Dissertation

Optimal interpolation data for image reconstructions

Laurent Arthur Hoeltgen

Saarbrücken
2014

Dissertation
zur Erlangung des Grades des
Doktors der Naturwissenschaften
der Naturwissenschaftlich-Technischen Fakultäten
der Universität des Saarlandes

Day of Colloquium
30 March 2015

Dean of Faculty
Univ.-Prof. Dr. Markus Bläser

Chair of the Committee
Prof. Dr. Matthias Hein

Reviewers
Prof. Dr. Joachim Weickert
Prof. Dr. Thomas Pock

Academic Assistant
Prof. Dr. Michael Bildhauer

To Damaris

Acknowledgements

I want to express my sincere thanks to all the people who have helped me to finish this PhD thesis. Without them, it would have been impossible to complete my work. I would like to thank Prof. Dr. Joachim Weickert for his excellent mentoring and support during the development of this thesis. I would also like to thank Prof. Dr. Thomas Pock for agreeing to become an external reviewer. My very special thanks go to the whole Mathematical Image Analysis group, especially Simon Setzer, for interesting scientific discussions and providing a great atmosphere that made me enjoy working here. Regarding the administrative problems, I am very grateful to our secretary Ellen Wintringer. Last but not least, I wish to thank Claudine Schiltz, Sylvain Delvaux, my whole family, in particular my parents and my girlfriend Damaris Gatzsche.

Kurzzusammenfassung

Diese Arbeit untersucht mehrere Ansätze zur Bestimmung optimaler dünn besetzter Datensätze für Bildrekonstruktionen mittels linearer homogener Diffusion. Es werden zwei Optimierungsverfahren zur Bestimmung der Position der Datenpunkte präsentiert. Das Erste besticht durch seine Einfachheit und basiert auf Resultaten aus der Spline Interpolationstheorie. Dieses Verfahren kann jedoch nur auf eindimensionale streng konvexe und stetig differenzierbare Signale angewendet werden. Wegen dieser Einschränkungen wird ein alternativer Ansatz hergeleitet der auf Erkenntnissen aus der Theorie der optimalen Steuerung beruht. Dieser neue Algorithmus kann auf beliebige Signale angewendet werden. Beide Methoden werden auf ihre Konvergenzeigenschaften untersucht.

Des Weiteren untersuchen wird das Problem zur Bestimmung guter Datenwerte für feste Positionen, welches im Rahmen der Methode der kleinsten Quadrate untersucht werden kann. Ein wesentlicher Zusammenhang zwischen optimalen Datenpositionen und Datenwerten wird hergeleitet und wir stellen effiziente numerische Verfahren zur Bestimmung dieser Datenwerte dar.

Abschließend präsentieren wir ein Bildkompressionsverfahren das auf den Resultaten aus dieser Arbeit basiert. Experimente beweisen, dass es möglich ist gängige Kompressionsalgorithmen zu schlagen.

Short abstract

This work analyses several approaches for determining optimal sparse data sets for image reconstructions by means of linear homogeneous diffusion. Two optimisation strategies for finding optimal data locations are presented. The first one impresses through its simplicity and is based on results from spline interpolation theory. However, this approach can only be applied to one dimensional strictly convex and differentiable functions. Due to these restrictions we derive an alternative approach which uses findings from optimal control theory. This new algorithm can be applied on arbitrary signals. Both approaches are analysed for their convergence behaviour.

Further, we discuss the problem of selecting good data values for fixed data positions. This problem can be analysed as a least squares problem. An important relationship between the optimal data locations and the data values is derived and we present efficient numerical schemes to obtain these values.

Finally, we present a image compression approach based on the findings from this work. Experiments show that is possible to outperform popular compression algorithms.

Abstract

Finding optimal inpainting data is a key problem for image compression with homogeneous diffusion. Not only the location of important pixels but also their values should be optimal to maximise the quality gain. Both tasks are analysed in this work. Two approaches to find optimal data locations are discussed. The first one is based on findings from spline interpolation theory. It is very simple and only applicable in presence of one dimensional, strictly convex and continuously differentiable signals. Nevertheless it offers important insight into the difficulties of the underlying optimisation task. Our second approach is very generic and applicable to arbitrary data signals. It uses a powerful optimal control based model where we augment the partial differential equation used for the inpainting process with a cost functional containing a data similarity as well as a sparsity inducing term. Both frameworks are analysed with respect to their convergence behaviour. We also establish conditions that assert optimality of the obtained solutions. For the latter model we also provide additional results on how to handle the occurring non-convex optimisation problem from a numerical point of view. Two approaches are discussed. Both methods proceed by an iterative linearisation of the constraint equations. The second method considers an additional reformulation in terms of convex duality. Experimental results on grey value as well as colour images confirm our theoretical insights.

Besides the results on optimal data positions we discuss the task of finding optimal data values for fixed spatial data locations. Our underlying inpainting scheme allows us to formulate this task as a least squares problem. An important relationship between the optimal data positions and the corresponding values is derived. This finding is a fundamental pillar in the design of an competitive image compression codec. Further, we provide efficient numerical schemes to find the best data values. Two algorithms are discussed. One of them excels for CPU based implementations and is based on a well known solver from the literature. The second method exploits findings from primal dual strategies for convex optimisation problems.

It is especially interesting for environments that provide massive parallel processing facilities.

Our work is completed by the presentation of a competitive lossy image compression codec. Experiments show that we outperform popular alternatives such as JPEG and JPEG 2000.

Contents

List of figures

List of tables

List of algorithms

Chapter 1

Introduction

> The beginning is the most important part of the work.
>
> *(Plato)*

A major challenge in data analysis is the reconstruction of a function, for example a 1D signal or an image, from a few data points. Already the ancient Greeks observed celestial movements and built lists, so called ephemerides, to know when to plant their crops. Due to atmospheric conditions hampering the observations these lists were incomplete and the missing positions had to be computed by hand [1]. In today's time, experimental settings in physical sciences usually allow only a limited number of discrete measurements in time or space. Often one wishes to know how the underlying process behaved in between these measurements. A popular example is the weather forecast where information about the current weather is collected at a small number of different locations in order to predict the amount of rainfall over the next few days for a whole region. Another modern use case is given by signals transmitted between electronic devices, potentially separated by any distance ranging from a few metres to several kilometres. The transmitted data could become corrupted by noise or suffer otherwise from loss of information. A phenomenon frequently encountered while listening to the radio or when using a cell phone. Finally, digital cameras with defect CCD sensors can yield photographs where parts of the image are completely missing.

From a mathematical point of view, the restoration of the lost data in all these examples can be regarded as an interpolation or approximation problem. In the context of digital photography the term image inpainting has been used predominantly during the last years. It has initially been

introduced by Masnou and Morel [2] and later again by Bertalmío et al. [3] to the image processing community. However the term had already been in use for decades among art restorers before.

Interpolation in 2D and higher dimensions, as it is required for image and video processing tasks, suffers from the curse of dimensionality. The amount of necessary data to achieve good results can be huge. Further if the data is not well structured (e.g. not laid out on a regular grid) then the reconstruction itself can become difficult to handle. Many approaches require specific layouts for the interpolation data. Unfortunately, setups like the introductory example specified above rarely provide us with such optimal settings.

Due to the special nature of digital photographs, several kinds of different inpainting approaches exist for their treatment. They differ by the application under consideration, for instance inpainting of textured images or movies, and by the mathematical tools used to model the problem (splines, wavelets, partial differential equations (PDEs), ...). Spline-based methods are attractive when speed is an issue. Variational and PDE-based methods stand out through their extraordinary flexibility in the modelling. Especially diffusion equations are appealing since they describe a very natural process to propagate information. One of the earliest variational approaches to inpainting was the pioneering work of Nitzberg et al. [4], even though the authors were not directly interested in the recovery of missing image parts. Their goal was to exploit findings from Gestalt psychology to segment and sort elements of an image scene according to their depth. In order to segment occluded objects the authors relied on the fact that our visual system smoothly extends occluded boundaries. They assumed that these extensions should be as short as possible and smooth and therefore proposed an energy based on Euler's elastica model. A curve C is said to be Euler's elastica if it minimises

$$\int_C \left(\alpha + \beta\kappa\left(s\right)^2\right) \mathrm{d}s$$

among all curves that join a given starting point with its corresponding endpoint. In this context α and β are two positive parameters, κ the curvature of the curve and $\mathrm{d}s$ its arc length. A thorough historical overview of the evolution of the elastica model is given by Levin [5].

Inspired by the results of Nitzberg et al., Masnou and Morel [2] propose an approach to reconstruct missing image parts. They suggest to interpolate

(a) Original image (b) Image with occlusions (c) Reconstruction

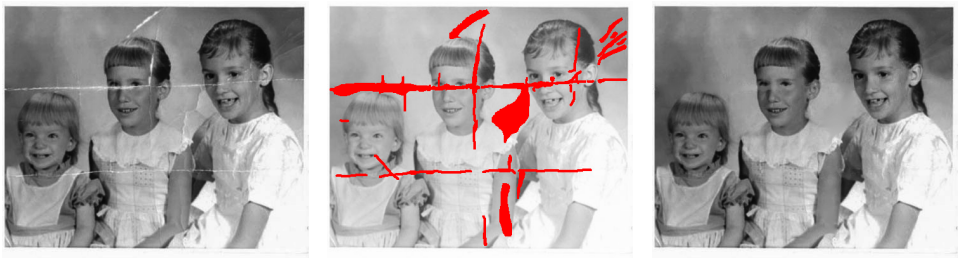
Figure 1.1: Example for the method of Masnou and Morel [2]: The missing parts are reconstructed by extrapolating the isophotes into the white region. The extension to colour images is obtained by applying the method to the luminance, hue and saturation channels separately. Source: [6]

a grey value image by completing the isophotes that arrive at a region with missing data through geodesic curves and refer to this process as disocclusion. The model of Masnou and Morel is a variational model based on the elastica and an example of its visual quality is given in Figure 1.1. Ballester and colleagues [7–9] also propose methods in similar spirit as [2, 4]. They suggest a variational approach based on the elastica model which performs a joint interpolation of certain vector fields and grey levels. Finally, total variation based approaches are proposed in [10–12]. In their simplest form, these methods solve

$$\arg\min_{u} \left\{ \int_{\Omega \setminus \Omega_K} |\nabla u\,(x)|\,\mathrm{d}x \;\middle|\; u(x) = u_0(x)\ \forall x \in \Omega_K \right\}$$

where Ω is the complete image domain and $\Omega_K \subseteq \Omega$ the known pixel data. An interesting feature of this approach is that additional denoising capabilities can be added by replacing the constraint $u = u_0$ with a suitable alternative.

The first PDE-based model that does not stem from a variational formulation is due to Bertalmío et al. [3]. By using a third order transport equation, information is transmitted from the known to the unknown image parts. The authors chose to propagate edge information along the isophotes. This



(a) Original image with undesired scratches

(b) Regions to be inpainted marked in red

(c) Reconstruction

Figure 1.2: Example for the method of Bertalmío et al. [3], Author: Bertalmío [13]

yields the following PDE

$$\begin{aligned} \frac{\partial}{\partial t} u(x,t) &= \langle \nabla(\Delta u(x,t)), \nabla u(x,t)^{\perp} \rangle, && x \in \Omega \setminus \Omega_K \\ u(x,t) &= u_0(x), && x \in \partial\Omega_K,\ t \geqslant 0 \\ u(x,0) &= u_0^{\text{ext}}(x), && x \in \Omega \setminus \Omega_K \end{aligned}$$

where Ω is again the image domain, $\Omega_K \subseteq \Omega$ the known image details and u_0^{ext} a continuous extension of u_0 (only known on Ω_K) onto the whole domain Ω. In order to ensure correct evolution of the direction field, an additional diffusion process has to be interleaved with the previous inpainting process. Basically any sharpness preserving formulation can be used. Bertalmío et al. suggest to use

$$\frac{\partial}{\partial t} u(x,t) = g_\varepsilon(x)\, \kappa(x,t)\, |\nabla u(x,t)|, \quad x \in (\Omega \setminus \Omega_K)_\varepsilon$$

where κ represents the curvature, $(\Omega \setminus \Omega_K)_\varepsilon$ a dilated version of $\Omega \setminus \Omega_K$ and g_ε a smooth function fulfilling $g_\varepsilon(x) \equiv 1$ for all $x \in \Omega \setminus \Omega_K$ and $g_\varepsilon(x) \equiv 0$ for all $x \in \partial(\Omega \setminus \Omega_K)_\varepsilon$. An example reconstruction done with this approach is depicted in Figure 1.2. We refer to [14] for a more complete presentation on PDE-based inpainting strategies and to [6] for a general overview.

So far, all the presented inpainting tasks had a fixed set of given and unknown data. Thus, the only way to improve the reconstruction quality was by using better suited models. Besides repairing corrupted datasets, the presented setups can also purposefully remove certain objects from an image

or a photograph. A closely related task is the seamless integration of new features into a given picture.

In some interesting applications however, one has the total freedom to choose and to modify the data used for the recovery. For instance, in data compression one starts with a full dataset and tries to reduce its size as much as possible. This reduction can be lossless or lossy. In the former case, the recovery is always perfect. The reconstruction yields the initial input signal whereas the latter framework only returns a close approximation to the original data. The advantage of lossy schemes is a significantly higher compression rate. They have been applied with much success in audio codecs such as MP3 [15] and image formats like JPEG [16]. These approaches remove features from a signal that can hardly be perceived by humans. The loss of accuracy goes almost completely unnoticed for most people.

The JPEG [16] compression algorithm is almost image agnostic in the sense that it does hardly analyse the image to find specific features (homogeneous regions, edges, textures, ...) that could be exploited to improve the compression rate. As a consequence it works well on almost any image without excelling for any particular image type. In recent approaches to PDE-based image compression different methods to choose suitable interpolation data were considered. Galić et al. [17] and Schmaltz et al. [18] use subdivision schemes based on the local error of the reconstruction. Mainberger et al. [19] extract edge information from the image to get good seed points for the colour propagation and in [20] searching strategies are employed in a similar manner as in [17, 18]. A related approach can also be found in the works of Demaret and Iske [21] and Demaret et al. [22]. These works show that it is possible to outperform current state-of-the-art compression schemes by optimising the data set used for the reconstruction. Data compression with PDEs is not restricted to images. As shown by Köstler et al. [23] and Schmaltz [24], one can also compress videos. An example for the extraordinary capabilities of PDE-based strategies is shown in Figure 1.3.

Usually it is quite difficult to predict the quality of the reconstruction for a known set of data, especially if the underlying reconstruction mechanism is sophisticated. Galić et al. [17] and Schmaltz et al. [18] use complicated non-linear PDEs for the reconstruction process. The best results are obtained with edge enhancing diffusion (EED)

$$\frac{\partial}{\partial t}u = \operatorname{div}\left(g\left(\nabla u_\sigma \nabla u_\sigma^\top\right)\nabla u\right)$$

(a) Original image (b) JPEG (c) JPEG 2000 (d) PDE-based approach from [19]

Figure 1.3: Comparison of lossy image compression methods: All images had a compression rate of 0.16 bits per pixel. Both JPEG and its successor JPEG 2000 contain visual artefacts whereas the the PDE-based reconstruction is almost flawless. Source: [19]

where u_σ is Gaussian smoothed version of u with standard deviation σ. The function g operates on the eigenvalues of diffusion tensor $\nabla u_\sigma \nabla u_\sigma^\top$. It should be chosen to favour diffusion along edges over diffusion across edges. Edge enhancing diffusion goes back to Weickert [25] where it is introduced for diffusion filtering. The results of Galić et al. and Schmaltz et al. are of remarkable quality. However, it is difficult to state the influence of a single data point on the whole reconstruction in their models. Generally, determining the best interpolation data yields a highly non-trivial optimisation task. Mainberger et al. [20] consider a simple model based on homogeneous diffusion:

$$\begin{aligned} -\Delta u &= 0, \quad \text{on } \Omega \setminus \Omega_K \\ u &= f, \quad \text{on } \partial\Omega_K \\ \partial_n u &= 0, \quad \text{on } \partial\Omega \setminus \partial\Omega_K \end{aligned} \tag{1.1}$$

where Ω is the image domain, Ω_K the known data and ∂_n represents the derivative in outer normal direction along the boundary. They analyse searching strategies for finding optimal positions Ω_K and suggest a least-squares model to obtain good corresponding data values f. Their findings show the tremendous benefits of a thorough optimisation. Unfortunately, their methods come not without restrictions. Although their searching strategies are applicable to any PDE-based inpainting method, they suffer from excessive performance issues, even in the simplest cases. Furthermore, their least-squares model is restricted to linear PDEs. A theoretical discussion

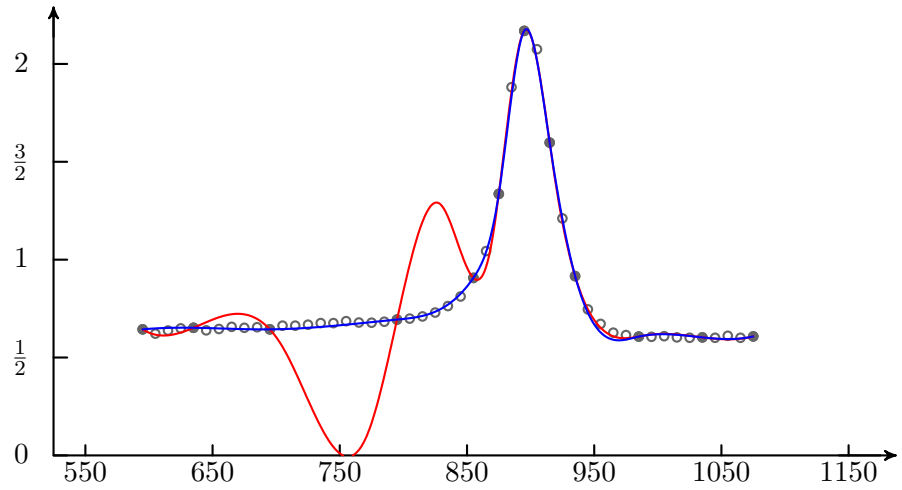


Figure 1.4: Interpolation of the popular titanium heat data (grey circles) [30] at the twelve locations marked by solid grey discs: The blue and red curve correspond to two cubic B-splines with different knot choices. Both splines interpolate the data at the same positions. The blue spline is considerably closer to the original data than the red spline.

on optimal distributions of data points can also be found in the work of Belhachmi et al. [26].

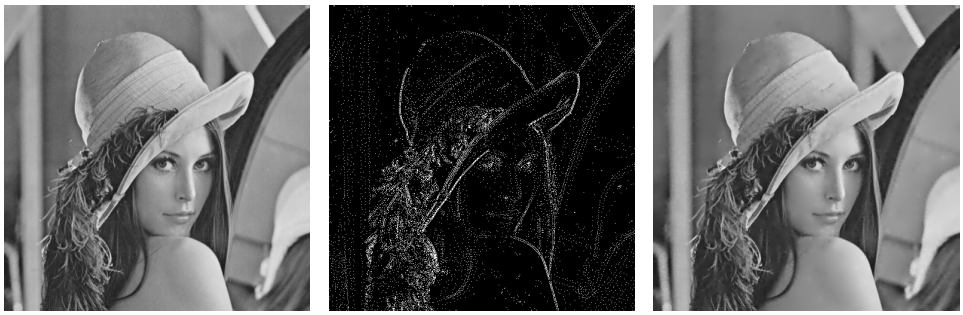
Finding good interpolation data is known in the mathematics literature as free knot (FK) problem. Often it is considered in combination with splines. The term free knot stems from the spline interpolation theory where knots are a characteristic of B-splines. The FK problem can be traced back to at least 1961 [27]. Due to the difficulties in deriving analytic expressions of good data distributions the FK problem has not been as popular as one would expect. Only very few satisfactory results exist, even though these clearly demonstrate the potential benefits of such an optimisation [28, 29]. Figure 1.4 depicts the example for a cubic B-spline interpolation with different knot choices. As one can see from this example, the impact of a badly chosen knot set can be tremendous. Finally, let us also remark that the theory of topology or shape optimisation could also be consulted to find optimal reconstruction data. Unfortunately, these setups often have prohibitive requirements for our purposes. A common assumption is that the considered set is connected or even simply connected. We refer to the the works of Bendsøe and Sigmund [31] and Allaire [32] for a more thorough

presentation and to [33, 34] for more introductory references.

In this work we deliberately restrict us to one of the simplest PDE-based inpainting methods, namely linear homogeneous diffusion, as stated in Eq. (1.1), and present a complete and thorough analysis on how to optimise the data used for the inpainting. Thus, we provide a solid mathematical foundation on which future work can be built upon. Our choice allows us to perceive the importance of individual data points. Equipped with this knowledge we formulate optimisation models yielding those data sets that offer the best accuracy for a given density. Several models are discussed. Our most powerful model being an optimal control (OC) formulation. Optimal control models generalise PDE-based and variational strategies and go back to Bellman [35] and Pontryagin et al. [36]. Their general form is

$$\begin{aligned} &\inf_{x,\lambda} \ \{E(x,\lambda)\} \\ &\text{such that} \quad L(x,\lambda) = 0 \end{aligned}$$

where E is an energy functional to be minimised and $L(x,\lambda) = 0$ a differential equation in x which additionally depends on a parameter λ. It is reasonable to require that for every feasible value of λ there exists exactly one solution x of the differential equation. The minimisation of the energy E helps finding those parameters which yield solutions most favourable to the task at hand. Surprisingly, OC models have received comparably little attention from the image processing community. One of the earliest works using OC for image processing tasks deals with the optical flow problem [37]. The OC model presented in this thesis combines a variational formulation penalising undesired datasets with Eq. (1.1) as a hard constraint. We consider those sets of data locations as undesirable which are either too large or which yield inaccurate reconstructions. Besides this optimisation on the data locations, we also consider strategies to find better data values for the reconstruction. There is no guarantee that the data values provided by the initial image are optimal for the considered location, thus it makes sense to optimise these as well. We follow the ideas already presented in [20] and provide a deep insight on the properties of optimal data values. Combining our OC approach with the findings on perfect grey values provides us a framework, which despite its simplicity, offers high quality reconstructions at compression rates competitive to many well established image compression methods. An example is given in Figure 1.5.



(a) Original image (b) Optimal data locations (c) Reconstruction

Figure 1.5: By optimally choosing 5 % of the pixels from (a) we obtain the result in (c) by solving Eq. (1.1). The selected pixels are marked in white in (b). A Method to find the pixels in (b) is analysed in the forthcoming chapters. Source of the input image: [38]

Structure of the work

In Chapter 2 we present in detail the reconstruction method to be used throughout this work. We use linear homogeneous diffusion inpainting [20, 26], as presented in Eq. (1.1) and which is sometimes also called Laplace- or membrane interpolation. Homogeneous diffusion inpainting is a fast and very flexible way to reconstruct large scattered datasets, especially in higher dimensions. We also discuss a formulation with more relaxed boundary conditions. Discrete analogues of our PDEs are presented and existence and uniqueness of solutions for these discrete versions are discussed. Since the discrete framework can be reduced to a linear system of equations, the spectral properties of the involved matrix are also considered.

Chapter 3 is concerned with the optimisation of the function values for fixed data sites. This task is also referred to as grey value optimisation (GVO) and has already been analysed by Mainberger et al. [20]. Our reconstruction method allows us to formulate this task as a linear least squares problem for which highly efficient strategies are presented. We provide several theoretical findings concerning the optimality of the data values and show how these results can be used to massively reduce the storage size of the data required for the reconstruction. They mark an important milestone in the design of a competitive image compression codec.

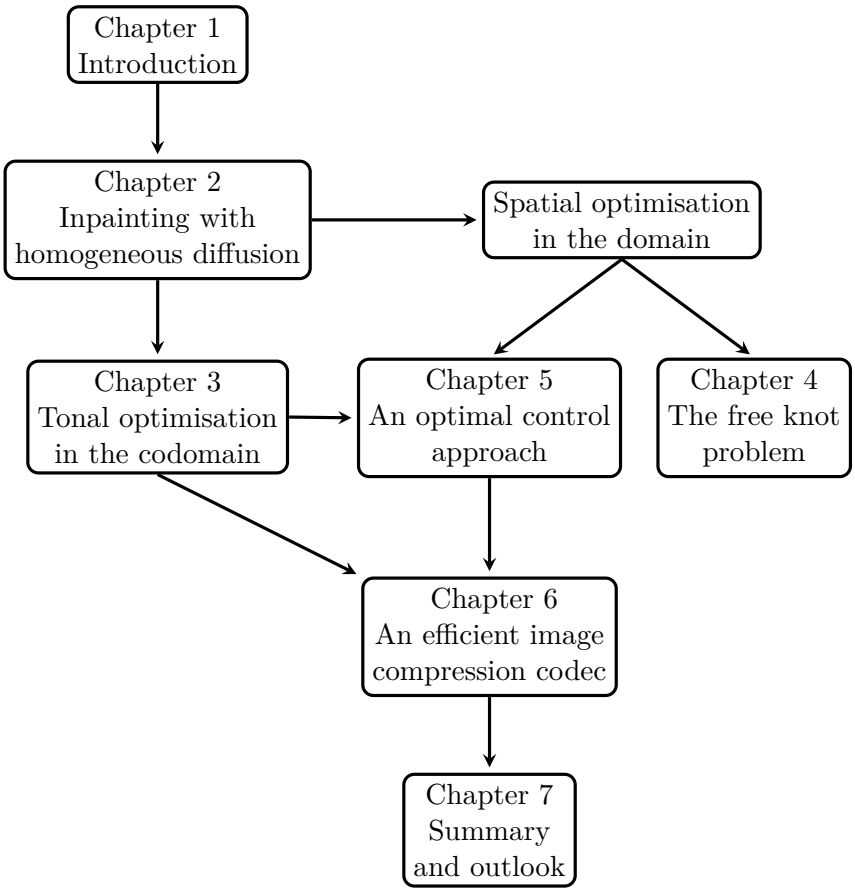
Our first model for finding optimal data locations is presented in Chapter 4. The considered setting is very restrictive. We analyse strictly convex

functions in a one dimensional environment. Within this framework we are able to exploit an important relationship between the input data and the reconstruction. As a consequence we gain some insight in the complexity of the underlying task and are able to derive a simple algorithm which works extraordinarily well. In 1D, our inpainting method coincides with piecewise linear spline interpolation. This fact allows us to bridge the gap between finding good data sites for image inpainting and the FK problem. Important findings from the literature are presented and discussed within our context. Further, we study how our first approach fares against similar strategies.

Chapter 5 presents a completely novel and generic approach based on an OC formulation. We present a highly flexible strategy that works in any dimension without assumptions on the underlying data. We perform a detailed analysis where optimality conditions and existence of solutions are discussed. Next, we propose efficient numeric algorithms to solve the occurring non-convex optimisation problem. An equivalent dual formulation is also given and a thorough convergence analysis of our iterative scheme is exhibited.

Finally, Chapter 6 discusses a complete setup for an image compression codec based on the findings from this thesis. This setup has originally been proposed by Peter [39]. A proper encoding of our optimised data allows us to beat modern image compression standards such as JPEG [16] and JPEG 2000 [40].

A graphical layout of the dependence of the forthcoming chapters is presented on the next page. Note how fundamental the choice of the reconstruction method is. Choosing a different inpainting scheme affects all the other chapters. However, optimisation in the codomain can be completely independent from the optimisation in the domain.


Chapter 1
Introduction
Chapter 2
Inpainting with
homogeneous diffusion
Spatial optimisation
in the domain
Chapter 3
Tonal optimisation
in the codomain
Chapter 5
An optimal control
approach
Chapter 4
The free knot
problem
Chapter 6
An efficient image
compression codec
Chapter 7
Summary
and outlook

Chapter 2

Inpainting with homogeneous diffusion

> Beauty is the first test: there is no permanent place in the world for ugly mathematics.
>
> *(Godfrey Harold Hardy)*

In this chapter we present the reconstruction method that we use throughout this whole work. In order to guarantee a certain freedom and flexibility in the forthcoming optimisation steps we want an interpolation framework that works in any dimension and which can handle arbitrarily scattered data sets. Further, the computation of the interpolated values should be relatively fast and efficient. Partial differential equation-based interpolation methods seem to fit well to our requirements. They can cope very well with highly scattered data and can easily be adapted to any dimensional setting. The interpolation data is usually represented by Dirichlet boundary conditions. Further, an adequate choice of the differential operator allows us to work in arbitrary dimensions and permits us to incorporate other wishful properties directly into the reconstruction process. Thus, one can for example design methods that preserve or even enhance edges or which exhibit a certain degree of smoothness in the results. Unfortunately, many PDEs, especially the non-linear ones, are computationally expensive to solve. This fact essentially restrains us to linear equations. The Laplace equation is among the most prominent and popular PDEs. A numerical reconstruction with the Laplace equation can be reduced to solving a large and sparse linear system. This simplicity of the Laplacian also renders it attractive for our

objectives. The influence of provided data onto the reconstruction can be studied very well and allows us to design efficient optimisation schemes.

Interpolation with the Laplacian is modelled as follows: Let $f : \Omega \to \mathbb{R}$ be a smooth function on some bounded domain $\Omega \subset \mathbb{R}^n$ with a sufficiently regular boundary $\partial\Omega$. Throughout this work we restrict ourselves to the cases $n \in \{1, 2, 3\}$. This choice covers three of the most common types of signals: Simple 1D signals, grey value images, and videos. Most results presented in this chapter can however be extended to settings with arbitrary $n \geqslant 1$ in a straightforward manner. Moreover, let us assume that there exists a closed set of known data $\Omega_K \subsetneq \Omega$ that we interpolate by the underlying diffusion process to recover the missing information on $\Omega \setminus \Omega_K$. Also, for technical reasons, the set Ω_K should be a set with positive measure. Homogeneous diffusion inpainting considers the following PDE with mixed boundary conditions:

$$\begin{aligned} -\Delta u &= 0, \quad \text{on } \Omega \setminus \Omega_K \\ u &= f, \quad \text{on } \partial\Omega_K \\ \partial_n u &= 0, \quad \text{on } \partial\Omega \setminus \partial\Omega_K \end{aligned} \tag{2.1}$$

where Δ represents the Laplacian operator and $\partial_n u$ denotes the derivative of u in outer normal direction. We assume that both boundary sets $\partial\Omega_K$ and $\partial\Omega \setminus \partial\Omega_K$ are non-empty. The setting from Eq. (2.1) is sketched in Figure 2.1. Equations of this type are commonly referred to as mixed boundary value problems and in rare cases also as Zaremba's problem, named after Stanisław Zaremba who studied such equations already in 1910 [41]. The existence and uniqueness of solutions has been extensively studied during the last century. Showing that Eq. (2.1) is indeed solvable is by no means a trivial feat. Generally, one can either show the existence of solutions in very weak settings or one has to impose strong regularity conditions on the domain. A general existence theory for solutions is given by Fichera [42]. Miranda [43] shows that a Hölder continuous solution exists if the data is regular enough and in [44] the author discusses solvability in a general way. More results concerning the existence of solutions have been provided a few years later by Azzam and Kreyszig [45]. Finally, Brown [46] discusses the regularity of solutions on Lipschitz domains. A particularly easy case is the 1D setting where $\Omega \subseteq \mathbb{R}$ and $f \colon \Omega \to \mathbb{R}$. Here, the solution can obviously be expressed by using piecewise linear splines that interpolate the data given on Ω_K.

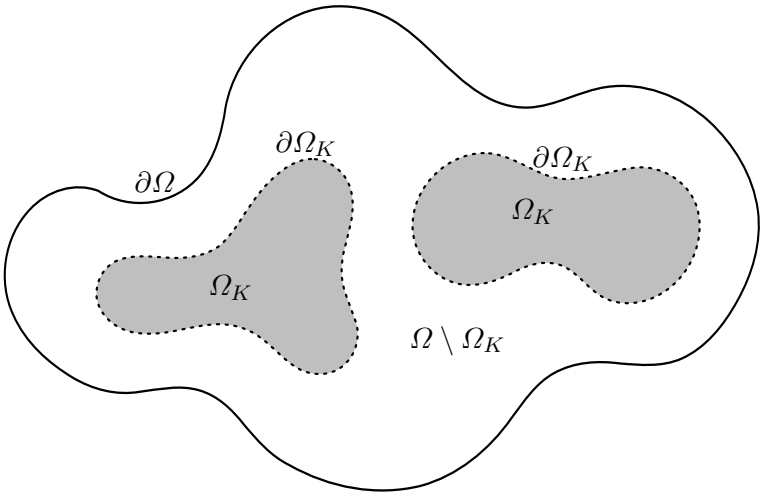


Figure 2.1: Example setup for a mixed boundary problem for PDE-based interpolation with the Laplacian. The set Ω_K, marked in grey, denotes known data and is used to recover the missing information on $\Omega \setminus \Omega_K$ by solving Eq. (2.1). Along the boundary $\partial\Omega_K$ we assume Dirichlet boundary conditions whereas $\partial\Omega \setminus \partial\Omega_K$ has Neumann conditions. Note that the set Ω_K must have positive measure but it is not necessarily connected.

Following Mainberger et al. [20], we introduce the confidence function c which states whether a point is known or not and which is defined by

$$c(x) \coloneqq \begin{cases} 1, & x \in \Omega_K, \\ 0, & x \in \Omega \setminus \Omega_K \end{cases} . \tag{2.2}$$

Note that the confidence function is, at least at this point, nothing else than the indicator function of the set Ω_K. In the forthcoming paragraphs we will relax its definition to allow arbitrary values in $\mathbb{R}$ and not just in the set $\{0, 1\}$. This seemingly insignificant extension greatly simplifies the upcoming optimisation tasks. The confidence function lets us rewrite Eq. (2.1) in a more compact functional form given by

$$\begin{aligned} c(x)\,(u(x) - f(x)) + (1 - c(x))\,(-\Delta)\,u(x) = 0, &\quad \text{on } \Omega \\ \partial_n u(x) = 0, &\quad \text{on } \partial\Omega \setminus \partial\Omega_K \end{aligned} . \tag{2.3}$$

For most parts of this text we will prefer the formulation from Eq. (2.3) as it is more comfortable to work with. Further, this formulation makes also

sense when c is not binary valued but takes arbitrary values. In the binary setting we can either use the diffusion or the data at a given point. If c is allowed to take arbitrary values in the range $[0, 1]$ we obtain combinations between $u - f$ and $-\Delta u$. Thus, we blend the information from the data term $u - f$ with the diffused data given by $-\Delta u$. These combinations are similar to convex combinations but differ in the fact that $c(x)$ may take different values in each x.

