



Mathematical Image Analysis Group  
Faculty of Mathematics and Computer Science  
Saarland University



# Evolutionary Models for Signal Enhancement and Approximation

Dissertation  
zur Erlangung des Grades des  
Doktors der Naturwissenschaften (Dr. rer. nat.)  
der Fakultät für Mathematik und Informatik der  
Universität des Saarlandes

submitted by

Leif Bergerhoff

Saarbrücken, 2020

**Date of the Defence**

28 June 2021

**Dean of the Faculty of Mathematics and Computer Science**

Prof. Dr. Thomas Schuster

**Members of the Examination Board**

Prof. Dr. Thomas Schuster  
Saarland University,  
Saarbrücken, Germany

Prof. Dr. Joachim Weickert  
Saarland University,  
Saarbrücken, Germany

Prof. Dr. Alfred M. Bruckstein  
Technion – Isreal Institute of Technology,  
Haifa, Israel

Prof. Dr. Martin Welk  
Private University for Health Sciences, Medical Informatics and Technology,  
Hall/Tyrol, Austria

Dr. Matthias Augustin  
Saarland University,  
Saarbrücken, Germany



## Short Abstract

This thesis deals with nature-inspired evolution processes for the purpose of signal enhancement and approximation. The focus lies on mathematical models which originate from the description of swarm behaviour. We extend existing approaches and show the potential of swarming processes as a modelling tool in image processing. In our work, we discuss the use cases of grey scale quantisation, contrast enhancement, line detection, and coherence enhancement. Furthermore, we propose a new and purely repulsive model of swarming that turns out to describe a specific type of backward diffusion process. It is remarkable that our model provides extensive stability guarantees which even support the utilisation of standard numerics. In experiments, we demonstrate its applicability to global and local contrast enhancement of digital images. In addition, we study the problem of one-dimensional signal approximation with limited resources using an adaptive sampling approach including tonal optimisation. We suggest a direct energy minimisation strategy and validate its efficacy in experiments. Moreover, we show that our approximation model can outperform a method recently proposed by Dar and Bruckstein.

## Kurzzusammenfassung

Die vorliegende Arbeit beschäftigt sich mit naturinspirierten Evolutionsprozessen, die sich zur Verbesserung und Approximation von Signalen eignen. Der Schwerpunkt liegt dabei auf mathematischen Modellen, die ihren Ursprung in der Beschreibung von Schwarmverhalten haben. Wir erweitern vorhandene Methoden und zeigen das Potenzial von Schwarmprozessen als Modellierungswerkzeug in der Bildverarbeitung auf. Im Rahmen dieser Arbeit besprechen wir folgende Anwendungsfälle: Grauwertquantisierung, Kontrastverstärkung, Geradenerkennung und Kohärenzverstärkung. Des Weiteren stellen wir ein neues und vollständig abstoßendes Schwarmmodell vor, mit dem es möglich ist, eine besondere Klasse von Rückwärtsdiffusionsprozessen zu beschreiben. Hervorzuheben sind hierbei weitreichende Stabilitätsgarantien unseres Modells, die sogar die Verwendung von Standardverfahren aus der Numerik miteinschließen. In Experimenten zeigen wir die Anwendbarkeit des Modells zur globalen und lokalen Kontrastverstärkung in digitalen Bildern. Darüber hinaus untersuchen wir das Problem der eindimensionalen Signalapproximation mit begrenzten Ressourcen unter der Verwendung eines adaptiven Abtastverfahrens mit tonaler Optimierung. Wir verfolgen eine direkte Energieminimierungsstrategie, deren Wirksamkeit wir in Experimenten bestätigen. Weiterhin zeigen wir, dass unser Modell einer kürzlich von Dar und Bruckstein veröffentlichte Methode überlegen ist.

## Abstract

This work has set itself the goal to study mathematically well-founded evolution processes and their application to the world of signal processing. The focus lies on the description of swarm behaviour and the usage of swarm models in the context of signal enhancement and signal approximation.

In a first step, we investigate the eligibility of swarm evolutions for the purpose of modelling image processing tasks and show that they have the potential to be way more than just optimisation tools as which they are often used. We propose an extension to attractive-repulsive discrete first-order models of swarming that complies with the theory of nonsymmetric nonlocal evolutions. Subsequently, we demonstrate that our generic swarming process suits well as a modelling tool: It allows us to describe grey scale quantisation, contrast enhancement, line detection, and coherence enhancement in terms of a swarm evolution.

Our next contribution deals with the ill-posed inverse problem of backward diffusion. Borrowing ideas from a purely repulsive swarm model, we come up with a smart model to describe a one-dimensional backward diffusion process. We achieve stability from reflecting boundary conditions in the diffusion co-domain, a concept which is new in the context of backward diffusion. Due to the fact that our model describes a gradient descent process on a convex energy, it provides excellent convergence guarantees and allows the application of simple numerics. This is in contrast to existing models which either require crude stabilisation terms or sophisticated numerics. In experiments, we show that our backward diffusion model allows to enhance the global and local contrast of digital greyscale and colour images.

Finally, we study the  $\ell^2$ -optimal approximation of arbitrary one-dimensional signals with piecewise-defined functions under the constraint of limited resources. This can also be interpreted in terms of adaptive sampling with tonal optimisation. We suggest an energy-based approach with minimal requirements that aims at minimising the mean squared error. Furthermore, we provide an alternative derivation of the recent Dar–Bruckstein model and disprove the optimality of error balancing for piecewise constant output functions. Due to the nonconvexity of the energy, we employ a particle swarm optimisation strategy. Additionally, we discuss the applicability of numerical first-order optimisation schemes. In our experiments, we estimate piecewise constant and piecewise linear approximation functions. We achieve high quality results and can beat the Dar–Bruckstein model in terms of the  $\ell^2$ -approximation error.

## Zusammenfassung

Diese Arbeit hat es sich zum Ziel gesetzt, mathematisch fundierte Evolutionsprozesse, sowie deren Anwendung im Bereich der Signalverarbeitung, zu untersuchen. Der Schwerpunkt liegt dabei auf der Beschreibung von Schwarmverhalten und der Verwendung von Schwarmmodellen im Zusammenhang mit Signalverbesserung und Signalapproximation.

In einem ersten Schritt prüfen wir die Eignung von Schwarmprozessen zur Modellierung von Bildverarbeitungsaufgaben und zeigen, dass diese das Potential dazu haben, mehr als nur Optimierungswerkzeuge zu sein, als die sie oft verwendet werden. Wir schlagen eine Erweiterung für diskrete Anziehungs-Abstoßungs-Swarmmodelle erster Ordnung vor, die mit der Theorie von nichtsymmetrischen nichtlokalen Evolutionsprozessen vereinbar ist. Anschließend zeigen wir, dass sich der von uns allgemein formulierte Schwarmprozess gut als Modellierungswerkzeug eignet: Er erlaubt es uns Grauwertquantisierung, Kontrastverstärkung, Geraden-erkennung und Kohärenzverstärkung im Sinne einer Schwarmevolution zu beschreiben.

Unser nächster Beitrag beschäftigt sich mit dem schlecht gestellten inversen Problem der Rückwärtsdiffusion. Basierend auf Ideen eines vollständig abstoßenden Schwarmmodells, entwickeln wir ein elegantes Modell zur Beschreibung eines eindimensionalen Rückwärtsdiffusionsprozesses. Dabei erreichen wir eine Stabilisierung des Prozesses mit Hilfe von reflektierenden Randbedingungen im Diffusions-Wertebereich, ein Konzept, das im Zusammenhang mit Rückwärtsdiffusion neu ist. Aufgrund der Tatsache, dass unser Modell einen Gradientenabstieg auf einer konvexen Energie beschreibt, bietet es hervorragende Konvergenzgarantien und erlaubt die Anwendung einfacher numerischer Verfahren. Dies steht im Gegensatz zu bereits existierenden Modellen, die entweder einen einflussreichen Stabilisierungsterm oder eine aufwändige Numerik voraussetzen. In Experimenten zeigen wir, dass sich unser Rückwärtsdiffusionsmodell zur globalen und lokalen Kontrastverstärkung von digitalen Grauwert- und Farbbildern eignet.

Zuletzt untersuchen wir die  $\ell^2$ -optimale Annäherung von beliebigen eindimensionalen Signalen mit Hilfe abschnittsweise definierter Funktionen bei eingeschränkten Ressourcen. Dies kann auch als adaptives Abtastverfahren mit tonaler Optimierung verstanden werden. Wir schlagen hierzu einen energiebasierten Ansatz mit minimalen Voraussetzungen vor, der die Minimierung des mittleren quadratischen Fehlers zum Ziel hat. Darüber hinaus liefern wir eine alternative Herleitung des kürzlich vorgestellten Dar-Bruckstein Modells und widerlegen die Optimalität der Fehlerbalancierung im Falle von abschnittsweise konstanten Ausgabefunktionen. Aufgrund der Nichtkonvexität der Energie, verwenden wir eine Partikel-Swarm-Optimierungsstrategie. Weiterhin diskutieren wir die Anwendbarkeit von numerischen Optimierungsverfahren erster Ordnung. In unseren Experimenten bestimmen wir abschnittsweise konstante, sowie abschnittsweise lineare Annäherungsfunktionen. Wir erzielen qualitativ hochwertige Resultate und können zudem die Ergebnisse des Modells von Dar und Bruckstein hinsichtlich des  $\ell^2$ -Approximationsfehlers unterbieten.

## Acknowledgements

At this point, I would like to take the opportunity and express my gratitude to the numerous people who have supported me throughout the last years. Without any doubt, this thesis wouldn't have been possible without you.

First of all, I would like to thank Prof. Joachim Weickert for offering me the chance to join the *Mathematical Image Analysis* (MIA) group and to do my doctoral studies at Saarland University. Not only do I really appreciate this but also the time and effort which he has spent on my supervision.

Furthermore, many thanks go to my collaborators and co-authors Prof. Martin Welk, Dr. Yehuda Dar, Dr. Marcelo Cárdenas, and Kireeti Bodduna for their support and contribution to the individual research projects.

Next, I would like to thank my colleagues from the MIA group for all their input, many fruitful discussions, and for providing such a pleasant working atmosphere. In particular, I want to thank Sarah Andris and David Hafner. Over the last years, I have also received a lot of support from closely connected research groups. For this reason, I would like to express my gratitude to Prof. Peter Ochs, Dr. Antoine Gautier, and Prof. Martin Reißel, too.

Special thanks go to the secretary of the MIA group, Ellen Wintringer, who never got tired to help me with all kinds of organisational issues in the most friendly way possible. Similarly, I want to thank our system administrator, Peter Franke, for doing a great job and his technical support in many non-standard setups.

Apart from that, I would like to thank my new colleagues at the German Aerospace Centre for their support throughout the final phase of my thesis.

It remains to say that I consider myself very lucky for all the scientific and personal support which I have received. The latter, of course, also includes the help of all my friends and my family who have always encouraged me to continue my research. Foremost, I want to thank my wife, whose patience I have certainly put to the test several times.

