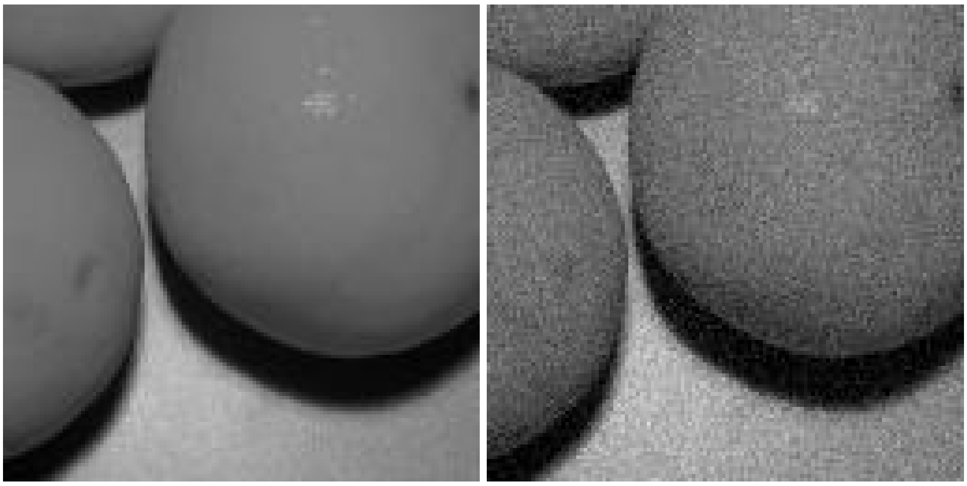

Figure 6.10: Original and noisy version of the 2D example.

Method	Order	SNR	PSNR	Iterations	τ	Parameter
Linear	1	27.57	32.78	4	$2.5 \cdot 10^{-1}$	-
Charbonnier	1	30.65	35.87	45	$2.5 \cdot 10^{-1}$	$\lambda = 0.7$
Perona-Malik	1	30.25	35.46	10	$3.125 \cdot 10^{-2}$	$\lambda = 8.4$
TV approx.	1	30.51	35.72	2991	$2.5 \cdot 10^{-3}$	$\varepsilon = 10^{-2}$
TV approx.	1	30.48	35.69	296650	$2.5 \cdot 10^{-5}$	$\varepsilon = 10^{-4}$
Linear	2	28.74	33.95	22	$3.125 \cdot 10^{-2}$	-
Charbonnier	2	29.71	34.92	2795	$3.125 \cdot 10^{-2}$	$\lambda = 0.1$
Perona-Malik	2	29.47	34.68	1614	$3.125 \cdot 10^{-2}$	$\lambda = 2.5$
TV approx.	2	29.71	34.92	28530	$3.125 \cdot 10^{-4}$	$\varepsilon = 10^{-2}$
TV approx.	2	28.53	33.74	2148300	$3.125 \cdot 10^{-6}$	$\varepsilon = 10^{-4}$

Table 6.7: Parameters and results of SNR/PSNR maximisation in 2D.