Let us now derive a discrete framework and fix certain notational conventions. The notations introduced in this paragraph are, unless mentioned differently, used throughout the whole thesis. We sample our image function f on a regular grid with n_r rows and n_c columns. Thus, we have a total of $n_r n_c$ pixels at our disposal. The distance between any two neighbouring samples on the same row or same column is constant and given by $h > 0$. Further, let $\{1, \dots, n_r n_c\}$ be the set of indices enumerating the discrete sample positions in a linear way (either row-wise or column-wise), and $K \subseteq \{1, \dots, n_r n_c\}$ the subset of indices of known samples. With a slight abuse of notation we can express the discrete version of f as a vector $f = (f_1, \dots, f_{n_r n_c})^\top$ and the corresponding solution of the PDE as a vector u of the same size. Since we work almost exclusively in a discrete setting in the forthcoming chapters of this thesis it is impossible to confuse the function f with its discrete version f. The binary mask c, where $c_i = 1$ if $i \in K$ and $c_i = 0$ otherwise, indicates the positions of the Dirichlet boundary data. At last, the Laplacian Δ is discretised by standard means of finite differences. Hence a straightforward discretisation of Eq. (2.3) on a regular grid yields

$$\operatorname{diag}(c)(u - f) + (I - \operatorname{diag}(c))(-L)u = 0 \tag{2.4}$$

where I is the identity matrix, $\operatorname{diag}(c)$ is a diagonal matrix with the components of the vector c as its entries, and L is the symmetric $n_r n_c \times n_r n_c$ matrix describing the discrete Laplace operator Δ with homogeneous Neumann boundary conditions along $\partial\Omega \setminus \partial\Omega_K$. We also refer to the glossary at the end of this thesis for a complete presentation of all the used symbols.

By a simple reordering of the terms, Eq. (2.4) can be rewritten as the following linear system

$$(\operatorname{diag}(c) + (I - \operatorname{diag}(c))(-L))u = \operatorname{diag}(c) f \ . \tag{2.5}$$

We will refer to this equation as the discrete inpainting equation or simply inpainting equation in the following. Mainberger et al. [19, Theorem 1] show

that this linear system of equations has a unique solution u if all c_i are either 0 or 1 and at least one c_i equals 1. They also show that Eq. (2.5) can be solved efficiently by using bidirectional multigrid methods.

To alleviate the forthcoming discussion we introduce two definitions related to the linear system from Eq. (2.5). The first definition introduces a convenient notation for the system matrix and the second one is useful to express the dependencies of solutions of our discretised PDE on the mask c and the data f.

Definition 2.1 (Inpainting matrix)
We call inpainting matrix *the following* $n_r n_c \times n_r n_c$ *matrix:*

$$A(c) \coloneqq \operatorname{diag}(c) + (I - \operatorname{diag}(c))(-L) \quad .$$

Note that the inpainting matrix lets us rewrite the linear system from Eq. (2.5) as $A(c)\,u = \operatorname{diag}(c)\,f$.

The next definition assumes that we have a mask c to our avail for which the inpainting matrix from the previous definition is invertible. The exact circumstances under which $A(c)$ is invertible will be analysed in the forthcoming section. We remark that invertibility of $A(c)$ is a desirable property since it asserts that solutions of Eq. (2.5) are unique.

Definition 2.2 (Reconstruction matrix)
We call reconstruction matrix *the following* $n_r n_c \times n_r n_c$ *matrix:*

$$M(c) \coloneqq A(c)^{-1} \operatorname{diag}(c) \quad .$$

With the help of the reconstruction matrix it is possible to express the solution of Eq. (2.5) in dependence of c and f. Clearly, we have $u = M(c)\,f$.

As already mentioned, we will make strong use of the fact that c may take arbitrary values. In this context it is important to know for which values of c the reconstruction matrix from the previous definition actually exists. The case of binary masks has already been discussed by Mainberger et al. [19]. In the next section we provide more accurate bounds on the mask values that assert invertibility. Further, we analyse the spectrum of the inpainting matrix and adherence to max-min principles for the reconstruction. These results extend the framework from [19] to the more general setting of non-binary masks.

2.1 Spectral analysis of the inpainting matrix

The eigenvalues of a matrix offer us a complete description of its behaviour. In the following we state upper and lower bounds that guarantee the spectrum of the inpainting matrix $A(c)$ to be real valued and we provide requirements that make sure that the matrix is also invertible.

Proposition 2.3
The inpainting matrix $A(c)$ has a real valued spectrum if $c_i \leqslant 1$ for all i.

Proof. We assume that $c_i \leqslant 1$ holds for all i. In order to show that the spectrum of $A(c)$ is real valued we first apply the following change of variables: $X = (x_{i,j})_{i,j} \coloneqq I - \operatorname{diag}(c)$. This change eases the computations and has no other impact on the results. Thus, we get $A(c) = I - X + X(-L)$ and we define further the diagonal matrix $S = (s_{i,j})_{i,j}$ with entries

$$s_{i,i} \coloneqq \begin{cases} 1, & x_{i,i} = 0 \\ x_{i,i}, & x_{i,i} > 0 \end{cases}, \quad \forall i \ .$$

All other entries of the matrix S are 0. Note that X is a diagonal matrix and, because of our assumption that all mask values are bounded above by 1, it cannot have any negative entries. Therefore, its square root $\sqrt{X}$ exists. Clearly, the matrix S is diagonal, positive definite and invertible, too. It follows that the matrices $\sqrt{S}$, $\sqrt{S}^{-1}$ and S^{-1} also exist. We consider now the following similarity transform

$$\begin{aligned} \sqrt{S}^{-1} A(c) \sqrt{S} &= \sqrt{S}^{-1} (I - X + X(-L)) \sqrt{S} \\ &= I - \sqrt{S}^{-1} X \sqrt{S} + \sqrt{S}^{-1} X(-L) \sqrt{S} \ . \end{aligned}$$

It is easy to see that the identities $\sqrt{S}^{-1} X = \sqrt{X}$ and $\sqrt{S}^{-1} X \sqrt{S} = X$ hold. By combining these two results we obtain the relation

$$\sqrt{S}^{-1} A(c) \sqrt{S} = I - X + \sqrt{X}(-L)\sqrt{S} \ .$$

If we further assume that $x_{i,i} > 0$ (e.g. $c_i < 1$) for all i, then $\sqrt{X} = \sqrt{S}$ and $\sqrt{S}^{-1} A(c) \sqrt{S}$ becomes $I - X + \sqrt{X}(-L)\sqrt{X}$. The latter matrix is symmetric and thus all eigenvalues of $\sqrt{S}^{-1} A(c) \sqrt{S}$ are real valued. We emphasise that at this point it is mandatory to chose a discretisation of the

0	0	0
α	α	α
0	0	0

Figure 2.2: Inpainting mask of a 3×3 image. For $\alpha > 1$, the corresponding inpainting matrix $A(c)$ has complex eigenvalues.

Laplacian such that L is symmetric. We cannot make any claims otherwise. We conclude that the spectrum of $A(c)$ is already real valued if $c_i < 1$ holds for all i. This result follows from the fact that the spectrum of a matrix is invariant under similarity transforms.

Let us now assume that $c_i \leqslant 1$ is fulfilled for all i and that $c_i = 1$ holds for certain i. Let again X be given by $I - \operatorname{diag}(c)$. Further, let us assume that $A(c)$ has at least one complex eigenvalue. Then there exists an eigenvalue $\lambda = \alpha + \imath\beta$ with $\alpha \in \mathbb{R}$ and $\beta < 0$ or $\beta > 0$. Finally, let $\varepsilon > 0$ be a fixed but arbitrary real positive number and define $X_\varepsilon \coloneqq X + \varepsilon I$. Our previous result states that the matrix $A(c)_\varepsilon \coloneqq I - X_\varepsilon + X_\varepsilon(-L)$ has a real spectrum for any $\varepsilon > 0$. Further, $A(c)_\varepsilon$ converges component wise towards $A(c)$ for ε going to 0. Since the eigenvalues depend continuously on the matrix entries [47, Appendix D], it follows that $A(c)$ cannot have a complex eigenvalue. Otherwise there would have to exist a $\varepsilon > 0$ such that $A(c)_\varepsilon$ already has a complex eigenvalue with imaginary part $\frac{\beta}{2}$. This would be a contradiction. □

Note that the upper bound of 1 for the mask entries is actually strict. Let us consider an arbitrary 3×3 image and the mask given in Figure 2.2. For such a small example all eigenvalues of the 9×9 inpainting matrix $A(c)$ can be stated explicitly. They are given by

$$\begin{gathered} 1, 1, 2, 4, -2(\alpha - 2), \\ \frac{1}{2}\left(-\sqrt{\alpha^2 - 10\alpha + 9} - \alpha + 3\right), \frac{1}{2}\left(\sqrt{\alpha^2 - 10\alpha + 9} - \alpha + 3\right), \\ \frac{1}{2}\left(-\sqrt{16\alpha^2 - 16\alpha + 9} - 4\alpha + 9\right), \frac{1}{2}\left(\sqrt{16\alpha^2 - 16\alpha + 9} - 4\alpha + 9\right) \end{gathered}$$

for which the values become complex valued as soon as $\alpha > 1$ since $\alpha^2 - 10\alpha + 9$ is negative then. In this example, we have discretised the Laplacian by using the stencil depicted in Figure 2.3 (a) and setting $h = 1$. The pixels have

0	$\frac{1}{h^2}$	0
$\frac{1}{h^2}$	$-\frac{4}{h^2}$	$\frac{1}{h^2}$
0	$\frac{1}{h^2}$	0

(a) Stencil of the discrete Laplacian

0	$\frac{-1+c_i}{h^2}$	0
$\frac{-1+c_i}{h^2}$	$c_i + \frac{4(1-c_i)}{h^2}$	$\frac{-1+c_i}{h^2}$
0	$\frac{-1+c_i}{h^2}$	0

(b) Stencil of the inpainting matrix

Figure 2.3: Convolution stencils of the discrete Laplacian and of the inpainting matrix $A(c)$ for inner pixels. The stencil of the inpainting matrix depends on the position where it is evaluated. At boundary pixels both stencils must be adapted to reflect the imposed Neumann boundary conditions.

been labelled column-wise. Finally, it is interesting to note that we have not encountered any lower bounds for ensuring a real valued spectrum. Basically, the mask values can even be negative.

2.2 Invertibility of the inpainting matrix

In the forthcoming paragraphs we turn our attention towards the invertibility of the inpainting matrix. Our goal is to exclude 0 from the eigenvalue spectrum. For technical reasons we restrict ourselves to the 2D case (e.g. $\Omega \subseteq \mathbb{R}^2$). Other environments can be handled analogously and require little to no changes on the proofs. Certain results from the 1D setting (e.g. $\Omega \subseteq \mathbb{R}$) are also required in the upcoming chapters. We mention them explicitly whenever they differ from the two dimensional environment.

As already mentioned, we assume that our image data is given on a regular grid with n_r rows, n_c columns, and a grid size of $h > 0$ in each direction. The simplest possible discretisation for the Laplacian can be expressed by a convolution with the stencil presented in Figure 2.3 (a). This choice will be analysed in the forthcoming paragraphs. Results for other discretisations can be done analogously. We remind that there exists an intimate one-to-one relationship between stencils for convolutions and matrices. The i-th row of a matrix contains the entries for the convolution stencil to be applied on the i-th data point of the signal. Thus, it is straightforward to derive the stencil corresponding to the inpainting matrix by computing the entries of $A(c)$ and vice versa. The resulting stencil for the inpainting matrix is depicted in Figure 2.3 (b). Note that the stencils from Figure 2.3 are only valid for inner

pixels. Along the image bounds the stencil has to be adapted to adhere to the imposed boundary conditions. It is easy to verify that in general the inpainting matrix $A(c)$ has the entry $c_i + |N(i)|(1 - c_i)h^{-2}$ on its i-th entry of the main diagonal. Here $N(i)$ is the set of existing direct neighbours to pixel i and $|N(i)|$ is its cardinality. The $|N(i)|$ non-zero off-diagonal entries in the same row are all given by $(c_i - 1)h^{-2}$.

Our next goal is to obtain an estimate on the allowed range for the mask values c_i such that we can assert invertibility of the inpainting matrix. Clearly, the i-th row of the matrix $A(c)$ depends only on c_i, whereas the i-th column may depend on all pixels from $N(i)$. Thus, we can derive pointwise estimates by applying Geršgorin's circle theorem [48, Theorem 1.1] onto the rows of $A(c)$. For the sake of completeness, we recall this important result of Geršgorin but refer to [48] for its proof.

Theorem 2.4 (Geršgorin's circle theorem)
For any matrix $A \in \mathbb{C}^{n,n}$ with entries $a_{i,j}$ and any eigenvalue $\lambda \in \mathbb{C}$ of A, there is a positive integer $k \in \{1, \dots, n\}$ such that

$$|\lambda - a_{k,k}| \leqslant \sum_{\substack{j=1 \\ j \neq k}}^{n} |a_{k,j}| \ .$$

The previous theorem allows a beautiful geometric interpretation. The sets

$$D_k := \left\{ z \in \mathbb{C} \;\middle|\; |z - a_{k,k}| \leqslant \sum_{j \neq k} |a_{k,j}| \right\}$$

with $k \in \{1, \dots, n\}$ represent discs in the complex plane. The theorem states that for every eigenvalue there exists such a disc which encloses it. Thus, by analysing the extent of all the discs it is possible to get estimates for eigenvalue candidates. We exploit this finding to obtain approximate bounds for the mask values that assert invertibility of the inpainting matrix.

Proposition 2.5
Let $A(c)$ be the inpainting matrix with mask c corresponding to the stencil in Figure 2.3 (b) and assume that the grid size is $h > 0$. Then, all eigenvalues of $A(c)$ are non-negative and 0 cannot lie in any of the row-wise computed Geršgorin discs of $A(c)$ if Eq. (2.6) holds for every mask value c_i. It follows that 0 cannot be an eigenvalue. Conversely, if any c_i takes the value 0 or

$2|N(i)|(2|N(i)| - h^2)^{-1}$ *then 0 lies on the boundary of the corresponding Geršgorin disc.*

Proof. In order to exclude 0 as an eigenvalue we have to assert that it is enclosed in none of the discs defined by the previous theorem. All the entries of the inpainting matrix are explicitly known and a straightforward computation reveals

$$\underbrace{c_i + |N(i)|\,(1 - c_i)\,h^{-2}}_{=A(c)_{i,i}} - |N(i)| \underbrace{|(1 - c_i)|h^{-2}}_{=A(c)_{i,j},\ \forall j \in N(i)} > 0, \quad \forall i$$

as a requirement to exclude 0 as candidate for an eigenvalue. This inequality can be reduced to

$$\begin{cases} 0 < c_i, & |N(i)| \leqslant \frac{h^2}{2} \\ 0 < c_i < \frac{2|N(i)|}{2|N(i)| - h^2}, & |N(i)| > \frac{h^2}{2} \end{cases} \quad \forall i \ . \tag{2.6}$$

Clearly if any mask value attains the value 0 or $2|N(i)|(2|N(i)| - h^2)^{-1}$, then the boundary of the corresponding Geršgorin disc will pass through 0. □

Note that $|N(i)| > \frac{h^2}{2}$ can also be rewritten as $h < \sqrt{2}\sqrt{|N(i)|}$. In our setting $|N(i)|$ can take at most the value 4. This implies that a grid size $h > 2\sqrt{2}$ leaves us with the sole requirement that all c_i must be positive to prevent 0 from being a candidate for an eigenvalue. One of the most frequent choices for the grid size is $h = 1$. This yields the upper bounds $\frac{8}{7}$ for inner pixels, $\frac{6}{5}$ for boundary pixels and $\frac{4}{3}$ for corner pixels. A visualisation of the Geršgorin discs of the inpainting matrix for different mask values is given in Figure 2.4. We also note that an almost identical computation for the 1D setting yields the upper bound of $\frac{4}{3}$ for the mask values. Finally let us remark that different discretisations of the Laplacian lead to different upper bounds. Nevertheless, the computations are identical in every case.

Unfortunately, enforcing all c_i to be positive to assert invertibility of the matrix $A(c)$ is too prohibitive for our goals. We would like to completely turn off the influence of the data at specific locations. Further, our results should generalise the findings of Mainberger et al. [19] where it was possible for the mask to take the value 0. Thus, we need to investigate on further results to weaken the requirements from Proposition 2.5.

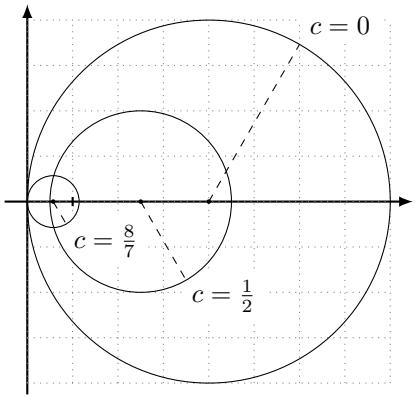


Figure 2.4: Visualisation of the Geršgorin discs for $c = 0$, $c = \frac{1}{2}$ and $c = \frac{8}{7}$ with grid size $h = 1$ for some inner pixel ($|N(i)| = 4$). The vertical tick is positioned at $x = 1$ and marks the position of the Geršgorin disc for $c = 1$ with radius 0. Note that the centre of the circles move from $x = 4$ to $x = 1$ when c changes from 0 to 1. At the same time the radius of the circles get smaller. For c varying from 1 to $\frac{8}{7}$ the centres move from $x = 1$ to $x = \frac{4}{7}$ and the radius increases from 0 to $\frac{4}{7}$.

Our next step is to show that the inpainting matrix is invertible as soon as one mask entry fulfils the requirements from Eq. (2.6). Our strategy employs ideas from [19]. However, due to the more general setting, a certain number of additional steps need to be taken. We proceed as follows: First, we show that our inpainting matrix is a so called block irreducible matrix. Then, by using a theorem from [49], we provide less restrictive conditions that exclude 0 as a candidate for an eigenvalue. To this end we show that 0 can only be an eigenvalue if it lies on the boundary of all Geršgorin discs. This setup is impossible as soon as as a single mask value fulfils the requirements from Proposition 2.5.

Several classic results from matrix analysis and graph theory are required. We provide them in the following paragraphs and begin our presentation on preliminary results with two common definitions concerning structured matrices. They stem from [48, Chapter 1.2].

Definition 2.6 (Permutation matrix.)
A matrix $P \in \mathbb{R}^{n,n}$ is said to be a permutation matrix *if there exists a permutation ϕ, i.e. a bijective mapping from the set $\{1, 2, \ldots, n\}$ onto itself,*

such that $P = \left(\delta_{i,\phi(j)}\right)_{i,j}$*, where* $\delta_{k,l}$ *is the Kronecker delta function*

$$\delta_{k,l} \coloneqq \begin{cases} 1, & k = l \\ 0, & k \neq l \end{cases} .$$

Definition 2.7 (Reducible matrix)
A matrix $A \in \mathbb{C}^{n,n}$ *with* $n \geqslant 2$ *is* reducible *if there exists a permutation matrix* $P \in \mathbb{R}^{n,n}$ *and a positive integer* r*, with* $1 \leqslant r < n$ *for which*

$$PAP^{\top} = \begin{pmatrix} A_{1,1} & A_{1,2} \\ 0 & A_{2,2} \end{pmatrix} .$$

Here, $A_{1,1} \in \mathbb{C}^{r,r}$*,* $A_{1,2} \in \mathbb{C}^{r,n-r}$ *and* $A_{2,2} \in \mathbb{C}^{n-r,n-r}$ *are arbitrary submatrices. If no such permutation matrix exists,* A *is said to be* irreducible*. For scalars we say that they are irreducible if they are non-zero and reducible else.*

The term reducible matrix stems from the following fact: Let us assume A is a reducible matrix and that we want to solve the linear system $Ax = b$. By applying the definition of reducibility, we can change the initial problem into two linear systems of smaller size:

$$\begin{aligned} A_{1,1}y + A_{1,2}z &= c \ , \\ A_{2,2}z &= d \ . \end{aligned}$$

By solving the second system $A_{2,2}z = d$, we can rewrite the first system as $A_{1,1}y = c - A_{1,2}z$. Often, these two reduced systems can be solved significantly faster than the original problem.

There exists a strong relationship between matrix analysis and graph theory. The idea is that one can associate matrices to graphs in such a way that properties of graphs reflect certain characteristics of matrices. This relationship is also quite useful to obtain a non-algebraic interpretation of the concept of an irreducible matrix.

Let us shortly introduce some very basic notions from graph theory. Consider any matrix $A \in \mathbb{R}^{n,n}$ with non-negative entries as well as n pairwise distinct elements v_j from an arbitrary set of the same size. The elements v_j are commonly called vertices. For each non-zero entry $a_{i,j}$ of A we connect the vertex v_i with the vertex v_j using a directed arc going from v_i to v_j

(denoted by $\overrightarrow{v_i v_j}$). The entry $a_{i,j}$ in the matrix A can be interpreted as a cost for going from v_i to v_j. The set of all such directed arcs is called directed graph and denoted by $\mathbb{G}(A)$. Further, a directed path in $\mathbb{G}(A)$ is a collection of abutting directed arcs $\overrightarrow{v_{k_1} v_{k_2}}$, $\overrightarrow{v_{k_2} v_{k_3}}$, …, $\overrightarrow{v_{k_{r-1}} v_{k_r}}$ connecting the initial vertex v_{k_1} with the terminal vertex v_{k_r}. In this context, the matrix A is also referred to as the (weighted) adjacency matrix of $\mathbb{G}(A)$ and the number of abutting arcs is called length of the path. The following definition can be found in [48, Definition 1.8] and denotes an important property of graphs.

Definition 2.8 (Strongly connected graph)
The directed graph $\mathbb{G}(A)$ of a matrix A is strongly connected *if, for each ordered pair v_i and v_j of vertices, there is a directed path in $\mathbb{G}(A)$ with initial vertex v_i and terminal vertex v_j.*

Strongly connected graphs are those graphs where any vertex can be reached from any other vertex in an arbitrary number of steps. Figure 2.5 (a) depicts an example of a strongly connected graph and Figure 2.5 (b) presents a graph that is not strongly connected. The corresponding adjacency matrices A_1 for Figure 2.5 (a) and A_2 for Figure 2.5 (b) are given by

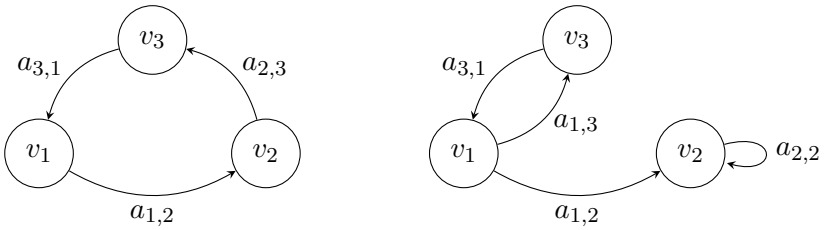
$$A_1 = \begin{pmatrix} 0 & a_{1,2} & 0 \\ 0 & 0 & a_{2,3} \\ a_{3,1} & 0 & 0 \end{pmatrix}, \quad A_2 = \begin{pmatrix} 0 & a_{1,2} & a_{1,3} \\ 0 & a_{2,2} & 0 \\ a_{3,1} & 0 & 0 \end{pmatrix} . \tag{2.7}$$

We remark that the non-zero entries in an adjacency matrix A tell us which nodes can be reached by traversing a path of length 1. Similarly, the non-zero entries in A^2 tell us which vertices are linked to each other by a path of length 2. In general, A^k indicates the vertices that are connected by a path of length k.

The next proposition exhibits the announced important relationship between graph theory and matrix analysis. It stems from [48, Theorem 1.9]. We refer to the source for a detailed proof of the statement.

Proposition 2.9
Any matrix $A \in \mathbb{R}^{n,n}$ with non-negative entries is irreducible if and only if its directed graph $\mathbb{G}(A)$ is strongly connected.

This finding yields a very comfortable way to verify that a matrix is irreducible and will be essential in our demonstration that the inpainting matrix is invertible.



(a) A graph which is strongly connected. The corresponding adjacency matrix is given as A_1 in Eq. (2.7)

(b) A graph which is not strongly connected. It is not possible to reach v_3 from v_2. The corresponding adjacency matrix is given as A_2 in Eq. (2.7)

Figure 2.5: Examples of a strongly connected and not strongly connected graphs. The first graph is not strongly connected, since there is no path going from v_2 to v_3.

The concept of irreducible matrices has a useful generalisation. Often, matrices suggest a block structure in the sense that its entries can be grouped into rectangular blocks and that these blocks follow a certain pattern. In this context it may be interesting to analyse the irreducibility with respect to these blocks. This idea is introduced in the following definition. It stems from [49, Definition 2].

Definition 2.10 (Block irreducible matrix)
Let A be a $n \times n$ matrix with complex entries, partitioned as follows:

$$A = \begin{pmatrix} A_{1,1} & A_{1,2} & \dots & A_{1,m} \\ A_{2,1} & A_{2,2} & \dots & A_{2,m} \\ \vdots & \vdots & \ddots & \vdots \\ A_{m,1} & A_{m,2} & \dots & A_{m,m} \end{pmatrix} .$$

The submatrices $A_{i,i}$ are assumed to be square of order $n_i < n$ for all i. We call the matrix A block irreducible *if the $m \times m$ matrix*

$$\begin{pmatrix} \|A_{1,1}\| & \|A_{1,2}\| & \dots & \|A_{1,m}\| \\ \|A_{2,1}\| & \|A_{2,2}\| & \dots & \|A_{2,m}\| \\ \vdots & \vdots & \ddots & \vdots \\ \|A_{m,1}\| & \|A_{m,2}\| & \dots & \|A_{m,m}\| \end{pmatrix}$$

is irreducible. Here, $\|\cdot\|$ is an arbitrary matrix norm.

Our goal in this section is to provide conditions for which the inpainting matrix $A(c)$ is block irreducible. Later we show how this property helps us to exclude 0 from the eigenvalue spectrum. Another common matrix structure is a band matrix which has non-zero entries only along certain diagonal bands. We now state a rigorous definition.

Definition 2.11 (Band matrix)
Let $p, q \in \mathbb{N}$ be positive numbers. We say that a matrix $A \in \mathbb{C}^{n,n}$ with entries $a_{i,j}$ and with $n \geqslant \max\{p, q\}$ is a band matrix with bandwidth $p+q+1$ if the entry $a_{i,j} = 0$ for all i, j such that $j + p < i$ or $i + q < j$. If $p = q = m$, we call the $2m+1$ diagonals of A that can be non-zero the main diagonals of A.

Band matrices often stem from the discretisation of differential equations. The discrete version L of the Laplacian as well as the inpainting matrix defined in this chapter are band matrices. Further, these matrices also have a block structure. An observation that we exploit in the following. A special case of band matrices is given by tridiagonal matrices. They correspond to setting $p = q = 1$ in the previous definition.

The next lemma shows how the bandwidth is changing if band matrices are applied onto each other.

Lemma 2.12
If $A \in \mathbb{R}^{n,n}$ is a band matrix where all entries on the $2m+1$ main diagonals are positive and $B \in \mathbb{R}^{n,n}$ is a tridiagonal matrix where all entries on the main diagonals are positive, too, then the product $AB \in \mathbb{R}^{n,n}$ is a band matrix where all entries on the $2m+3$ main diagonals are positive.

Proof. Clearly the entry $a_{i,r}$ is positive if and only if $|i - r| \leqslant m$ and similarly the entry $b_{r,j}$ is positive if and only if $|r - j| \leqslant 1$. Since the (i, j)-th entry of AB is given by

$$\sum_{r=1}^{n} a_{i,r} b_{r,j} \ ,$$

it remains to show that this sum differs from 0 if and only if $|i - j| \leqslant m + 1$. Note that all entries in A and B are non-negative. Thus it cannot happen that multiple terms of the sum cancel out. Obviously the term $\sum_{r=1}^{n} a_{i,r} b_{r,j}$ is non-zero if and only $[i - m, i + m] \cap [j - 1, j + 1] \cap \mathbb{Z}$ is not the empty set. But this assertion is equivalent to

$$i + m \geqslant j - 1 \quad \text{or} \quad i - m \leqslant j + 1 \ .$$

Finally, the latter two inequalities can be merged into a single expression given by $|i-j| \leqslant m+1$. This concludes the proof. □

Corollary 2.12.1
If $A \in \mathbb{R}^{n,n}$ is a tridiagonal matrix where all entries on the three main diagonals are positive, then there exists an integer $k > 1$ such that A^k is a full matrix where all entries are positive.

Proof. Applying Lemma 2.12 iteratively on the matrices A^k and A for $k \geqslant 1$ shows that the bandwidth is increasing by one with each power. Thus, the matrix will be full for $k = n-1$. □

Corollary 2.12.2
A tridiagonal matrix $A \in \mathbb{R}^{n,n}$ where all the entries on the 3 main diagonals are positive is irreducible.

Proof. Corollary 2.12.1 asserts that there exists a k such that A^k is a full matrix. This implies that any vertex in $\mathbb{G}(A)$ can be reached from any other vertex by a path of length k. Thus, A is irreducible. □

Let us assume for a moment that the pixels in the image are labelled column-wise. Then our inpainting matrix $A(c)$ has block tridiagonal structure. This means that we can partition it as follows:

$$\begin{pmatrix} A_{1,1} & A_{1,2} & 0 & 0 & \dots & 0 & 0 \\ A_{2,1} & A_{2,2} & A_{2,3} & 0 & \dots & 0 & 0 \\ 0 & A_{3,2} & A_{3,3} & A_{3,4} & \dots & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & 0 & 0 & \dots & A_{n_c,n_c-1} & A_{n_c,n_c} \end{pmatrix}.$$

Obviously, the tridiagonal submatrices $A_{k,k} \in \mathbb{R}^{n_r,n_r}$ on the main diagonal can be never be 0 in each entry. This follows immediately from the stencil structure stated in Figure 2.3 (b). However, in the case that a whole column k of our rectangular image domain Ω contains only mask values equal to 1, the 2 diagonal submatrices $A_{k,k-1}$ and $A_{k,k+1}$ along the off-diagonals are zero matrices. By adapting the size of the submatrices $A_{k-1,k-1}$ and $A_{k+1,k+1}$ it is possible to create a tridiagonal block structure such that none of the blocks contains 0 as only entry. The same strategy can also be applied to the block corresponding to the first or last column. The drawback is that

not all blocks have the same size anymore. However, this fact is completely irrelevant for our purposes. An example of such a restructuring is visualised in Figure 2.6. Let us remark, that the previous reasoning can also be done if the pixels are labelled row-wise. In that case, the inpainting matrix can be partitioned into $n_r \times n_r$ blocks, each of size $n_c \times n_c$.

Using Corollary 2.12.2 and Definition 2.10 it easily follows that $A(c)$ is block irreducible. Let us also remark that any labelling of the pixels in the image domain can be reduced to the previous case by applying a permutation of the labels. The underlying graph structure does not change and the block irreducibility is always preserved.

We now use the following result from [49, Theorem 3] and refer to the source for a proof of this important claim.

Proposition 2.13
Let $A \in \mathbb{R}^{n,n}$ be a block irreducible matrix and $\lambda \in \mathbb{C}$ an eigenvalue of A. If λ is a boundary point of the union of all the Geršgorin discs, then it is a boundary point of each Geršgorin disc.

Theorem 2.14 (Invertibility of the inpainting matrix)
Let $|N(i)| > \frac{h^2}{2}$ hold for all pixel indices i and define $\ell_{\max}$ as follows

$$\ell_{\max} \coloneqq \min_i \left\{ \frac{2|N(i)|}{2|N(i)| - h^2} \right\} .$$

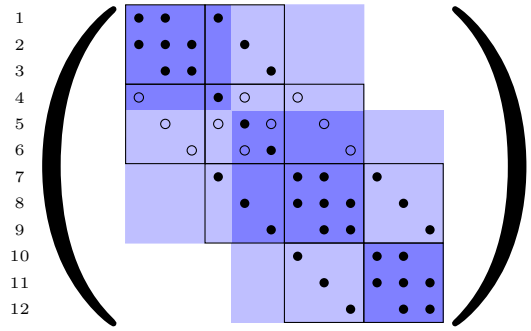
The inpainting matrix $A(c)$ is invertible if all mask entries are in the interval $[0, \ell_{\max}]$ and at least one mask entry is in the open interval $(0, \ell_{\max})$.

Proof. It follows from Proposition 2.5 that all mask entries must lie in the open interval $(0, \ell_{\max})$ to discard 0 as a potential eigenvalue. Further, if all c_i are in the interval $[0, \ell_{\max}]$, then 0 can only be a boundary point of the union of all Geršgorin discs. Using Proposition 2.13, it follows that 0 must be a boundary point of each Geršgorin disc if it is an eigenvalue. However, because of Proposition 2.5, this becomes impossible as soon as $0 < c_i < \ell_{\max}$ holds for at least one c_i. Thus, it is enough that a single mask value lies in the interval $(0, \ell_{\max})$ and all the others in the interval $[0, \ell_{\max}]$ to assert invertibility of the inpainting matrix. □

Note that the previous theorem indeed generalises the findings of Mainberger et al. [19]. They showed that the inpainting matrix with a binary

c_1	c_4	c_7	c_{10}
c_2	c_5	c_8	c_{11}
c_3	c_6	c_9	c_{12}

(a) A 3×4 image with the a column-wise ordering of the pixels. In this example, we assume that c_4, c_5, and c_6 are 0.



(b) The corresponding 12×12 inpainting matrix. Black discs indicate non-zero entries. Black circles indicate entries that are 0 due to the choice of the mask values. All other entries are 0. The labels of the image have been placed next to the matrix rows for better orientation.

Figure 2.6: Example for a block irreducible partitioning. We consider the mask given in (a) with c_4, c_5, and c_6 being 0 and obtain the matrix structure in (b). The black rectangles in (b) indicate the canonical tridiagonal block structure of the matrix obtained by column-wise labelling. The matrix is not block irreducible with respect to this ordering. The dark and clear blue blocks yield a tridiagonal block structure with respect to which the inpainting matrix is block irreducible.

mask is invertible if at least a single mask point is equal to 1. Our findings yield two significant extensions. First of all, we are not restricted to the binary setting anymore. Our mask points are free to take any value in the interval $[0, \ell_{\max}]$. Secondly, we have found a slightly better upper limit for the mask values. Indeed, it is easy to see that $\ell_{\max} > 1$ for any valid choice of $h > 0$ and $|N(i)|$.

2.3 Extremum principles

Desirable properties in the context of diffusion processes are the preservation of the mean value and the adherence to the max-min principle. Preservation of the mean value means that the average value of output signal should be identical to the average value of the input signal. Such a property is important for image processing purposes as it guarantees that images do not get darker or brighter. The max-min principle serves a similar purpose. It asserts that all signal values remain in the convex hull of the input data and it can be seen as a stability criteria for certain algorithms. While it is clear that an inpainting process will, in general, not preserve the mean value, we are still in the position to state constraints that guarantee that the extrema of the reconstruction do not exceed the extrema of the input data. The strategy is straightforward and rather simple. We show that our inpainting matrix is a so-called M-matrix. In combination with the fact that the inpainting matrix is invertible we can deduce that all the entries of $A(c)^{-1}$ are positive. Finally, we derive the max-min principle from the latter result. The same strategy was already presented for binary masks by Mainberger et al. [19]. A certain number of concepts are essential in the presentation of the just mentioned strategy. They are presented in the following and stem from [47, Section 8.1] and [50, Definitions 2.1.1, 2.1.2, 2.5.1, and 2.5.2].