*ACKNOWLEDGEMENTS*

---

# Contents

<b>1</b>	<b>Introduction</b>	<b>1</b>
1.1	Motivation . . . . .	1
1.2	Scope and Contributions . . . . .	5
1.3	Thesis Outline . . . . .	7
<b>2</b>	<b>Foundations</b>	<b>9</b>
2.1	Preliminary Concepts and Definitions . . . . .	10
2.1.1	Basic Notations . . . . .	10
2.1.2	Interpretation of Digital Signals as Functions . . . . .	10
2.1.3	Important Function Classes . . . . .	11
2.1.4	Mean Squared Error . . . . .	12
2.1.5	Eigenvalue Analysis of Matrices . . . . .	13
2.1.6	Initial Value Problems . . . . .	13
2.2	Numerical Optimisation Techniques . . . . .	14
2.2.1	Gradient Descent Method . . . . .	14
2.2.2	Heavy Ball Method . . . . .	15
2.2.3	Adaptive FSI . . . . .	16
2.2.4	Backtracking Line Search . . . . .	17
2.2.5	Particle Swarm Optimisation . . . . .	18
<b>3</b>	<b>Attractive-Repulsive Swarming Models for Image Processing</b>	<b>21</b>
3.1	Introduction . . . . .	22
3.2	Discrete Modelling of Swarm Behaviour . . . . .	24
3.2.1	Basic Notations and Definitions . . . . .	24
3.2.2	Potential Energies and Forces . . . . .	24
3.2.3	Discrete First-Order Models of Swarming . . . . .	25
3.2.4	Time Discretisation . . . . .	27
3.3	Application to Image Processing Problems . . . . .	28
3.3.1	Grey Scale Quantisation . . . . .	28
3.3.2	Contrast Enhancement . . . . .	30
3.3.3	Line Detection . . . . .	31
3.3.4	Coherence Enhancement . . . . .	32
3.4	Conclusions and Outlook . . . . .	37

<b>4</b>	<b>Purely Repulsive Models and Backward Diffusion</b>	<b>39</b>
4.1	Introduction . . . . .	40
4.2	Model . . . . .	43
4.2.1	Motivation from Swarm Dynamics . . . . .	43
4.2.2	Discrete Variational Model . . . . .	44
4.3	Theory . . . . .	47
4.3.1	General Results . . . . .	47
4.3.2	Global Model . . . . .	51
4.3.3	Relation to Variational Signal and Image Filtering . . . . .	54
4.4	Explicit Time Discretisation . . . . .	56
4.5	Application to Image Enhancement . . . . .	60
4.5.1	Greyscale Images . . . . .	60
4.5.2	Colour Images . . . . .	62
4.5.3	Parameters . . . . .	64
4.5.4	Related Work from an Application Perspective . . . . .	65
4.6	Conclusions and Outlook . . . . .	68
4.A	Supplementary Material . . . . .	69
4.A.1	Derivations . . . . .	69
4.A.2	Parameters for Local Contrast Enhancement . . . . .	73
<b>5</b>	<b>Evolutions for One-Dimensional Signal Approximation</b>	<b>75</b>
5.1	Introduction . . . . .	76
5.2	Modelling One-Dimensional Signal Approximation . . . . .	78
5.2.1	Problem Statement . . . . .	78
5.2.2	The Approximation Functions $u$ . . . . .	78
5.2.3	Digital Input Signals $f$ . . . . .	84
5.3	The Dar–Bruckstein Method . . . . .	88
5.3.1	Compact Reformulation of the Dar–Bruckstein Method . . . . .	89
5.3.2	Limitations of the Dar–Bruckstein Method . . . . .	90
5.4	Direct Energy Optimisation . . . . .	91
5.4.1	Particle Swarm Optimisation (PSO) . . . . .	91
5.4.2	First-Order Optimisation Methods . . . . .	92
5.5	Experiments . . . . .	97
5.5.1	Piecewise Constant Approximation Functions $u_c(x)$ . . . . .	98
5.5.2	Piecewise Linear Approximation Functions $u_\ell(x)$ . . . . .	116
5.6	Conclusions and Outlook . . . . .	120
<b>6</b>	<b>Conclusions and Outlook</b>	<b>125</b>
6.1	Summary and Conclusions . . . . .	125
6.2	Outlook . . . . .	127
<b>A</b>	<b>Bibliography</b>	<b>129</b>
<b>B</b>	<b>Own Publications</b>	<b>141</b>
<b>C</b>	<b>Glossary</b>	<b>143</b>
<b>D</b>	<b>List of Symbols</b>	<b>145</b>



---

# Chapter 1

## Introduction

“I can’t understand why people are frightened of new ideas. I’m frightened of the old ones.”

---

John Cage, *Conversing with Cage*

### Contents

---

<b>1.1</b>	<b>Motivation</b>	<b>1</b>
<b>1.2</b>	<b>Scope and Contributions</b>	<b>5</b>
<b>1.3</b>	<b>Thesis Outline</b>	<b>7</b>

---

## 1.1 Motivation

Questioning things is an inherent part of human history. Essentially, it’s our curiosity which motivates us to explore the unknown and helps to understand and explain supposedly unexplainable phenomena. It turns out that, in particular, our most important teacher, the nature, follows fascinating and surprisingly simple rules which can assist us in solving complex problems.

One example out of many represents the Giant’s Causeway in Northern Ireland (see Figure 1.1). According to the Gaelic mythology it represents a remainder of a causeway connecting Northern Ireland and Scotland which was built and destroyed by a giant [Jon06]. In fact, it consists of 40000 basalt columns which emerged from the cooling process of lava around 50 to 60 million years ago [UNE20]. The Giant’s Causeway does – however – not only serve as a good basis for legends, or as tourist attraction. Taking a closer look at the structure and the ordering of the basalt columns one can identify a so-called centroidal Voronoi tessellation [DFG99]: nature presents us an optimal solution for a partitioning problem! This concept has been successfully transferred and exploited within the context of vector quantisation or clustering. There, it is common practice to use algorithms like Lloyd’s algorithm [Llo82] or the k-means algorithm [Mac67] to obtain a centroidal Voronoi tessellation.

The answers – as we can see perfectly in this particular case – lie straight ahead



Figure 1.1: The Giant's Causeway near Bushmills, Northern Ireland. Sources: <https://pixabay.com/images/id-539859/> (left picture), <https://pixabay.com/images/id-3801174/> (right picture).

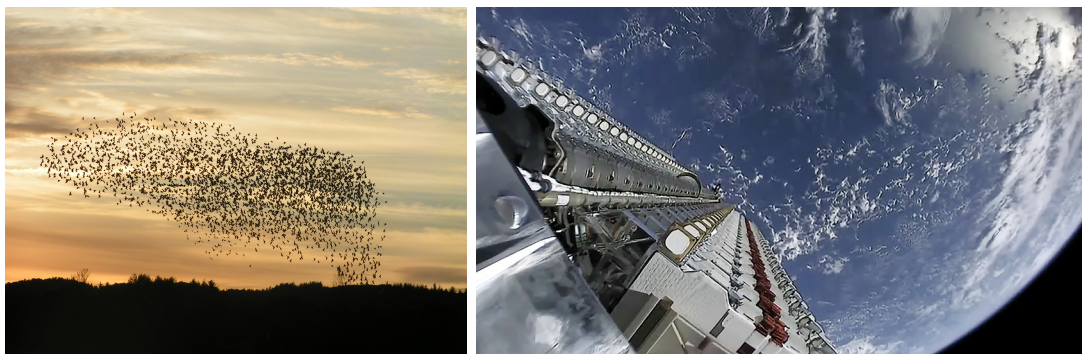


Figure 1.2: **Left:** Swarm of starlings looking for a roost near Silkeborg, Denmark. **Right:** 60 stacked satellites of the Starlink mission right after being launched in May 2019. Sources: [https://en.wikipedia.org/wiki/File:Sort\\_sol\\_ved\\_Ørnsø\\_2007.jpg](https://en.wikipedia.org/wiki/File:Sort_sol_ved_Ørnsø_2007.jpg) (left image), <https://flickr.com/photos/130608600@N05/47926144123> (right image).

and it is up to us to discover and ask the right questions in order of being able to learn from them. Already the sheer scale of the (known and partially explored) universe allows the conclusion that numerous alternative examples exist of which we might not be even aware of yet. In other cases, we have probably not been able to realise their whole extent so far. Anyhow, we should understand nature as a mentor which provides us with inspiration and guidance. Just consider our everyday lives. We navigate through the latter right in the middle of an environment which is equipped with many things we take for granted. However, for a lot of them we are far away from being able to completely understand or explain them. The movement of bird flocks (see Figure 1.2) – as well as the organisation therein – serves as a good example and is dealt with in the research areas of collective or swarm behaviour. Some important questions which arise cover very elementary concepts: How do single birds align? How do swarm members influence each other? How do they avoid collisions? All these questions have in common that we don't know a final answer yet. Nonetheless, their investigation is highly relevant for multiple applications which are in the focus of our society



Figure 1.3: **Left:** Diffusion of ink in water. **Centre:** Picture of a cow taken by the author. **Right:** Blurred cow image resulting from homogeneous diffusion. *Sources:* <https://pixabay.com/images/id-2427263/> (left picture).

today. We live in a world in which companies have planned and recently started to shoot 42000 additional satellites into the earth's atmosphere which will increase the number of human made spacecraft by a factor of five [Hen19] (see also Figure 1.2). Not only it is important for satellites to avoid collisions with space junk or asteroids. The higher the number of satellites will get the more important also becomes an autonomous strategy for swerving: a task which would definitely benefit from – and which should be based on – reliable mathematical models. Of course, concepts for future autonomous vehicles like cars, trains, or drones require the same in order of being able to improve public transport and to reach a maximum possible amount of reliability and safety.

A mathematical model which is capable of describing interactions amongst a large number of entities exists e.g. for diffusion processes (see Figure 1.3) where one is interested in distribution changes of some quantity (e.g. heat) over time. Weickert demonstrates in [Wei98] how this model can be applied as a denoising filter to digital greyscale and colour images, too. It turns out that the diffusion models used in image processing provide a lot of features which would also be desirable for (predictable) swarm behaviour. This becomes clear if one interprets the pixels of a digital image as members of an artificial swarm (the swarm represents the image as a whole). Furthermore, the individual position within the swarm corresponds to a pixel's grey or colour value and is subject to change over time. Analogously to the previously mentioned case of autonomous vehicles, it is desirable that the positions lie within a pre-defined domain which they never leave. The behaviour of all swarm members should be predictable and controllable while changing some properties of the swarm. Apart from that, it is of fundamental importance that the most important characteristics of the swarm remain unaffected. Speaking in terms of digital images again, removing noise should – of course – not change the meaning of the perceived content itself. Diffusion models allow to describe direction-dependent and attractive-repulsive swarming processes. A very interesting but challenging – and only partially solved – problem remains the description of pure backward diffusion. The latter is e.g. useful in the context of inverse heat propagation or a swarm which consists of repelling members only. Such complex dynamical systems are omnipresent in the world we live in, al-

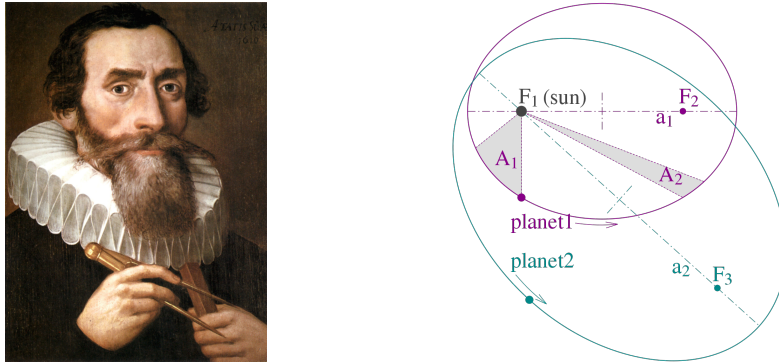


Figure 1.4: **Left:** Portrait of Johannes Kepler (1571–1630). **Right:** Illustration of Kepler's laws. *Sources:* [https://en.wikipedia.org/wiki/File:Johannes\\_Kepler\\_1610.jpg](https://en.wikipedia.org/wiki/File:Johannes_Kepler_1610.jpg) (left picture), [https://en.wikipedia.org/wiki/File:Kepler\\_laws\\_diagram.svg](https://en.wikipedia.org/wiki/File:Kepler_laws_diagram.svg) (right picture).