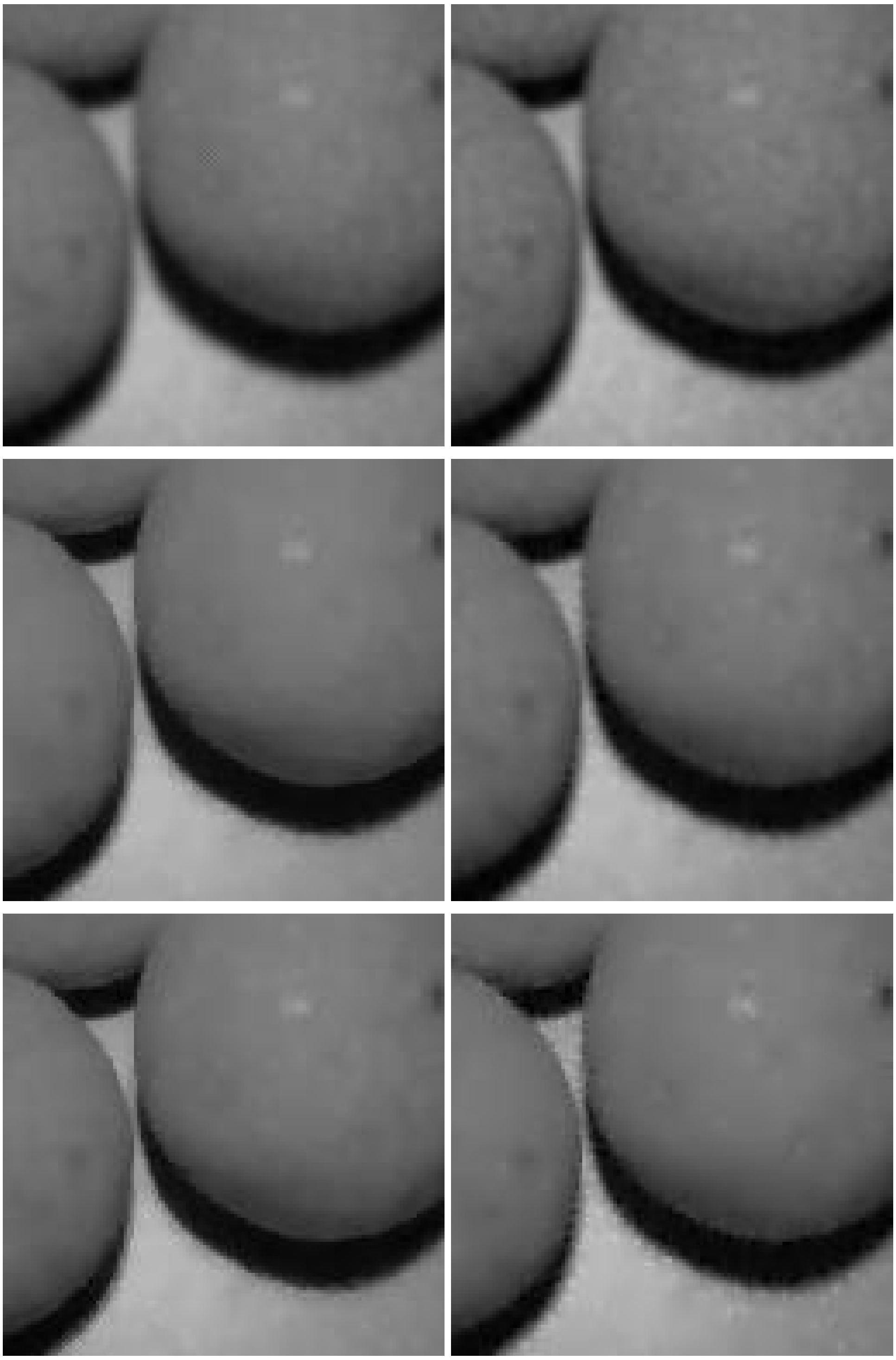

Figure 6.11: Results of SNR maximisation in 2D. Left column: First order filtering, Right column: Second order filtering with Neumann boundary conditions. First row: Linear penaliser, Second row: Charbonnier penaliser, Third row: Perona-Malik penaliser.

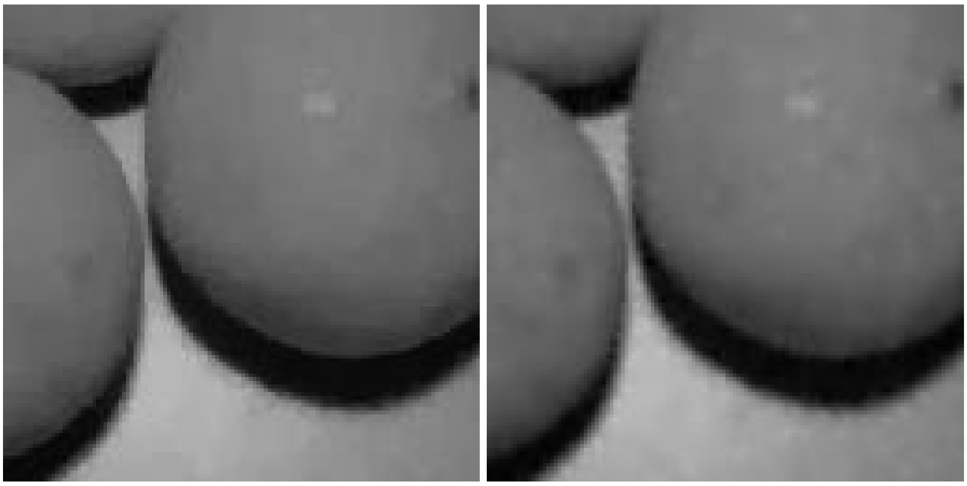

Figure 6.12: Results of SNR maximisation in 2D, Total variation approximation with $\varepsilon = 10^{-2}$. Left column: First order filtering, Right column: Second order filtering with Neumann boundary conditions.

filters. We see that in this case the SNR and PSNR of second order filtering are a bit smaller than the ones for first order filtering.