Definition 2.15 (Non-negative matrix)
We say that a matrix $A \in \mathbb{R}^{n,n}$ is non-negative *if $a_{i,j} \geqslant 0$ for all i and j.*

Definition 2.16 (Inertia of a matrix)
Let $A \in \mathbb{C}^{n,n}$, we define $i_+(A)$ as the number of eigenvalues of A with positive real part, $i_-(A)$ as the number of eigenvalues of A with negative real part, and $i_0(A)$ as the the number of eigenvalues of A with zero real part. In each case multiplicities of the eigenvalues are taken into consideration. It

follows that $i_-(A) + i_0(A) + i_+(A) = n$. *Finally, the row vector*

$$i\,(A) \coloneqq \begin{pmatrix} i_+(A) & i_-(A) & i_0(A) \end{pmatrix}$$

is called the inertia *of the matrix* A.

Definition 2.17 (Positive stable matrix)
A matrix $A \in \mathbb{C}^{n,n}$ *is said to be* positive stable *if* $i_+(A) = n$.

The next definition is required as an intermediate step for the definition of a so called M-matrix.

Definition 2.18
The set $Z_n \subseteq \mathbb{R}^{n,n}$ *is defined by*

$$Z_n \coloneqq \{A \in \mathbb{R}^{n,n} \mid a_{i,j} \leqslant 0 \ \forall i \neq j\} \quad .$$

It contains all real valued matrices that have non-positive elements at each position not on the main-diagonal. The entries on the main-diagonal can be arbitrary.

Definition 2.19 (M-matrix)
A matrix $A \in \mathbb{R}^{n,n}$ *is called an* M-matrix *if* $A \in Z_n$ *and if* A *is positive stable.*

Thus, a M-matrix is a matrix which has only eigenvalues with positive real part and where all the entries not on the main diagonal are non-positive. M-matrices appear in many fields such as probability theory, economics and matrix analysis. They also appear in the context of discretisations of PDEs and in the analysis of finite Markov chains as well as in the study of population dynamics. Clearly, the inpainting matrix $A(c)$ is positive stable and a M-matrix if all mask entries are in the interval $[0, 1]$ and at least one mask entry is non-zero. This result follows immediately from the spectral analysis that we performed at the beginning of this section and the fact that the off-diagonals are given by $(-1 + c_i)\, h^{-2}$ which stays non-positive as long as $c_i \leqslant 1$ holds. Berman and Plemmons [51, Chapter 6, Theorem 2.3] list fifty conditions on a matrix $A \in Z_n$ which are equivalent for a matrix A to being a non-singular M-matrix. Especially their condition N_{38} is interesting for us. It states that A is inverse positive; that is, the inverse A^{-1} exists and all its entries are non-negative. This result allows us to show that signals obtained by applying the reconstruction matrix $M\,(c)$ do not exceed the range of the initial data. In [19] this result was shown for the case where the

inpainting mask is binary valued. The statement remains valid if all mask values stay in the range $[0, 1]$ and the actual proof is very similar. We state this result in the following proposition.

Proposition 2.20 (Max-min principle for homogeneous inpainting)
Let f be our signal to be reconstructed and c a given inpainting mask with $c_i \in [0, 1]$ for all mask positions i and at least one mask value in the interval $(0, 1]$. Further let $f_{\min} \coloneqq \min_i \{f_i \mid c_i > 0\}$ and $f_{\max} \coloneqq \max_i \{f_i \mid c_i > 0\}$ be the minimal and maximal known data value of our signal f. If $u = M(c)f$ is our reconstruction, then $u_i \in [f_{\min}, f_{\max}]$ for all i.

Proof. Note that the sum of all entries in the i-th row of the inpainting matrix $A(c)$ is c_i. From this it follows that

$$A(c) \begin{pmatrix} 1 \\ \vdots \\ 1 \end{pmatrix} = \begin{pmatrix} c_1 \\ \vdots \\ c_{n_r n_c} \end{pmatrix} \quad .$$

Due to our assumptions, the inpainting matrix $A(c)$ is invertible. Thus we can deduce

$$\sum_j \left(A(c)^{-1}\right)_{i,j} c_j = 1 \quad \forall i \ .$$

We know that $A(c)$ is inverse positive. In combination with the previous equation we conclude that $(A(c)^{-1})_{i,j} c_j \in [0, 1]$ for all i and j. Let us now consider the reconstruction $u = M(c) f$. By applying the definition of the reconstruction matrix, we obtain

$$u_i = \sum_j \left(A(c)^{-1}\right)_{i,j} c_j f_j \leqslant \underbrace{\sum_j \left(A(c)^{-1}\right)_{i,j} c_j}_{=1} f_{\max} \leqslant f_{\max} \ .$$

The estimate with respect to $f_{\min}$ is done in the very same way. □

2.4 Conclusion

The findings in this chapter show how to generalise the initial Laplace interpolation to a more general approach by blending the data values with their diffused counterparts. Our findings show that a reasonable range for

the mask values is the interval $[0, 1]$. It guarantees that the spectrum of the inpainting matrix stays real and that the reconstruction matrix exists, regardless of the grid size h. Furthermore, mask values within the unit interval assert adherence to the max-min principle for the reconstruction.

Chapter 3

Optimisation in the codomain

> Success consists of going
> from failure to failure
> without loss of enthusiasm.
>
> *(Winston Churchill)*

Mainberger et al. [20] describe an additional optimisation step to decrease the reconstruction error with homogeneous diffusion inpainting. They optimise the data values for a given, fixed, and non-empty binary mask. Experimental setups demonstrate that such an additional tuning of the interpolation data has a significant impact on the accuracy. Their so called grey value optimisation (GVO) method is analysed in the following paragraphs and several new findings concerning the optimality of grey values and the corresponding mask values are presented. We emphasise that all the results from this chapter assume that the mask locations have already been fixed beforehand. We only consider the corresponding mask values for these predefined positions. We further remark that the results from this chapter have also been presented in [52].

Mainberger et al. [20] discuss the GVO exclusively in combination with binary masks. In the previous chapter we have laid the foundation for inpainting strategies that can handle arbitrary mask values. Since the GVO by Mainberger et al. [20] is merely a post processing step, it can likewise be applied to non-binary masks, too. Thus, the important question is whether we can achieve even more accurate reconstructions with this additional step if the mask values have already been optimised. Another interesting topic is the numerical handling of the problem. Grey value optimisation can be expressed as a least squares problem. The question arises how efficiently this

problem can be solved.

The GVO task seeks for a given and fixed mask the data that yields the most accurate reconstruction in the least squares sense. This task can be expressed as

$$g = \underset{x \in \mathbb{R}^n}{\arg\min} \left\{ \frac{1}{2} \| M(c)\, x - f \|_2^2 \right\} \quad . \tag{3.1}$$

For convenience we define the reconstruction error as the value of the cost function from Eq. (3.1) for a given mask c and corresponding data x. The reconstruction error coincides with the popular mean squared error (MSE) up to a scaling factor. For our purposes the reconstruction error is the more natural choice, however.

Definition 3.1 (Reconstruction error)
We call reconstruction error*, the following quantity:*

$$E(c, x) \coloneqq \frac{1}{2} \| M(c)\, x - f \|_2^2 \quad . \tag{3.2}$$

During this whole chapter we assume that the reconstruction matrix $M(c)$ always exists. The necessary requirements have already been discussed in the previous chapter.

3.1 Optimal mask values and optimal data values

We follow the notational conventions from the previous chapters and assume that we have a total of $n \coloneqq n_r n_c$ linearly indexed pixels in our image and that we are given a fixed and non-empty set $K \subseteq \{1, \ldots, n\}$ of locations. They indicate the positions of our inpainting data x as well as the entries in our mask c which are non-zero. We will also refer to the set K as the sparsity pattern of our mask. For $i \in K$ we are left with three possibilities to improve the reconstruction. Either we fix the mask value c_i at some arbitrary value for all $i \in K$ and manipulate the corresponding pixel value x_i to reduce the error $E(c, x)$, or we fix x_i and optimise the value of c_i. Lastly, we could also try to optimise both x_i and c_i for all $i \in K$ simultaneously. In this chapter we are interested in the following two settings.

1. We fix $c_i = 0$ for all $i \notin K$, $c_i = 1$ for all $i \in K$ and solve Eq. (3.1) to obtain the optimal data g with respect to this binary mask.

2. We fix $c_i = 0$ for all $i \notin K$ and assume that all values c_i with $i \in K$ have been chosen such that $E(c, f)$ is minimal. Here, f represents the original input data.

The former setting corresponds to the GVO framework of Mainberger et al. [20] while the latter case optimises the mask values and leaves data at its original values. The question arises whether there is any benefit in the latter approach over the former one. After all, we are optimising data at the exact same positions in each case. In this chapter we show that it does not matter which strategy is employed. Both methods yield the same error.

The two just mentioned optimisation tasks are unconstrained. It follows that the necessary conditions for a minimum of E with respect to the data x (respectively the mask c) are given by

$$\begin{aligned} &\frac{\partial}{\partial x_i} E(c, x) = 0, \quad \forall i \in \{1, \dots, n\} \quad , \\ \text{resp.} \quad &\frac{\partial}{\partial c_i} E(c, x) = 0, \quad \forall i \in K \quad . \end{aligned} \tag{3.3}$$

For the upcoming discussion it is mandatory to have the analytic expressions of these derivatives to our avail. The partial derivatives with respect to the x_i are easy to derive and well known. Due to the complex dependency of E on c, the partial derivatives with respect to the mask entries are much more complicated to obtain.

The following proposition is an adapted version of a similar result found by Ochs et al. [53, Lemma 9]. There, the authors state it for the case $x = f$. Its reformulation to the more general setting is straightforward. We refer to the original work [53] for a detailed proof.

Proposition 3.2 (Gradients of the reconstruction error)
We consider the reconstruction error E from Eq. (3.2). The gradients of E with respect to the data x (denoted by ∇_x) and the mask c (denoted by ∇_c) are given by

$$\nabla_x E(c, x) = M(c)^\top (M(c)\, x - f) \quad , \tag{3.4}$$

$$\nabla_c E(c, x) = \operatorname{diag}\Big(x - (I + L)\, M(c)\, x\Big) A(c)^{-\top} (M(c)\, x - f) \quad . \tag{3.5}$$

A close look onto the previous expressions reveals an intriguing similarity between the gradients.

Corollary 3.2.1
The gradients of E with respect to x and c have a certain similarity. If we denote:

$$\begin{aligned} T(c,x) &:= A(c)^{-\top}\left(A(c)^{-1}\operatorname{diag}(c)\,x - f\right) \\ &= \left(\nabla_z\left(\frac{1}{2}\left\|A(c)^{-1}z - f\right\|_2^2\right)\right)\Big|_{z=\operatorname{diag}(c)x} \quad . \end{aligned}$$

Then we have

$$\begin{aligned} \nabla_x E(c,x) &= \operatorname{diag}(c)\,T(c,x) \quad , \\ \nabla_c E(c,x) &= \operatorname{diag}\left(x - (I+L)\,A(c)^{-1}\operatorname{diag}(c)\,x\right)T(c,x) \quad . \end{aligned}$$

Thus, both gradients coincide if

$$c = x - (I+L)\,A(c)^{-1}\operatorname{diag}(c)\,x \quad .$$

The next theorem is an important first step in showing the equivalence between mask value optimisation and GVO when the sparsity pattern is fixed. It shows that, regardless of what we optimise, all necessary optimality conditions for minimising the reconstruction error are fulfilled.

Theorem 3.3 (Fulfilment of optimality conditions)
Optimising the mask values and keeping the grey values fixed at the original data yields a pair of variables that fulfils all necessary optimality conditions from Eq. (3.3) *with respect to the data values x and the mask values c. Similarly, fixing a binary sparsity pattern for the inpainting mask and optimising the grey values also returns a pair of variables that fulfils all necessary optimality conditions for minimising the reconstruction error.*

Proof. Assume first that for a fixed non-empty sparsity pattern K we have found the optimal mask values $\tilde{c}$ for the reconstruction with respect to the original data f. This means that the entries $\tilde{c}_i$ with $i \in K$ have been obtained by seeking the minimiser of $E(c,f)$ with respect to all c_i with $i \in K$ and that $\tilde{c}_i = 0$ for all $i \notin K$. Further, it holds that

$$\left(\nabla_c E(c,f)\Big|_{c=\tilde{c}}\right)_{i\in K} = 0 \quad . \tag{3.6}$$

Replacing the gradient in Eq. (3.6) with the expression from Eq. (3.5) yields

$$\left(\operatorname{diag}\left(f-(I+L)\,M\left(\tilde{c}\right)f\right)A\left(\tilde{c}\right)^{-\top}\left(M\left(\tilde{c}\right)f-f\right)\right)_{i\in K}=0\ .$$

The previous equation is a product between the diagonal matrix

$$\operatorname{diag}\left(f-(I+L)\,M\left(\tilde{c}\right)f\right)$$

and the vector

$$A\left(\tilde{c}\right)^{-\top}\left(M\left(\tilde{c}\right)f-f\right)\ .$$

This product comes down to a component wise multiplication between the diagonal entries of the matrix and the vector entries. Therefore, at least one of the two following equations must hold for each $i \in K$:

$$\left(f-(I+L)\,M\left(\tilde{c}\right)f\right)_{i\in K}=0\ , \tag{3.7}$$

$$\left(A\left(\tilde{c}\right)^{-\top}\left(M\left(\tilde{c}\right)f-f\right)\right)_{i\in K}=0\ . \tag{3.8}$$

Our goal is to show that Eq. (3.8) actually holds for all $i \in K$. If for a certain entry $i \in K$, Eq. (3.7) differs from 0, then Eq. (3.8) must be 0. Thus we only need to show, that Eq. (3.7) can never hold. To this end note that $u \coloneqq M\left(\tilde{c}\right)f$ solves by definition the equation

$$\operatorname{diag}(\tilde{c})\left(u-f\right)+\left(I-\operatorname{diag}\left(\tilde{c}\right)\right)\left(-L\right)u=0 \tag{3.9}$$

and that Eq. (3.7) is equivalent to

$$\operatorname{diag}\left(\tilde{c}\right)\left(f-(I+L)\underbrace{M\left(\tilde{c}\right)f}_{=u}\right)=0\ . \tag{3.10}$$

From Eq. (3.9) it follows that

$$\operatorname{diag}\left(\tilde{c}\right)\left(u-f+Lu\right)=Lu\ . \tag{3.11}$$

Plugging Eq. (3.11) into Eq. (3.10) yields the requirement $-Lu = 0$. Thus, if Eq. (3.7) holds, then the reconstruction $u = M\left(\tilde{c}\right)f$ also needs to solve $-Lu = 0$. This contradicts our assumption that $\tilde{c}$ has at least one non-zero entry and that the set K is non-empty. Therefore, Eq. (3.7) can never hold and Eq. (3.8) is valid for all $i \in K$.

Next we observe that Eq. (3.8) can be extended to all indices i by multiplying it from the left with $\operatorname{diag}(\tilde{c})$. This gives us

$$\operatorname{diag}(\tilde{c})\, A(\tilde{c})^{-\top} \left(M(\tilde{c})\, f - f\right) = 0$$

which implies that

$$\nabla_x E(\tilde{c}, x)\Big|_{x=f} = 0 \ .$$

Thus, the necessary optimality conditions with respect to x are also fulfilled.

On the other hand, let us define a binary mask $\overline{c}$ with $c_i = 1$ if $i \in K$ and $c_i = 0$ else. Using this mask and optimising the grey values for the reconstruction yields the following requirement on the data x:

$$\begin{aligned} & \left(\nabla_x E(\overline{c}, x)\right)_i = 0 \quad \forall i \in \{1, \dots, n\} \\ \Leftrightarrow \quad & \left(\operatorname{diag}(\overline{c})\, A(\overline{c})^{-\top} \left(A(\overline{c})^{-1} \operatorname{diag}(\overline{c})\, x - f\right)\right)_i = 0 \quad \forall i \in \{1, \dots, n\} \\ \Leftrightarrow \quad & \left(A(\overline{c})^{-\top} \left(A(\overline{c})^{-1} \operatorname{diag}(\overline{c})\, x - f\right)\right)_{i \in K} = 0 \ . \end{aligned} \tag{3.12}$$

Let g be the optimal data that fulfils Eq. (3.12). Using Corollary 3.2.1, we see that $T(\overline{c}, g)_i = 0$ holds true for all $i \in K$. From this equality we can further conclude that

$$\left(\nabla_c E(c, g)\Big|_{c=\overline{c}}\right)_{i \in K} = 0 \ .$$

Thus, our pair $(\overline{c}, g)$ also fulfils the necessary optimality conditions with respect to the mask. □

We are now in the position to show the main result of this chapter. It proves that the optimisation of the mask values is equivalent to the optimisation of the grey values if the sparsity pattern of the mask is fixed.

Theorem 3.4 (Equivalence between mask and grey value optimisation)
Let, for a fixed and non-empty sparsity pattern K, the mask $\tilde{c}$ be given and assume that this mask minimises the reconstruction error when used in conjunction with the original data f. Then there exists inpainting data g, a binary mask $\overline{c}$ with the same sparsity pattern K as $\tilde{c}$ and we have

$$g = \operatorname*{arg\,min}_{x \in \mathbb{R}^n} \left\{E(\overline{c}, x)\right\} \ , \tag{3.13}$$

$$E(\tilde{c}, f) = E(\overline{c}, g) \ . \tag{3.14}$$

Proof. We have a pair $(\tilde{c}, f)$ to our avail which fulfils

$$\tilde{c}_i = \left(\underset{c_j,\, j \in K}{\arg\min} \{E(c, f)\} \right)_i \quad \forall i \in K \quad \text{and} \quad \tilde{c}_i = 0 \; \forall i \notin K \ .$$

One possible way to satisfy the claims of the theorem is to assume the availability of a vector g such that $M(\overline{c})\, g - f = M(\tilde{c})\, f - f$ holds. Expanding the right-hand side of this equality and exploiting the fact that $\overline{c}$ is binary valued, yields

$$g = \operatorname{diag}(\overline{c})\, A(\overline{c})\, M(\tilde{c})\, f \ . \tag{3.15}$$

We conclude that the existence of g is verified and that Eq. (3.14) holds. It only remains to show that our variable g from Eq. (3.15) is also a solution of the normal equations

$$M(\overline{c})^\top (M(\overline{c})\, g - f) = 0 \tag{3.16}$$

to assert optimality. Since $\overline{c}$ and $\tilde{c}$ have the same sparsity pattern it follows that $\ker \operatorname{diag}(\overline{c}) = \ker \operatorname{diag}(\tilde{c})$ and consequently $\ker M(\overline{c}) = \ker M(\tilde{c})$, too. Further, we remark that for any linear operator J from $\mathbb{R}^n$ to $\mathbb{R}^n$ we have $\ker(J^\top) = \operatorname{ran}(J)^\perp$. Combining these identities with the first isomorphism theorem [54, Theorem 6.23] yields

$$\begin{aligned} \ker M(\overline{c})^\top &= (\operatorname{ran} M(\overline{c}))^\perp \simeq (\mathbb{R}^n / \ker(M(\overline{c})))^\perp \\ &= (\mathbb{R}^n / \ker(M(\tilde{c})))^\perp \simeq (\operatorname{ran} M(\tilde{c}))^\perp = \ker M(\tilde{c})^\top \ . \end{aligned} \tag{3.17}$$

The importance of this identity will become clear in a moment. By assumption, $\tilde{c}$ is optimal. This implies that

$$\left(\nabla_c E(c, f) \Big|_{c=\tilde{c}} \right)_{i \in K} = 0 \ .$$

Because of Theorem 3.3 it follows that

$$\nabla_x E(\tilde{c}, x) \Big|_{x=f} = 0 \ .$$

Expanding the latter expression and using the identity $M(\overline{c})\, g = M(\tilde{c})\, f$ yields $M(\tilde{c})^\top (M(\overline{c})\, g - f) = 0$. At this point two possibilities exist. Either $M(\overline{c})\, g - f = 0$, in which case Eq. (3.16) holds trivially. In the other case it

follows that $0 \neq (M(\overline{c})\, g - f) \in \ker M(\tilde{c})^\top$. Equation (3.17) implies that $M(\overline{c})^\top (M(\overline{c})\, g - f) = 0$ holds as well. We conclude that Eq. (3.16) is again verified and that g always fulfils the GVO optimality conditions. □

Corollary 3.4.1
Assume that the optimal mask values $\tilde{c}$ for a fixed non-empty sparsity pattern K are known. Then the tonal optimisation can be obtained from Eq. (3.15)*.*

Our findings show that for a fixed sparsity pattern of the mask it is irrelevant whether we optimise mask values or data values. From a practical point of view there is however a significant difference. The optimisation of the grey values is a linear least squares problem and much easier to solve than finding the best mask values. Therefore, Eq. (3.15) is usually of very little use.

In the forthcoming paragraphs we discuss two methods that help us solve the tonal optimisation problem in an highly efficient manner. We remind that the GVO seeks a solution of

$$\underset{x \in \mathbb{R}^n}{\arg\min} \left\{ \frac{1}{2} \| M(c)\, x - f \|_2^2 \right\} \tag{3.18}$$

where the mask c is fixed and where $M(c)$ is the reconstruction matrix from Definition 2.2. Note that $M(c)$ is usually not invertible. One could find the optimal grey values by solving the normal equations. However, this approach is often unfeasible due to the high condition number of the reconstruction matrix and its prohibitive memory requirements for storing all the entries. One possible approach to handle this task has been proposed by Mainberger et al. [20]. They suggest to apply a Gauß-Seidel solver on the normal equation $M(c)^\top (M(c)\, x - f) = 0$. Unfortunately, their strategy is rather slow. Possible alternatives are presented in the following sections.

3.2 Fast algorithms for tonal optimisation

In this section we derive two algorithms for a fast and efficient handling of the GVO problem from Eq. (3.18). The algorithms provide state-of-the-art performance for implementations on the central processing unit (CPU) as well as for the graphics processing unit (GPU).

An approach based on the LSQR method

The least squares algorithm (LSQR) of Paige and Saunders [55, 56] is a highly efficient method to solve general least squares problems of the form

$$\arg\min_{x\in\mathbb{R}^n} \{\|Kx - b\|_2\}$$

with a large and possibly sparse and unsymmetric matrix $K \in \mathbb{R}^{n,n}$ and an arbitrary vector $b \in \mathbb{R}^n$. The underlying iterative strategy applies the bidiagonalisation process of Golub and Kahan [57] and decreases the norm of the residual in each step. Although the algorithm generates a sequence of iterates that has the same properties as those from standard conjugate gradient methods it tends to behave much better in numerically ill posed situations. Further, it is easy to implement and only requires the matrix K for computing matrix-vector products of the form Ku and $K^\top v$ for various vectors u and v. In presence of routines capable of computing these products efficiently it is not even necessary to know the matrix explicitly. This fact makes the algorithm attractive for solving Eq. (3.18). The adaptation is straightforward. It suffices to find a fast way to compute the products $M(c)\,x$ and $M(c)^\top x$. To this end we use the definition of the reconstruction matrix and rewrite the task as a linear system of equations:

$$\begin{aligned} y = M(c)\,x &\Leftrightarrow A(c)\,y = \operatorname{diag}(c)\,x \ , \\ y = M(c)^\top x &\Leftrightarrow A(c)^\top z = x,\ y = \operatorname{diag}(c)\,z \ . \end{aligned} \tag{3.19}$$

The linear systems $A(c)\,y = \operatorname{diag}(c)\,x$ and $A(c)^\top z = x$ can for example be solved in a highly efficacious manner with the multifrontal sparse LU decomposition from [58–60]. Since the mask c is fixed, the decomposition of the matrix $A(c)$ need only be done once during the first iteration of the LSQR algorithm. Forthcoming iterations can then be computed at almost no additional cost. Alternatively one can also use the multigrid solver from [19] or any other method to obtain a solution for the linear system. In our tests the sparse LU solver of Davis and Duff [58–60] performs best. The complete algorithm for solving the GVO problem is depicted in Algorithm 3.1

A primal dual formulation

An alternative strategy to the LSQR algorithm goes as follows: We start with Eq. (3.18) and rewrite the optimisation problem by introducing a dummy

Algorithm 3.1: Tonal optimisation with the LSQR algorithm: The computations of $M(c)\, v^{(k)}$ and $M(c)^{\top} u^{(k)}$ can be performed by solving the equations from Eq. (3.19) with the sparse LU decomposition.

Input: Reconstruction matrix $M(c)$, data f, number of iterations N
Output: Solution of the least squares problem Eq. (3.18): $x^{(N+1)}$
Initialise:

$$\bar{u}^{(1)} = f,\ \beta^{(1)} = \left\|\bar{u}^{(1)}\right\|,\ u^{(1)} = \frac{\bar{u}^{(1)}}{\beta^{(1)}}$$

$$\bar{v}^{(1)} = M(c)^{\top} u^{(1)},\ \alpha^{(1)} = \left\|\bar{v}^{(1)}\right\|,\ v^{(1)} = \frac{\bar{v}^{(1)}}{\alpha^{(1)}}$$

$$w^{(1)} = v^{(1)},\ x^{(0)} = 0,\ \bar{\phi}^{(1)} = \beta^{(1)},\ \bar{\rho}^{(1)} = \alpha^{(1)}$$

for k **from** *1* **to** N **do**

$$\bar{u}^{(k+1)} = M(c)\, v^{(k)} - \alpha^{(k)} u^{(k)}$$

$$\beta^{(k+1)} = \left\|\bar{u}^{(k+1)}\right\|,\ u^{(k+1)} = \frac{\bar{u}^{(k+1)}}{\beta^{(k+1)}}$$

$$\bar{v}^{(k+1)} = M(c)^{\top} u^{(k+1)} - \beta^{(k+1)} v^{(k)}$$

$$\alpha^{(k+1)} = \left\|\bar{v}^{(k+1)}\right\|,\ v^{(k+1)} = \frac{\bar{v}^{(k+1)}}{\alpha^{(k+1)}}$$

$$\rho^{(k)} = \sqrt{\left|\bar{\rho}^{(k)}\right|^2 + \left|\beta^{(k+1)}\right|^2}$$

$$c^{(k)} = \frac{\bar{\rho}^{(k)}}{\rho^{(k)}},\ s^{(k)} = \frac{\beta^{(k+1)}}{\rho^{(k)}},\ \theta^{(k+1)} = s^{(k)} \alpha^{(k+1)}$$

$$\bar{\rho}^{(k+1)} = -c^{(k)} \alpha^{(k+1)}$$

$$\phi^{(k)} = c^{(k)} \bar{\phi}^{(k)},\ \bar{\phi}^{(k+1)} = s^{(k)} \bar{\phi}^{(k)}$$

$$x^{(k+1)} = x^{(k)} + \frac{\phi^{(k)}}{\rho^{(k)}} w^{(k)}$$

$$w^{(k+1)} = v^{(k+1)} - \frac{\theta^{(k+1)}}{\rho^{(k)}} w^{(k)}$$

end

variable $d \in \mathbb{R}^n$ and enforce that it coincides with the reconstruction $M(c)\,x$.

$$\operatorname*{arg\,min}_{x \in \mathbb{R}^n} \left\{ \frac{1}{2} \| M(c)\,x - f \|_2^2 \right\} = \operatorname*{arg\,min}_{x,d \in \mathbb{R}^n} \left\{ \frac{1}{2} \| d - f \|_2^2 + \iota_{\{0\}} (d - M(c)\,x) \right\}$$

Here, ι_S is the characteristic function of the set S. It is defined as

$$\iota_S(x) := \begin{cases} 0, & x \in S \\ +\infty, & x \notin S \end{cases} .$$

Note that d is equal to $M(c)\,x$ if and only if $A(c)\,d = \operatorname{diag}(c)\,x$ holds. Thus, Eq. (3.18) is equivalent to

$$\operatorname*{arg\,min}_{x,d \in \mathbb{R}^n} \left\{ \frac{1}{2} \| d - f \|_2^2 + \iota_{\{0\}} (A(c)\,d - \operatorname{diag}(c)\,x) \right\} . \tag{3.20}$$

Equation (3.20) can be handled efficiently with the algorithm presented by Chambolle and Pock [61]. The advantage of this approach is that we have eliminated the inverse of the inpainting matrix $A(c)$ from our formulation. Applying the primal dual method from [61] only requires the evaluation of $A(c)\,u$ and $A(c)^\top u$ for a vector u. The matrix $A(c)$ is structured and extremely sparse. It follows that these products can be handled in an time-saving manner and lead to a high performing tonal optimisation strategy. A straightforward application of Algorithm 1 from [61] gives us the simple iterative strategy shown in Algorithm 3.2. This algorithm is better suited for parallel processing environments than Algorithm 3.1. Almost all operations are pointwise and do not depend on each other. Let us also remark that additional optimisations like preconditioning strategies, as presented by Pock and Chambolle [62], can further improve the performance of Algorithm 3.2.

Performance comparison

We analyse the performance of the stochastic tonal optimisation method from [20] (as presented in Algorithm 3.3), Algorithm 3.1 and Algorithm 3.2 in terms of speed. The results are depicted in Table 3.1. We use different sizes of the *Trui* test image (see Figure 5.3) for our benchmark. For each image size we compute a single inpainting mask by using the optimal control framework from Chapter 5. This mask is then binarised by thresholding its entries at 0.01 and used as inpainting mask for all the tonal optimisation

Algorithm 3.2: Tonal optimisation based on the primal dual method from [61]. Good estimates for the parameters τ and σ may be obtained by applying power iterations onto $\Big(A\,(c) \quad -\operatorname{diag}(c)\Big)$.

Input: N the number of iterations.
Output: Vectors $x^{(N+1)}$ and $d^{(N+1)}$ solving Eq. (3.20)
Initialise:
$\theta \in [0,1]$ arbitrary and τ, $\sigma > 0$ such that

$$\left\|\Big(A\,(c) \quad -\operatorname{diag}(c)\Big)\right\|_2^2 < \frac{1}{\sigma\tau}$$

$x^{(0)}$, $c^{(0)}$ and $y^{(0)}$ arbitrary
$\overline{x}^{(0)} = x^{(0)}$ and $\overline{d}^{(0)} = d^{(0)}$

for k **from** *1* **to** N **do**

$$\begin{aligned}
y^{(k+1)} &= y^{(k)} + \sigma\left(A\,(c)\,\overline{d}^{(k)} - \operatorname{diag}(c)\,\overline{x}^{(k)}\right)\\
d^{(k+1)} &= \frac{d^{(k)} - \tau\left(A\,(c)^{\top} y^{(k+1)} - f\right)}{1+\tau}\\
x^{(k+1)} &= x^{(k)} + \tau\operatorname{diag}(c)\,y^{(k+1)}\\
\overline{d}^{(k+1)} &= d^{(k+1)} + \theta\left(d^{(k+1)} - d^{(k)}\right)\\
\overline{x}^{(k+1)} &= x^{(k+1)} + \theta\left(x^{(k+1)} - x^{(k)}\right)
\end{aligned}$$

end

Table 3.1: Speed comparison between the different algorithms for tonal optimisation on the CPU and GPU: All times are given in seconds and represent the average of three runs. Algorithm 3.3 by Mainberger et al. [20] performs worst on every image size and its run time increases much faster for larger images than for the other two algorithms. For small images, the transfer of the data to the GPU and back requires a significant amount of time. Therefore, there is little difference in the run times for the very small images on the GPU. We refer to Figure 5.3 for a visualisation of the reconstruction quality. (GPU results courtesy of Sebastian Hoffmann)

Image size	Run time CPU			Run time GPU
$n_r \times n_c$	Alg. 3.3	Alg. 3.1	Alg. 3.2	Alg. 3.2
48×48	32.57	1.23	2.90	1.36
64×64	156.33	2.69	5.82	1.28
80×80	360.42	4.63	8.50	1.47
96×96	783.87	7.72	14.89	2.30
112×112	1633.82	12.02	35.86	2.60
128×128	3116.70	18.73	52.57	3.33
256×256	95 832.64	113.07	260.26	9.01

methods. All masks have a density within the range of $5.0 \pm 0.1\,\%$. We use the algorithm from [20] as a reference method and compare how well our algorithms compete in terms of speed. For each approach, the parameters are tuned such that all algorithms converge towards the same solution. The method from [20] uses a powerful multigrid solver to compute the inpaintings $M(c)e_i$ in Algorithm 3.3. It stops when the error between two iterates drops below 10^{-3}. Algorithm 3.1 stops when the increment in the solution drops in norm below 10^{-10} whereas Algorithm 3.2 halts its execution when the update in any variable is smaller than 10^{-15} in norm. These tolerances assert that all algorithms reach the same reconstruction error within a tolerance of 10^{-6}. The algorithms have been implemented in Fortran 2003 and ANSI C. All tests are done on a standard desktop PC with an Intel Xeon processor clocked at 3.2 GHz and 24 GB of memory. We also use a Nvidia GeForce GTX 460 for the GPU experiments. The represented timings in Table 3.1 are the averages of three runs for each test case.

The exceptional performance of the LSQR based algorithm stems from

the fact that it reaches a convergent state within 10 to 30 iterations. As a consequence it requires less than 100 inpaintings. On the other hand, the method of Mainberger et al. [20] has to compute an inpainting for every mask pixel during each iteration. The method can be sped up by precomputing the inpainting results of all the individual image impulses e_i in Algorithm 3.3. However, this strategy requires excessive amounts of memory. For an image of size 256×256 with a mask density of 5 % it is necessary to store more than 3250 reconstructions. This corresponds roughly to 1.6 GB of data if the necessary results are stored with double precision. We conclude that Algorithm 3.1 is best suited for CPU implementations. For GPUs, the method of choice is Algorithm 3.2. There, most computations can be done in parallel and no linear systems must be solved. Neither the method of Mainberger et al. [20] nor the LSQR-based approach from Algorithm 3.1 are capable of exploiting the massive parallelism of a GPU efficiently enough to be competitive to the primal dual strategy.

Algorithm 3.3: Tonal optimisation of Mainberger et al. [20]. The image e_i has a pixel value of 1 in pixel i and 0 in any other pixel.

Input: Image f, Stopping threshold $\varepsilon > 0$
Output: Optimised tonal values g
Initialise: $u = M(c)f$ and $g = f$
repeat
  Set $u_{\text{old}} = u$
  for all *pixel positions* $i \in \{1, \dots, n\}$ **do**

$$g = g + \frac{\langle M(c)e_i, f - u\rangle}{\|M(c)e_i\|_2^2} e_i$$

$$u = u + \frac{\langle M(c)e_i, f - u\rangle}{\|M(c)e_i\|_2^2} M(c)e_i$$

  end
until $|\|u - f\| - \|u_{old} - f\|| \leqslant \varepsilon$
return g

3.3 Conclusion

In this chapter we have analysed the benefits of a tonal optimisation. Our findings show that it does not matter whether we optimise the mask values or the grey values if the sparsity pattern is fixed. This result is important within two contexts. The forthcoming chapters show that it is easier to find a good sparsity pattern with a rough approximation to the optimal mask values when those are allowed to take arbitrary values. Once the sparsity pattern is found we can binarise the mask and perform the tonal optimisation. The outcome is identical to performing a tedious mask value optimisation. Furthermore, we have highly efficient algorithms for the GVO problem to our avail that are hard to outperform. Secondly, our findings allow a tremendous reduction of the storage size in the context of image compression. Instead of storing data positions, data values, and optimal mask values, we only need to store the data positions and the respective optimised data values. The binarisation of the mask comes at no loss. This insight marks a cornerstone in the design of a competitive image compression codec.