though, we often restrict ourselves to simplified or idealistic model variants in order of being able to cope with their original complexity. Consider e.g. the heliocentric model of the universe and Kepler's laws of planetary motion (see Figure 1.4). In fact, we know that they only represent an approximation to reality (see e.g. [GPS01]). Nevertheless, they cover the most important characteristics and provide sufficient accuracy for many applications (like the state estimation of earth-orbiting objects like satellites). This idea has also been transferred to the world of signal processing, e.g. for the purpose of signal compression and approximation. Although, there exist important and frequently used concepts for exact signal reconstruction, we usually don't require their precision in many situations in our everyday lives. Lossy compression techniques for images became part of our daily routine and allow efficient data storage at acceptable quality for many purposes as shown in Figure 1.5. Without any doubt, the method of JPEG compression [PM92] established as a standard format for digital images shared amongst millions or even billions of people around the globe. One interesting aspect within this context is the fact that lossy compression techniques are heavily-used those days although often there doesn't even exist a clear definition or consistent idea of what *optimal* signal approximation means. In case of limited resources – usually this refers to limited storage capacities – it is still unknown how the optimal approximation of a (non-trivial) piecewise constant signal should look like (e.g. in terms of a mean squared error).

It becomes clear that there are many (essential and) highly relevant open questions in the closely connected research areas of swarm behaviour, backward diffusion, and signal approximation. This is the starting point and main motivation of this dissertation.





Figure 1.5: **Left:** Original image ( $4096 \times 3072$  px, 2450786 bytes using JPEG compression) taken by the author near Neustrelitz, Germany. **Right:** The same image with reduced JPEG quality obtained using GIMP [Tea20] (quality = 25, 582106 bytes).



Figure 1.6: Example of swarm-based quantisation. **Left:** Original image from [Sig15] with 255 greyscales. **Centre:** Quantised image with 16 greyscales. **Right:** Quantised image with 8 greyscales.

## 1.2 Scope and Contributions

This thesis connects the mathematical modelling of swarm dynamics for the purpose of image processing, with a stable model for pure backward diffusion, and optimal signal approximation. We provide new insights into related complex processes and give answers to fundamental yet unanswered questions arising in all three domains.

**Attractive-Repulsive Swarming Models for Image Processing.** Emerged from simulation applications, swarming models have established as an eligible tool for optimisation. We follow a different approach describing their potential as a modelling instrument for image processing tasks. This enables us to use swarming models for the purpose of grey scale quantisation, contrast enhancement, and line detection. Exemplary results for grey scale quantisation are illustrated in Figure 1.6. Additionally, we explain the construction of a coherence enhancing image filter based on the theory of swarm models (see Figure 1.7).



Figure 1.7: Example of swarm-based coherence enhancing image filtering. **Left:** Original fingerprint image from [WWS06]. **Right:** Coherence enhanced image.



Figure 1.8: Example for contrast enhancement using our backward diffusion model. **Left:** Original image from [Kod]. **Centre:** Result with globally enhanced contrast. **Right:** Result with locally enhanced contrast.

**Purely Repulsive Models and Backward Diffusion.** The backward diffusion – or inverse heat propagation – problem is well-known to be ill-posed (cf. [Joh55]) and its solution requires extensive stabilisation. The latter can e.g. be achieved with the help of additional regularisation terms or sophisticated numerics. Inspired by purely repulsive swarm behaviour, we present a novel and smart model that implements globally negative diffusivities. Its stabilisation results from reflecting boundary conditions which we impose in the co-domain. Surprisingly, this simple trick allows us to formulate backward diffusion as a gradient descent process on a convex energy. The associated well-posedness properties permit the usage of standard numerics and guarantee a stable evolution. In experiments, we demonstrate the applicability of our model for global and local contrast enhancement of grey scale and colour images. Representative results for colour images are given in Figure 1.8.

**Evolutions for One-Dimensional Signal Approximation.** Our third contribution deals with piecewise constant and piecewise linear approximations of arbitrary one-dimensional signals (see Figure 1.9). We consider the yet unsolved problem of estimating optimal solutions which are based on a limited number of samples and minimise the mean squared error w.r.t. the original signal. Part of our work is the analysis of the connected nonconvex energy minimisation problem in general and for the specific cases of piecewise constant and linear approximations. This discussion also includes a compact and transparent reformulation of

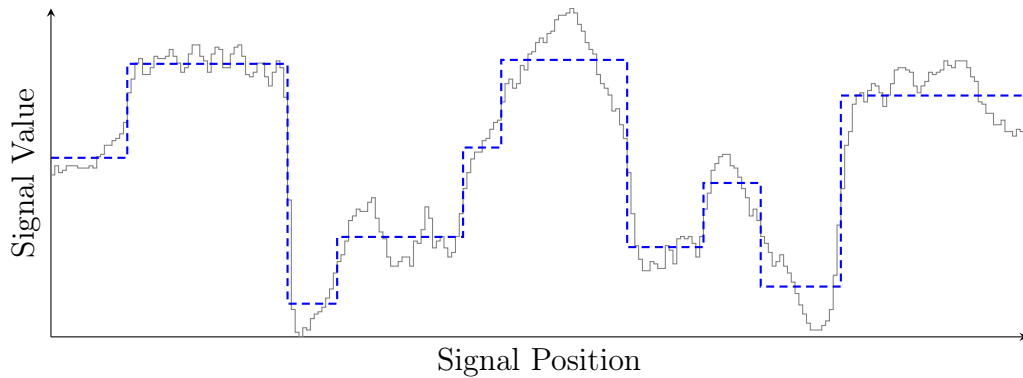


Figure 1.9: Illustration of piecewise constant signal approximation: The piecewise constant true signal – from Chapter 5 – in grey and our best piecewise constant approximation when using 10 samples in blue.

the Dar–Bruckstein model for adaptive sampling [DB19]. Furthermore, we disprove the suitability of error balancing as a criterion for  $\ell^2$ -optimality of piecewise constant approximations. In order to solve the nonconvex optimisation problem efficiently, we evaluate the applicability of nature-inspired and numerical first-order optimisation strategies.

### 1.3 Thesis Outline

The present work is structured as follows: First, we introduce notational conventions and discuss the mathematical foundations which build the basis of this thesis in Chapter 2. Subsequently, in Chapter 3, we present attractive-repulsive swarming models and their application to image processing problems. In Chapter 4, we introduce our model for purely repulsive swarm behaviour and describe how it can be used to solve the backward diffusion problem. Chapter 5 explains our energy-based evolution for one-dimensional signal approximation. Afterwards, we provide a summary and outlook in Chapter 6. Appendix A contains the bibliography and a list of our individual publications can be found in Appendix B. For the glossary and the list of symbols we refer to Appendix C and D accordingly.





---

# Chapter 2

## Foundations

“The aspects of a thing that are most important to us are hidden to us because of their simplicity and familiarity.”

---

Ludwig Wittgenstein, *Philosophical Investigations*

### Contents

---

<b>2.1 Preliminary Concepts and Definitions . . . . .</b>	<b>10</b>
2.1.1 Basic Notations . . . . .	10
2.1.2 Interpretation of Digital Signals as Functions . . . . .	10
2.1.3 Important Function Classes . . . . .	11
2.1.4 Mean Squared Error . . . . .	12
2.1.5 Eigenvalue Analysis of Matrices . . . . .	13
2.1.6 Initial Value Problems . . . . .	13
<b>2.2 Numerical Optimisation Techniques . . . . .</b>	<b>14</b>
2.2.1 Gradient Descent Method . . . . .	14
2.2.2 Heavy Ball Method . . . . .	15
2.2.3 Adaptive FSI . . . . .	16
2.2.4 Backtracking Line Search . . . . .	17
2.2.5 Particle Swarm Optimisation . . . . .	18

---

This chapter addresses the basic notation used in this thesis and introduces essential tools such as

- mathematical concepts and definitions, and
- numerical optimisation algorithms,

which are used in the main part afterwards.

## 2.1 Preliminary Concepts and Definitions

### 2.1.1 Basic Notations

Throughout our work we make use of the subsequent typesetting conventions in order to accentuate and differentiate between scalars, vectors, matrices, functions, and sets:

- We use lower-case letters to express scalar values and employ bold lower-case letters for vectors. For example,  $\mathbf{f} = (f_1, \dots, f_n)^T \in \mathbb{R}^n$  represents a real-valued column vector of length  $n \in \mathbb{N}$  with first scalar element  $f_1 \in \mathbb{R}$ . Apart from that, we employ capital letters to refer to important constants such as the Lipschitz constant  $L$ , or the total number of samples  $N$ .
- Matrices use bold upper-case letters, such that  $\mathbf{A} \in \mathbb{R}^{n \times m}$  refers to a real-valued matrix which consists of  $n$  rows and  $m$  columns. We refer to the corresponding matrix element in row  $i$  and column  $j$  by  $a_{i,j} \in \mathbb{R}$ .
- In general, we denote functions by lower-case letters. Only for energy functions, forces, and integrated functions, we use upper-case letters. In contrast to scalar-valued functions we write the letters of vector-valued functions in bold face: e.g. we write  $f : \mathbb{N}^2 \rightarrow \mathbb{R}$ , but  $\mathbf{g} : \mathbb{N}^2 \rightarrow \mathbb{R}^3$ .
- We utilise upper-case letters to represent sets. An example is given by  $Q \subseteq \mathbb{R}^n$ .

### 2.1.2 Interpretation of Digital Signals as Functions

We often find ourselves in the situation where we want to apply a mathematical model to some given digital input signal. The latter consists of discrete measurements and is typically given either as a real-valued vector or a real-valued matrix. In order of being able to process this pointwise information we make use of the following mappings which can – if necessary – also be adapted to vector-valued data.

**One-dimensional Signals.** We consider a signal vector  $\mathbf{f} \in \mathbb{R}^n$  as a function

$$f : \mathbb{N} \rightarrow \mathbb{R} \tag{2.1}$$

which maps a discrete position  $i \in \{1, \dots, n\}$  to the corresponding value  $f_i \in \mathbb{R}$ .

**Two-dimensional Signals.** Similarly, we consider a signal matrix  $\mathbf{A} \in \mathbb{R}^{n \times m}$  as a function

$$f : \mathbb{N} \times \mathbb{N} \rightarrow \mathbb{R} \tag{2.2}$$

which maps a discrete grid position  $(i, j) \in \{1, \dots, n\} \times \{1, \dots, m\}$  to the corresponding value  $a_{i,j} \in \mathbb{R}$ .