One can state that the second order filtering results seem to contain isolated white or black pixels or speckle artifacts near edges. Similar observations are mentioned in [34] where an algorithm based on mean and variance in a neighbourhood for the removal of these artifacts is proposed. This would require a postprocessing step after the filtering to remove the speckle artifacts. In Section 6.3 we will give an indication that it could be possible to avoid these artifacts within the higher order filtering framework.

6.2 Different Discretisation Methods

In this section we give an example for the comparison of different filter implementations. We compare explicit schemes for Perona-Malik filtering of order 2 with Neumann boundary conditions. To discretise the Laplacian we use the matrices L (which will be called Laplacian 1) and $\tilde{L}$ (Laplacian 2) from Section 4.2.4 and the matrix D^S from Section 4.3.3. All three filters were started with the same parameters (step size, number of iterations and λ) to get comparable conditions. Figure 6.2 shows the filtering results for all three filters and the pairwise differences between them. Here we take the complete 256x256 pixels of our test image. The image values were rounded and truncated to values from 0 to 255 before taking the differences. For better visualisation the grey values of the difference images were linearly rescaled to the range 0 to 255. To give a quantitative impression the following table contains the minimum, maximum, mean and standard deviation of the difference images before they were rescaled.

	Min	Max	Mean	Standard deviation
L_1 and L_2	0.0	18.0	0.2	0.65
L_1 and S	0.0	24.0	2.0	1.6
L_2 and S	0.0	26.0	2.0	1.7

With a mean of up to 2.0 the differences can be considered as rather small. For practical purposes this could be interesting since spectral methods require two Fourier