Chapter 4

Optimisation in the domain

> Simplicity is the ultimate sophistication.
>
> *(Leonardo da Vinci)*

So far we have discussed the reconstruction process for given data and a fixed mask. Further, we have shown how to optimise the interpolation data for a fixed sparsity pattern of the mask. However, we have not discussed any approaches that yield good data locations yet. The next two chapters catch up on this topic. In this chapter we first present a very simple method that works only in the 1D setting for strictly convex functions. This restrictive framework is rewarded with a very simple algorithm and some interesting insight into the difficulties of localising good mask positions. A more generic approach is discussed in Chapter 5. We remind that our inpainting method with known binary valued masks corresponds to piecewise linear spline interpolation in the 1D setting. Thus, we do have an analytic expression for the solutions of the PDE to our avail. Further, only considering the recovery of convex functions allows us to state the reconstruction error in a convenient form which lends itself to an effective optimisation scheme. Unfortunately, this simple setting already reveals the difficulties related to mask optimisation. Finding optimal binary masks is a non-convex problem. This fact prevents us from exploiting many efficient and well studied strategies.

Interpolation is also closely linked to approximation. The main difference is that we do not require exact reconstructions at the mask positions in the approximation framework. From our point of view one could interpret a combined optimisation of the mask positions and the respective mask/function values as an approximation problem. This observation moti-

vates further investigations in this direction and parallels to GVO and mask value optimisation are drawn as well.

The results from this chapter are inspired by the findings on free knot optimisation from spline interpolation theory. We refer to the works [27–30] for a general overview on this topic. Results closely related to our presentation can also be found in the works of Hamideh [63] and Kioustelidis and Spyropoulos [64].

4.1 Optimal masks for linear spline interpolation

Let us now formalise the setting which we consider in the forthcoming paragraphs. It is closely related to the notational conventions we have used before. We suppose that our domain $\Omega \subseteq \mathbb{R}$ is a closed and bounded interval of the form $[a, b]$ with $-\infty < a < b < \infty$ and that $f\colon \mathbb{R} \to \mathbb{R}$ is a strictly convex function on Ω. Further, we consider a mask set $\Omega_K \subseteq \Omega$ with $|\Omega_K| = N + 1$ distinct positions k_i distributed over the whole domain Ω:

$$\Omega_K \coloneqq \{k_i \mid k_0 = a,\ k_N = b,\ k_{i-1} < k_i < k_{i+1},\ i = 1, \dots, N-1\} \quad . \tag{4.1}$$

These mask positions k_i denote the locations where the function f is interpolated. Our goal consists in finding a set Ω_K such that the interpolation error between f and a piecewise linear spline u becomes minimal in the L_1 norm. In concrete terms, we seek a piecewise linear and continuous function u defined on Ω which has the form

$$u(x) \coloneqq \begin{cases} \frac{f(k_{i+1}) - f(k_i)}{k_{i+1} - k_i}(x - k_i) + f(k_i), & x \in (k_i, k_{i+1}) \quad , \\ f(k_i), & x = k_i \quad . \end{cases}$$

This linear spline u should have minimal distance to f in the L_1 sense. Note that u is completely determined by specifying the interpolation locations Ω_K. Thus, we can consider the error E in function of the mask

$$E(\Omega_K) \coloneqq \|u - f\|_{L_1(\Omega)} = \int_\Omega |u(x) - f(x)| \, \mathrm{d}x \quad . \tag{4.2}$$

We remind, that Ω_K represents the location of the Dirichlet boundary data in Eq. (2.1), whereas the piecewise linear spline u represents the solution of the corresponding PDE. Finding the best Ω_K inside this context is known

in the literature as free knot (FK) problem and the mask positions are often referred to as knots. In order to remain consistent with the nomenclature from the previous chapters we continue to call them mask points. The FK problem has been studied for more than fifty years already but only few satisfactory solutions exist. We refer to Hamideh [63] and Kioustelidis and Spyropoulos [64] for similar considerations as in our work and to Jupp [29], Boor [65], DeVore and Popov [66], and Dikoussar and Török [67] and the references therein for more general approaches. Further ways to optimise linear spline interpolation can also be found in the works of Blu et al. [68].

The mask points k_0 and k_N are fixed in Eq. (4.1) at the boundary of the considered interval for technical reasons which we elucidate in a moment. Also, the choice of the L_1 norm is especially attractive in this case: Due to the strict convexity of f the difference $u - f$ is non-negative for all x in Ω and thus we can simply omit the absolute value in Eq. (4.2). If we had not fixed k_0 and k_N at the interval boundaries, then the previous estimate would not necessarily hold. The fact that we can omit the absolute value is the foundation upon which the forthcoming optimisation strategy is built on. The non-negativity of our integrand $u - f$ allows us to specify simple necessary optimality conditions for the mask. Plugging the analytic expression of the linear spline into our energy and evaluating it yields

$$\begin{aligned} E(\Omega_K) &= \int_\Omega u(x) - f(x)\,\mathrm{d}x \\ &= \frac{1}{2}\sum_{i=0}^{N-1} (k_{i+1} - k_i)\,(f(k_{i+1}) + f(k_i)) - \int_\Omega f(x)\,\mathrm{d}x \ . \end{aligned} \tag{4.3}$$

We remark that the expression obtained in Eq. (4.3) also corresponds to the error of the composite trapezoidal rule for the numerical integration of f with the non-equidistant integration intervals $[k_i, k_{i+1}]$. In that sense our problem is equivalent to finding the best data set for numerical quadrature. We refer to [69] for more information on numerical integration. An example visualisation of the FK problem is given in Figure 4.1.

The next result shows the difficulty of the FK optimisation task.

Proposition 4.1
If the function $f\colon \mathbb{R} \to \mathbb{R}$ is strictly convex on Ω and twice continuously differentiable in the interior of Ω, then the energy functional given in Eq. (4.2) *is convex in Ω_K for three mask points and in general not convex for any*

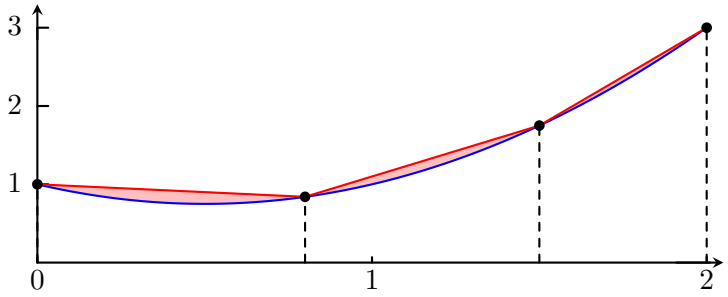


Figure 4.1: Visualisation of the free knot problem: We seek those positions of the the black dots along the blue curve such that the surface between the red and the blue curve becomes minimal (surface marked in pale red). Here, the blue curve represents the function $x^2 - x + 1$ and the red curve is the linear spline interpolating the function at the positions marked by the black dots.

other number of mask points larger than three.

Proof. In the case of three mask points we only have one free variable, namely k_1, and it follows from Eq. (4.3) that the error is given by

$$\frac{1}{2}\Big((k_1 - a)\left(f(k_1) + f(a)\right) + (b - k_1)\left(f(b) + f(k_1)\right)\Big) - \int_a^b f(x)\,\mathrm{d}x \quad .$$

Further, the second derivative of E with respect to k_1 is given by

$$\frac{\partial^2}{\partial k_1^2} E(\Omega_K) = \frac{b-a}{2} \frac{\partial^2}{\partial k_1^2} f(k_1)$$

and obviously positive for all valid $k_1 \in \Omega$. Thus E is a convex function in k_1. In order to demonstrate that the error function can be non-convex for a higher number of interpolation points it suffices to provide a counterexample. Let us consider the function $f(x) = \exp(x)$ on the interval $\Omega = [-15, 15]$ as well as the two masks

$$\Omega_K^1 := \{-15, 10.65, 14.65, 15\} \quad ,$$
$$\Omega_K^2 := \{-15, -1.2, 12.5, 15\} \quad .$$

If E were convex, then it must also be convex along the line in $\mathbb{R}^4$ that connects Ω_K^1 and Ω_K^2 (interpreting both sets as vectors in $\mathbb{R}^4$). However, the plot of $E((1-\lambda)\Omega_K^1 + \lambda\Omega_K^2)$ with $\lambda \in [0, 1]$ depicted in Figure 4.2 clearly displays a non-convex behaviour. □

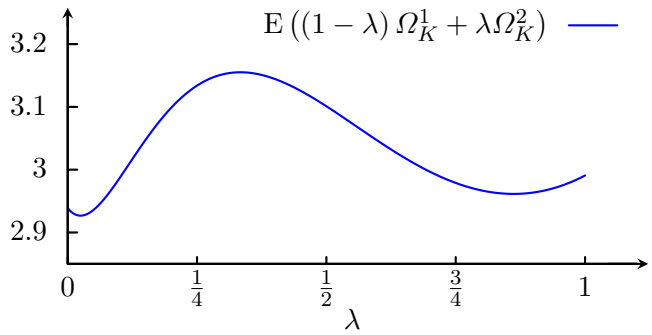


Figure 4.2: Evolution of the interpolation error with a linear spline for the function $\exp(x)$ with mask points along the segment with endpoints $\Omega_K^1 = \{-15, 10.65, 14.65, 15\}$ and $\Omega_K^2 = \{-15, -1.2, 12.5, 15\}$. It clearly depicts a non-convex behaviour. Note that the function values have been rescaled by a factor 10^{-6} for better readability.

We remark that the previous proposition does not claim that the energy can never be convex for more than three mask points. Indeed for affine functions of the form $\alpha x + \beta$ with real coefficients α and β the energy is identically zero for any number of data locations and thus also convex. This shows that even under weaker conditions as in the proposition, the energy may be convex.

Let us now derive a new algorithm for finding optimal mask points. A necessary condition for a minimiser Ω_K^* of Eq. (4.3) is $\nabla E\,(\Omega_K^*) = 0$. Here $\nabla E\,(\Omega_K^*)$ denotes the gradient of the error E with respect to the individual mask points k_i with $i \in \{1, \ldots, N-1\}$ and evaluated at the knot sites in Ω_K^*. The points k_0 and k_N are not considered since they are fixed anyway. Note that it follows from Proposition 4.1 that this condition is not sufficient. There may very well exist several global and/or local minima. A simple computation of $\nabla E\,(\Omega_K^*) = 0$ leads us immediately to the following system of $N-1$ nonlinear equations in the $N-1$ unknowns k_i, with $i \in \{1, \ldots, N-1\}$.

$$f'\,(k_i) = \frac{f\,(k_{i+1}) - f\,(k_{i-1})}{k_{i+1} - k_{i-1}}, \quad i = 1, \ldots, N-1 \ . \tag{4.4}$$

The next observations follow immediately from Eq. (4.4). Each optimal mask point only depends on its direct neighbours. Therefore, even indexed points only depend on odd indexed ones and vice versa. Since f is strictly convex on Ω, it follows that its derivative f' is strictly monotonically increasing

on Ω. Thus, its inverse $(f')^{-1}$ exists and is unique at every point of the considered domain. This motivates us to alternatingly update all the even indices and then all the odd indices by solving Eq. (4.4) for k_i with an even and an odd index i respectively. By iterating this two-step update strategy we hope to reach a fixed-point. A detailed description of the algorithm is given in Algorithm 4.1. Note that the initialisation can be arbitrary. In practice a uniform distribution has proven to work quite well.

Algorithm 4.1: Mask optimisation in the 1D domain $\Omega = [a, b]$ for strictly convex functions $f\colon \Omega \to \mathbb{R}$ with $N+1$ mask points

Input: The desired number of mask points: $N+1$.
Output: Optimal mask Ω_K^*.
Initialise: Choose arbitrary initial distribution $\Omega_K^{(0)}$ with $k_0^{(0)} = a$, $k_N^{(0)} = b$ and $k_{i-1}^{(0)} < k_i^{(0)} < k_{i+1}^{(0)}$ for all $i \in \{1, \dots, N-1\}$
repeat
  for all *even indices i in* $\{1, \dots, N-1\}$ **do**

$$k_i^{(j+1)} := (f')^{-1}\left(\frac{f\left(k_{i+1}^{(j)}\right) - f\left(k_{i-1}^{(j)}\right)}{k_{i+1}^{(j)} - k_{i-1}^{(j)}}\right) \tag{4.5}$$

  end
  for all *odd indices i in* $\{1, \dots, N-1\}$ **do**

$$k_i^{(j+1)} := (f')^{-1}\left(\frac{f\left(k_{i+1}^{(j+1)}\right) - f\left(k_{i-1}^{(j+1)}\right)}{k_{i+1}^{(j+1)} - k_{i-1}^{(j+1)}}\right) \tag{4.6}$$

  end
until *fixed-point is reached*
return *Mask Ω_K^*.*

Observe that the strategy in Algorithm 4.1 is similar to a Red-Black Gauß-Seidel scheme for solving linear systems (see for example the book of Saad [70] for a more detailed presentation). We update the variables iteratively and use newly gained information as soon as it becomes available without interfering with the direct neighbours of the data point.

An important issue is that the mask points k_i are not allowed to fall together in our approach. General spline theory allows such a situation.

Boor [30, Chapter IX] shows that the multiplicity of a mask point is linked to the smoothness of the corresponding spline at that position. In our case two overlapping mask points would imply that our interpolating spline function would exhibit discontinuities. A characteristic that we wish to avoid. Furthermore, our goal is to obtain the smallest possible error for a given amount of mask points. Intuitively it seems to be clear that the best solutions must necessarily be attained with the largest possible quantity of data. The number of available mask points corresponds to the quota of information that we want to exploit. The more knowledge we have to our avail, the better we can reconstruct the function. Thus, there is little motivation in allowing the number of mask points to be smaller than specified. Kioustelidis and Spyropoulos [64, Theorem 3] also show that these requirements must necessarily be fulfilled in an approximation framework. The following proposition shows that mask points preserve the order in which they are initialised by Algorithm 4.1. Thus, they can never fall together.

Proposition 4.2
The iterative scheme proposed Algorithm 4.1 preserves the ordering of the mask point positions. This means we have

$$k_{i-1}^{(j)} < k_i^{(j)} < k_{i+1}^{(j)} \quad \Rightarrow \quad k_{i-1}^{(j+1)} < k_i^{(j+1)} < k_{i+1}^{(j+1)}$$

for all $i \in \{1, \ldots, N-1\}$ *and all* $j \geqslant 0$.

Proof. Since f is differentiable on Ω the mean value theorem guarantees the existence of a k_i in (k_{i-1}, k_{i+1}) such that

$$f'(k_i) = \frac{f(k_{i+1}) - f(k_{i-1})}{k_{i+1} - k_{i-1}} \ .$$

Thus, our iterative scheme must necessarily preserve the order of the mask points. □

Up to this point we have presented a simple iterative strategy to find good mask points and we have shown that the algorithm yields the expected number of data points. We do not know yet if our approach converges. Also, even if it were to converge, we cannot claim yet that the obtained solution is optimal. These issues are considered in the following statements.

The next theorem shows that the iterates from our algorithm monotonically decrease the considered energy.

Theorem 4.3
If the function $f : \mathbb{R} \to \mathbb{R}$ is strictly convex on Ω and twice continuously differentiable in the interior of Ω, then the sequence of iterates $(\Omega_K^{(k)})_k$ obtained in Algorithm 4.1 decreases the L_1 error from Eq. (4.2) in each step. This means we have $E(\Omega_K^{(k+1)}) \leqslant E(\Omega_K^{(k)})$ for all k.

Proof. The decrease in the error is essentially due to two facts. By alternating between the update of the odd and even indexed sites the problem decouples. The new value $k_i^{(j+1)}$ will only depend on $k_{i-1}^{(j)}$ and $k_{i+1}^{(j)}$, which are fixed. Therefore, the problem is localised and we can update all the even/odd indexed mask points independently of each other. It follows that one iteration step is equivalent to finding the optimal $k_i^{(j+1)}$ such that the interpolation error becomes minimal on $[k_{i-1}^{(j)}, k_{i+1}^{(j)}]$ for all even/odd i. The global error can now be written as the sum of all the errors over the intervals $[k_{i-1}^{(j)}, k_{i+1}^{(j)}]$ and will necessarily decrease when each term of this sum decreases. Further, Proposition 4.1 shows that the considered energy is convex for three mask points. Thus,

$$\left(\frac{\partial}{\partial k} E\left(\left\{k_{i-1}^{(j)},\ k,\ k_{i+1}^{(j)}\right\}\right)\right)\Big|_{k=k_i^{(j+1)}} = 0$$

is not only a necessary, but also a sufficient condition on $k_i^{(j+1)}$ for minimising the error on the interval $[k_{i-1}^{(j)}, k_{i+1}^{(j)}]$. This means that Eq. (4.5) will not increase the error when updating even indexed mask points and subsequently, Eq. (4.6) will not increase the error while updating the odd numbered sites. Therefore, it follows that the overall error cannot increase in an iteration step. □

From the previous theorem we can conclude, that the errors of all our iterates lie in the interval $[0, E(\Omega_K^{(0)})]$. Thus, the sequence of errors is bounded and monotonically decreasing. It follows that $(E(\Omega_K^{(k)}))_k$ is converging for k running to infinity. Note that we cannot claim convergence of the sequence of mask points $(\Omega_K^{(k)})_k$ itself. Since the problem is non-convex, the global minimum of the considered energy is not assured to be unique. Our algorithm might alternate between several of the minimisers. These minimisers are, from a qualitative point of view, all identical since they yield the same (minimal) error which must not necessarily be the global minimum. However,

note that the theorem of Bolzano-Weierstrass asserts that $(\Omega_K^{(k)})_k$ contains at least one convergent subsequence since all mask points are always required to lie in the compact domain Ω.

4.2 Optimal masks for linear spline approximation

So far we have only discussed the case of linear spline interpolation. This framework offers us already an optimisation strategy for the mask positions. Of course, Algorithm 4.1 can also be combined with the findings from Chapter 3 as a subsequent post processing step to obtain a corresponding optimisation of the data values. The question arises, if a sequential optimisation of the mask positions followed by a tonal optimisation can be competitive to a combined optimisation of mask positions and mask values (respectively, mask positions and data values). As already mentioned in the beginning of this chapter, a combined optimisation can be treated like a spline approximation model. The fact that we are operating in a rather restrictive framework of reconstructing real valued strictly convex functions allows us to gain essential insight into these optimisation problems. Compared to the interpolation approach, the optimal approximation with linear splines has received significantly more attention in the literature and many results are well known. Let us briefly describe the approximation setting. As in the previous section we assume that some bounded and closed interval $\Omega \subset \mathbb{R}$ is given. The corresponding mask Ω_K of cardinality $N+1$ is identically defined as in the previous section. Further, we have a strictly convex function $f\colon \mathbb{R} \to \mathbb{R}$ to our avail and consider a piecewise linear spline $u_{\text{app}} : \Omega \to \mathbb{R}$ of the form

$$u_{\text{app}}(x) \coloneqq \begin{cases} \alpha_i (x - k_{i-1}) + \beta_i, & x \in [k_{i-1}, k_i)\,,\ i \in \{1, \dots, N-1\} \\ \alpha_N (x - k_{N-1}) + \beta_N, & x \in [k_{N-1}, k_N] \quad . \end{cases}$$

Similarly as before we fix the first mask point k_0 at the beginning of the domain and the last one, k_N, at the end of the domain. However, we do not require anymore that $u_{\text{app}}(k_i) = f(k_i)$ for any mask point k_i. Nevertheless, the mask points $k_i \in \Omega_K$ with $i \in \{1, \dots, N-1\}$ still represent the locations where the individual linear parts of the spline u_{app} blend into each other. The task consists now in finding those parameters α_i, β_i and Ω_K such that $\|u_{\text{app}} - f\|_{L_1(\Omega)}$ becomes minimal.

As already mentioned, the topic of optimal approximations seems to have received more attention in the past than optimal interpolation. Stone [27] analyses the best approximation of strictly convex functions in the least squares sense while Davis [71] cites general conditions for determining best approximations of strictly convex functions in the L_∞ sense. Theoretical results can also be found in the work of Jupp [29]. Nürnberger and Braess [72] show that an optimal approximation of convex functions with splines does not necessarily have a unique solution, a problem which we already mentioned in the stricter case of convex spline interpolation. Finally, Cox [73] and Phillips [74] supply algorithms for determining optimal approximations. A textbook covering in detail the topic of function approximations has also been written by Rice [75].

In this section we focus on two works that yield similar results as our findings from Section 4.1. Kioustelidis and Spyropoulos [64] do a thorough theoretical analysis concerning optimality of linear spline approximations. Hamideh [63] presents an algorithmic approach to find optimal masks. Essential for the results discovered by Kioustelidis and Spyropoulos and Hamideh is the following observation which can be found in the book written by Rice [75]. The proof of this finding can be verified by direct computation but a more elegant way can also be found in [75, Chapter 4-4].

Proposition 4.4 (Optimal line approximation for convex functions.)
For any function $f : \mathbb{R} \to \mathbb{R}$ which is strictly convex on its bounded domain $\Omega = [a, b]$, the optimal straight line approximation to f in the L_1 sense on Ω interpolates f at the points

$$\xi_1 = \frac{3a + b}{4} \quad \textit{and} \quad \xi_2 = \frac{a + 3b}{4} \ .$$

The previous proposition shows that finding the optimal line, that means when $\Omega_K = \{a, b\}$ and $\Omega = [a, b]$, is trivial. It remains however a daunting challenge to optimise the mask set if we are asked to place more than two data points and require the corresponding spline to be continuous. Clearly, the difficulty lies in enforcing the continuity while preserving the optimality. An immediate idea based the previous proposition would be to build an optimal spline by constructing locally optimal lines on each interval $[k_i, k_{i+1}]$ and considering the corresponding linear spline function u_{app}. Note that in general such a resulting linear spline will not be continuous. Figure 4.3 depicts an example of an approximation where the mask positions have been

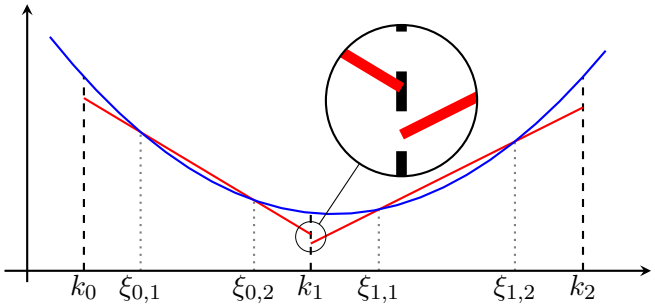


Figure 4.3: Approximation of $f(x) = \left(x - \frac{3}{2}\right)^2 + \frac{1}{2}$ (in blue) on the interval $\left[\frac{4}{10}, \frac{13}{5}\right]$ with a piecewise linear spline (in red). On each interval $[k_i, k_{i+1}]$ the line has been computed with Proposition 4.4. The corresponding optimal interpolation points given by this proposition are indicated as $\xi_{i,1}$ and $\xi_{i,2}$. Note that the corresponding spline function has a discontinuity in the knot k_1. Also, the function values of the spline do not coincide with the function values of f at the positions k_i.

fixed randomly and the lines on each subinterval $[k_i, k_{i+1}]$ have been set according to Proposition 4.4. As we can see, the obtained piecewise linear spline has a discontinuity in k_1.

Kioustelidis and Spyropoulos [64] consider the approximation of strictly convex functions and provide a link between the best mask Ω_K and the continuity of the corresponding linear spline built with locally optimal lines [64, Theorem 2]. They show that an optimal Ω_K can be obtained by solving the tridiagonal system of nonlinear equations

$$0 = \frac{1}{2}\left(f\left(\frac{3k_{i-1} + k_i}{4}\right) - 3f\left(\frac{k_{i-1} + 3k_i}{4}\right) + 3f\left(\frac{3k_i + k_{i+1}}{4}\right) - f\left(\frac{k_i + 3k_{i+1}}{4}\right)\right) \quad (4.7)$$

for all $i = 1, \ldots, N-1$. Further, the obtained piecewise linear function consisting of the corresponding locally optimal lines will be continuous. Due to this relationship, Eq. (4.7) is also referred to as continuity condition.

Hamideh [63] bases his research upon the results from Kioustelidis and Spyropoulos and presents a simple algorithm to determine the position and value of the optimal mask for strictly convex functions. His method optimises both quantities simultaneously and the resulting reconstruction is an

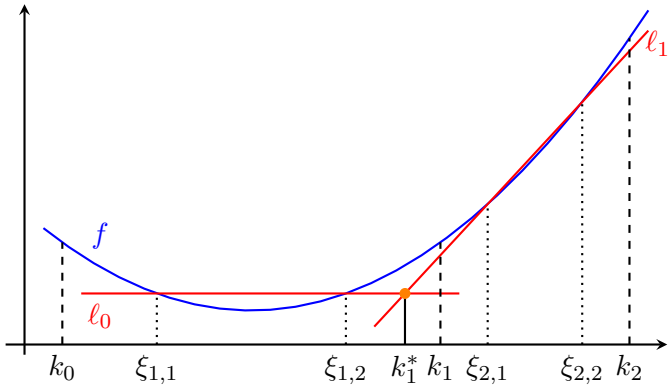


Figure 4.4: Visualisation of Hamideh's algorithm for updating the location of k_1. The function f (marked in blue) represents our data function. The points k_0, k_1 and k_2 denote the locations of the mask points from the previous iteration. On each interval $[k_{i-1}, k_i]$ we compute the optimal interpolation points $\xi_{i,1}$ and $\xi_{i,2}$ using Proposition 4.4. These points yield two lines ℓ_0 and ℓ_1 (marked in red). The intersection point of these two lines (marked in orange) represents the next location of the mask point k_1^*. This strategy is repeated until a fixed-point is reached.

approximating spline to the given input function. The underlying strategy is similar to ours from Algorithm 4.1. By exploiting the result from Proposition 4.4 for a given mask, Hamideh's method constructs locally optimal lines. The intersection points of these lines yield a new mask. This step is simply repeated until a fixed-point is reached. The approach is visualised in Figure 4.4 and a detailed presentation is given in Algorithm 4.2. The benefits of Hamideh's algorithm over a straightforward solving strategy for the continuity condition lie in the simplicity of the method and the existence of a convergence theory. Furthermore, Hamideh's method is constructive and allows a simple geometric interpretation. We also note that Chieppa [76] showed that Hamideh's algorithm can be interpreted as a Jacobi method for solving the continuity condition from Eq. (4.7) for each mask point.

4.3 Numerical experiments

Our main interest lies in the reconstruction quality of a sequential application of the mask tuning followed by a tonal optimisation compared against the combined mask position and value optimisation done by the algorithm of

Algorithm 4.2: Mask optimisation of Hamideh [63] for strictly convex functions $f\colon \Omega \to \mathbb{R}$ with $\Omega = [a, b]$ and $N + 1$ mask points

Input: $N + 1$ the number of desired mask points.
Output: Optimal mask Ω_K^*
Initialise: Choose any initial distribution $\Omega_K^{(0)}$ with $k_0^{(0)} = a$ and $k_N^{(0)} = b$ and $k_{i-1}^{(0)} < k_i^{(0)} < k_{i+1}^{(0)}$ for all $i \in \{1, \dots, N-1\}$
repeat
  for all *subintervals* $\left[k_{i-1}^{(j)}, k_i^{(j)}\right]$ **do**
    Define locally optimal points $\xi_{i,1}^{(j)}$ and $\xi_{i,2}^{(j)}$

$$\xi_{i,1}^{(j)} := \frac{3k_{i-1}^{(j)} + k_i^{(j)}}{4}, \qquad \xi_{i,2}^{(j)} := \frac{k_{i-1}^{(j)} + 3k_i^{(j)}}{4}$$

    Define the line $\ell_{i-1}^{(j)}$ passing through $\xi_{i,1}^{(j)}$ and $\xi_{i,2}^{(j)}$

$$\ell_{i-1}^{(j)}(x) := \frac{f\left(\xi_{i,2}^{(j)}\right) - f\left(\xi_{i,2}^{(j)}\right)}{\xi_{i,2}^{(j)} - \xi_{i,1}^{(j)}} \left(x - \xi_{i,1}^{(j)}\right) + f\left(\xi_{i,1}^{(j)}\right)$$

    for all *indices* i **do**
      Determine the new mask point position $k_i^{(j+1)}$ by intersecting the lines $\ell_{i-1}^{(j)}$ and $\ell_i^{(j)}$, i.e. solve

$$\ell_{i-1}^{(j)}\left(k_i^{(j+1)}\right) = \ell_i^{(j)}\left(k_i^{(j+1)}\right)$$

      for $k_i^{(j+1)}$.
    end
  end
until *fix point is reached*
return *Optimal mask* Ω_K^*

Hamideh. We expect that the approach of Hamideh yields the best results but also believe that our sequential strategy can get relatively close in terms of accuracy. To this end we also analyse the potential impact of a tonal optimisation onto the obtained masks. Since we operate in a L_1 setting in this chapter we will perform the tonal optimisation with respect to this norm, too. We note that Chapter 3 discusses the squared euclidean norm and thus differs from the approach employed in this chapter. Due to the difficulties of minimising the L_1 norm in the continuous setting, we approximate this optimisation task through a full discretisation. We sample our domain at $M \gg N$ uniformly distributed positions x_i and denote the values taken by our spline u and those of the data function f at these locations by u_i, respectively f_i. Next, we exploit the fact that any piecewise linear function can be expressed as a linear combination of B-splines $B_{1,j}$ of degree one:

$$u = \sum_{j=0}^{N} \gamma_j B_{1,j} \ .$$

The functions $B_{1,j}$ with corresponding knot set Ω_K are easily obtained by the Cox-de Boor recursion formula [30, Chapter IX, B-Spline Property (i)]. We shortly remark that the first and last mask point in Ω_K must be used with knot multiplicity 2 to obtain the correct number of basis functions for a complete representation.

The previous identity implies that $u_i = \sum_{j=0}^{N} \gamma_j B_{1,j}(x_i)$ for all i. By packing all the values $B_{1,j}(x_i)$ into a matrix $B \in \mathbb{R}^{M,N+1}$ and all γ_j and f_i into vectors $\gamma \in \mathbb{R}^{N+1}$ (resp. $f \in \mathbb{R}^M$), we obtain the following expression for the tonal optimisation

$$\arg\min_{\gamma \in \mathbb{R}^{N+1}} \{\|B\gamma - f\|_1\} \ .$$

By introducing an additional variable $z \in \mathbb{R}^M$ we can rewrite the previous problem as a linear program

$$\min_{\gamma,z} \{z\}$$
$$\text{such that} \begin{cases} B\gamma - f \preccurlyeq z \\ -B\gamma + f \preccurlyeq z \\ z \succcurlyeq 0 \end{cases}$$

Table 4.1: Error measures for our interpolation algorithm and the approximation algorithm of Hamideh for different numbers of mask points applied to the function $x \mapsto \exp(2x - 3) + x$ on the interval $[-4, 4]$. The corresponding masks of size 7 are visualised in Figure 4.5. For our method we list the error without additional tonal optimisation and with additional tonal optimisation. For Hamideh's method we list the errors of the resulting mask from his algorithm and with the extra tonal optimisation. As expected, Hamideh's method performs best in each case but there are no visible improvements for the tonal optimisation in his algorithm. However, applying a tonal optimisation boosts the results from the interpolation framework and yields competitive error measures.

$\lvert\Omega_K\rvert$	Our method		Hamideh	
	without optim.	with optim.	Initial	with optim.
5	12.501	4.229	3.982	3.982
7	5.134	1.810	1.748	1.748
9	2.785	0.999	0.977	0.977

where $\preccurlyeq$ and $\succcurlyeq$ denote element wise inequalities. Linear programs belong to the best studied optimisation tasks and many highly efficient solvers exist. Their study can be traced back to 1939 and has been initiated by Kantorovich [77]. We refer to the work of Luenberger and Ye [78] for more information on how to solve these problems. Due to its ease of use and efficiency we opt for the popular simplex algorithm [78, Chapter 3] and fix $M = 2^{16}$ for all our experiments. Even for such a large number of samples the simplex algorithm converged in each case within a few seconds.

Let us now apply all our presented strategies onto the convex function $f(x) = \exp(2x - 3) + x$ on the interval $[-4, 4]$. The obtained errors are specified in Table 4.1. The experiments are done with a randomised initial distribution of the mask and 5000 iterations. For each setup the iterates converge already after very few iterations and the upper limit of 5000 iterations is more than sufficient. The distance between the two last iterates is always below the machine precision of 10^{-16}. The final distribution for a mask of size 7 is visualised for both algorithms in Figure 4.5. In accordance with the theory from the previous section we note that the error is monotonically decreasing, both with respect to the number of mask points and with the number of iterations. Another interesting observation is the influence of an additionally introduced point onto the whole mask. Adding further mask

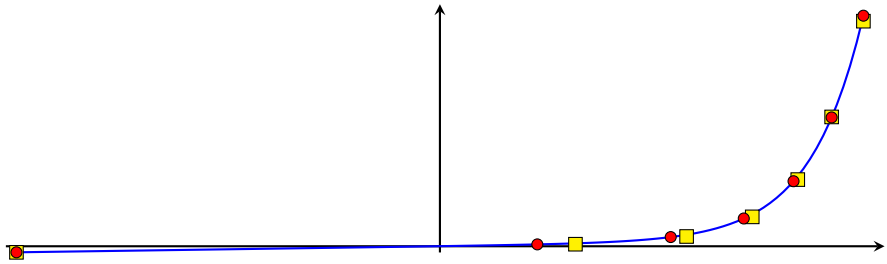


Figure 4.5: Distribution of 7 mask points along the interval $[-4, 4]$ for our interpolation approach (red circles) and the method of Hamideh (yellow squares) for the function $x \mapsto \exp(2x - 3) + x$ (marked in blue). Note that there are almost no points in flat regions, whereas there is a high density in regions with large curvature. The mask points of Hamideh do not necessarily lie on the function curve. Their positions corresponds to the coordinates yielded by Hamideh's algorithm (well visible for the last mask point at the right).

points has a global influence. Each element from Ω_K converges towards a different location after the insertion of an additional mask point. Also, we encounter a higher density in regions with large curvature than in flat regions. A behaviour which seems reasonable and could have been predicted. Although the mask distribution for the algorithm of Hamideh is similar to the one found with the interpolation framework, the error is significantly lower. The additional tuning of the data values allows a higher accuracy in the reconstruction. Hamideh's algorithm outperforms our method in each case if we do not apply a tonal optimisation as post processing. Finally, we also examine the benefits of the tonal optimisation. The accuracy gain is tremendous for our method but it does not yield any visible improvements for the method of Hamideh. With the additional tonal optimisation our interpolation masks are competitive with the results of Hamideh even though we still cannot beat them. Experiments with other strictly convex functions yield identical results.