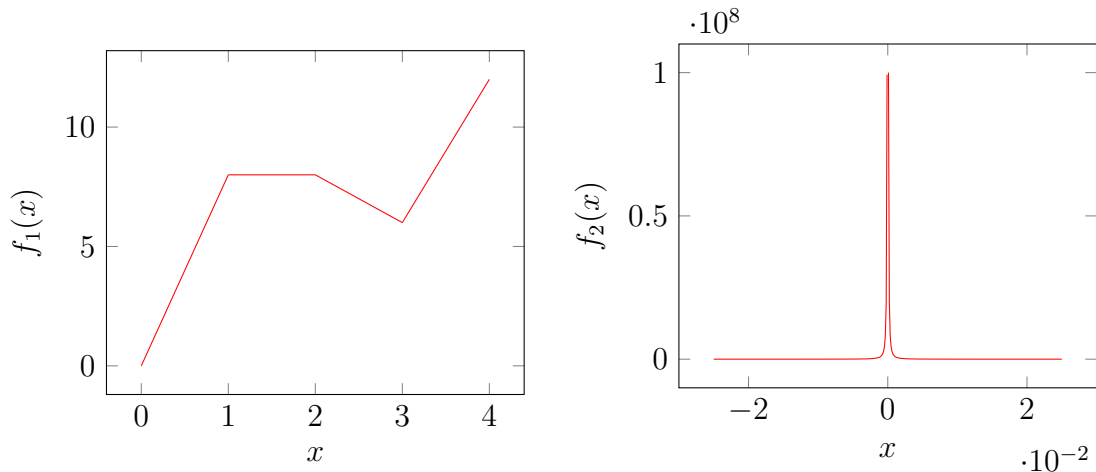


Figure 2.1: While  $f_1 \in C_8^{0,0}([0, 4])$ ,  $f_2$  is not Lipschitz continuous on  $\mathbb{R} \setminus \{0\}$ .

### 2.1.3 Important Function Classes

Within the next section, we introduce two function classes which are of particular importance in conjunction with mathematical optimisation problems: Lipschitz continuous functions and convex functions. Both provide support for reasonable assumptions on a given task and allow to differentiate between distinct types of optimisation problems. As a consequence, they are essential in the process of determining an appropriate optimisation strategy.

**Lipschitz Continuity.** A function  $f : Q \rightarrow \mathbb{R}$  is said to be Lipschitz continuous on  $Q \subseteq \mathbb{R}^n$  if it fulfils the Lipschitz condition

$$\|f(\mathbf{x}) - f(\mathbf{y})\| \leq L\|\mathbf{x} - \mathbf{y}\|, \quad \text{for some } L \geq 0, \quad (2.3)$$

for all  $\mathbf{x}, \mathbf{y} \in Q$ . This work borrows the notation  $f \in C_L^{k,p}(Q)$  as used by Nesterov in [Nes04] in order to express that a function  $f$  is  $k$ -times differentiable on  $Q$  while its  $p$ -th derivative is Lipschitz continuous with constant  $L$ . It holds that  $0 \leq p \leq k$ . All functions of this class have in common that the variation of the magnitude of the  $p$ -th signal derivative is bounded. If we neglect the Lipschitz continuity of the  $p$ -th derivative and refer to the class of  $k$ -times differentiable functions on  $Q$  we simply write  $f \in C^k(Q)$ . The functions

$$f_1(x) := \begin{cases} 8x, & \text{for } 0 \leq x < 1 \\ 8, & \text{for } 1 \leq x < 2 \\ -2x + 12, & \text{for } 2 \leq x < 3 \\ 6x - 12, & \text{for } 3 \leq x \leq 4 \end{cases}, \quad f_2(x) := \frac{1}{x^2}, \quad \text{for } x \in \mathbb{R} \setminus \{0\}, \quad (2.4)$$

serve as an example for a Lipschitz continuous and a non-Lipschitz continuous function (see also Figure 2.1).

**Convexity.** According to [BV04] a function  $f : Q \rightarrow \mathbb{R}$  is convex on  $Q \subseteq \mathbb{R}^n$  if it fulfils Jensen's inequality

$$f(\gamma\mathbf{x} + (1 - \gamma)\mathbf{y}) \leq \gamma f(\mathbf{x}) + (1 - \gamma)f(\mathbf{y}), \quad (2.5)$$

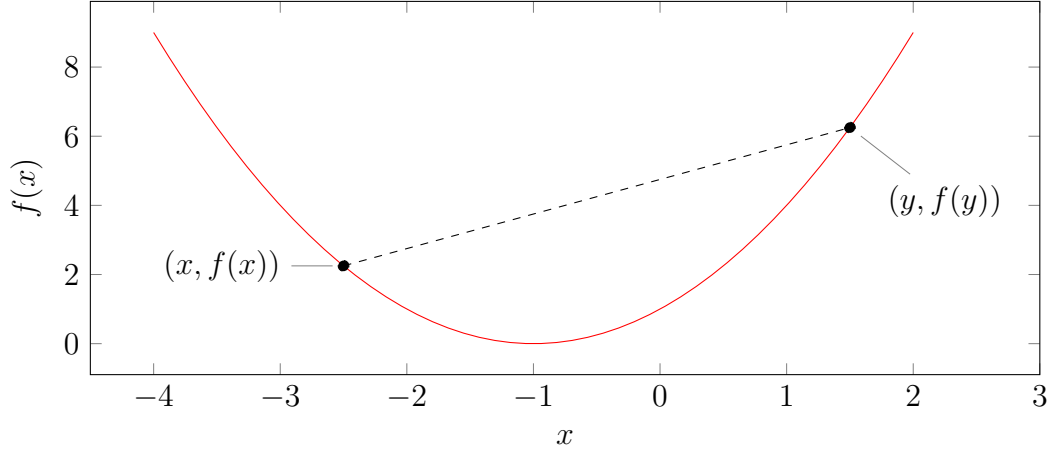


Figure 2.2: Graph of the convex function  $f(x) = x^2 + 2x + 1$ .

for  $0 \leq \gamma \leq 1$ , where  $\mathbf{x}, \mathbf{y} \in Q$ , and  $Q$  is a convex set. As illustrated in Figure 2.2, this inequality involves that the value of a convex function  $f$  between any two positions  $\mathbf{x}$  and  $\mathbf{y}$  never lies above the line segment through  $f(\mathbf{x})$  and  $f(\mathbf{y})$ . Every function  $f$  for which the inequality (2.5) is strict – assuming  $0 < \gamma < 1$  and  $\mathbf{x} \neq \mathbf{y}$  – is said to be strictly convex.

### 2.1.4 Mean Squared Error

A classical measure to quantify the differences between two discrete signals is the mean squared error

$$\text{MSE}(\mathbf{F}, \mathbf{G}) := \frac{1}{N} \|\mathbf{F} - \mathbf{G}\|_F^2, \quad \text{for } \mathbf{F}, \mathbf{G} \in \mathbb{R}^{n \times m}, \quad (2.6)$$

where  $N := n \cdot m$  denotes the number of samples and  $\|\cdot\|_F$  represents the Frobenius norm (see also [HTF09]). This simple but effective method sums up the squared differences at each signal position, i.e. the MSE is always greater than or equal to zero and reflects the average element-wise difference of both signals.

It is well-known that the MSE is not invariant w.r.t. translations or rotations of signals. However, these scenarios are not relevant within the scope of this thesis. For this reason, the MSE represents our method of choice when comparing two signals.

**Examples.** Two simple examples illustrate the estimation of the MSE for one- and two-dimensional signals.

$$\mathbf{f}_1 := (2, 0, 2, 0)^T, \quad \mathbf{g}_1 := (0, 2, 0, 2)^T, \quad \text{MSE}(\mathbf{f}_1, \mathbf{g}_1) = 4 \quad (2.7)$$

$$\mathbf{F}_2 := \begin{pmatrix} 2 & 0 & 0 & 0 \\ 0 & 2 & 0 & 0 \\ 0 & 0 & 2 & 0 \\ 0 & 0 & 0 & 2 \end{pmatrix}, \quad \mathbf{G}_2 := \begin{pmatrix} 0 & 0 & 0 & 2 \\ 0 & 0 & 2 & 0 \\ 0 & 2 & 0 & 0 \\ 2 & 0 & 0 & 0 \end{pmatrix}, \quad \text{MSE}(\mathbf{F}_2, \mathbf{G}_2) = 2 \quad (2.8)$$

### 2.1.5 Eigenvalue Analysis of Matrices

**Spectral Radius.** In accordance with [SK04] the spectral radius of some matrix  $\mathbf{A} \in \mathbb{R}^{n \times n}$  is defined as

$$\rho(\mathbf{A}) := \max_i |\lambda_i|, \quad \text{for } i = 1, 2, \dots, n, \quad (2.9)$$

where  $\lambda_i$  denotes the  $i$ -th eigenvalue of  $\mathbf{A}$ .

**Gershgorin Circle Theorem.** In order of being able to numerically estimate the solution to large-scale optimisation problems it is often sufficient to find an approximation to the upper boundary of the spectral radius of a given system matrix. This can be done in a comfortable way with the help of Gershgorin's Circle Theorem [Ger31, SK04]. Let

$$K_i := \{z \in \mathbb{C} \mid |z - a_{ii}| \leq r_i := \sum_{\substack{k=1 \\ k \neq i}}^n |a_{ik}|\}, \quad \text{for } i = 1, \dots, n, \quad (2.10)$$

then every eigenvalue of  $\mathbf{A} \in \mathbb{R}^{n \times n}$  lies within  $\bigcup_{i=1}^n K_i$ .

**Example.** Let's consider the symmetric real-valued  $4 \times 4$  matrix

$$\mathbf{A}_1 := \begin{pmatrix} 4 & 1 & 0 & 0 \\ 1 & 5 & 1 & 0 \\ 0 & 1 & 4 & 2 \\ 0 & 0 & 2 & 3 \end{pmatrix}. \quad (2.11)$$

As a consequence of its symmetry and the Gershgorin circle theorem all eigenvalues of  $\mathbf{A}_1$  lie within the real-valued subset of  $\bigcup_{i=1}^4 K_i$ . More precisely, this means that all eigenvalues fulfil

$$\lambda_i \in [1, 7], \quad \text{for } i = 1, 2, 3, 4, \quad (2.12)$$

and  $\rho(\mathbf{A}_1) \leq 7$ . These findings can e.g. be validated numerically using Maple [Map18] which returns the eigenvalues

$$\lambda_1 \approx 6.3234, \quad \lambda_2 = 5, \quad \lambda_3 \approx 3.35793, \quad \lambda_4 \approx 1.31867. \quad (2.13)$$

### 2.1.6 Initial Value Problems

The tasks in this work are formulated as time evolutions in a one- or two-dimensional signal domain. From a theoretical viewpoint this comes down to solve an initial value problem of type

$$\dot{\mathbf{x}}(t) = \mathbf{f}(\mathbf{x}(t)) \quad (2.14)$$

$$\mathbf{x}(0) = \mathbf{x}_0 \quad (2.15)$$

with  $t \in \mathbb{R}_0^+$ ,  $\mathbf{x}_0 \in \mathbb{R}^n$ ,  $\mathbf{x} : \mathbb{R}_0^+ \rightarrow \mathbb{R}^n$ , and  $\mathbf{f} : \mathbb{R}^n \rightarrow \mathbb{R}^n$ . The goal is to estimate the dynamical system  $\mathbf{x}(t)$  for which the initial state  $\mathbf{x}_0$  is known and whose change in time is given by  $\mathbf{f}$ . Usually, the dynamical system is evaluated for a certain time  $t$  or one is interested in its steady state

$$\mathbf{x}^* := \lim_{t \rightarrow \infty} \mathbf{x}(t). \quad (2.16)$$

A good resource for more detailed information about dynamical systems provides the book by Perko [Per01].