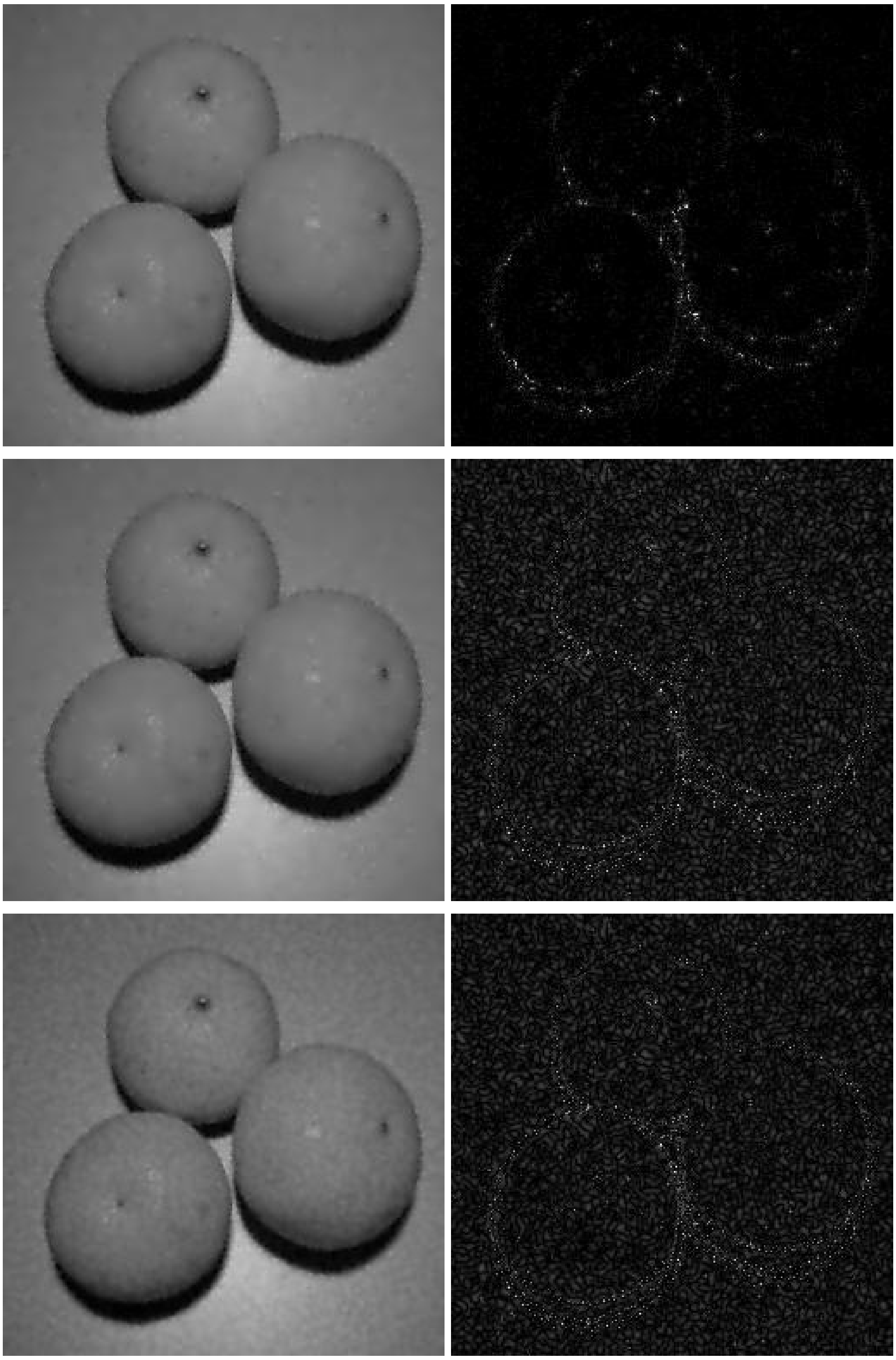

Figure 6.13: Left column: Filtering results. Top left: Laplace 1, Middle left: Laplace 2, Bottom left: Spectral methods. Right column: Differences. Top right: Laplace 1 vs. Laplace 2, Middle right: Laplace 1 vs. Spectral methods, Bottom right: Laplace 2 vs. Spectral methods.

transforms in each step. The complexity is then $O(n \log n)$ if the Fast Fourier Transform can be used or $O(n^2)$ for the regular Discrete Fourier Transform. For finite difference methods one step has complexity $O(n)$ since for each pixel at most a neighbourhood of fixed size has to be considered. If the qualitative differences are not very large, one would prefer the finite difference implementations because they are faster than the spectral methods when an explicit time discretisation is used and the problem is nonlinear. The difference image on the right-hand side of Figure 6.2 shows where the greatest differences appear. We note that the influence of the discretisation methods gets higher near edges of the image.

6.3 Combinations of Different Orders

To conclude this chapter we show some examples of filtering processes of combined first and second order. A nonlinear energy functional in 1D for these filters was presented in Example 2.11. For our filtering example we have used a convex combination of the smoothness terms of first and second order in 2D:

$$\mathcal{E}(u) = \int\limits_{\Omega} \left((u-f)^2 + \alpha \left((1-\beta)\varphi_1 \left(|\nabla u|^2 \right) + \beta \varphi_2 \left((\Delta u)^2 \right) \right) \right) dz.$$

The corresponding diffusion equation in 2D is

$$\begin{aligned} u_t &= (1-\beta)\operatorname{div}\left(\varphi_1' \left(|\nabla u|^2\right) \nabla u\right) - \beta \Delta \left(\varphi_2' \left((\Delta u)^2 \right) \Delta u \right) \\ u(\cdot, 0) &= f \end{aligned}$$

with stopping time α and $\beta \in [0,1]$.

Figure 6.14 shows some filtering tests with a combination of first and second order Perona-Malik filtering. So we are using the diffusivities

$$\varphi_k'(x) = \frac{1}{1 + \frac{x}{\lambda_k^2}} \qquad k = 1, 2$$

in the above equation. The example was obtained using the paramters $\beta = 0.5, \lambda_1 = 5.0$, and $\lambda_2 = 1.0$. The images have size 256x256 pixels. We show here a section of size 128x128 pixels.

The first order filter produces stair-casing in the background while in the second order example the edges contain a lot of speckle artifacts. The combined method reduces both shortcomings without a postprocessing step. We should add that the above results are not optimised with respect to SNR or PSNR. With such an optimisation each of the methods also could lead to better results. This example indicates that the combination of different filter orders could possibly improve filtering results.