4.4 Conclusion

Finding the optimal mask positions is a complicated task. It can be formulated as a non-convex optimisation problem. Two algorithmic approaches have been analysed that perform well in a very restrictive setting. The

algorithm of Hamideh [63] can be interpreted as a combined tuning of tonal and spatial values. It outperforms our pure positional optimisation, even when the tonal optimisation is applied as a post processing. However, the differences become marginal. These results inspire us to pursue the optimisation of the reconstruction data and to combine our results with the findings from Chapter 3. Finally, the excellent quality of all the masks and their reconstructions motivates further investigations in their optimisation and the extension to arbitrary image data sets such that we can recover these from as little data as possible. Unfortunately, the methods presented in this chapter are neither extendable to non-convex functions nor to two dimensional environments in a straightforward way. The convexity of the underlying data is an essential requirement for the derivation of the formulas presented in this chapter. Next, for higher dimensions we do not have any explicit representations for the reconstruction or the corresponding error to our avail. Thus, we cannot simply compute gradients and set them to zero to get a system of equations whose solutions are our sought optimal mask points. Heuristics, such as applying our algorithms along each dimension iteratively, also prohibit any convergence claims. It follows that we need to investigate on alternative models to obtain good interpolation data.

Chapter 5

Optimisation in the domain and codomain

> The difficulty lies not so much in developing new ideas as in escaping from old ones.
>
> *(John Maynard Keynes)*

Chapter 4 contains a novel method for finding optimal mask positions for strictly convex functions. The comparison against Hamideh's method also demonstrates the benefits of an additional tuning in the codomain. We have already discussed some surrogates to perform such a tuning as a post processing step in Chapter 3. In order to develop a competitive image compression codec we need to overcome the previously discovered shortcomings and find an alternative approach to those from Chapter 4. In the forthcoming paragraphs we develop a novel formulation that does not suffer anymore from the severe restrictions imposed on our data in the foregoing chapters. The resulting method is applicable to any function in arbitrary dimensions. Further, our algorithm yields solutions which are optimised both in the domain and codomain. It follows that the findings from this chapter pave the way for a new perspective on partial differential equation based image compression.

The underlying model that we present is inspired by optimal control theory and has initially been presented in [79]. We use the inpainting PDE from Eq. (2.3) as a starting point and relax the restrictions placed upon the confidence function c. In this chapter it is free to take any value in $\mathbb{R}$ and it is considered to be a parameter that allows us to steer the inpainting.

Further, Eq. (2.3) is combined with a strictly convex energy to penalise poor reconstructions and non-sparse sets of interpolation data. Our complete framework consists of a large-scale optimisation task with a strictly convex but non-differentiable objective and non-convex constraints. We proceed as follows for its detailed presentation: The concrete formulation of the model is derived in Section 5.1. In Section 5.2 we discuss a strategy to handle the occurring difficulties in this optimisation problem. Several hurdles need to be overcome. We have to take care of the non-differentiability of the energy and the non-convexity of the constraints. The underlying idea of our solver is to replace the original problem by a series of simpler convex optimisation tasks that can be solved efficiently. Section 5.3 provides an alternative strategy by exploiting results on convex conjugacy. These also offer an additional understanding of the properties of the underlying task and allow us to state simple optimality conditions. Next, Section 5.4 offers some further insight into our framework by expressing supplementary optimality conditions and requirements for a monotonic convergence towards a solution. Finally, in Section 5.5 we describe experiments that show the general usefulness of our model, both in the 1D and 2D setting. Extensions to colour images and video sequences are also briefly demonstrated.

5.1 A novel optimal control model for good interpolation data

We shortly remind a few facts from Chapter 2. If the confidence function c from Eq. (2.2) maps to $\{0, 1\}$ for all points within the image domain Ω, then our inpainting PDEs from Eq. (2.1) and Eq. (2.3) represent equivalent formulations of the same mixed boundary value problem. As already mentioned, the latter equation makes also sense if c is allowed to take a continuous range of values such as $\mathbb{R}$. One may regard continuously valued functions c as a relaxation of the initial formulation. We are no longer in the presence of a combinatorial optimisation task once the PDE has been discretised. Our goal is to optimise such $\mathbb{R}$-valued masks with respect to the accuracy of the reconstruction and to the sparsity of the interpolation data. Note that these two objectives cannot be perfectly fulfilled at the same time. If $c\,(x) \equiv 1$, then the reconstruction obtained by solving Eq. (2.3) is perfect. On the other hand, the sparsest possible choice would be $c\,(x) \equiv 0$ which does not allow

any reconstruction at all. Therefore, we suggest to complement Eq. (2.3) by an energy that reflects exactly this trade-off between the quality of the reconstruction and the amount of interpolation data. This leads us to the following constrained optimisation problem:

$$\begin{aligned} &\arg\min_{u,c} \left\{ \int_\Omega \frac{1}{2} \left(u\left(x\right) - f\left(x\right)\right)^2 + \lambda |c\left(x\right)| + \frac{\varepsilon}{2} c\left(x\right)^2 \mathrm{d}x \right\} \\ &c\left(x\right)\left(u\left(x\right) - f\left(x\right)\right) - \left(1 - c\left(x\right)\right) \Delta u\left(x\right) = 0, \quad \text{on } \Omega \\ &\qquad\qquad\qquad\qquad\qquad\qquad \partial_n u = 0, \quad \text{on } \partial\Omega \setminus \partial\Omega_K \end{aligned} \tag{5.1}$$

with positive parameters λ and ε. The first term in the energy penalises deviations of the reconstruction from the original data f. As in many other imaging applications, such as image segmentation [80], we encourage a sparse mask by also penalising the L_1 norm of c. The choice of λ lets us steer the sparsity of the mask. For $\lambda = 0$, the optimal solution is $c(x) \equiv 1$. If λ increases, the mask will become sparser. On the other hand, letting λ run towards infinity will require $c(x)$ to be 0 almost everywhere. Finally, we add an additional quadratic penalisation on c with a positive weight ε to the energy for technical reasons. As we will see in the forthcoming section, our numerical solver will require us to solve intermediate problems with a linear instead of non-convex constraint. These problems are related to optimal control problems of the form

$$\begin{aligned} &\arg\min_{u,c} \left\{ \int_\Omega \frac{1}{2} \left(u\left(x\right) - h_1\left(x\right)\right)^2 + \lambda \left|c\left(x\right)\right| + \frac{\varepsilon}{2} c\left(x\right)^2 \mathrm{d}x \right\} \\ &\qquad Du = h_2 + c \end{aligned} \tag{5.2}$$

with a second-order elliptic and linear differential operator D, a state u, a control variable c, and given data h_1 and h_2. Existence and regularity of such formulations is analysed by Clason and Kunisch [81], Stadler [82], and Wachsmuth and Wachsmuth [83]. The problem in Eq. (5.2) may not necessarily have a solution c if $\varepsilon = 0$. Clason and Kunisch [81] show that one may be forced to resort to measures to assert solvability in such a setting. In order to avoid these ill-posed formulations it is however sufficient to fix ε at a small positive value. A convergence analysis when ε decreases towards 0 is presented by Wachsmuth and Wachsmuth [83]. Although an analytic discussion of the variational model in Eq. (5.1) is out of the scope of this work, we remark that we include the penaliser on the squared L_2 norm of

c for the same regularity reasons. Furthermore, we will see in one of the upcoming sections that a positive value for ε has another advantage. It helps us in the derivation of a so called dual formulation.

5.2 A solution strategy

Our optimal control model proposed in Eq. (5.1) is challenging for two reasons. First of all, the energy contains a non-differentiable term and secondly, the occurring mixed products $c(x)u(x)$ and $c(x)\Delta u(x)$ in the constraint render the set of tuples (u, c) that fulfil the PDE non-convex. In order to devise a solution strategy we opt for a discretise-first-then-optimise approach. The PDE is discretised as described in Chapter 2. In addition to the notation introduced in that chapter we further denote the total number of samples by n. This allows us to formulate our optimisation problems in the same way for any dimension of the underlying data set. Concerning the energy, we simply transform the L_p norms in their discrete analogues. Thus, the discrete version of Eq. (5.1) is given by

$$\begin{aligned} &\underset{u,c\in\mathbb{R}^n}{\arg\min} \left\{ \frac{1}{2}\|u-f\|_2^2 + \lambda\|c\|_1 + \frac{\varepsilon}{2}\|c\|_2^2 \right\} \\ &\operatorname{diag}(c)\,(u-f) + (I - \operatorname{diag}(c))\,(-L)\,u = 0 \ . \end{aligned} \tag{5.3}$$

In order to tackle Eq. (5.3) numerically, we will replace it by a series of simpler convex optimisation problems. This idea is related to several well-known methods from the literature. One of the simplest strategies is known as sequential linear programming (SLP) and has been discovered by Griffith and Stewart [84]. Sequential linear programming methods replace a single non-linear optimisation problem by a sequence of linear programs. These linear programs are obtained through a first-order Taylor approximation of the objective and the constraints. This method sounds appealing because it significantly reduces the complexity of the problem. However, it has a major drawback. In order to achieve an accurate result, the solution must necessarily lie at a vertex of the linearised constraint space. This requirement is usually not fulfilled. As an alternative, one may consider linearly constrained Lagrangian methods (LCL). They have originally been presented by Friedlander and Saunders [85] and Murthagh and Saunders [86] and differ from SLP formulations by the fact that they do not linearise

the objective function. They only consider a linear approximation of the constraints and try to minimise the (augmented) Lagrangian of the original problem. LCL methods are popular and quite effective. Robinson [87] shows that under suitable conditions one can achieve quadratic convergence rates with them.

The main difference between these methods and ours will be the treatment of the objective function. We keep the original energy and merely augment it by an additional penalty term. This way we can circumvent the need to differentiate the objective and provide an alternative approach to LCL methods that often require the involved data to be differentiable. A similar strategy to ours is also briefly mentioned by Tröltzsch [88, Section 2.16] as a possibility to derive optimality conditions for non-linear optimal control problems. Our approach also presents certain similarities to majorise/minimise methods (MM). These methods perform step wise approximations to the original objective with convex majorisations or minorisations. MM strategies go back to Orthega and Rheinboldt [89] and have since then reappeared regularly under various names. Finally we remark that matrix factorisation and completion problems have a similar non-convex structure as the problem discussed in this section. Hence, alternative methods as ours have recently been proposed by Lin [90], Xu and Yin [91], and Xu et al. [92].

As already mentioned, our goal is to replace the problem in Eq. (5.3) by a series of convex problems that are easier to solve. Therefore, we will replace the constraints by linear counterparts that approximate the original conditions. We define a mapping T which evaluates the constraints for given vectors u and c.

$$\begin{aligned} T : \mathbb{R}^n \times \mathbb{R}^n &\to \mathbb{R}^n \\ (u, c) &\mapsto \operatorname{diag}(c)(u - f) + (I - \operatorname{diag}(c))(-L)\, u \ . \end{aligned} \tag{5.4}$$

Its first-order approximation around some point $(\bar{u}, \bar{c})$ can be written as

$$T(u, c) \approx T(\bar{u}, \bar{c}) + D_u T(\bar{u}, \bar{c})(u - \bar{u}) + D_c T(\bar{u}, \bar{c})(c - \bar{c})$$

where $D_u T(\bar{u}, \bar{c})$ and $D_c T(\bar{u}, \bar{c})$ describe the Jacobi matrices for the differentiation with respect to u and c at position $(\bar{u}, \bar{c})$. It is easy to check that

$$\begin{aligned} D_u T(\bar{u}, \bar{c}) &= \operatorname{diag}(\bar{c}) + (I - \operatorname{diag}(\bar{c}))(-L) \stackrel{!}{=} A(\bar{c}) \ , \\ D_c T(\bar{u}, \bar{c}) &= \operatorname{diag}(\bar{u} - f + L\bar{u}) \ . \end{aligned} \tag{5.5}$$

Note that $D_u T(\bar{u}, \bar{c})$ is actually the inpainting matrix $A(\bar{c})$ from Definition 2.1 for the mask $\bar{c}$. It follows that our initial problem is approximated by

$$\begin{aligned} &\operatorname*{arg\,min}_{u,c \in \mathbb{R}^n} \left\{ \frac{1}{2}\|u - f\|_2^2 + \lambda \|c\|_1 + \frac{\varepsilon}{2}\|c\|_2^2 \right\} \\ &D_u T(\bar{u}, \bar{c})(u - \bar{u}) + D_c T(\bar{u}, \bar{c})(c - \bar{c}) = -T(\bar{u}, \bar{c}) \quad . \end{aligned}$$

However, the previous formulation is only reliable for pairs (u, c) in a neighbourhood of $(\bar{u}, \bar{c})$. Therefore, we additionally penalise large differences from $(\bar{u}, \bar{c})$ by adding a proximal term with positive weight μ to oppose strong deviations from the linearisation point:

$$\begin{aligned} &\operatorname*{arg\,min}_{u,c \in \mathbb{R}^n} \left\{ \frac{1}{2}\|u - f\|_2^2 + \lambda \|c\|_1 + \frac{\varepsilon}{2}\|c\|_2^2 + \frac{\mu}{2}\left\| \begin{pmatrix} u \\ c \end{pmatrix} - \begin{pmatrix} \bar{u} \\ \bar{c} \end{pmatrix} \right\|_2^2 \right\} \\ &D_u T(\bar{u}, \bar{c})(u - \bar{u}) + D_c T(\bar{u}, \bar{c})(c - \bar{c}) = -T(\bar{u}, \bar{c}) \quad . \end{aligned} \tag{5.6}$$

Our goal is to iterate Eq. (5.6). We solve the previous problem for some given $u^{(k)}$ and $c^{(k)}$ to obtain a pair $(u^{(k+1)}, c^{(k+1)})$ which we use as new linearisation point. This iteration step is repeated until convergence. Thus, we compute

$$\begin{aligned} \left(u^{(k+1)}, c^{(k+1)}\right) = \operatorname*{arg\,min}_{u,c \in \mathbb{R}^n} \Bigg\{ &\frac{1}{2}\|u - f\|_2^2 + \lambda \|c\|_1 + \\ &\frac{\varepsilon}{2}\|c\|_2^2 + \frac{\mu}{2}\left\| \begin{pmatrix} u \\ c \end{pmatrix} - \begin{pmatrix} u^{(k)} \\ c^{(k)} \end{pmatrix} \right\|_2^2 \Bigg\} \\ D_u T\left(u^{(k)}, c^{(k)}\right)\left(u - u^{(k)}\right) + D_c T\left(u^{(k)}, c^{(k)}\right)\left(c - c^{(k)}\right) = -T\left(u^{(k)}, c^{(k)}\right) \quad . \end{aligned} \tag{5.7}$$

for all k until a fixed-point is reached. For the sake of brevity, we now introduce the following aliases:

$$\begin{aligned} A &\coloneqq D_u T\left(u^{(k)}, c^{(k)}\right) \quad , & B &\coloneqq D_c T\left(u^{(k)}, c^{(k)}\right) \quad , \\ g &\coloneqq Au^{(k)} + Bc^{(k)} - T\left(u^{(k)}, c^{(k)}\right) \quad , & & \\ \zeta_1 &\coloneqq 1 + \mu \quad , & \zeta_2 &\coloneqq \varepsilon + \mu \quad , \\ z_1 &\coloneqq \frac{\mu u^{(k)} + f}{1 + \mu} \quad , & z_2 &\coloneqq \frac{\mu c^{(k)}}{\varepsilon + \mu} \quad . \end{aligned} \tag{5.8}$$

They help us in rewriting our optimisation task from the previous equation in a more compact form. A straightforward computation also reveals that $g = \operatorname{diag}(c^{(k)})(I + L)u^{(k)}$. It suffices to insert the expressions for the Jacobians $D_u T$ and $D_c T$ and for T. All in all, we are led to the final form of our discrete approximation of Eq. (5.1):

$$\begin{aligned} &\underset{u,c\in\mathbb{R}^n}{\arg\min} \left\{ \frac{\zeta_1}{2}\|u - z_1\|_2^2 + \lambda\|c\|_1 + \frac{\zeta_2}{2}\|c - z_2\|_2^2 \right\} \\ &\qquad\qquad Au + Bc = g \end{aligned} \tag{5.9}$$

where all the quadratic terms from Eq. (5.7) have been regrouped into a single term in u and c respectively. This reformulation is achieved by applying quadratic completion and adding or removing constant terms that do not alter the minimiser but only shift the minimum. Before we discuss the overall behaviour of the iterates, we first have to analyse the linearised problem from Eq. (5.9) outside of our iterative strategy from Eq. (5.7). Existence and uniqueness of solutions are important topics that need to be considered. There is no point in investigating the existence of fixed-points of Eq. (5.7) if these linearised tasks from Eq. (5.9) cannot be solved properly. Once we have ensured that our iterates are well posed, we can analyse the convergence properties of our strategy as a whole.

Note that the set of feasible points in Eq. (5.9) may be empty. Even though we have twice as many variables as equations in our constraints it may happen that the equations are contradicting each other. Clearly we can also rewrite our problem in an unconstrained form

$$\underset{u,c\in\mathbb{R}^n}{\arg\min} \left\{ \frac{\zeta_1}{2}\|u - z_1\|_2^2 + \lambda\|c\|_1 + \frac{\zeta_2}{2}\|c - z_2\|_2^2 + \iota_{\{g\}}(Au + Bc) \right\} . \tag{5.10}$$

Here $\iota_{\{g\}}$ is the characteristic function of the set $\{g\}$. This latter form will be useful in the derivation of a numerical scheme for finding optimal vectors u and c.

Let us also shortly discuss the existence of minimisers of Eq. (5.10) if there exist solutions of the linear systems $Au+Bc = g$. If we assume that the linear system is solvable, then Eq. (5.10) represents a minimisation problem with a proper, lower semi-continuous, strictly convex and coercive cost function. These properties assert the existence of a unique solution ([93, Theorem 2.6] and [94, Satz 2.13]). We remind that a function ψ is coercive if $\psi(x)$

diverges towards $+\infty$ whenever $|x|$ runs towards $+\infty$. A function ψ is lower semi-continuous in x_0 if

$$\psi(x_0) \leqslant \liminf_{x \to x_0} \psi(x) \quad .$$

Finally, a convex function is said to be proper if it is finite in at least one point and if it does not take the value $-\infty$.

In the case where the matrix A is even invertible, we can express u as $A^{-1}(g - Bc)$. Clearly, it follows that the linear system has infinitely many solutions, namely one for each choice of c, if B is not the zero matrix and a single unique solution, namely $A^{-1}g$, else. In each of these cases the solution of Eq. (5.9) will be unique as we have to solve

$$\arg\min_{c \in \mathbb{R}^n} \left\{ \frac{\zeta_1}{2} \left\| A^{-1}Bc - \left(A^{-1}g - z_1 \right) \right\|_2^2 + \lambda \|c\|_1 + \frac{\zeta_2}{2} \|c - z_2\|_2^2 \right\} \quad .$$

The latter task is clearly an optimisation problem with a strictly convex, continuous, and coercive cost function. This is enough to guarantee uniqueness. We point to the findings from Section 2.2 for conditions that assert the existence of A^{-1}.

We emphasise further that the invertibility of the inpainting matrix A is a useful feature but not absolutely necessary for our approach. None of our forthcoming numerical strategies requires the explicit existence of A^{-1}. By not limiting the mask values in any way we can exploit the full potential of a complete mask optimisation and achieve the best possible results. Nevertheless, we will implicitly assume for any of the forthcoming discussions that the linear system $Au + Bc = g$ has at least one solution. This slight restriction asserts that our iterates always exist. Let us also remark that the existence of A^{-1} may still be required to guarantee convergence. A fact that we cannot discard yet. All we know at this point is that the existence of the inverse of A is not necessary to carry out the iterations.

In view of the forthcoming results we also mention the related problem of finding the minimal value of the previous energy and embed this task via a perturbation w into a family of optimisation problems. This gives us

$$(P_w) \qquad \begin{aligned} \min_{u,c \in \mathbb{R}^n} & \left\{ \frac{\zeta_1}{2} \|u - z_1\|_2^2 + \lambda \|c\|_1 + \frac{\zeta_2}{2} \|c - z_2\|_2^2 \right\} \\ & Au + Bc + w = g \quad . \end{aligned}$$

The introduction of this perturbation $w \in \mathbb{R}^n$ is inspired by the results presented in the book of Bonnans and Shapiro [93, Section 2.5] and allows an elegant analysis of the behaviour of convex optimisation problems. Note that for $w = 0$ we get our original problem back. We call (P_w) the primal problem.

Let us now return to the linearised approximation of our original optimal control model. Equation (5.9) is a constrained optimisation problem with a continuous, strictly convex, and coercive cost and linear constraints. Such problems are well studied and many highly efficient algorithms exist from which we can freely chose. For our purpose it does not matter how Eq. (5.9) is solved. We use a primal-dual algorithm for convex problems from Chambolle and Pock [61] and Esser et al. [95] where it is referred to as Algorithm 1 and modified primal dual hybrid gradient (PDHGMu), respectively. This algorithm represents an excellent trade-off between simplicity and efficiency. For convex functions $F\colon \mathbb{R}^\ell \to \mathbb{R}$, $G\colon \mathbb{R}^k \to \mathbb{R}$, and a linear and continuous operator $K : \mathbb{R}^k \to \mathbb{R}^\ell$ this algorithm solves

$$\min_{x \in \mathbb{R}^k} \{F(Kx) + G(x)\} \quad . \tag{5.11}$$

It is a well known fact from convex analysis (see [93, Section 2.5.2]) that this formulation is equivalent to the saddle point problem

$$\min_{x \in \mathbb{R}^k} \max_{y \in \mathbb{R}^\ell} \{\langle Kx, y\rangle + G(x) - F^*(y)\} \tag{5.12}$$

where F^* is the convex conjugate of F. It is defined as follows:

Definition 5.1 (Convex conjugate)
Let $f : \mathbb{R}^k \to \overline{\mathbb{R}}$ be a convex function. We call convex conjugate *the function f^* given by*

$$f^*(y) \coloneqq \sup_{x \in \mathbb{R}^k} \{\langle x, y\rangle - f(x)\} \quad .$$

Here, $\overline{\mathbb{R}}$ represents the extended real line which augments the set $\mathbb{R}$ with its two endpoints $-\infty$ and $+\infty$. Working with $\overline{\mathbb{R}}$ instead of $\mathbb{R}$ is a rather common strategy in convex analysis. By allowing functions to take the value $\pm\infty$ one can simply assume that functions are defined on the whole $\mathbb{R}^k$. Sometimes such approaches avoid cumbersome discussions that involve the domain of definition of a function. We refer to [96, Section 4] for more information on this topic. The convex conjugate is a very popular tool.

Besides optimisation theory it also finds applications in convex analysis, Lagrangian and Hamiltonian mechanics as well as thermodynamics. We point to the excellent books by Rockafellar [96] and Bonnans and Shapiro [93] for more details on this transform.

The just mentioned primal dual strategy computes iteratively

$$\begin{aligned} y^{(k+1)} &= \operatorname*{arg\,min}_{y \in \mathbb{R}^\ell} \left\{ \frac{1}{2} \left\| y - \left(y^{(k)} + \sigma K \hat{x}^{(k)} \right) \right\|_2^2 + \sigma F^* \left(y \right) \right\} \quad , \\ x^{(k+1)} &= \operatorname*{arg\,min}_{x \in \mathbb{R}^k} \left\{ \frac{1}{2} \left\| x - \left(x^{(k)} - \tau K^\top y^{(k+1)} \right) \right\|_2^2 + \tau G \left(x \right) \right\} \quad , \\ \hat{x}^{(k+1)} &= x^{(k+1)} + \theta \left(x^{(k+1)} - x^{(k)} \right) \quad . \end{aligned} \tag{5.13}$$

Chambolle and Pock [61] show that if $\tau\sigma\|K\|_2^2 < 1$, $\theta \in [0,1]$ and a few other regularity conditions concerning Eq. (5.11) are met, then the sequences $(x^{(k)})_k$, $(y^{(k)})_k$ generated by Eq. (5.13) converge towards a solution of Eq. (5.12). The strategy can be further improved by employing preconditioning ideas as presented in the work of Pock and Chambolle [62]. We also note that the updates in y and x are in fact proximal mappings.

Definition 5.2 (Proximal Mapping)
Let $f \colon \mathbb{R}^k \to \mathbb{R}$ be a proper, lower semi-continuous convex function and let $\gamma > 0$ be a positive real valued parameter. We call proximal mapping *the function* $\operatorname{prox}_{\gamma f}$ *given by*

$$\begin{aligned} \operatorname{prox}_{\gamma f} \left(x \right) &\coloneqq \operatorname*{arg\,min}_{z \in \mathbb{R}^k} \left\{ \gamma f \left(z \right) + \frac{1}{2} \|x - z\|_2^2 \right\} \\ &= \operatorname*{arg\,min}_{z \in \mathbb{R}^k} \left\{ f \left(z \right) + \frac{1}{2\gamma} \|x - z\|_2^2 \right\} \quad . \end{aligned}$$

The Moreau envelope is a closely related operator. Instead of yielding the minimisers, it returns the minimal value.

Definition 5.3 (Moreau-Yosida Regularisation, Moreau envelope)
Let $f \colon \mathbb{R}^k \to \mathbb{R}$ be a proper, lower semi-continuous convex function and let $\gamma > 0$ be a positive real valued parameter. We call Moreau-Yosida regularisation *or* Moreau envelope *the mapping ${}^\gamma f$ given by*

$${}^\gamma f \left(x \right) \coloneqq \inf_{z \in \mathbb{R}^k} \left\{ f \left(z \right) + \frac{1}{2\gamma} \|z - x\|_2^2 \right\} \quad .$$

These two operators go back to 1965 and were introduced by Moreau [97]. Due to their particularly advantageous properties they have been studied extensively in the literature during the last decades and form the building blocks of many modern optimisation strategies. A detailed analysis of the the Moreau envelope and the proximal mapping can for example be found in the book of Geiger and Kanzow [94, Chapter 6.4]. We will also come back to them later.

A detailed listing of the primal dual algorithm corresponding to Eq. (5.13) is given in Algorithm 5.1. We remark that this formulation does not include the preconditioning mentioned before. Suitable values for τ and σ can easily be computed with power iterations if the operator norm of K is not known exactly.

Algorithm 5.1: Primal dual algorithm from [61] for solving Eq. (5.12).

Input: τ, $\sigma > 0$, such that

$$\|K\|_2^2 < \frac{1}{\tau\sigma}$$

$\theta \in [0,1]$, $(x^{(0)}, y^{(0)}) \in \mathbb{R}^k \times \mathbb{R}^\ell$ arbitrary
Output: Optimal values x^* and y^*
Initialise: $\hat{x}^{(0)} = x^{(0)}$
repeat

$$\begin{aligned} y^{(n+1)} &= \operatorname{prox}_{\sigma F^*}\left(y^{(n)} + \sigma K \hat{x}^{(n)}\right) \\ x^{(n+1)} &= \operatorname{prox}_{\tau G}\left(x^{(n)} - \tau K^\top y^{(n+1)}\right) \\ \hat{x}^{(n+1)} &= x^{(n+1)} + \theta\left(x^{(n+1)} - x^{(n)}\right) \end{aligned}$$

until *convergence of the* $x^{(n)}$ *and* $y^{(n)}$
return *optimal* x^* *and* y^*

In order to apply the just described primal dual algorithm to our framework we only have to map parts of the energy in Eq. (5.10) to the functions F and G from Eq. (5.11). We opt for the following choice:

$$\begin{aligned} G(u,c) &\coloneqq \frac{\zeta_1}{2}\|u - z_1\|_2^2 + \lambda\|c\|_1 + \frac{\zeta_2}{2}\|c - z_2\|_2^2 \ , \\ F(Au + Bc) &\coloneqq \iota_{\{g\}}(Au + Bc) \ . \end{aligned} \tag{5.14}$$

Our choice is motivated by the fact that we obtain a particularly simple

expression for F^*. Indeed, a simple computation reveals that $F^*(x) = \langle x, g\rangle$. All in all this yields an algorithm that consists essentially of the following optimisation steps:

$$
\begin{aligned}
y^{(k+1)} &= \operatorname*{arg\,min}_{z\in\mathbb{R}^n} \left\{ \frac{1}{2}\left\| z - \left(y^{(k)} + \sigma\left(Au^{(k)} + Bc^{(k)} \right) \right) \right\|_2^2 + \sigma\langle z, g\rangle \right\} \quad , \\
u^{(k+1)} &= \operatorname*{arg\,min}_{z\in\mathbb{R}^n} \left\{ \frac{1}{2}\left\| z - \left(u^{(k)} - \tau A^\top y^{(k+1)} \right) \right\|_2^2 + \tau\frac{\zeta_1}{2}\|z - z_1\|_2^2 \right\} \quad , \\
c^{(k+1)} &= \operatorname*{arg\,min}_{z\in\mathbb{R}^n} \left\{ \frac{1}{2}\left\| z - \left(c^{(k)} - \tau B^\top y^{(k+1)} \right) \right\|_2^2 + \tau\left(\frac{\zeta_2}{2}\|z - z_2\|_2^2 + \lambda\|z\|_1 \right) \right\} .
\end{aligned}
$$

The first two optimisations are straightforward. We immediately obtain

$$
\begin{aligned}
y^{(k+1)} &= y^{(k)} + \sigma\left(Au^{(k)} + Bc^{(k)} - g \right) \quad , \\
u^{(k+1)} &= \frac{u^{(k)} - \tau A^\top y^{(k+1)} + \tau\zeta_1 z_1}{1 + \tau\zeta_1} \quad .
\end{aligned}
$$

The update in c requires a few additional preliminary results. First of all we remark that the proximal mapping of the absolute value can be expressed in closed form as a soft shrinkage [98]:

$$
\operatorname{prox}_{\gamma|\cdot|}(x) = \operatorname*{shrink}_{\gamma}(x) \coloneqq \begin{cases} x - \gamma, & x > \gamma \\ 0, & |x| \leqslant \gamma \\ x + \gamma, & x < -\gamma \end{cases} \quad .
$$

This identity can also be applied in higher dimensions to $\|\cdot\|_1$ since the minimisation decouples into independent optimisation steps. In that case the soft shrinkage operates component wise on the entries of the input vector. The following two findings are straightforward but useful to simplify certain optimisation problems with many summands.

Lemma 5.4
Let γ, λ and x be arbitrary real numbers with $\gamma > 0$ and $\lambda > 0$. Then we have the following relationship

$$
\operatorname*{shrink}_{\frac{\lambda}{\gamma}}\left(\frac{x}{\gamma} \right) = \frac{1}{\gamma}\operatorname*{shrink}_{\lambda}(x) \quad .
$$

Proof. This identity follows immediately from the definition of the soft shrinkage operator.

$$\begin{aligned}\underset{\frac{\lambda}{\gamma}}{\operatorname{shrink}}\left(\frac{x}{\gamma}\right) &= \begin{cases} \frac{x}{\gamma} - \frac{\lambda}{\gamma}, & \frac{x}{\gamma} > \frac{\lambda}{\gamma} \\ 0, & \left|\frac{x}{\gamma}\right| \leqslant \frac{\lambda}{\gamma} \\ \frac{x}{\gamma} + \frac{\lambda}{\gamma}, & \frac{x}{\gamma} < -\frac{\lambda}{\gamma} \end{cases} \\ &= \frac{1}{\gamma} \underset{\lambda}{\operatorname{shrink}}(x)\end{aligned}$$

□

Proposition 5.5
Let $(f^{(i)})_i$ be a family of m arbitrary vectors in $\mathbb{R}^k$. We consider the convex optimisation problem

$$\underset{x\in\mathbb{R}^k}{\arg\min}\left\{\lambda\|x\|_1 + \sum_{i=1}^{m}\frac{\alpha_i}{2}\left\|x - f^{(i)}\right\|_2^2\right\}$$

with positive real valued weights α_i and λ. This problem has the same unique solution as

$$\underset{x\in\mathbb{R}^k}{\arg\min}\left\{\frac{\lambda}{\sum_{j=1}^m \alpha_j}\|x\|_1 + \frac{1}{2}\left\|x - \frac{\sum_{i=1}^m \alpha_i f^{(i)}}{\sum_{j=1}^m \alpha_j}\right\|_2^2\right\}$$

and the minimiser is given by

$$\frac{1}{\sum_{j=1}^m \alpha_j} \underset{\lambda}{\operatorname{shrink}}\left(\sum_{i=1}^{m}\alpha_i f^{(i)}\right) \quad ,$$

where the soft shrinkage is applied component wise onto its argument.

Proof. We perform a simple quadratic completion and remove all constant terms. This change has no influence on the minimiser. It merely shifts the minimal value. It follows that we can regroup all the quadratic terms into a single one:

$$\begin{aligned}&\underset{x\in\mathbb{R}^k}{\arg\min}\left\{\lambda\|x\|_1 + \sum_{i=1}^{m}\frac{\alpha_i}{2}\left\|x - f^{(i)}\right\|_2^2\right\} \\ &\qquad = \underset{x\in\mathbb{R}^k}{\arg\min}\left\{\frac{\lambda}{\sum_{j=1}^m \alpha_j}\|x\|_1 + \frac{1}{2}\left\|x - \frac{\sum_{i=1}^m \alpha_i f^{(i)}}{\sum_{j=1}^m \alpha_j}\right\|_2^2\right\} \quad .\end{aligned}$$

The latter problem has a closed form solution in terms of the soft shrinkage operator. In conjunction with Lemma 5.4 this gives us

$$\arg\min_{x\in\mathbb{R}^k}\left\{\lambda\|x\|_1+\sum_{i=1}^{m}\frac{\alpha_i}{2}\left\|x-f^{(i)}\right\|_2^2\right\}=\frac{1}{\sum_{j=1}^{m}\alpha_j}\operatorname*{shrink}_{\lambda}\left(\sum_{i=1}^{m}\alpha_i f^{(i)}\right)\quad.$$

□

Using Lemma 5.4 and Proposition 5.5 it follows that the update in the mask c can be expressed as

$$c^{(k+1)}=\operatorname*{shrink}_{\frac{\tau\lambda}{1+\tau\zeta_2}}\left(\frac{c^{(k)}-\tau B^\top y^{(k+1)}+\tau\zeta_2 z_2}{1+\tau\zeta_2}\right)\quad.$$

All in all, after exchanging the aliases ζ_1, z_1, ζ_2 and z_2 with their respective values we obtain the iterative strategy depicted in Algorithm 5.2. Note that all the involved operations are favourable to parallelisation and can be performed relatively fast.

It is important to remember that the optimal pair (u^*, c^*) obtained from Algorithm 5.2 is in general not a feasible point for the problem stated in Eq. (5.3). It only represents a solution of Eq. (5.9). As a remedy, we use this pair to compute a new first-order approximation of the previously defined function T and repeat all the steps until a fixed-point is reached. The complete algorithm to solve Eq. (5.3) is given in Algorithm 5.3.

The back projection step at Line 1 in Algorithm 5.3 forces each iterate to be feasible. At first glance, such a behaviour sounds appealing for practical purposes. We can abort the algorithm at any moment and be sure that we have a mask with a corresponding solution of the inpainting equation to our avail, even if we have not yet reached a fixed-point. However, this additional step renders the convergence analysis significantly more difficult. As we will see in Section 5.4, we cannot assert an unconditional monotonic decrease in the energy if this optional inpainting step is performed.