**Energy Minimisation Problems.** Within the scope of this thesis we restrict ourselves to initial value problems which can be interpreted as energy minimisation problems. More detailed, we assume that there exists an energy function  $E : \mathbb{R}^n \rightarrow \mathbb{R}$  which is minimised by a time evolution (2.14)

$$\dot{\mathbf{x}}(t) = -\nabla_{\mathbf{x}(t)} E(\mathbf{x}(t)) \quad (2.17)$$

in terms of a gradient descent.

## 2.2 Numerical Optimisation Techniques

Often, it is not feasible to derive an analytical solution to a given initial value problem. In these cases, numerical methods are of essential importance and allow to estimate an approximative solution (see e.g. [SK04, BV04, Nes04]). Subsequently, we introduce different numerical optimisation techniques which are used in the context of this thesis.

### 2.2.1 Gradient Descent Method

An elementary method for unconstrained minimisation of functions  $f \in C_L^{1,1}(\mathbb{R}^n)$  which are bounded from below represents the *gradient descent method*. Other frequently used names are *gradient algorithm* or *gradient method* (see e.g. [Nes04, BV04, Pol87]). The gradient descent method belongs to the class of first-order optimisation techniques due to the fact that it makes use of the first-order derivative of  $f$ . Its basic idea is described in Algorithm 1. Therein the positive scalar  $\alpha$  denotes the step size and  $\mathbf{x}^k$  refers to a position after  $k$  iterations.

---

**Algorithm 1** Gradient Descent Method

(based on [Nes04, (1.2.9),(1.2.16)] and [BV04, Algorithm 9.3])

---

$\mathbf{x}^0 \in \mathbb{R}^n, k := 0$

**while** stopping criterion is not met **do**

$\mathbf{x}^{k+1} \leftarrow \mathbf{x}^k - \alpha \nabla f(\mathbf{x}^k)$

$k \leftarrow k + 1$

**end while**

---

**Convergence and Optimal Time Step Size.** The gradient descent method converges to a local minimum of  $f$  for every step size  $\alpha$  which satisfies

$$0 < \alpha < \frac{2}{L}. \quad (2.18)$$

A proof for this statement can e.g. be found in [Nes04]. Furthermore, there exists an optimal step size

$$\alpha^* := \frac{1}{L} \quad (2.19)$$

which leads to most rapid descent of the objective function  $f$ . Based on [Nes04, (1.2.12)], one can verify that for  $\alpha = \alpha^*$  the decrease of the value of  $f$  follows at least

$$f(\mathbf{x}^{k+1}) \leq f(\mathbf{x}^k) - \frac{\|\nabla f(\mathbf{x}^k)\|_2^2}{2L}. \quad (2.20)$$

**Stopping Criterion.** Usually, the stopping criterion is given by  $\|\nabla f(\mathbf{x}^k)\|_2 \leq \varepsilon$ , where  $\varepsilon$  is a small and positive constant [BV04]. Unless stated otherwise we make use of

$$\varepsilon := \frac{\|\nabla f(\mathbf{x}^0)\|_2}{10^6}, \quad (2.21)$$

where we assume that  $\|\nabla f(\mathbf{x}^0)\|_2 > 0$ .

**Subspaces of  $\mathbb{R}^n$ .** As long as one can ensure that  $\mathbf{x}^k \in Q$  for all  $k \geq 0$  the gradient descent method can be applied without modifications to functions  $f \in C_L^{1,1}(Q)$ , where  $Q \subseteq \mathbb{R}^n$ .

### 2.2.2 Heavy Ball Method

Another first-order minimisation method – called the *heavy ball method* – was proposed by Polyak [Pol64]. It belongs to the class of so-called multi-step methods since – in every iteration – it makes use of information from multiple preceding steps. The scheme of the heavy ball method is given in Algorithm 2. The para-

---

**Algorithm 2** Heavy Ball Method  
(based on [Pol87, 3.2.1])

---

```

 $\mathbf{x}^0 \in \mathbb{R}^n, \mathbf{x}^{-1} := \mathbf{x}^0, k := 0$ 
while stopping criterion is not met do
     $\mathbf{x}^{k+1} \leftarrow \mathbf{x}^k - \alpha \nabla f(\mathbf{x}^k) + \beta(\mathbf{x}^k - \mathbf{x}^{k-1})$ 
     $k \leftarrow k + 1$ 
end while

```

---

meter  $\beta$  steers the inertia of the method while  $\alpha$ , again, denotes the step size. If the parameter values are chosen such that

$$\beta \in [0, 1), \quad \text{and} \quad \alpha \in \left(0, \frac{2(1+\beta)}{L}\right), \quad (2.22)$$


---

then the heavy ball method converges to a local minimum of the objective function  $f$  [Pol87, 3.2.1, Theorem 1]. For  $\beta = 0$ , the algorithm simplifies to the gradient descent method discussed in Section 2.2.1. Within this thesis we set

$$\beta = 0.75 \quad (2.23)$$

and use the same stopping criterion as for the gradient descent method. As illustrated in Figure 2.3, the introduction of the inertia term  $\beta(\mathbf{x}^k - \mathbf{x}^{k-1})$  may increase the convergence of the algorithm: While in the particular example the gradient descent method tends to zigzag motion, the heavy ball method has a much smoother trajectory.

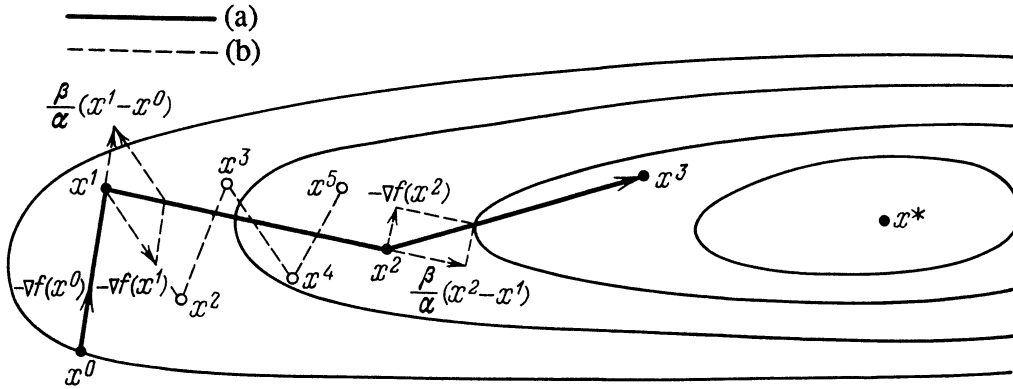


Figure 2.3: Trajectories of the heavy ball method (a) and the gradient descent method (b) from an initial point  $\mathbf{x}^0$  towards the minimum  $\mathbf{x}^*$  (image taken from [Pol87]).

### 2.2.3 Adaptive FSI

Recently, Tómasson et al. [TOW19] came up with *Adaptive FSI (AFSI)* schemes. Also their algorithm represents a first-order multi-step method and can be interpreted as a cyclic variant of the heavy ball method. It is illustrated in Algorithm 3. Therein, a new cycle is initiated every time when the reset condition

$$\nabla f(\mathbf{x}^k)^\top (\mathbf{x}^k - \mathbf{x}^{k-1}) > 0 \quad (2.24)$$

is satisfied and the index  $k$  is set to zero again. By doing so, AFSI implements an adaptive restart of a cycle which means that the length of a cycle is determined automatically by the algorithm. Within each cycle the index  $k$  increases sequentially and the extrapolation parameter  $\alpha_k$  is updated accordingly. In case the step size parameter  $\omega$  satisfies

$$\omega \in \left(0, \frac{2}{L}\right), \quad (2.25)$$

convergence of the method to a local minimum is guaranteed using the same reasoning as for the heavy ball method: One can show that – for every  $k$  – the conditions in (2.22) are fulfilled, where we assume  $\beta = a_k - 1$  and  $\alpha = \alpha_k \omega$ .



---

**Algorithm 3** AFSI

(based on [TOW19, Algorithm 1])

---

```

 $\mathbf{x}^0 \in \mathbb{R}^n, \mathbf{x}^{-1} := \mathbf{x}^0, k := 0$ 
while stopping criterion is not met do
  if  $\nabla f(\mathbf{x}^k)^\top (\mathbf{x}^k - \mathbf{x}^{k-1}) > 0$  then
     $\mathbf{x}^{-1}, \mathbf{x}^0 \leftarrow \mathbf{x}^{k-1}$ 
     $k \leftarrow 0$ 
  end if
   $\alpha_k \leftarrow \frac{4k+2}{2k+3}$ 
   $\mathbf{x}^{k+1} \leftarrow \mathbf{x}^k - \alpha_k \omega \nabla f(\mathbf{x}^k) + (\alpha_k - 1)(\mathbf{x}^k - \mathbf{x}^{k-1})$ 
   $k \leftarrow k + 1$ 
end while

```

---

In summary, AFSI combines the simplicity of the gradient descent method with the convergence benefits of a multi-step procedure like the heavy ball method without the requirement of additional parameters.

### 2.2.4 Backtracking Line Search

While for AFSI the step size  $\alpha = \alpha_k \omega$  depends on the index  $k$  and varies over time, we have – so far – assumed a constant step size  $\alpha$  for the gradient descent and the heavy ball method. In order to adapt both algorithms better to the shape of the objective function  $f$  it is possible to set the step size  $\alpha$  via *backtracking line search* [NW99, 3.1]. The latter is a simple and efficient strategy for automatic step size selection and can in practice also be used to handle jumps and kinks of the objective function. We show the most basic form of backtracking line search in Algorithm 4. The parameter  $\bar{\alpha} > 0$  represents the initial step length and is

---

**Algorithm 4** Backtracking Line Search

(based on [NW99, Procedure 3.1])

---

```

 $\alpha \leftarrow \bar{\alpha}$ 
while  $f(\mathbf{x}^k + \alpha \mathbf{p}^k) > f(\mathbf{x}^k) + d\alpha \nabla f(\mathbf{x}^k)^\top \mathbf{p}^k$  do
   $\alpha \leftarrow \rho \alpha$ 
end while

```

---

– according to [NW99] – often set to one. Another option is to set  $\bar{\alpha}$  to the optimal step size given in (2.19). The variable  $\mathbf{p}^k$  denotes some descent direction of the objective function  $f$  in the current iteration  $k$ , i.e.  $\nabla f(\mathbf{x}^k)^\top \mathbf{p}^k < 0$ . The contraction parameter  $\rho \in (0, 1)$  allows to steer the crudeness of the line search. Furthermore, the constant  $d \in (0, 1)$  sets the minimum desired decrease of the objective function as a fraction of the linear extrapolation of  $f$  (cf. [BV04, 9.2]). If not stated otherwise we set

$$d = 10^{-4}, \tag{2.26}$$

$$\rho = 10^{-1}. \tag{2.27}$$

## 2.2.5 Particle Swarm Optimisation

Based on a simulation of social behaviour, Kennedy and Eberhart [KE95] proposed the iterative *Particle Swarm Optimisation* (PSO) algorithm for nonlinear function optimisation. In contrast to the numerical minimisation techniques mentioned before, PSO does not require a differentiable objective function. This comes at the price of losing convergence guarantees, however, experimental studies have proven the usefulness and competitiveness of PSO and its variants.