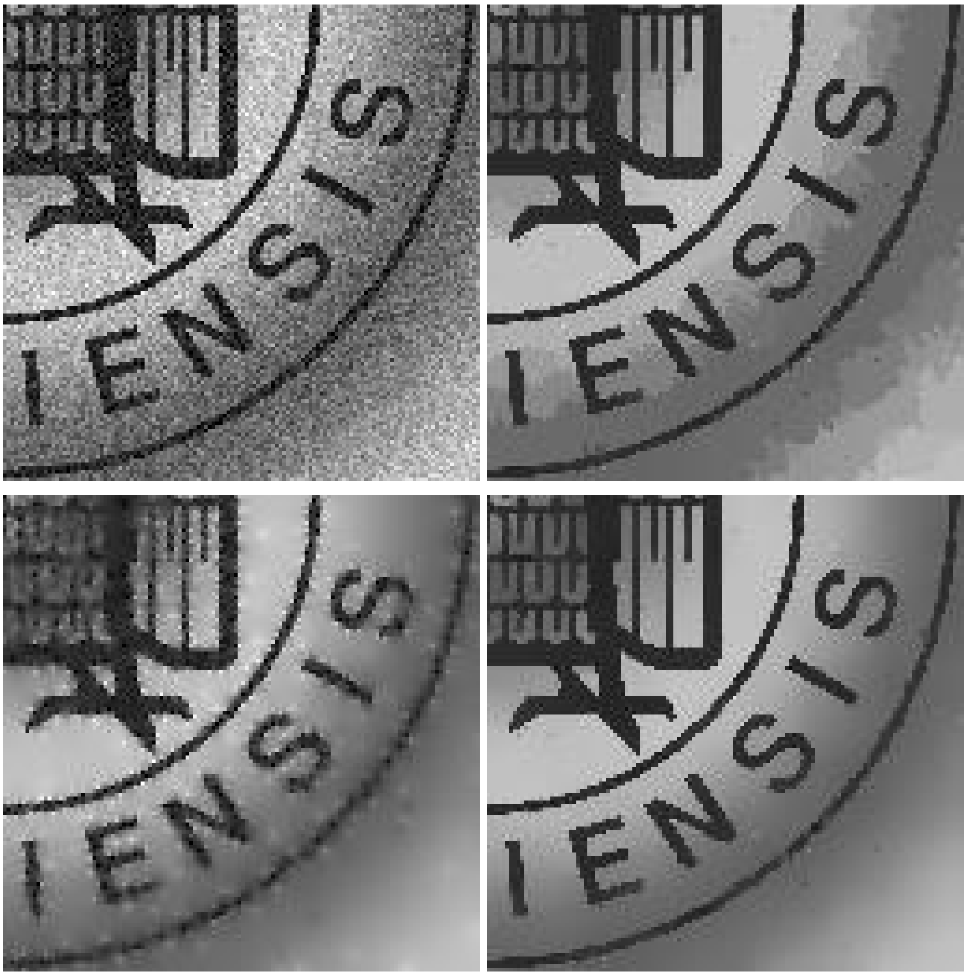

Figure 6.14: Top left: Noisy input image, Top right: First order Perona-Malik, Bottom left: Second order Perona-Malik, Bottom right: Combined filter with first and second order.

Chapter 7

Summary and Outlook

In this thesis we have considered higher order methods for signal and image restoration. We now summarise the main results and point out some open questions and directions for further work.

In Chapter 2 we have discussed necessary and sufficient conditions for minimisers of energy functionals. Here an image is represented by a real-valued function depending on one or two real variables. That means we considered only grey value images. We have developed the Euler-Lagrange equations with the natural boundary conditions for one- and two-dimensional energy functionals.

After investigating energy functionals and corresponding necessary conditions for a minimiser we have turned our attention to strategies how to use these conditions to obtain such a minimiser. In Chapter 3 we considered this filtering process in the framework of continuous functions. For higher order filtering with linear penalisers we oriented at [18]. We discussed also some scale-space properties of linear filtering using higher order linear diffusion equations. Analogously to first order filtering we introduced some well-posedness considerations for nonlinear second order equations of diffusion type. As applications we studied the behaviour for different penalisers. We found out that linear, Charbonnier and total variation approximations always perform forward diffusion. Total variation is the border case between forward and backward diffusion while the Perona-Malik penaliser can perform both forward and backward diffusion and can therefore be ill-posed.