At this point we still ignore if the iterative scheme from Algorithm 5.3 yields meaningful results. We postpone the thorough discussion of this topic to Section 5.4. The next section deals with an alternative approach to solve Eq. (5.10). We remind that this expression is similar to the primal problem (P_w) mentioned at the beginning of this chapter. As we will see now, there exists a closely linked optimisation task which is known in the literature as

Algorithm 5.2: Minimisation strategy for solving Eq. (5.9)

Input: N the number of iterations.
Output: Vectors u^* and c^* solving Eq. (5.9)
Initialise: τ, $\sigma > 0$ such that

$$\left\| \begin{pmatrix} A & B \end{pmatrix} \right\|_2^2 < \frac{1}{\sigma\tau} \ ,$$

$\theta \in [0,1]$, $u^{(0)}$, $c^{(0)}$, $y^{(0)}$ arbitrary,
$\hat{u}^{(0)} = u^{(0)}$ and $\hat{c}^{(0)} = c^{(0)}$
repeat

Compute proximal update steps:

$$y^{(k+1)} = y^{(k)} + \sigma \left(A\hat{u}^{(k)} + B\hat{c}^{(k)} - g \right)$$

$$u^{(k+1)} = \frac{u^{(k)} - \tau \left(A^\top y^{(k+1)} - f - \mu\bar{u} \right)}{1 + \tau + \mu\tau}$$

$$c^{(k+1)} = \underset{\frac{\tau\lambda}{1+\tau\varepsilon+\tau\mu}}{\text{shrink}} \left(\frac{c^{(k)} - \tau B y^{(k+1)} + \tau\mu\bar{c}}{1 + \tau\varepsilon + \tau\mu} \right)$$

Perform extrapolation of the iterates:

$$\hat{u}^{(k+1)} = u^{(k+1)} + \theta \left(u^{(k+1)} - u^{(k)} \right)$$

$$\hat{c}^{(k+1)} = c^{(k+1)} + \theta \left(c^{(k+1)} - c^{(k)} \right)$$

until *convergence of* $u^{(k)}$ *and* $c^{(k)}$
return *Optimal* u^* *and* c^*

Algorithm 5.3: Minimisation strategy for solving Eq. (5.3)

Input: Image f, parameters λ, ε, μ.
Output: Vectors u^* and c^* solving Eq. (5.3)
Initialise: $u^{(0)} = f$, $c_i^{(0)} = 1\ \forall i$
for $k \geqslant 0$ **do**
  Compute first-order approximation of $T(u, c)$ around $\left(u^{(k)}, c^{(k)}\right)$.
  Obtain $u^{(k+1)}$ and $c^{(k+1)}$ by solving Eq. (5.9) with Algorithm 5.2.
1 Optionally compute feasible $u^{(k+1)}$ from $c^{(k+1)}$ by solving the inpainting equation $T(u, c^{(k+1)}) = 0$ for u.
  if $u^{(k+1)} = u^{(k)}$ *and* $c^{(k+1)} = c^{(k)}$ **then**
    Set $c^* = c^{(k+1)}$
    Compute feasible u^* from c^* by solving the inpainting equation $T(u, c^*) = 0$ for u.
    Exit for loop.
  end
end
return *Optimal* u^* *and* c^*

dual problem. Primal and dual problems are related via the convex conjugate transform and it is possible to derive the solution of one of these problems if the solution of the other one is known. Our next goal is to find this dual formulation. As we will see, it is well suited for numerical optimisations and can be solved efficiently with simple tools like a gradient descent.

5.3 Dual formulation of the linearised optimal control model

We have introduced an optimal control model with a strictly convex but non-differentiable objective function and a non-linear PDE as a constraint in the previous section. In order to find a solution we have suggested to linearise the corresponding constraint and to solve the resulting sequence of convex optimisation problems iteratively. As announced before, we now derive an alternative formulation of our linearised problem by means of conjugate duality. To this end we will use the associated primal minimisation problem (P_w) and base our presentation on the theory from the book of Bonnans and Shapiro [93, Chapter 2.5]. An in-depth study of the implications of duality approaches on convex programming can also be found in the work of Boţ [99]. We recall that (P_w) is given by

$$(P_w) \qquad \min_{u,c\in\mathbb{R}^n} \left\{ \frac{\zeta_1}{2}\|u - z_1\|_2^2 + \lambda\|c\|_1 + \frac{\zeta_2}{2}\|c - z_2\|_2^2 \right\} \\ Au + Bc + w = g \ .$$

Conjugate duality offers fruitful insights into many optimisation problems. The excellent reputation of duality in optimisation theory comes from its major role in formulating necessary and sufficient optimality conditions and, consequently, in its usefulness in the quest for new algorithmic approaches for solving mathematical programming tasks. The results from this section are mainly driven by our curiosity in investigating the benefits of duality within our framework.

By applying the convex conjugate twice onto an optimisation problem we obtain a completely new but equivalent description of our initial task. This new formulation is called dual problem. Further, the usage of the convex conjugate induces a symmetry between the primal and dual problem. This symmetry yields a one-to-one mapping between the properties of both

models. As a consequence, the solutions of the two tasks are also related in a very concise manner. In view of these claims it becomes clear that conjugate duality is as important for convex analysis as the Fourier transform is for linear system theory.

Our goal is to inspect these acclaimed benefits. We want to analyse the optimality conditions of the dual model as well as suitable numerical solving strategies. Our hope is that we gain faster numerics and a more profound insight into the nature of the underlying task.

Before stating the dual model we briefly sketch the steps to be taken and introduce certain essential concepts and results. We recall that the convex conjugate of a function $f : \mathbb{R}^k \to \overline{\mathbb{R}}$ is defined as

$$f^*(y) \coloneqq \sup_{x \in \mathbb{R}^k} \{\langle x, y \rangle - f(x)\} \quad .$$

If f is proper, lower semi-continuous and convex, then its conjugate is also proper and convex. We refer to [93, Proposition 2.112] for a proof of this result. In similar spirit we define the biconjugate function f^{**} as the conjugate of the conjugate, i.e. $(f^*)^*$. Under certain mild assumptions, which we will specify in a few moments, one has $f = f^{**}$. This observation is crucial for the forthcoming presentation. Finally, we briefly note that the conjugate is a generalisation of the Legendre Fenchel transform and refer to [96] for more details.

In order to explain the derivation of the dual problem in detail we consider the following exemplary optimisation task:

$$\inf_{x \in \mathbb{R}^k} \{f(x)\} \quad .$$

Here, f can be any function which maps from $\mathbb{R}^k$ to $\overline{\mathbb{R}}$. The fundamental idea behind conjugate duality is to embed the previous optimisation into a family of parametric problems of the form:

$$\operatorname{val}(w) \coloneqq \inf_{x \in \mathbb{R}^k} \{\varphi(x, w)\}$$

where $\varphi \colon \mathbb{R}^k \times \mathbb{R}^\ell \to \overline{\mathbb{R}}$ is an arbitrary function such that $\varphi(\cdot, 0) = f(\cdot)$. We emphasise that in general neither f nor φ are required to be convex. The unknown $w \in \mathbb{R}^\ell$ is often referred to as a perturbation parameter. It slightly alters the optimisation task with its fluctuations. The function $\operatorname{val} \colon \mathbb{R}^\ell \to \overline{\mathbb{R}}$

is also called value function. Following the presentation of Bonnans and Shapiro [93, Section 2.5] we define the dual problem to be the biconjugate val^{**}. A simple computation [93, Eq. 2.265] reveals that it can be written as

$$\mathrm{val}^{**}(w) = \sup_{d \in \mathbb{R}^\ell} \{\langle d, w\rangle - \varphi^*(0, d)\} \quad .$$

The conjugate of φ is considered with respect to all of its arguments. Thus, we have

$$\varphi^*(y, d) \coloneqq \sup_{\substack{x \in \mathbb{R}^k \\ w \in \mathbb{R}^\ell}} \{\langle x, y\rangle + \langle w, d\rangle - \varphi(x, w)\} \quad .$$

Now we take the just mentioned steps and apply them to our linearised convex optimisation problem (P_w). Using the definitions from Eq. (5.14), we can rewrite it in a compact form as

$$(P_w) \qquad \mathrm{val}(w) = \inf_{u,c \in \mathbb{R}^n} \{G(u, c) + F(Au + Bc + w)\} \quad .$$

Note that we have introduced the perturbation parameter w into the linear constraints. This rather common choice leads to a particularly useful structure. The dual problem can now be stated as

$$(D_w) \qquad \mathrm{val}^{**}(w) = -\inf_{p \in \mathbb{R}^n} \left\{F^*(p) + G^*\left(-A^\top p, -B^\top p\right) - \langle p, w\rangle\right\} \quad .$$

We refer to the the book of Bonnans and Shapiro [93, Sections 2.5.3 and 2.5.4] for detailed derivations of this result. It is a well known fact from convex analysis that the biconjugate is a lower bound for a considered function. This means we always have $f^{**}(p) \leqslant f(p)$ for any function f [93, Section 2.4.2]. The difference $f(p) - f^{**}(p)$ is called duality gap. In view of our original problem (P_0) we have in particular $\mathrm{val}(0) - \mathrm{val}^{**}(0) \geqslant 0$. If this duality gap is 0, then it is possible to obtain solutions for (P_0) by solving the corresponding dual problem (D_0). A result from Bonnans and Shapiro [93, Remark 2.172] asserts that the duality gap is indeed 0 if the linear system $Au + Bc = g$ has at least one solution. An assertion that we implicitly assume to hold true. Thus, we have $\mathrm{val}(0) = \mathrm{val}^{**}(0)$ and can use all our findings from the dual model to improve our understanding of the primal formulation.

In a next step we want to state an explicit expression for the cost function of the dual problem $\mathrm{val}^{**}(w)$. To this end we only need to derive the

expressions for the convex conjugates of F and G. The convex conjugate of F is well known and given by $F^*(p) = \langle g, p \rangle$. A short computation also reveals that G^* is given by

$$\begin{aligned} G^*(r,s) &= \sup_{x,y \in \mathbb{R}^n} \{\langle r, x \rangle + \langle s, y \rangle - G(x,y)\} \\ &= \frac{1}{2\zeta_1}\|r\|_2^2 + \langle r, z_1 \rangle - \frac{\zeta_2}{2}\|z_2\|_2^2 + \frac{1}{2\zeta_2}\|s + \zeta_2 z_2\|_2^2 - \\ &\qquad \lambda \inf_{y \in \mathbb{R}^n} \left\{ \|y\|_1 + \frac{\zeta_2}{2\lambda} \left\| y - \frac{s + \zeta_2 z_2}{\zeta_2} \right\|_2^2 \right\} \quad . \end{aligned} \tag{5.15}$$

The remaining infimum corresponds to the Moreau envelope of $\|\cdot\|_1$ in $\mathbb{R}^n$. Clearly, the computation of this infimum decouples into the sum of n evaluations of the Moreau envelope of the absolute value function. An analytic expression of the latter can be given in terms of the Huber penalty- or Huber loss function H_γ with positive parameter γ:

$$H_\gamma(x) \coloneqq \begin{cases} \frac{1}{2}x^2, & |x| \leqslant \gamma \\ \gamma\left(|x| - \frac{\gamma}{2}\right), & |x| > \gamma \end{cases} \quad .$$

It originally stems from the work of Huber [100] and currently represents one of the most popular robust penalisers in regression analysis. We already know that the proximal mapping of the absolute value is given by a soft shrinkage. Plugging the definition of the soft shrinkage operator into the cost function of the Moreau envelope of the absolute value reveals that ${}^{\gamma}|\cdot|(x) = \gamma^{-1} H_\gamma(x)$. As a consequence the Moreau envelope of $\|\cdot\|_1$ in $\mathbb{R}^k$ becomes

$${}^{\gamma}\|\cdot\|_1(x) = \frac{1}{\gamma}\sum_{i=1}^{k} H_\gamma(x_i) \quad . \tag{5.16}$$

The next identity will also be useful in the forthcoming discussion.

Lemma 5.6
Let λ and γ be two positive real valued scalar parameters and $x \in \mathbb{R}$ arbitrary. Then we have the following identity for the Huber penalty function:

$$H_{\lambda\gamma^{-1}}\left(\gamma^{-1}x\right) = \gamma^{-2} H_\lambda(x) \quad .$$

Proof. The proof consists of a straightforward computation. By definition of the Huber penalty function we have

$$H_{\lambda\gamma^{-1}}\left(\gamma^{-1}x\right) = \begin{cases} \frac{1}{2}\gamma^{-2}x^2, & |\gamma^{-1}x| \leqslant \lambda\gamma^{-1} \\ \lambda\gamma^{-1}\left(|\gamma^{-1}x| - \frac{\lambda\gamma^{-1}}{2}\right), & |\gamma^{-1}x| > \lambda\gamma^{-1} \end{cases}$$
$$= \gamma^{-2}H_{\lambda}(x) \quad .$$

□

Now we exploit our newly gained knowledge to simplify the expression of G^*. Applying Eq. (5.16) and Lemma 5.6 on Eq. (5.15) yields

$$G^*(r,s) = \frac{1}{2\zeta_1}\|r\|_2^2 + \langle r, z_1\rangle - \frac{\zeta_2}{2}\|z_2\|_2^2 + \frac{1}{2\zeta_2}\|s+\zeta_2 z_2\|_2^2 - \frac{1}{\zeta_2}\sum_{i=1}^{n} H_{\lambda}\Big((s+\zeta_2 z_2)_i\Big) \quad .$$

As a final step we exchange the aliases with their original definitions. This gives us the following relationship:

$$G^*(r,s) = \frac{1}{2(1+\mu)}\|r\|_2^2 + \left\langle r, \frac{\mu\bar{u}+f}{1+\mu}\right\rangle - \frac{1}{2(\varepsilon+\mu)}\|\mu\bar{c}\|_2^2 + \frac{1}{2(\varepsilon+\mu)}\|s+\mu\bar{c}\|_2^2 - \frac{1}{\varepsilon+\mu}\sum_{i=1}^{n} H_{\lambda}(s_i+\mu\bar{c}_i) \quad .$$

It remains to evaluate $G^*(r,s)$ at $r=-A^\top p$ and $s=-B^\top p$ to obtain the explicit form of the dual problem (D_w):

$$(D_w)\quad \mathrm{val}^{**}(w) = -\inf_p \left\{ \left\langle p, g-w-\frac{\mu A\bar{u}+Af}{1+\mu}\right\rangle + \frac{1}{2(1+\mu)}\left\|A^\top p\right\|_2^2 - \frac{1}{2(\varepsilon+\mu)}\|\mu\bar{c}\|_2^2 + \frac{1}{2(\varepsilon+\mu)}\|-Bp+\mu\bar{c}\|_2^2 - \frac{1}{\mu+\varepsilon}\sum_{i=1}^{n} H_{\lambda}(-B_{i,i}p_i+\mu\bar{c}_i)\right\} \quad .$$

We remark that a positive value of ε in (D_w) avoids divisions by 0 if we would let μ run towards 0. For convenience we will refer to the cost function

in (D_w) as $h_D(p)$ in the forthcoming paragraphs. Clearly h_D is a convex function since it is a sum of convex and affine functions. Obviously it is also continuous. However, it is not necessarily a strictly convex function. It is coercive if $\ker(A^\top) \cap \ker(B) = \{0\}$. Finally we note that the cost function of our dual formulation is continuously differentiable. This follows immediately from the observation that the Huber loss function is a rescaled Moreau envelope and that the Theorem of Danskin (Theorem 6.37 in [94]) asserts the differentiability of the Moreau envelope.

The fact that the dual problem is an unconstrained optimisation task with a continuously differentiable cost function renders it significantly more attractive than the primal approach. We claim that the dual problem is much easier to tackle from a numerical point of view than the primal problem (P_w). We benefit from the fact that the energy has such a structure where all terms are either linear or quadratic and therefore easy to manipulate. A simple but efficient way to approach the dual problem numerically consists in applying a gradient descent scheme as depicted in Algorithm 5.4. Determining the optimal step size can efficiently be performed by a line search (i.e. an inexpensive 1D optimisation). We refer to the textbook of Nocedal and Wright [101] for an extensive discussion on line search methods.

Algorithm 5.4: Gradient descent scheme for solving the dual problem

Output: Optimal dual solution p^*
Initialise: Choose arbitrary initial $p^{(0)}$
repeat

$$\alpha^{(k+1)} = \operatorname*{arg\,min}_{\alpha>0} \left\{ h_D \left(p^{(k)} - \alpha \nabla h_D \left(p^{(k)} \right) \right) \right\}$$
$$p^{(k+1)} = p^{(k)} - \alpha^{(k+1)} \nabla h_D \left(p^{(k)} \right)$$

until *fixed-point* p^* *is reached*
return *Optimal* p^*

Solving the dual problem yields an optimal value for the unknown p. However, our desired image and mask information are still encoded in the variables u and c when $w = 0$. Since we are mostly interested in solutions of our original problem we present the forthcoming results for the case $w = 0$

only. In order to obtain an optimal mask and the corresponding image data we have to use the Karush Kuhn Tucker (KKT) conditions. They state the relationship between the primal and dual solution. To this end we also need the standard Lagrangian (in the following simply referred to as Lagrangian) of Eq. (5.9). It is given by

$$L(u, c, p) \coloneqq G(u, c) + \langle Au + Bc, p \rangle \ . \tag{5.17}$$

We refer to [93, Section 2.5.3] for a more exhaustive presentation of the Lagrangian and its related concepts. In [93, Eq. (2.301), Theorem 2.158 and Proposition 3.3] the KKT conditions for optimal u^*, c^* and p^* are discussed in detail and stated as

$$\begin{cases} (u^*, c^*) = \underset{u,c \in \mathbb{R}^n}{\arg\min} \{L(u, c, p^*)\} \ , \\ p^* \in \partial(F)(Au^* + Bc^*) \ , \end{cases} \tag{5.18}$$

where $\partial(F)(x)$ denotes the subdifferential of F at position x. We refer to [93, Section 2.4.3] and [96, Section 23] for a thorough presentation on the concepts of subdifferentiability and subgradients and simply mention that the subdifferential of a convex function $\psi \colon \mathbb{R}^\ell \to \mathbb{R}$ is given by

$$\partial(\psi)(x) \coloneqq \left\{ y \in \mathbb{R}^\ell \;\middle|\; \psi(z) - \psi(x) \geqslant \langle y, z - x \rangle \ \forall z \in \mathbb{R}^\ell \right\}$$

and that the individual elements of this set are called subgradients. They provide a comfortable framework with similar properties as classical gradients in presence of non-differentiable convex functions. A particularly important property of the subdifferential is the following fact: An unknown x is a minimiser of a convex function ψ if and only if $0 \in \partial(\psi)(x)$.

Note that the Lagrangian from Eq. (5.17) is strictly convex and coercive for fixed p. Therefore, the first KKT condition is equivalent to

$$\begin{cases} (1 + \mu)\left(u^* - \dfrac{\mu\bar{u} + f}{1 + \mu}\right) + A^\top p^* = 0 \ , \\ \lambda\partial(\|\cdot\|_1)(c^*) + (\varepsilon + \mu)\left(c^* - \dfrac{\mu\bar{c}}{\varepsilon + \mu}\right) + B^\top p^* \ni 0 \ . \end{cases} \tag{5.19}$$

This result follows immediately by computing the (sub-) gradient of $L(u, c, p^*)$ with respect to u and c and setting it to 0. Here and in the subsequent

equations, the tuple $(\bar{u}, \bar{c})$ represents the point around with the linearisation of $T(u, c)$ was performed. In [96, Corollary 23.5.1] it is further shown that

$$x^* \in \partial(\psi)(x) \quad \Leftrightarrow \quad x \in \partial(\psi^*)(x^*)$$

if ψ is a closed, proper, and convex function and where ψ^* represents its convex conjugate. In view of this fact, the second KKT condition in Eq. (5.18) can be rewritten as

$$Au^* + Bc^* \in \partial(F^*)(p^*) = g \ . \tag{5.20}$$

Thus, the previous condition simply requires that optimal solutions must fulfil the linear system $Au + Bc = g$. Let us now assume that we have an optimal p^* which solves the dual problem to our avail. Then we get u^* from Eq. (5.19) via

$$u^* = \frac{\mu\bar{u} + f - A^\top p^*}{1 + \mu}$$

and Eq. (5.20) gives us c^* by solving $Bc^* = g - Au^*$. Note that B is a diagonal matrix. Unfortunately we are unable to make any statements about the entries on the main diagonal and some of them could very well be zero. Therefore, this linear system does not necessarily have a unique solution. Using the Moore-Penrose pseudoinverse $B^\dagger$ of the matrix B gives us the best approximation in the least squares sense and a result with maximal sparsity. Another disadvantage of this strategy for recovering c^* is given by the numerical difficulties related to the Moore-Penrose pseudoinverse. In the presence of very small entries in B and $g - Au^*$ we might suffer from severe rounding and cancellation errors during the computation of $B^\dagger(g - Au^*)$. Therefore, we should consider an alternative approach to recover the optimal value for c^* from p^* and u^*. To this end note that the second equation in Eq. (5.19) corresponds to $\mu\bar{c} - Bp^* \in ((\varepsilon + \mu) I + \lambda\partial(\|\cdot\|_1))(c^*)$. The latter expression is exactly the optimality condition for c^* being a solution of

$$\operatorname*{arg\,min}_{x \in \mathbb{R}^n} \left\{ \|x\|_1 + \frac{\varepsilon + \mu}{2\lambda} \left\| x - \frac{\mu\bar{c} - Bp^*}{\varepsilon + \mu} \right\|_2^2 \right\} \ .$$

To acknowledge this claim it suffices to compute the subdifferential of the previous cost function and require that 0 is a subgradient. As mentioned already several times, we can solve the above optimisation task by using

the soft shrinkage operator. Thus, we can retrieve the optimal mask by computing

$$c^* = \underset{\frac{\lambda}{\varepsilon+\mu}}{\operatorname{shrink}}\left(\frac{\mu\bar{c} - Bp^*}{\varepsilon+\mu}\right) = \frac{1}{\varepsilon+\mu}\underset{\lambda}{\operatorname{shrink}}\left(\mu\bar{c} - Bp^*\right) \quad .$$

This latter method is significantly more reliable and stable compared to inverting the non-zero diagonal entries of the matrix B.

At this point, we have two distinct strategies to our avail to solve our linearised optimal control model. These approaches are equivalent in the sense that they yield exactly the same solutions. However, their performance might differ. We refer to Section 5.5 for a detailed analysis of this topic. Our next goal is the analysis of the convergence behaviour of our strategy with respect to the iterative linearisation detailed in Eq. (5.7).

5.4 Convergence analysis

Let us remind that our initial problem formulation from Eq. (5.3) has a non-linear constraint. Our suggestion to avoid the explicit handling of these non-linearities is to use an iterative linearisation. The convergence behaviour of our scheme has not been analysed until now. The forthcoming paragraphs deal with this outstanding question. We discuss the optimality of fixed-points of our iterative strategy from Eq. (5.7) and the overall properties of the iterates. We provide clear statements for two different setups, namely with the back projection step onto the feasible set at Line 1 in Algorithm 5.3 and without this back projection.

The following proposition gives us necessary optimality conditions for Eq. (5.3). It is based on a result from Tröltzsch [88, Equation (2.109)] and is valid with and without additionally solving $T(u, c) = 0$ for u. In [88] the author suggests necessary optimality conditions for an optimal control model with non-linear constraints. We provide a version of these statements that is adapted to our task at hand.

Proposition 5.7
Let us assume that there exists a solution $(u^*, c^*) \in \mathbb{R}^n \times \mathbb{R}^n$ *of Eq.* (5.3) *and let* T *be the operator from Eq.* (5.4)*. Let* $D_u T\,(u^*, c^*)$ *be invertible. Then, for any optimal pair* $(u^*, c^*) \in \mathbb{R}^n \times \mathbb{R}^n$ *of Eq.* (5.3) *there must exist a vector*

$p^* \in \mathbb{R}^n$ such that the following relations are fulfilled.

$$\begin{aligned} u^* - f + D_u T\left(u^*, c^*\right)^\top p^* &= 0 \ , \\ \lambda \partial \left(\|\cdot\|_1\right)\left(c^*\right) + \varepsilon c^* + D_c T\left(u^*, c^*\right)^\top p^* &\ni 0 \ , \\ T\left(u^*, c^*\right) &= 0 \ . \end{aligned} \tag{5.21}$$

Proof. The implicit function theorem asserts that there exist open neighbourhoods $\mathcal{O}\left(u^*\right)$ and $\mathcal{O}(c^*)$ around u^* and c^* as well as a continuously differentiable mapping $S\colon \mathcal{O}(c^*) \to \mathcal{O}(u^*)$ with $S\left(c^*\right) = u^*$ such that

$$T\left(S\left(c\right), c\right) = 0 \quad \forall c \in \mathcal{O}\left(c^*\right) \ .$$

Further, the Jacobian $DS\left(c\right)$ of S is given by

$$DS(c) = -D_u T(S(c), c)^{-1} D_c T(S(c), c) \ .$$

Here, $D_u T(u, c)$ denotes the Jacobian of T with respect to the first variable and likewise $D_c T(u, c)$ denotes the Jacobian of T with respect to the second variable. For the sake of simplicity, let us now denote our discrete energy functional from Eq. (5.3) by $J(u, c)$. Plugging $S(c)$ and c into J and requiring that 0 is a subgradient leads us to the following system of equations

$$DS(c)^\top \nabla_u J\left(S\left(c\right), c\right) + \partial_c(J)(S(c), c) \ni 0 \ ,$$

where $\nabla_u J$ represents the gradient of J with respect to its first variable and $\partial_c(J)$ the subgradient with respect to the second variable. This system contains the necessary optimality conditions that must be fulfilled. Expanding the expression for the Jacobian of S in c gives us

$$-D_c T(S(c), c)^\top D_u T\left(S(c), c\right)^{-\top} \nabla_u J\left(S\left(c\right), c\right) + \partial_c(J)(S(c), c) \ni 0$$

as a requirement for an optimum. Following the splitting proposed in [88], we introduce an additional variable p to avoid the explicit usage of the matrix inverse:

$$\begin{aligned} D_u T\left(S(c), c\right)^\top p + \nabla_u J\left(S\left(c\right), c\right) &= 0 \ , \\ D_c T(S(c), c)^\top p + \partial_c(J)(S(c), c) &\ni 0 \ . \end{aligned}$$

By considering the previous system of equations at position c^* and $u^* = S(c^*)$ and inserting the respective expressions for the (sub-) gradients of J we obtain the final form of our optimality conditions:

$$\begin{aligned} u^* - f + D_u T(u^*, c^*)^\top p^* &= 0 \ , \\ \lambda \partial \left(\|\cdot\|_1 \right)(c^*) + \varepsilon c^* + D_c T(u^*, c^*)^\top p^* &\ni 0 \ . \end{aligned}$$

The last requirement $T(u^*, c^*) = 0$ is obvious and does not need any proof. □

Note that the previous proposition requires the invertibility of the inpainting matrix at the optimum. As already mentioned, our iterative scheme itself does not impose any such restrictions. We also remind that Section 2.2 provides simple criteria that allow us to verify if the inpainting matrix is indeed invertible.

Our next result shows that feasible fixed-points of Algorithm 5.3 fulfil the necessary optimality conditions from Eq. (5.18) and Eq. (5.21). As a consequence, they represent good candidates for a solution. We emphasise that this result does not explicitly rely on the back projection. It merely requires that fixed-points are feasible. Nevertheless, we do not know yet if our iterates converge towards such a feasible solution if the back projection is omitted.

Proposition 5.8
If Algorithm 5.2 has reached a feasible fixed-point (i.e. one that fulfils $T(u, c) = 0$) with respect to the linearisation point (that means the minimiser of Eq. (5.9) *is equal to the point around which PDE was linearised), then this fixed-point (u^*, c^*) must fulfil the conditions in Eq.* (5.18) *and it also fulfils the necessary optimality conditions derived in Proposition 5.7.*

Proof. By requirement u^* and c^* are feasible and thus $T(u^*, c^*) = 0$ holds. Further, u^* and c^* are solutions of the linearised problem and hence they fulfil the KKT optimality conditions stated in Eq. (5.18), respectively Eq. (5.19), by construction. Since we are in the presence of a fixed-point it follows that our solution (u^*, c^*) coincides with the location $(\overline{u}, \overline{c})$ around which the linearisation took place. Thus, $u^* = \overline{u}$ and $c^* = \overline{c}$ hold. We conclude further that $A = D_u T(u^*, c^*)$ and $B = D_c T(u^*, c^*)$. These four relations imply that the equations from Eq. (5.19) coincide with the first two identities in Eq. (5.21). □

The previous proposition has required that fixed-points are feasible. The following proposition shows that such an assumption is not necessary. If we reach a fixed-point with our iterative scheme, then this point will automatically be feasible.

Proposition 5.9
If Algorithm 5.2 has reached a fixed-point with respect to the linearisation point (that means the minimiser of Eq. (5.9) *is equal to the point around which PDE was linearised), then this fixed-point* (u^*, c^*) *is always feasible. This means it fulfils* $T(u^*, c^*) = 0$.

Proof. We have the following matrices and vectors occurring in the Taylor approximation when a fixed-point is reached:

$$\begin{aligned} A\left(u^*, c^*\right) &= D_u T\left(u^*, c^*\right) \quad , \\ B\left(u^*, c^*\right) &= D_c T\left(u^*, c^*\right) \quad , \\ g\left(u^*, c^*\right) &= D_u T\left(u^*, c^*\right) u^* + D_c T\left(u^*, c^*\right) c^* - T\left(u^*, c^*\right) \quad . \end{aligned}$$

Fixed-points solve the linearised problem. Thus,

$$A\left(u^*, c^*\right) u^* + B\left(u^*, c^*\right) c^* - g\left(u^*, c^*\right) = 0$$

holds. Inserting the corresponding definitions immediately shows that this implies $T(u^*, c^*) = 0$. □

Proposition 5.8 and Proposition 5.9 show that if our iterates converge towards a fixed-point, then this fixed-point also fulfils all the necessary optimality conditions of the initial problem. Whether the back projection onto the feasible set of solutions is performed or not is completely irrelevant. The final result always fulfils $T(u, c) = 0$.

Since our initial problem was not convex, the conditions from Proposition 5.7 are only necessary conditions and not sufficient ones for a minimum. Even if our fixed-points fulfil them, we cannot say anything about the nature of the solution. It could just as well be a local maximum or a saddle point. The following proposition gives some insight into the behaviour of our iterates and states conditions under which the energy will necessarily decrease. This result is specially tailored to take the additional back projection into account. It shows that the decrease of the energy is not unconditional.

Proposition 5.10
Let the solution obtained from Algorithm 5.2 be given by $(u^{(k+1)}, c^{(k+1)})$ *and the point around which the linearisation has been performed by* $(u^{(k)}, c^{(k)})$. *Further assume that* $(u^{(k)}, c^{(k)})$ *fulfils* $T(u^{(k)}, c^{(k)}) = 0$ *and let* $\tilde{u}^{(k+1)}$ *fulfil* $T(\tilde{u}^{(k+1)}, c^{(k+1)}) = 0$, *too. Thus,* $(\tilde{u}^{(k+1)}, c^{(k+1)})$ *is the back projection of* $(u^{(k+1)}, c^{(k+1)})$ *onto the set of feasible locations. Then* $(\tilde{u}^{(k+1)}, c^{(k+1)})$ *will be a feasible pair of iterates that decreases the energy if the following condition is valid.*

$$\begin{aligned}\frac{1}{2}\left(\left\|\tilde{u}^{(k+1)} - f\right\|_2^2 - \|u^{(k)} - f\|_2^2\right) \leqslant \\ \lambda\|c^{(k)}\|_1 + \frac{\varepsilon}{2}\|c^{(k)}\|_2^2 - \left(\lambda\left\|c^{(k+1)}\right\|_1 + \frac{\varepsilon}{2}\left\|c^{(k+1)}\right\|_2^2\right)\end{aligned} \tag{5.22}$$

Proof. By exploiting the minimality of $(u^{(k+1)}, c^{(k+1)})$ and the properties of $(u^{(k)}, c^{(k)})$ we have

$$\begin{aligned}&\frac{1}{2}\left\|u^{(k+1)} - f\right\|_2^2 + \lambda\left\|c^{(k+1)}\right\|_1 + \frac{\varepsilon}{2}\left\|c^{(k+1)}\right\|_2^2 \\ &\qquad\leqslant \frac{1}{2}\left\|u^{(k+1)} - f\right\|_2^2 + \lambda\left\|c^{(k+1)}\right\|_1 + \frac{\varepsilon}{2}\left\|c^{(k+1)}\right\|_2^2 + \\ &\qquad\qquad \frac{\mu}{2}\left\|\begin{pmatrix}u^{(k+1)}\\ c^{(k+1)}\end{pmatrix} - \begin{pmatrix}u^{(k)}\\ c^{(k)}\end{pmatrix}\right\|_2^2 \\ &\qquad\leqslant \frac{1}{2}\|u^{(k)} - f\|_2^2 + \lambda\|c^{(k)}\|_1 + \frac{\varepsilon}{2}\|c^{(k)}\|_2^2 \ .\end{aligned} \tag{5.23}$$

Note that the last estimate is only valid if $(u^{(k)}, c^{(k)})$ represents a feasible pair of variables. If $(u^{(k)}, c^{(k)})$ solve $T(u, c) = 0$, then they also solve the linearised constraints

$$D_u T(u^{(k)}, c^{(k)})(u - u^{(k)}) + D_c T(u^{(k)}, c^{(k)})(u - c^{(k)}) = -\underbrace{T(u^{(k)}, c^{(k)})}_{=0} \ .$$

However, the tuple $(u^{(k+1)}, c^{(k+1)})$ minimises our energy over the set of solutions of these linearised constraints. Therefore, the estimate in Eq. (5.23) follows. Finally, replacing $u^{(k+1)}$ by $\tilde{u}^{(k+1)}$ and reordering the terms yields the sought expression. □

Equation (5.22) yields two interesting results. Let us assume that Eq. (5.22) holds for all iterations k. Since the energy is obviously bounded, it follows from the theorem of Bolzano-Weierstrass that there must exist a convergent subsequence of energy values. Unfortunately, we cannot assert the convergence of the sequence of iterates $((u^{(k)}, c^{(k)}))_k$ themselves. It is also clear that the back projection will in general yield a pair of feasible iterates that may increase the value of the cost function again. As a consequence, we lose the existence of a converging subsequence if we enforce all iterates to be feasible and if the requirement from Eq. (5.22) is violated.

Our previous requirement also allows an interesting interpretation. The left-hand side of Eq. (5.22) can be seen as the loss in accuracy for one iteration step whereas the right-hand side can be considered as the simultaneous gain in sparseness. It follows that the energy must necessarily decrease as long as the gain in sparseness outweighs the loss in precision.

The complicated nature of the task at hand prevents a more rigorous convergence proof without imposing additional restrictions. We remark that an alternative but closely related numerical scheme, for which convergence can be shown, has recently been presented by Ochs et al. [53]. This approach has also been discussed and evaluated by Chen et al. [102].