In the context of PSO, a “swarm” describes a given number of  $n$  virtual particles which explore the solution space with the help of their memory as well as social interactions. In iteration  $k$ , the velocity  $\mathbf{v}_i^k$  and position  $\mathbf{x}_i^k$  of particle  $i$  are updated, where  $i = 1, 2, \dots, n$ . It is common to distribute the initial particle positions  $\mathbf{x}_i^0$  uniformly in the solution space. Each particle position represents a potential minimiser of the objective function with corresponding value  $f(\mathbf{x}_i^k)$ .

**Original Algorithm.** As stated in [BM08, Chapter 3], the original PSO algorithm updates the particle velocities and positions via

$$\mathbf{v}_i^{k+1} = \mathbf{v}_i^k + c_1 \mathbf{u}_1^k \odot (\bar{\mathbf{p}}_i^k - \mathbf{x}_i^k) + c_2 \mathbf{u}_2^k \odot (\bar{\mathbf{g}}^k - \mathbf{x}_i^k), \quad (2.28)$$

$$\mathbf{x}_i^{k+1} = \mathbf{x}_i^k + \mathbf{v}_i^{k+1}, \quad (2.29)$$

where

$$\bar{\mathbf{p}}_i^k := \arg \min_{j=0,1,\dots,k} f(\mathbf{x}_i^j) \quad (2.30)$$

is the preceding position of particle  $i$  which induces the lowest objective value. The best previous position of the whole swarm is given by

$$\bar{\mathbf{g}}^k := \arg \min_{i=1,2,\dots,n} f(\bar{\mathbf{p}}_i^k). \quad (2.31)$$

Consequently,  $\bar{\mathbf{g}}^k$  represents the best minimiser of the objective function found up to iteration  $k$ . Within (2.28) the symbol  $\odot$  denotes element-wise vector multiplication, whereas  $\mathbf{u}_1^k$  and  $\mathbf{u}_2^k$  represent independent and uniformly distributed random vectors with components in  $[0, 1]$ . The cognitive and social acceleration coefficients are given by the nonnegative weights  $c_1$  and  $c_2$ . Kennedy and Eberhart [KE95] set both values to 2.

People have suggested many improvements and variations for the original PSO algorithm. A recent and comprehensive review is e.g. given in [BM17].

**Standard Particle Swarm Optimisation 2011.** Within the scope of this thesis we make use of the *Standard Particle Swarm Optimisation 2011* (SPSO-2011) algorithm which incorporates an adaptive random particle neighbourhood topology and rotation invariance [ZCR13]. For SPSO-2011 the update rule for

the particle velocity  $\mathbf{v}_i^k$  in iteration step  $k$  reads

$$\mathbf{p}_i^k = \mathbf{x}_i^k + c_1 \mathbf{u}_1^k \odot (\bar{\mathbf{p}}_i^k - \mathbf{x}_i^k), \quad (2.32)$$

$$\mathbf{l}_i^k = \mathbf{x}_i^k + c_2 \mathbf{u}_2^k \odot (\bar{\mathbf{l}}_i^k - \mathbf{x}_i^k), \quad (2.33)$$

$$\mathbf{g}_i^k = \frac{1}{3} (\mathbf{x}_i^k + \mathbf{p}_i^k + \mathbf{l}_i^k), \quad (2.34)$$

$$\mathbf{v}_i^{k+1} = \omega \mathbf{v}_i^k + \mathcal{H}_i(\mathbf{g}_i^k, |\mathbf{g}_i^k - \mathbf{x}_i^k|) - \mathbf{x}_i^k. \quad (2.35)$$

Using (2.32) and (2.33), the algorithm estimates the two points  $\mathbf{p}_i^k$  and  $\mathbf{l}_i^k$  first. While  $\mathbf{p}_i^k$  lies in direction of  $\bar{\mathbf{p}}_i^k$  (the previous best position of particle  $i$ ), the point  $\mathbf{l}_i^k$  implements the influence of neighbour knowledge and is located near  $\bar{\mathbf{l}}_i^k$ . The latter denotes the best previous position – concerning the objective value  $f$  – among the  $n_{neigh}$  neighbours of particle  $i$ . Next, the centre of gravity  $\mathbf{g}_i^k$  of  $\mathbf{x}_i^k$ ,  $\mathbf{p}_i^k$ , and  $\mathbf{l}_i^k$  is set in (2.34). In (2.35), we update the velocity  $\mathbf{v}_i^{k+1}$  of particle  $i$  based on its current velocity  $\mathbf{v}_i^k$  and  $\mathbf{g}_i^k$ . The nonnegative weight  $\omega$  allows to steer the particle inertia. By  $\mathcal{H}_i(\mathbf{g}_i^k, |\mathbf{g}_i^k - \mathbf{x}_i^k|)$ , we refer to a random point from the hypersphere around  $\mathbf{g}_i^k$  with radius  $|\mathbf{g}_i^k - \mathbf{x}_i^k|$  ( $|\cdot|$  denotes the Euclidean norm). Equivalent to the original PSO algorithm, we use (2.29) to move  $\mathbf{x}_i^k$  to its new position  $\mathbf{x}_i^{k+1}$ . In case the global optimum does not improve, each particle randomly selects  $n_{neigh}$  new neighbours.

Zambrano-Bigiarini et al. [ZCR13] recommend the parameter values

$$c_1 = 0.5 + \ln(2), \quad (2.36)$$

$$c_2 = 0.5 + \ln(2), \quad (2.37)$$

$$\omega = \frac{1}{2 \ln(2)}, \quad (2.38)$$

which we use throughout this thesis. Besides this, we employ a neighbourhood size of  $n_{neigh} = 20$  and reinitialise the particle neighbourhoods after 15 iterations without an improvement of the global optimum.

**Stopping Criterion.** The PSO and SPSO-2011 algorithm stop when a maximum number of iterations or a tolerable objective value is reached (see e.g. [ZCR13]).



---

# Chapter 3

## Attractive-Repulsive Swarming Models for Image Processing

“It seems safe to look forward to the time when the conception of attractive and repulsive forces, having served its purpose as a useful piece of scientific scaffolding, will be replaced by the deduction of the phenomena known as attraction and repulsion, from the general laws of motion.”

---

T. H. Huxley, *The Advance of Science in the Last Half-Century*

### Contents

---

<b>3.1</b>	<b>Introduction</b>	<b>22</b>
<b>3.2</b>	<b>Discrete Modelling of Swarm Behaviour</b>	<b>24</b>
3.2.1	Basic Notations and Definitions	24
3.2.2	Potential Energies and Forces	24
3.2.3	Discrete First-Order Models of Swarming	25
3.2.4	Time Discretisation	27
<b>3.3</b>	<b>Application to Image Processing Problems</b>	<b>28</b>
3.3.1	Grey Scale Quantisation	28
3.3.2	Contrast Enhancement	30
3.3.3	Line Detection	31
3.3.4	Coherence Enhancement	32
<b>3.4</b>	<b>Conclusions and Outlook</b>	<b>37</b>

---

Main parts of this chapter base on our conference publication [BW16] and on joint work with Marcelo Cárdenas, Kireeti Bodduna, and Joachim Weickert which was published as part of the PhD thesis of Marcelo Cárdenas [Cár18].

So far most applications of swarm behaviour in image analysis use swarms as models for *optimisation* tasks. In this chapter, we follow a different philosophy

and propose to exploit them as valuable tools for *modelling* image processing problems. To this end, we consider models of swarming that are individual-based and of first order. We show that a suitable adaptation of the potential forces allows us to model three classical image processing tasks: grey scale quantisation, contrast enhancement, and line detection. A more advanced scenario represents the construction of a coherence enhancing image filter based solely on our swarming theory. For this purpose, we propose a novel two-step approach which employs a swarm model in the gradient domain and – on top of that – another one in the grey value domain. These proof-of-concept applications demonstrate that modelling image analysis tasks with swarms can be simple, intuitive, and highly flexible.

### 3.1 Introduction

The interest of understanding and imitating nature plays an elementary role in human history. In this context the phenomenon of swarm behaviour recently as an example which has received increasing attention. The investigation of bird flocks, fish schools, locust swarms, bat populations, fireflies, ant or termite tribes, and many others has led to numerous models of swarming in literature. Going along with this, it is not surprising that research on those models exists in a huge variety of fields. The latter include biology [CKJ<sup>+</sup>02], computer science [Rey87], robotics [RS09], mathematics [CS07], physics [VCBJ<sup>+</sup>95], and philosophy [Sum05]. However not only its interdisciplinarity stresses the attractiveness of swarm behaviour as a research topic, but also the fact that although it has been on active research for more than six decades [CE54] many questions still remain unanswered.

Models of swarming can be split into two general classes:

1. continuum / population-based / Eulerian / macroscopic models,
2. discrete / individual-based / Lagrangian / microscopic models.

Models of first class treat the swarm as a whole and describe the evolution of the swarm's population density in space and time. Consequently, they offer the possibility to get insight into general swarm characteristics, but do not allow to distinguish between individual swarm members.

Discrete models of swarming – in contrast – allow this differentiation. They depict the change in position, velocity and other properties of each swarm member individually. For this purpose, a discrete model considers generic rules that describe either the sociological behaviour of animals or a task that needs to be fulfilled. Those rules directly affect the attraction, repulsion and orientation behaviour of the individuals. As popular examples serve the rules of the so-called Boids model [Rey87] which we illustrate in Figure 3.1. The integration of such rules in equations of motion allows to describe the temporal evolution of the individual swarm members. Therefore, the estimation of a swarm's state for a specific point in time requires to solve a system of ordinary differential equations (ODEs). Discrete models of swarming usually describe the velocity

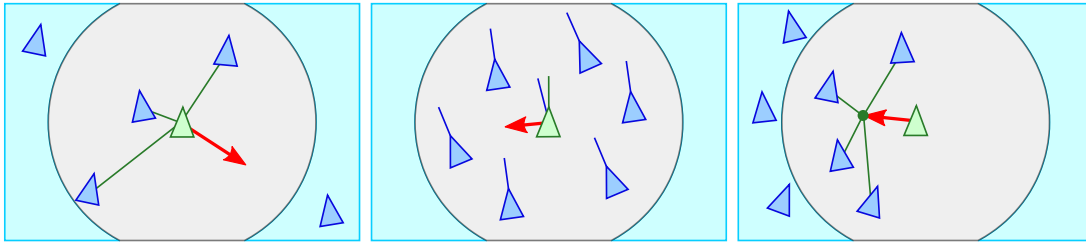


Figure 3.1: Illustration of the behavioural rules used in the Boids model [Rey87] (images based on [Rey07]). **Left:** *separation*, i.e. repulsion from the neighbours. **Centre:** *alignment*, i.e. orientation towards the average heading of the neighbours. **Right:** *cohesion*, i.e. attraction towards the centre of gravity of the neighbourhood.

or the acceleration of individuals and are classified as first- and second-order models accordingly. Well-known first-order models of swarming are defined in [GP03, GP04, CKJ<sup>+</sup>02, SGBW10, Yan10b]. On the other hand, examples for second-order models can be found in [Rey87, KE95, VCBJ<sup>+</sup>95, DCBC06, CS07, Yan10a, Gaz13, SP15]. For a recent review of discrete models of swarming, we refer the reader to [VZ12, YBEM10, FS13] and the references therein. Especially [FS13] provides a compact but comprehensive comparison of existing models.