The discrete part of the work started with giving different ways for derivative approximation with finite differences and spectral methods. Our special interest was on natural and Neumann boundary conditions. We derived matrix notations for finite difference and spectral derivative approximations in one dimension with arbitrary derivative order and both types of boundary conditions. We investigated the properties of these matrices in terms of spectral norm and kernel.

We concluded the theoretical part with Chapter 5 by giving discretisations of the approaches in Chapter 3 that were used in concrete algorithms. The semi-discrete case for the corresponding diffusion equations was also investigated. We have derived that one-dimensional finite difference methods with natural boundary conditions yield least-square polynomial data fittings for $t \longrightarrow \infty$ while Neumann boundary conditions lead to the average grey value. Then the main focus was on explicit discretisations of the corresponding diffusion equations. Stability limits for the time step size τ were derived for different derivative orders and approximations. The results of our numerical tests were presented in Chapter 6.

We can conclude that the presented methods seem to produce really better results for signal processing. Besides the optical impression also the SNR and PSNR results are clearly better for higher order methods. Especially the total variations approximations fit in the imagination of yielding piecewise polynomial signals. The second order seems to be a compromise between avoiding stair-casing and too high adaption at the noise. The image processing examples with second order filtering often suffer from artifacts around edges. Our numerical tests suggest that combined methods with first and second order derivatives could reduce or even completely avoid this shortcoming. It also seems to be possible to choose parameters sets such that combined methods of first and second order satisfy a minimum-maximum property. If this observation could be mathematically confirmed the combined nonlinear methods of first and second order would yield a compromise which avoids the stair-casing of first order methods and the edge artifacts of second order methods.

After this short summary of the main results we consider some open questions which invite us to further studies: In real world applications multi-channel images with values in $\mathbb{R}^n$ appear in a natural way. An obvious generalisation for colour images requires the theory for images with values in $\mathbb{R}^3$, for example. One may also think of medical data like NMR images which are matrix-valued. The images may also depend on more than two variables. For such applications the results presented in Chapter 2 have to be extended. For functionals depending only on the first derivatives these extensions are covered in [9]. For higher derivatives this should also be possible without great changes in the reasoning.

For our numerical tests we only used explicit schemes. To make the methods practically useful other algorithms have to be applied which reduce the running time. With implicit schemes for example higher values for the temporal step size τ could be chosen and the number of iterations would become smaller.

The properties of the combined filters of first and second order are another interesting point. In particular the existence of conditions for a maximum-minimum property of these filters seems to be possible.

We should also mention that all methods presented in this work are isotropic. It would be interesting to investigate how higher order anisotropic filters would behave. In the first order case a description of anisotropic filters can be found in [31]. In the higher order framework one may think of second order diffusion along the eigenvectors of the Hessian. Perhaps one could implement coherence-enhancing or edge-enhancing filters similar to the first order case.

Bibliography

[1] R. Acar and C. R. Vogel. Analysis of bounded variation penalty methods for ill-posed problems. *Inverse Problems*, 10:1217–1229, 1994.

[2] A. M. Arthurs. *Calculus of Variations.* Routledge & Kegan Paul Ltd., 1975.

[3] M. Bertero, T. A. Poggio, and V. Torre. Ill-posed problems in early vision. *Proc. IEEE*, 76:869–889, 1988.

[4] G. Boole. *Calculus of Finite Differences.* Chelsea Publishing Company, fourth edition, 1872.

[5] T. Chan, A. Marquina, and P. Mulet. High-order total variation-based image restoration. *SIAM Journal of Scientific Computing*, 22(2):503–516, 2000.

[6] P. Charbonnier, L. Blanc-Féraud, G. Aubert, and M. Barlaud. Two deterministic half-quadratic regularization algorithms for computed imaging. *Proc. IEEE International Conference on Image Processing (ICIP-94, Austin, Now. 13-16, 1994)*, 2:168–172, 1994.

[7] G. Fischer. *Lineare Algebra.* Vieweg Verlag, eleventh edition, 1997.

[8] I. M. Gelfand and S. V. Fomin. *Calculus of Variations.* Prentice-Hall, Inc., Englewood Cliffs, New Jersey, revised english edition, 1963.