5.5 Numerical experiments

This section contains an extensive performance benchmark of our optimal control approach. Several things are tested. We discuss the quality of our optimal control model on 1D signals and evaluate the results against the methods discussed in Chapter 4. We also compare the efficiency of the primal and dual solvers for the linearised problem on 2D data sets. The efficiency is measured in terms of run time required to reach an accurate solution. Qualitative comparisons to other methods from the literature are also carried out. Furthermore, straightforward extensions to colour images and image sequences are presented. Finally, we shortly mention a simple heuristic strategy to speed up the computations. While the gain in speed can be significant, it does not come without a certain loss in accuracy. However, for practical purposes this loss is likely to be negligible.

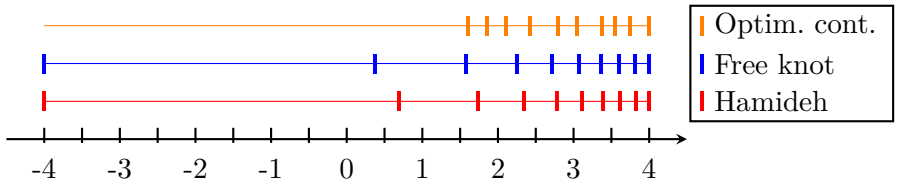


Figure 5.1: Mask distribution obtained by our frameworks for the function $\exp(2x-3)+x$ along $[-4,4]$. Each rectangle represents the position of a mask point. The free knot method and the approach of Hamideh [63] fix the first and last knot at the endpoints of the considered interval. For the optimal control method all knots are free to move. For this experiment the additional back projection in the optimal control algorithm is not performed.

Optimal masks for signals in 1D

In a first example we compare our optimal control model to the methods from Chapter 4. To this end we choose the convex function $x \mapsto \exp(2x-3)+x$ on the interval $[-4,4]$. This function has already been used for the experimental setups in Table 4.1 and Figure 4.5. We consider the reconstruction error for a mask of size 10 with our optimal control method, our optimal interpolation algorithm from Chapter 4 and the method of Hamideh [63]. The methods from Chapter 4 do not have any parameters except for the maximal number of iterations, which we fix at 10 000. This choice is sufficiently large enough to assert convergence. In order to apply our optimal control solver we sample the data function at 128 equidistant positions along the considered interval to obtain a discrete version of the signal. Additionally we use the following parameter choices: $\lambda = 36$, $\mu = 50$, $\varepsilon = 10^{-4}$, and apply 500 outer iterations. The linearised problem is solved with the primal approach with a maximum of 25 000 iterations. These choices are enough to ensure that the final difference between two iterates is smaller than 10^{-15} in norm. The resulting distributions are visualised in Figure 5.1. All methods behave similarly in the sense that the knot density is larger in regions with higher curvature. However, the free knot algorithm and the method of Hamideh [63] from Chapter 4 fix a mask point at each end of the interval. The additional knot at the left end allows much more accurate reconstructions of the long and flat tail. This observation is also reflected in the errors. Algorithm 4.1 yields an L_1 error of 2.17 without tonal optimisation and 0.8 with an additional grey value optimisation. Algorithm 4.2 yields the best reconstructions with

an error of 0.77 (with and without tonal optimisation). A fair comparison with the optimal control model is difficult to achieve since the latter operates exclusively in the discrete setting while the former methods use continuous formulations. We proceed as follows: The non-zero entries of the discrete mask are binarised and used for the construction of a linear spline which interpolates tonal optimised data. The tonal optimisation is carried out with respect to the ℓ_1 norm and done in the same way as in Section 4.3. The final error is 9.62. The fact that the optimal control approach does not return any information on the function within the vicinity of the left interval boundary causes a significant increase in the error. Furthermore, the strategies from Chapter 4 know that the underlying function is strictly convex and differentiable and actively exploit this fact. The optimal control method is more generic and flexible but unable to use any additional knowledge to improve the error.

In a second step we investigate the performance of our optimal control algorithm for arbitrary signals. To demonstrate the benefits of our approach for such settings we choose the piecewise polynomial and non-continuous signal *Piece-Polynomial* from the WAVELAB 850 toolbox [103] and normalise it to the interval $[0, 1]$ to ease the simultaneous visualisation of signal, reconstruction and mask. The result is shown in Figure 5.2. We remark that the obtained mask is sparse and that the non-zero entries are placed at positions where one would naturally expect them, e.g. two mask points are used to encode a step in the signal. Also note the excellent quality of the reconstruction. The mask is computed with the following parameter choices. We initialise our method with u being the original signal and a full mask, i.e. $c_i = 1$ for all i, and set $\varepsilon = 10^{-9}$, $\mu = 1.0$, $\lambda = 0.02$. For Algorithm 5.2 we set $\theta = 1$ and $\tau = 0.25$. In order to fulfil the step length constraint $\tau\sigma L^2 < 1$ where $L = \|\begin{pmatrix} A & B \end{pmatrix}\|$ we approximate L through power iterations and set $\sigma = ((L^2 + 0.1)\tau)^{-1}$. The method aborts when the distance between two consecutive iterates drops below $3 \cdot 10^{-16}$. In order to reach this precision we require about 225 000 iterations of Algorithm 5.2. After 630 iterations of Algorithm 5.3 the distance between two successive versions of c drops below 10^{-15} at which point the iterative scheme stops. The whole approach is implemented in Matlab with Algorithm 5.2 as a mex function written in C. All the tests are done on a standard desktop PC with an Intel Xeon processor clocked at 3.2 GHz and 24 GB of memory. The total run time is

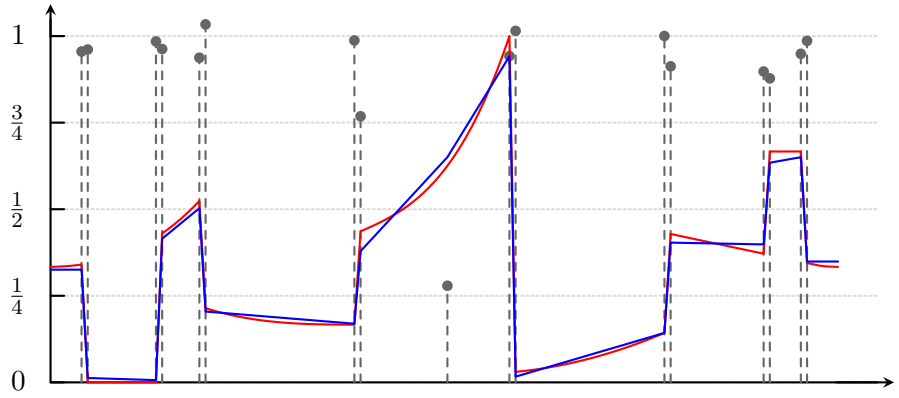


Figure 5.2: The original signal (red line), the reconstruction (blue line) as well as the used mask (grey dots). As expected, the mask is sparse (17 non-zero entries out of 128) and not binary-valued. Some knots even exceed the value 1. The mask point in the middle of the signal with the smallest value allows to better adapt to the curvature of the input signal by blending the diffusion result with the data. Also note that the mask entries neatly align with the discontinuities of the signal. This result is obtained with the parameter settings: $\varepsilon = 10^{-9}$, $\mu = 1.0$, $\lambda = 0.02$. For this experiment the additional back projection is always performed.

roughly ten minutes. The squared Euclidean distance between the input signal and the reconstruction is 0.0377.

Optimal masks for grey scale images

To show that our approach performs as well on 2D data sets as it does on 1D signals we apply our algorithm to three different test images and compare our method to the state-of-the-art approach from [20]. In [20] the authors propose a greedy method, called stochastic sparsification, that iteratively selects a set of candidate points and discards those pixels that yield the smallest increase in the error when removed from the mask. This step is repeated until a desired density is reached. In a second step, called non-local pixel exchange, random mask and non-mask pixels are swapped. If the reconstruction error increases, the swap is undone, otherwise it is kept. This latter step is repeated until the desired error or the maximal number of swaps is reached. The method of Mainberger et al. [20] is capable of reaching a global minimum if a sufficiently large number of iterations is carried out.

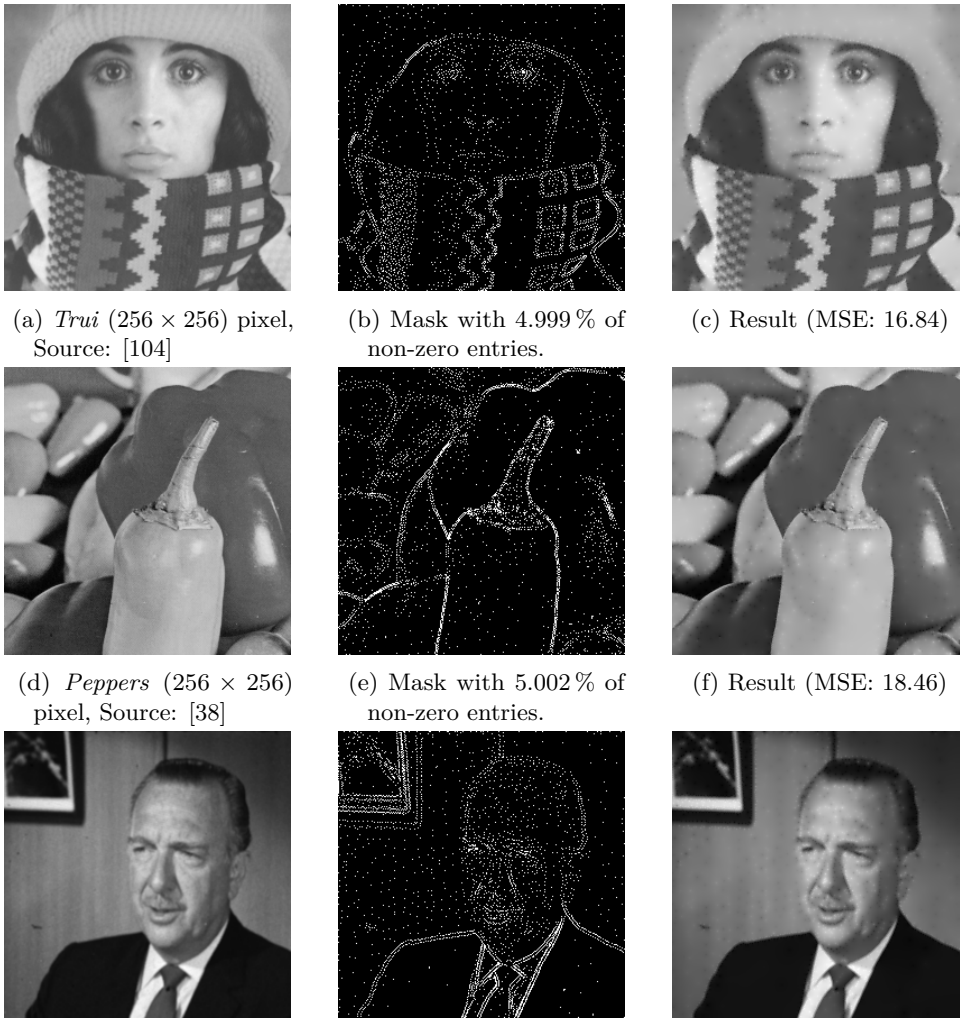
Unfortunately, this approach is also very time consuming.

The results of our method are depicted in Figure 5.3 and a summary with a comparison to the approach from [20] is given in Table 5.1. As an error measure we use the MSE which is computed by

$$\mathrm{MSE}\,(f,u) \coloneqq \frac{1}{n}\sum_{i=1}^{n}(f_i - u_i)^2$$

for an image with n linearly indexed pixels. For the computation of the MSE we assume that the image values lie in the interval $[0, 255]$.

Our optimal control solver is remarkably stable with respect to parameter choices. Only the parameter λ, which is responsible for the sparsity of the mask, needs specific tuning for each image. All other parameters can be set to sane default values for almost all test suites. We opted for the following parameter settings: All experiments use as initialisation a full mask and the complete image data. We set $\mu = 0.1$ and $\varepsilon = 10^{-7}$. The linearised problem is always solved with a gradient descent scheme applied to the dual model. The step size for each descent step is optimised via a simple line search method and a maximum of 10 000 iterations. If the increment drops below 10^{-9} in norm before reaching the maximal number of iterations, the method aborts. Finally, we always used at most 300 linearisations. The reconstructions and error measures presented in Figure 5.3 have been obtained by binarising the mask and applying the LSQR based GVO method from Algorithm 3.1. The comparisons in Table 5.1 to the methods from Mainberger et al. [20], Ochs et al. [53], and Chen et al. [102] demonstrate the state-of-the-art performance of our approach. If combined with a grey value optimisation, our binarised mask outperforms all other approaches in terms of reconstruction quality. The errors obtained from the continuous masks yielded by our optimal control solver are often slightly higher than those obtained by a binarisation with a subsequent GVO and indicated in Table 5.1. However, this difference is usually in the range of 0.1 never larger than 0.2. It probably stems from the fact that there exists a more rigorous convergence guarantee for the GVO method than for our OC strategy. Further, the LSQR algorithm used for the tonal optimisation in all our experiments is known to perform extremely well in ill-posed situations and likely capable of returning superior results when compared to our algorithms. Also note that the mask density in [20] is exactly 5 % for each image. For our experiments we have



(a) *Trui* (256 × 256) pixel, Source: [104]
(b) Mask with 4.999 % of non-zero entries.
(c) Result (MSE: 16.84)
(d) *Peppers* (256 × 256) pixel, Source: [38]
(e) Mask with 5.002 % of non-zero entries.
(f) Result (MSE: 18.46)
(g) *Walter* (256 × 256) pixel, Source: [38]
(h) Mask with 5.002 % of non-zero entries
(i) Result (MSE: 7.70)

Figure 5.3: Results for three test images and a target density of 5 %. All masks are sparse and yield a remarkable reconstruction quality. Note that the bright spots visible in the reconstruction are an artefact stemming from the fact that the Laplace operator is used. We set λ to 3.7×10^{-3} for *Trui*, 3.26×10^{-3} for *Peppers*, and 1.75×10^{-3} for *Walter*. For these experiments the additional back projection is not performed.

Table 5.1: MSE for our experimental results. The best results for each image are marked in boldface. Unavailable results are marked with a —. The column c_B denotes the error for the binarised mask and c_B (GVO) the result for the binarised mask with additional tonal optimisation. According to the results from Chapter 3, these errors coincide with those obtained by optimal mask values. The errors obtained from the continuous mask yielded by our optimal control solver are slightly higher that those indicated in the table below. This difference is never larger than 0.2 and stems probably from the fact that the GVO is numerically more stable than our OC solver. Also note that the mask densities are not completely identical in the comparisons to the other methods but only very close to each other.

Image	Algorithm	c_B	c_B (GVO)
Trui	Our method	46.96	**16.84**
	Method of [20]	23.21	17.17
	Method of [53, 102]	—	16.89
Peppers	Our method	30.64	**18.46**
	Method of [20]	—	19.38
	Method of [53, 102]	—	18.99
Walter	Our method	21.20	**7.70**
	Method of [20]	—	8.14
	Method of [53, 102]	—	8.03

a density of 4.999 % for *Trui*, 5.002 % for *Peppers*, and 5.002 % for *Walter*. The results from [102] have a density of 4.98 % for *Trui*, 4.84 % for *Peppers*, and 4.82 % for *Walter*.

Performance comparison between the primal approach and the dual approach

In this section we demonstrate the strengths and weaknesses of the primal and the dual solver for the linearised formulation. As we have seen in the previous sections, the dual approach yields a much simpler optimisation problem which can easily be handled through a gradient descent scheme. In order to evaluate the performance of both strategies we consider the *Trui* test image. We set $\lambda = 0.001$, $\mu = 0.1$, $\varepsilon = 10^{-9}$. Our goal is to examine

Table 5.2: Performance comparison between the primal and dual method with same stopping criteria on the iterates for the *Trui* test image. For each method, the first result represents the absolute residual of the constraint and the second depicts the relative residual. The dual method yields consistently the higher accuracy and converges in the initial phase much faster. The primal method exhibits surprisingly constant run times.

It.	Primal formulation			Dual formulation		
	Residuals (abs./rel.)		Time	Residuals (abs./rel.)		Time
10	$2.0 \cdot 10^{-7}$	$1.5 \cdot 10^{-9}$	5.41	$1.5 \cdot 10^{-9}$	$1.2 \cdot 10^{-11}$	0.51
15	$2.6 \cdot 10^{-7}$	$2.1 \cdot 10^{-9}$	5.03	$1.4 \cdot 10^{-9}$	$1.2 \cdot 10^{-11}$	0.61
20	$3.4 \cdot 10^{-7}$	$2.9 \cdot 10^{-9}$	5.04	$2.9 \cdot 10^{-9}$	$2.5 \cdot 10^{-11}$	0.87
25	$4.4 \cdot 10^{-7}$	$4.0 \cdot 10^{-9}$	5.16	$4.5 \cdot 10^{-9}$	$4.2 \cdot 10^{-11}$	1.24
35	$7.4 \cdot 10^{-7}$	$7.9 \cdot 10^{-9}$	5.07	$1.1 \cdot 10^{-8}$	$1.2 \cdot 10^{-11}$	2.77
45	$1.3 \cdot 10^{-6}$	$1.6 \cdot 10^{-8}$	5.17	$2.0 \cdot 10^{-8}$	$2.5 \cdot 10^{-10}$	5.40
55	$3.6 \cdot 10^{-6}$	$4.8 \cdot 10^{-8}$	5.81	$3.8 \cdot 10^{-8}$	$5.7 \cdot 10^{-10}$	11.15
65	$7.9 \cdot 10^{-6}$	$1.4 \cdot 10^{-7}$	13.03	$6.9 \cdot 10^{-8}$	$1.2 \cdot 10^{-9}$	29.52
75	$1.4 \cdot 10^{-5}$	$3.0 \cdot 10^{-7}$	37.13	$1.2 \cdot 10^{-7}$	$2.5 \cdot 10^{-9}$	103.32

the speed and accuracy of both methods for each linearised problem during a single run of our optimal control solver. The primal method stops when either the increment in u or c drops in norm below 10^{-9}. For the gradient descent scheme we apply the same stopping criteria on our iterates, too. Further, its step length is optimised for each linearised problem but kept constant over all iterations. This optimisation is done by testing in advance a range of potential step sizes for their convergence behaviour. In our tests a complete optimisation of the step length in each iteration for the gradient descent has proven to be too expensive in terms of run time. Table 5.2 exhibits the run times and residuals for some of the iterates. We measure both the absolute residual $\|Au + Bc - g\|_2$ as well as the relative residual $\|Au + Bc - g\|_2 \|g\|_2^{-1}$. The energies of both approaches are identical for each presented case. It follows that the difference in performance is only due to the accuracy in solving the constraint. As we can see from Table 5.2, the dual method yields more accurate solutions and is significantly faster in

the beginning. Towards the end, the step length must be reduced to ensure convergence. This causes a considerable slow down of the approach. While it is possible to set the step length to 0.4 in the beginning, it drops below 0.1 for the final tests. At the same time the required number of iterations increases by more than a factor 10. The run time of the primal method is almost identical for every iterate but it cannot offer the accuracy of the gradient descent scheme. Nevertheless, the primal solver has significantly lower run times for later iterations.

The results from Table 5.2 suggest that it is beneficial to use the dual solver for early iterations and switch to the primal approach for minimising later linearised problems if speed is an issue. Unfortunately, there is no concise way to predict the moment for switching from one model to the other. Finally, let us mention that the ultimate masks obtained from both approaches were identical.

Extension to colour images and videos

Several approaches can be used to handle colour valued images. A straightforward method would be to apply the algorithm channel wise in the RGB colour space. This yields a different interpolation mask for each channel. Alternatively, one could convert the image first to the YCbCr space and determine the optimal reconstruction points for the Y channel only. The Cb and Cr channels could then be subsampled with the same interpolation mask. The latter strategy has several benefits. First of all, it is significantly faster since we only need to compute a single mask instead of three. Secondly, in view of applications for image compression, we benefit from the fact that we only have to save the positions for a single mask. The Cb and Cr channels do not require highly accurate reconstructions. A fact which is also exploited in many other image compression techniques. Therefore, using the mask from the Y channel for all reconstructions is a viable choice in this approach. Finally, a channel wise computation in the RGB space can lead to visually unpleasant artefacts when the mask information from the individual channels is inconsistent.

Figure 5.4 depicts an example of an optimisation within the YCbCr space. We use the parameter settings $\lambda = 5 \cdot 10^{-3}$, $\varepsilon = 10^{-6}$, $\mu = 0.1$, 50 000 iterations for solving the linearised primal problem and 500 linearisations. The mask has a density of 3.865 % and yields an extraordinary high quality

(a) Input image, Source: [38] (b) Binary Mask (c) Reconstruction

Figure 5.4: Test image *Peppers* in colour (511×511): The reconstruction is done with an additional GVO on the binarised mask and presents a remarkable quality.

reconstruction. The MSE in the Y channel is 24.343.

Our framework can be applied in a straightforward manner to image sequences as well. Instead of working in $\mathbb{R}^2$, we have to switch over to $\mathbb{R}^3$. No significant changes need to be done. It suffices to replace the 2D Laplacian with its 3D variant. Many video compression approaches work frame by frame or over groups of frames. Our approach considers the sequence in its entirety when localising the best interpolation data. Thus, the algorithm is capable of handling almost static sequences as well as sudden scene changes by adapting the mask density in the spatio-temporal domain of the video data. The unfortunate downside of such an approach is the prohibitive amount of data that has to be optimised simultaneously. The complete sequence is optimised as a whole. Such a strategy requires large amounts of memory and computing capacity. Computing a mask for a sequence of 5 frames with size 640×480 pixels requires up to 90 hours on standard desktop PC with an Intel Xeon processor clocked at 3.2 GHz and 24 GB of memory.

A greedy speed-up strategy

The run times for the previous experiments are often in the range of several hours. The methods of Mainberger et al. [20] take similar amounts of time, whereas the approaches of Ochs et al. [53] and Chen et al. [102] are faster. Our experiments show that the time consuming back projection can be omitted. The final results are always very similar. For practical purposes one can

also significantly reduce the number of iterations in Algorithm 5.2 without encountering a notable loss in quality. Moreover, the results from Chapter 3 suggest an interesting heuristic to further speed up the computation of the mask. Our algorithm usually starts with a full mask which is made gradually sparser. Since the smallest errors can be obtained with a binarised mask combined with a GVO it is not really necessary to know the exact optimal value of the mask at every position. All we need to know is the sparsity pattern of the mask. Therefore, one can threshold the mask during the iterations and check whether this thresholded version differs significantly from the thresholded mask of the previous iterate. If not, one aborts the iteration. Using this heuristic for the *Peppers* image with $\lambda = 3.25 \times 10^{-3}$, $\mu = 0.01$, $\varepsilon = 10^{-9}$, 1250 iterations of Algorithm 5.2 and 50 outer iterations we obtain a binary mask with a density of 5.01 %. In combination with GVO, the MSE is 19.38, which is identical to the result from [20]. The total run time is 272 seconds. Even though the obtained mask yields a slightly larger error when compared to the results from Table 5.1, the run time is reduced from 15 hours down to less than 5 minutes. This corresponds to a speed-up factor of almost 200.

5.6 Conclusion

We have analysed an optimal control based approach to find good masks for inpainting with the Laplacian. The method is flexible, easy to implement, and can be applied to arbitrary signals. We have derived a solution strategy and discussed optimality conditions and convergence behaviour of our iterative scheme. The convergence analysis yields a similar situation as in Chapter 4. We can assert a rather well-behaved evolution of the iterates but are unable to show convergence towards a global minimum. Duality concepts have also been discussed in this context. They offer valuable insight into the underlying task. The experimental findings confirm our expectations. The algorithm excels for arbitrary non-convex signals and outperforms other state-of-the-art methods. However, it cannot compete with specialised methods that exploit additional information such as the convexity of the underlying signal. Nevertheless, we believe that our method can serve as a solid base in the design of a novel image compression codec.

Chapter 6

Applications to image compression

> Happiness lies in the joy of achievement and the thrill of creative effort.
>
> *(Franklin D. Roosevelt)*

In this chapter we briefly present a lossy image compression codec based on the results from this work. The codec has been completely suggested by Peter [39] and is capable of compressing single grey scale images. It outperforms both JPEG and sometimes even its successor JPEG 2000 in terms of reconstruction quality. Unfortunately, our strategy is comparatively slow. It serves as a viable alternative to highly popular methods if speed is not an issue. Let us also remark that there exists already a certain number of PDE based codecs such as those of Galić et al. [17], Schmaltz et al. [18], and Mainberger et al. [19]. The main difference to these approaches lies in the fact that we use homogeneous diffusion inpainting for the reconstruction and that our mask is determined by a powerful optimisation scheme.

6.1 The strategy

We proceed as follows: For a given image we first determine a mask by applying the optimal control framework from Chapter 5. In view of the findings from Chapter 3 it suffices to store a binary mask. The positions of the non-zero entries are efficiently saved by using a block encoding scheme. Next we complement this mask by a set of tonal values. In order to increase the compression rate we extend the GVO strategies from Chapter 3 with a few additional steps. Storing the optimal colour values in full precision is

prohibitively expensive. Instead, we quantise them and perform a second discrete optimisation on these quantised values. This strategy yields sub-optimal data but the gain in the compression ratio outweighs the loss of accuracy. Finally, the whole data is stored in a container file and compressed with a state-of-the-art lossless compression algorithm. Let us also remark that our strategy has two steps where unrecoverable loss of information occurs. Partial differential equation based inpainting is never a perfect reconstruction method. Thus, the optimal control algorithm for determining the mask is the first lossy step. Secondly, the quantisation of the tonal values is also non-reversible. The full details to the individual steps are now given below.

Mask and data encoding

Our experiments suggest that most specialised entropy encoders such as those presented in [105–107] are not capable of improving significantly enough on simpler approaches to justify their use. Therefore, we propose to use a simple, fixed-size block coding algorithm and combine it with the high performing lossless compression algorithm PAQ developed by Mahoney [108, 109]. We divide each side of the image into b parts. This results in a decomposition of the image into b^2 blocks, all having the same aspect ratio as the original image. For every block that contains only zeroes, just a 0 is stored. For the remaining blocks we store a 1 followed by the row-wise linearised content of the block. The corresponding grey values are stored in byte precision on a regular grid.

Tonal optimisation

The GVO algorithms described in Chapter 3 yield optimal results for grey values in a continuous range. However, in order to achieve competitive file sizes the number of admissible grey values must be reduced. This so called quantisation step is done by sampling the complete grey value range at q equidistant points. A naive application of this sampling step after a GVO leads to a large degradation in quality. We counter this loss by using the quantisation aware brightness optimisation technique of Schmaltz et al. [18]. This iterative approach traverses all pixels in random order. For each grey value the effect of a change to the next lower or higher quantised colour on

the error is considered. Sequentially, each pixel is assigned the best grey value. This procedure is repeated until convergence. In our experiments we choose a grey value range of $[0, 255]$ and specify a MSE change of 0.001 between the results of two subsequent iterations as a convergence criterion. Let us also emphasise that this strategy is very time consuming. We have to perform an inpainting for every single change in a pixel value.

The quantisation parameter q is independent of the number of mask points but it still influences the overall file size. The entropy coding of the grey values becomes more efficient for smaller numbers of distinct colours. Thus, the file size is directly proportional to q. Simultaneously, the error increases with decreasing q. The fewer colours we allow the more misrepresentations occur. It follows that a suitable parameter q must be found that offers the best trade-off between file size and reconstruction quality. To this end we define the quantity

$$\frac{s\left(v_{\max}\right) - s\left(q\right)}{s\left(v_{\max}\right)} - \frac{e\left(q\right) - e\left(v_{\max}\right)}{e\left(v_{\max}\right)} \quad . \tag{6.1}$$

Here, $s\colon \{1, 2, \ldots, v_{\max}\} \to \mathbb{N}$ is the file size in bytes for a given quantisation level and $e\colon \{1, 2, \ldots, v_{\max}\} \to \mathbb{R}$ the corresponding MSE. The integer $v_{\max}$ is the highest possible number of distinct grey values. Equation (6.1) represents the difference between the relative file size decrease and the relative error increase. The larger this number, the better the trade-off. Maximising Eq. (6.1) returns the best quantisation level for a given mask. The parameter q should also be optimised with the subsequent brightness optimisation in mind. To this end one should in fact perform the quantisation aware brightness optimisation technique of Schmaltz et al. [18] for every valid q and finally use Eq. (6.1) as criteria to select the best number of distinct colours. However, this approach requires a massive computational workload. As a heuristic to cut down the run time we suggest to compute a suitable approximation to the brightness optimised result by changing the MSE threshold as a stopping criterion. In our experiments, 0.1 yields a good compromise between speed and accuracy and allows us to perform the necessary computations within a few hours.

The complete codec

Combining all steps leads to a compression algorithm that consists of the following five steps.

1. Computation of good inpainting data with the approaches from Chapter 5
2. Block coding of the binary mask from the previous point
3. Quantisation optimisation to determine the best q by maximising Eq. (6.1)
4. Tonal optimisation with the GVO algorithms from Chapter 3 and the brightness optimisation technique of Schmaltz et al. [18]
5. Container compression with the lossless compression scheme PAQ

The final compressed file is obtained by storing first the dimension of the image with a variable size of up to two byte per dimension, a 1 bit flag that indicates if the dimensions are encoded using one or two bytes and a 1 bit flag for quadratic images, where the dimension is only stored once. Additionally, we store the number of blocks b and the quantisation parameter q as one byte each. This header is followed by the block-encoded mask information and finally, the quantised grey values are appended. The whole binary file is then stored in a PAQ container. Decompression is straightforward and simply done in reverse order. The final image is obtained by solving the inpainting equation with the extracted mask and grey values.

Let us also remark that if our mask is full, then it suffices to store the complete original image data since no inpainting and GVO needs to be done. The reconstruction is always perfect. In this setting the only compression stems from the PAQ encoding and our codec becomes a lossless method.

Finally note that our choice of storing q with a single byte also restricts us to a byte wise coding of the pixel values. The value of $v_{\max}$ cannot exceed 256.

6.2 Numerical experiments

We test our codec on the grey scale version of the *Peppers* image from Figure 5.3. We save it with different compression ratios and consider the

MSE between the original and the compressed result. The performance of the open source JPEG and JPEG 2000 encoders from the Image Magick suite [110] in version 6.8.3-6 serve as a reference benchmark. The outcome is depicted in Figure 6.1. The corresponding images for a compression ratio of 13.6 to 1 are also visualised in Figure 6.2. Other data sets yield similar results.

As we can see in Figure 6.1, we outperform JPEG for every compression ratio by a significant margin. For certain very low compression ratios we even outperform the state-of-the-art JPEG 2000 standard. Nevertheless, this quality gain does not come without a price. Encoding an image with JPEG takes a few hundred milliseconds to complete. Our mask optimisation scheme alone has an average run time in the range of several hours. Another disadvantage of our method is the tuning of the parameters. Both JPEG and JPEG 2000 allow the specification of a target quality, respectively compression ratio. Even though the mask density correlates in our codec with the final file size, it is still difficult to predict good parameter values to achieve a desired compression ratio. Furthermore, different combinations of mask density and quantisation may lead to the same compression ratio but with different reconstruction errors. It is not possible in advance to state which choice might yield the better outcome. Finally, our approach fails when the image contains large textured regions. In order to reconstruct highly oscillating patterns the strong smoothing effect of the Laplacian during the inpainting has to be countered with a significant amount of mask points. This causes an overall increase in the file size. The JPEG codec is much better suited to handle such ill posed situations by acting on the coefficients of the discrete cosine transform. The JPEG 2000 method uses wavelet decompositions and has similar advantages in the presence of structured patterns. Potential remedies to improve the handling of textures with our codec could be based on the incorporation of patch-based strategies. These approaches usually store small patches of a given image and use them for the reconstruction process. The recent work of Facciolo et al. [111] demonstrates that high quality reconstructions are possible with very sparse data patches. We refer to [112, 113] and the references therein for more information on exemplar based inpainting schemes.

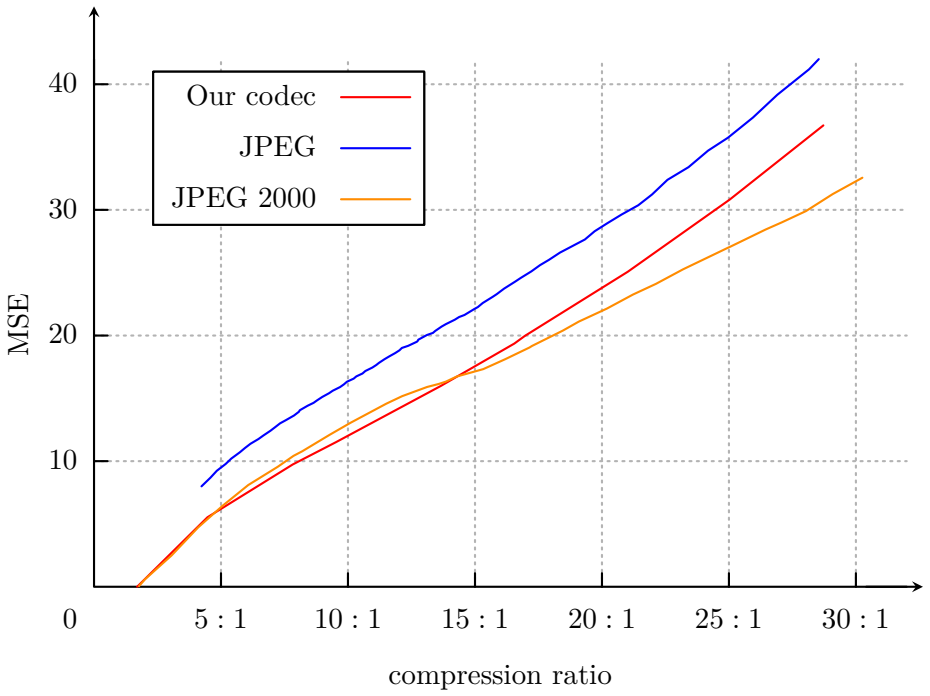
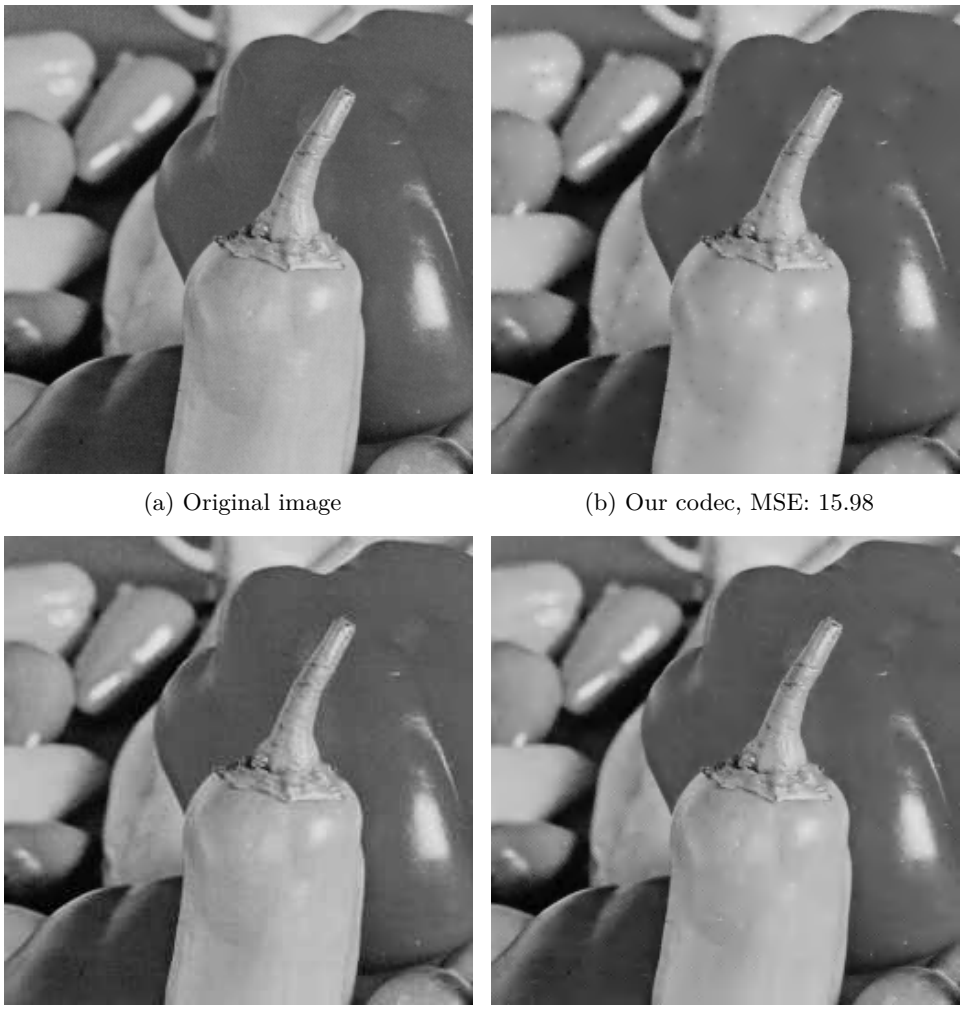


Figure 6.1: Comparison between our image compression codec, JPEG and JPEG 2000 for the grey scale version of the *Peppers* image. We consistently outperform the JPEG standard. For certain compression ratios between $5:1$ and $15:1$ we even outperform the JPEG 2000 codec. Also note that the JPEG format is always lossy. Even with parameters set to maximise the quality, loss of information occurs. Therefore, the JPEG curve cannot be extended to the left as far as for the other two codecs. The lowest compression ratio for our codec is achieved with a full mask. In that case the error is indeed 0 as we can simply store the original data and exploit the performance of the lossless PAQ compression.