Nowadays, discrete models are extensively used in the field of *optimisation*, since the heuristic character of the models is well-suited to approximate solutions for difficult optimisation problems. In this case, popular representatives of discrete models are ant colony optimization (ACO) [CDM91] and particle swarm optimization (PSO) [KE95].

Contrarily, the number of cases in which models of swarming have been used for the purpose of *modelling* problems is relatively small. This holds, in particular, for the domain of image analysis. Existing approaches deal with image halftoning [SGBW10], colour correction [SAF<sup>+</sup>14], segmentation [LT99], contour detection [KHD13], boundary identification and tracking [MB04, TS05], and the detection of fibre pathways [ARRM14]. Most of these modelling applications are fairly new and show convincing performance. However, all these approaches have in common, that they focus on a specific application and do not exploit the genericity behind the models of swarming.

**Contributions of this Chapter.** Motivated by these recent encouraging results, the goal of this chapter is to present novel applications of discrete first-order models of swarming in image analysis. To this end, we define behavioural rules for four fairly different image processing problems: grey scale quantisation, contrast enhancement, line detection with the Hough transform, and coherence enhancement. In all scenarios, we use essentially the same model and modify only some of its features. This emphasises the versatility and genericity of models of swarming.

**Structure of the Chapter.** Section 3.2 reviews the modelling of swarm behaviour in a discrete setup. We present our different behavioural rules and discuss related potential energies and forces. Furthermore, we discuss some model char-

acteristics and a time discretisation. Section 3.3 adapts this modelling framework to four different applications in image processing and shows experimental results. Section 3.4 summarises our contributions and gives an outlook to future work.

## 3.2 Discrete Modelling of Swarm Behaviour

### 3.2.1 Basic Notations and Definitions

We consider a set

$$S = \{i \mid i = 1, \dots, N\}, \quad (3.1)$$

called *swarm*, which is composed of  $N$  *agents*. In the following, we use the terms *agent*, *particle* and *individual* interchangeably. By  $\mathbf{x}_i \in \mathbb{R}^n$  we denote the position of an agent, whereas  $\partial_t \mathbf{x}_i \in \mathbb{R}^n$  describes its velocity. Both, the particle position and its velocity are functions over time  $t \in [0, \infty)$ :

$$\mathbf{x}_i = \mathbf{x}_i(t), \quad \partial_t \mathbf{x}_i = \partial_t \mathbf{x}_i(t). \quad (3.2)$$

Based on this, the position of a swarm  $\mathbf{x} \in \mathbb{R}^{N \times n}$  and its velocity  $\partial_t \mathbf{x} \in \mathbb{R}^{N \times n}$  are given by

$$\mathbf{x} = \mathbf{x}(t) = \begin{pmatrix} \mathbf{x}_1 \\ \mathbf{x}_2 \\ \vdots \\ \mathbf{x}_N \end{pmatrix}, \quad \partial_t \mathbf{x} = \partial_t \mathbf{x}(t) = \begin{pmatrix} \partial_t \mathbf{x}_1 \\ \partial_t \mathbf{x}_2 \\ \vdots \\ \partial_t \mathbf{x}_N \end{pmatrix}. \quad (3.3)$$

If the agents are intended to have a limited field of perception, many discrete models such as [Rey87] make use of a disk-shaped neighbourhood set of agents (cf. Figure 3.1). For agent  $i$  its neighbourhood set of radius  $\delta > 0$  is given by

$$\mathcal{N}_{i,\delta} = \mathcal{N}_{i,\delta}(t) = \{j \in S \mid j \neq i \text{ and } |\mathbf{x}_i - \mathbf{x}_j| \leq \delta\}, \quad (3.4)$$

where  $|\cdot|$  denotes the euclidean norm. If all neighbourhoods  $\mathcal{N}_{i,\delta}$  contain all swarm mates  $j$  for all times  $t$ , a model is said to be *global*. Otherwise, it is called *local*.  $|\mathcal{N}_{i,\delta}|$  represents the number of neighbours of agent  $i$  at a specific point in time.

### 3.2.2 Potential Energies and Forces

To describe a desired collective behaviour, discrete models of swarming define update rules for the positions and velocities of the individuals. These rules include effects based on attractive, repulsive, and orientating behaviour among agents [CKJ+02, CS07, Rey87, VCBJ+95], as well as on the environment [KE95], or a combination of both [Gaz13, SGBW10]. In this chapter, we restrict ourselves to the treatment of attraction and repulsion among the agents.

The influence of a swarm mate on an individual is described by a pairwise function  $U : \mathbb{R}^n \rightarrow \mathbb{R}$  that denotes the *potential energy*. Another common name for  $U$  is *potential function*. The total potential energy of a particle  $i \in S$  is given by

$$E_i(\mathbf{x}) = \sum_{j \in S \setminus \{i\}} U(\mathbf{x}_i - \mathbf{x}_j). \quad (3.5)$$



Obviously, the total potential energy (3.5) only depends on the relative position of a particle to its fellows. Accordingly, the total potential energy of the swarm  $S$  reads

$$E(\mathbf{x}) = \frac{1}{2} \sum_{i \in S} E_i(\mathbf{x}), \quad (3.6)$$

where the division by two implements the idea of counting each mutual particle relation only once.

Another important factor is the *potential force* that acts on an individual  $i$ . The potential force is defined as the negative gradient of  $U$  in direction of  $\mathbf{x}_i$ , i.e.  $-\nabla_{\mathbf{x}_i} U : \mathbb{R}^n \rightarrow \mathbb{R}^n$ .

### 3.2.3 Discrete First-Order Models of Swarming

Discrete first-order models of swarming apply potential forces to express the evolution of a swarm in time. Assuming that the initial state at time  $t = 0$  is known, the particle velocities are given by

$$\partial_t \mathbf{x}_i(t) = - \sum_{j \in S \setminus \{i\}} \nabla_{\mathbf{x}_i} U(\mathbf{x}_i - \mathbf{x}_j), \quad \text{for } i \in S. \quad (3.7)$$

Such models can be interpreted as a physically simplified adaptation of Newton's second law. Gazi and Passino [GP04] state that (3.7) describes a second-order model with an individual particle mass  $m_i \approx 0$  and a damping term  $-\partial_t \mathbf{x}_i$ . Furthermore, the model (3.7) implements a gradient descent on the potential energy of the swarm, i.e. we have

$$\partial_t \mathbf{x}_i(t) = -\nabla_{\mathbf{x}_i} E_i(\mathbf{x}) = -\nabla_{\mathbf{x}_i} E(\mathbf{x}), \quad \text{for } i \in S. \quad (3.8)$$

In accordance with the results on nonsymmetric nonlocal evolutions presented by Cárdenas [Cár18], we extend (3.7) to

$$\partial_t \mathbf{x}_i(t) = - \sum_{j \in S \setminus \{i\}} w(\mathbf{x}_i, \mathbf{x}_j) \cdot \nabla_{\mathbf{x}_i} U(\mathbf{x}_i - \mathbf{x}_j), \quad \text{for } i \in S. \quad (3.9)$$

This swarming model introduces support for nonsymmetric, anisotropic evolutions with the help of a non-negative weighting function  $w : \mathbb{R}^n \times \mathbb{R}^n \rightarrow \mathbb{R}_0^+$ .

It is common practice to define potential forces in terms of an *attraction-repulsion function*

$$-\nabla_{\mathbf{x}_i} U(\mathbf{y}) := -(k_a(|\mathbf{y}|) - k_r(|\mathbf{y}|)) \cdot \mathbf{y}, \quad \text{where } \mathbf{y} \in \mathbb{R}^n. \quad (3.10)$$

The non-negative kernel functions  $k_a : \mathbb{R}_0^+ \rightarrow \mathbb{R}_0^+$  and  $k_r : \mathbb{R}_0^+ \rightarrow \mathbb{R}_0^+$  describe the magnitude of attraction and repulsion amongst two swarm members. In this

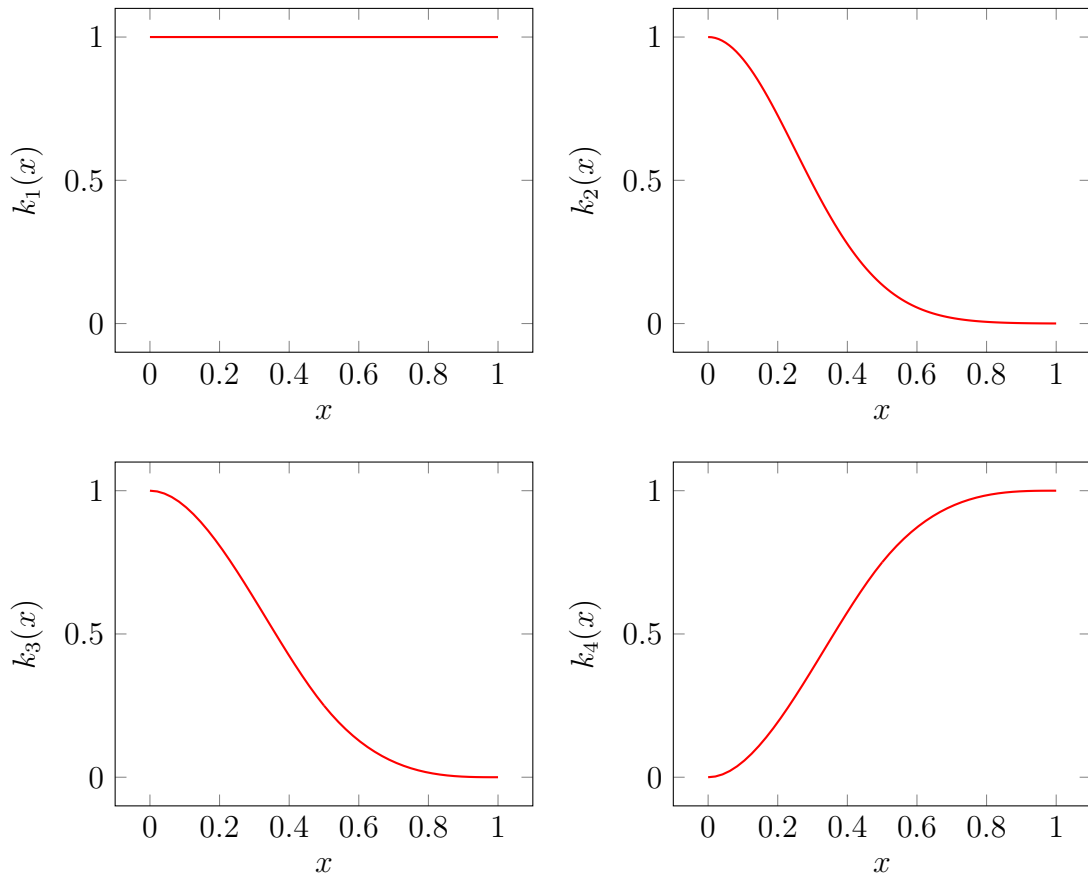


Figure 3.2: Plots of our kernel functions defined in (3.11)-(3.14) for  $x \in [0, 1]$  and  $c = 0.125$ .

chapter, we employ the kernel functions

$$k_1(x) := 1, \quad (3.11)$$

$$k_2(x) := \exp\left(-\frac{x^2}{c^2}\right), \quad \text{with } c \in \mathbb{R} \setminus \{0\}, \quad (3.12)$$

$$k_3(x) := \begin{cases} 1 - 6x^2 + 6x^3, & \text{if } 0 \leq x < \frac{1}{2}, \\ 2 \cdot (1 - x)^3, & \text{if } \frac{1}{2} \leq x < 1, \\ 0, & \text{else,} \end{cases} \quad (3.13)$$

$$k_4(x) := 1 - k_3(x), \quad (3.14)$$

with  $k_1, k_2, k_3, k_4 \in [0, 1]$ . While (3.11) and (3.12) represent a constant function and a Gaussian function, (3.13) denotes a scaled-cubic B-spline which can be understood as an approximation of a Gaussian in  $[0, 1]$  – however – with compact support. The last function (3.14) is a flipped version of (3.13). We provide plots of all kernel functions for  $x \in [0, 1]$  in Figure 3.2.