[9] M. Giaquinta and S. Hildebrandt. *Calculus of Variations I – The Lagrangian Formalism.* Springer-Verlag, 1996.

[10] J. B. Greer and A. L. Bertozzi. H^1 solutions of a class of fourth order nonlinear equations for image processing. *Discrete and Continuous Dynamical Systems*, 10(1 and 2), January and March 2004.

[11] C. Grossmann and H.-G. Roos. *Numerik partieller Differentialgleichungen.* B. G. Teubner Stuttgart, second edition, 1994.

[12] H. Heuser. *Gewöhnliche Differentialgleichungen – Einführung in Lehre und Gebrauch.* B. G. Teubner Stuttgart, second edition, 1991.

[13] H. Heuser. *Funktionalanalysis – Theorie und Anwendung.* B. G. Teubner Stuttgart, third edition, 1992.

[14] F. B. Hildebrand. *Finite-Difference Equations and Simulations.* Prentice-Hall Inc., Englewood Cliffs, New Jersey, 1968.

[15] R. A. Horn and C. A. Johnson. *Matrix Analysis.* Cambridge University Press, 1985.

[16] T. Iijima. Theory of pattern recognition. *Electronics and Communications in Japan*, pages 123–124, 1963.

[17] M. Lysaker, A. Lundervold, and X.-C. Tai. Noise removal using fourth-order partial differential equations with applications to medical magnetic resonance images in space and time. Technical report, University of California, Los Angeles, Juli 2002.

[18] M. Nielsen, L. Florack, and R. Deriche. Regularization, scale-space and edge detection filters. *Journal of Mathematical Imaging and Vision*, 7:291–307, 1997.

[19] N. Nordström. Biased anisotropic diffusion – a unified regularization and diffusion approach to edge detection. *Image and Vision Computing*, 8:318–327, 1990.

[20] P. Perona and J. Malik. Scale space and edge detection using anisotropic diffusion. *IEEE Trans. Pattern Anal. Mach. Intell.*, 12:629–639, 1990.

[21] L. I. Rudin, S. Osher, and E. Fatemi. Nonlinear total variation based noise removal algorithms. *Physica D*, 60:259–268, 1992.

[22] O. Scherzer and J. Weickert. Relations between regularization and diffusion filtering. *Journal of Mathematical Imaging and Vision*, 12:43–63, 2000.

[23] C. Schnörr. Unique reconstruction of piecewise smooth images by minimizing strictly convex non-quadratic functionals. *Journal of Mathematical Imaging and Vision*, 4:189–198, 1994.

[24] H. R. Schwarz. *Numerische Mathematik*. B. G. Teubner Stuttgart, 1997.

[25] G. D. Smith. *Numerical Solution of Partial Differential Equations: Finite Difference Methods*. Clarendon Press, Oxford, 1978.

[26] J. Stoer and R. Bulirsch. *Introduction to Numerical Analysis*. Springer-Verlag New York, Inc., third edition, 2002.

[27] M. E. Taylor. *Partial Differential Equations I*. Springer-Verlag New York, Inc., 1996.

[28] A. N. Tikhonov. Solution of incorrectly formulated problems and the regularization method. *Soviet Mathematics Doklady*, 4(2):1035–1038, 1963.

[29] L. N. Trefethen. *Finite Difference and Spectral Methods for Ordinary and Partial Differential Equations*. Unpublished, 1996. Available at the URL http://web.comlab.ox.ac.uk/oucl/work/nick.trefethen/pdetext.html.

[30] W. Walter. *Analysis 2*. Springer-Verlag Berlin Heidelberg, fourth edition, 1995.

[31] J. Weickert. *Anisotropic Diffusion in Image Processing*. B. G. Teubner Stuttgart, 1998.

[32] J. Weickert, S. Ishikawa, and A. Imiya. Linear scale-space has first been proposed in Japan. *Journal of Mathematical Imaging and Vision*, 10:237–252, 1999.

[33] D. Werner. *Funktionalanalysis*. Springer-Verlag Berlin Heidelberg, third edition, 2000.

[34] Y.-L. You and M. Kaveh. Fourth-order partial differential equations for noise removal. *IEEE Transactions on Image Processing*, 9(10):1723–1730, October 2000.

[35] F. Zhang. *Matrix Theory – Basic Results and Techniques*. Springer-Verlag New York, Inc., 1999.