(a) Original image

(b) Our codec, MSE: 15.98

(c) JPEG, MSE: 20.71

(d) JPEG 2000, MSE: 16.14

Figure 6.2: Example images for the compression ratio 13.6 to 1. Visual differences are barely visible. The JPEG image depicts some ringing artefacts whereas in our codec the mask points are still apparent. Also the JPEG 2000 image contains a certain number of very small singularities.

6.3 Conclusion

Image compression schemes based on simple partial differential equations can outperform well established and sophisticated codecs that are widely used today. We achieve qualitative better results at the expense of a significantly larger computational burden. Another deficiency of our codec lies in the predictability of the parameter influences. Currently, trial and error strategies are the only way to achieve accurate target quality settings. If run time is not an issue and if the data does not contain too many textures, then our codec presents a viable alternative to prevailing methods. Future improvements and extensions of the codec should include an improved handling of textures and the ability to compress colour images and video sequences.

Chapter 7

Summary and outlook

> The whole point of getting things done is knowing what to leave undone.
>
> *(Oswald Chambers)*

In this work we have presented several new findings concerning Laplace interpolation for image reconstructions with extremely sparse data. We have generalised the inpainting PDE with homogeneous diffusion from an original setting with binary masks to arbitrarily valued masks. We have shown that a well suited choice for the range of the mask values is the interval $[0, 1]$. It asserts that the inpainting matrix is invertible, that all its eigenvalues are real, and that adherence to a max-min principle is guaranteed. Our findings extend previously well known results from [19] to a more generic setting.

The formulation with continuously valued masks allows us to state a new control model for the determination of optimal interpolation data. Our optimal control based model from Chapter 5 is able to process arbitrary signals and, in combination with the tonal optimisation findings from Chapter 3, it is possible to achieve extraordinary image reconstructions with as little as 5 % of the pixels. Currently, these results outperform all other competing strategies in terms of quality. An alternative approach for one dimensional and strictly convex signals is also considered in Chapter 4. Unfortunately, this latter framework is not flexible enough to be used for our purposes. Nevertheless it offers some valuable insight into the difficulties behind the underlying problem. Besides these improvements in the modelling we have also found new numerical schemes that allow an even more efficient handling of the optimisation tasks at hand. Our algorithms outperform previous

attempts in terms of accuracy and speed. Our grey value optimisation algorithms from Chapter 3 also cover the most frequently used computing environments. We have provided methods that work well on CPUs and that exploit the massive parallelism offered by modern GPUs. Finally, the results from Chapter 6 prove the practical applicability of the findings from this thesis to image compression tasks. In certain cases we can outperform popular codecs such as JPEG and JPEG 2000.

From a theoretical point of view almost all important questions are answered. We are able to state criteria that assert that the inpainting PDE is solvable and we have analysed the optimality conditions for the optimisation in the codomain. Our methods for finding good interpolation sites from Chapter 4 and Chapter 5 have been investigated in detail as well. Equivalent formulations in terms of conjugate duality are presented. We also provide conditions that assert that our iterates decrease the cost functions in the optimisation models. Unfortunately we have been unable to prove convergence of our iterates towards a global minimum for the non-convex tasks at hand. Therefore, it remains an open question how good the obtained solutions really are. Experimental setups suggest however that our solutions perform surprisingly well.

While this work concentrates exclusively on the Laplace equation, it is clearly possible to formulate inpainting models with other PDEs, too. Incorporating them into our models requires an approach in similar style as in Chapter 2 and Section 5.2. An extension to other linear operators is straightforward and investigating the properties of the biharmonic operator should be done as a next step. In view of the findings from [102] we expect the biharmonic operator to yield more accurate reconstructions and to be more difficult to handle numerically. On the other hand, the analysis of non-linear PDEs is likely to remain challenging. Many results from this work rely on the linearity of the differential operator. Nevertheless, our work can still serve as a foundation for future research in this domain. Handling non-linear operators could for example be done by lagged linearity approaches.

Even though our numerical schemes are already quite efficient, they are not fast enough for a day-to-day usage. The computation of mask points for video sequences and the compression codec for grey scale images require significant speedups to become feasible alternatives. Currently the best performing codecs require fractions of a second to compress large images. Our methods require at least several hours for the same task. High performing algorithms

would also facilitate the tuning of the parameters to achieve desired mask densities and compression ratios. Another topic for future research in this domain would be the handling of colour images and the integration of a more efficient texture processing. Colour images contain a certain amount of redundant information in their individual channels. A fact that must be taken into consideration. A first tentative attempt has been suggested in Section 5.5. The difficulty in handling textured data stems from the usage of the Laplacian. There exist two ways to overcome the deficiencies. Either we replace the Laplacian with another differential operator that is better suited to handle texture or we combine our findings from this work with patch based strategies as suggested in Chapter 6. The latter ones are known to work very well with regular and non-smooth patterns. Finally, another important and outstanding topic is the global convergence and optimality of our approaches. Experiments show that our solutions yield state-of-the-art results but so far we cannot claim convergence towards a global minimum in any of our methods. The non-convexity of our models prevents us from overcoming this obstacle.

All in all we conclude that this thesis has contributed a number of fundamental results to the research on image inpainting and the accurate determination of good reconstruction data sets. These findings will certainly allow us in the future to perform significant advances in the domain of PDE-based image and video compression.

List of symbols

The following nomenclature is used throughout the whole document. Functions, scalar- and vector-valued unknowns are denoted by lowercase roman letters. The individual entries of a vector are marked with a single subscript index. If f is a vector, then its entries are given by f_i. Matrices and other operators use capital roman letters. Matrix entries are also denoted by a lowercase roman letter and a double subscript index. Thus a matrix A has the entries $a_{i,j}$. Alternatively we may also address a matrix A by writing $(a_{i,j})_{i,j}$. Submatrices use a capital roman letter with a double subscript index. It follows that a submatrix of the matrix A can be referenced by $A_{i,j}$. Sequences are enclosed in parentheses and have a running subscript index appended. This index is also added as a superscript with parentheses to the individual elements. A sequence of scalars $x^{(i)}$ is therefore written as $(x^{(i)})_i$. Parameters are stated in lowercase Greek letters whereas sets use uppercase Greek letters. Further notations and all exceptions deviating from this convention are designated below.

Notation	**Description**
$\equiv$	Identical equality: $f(x) \equiv \alpha \Leftrightarrow f(x) = \alpha \ \forall x$
$\succcurlyeq$	Component wise larger than or equal
$\preccurlyeq$	Component wise less than or equal
$\gg$	Significantly larger than
$A\,(c)$	Inpainting matrix with mask c
$B^\dagger$	Moore Penrose pseudoinverse of the matrix B
$\mathbb{C}$	Set of complex numbers
$\mathbb{C}^{n,n}$	Set of $n \times n$ matrices with complex entries
$\partial\Omega$	Boundary of the set Ω

Notation	**Description**
$\operatorname{diag}(c)$	Diagonal matrix with the vector c on its main diagonal
$D_c T(\bar{u}, \bar{c})$	Jacobi matrix of $T(u, c)$ with respect to c evaluated at $(\bar{u}, \bar{c})$
$D_u T(\bar{u}, \bar{c})$	Jacobi matrix of $T(u, c)$ with respect to u evaluated at $(\bar{u}, \bar{c})$
$\delta_{k,l}$	Kronecker delta function
Δ	Laplace operator
∂_n	Derivative in outer normal direction
$\partial(F)(x)$	Subgradient of F at position x
f^*	Conjugate of the function f
f^{**}	Biconjugate of the function f
$f(x)\Big\vert_{x=z}$	$f(x)$ evaluated at $x = z$
$\mathbb{G}(A)$	Directed graph corresponding to the matrix A
H_γ	Huber loss function with parameter γ
$\imath$	Complex unit
I	Identity matrix
$i(A)$	Inertia of the matrix A
$i_-(A)$	Number of eigenvalues A with negative real part
$i_0(A)$	Number of eigenvalues A with zero real part
$i_+(A)$	Number of eigenvalues A with positive real part
ι_S	Characteristic function of the set S
$J^\top$	Transpose of the linear operator J.
L	discrete Laplace operator
L_1	Space of integrable functions
L_2	Space of square integrable functions
L_p	Space of p-integrable functions
$M(c)$	Reconstruction matrix with mask c
∇	Gradient operator

Notation	**Description**
$\nabla_c E(c,x)$	Gradient of $E(c,x)$ with respect to c.
$\nabla_x E(c,x)$	Gradient of $E(c,x)$ with respect to x.
$\mathbb{N}$	Set of natural numbers
$\lvert N(i)\rvert$	Number of direct neighbours of pixel i
$N(i)$	Set of direct neighbourhood pixels of pixel i
Ω	Image domain
Ω_K	Set of known data locations
$\mathcal{O}(u^*)$	Open neighbourhood around u^*
$\operatorname{ran}(J)$	Range of the operator J
$\operatorname{ran}(J)^\perp$	Orthogonal complement of $\operatorname{ran}(J)$
$\mathbb{R}$	Set of real numbers
$\overline{\mathbb{R}}$	Extended real line: $\mathbb{R} \cup \{-\infty, \infty\}$
$\mathbb{R}^{n,n}$	Set of $n \times n$ matrices with real entries
Z_n	Set of all real valued $n \times n$ matrices with non-positive elements at each position not on the main-diagonal

List of abbreviations

Notation	Description
CCD	charge coupled device
CPU	central processing unit
EED	edge enhancing diffusion
FK	free knot
GPU	graphics processing unit
GVO	grey value optimisation
JPEG	Joint Photographic Experts Group
KKT	Karush Kuhn Tucker
LCL	linearly constrained Lagrangian methods
LSQR	least squares algorithm
MM	majorise/minimise methods
MP3	MPEG Audio Layer III
MSE	mean squared error
OC	optimal control
PDE	partial differential equation
PDHGMu	modified primal dual hybrid gradient
SLP	sequential linear programming

Index

References

[1] E. Meijering. „A Chronology of Interpolation: From Ancient Astronomy to Modern Signal and Image Processing“. *Proceedings of the IEEE* 90(3), 2002, pp. 319–342.

[2] S. Masnou and J.-M. Morel. „Level Lines Based Disocclusion“. *Proc. 1998 IEEE International Conference on Image Processing*. Vol. 3. 1998, pp. 259–263.

[3] M. Bertalmío, G. Sapiro, V. Caselles, and C. Ballester. „Image Inpainting“. *Proc. 27th Annual Conference on Computer Graphics and Interactive Techniques*. ACM Press/Addison-Wesley Publishing Company, 2000, pp. 417–424.

[4] M. Nitzberg, D. Mumford, and T. Shiota. *Filtering, Segmentation and Depth*. Vol. 662. Lecture Notes in Computer Science. Springer, 1993.

[5] R. Levin. *The Elastica: A Mathematical History*. Technical Report UCB/EECS-2008-103. EECS Department, University of California, Berkeley, 2008.

[6] M. Bertalmío, V. Caselles, S. Masnou, and G. Sapiro. „Inpainting“. *Computer Vision. A reference guide*. Ed. by K. Ikeuchi. Springer, 2011, pp. 401–416.

[7] C. Ballester, V. Caselles, J. Verdera, M. Bertalmío, and G. Sapiro. „A Variational Model for Filling-in Gray Level and Color Images“. *Computer Vision, 2001. ICCV 2001. Proceedings. Eighth IEEE International Conference on*. Vol. 1. Institute of Electrical and Electronics Engineers, 2001, pp. 10–16.

[8] C. Ballester, M. Bertalmío, V. Caselles, G. Sapiro, and J. Verdera. „Filling-in by joint interpolation of vector fields and gray levels“. *IEEE Transactions on Image Processing* 10(8), 2001, pp. 1200–1211.

[9] C. Ballester, V. Caselles, and J. Verdera. „Disocclusion by Joint Interpolation of Vector Fields and Gray Levels“. *Multiscale Modeling & Simulation* 2(1), 2003, pp. 80–123.

[10] J. Shen, S. H. Kang, and T. F. Chan. „Euler's Elastica and Curvature-based Inpainting“. *SIAM Journal on Applied Mathematics* 63(2), 2002, pp. 564–592.

[11] T. F. Chan and J. Shen. *Mathematical Models for Local Deterministic Inpaintings*. Technical Report TR 00-11. University of California, Los Angeles, Department of Mathematics, 2000.

[12] T. F. Chan and J. Shen. „Non-texture Inpainting by Curvature-driven Diffusions (CDD)“. *Journal of Visual Communication and Image Representation* 12(4), 2001, pp. 436–449.

[13] M. Bertalmío. *Image Inpainting*. 2003. `http://www.dtic.upf.edu/~mbertalmio/restoration0.html` (visited on Nov. 14, 2014).

[14] M. Bertalmío, V. Caselles, G. Haro, and G. Sapiro. „PDE-based Image and Surface Inpainting“. *Handbook of Mathematical Models in Computer Vision*. Ed. by N. Paragios, Y. Chen, and O. Faugeras. Springer, 2006. Chap. 3, pp. 33–61.

[15] M. Nilsson, ed. *The audio/mpeg Media Type*. 2000. `http://tools.ietf.org/html/rfc3003` (visited on Nov. 17, 2014).

[16] R. Clark, ed. *JPEG Homepage*. 2014. `http://jpeg.org/jpeg/index.html` (visited on Oct. 30, 2014).

[17] I. Galić, J. Weickert, M. Welk, A. Bruhn, A. Belyaev, and H.-P. Seidel. „Image Compression with Anisotropic Diffusion“. *Journal of Mathematical Imaging and Vision* 31(2-3), 2008: *Special Issue: Tribute to Peter Johansen*, pp. 255–269.

[18] C. Schmaltz, J. Weickert, and A. Bruhn. „Beating the Quality of JPEG 2000 with Anisotropic Diffusion“. *Pattern Recognition*. Ed. by J. Denzler, G. Notni, and H. Süße. Vol. 5748. Lecture Notes in Computer Science. Springer, 2009, pp. 452–461.

[19] M. Mainberger, A. Bruhn, J. Weickert, and S. Forchhammer. „Edge-based Compression of Cartoon-like Images with Homogeneous Diffusion“. *Pattern Recognition* 44(9), 2011. Ed. by C. Y. Suen, pp. 1859–1873.

[20] M. Mainberger, S. Hoffmann, J. Weickert, C. H. Tang, D. Johannsen, F. Neumann, and B. Doerr. „Optimising Spatial and Tonal Data for Homogeneous Diffusion Inpainting“. *Scale Space and Variational Methods in Computer Vision. Proc. Third International Conference.* (Ein-Gedi, Israel, May 29–June 2, 2011). Ed. by A. M. Bruckstein, B. M. ter Haar Romeny, A. M. Bronstein, and M. M. Bronstein. Vol. 6667. Lecture Notes in Computer Science. Springer, 2012, pp. 26–37.

[21] L. Demaret and A. Iske. „Advances in Digital Image Compression by Adaptive Thinning“. *Annals of the Marie Curie Fellowship Association (MCFA)* 3, 2004, pp. 105–109.

[22] L. Demaret, N. Dyn, and A. Iske. „Image Compression by Linear Splines Over Adaptive Triangulations“. *Signal Processing* 86(7), 2006, pp. 1604–1616.

[23] H. Köstler, M. Stürmer, C. Freundl, and U. Rüde. *PDE-based Video Compression in Real Time.* Technical Report 07-11. Friedrich-Alexander-Universität Erlangen-Nürnberg, 2011.

[24] C. Schmaltz. „Compression, Pose Tracking, and Halftoning“. Dissertation. Saarland University, Saarbrücken, Germany, 2012.

[25] J. Weickert. „Theoretical Foundations of Anisotropic Diffusion in Image Processing“. *Computing Supplement* 11, 1996, pp. 221–236.

[26] Z. Belhachmi, D. Bucur, B. Burgeth, and J. Weickert. „How to Choose Interpolation Data in Images“. *SIAM Journal on Applied Mathematics* 70(1), 2009, pp. 333–352.

[27] H. Stone. „Approximation of Curves by Line Segments“. *Mathematics of Computation* 15(73), 1961, pp. 40–47.

[28] H. G. Burchard. „Splines (with optimal knots) are better“. *Applicable Analysis* 3(4), 1974, pp. 309–319.

[29] D. L. B. Jupp. „Approximation to Data by Splines with Free Knots“. *SIAM Journal on Numerical Analysis* 15(2), 1978, pp. 328–343.

[30] C. de Boor. *A Practical Guide to Splines.* 2nd ed. Vol. 27. Applied Mathematical Sciences. Springer, 2001.

[31] M. P. Bendsøe and O. Sigmund. *Topology Optimization.* 2nd ed. Springer, 2003.

[32] G. Allaire. *Conception Optimale de Structures*. Vol. 58. Mathématiques et Applications. Springer, 2007.

[33] J. Haslinger and R. A. E. Mäkinen. *Introduction to Shape Optimization: Theory, Approximation, and Computation*. Society for Industrial and Applied Mathematics, 1987.

[34] J. Sokolowski and J. P. Zolesio. *Introduction to Shape Optimization*. Springer, 1992.

[35] R. Bellman. *Dynamic Programming*. Princeton University Press, 1957.

[36] L. S. Pontryagin, V. G. Boltyanskii, R. V. Gamkrelidze, and E. F. Mishchenko. *The Mathematical Theory of Optimal Processes*. Interscience, 1962.

[37] A. Borzì, K. Ito, and K. Kunisch. „Optimal Control Formulation for Determining Optical Flow“. *SIAM Journal on Scientific Computing* 24(3), 2002, pp. 818–847.

[38] A. Weber, ed. *The USC-SIPI Image Database*. 2014. `http://sipi.usc.edu/database/` (visited on Oct. 30, 2014).

[39] P. Peter. *A General Purpose Image Compression Codec with Homogeneous Diffusion*. Private communication. 2014.

[40] R. Clark, ed. *JPEG 2000 Homepage*. 2014. `http://jpeg.org/jpeg2000/index.html` (visited on Oct. 30, 2014).

[41] S. Zaremba. „Sur un Problème Mixte Relatif à l'Équation de Laplace“. *Bulletin de l'Academie des Sciences de Cracovie*. Classe des Sciences Mathématiques et Naturelles. Série A, 1910, pp. 313–344.

[42] G. Fichera. „Analisi Esistenziale per le Soluzioni dei Problemi al Contorno Misti, Relativi all Equazione e ai Sistemi di Equazioni del Secondo Ordine di Tipo Elliptico, Autoaggiunti“. *Annali della Scuola Normale Superiore di Pisa — Classe di Scienze* 3(1), 1949, pp. 75–100.

[43] C. Miranda. „Sul Problema Misto per le Equazioni Lineari Ellittiche.“ *Annali di Matematica Pura ed Applicata* 39, 1955, pp. 279–303.

[44] C. Miranda. *Partial Differential Equations of Elliptic Type*. 2nd ed. Springer, 1970.

[45] A. Azzam and E. Kreyszig. „On Solutions of Elliptic Equations Satisfying Mixed Boundary Conditions“. *SIAM Journal of Mathematical Analysis* 13(2), 1982, pp. 254–262.

[46] R. Brown. „The Mixed Problem for Laplace's Equation in a Class of Lipschitz Domains“. *Communications in Partial Differential Equations* 19(7-8), 1994, pp. 1217–1233.

[47] R. A. Horn and C. R. Johnson. *Matrix Analysis*. Cambridge University Press, 2010.

[48] R. S. Varga. *Gersgorin and His Circles*. Vol. 36. Springer Series in Computational Mathematics. Springer, 2004.

[49] D. G. Feingold and R. S. Varga. „Block Diagonally Dominant Matrices and Generalisations of the Gerschgorin Circle Theorem“. *Pacific Journal of Mathematics* 12(4), 1962. Ed. by R. S. Phillips, A. L. Whiteman, M. G. Arsove, and L. J. Paige, pp. 1241–1250.

[50] R. A. Horn and C. R. Johnson. *Topics in Matrix Analysis*. Cambridge University Press, 2008.

[51] A. Berman and R. J. Plemmons. *Nonnegative Matrices in the Mathematical Sciences*. Vol. 9. Classics in Applied Mathematics. Society for Industrial and Applied Mathematics, 1994.

[52] L. Hoeltgen and J. Weickert. „Why Does Non-binary Mask Optimisation Work for Diffusion-based Image Compression?“ *Energy Minimization Methods in Computer Vision and Pattern Recognition*. Ed. by X.-C. Tai, E. Bae, T. F. Chan, S. Y. Leung, and M. Lysaker. Lecture Notes in Computer Science. Springer, 2015.

[53] P. Ochs, Y. Chen, T. Brox, and T. Pock. „iPiano: Inertial Proximal Algorithm for Non-convex Optimization“. *SIAM Journal on Imaging Sciences* 7(2), 2014, pp. 1388–1419.

[54] V. Shoup. *A Computational Introduction to Number Theory and Algebra*. 2nd ed. Cambridge University Press, 2008.

[55] C. C. Paige and M. A. Saunders. „LSQR: An Algorithm for Sparse Linear Equations and Sparse Least Squares“. *ACM Transactions on Mathematical Software (TOMS)* 8(1), 1982, pp. 43–71.

[56] C. C. Paige and M. A. Saunders. „Algorithm 583; LSQR: Sparse Linear Equations and Least Squares Problems“. *ACM Transactions on Mathematical Software (TOMS)* 8(2), 1982, pp. 195–209.

[57] G. H. Golub and W. Kahan. „Calculating the Singular Values and Pseudoinverse of a Matrix“. *Journal of the Society for Industrial and Applied Mathematics.* Series B: Numerical Analysis 2(2), 1965, pp. 205–224.

[58] T. A. Davis and I. S. Duff. „A Combined Unifrontal/Multifrontal Method for Unsymmetric Sparse Matrices“. *ACM Transactions on Mathematical Software (TOMS)* 25(1), 1999, pp. 1–19.

[59] T. A. Davis. „Algorithm 832: UMFPACK, an Unsymmetric-pattern Multifrontal Method“. *ACM Transactions on Mathematical Software (TOMS)* 30(2), 2004, pp. 196–199.

[60] T. A. Davis and I. S. Duff. „An Unsymmetric-pattern Multifrontal Method for Sparse LU Factorization“. *SIAM Journal on Matrix Analysis and Applications* 18(1), 1997, pp. 104–158.

[61] A. Chambolle and T. Pock. „A First-order Primal-dual Algorithm for Convex Problems with Applications to Imaging“. *Journal of Mathematical Imaging and Vision* 40(1), 2011, pp. 120–145.

[62] T. Pock and A. Chambolle. „Diagonal Preconditioning for First Order Primal-dual Algorithms in Convex Optimization“. *2011 International Conference on Computer Vision (ICCV 2011).* (Nov. 6–13, 2011). Ed. by D. Metaxas, L. Quan, A. Sanfeliu, and L. V. Gool. Institute of Electrical and Electronics Engineers, 2011, pp. 1762–1769.

[63] H. Hamideh. „On the Optimal Knots of First Degree Splines“. *Kuwait Journal of Science and Engineering* 29(1), 2002, pp. 1–13.

[64] J. B. Kioustelidis and K. J. Spyropoulos. „L_1 Approximations of Strictly Convex Functions by Means of First Degree Splines“. *Computing* 20(1), 1978, pp. 35–45.

[65] C. de Boor. „Good Approximation by Splines with Variable Knots II“. *Conference on the Numerical Solution of Differential Equations.* Ed. by G. Watson. Vol. 363. Lecture Notes in Mathematics. Springer, 1974, pp. 12–20.

[66] R. A. DeVore and V. A. Popov. „Free Multivariate Splines“. *Constructive Approximation* 3, 1987, pp. 239–248.

[67] N. D. Dikoussar and C. Török. „Data Smoothing by Splines with Free Knots“. *Physics of Particles and Nuclei Letters* 5(3), 2008, pp. 324–327.

[68] T. Blu, P. Thévenaz, and M. Unser. „Linear Interpolation Revitalized“. *IEEE Transactions on Image Processing* 13(5), 2004, pp. 710–719.

[69] P. J. Davis and P. Rabinowitz. *Methods of Numerical Integration*. 2nd ed. Computer Science and Applied Mathematics. Academic Press, 1984.

[70] Y. Saad. *Iterative Methods for Sparse Linear Systems*. 2nd ed. Society for Industrial and Applied Mathematics, 2003.

[71] P. J. Davis. *Interpolation and Approximation*. 1st ed. Blaisdell Publishing Co., 1963, p. 393.

[72] G. Nürnberger and D. Braess. „Nonuniqueness of Best L_p Approximation for Generalized Convex Functions by Splines with Free Knots“. *Numerical Functional Analysis and Optimization* 4(2), 1982, pp. 199–209.

[73] M. G. Cox. „An Algorithm for Approximating Convex Functions by Means of First Degree Splines“. *The Computer Journal* 14(3), 1971, pp. 272–275.

[74] G. M. Phillips. „Algorithms for Piecewise Straight Line Approximations“. *The Computer Journal* 11(2), 1968, pp. 211–212.

[75] J. R. Rice. *The Approximation of Functions. Volume 1: Linear Theory*. 1st ed. Addison-Wesley series in computer science and information processing. Addison-Wesley Pub. Co., 1964, p. 206.

[76] L. Chieppa. „Numerical Algorithms for Curve Approximation and Novel User Oriented Interactive Tools“. Phd Thesis. Università degli studi di Bari, 2009.

[77] L. V. Kantorovich. „Mathematical Methods of Organizing and Planning Production“. *Management Science* 6(4), 1960, pp. 366–422.

[78] D. G. Luenberger and Y. Ye. *Linear and Nonlinear Programming*. International Series In Operations Research & Management Science. Springer, 2008.

[79] L. Hoeltgen, S. Setzer, and J. Weickert. „An Optimal Control Approach to Find Sparse Data for Laplace Interpolation“. *Energy Minimization Methods in Computer Vision and Pattern Recognition. 9th International Conference*. Ed. by A. Heyden, F. Kahl, C. Olsson, M. Oskarsson, and X.-C. Tay. Vol. 8081. Lecture Notes in Computer Science. Springer, 2013.

[80] T. Pock, T. Schoenemann, G. Graber, H. Bischof, and D. Cremers. „A Convex Formulation of Continuous Multi-label Problems“. *Computer Vision - ECCV 2008*. Ed. by D. Forsyth, P. Torr, and A. Zisserman. Vol. 5304. LNCS. Springer, 2008, pp. 792–805.

[81] C. Clason and K. Kunisch. „A Duality-based Approach to Elliptic Control Problems in Non-reflexive Banach Spaces“. *ESAIM: Control, Optimisation and Calculus of Variations* 17(1), 2011, pp. 243–266.

[82] G. Stadler. „Elliptic Optimal Control Problems with L_1-Control Cost and Applications for the Placement of Control Devices“. *Computational Optimization and Applications* 44(2), 2009, pp. 159–181.

[83] G. Wachsmuth and D. Wachsmuth. „Convergence and Regularization Results for Optimal Control Problems with Sparsity Functional“. *ESAIM: Control, Optimisation and Calculus of Variations* 17(3), 2011, pp. 858–886.

[84] R. E. Griffith and R. A. Stewart. „A Nonlinear Programming Technique for the Optimization of Continuous Processing Systems“. *Management Science* 7(4), 1961, pp. 379–392.

[85] M. P. Friedlander and M. A. Saunders. „A Globally Convergent Linearly Constrained Lagrangian Method for Nonlinear Optimization“. *SIAM Journal on Optimization* 15(3), 2005, pp. 863–897.

[86] B. A. Murthagh and M. A. Saunders. „A Projected Lagrangian Algorithm and its Implementation for Sparse Nonlinear Constraints“. *Mathematical Programming Study* 16, 1982, pp. 84–117.

[87] S. M. Robinson. „A Quadratically-convergent Algorithm for General Nonlinear Programming Problems“. *Mathematical Programming* 3, 1972, pp. 145–156.

[88] F. Tröltzsch. *Optimale Steuerung partieller Differentialgleichungen*. 2nd ed. Vieweg+Teubner, 2009.

[89] J. M. Orthega and W. C. Rheinboldt. *Iterative Solutions of Nonlinear Equations in Several Variables*. New York Academic, 1970.

[90] C.-J. Lin. „Projected Gradient Methods for Nonnegative Matrix Factorization“. *Neural Computation* 19(10), 2007, pp. 2756–2779.

[91] Y. Xu and W. Yin. *A Block Coordinate Descent Method for Multiconvex Optimization with Applications to Nonnegative Tensor Factorization and Completion*. Rice CAAM Technical Report TR12-15. Rice University, 2012.

[92] Y. Xu, W. Yin, Z. Wen, and Y. Zhang. „An Alternating Direction Algorithm for Matrix Completion with Nonnegative Factors“. *Frontiers of Mathematics in China* 7(2), 2012, pp. 365–384.

[93] J. F. Bonnans and A. Shapiro. *Perturbation Analysis of Optimization Problems*. Springer Series in Operations Research and Financial Engineering. Springer, 2000.

[94] C. Geiger and C. Kanzow. *Theorie und Numerik Restringierter Optimierungsaufgaben*. Mit 140 Übungsaufgaben. Springer-Lehrbuch Masterclass. Springer, 2002.

[95] E. Esser, X. Zhang, and T. F. Chan. „A General Framework for a Class of First Order Primal-dual Algorithms for Convex Optimization in Imaging Science“. *SIAM Journal on Imaging Sciences* 3(4), 2010, pp. 1015–1046.

[96] R. T. Rockafellar. *Convex Analysis*. Princeton Landmarks in Mathematics and Physics. Princeton University Press, 1997.

[97] J.-J. Moreau. „Proximité et Dualité dans un Espace Hilbertien“. *Bulletin de la Société Mathématique de France* 93, 1965, pp. 273–299.

[98] R. Chartrand. „Shrinkage mappings and their induced penalty functions“. *IEEE International Conference on Acoustics, Speech, and Signal Processing*. Ed. by Institute of Electrical and Electronics Engineers. 2014.

[99] R. I. Boţ. *Conjugate Duality in Convex Optimization*. Vol. 637. Lecture Notes in Economics and Mathematical Systems. Springer, 2010.

[100] P. J. Huber. „Robust Estimation of a Location Parameter“. *The Annals of Mathematical Statistics* 35(1), 1964, pp. 73–101.

[101] J. Nocedal and S. J. Wright. *Numerical Optimization*. Springer Series in Operations Research. Springer, 1999.

[102] Y. Chen, R. Ranftl, and T. Pock. „A Bi-level View of Inpainting-based Image Compression“. (Chateau Krtiny, Czech Republic, Feb. 3–5, 2014). Available from: `http://arxiv.org/abs/1401.4112`. 2014.

[103] J. Buckheit, S. S. Chen, D. Donoho, X. Huo, I. Johnstone, O. Levi, J. Scargle, and T. Yu. *WAVELAB 850 Toolbox for MATLAB*. 2012.

[104] C. Solomon and T. Breckon. *Fundamentals of Digital Image Processing*. 2014. `http://breckon.eu/toby/fundipbook/materials/` (visited on Oct. 30, 2014).

[105] P. Fränti and O. Nevalainen. „Compression of Binary Images by Composite Methods Based on Block Coding“. *Journal of Visual Communication and Image Representation* 6(4), 1995, pp. 366–377.

[106] S. A. Mohamed and M. Fahmy. „Binary image compression using efficient partitioning into rectangular regions“. *IEEE Transactions on Communications* 43(5), 1995, pp. 1888–1893.

[107] G. Zeng and N. Ahmed. „A block coding technique for encoding sparse binary patterns“. *IEEE Transactions on Acoustics Speech and Signal Processing* 37(5), 1989, pp. 778–780.

[108] M. Mahoney. *Adaptive Weighing of Context Models for Lossless Data Compression*. Technical Report CS-2005-16. Florida Institute of Technology, Computer Science Department, 2005.

[109] M. Mahoney. *Data Compression Programs*. Overview over PAQ based compression software. `http://mattmahoney.net/dc/` (visited on Oct. 13, 2014).

[110] ImageMagick Studio LLC. *Image Magick Graphics Tools*. 2014. `http://www.imagemagick.org/` (visited on Nov. 25, 2014).

[111] G. Facciolo, P. Arias, V. Caselles, and G. Sapiro. „Exemplar-based interpolation of sparsely sampled images“. *Proceedings of the 7th International Conference on Energy Minimization Methods in Computer Vision and Pattern Recognition*. Ed. by D. Cremers, Y. Boykov, A. Blake, and F. Schmidt. Vol. 5681. Lecture Notes in Computer Science. Springer, 2009, pp. 331–344.

[112] A. Criminisi, P. Pérez, and K. Toyama. „Region Filling and Object Removal by Exemplar-based Image Inpainting“. *IEEE Transactions on Image Processing* 13(9), 2004, pp. 1200–1212.

[113] P. Arias, G. Facciolo, V. Caselles, and G. Sapiro. „A variational Framework for Exemplar-Based Image Inpainting“. *International Journal of Computer Vision* 93(3), 2011, pp. 319–347.