From (3.7) and (3.9) it becomes clear that discrete first-order models only depend on potential forces and do not necessarily require an expression for the potential energy. Actually, there even exist models for which no closed-form expression

of a potential energy potential energy can be formulated. However, in case it is possible to find a valid formula for the potential energy, this allows an elegant interpretation of the underlying dynamical system in terms of (3.8). For this purpose, let us take a look at two potential forces

$$-\nabla_{\mathbf{x}_i} U(\mathbf{x}_i - \mathbf{x}_j) = -k_1(|\mathbf{x}_i - \mathbf{x}_j|) \cdot (\mathbf{x}_i - \mathbf{x}_j), \quad (3.15)$$

$$-\nabla_{\mathbf{x}_i} U(\mathbf{x}_i - \mathbf{x}_j) = k_2(|\mathbf{x}_i - \mathbf{x}_j|) \cdot (\mathbf{x}_i - \mathbf{x}_j), \quad (3.16)$$

for  $i \in S$ , which we use in our experiments later on. Combining (3.7) and (3.10), we observe that (3.15) describes an attractive swarming process, while (3.16) models repulsive behaviour amongst the swarm members. In case of the attractive model (3.15), the formula for the potential function reads

$$U(\mathbf{x}_i - \mathbf{x}_j) = \frac{1}{2} \cdot |\mathbf{x}_i - \mathbf{x}_j|^2. \quad (3.17)$$

Together with (3.5) and (3.6), we see that the potential forces (3.15) induce a minimisation of the  $\ell^2$ -distance between the swarm members. For our repulsive model (3.16), we end up in the potential energy

$$U(\mathbf{x}_i - \mathbf{x}_j) = \frac{c^2}{2} \cdot \exp\left(-\frac{|\mathbf{x}_i - \mathbf{x}_j|^2}{c^2}\right), \quad \text{with } c \in \mathbb{R} \setminus \{0\}. \quad (3.18)$$

Consequently, the total potential energy of a particle – and the swarm – is minimised when the  $\ell^2$ -distance between the swarm members is maximal.

### 3.2.4 Time Discretisation

Since we cannot expect to find an analytical solution to the dynamical system (3.9), we have to approximate it numerically on the computer. This requires to discretise it in time.

Let  $\alpha > 0$  denote some time step size, and let  $t_k := k\alpha$ . Moreover, we abbreviate  $\mathbf{x}_i(t_k)$  by  $\mathbf{x}_i^k$ . The simplest time discretisation of Equation 3.9 approximates the time derivative by its forward difference:

$$\partial_t \mathbf{x}_i(t) \approx \frac{\mathbf{x}_i^{k+1} - \mathbf{x}_i^k}{\alpha}. \quad (3.19)$$

This turns (3.9) into the following explicit update scheme:

$$\mathbf{x}_i^{k+1} = \mathbf{x}_i^k - \alpha \cdot \sum_{j \in S \setminus \{i\}} w(\mathbf{x}_i^k, \mathbf{x}_j^k) \cdot \nabla_{\mathbf{x}_i^k} U(\mathbf{x}_i^k - \mathbf{x}_j^k) \quad (k = 0, 1, \dots) \quad (3.20)$$

with some appropriate initialisation  $\mathbf{x}_i^0$  for all  $i \in S$ .

If we restrict the interactions of agent  $i$  to its  $\delta$ -neighbourhood  $\mathcal{N}_{i,\delta}^k = \mathcal{N}_{i,\delta}(t_k)$  from (3.4), we can replace (3.20) by the local update rule

$$\mathbf{x}_i^{k+1} = \mathbf{x}_i^k - \alpha \cdot \sum_{j \in \mathcal{N}_{i,\delta}^k} w(\mathbf{x}_i^k, \mathbf{x}_j^k) \cdot \nabla_{\mathbf{x}_i^k} U(\mathbf{x}_i^k - \mathbf{x}_j^k). \quad (3.21)$$

It is well-known from the theory of numerical methods for differential equations that such explicit schemes may require a fairly small time step size  $\alpha$  in order to be stable [LeV07], in particular if the right hand side fluctuates strongly w.r.t. its argument.<sup>1</sup>

### 3.3 Application to Image Processing Problems

Discrete first-order models of swarming allow a new interpretation of image processing problems and offer an elegant way to solve them. We demonstrate how to model grey scale quantisation, contrast enhancement, line detection, and coherence enhancement in terms of the previously discussed model. This includes the definition of swarming agents and corresponding potential forces for each problem in an appropriate way.

For our first three settings, we assume the input to be given by a *digital grey scale image*  $f$  which is discrete in both, its domain and its codomain:

$$f : \{1, \dots, n\} \times \{1, \dots, m\} \rightarrow \{0, \dots, 255\}. \quad (3.22)$$

The domain consists of  $n$  equally spaced pixels in  $x$ -direction and  $m$  pixels in  $y$ -direction. The image  $f$  maps each pixel position to an eight bit grey value from the set  $\{0, \dots, 255\}$ .

For such a two-dimensional image, the *histogram*  $h[f](u)$  specifies the frequency of each grey value  $u \in \{0, \dots, 255\}$ . Accordingly,  $h[f]$  is a one-dimensional function from  $\{0, \dots, 255\}$  to  $\mathbb{N}_0$ .

In our last application, coherence enhancement, we assume that  $f$  is given by the one-dimensional function

$$f : \{1, \dots, mn\} \rightarrow [0, 1], \quad (3.23)$$

which maps a pixel  $i$  to its corresponding grey value  $f(i)$ . In contrast to (3.22), the grey values are scaled by a factor of  $1/255$ .

#### 3.3.1 Grey Scale Quantisation

The discretisation of the codomain of an image is called *quantisation*. Obviously, the number of different greyscales in an image determines how expensive it is to store them: While 256 different values require a full byte, 8 values can be encoded already with 3 bits. Since humans cannot distinguish many greyscales, one can compress image data without severe visual degradations by reducing the number of quantisation levels.

To design a model of swarming for obtaining a coarser quantisation of some digital greyscale image  $f$ , we proceed as follows. We consider its histogram  $h[f]$  and identify some histogram value  $h[f](u) = c_u$  with  $c_u$  agents sharing the same position  $x_i = u$ . Thus, we have a one-dimensional model of swarming. Note that multiple agents that share the same position have to undergo the same joint

---

<sup>1</sup>If this becomes too time-consuming, one can also consider more efficient, so-called implicit schemes [LeV07]. However, they require to solve linear or nonlinear systems of equations.

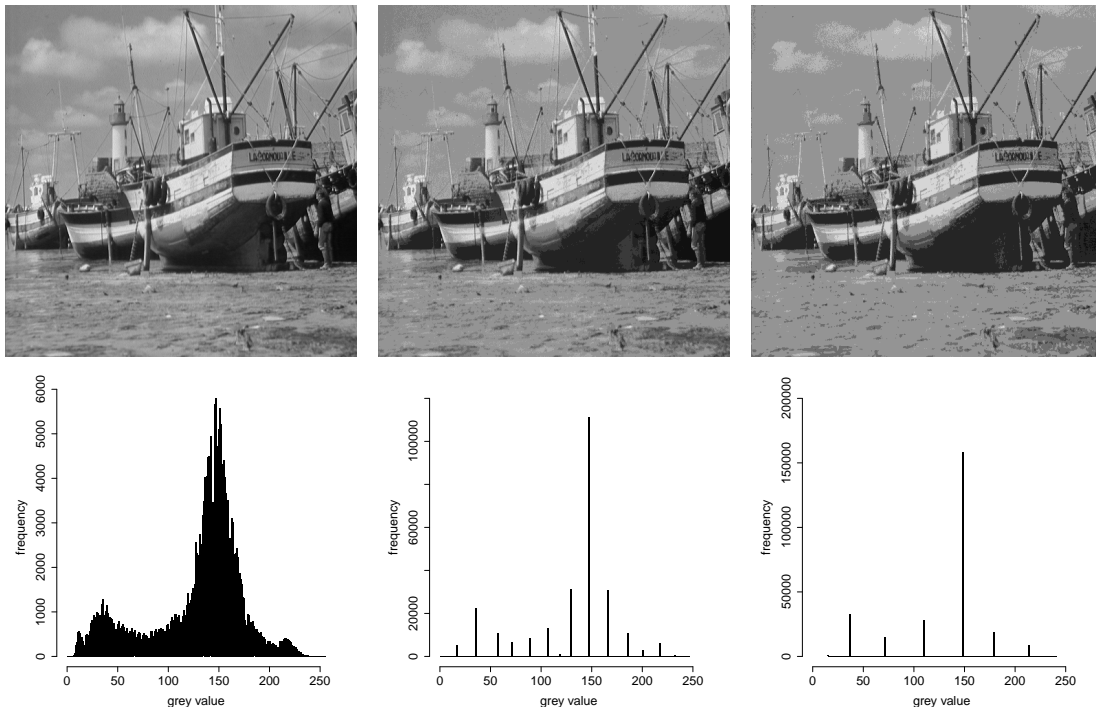


Figure 3.3: Swarm-based image quantisation. **Top, from left to right:** (a) Original image from [Sig15],  $512 \times 512$  pixels,  $q = 255$  greyscales. (b) Swarm-based quantisation with  $\delta = 8$ , yielding  $q = 16$  greyscales. (c)  $\delta = 16$ ,  $q = 8$  greyscales. **Bottom, from left to right:** (d)–(f) Corresponding histograms.

motion. This reduces the computational complexity in a substantial way: The computational effort becomes proportional to the number of greyscales instead of the number of pixels.

In order to cluster multiple quantisation levels into a single level, we use the linear attraction force from (3.15). As we will see below, it makes sense to localise the interaction to a  $\delta$ -neighbourhood, which requires the update scheme (3.21). We set the weighting function  $w$  to a constant value of 1.

In our quantisation experiments we have chosen  $\alpha = 10^{-5}$ . This leads to a stable steady state solution after at most  $4 \cdot 10^4$  iterations. For a  $512 \times 512$  image, this can be accomplished in far less than one minute on a single core of a standard PC. Figure 3.3 illustrates the effect of our model of swarming for different  $\delta$  values. We observe that increasing  $\delta$  reduces the number  $q$  of quantisation levels. Interestingly there seems to be an almost inverse relation, such that  $2\delta q$  is roughly equal to the length of the original greyscale interval (255 in our case). This suggests that our model of swarming clusters the grey scales into  $q$  bins of approximately<sup>2</sup> the same size  $2\delta$ . Note that the interval length  $2\delta$  is the diameter of the neighbourhood  $\mathcal{N}_{i,\delta}$ . Hence, the model of swarming can be interpreted and handled in a very intuitive way.

<sup>2</sup>It is clear from the structure of our approach and the experiments that the quantisation levels depend on the actual image histogram and are not necessarily equidistant.

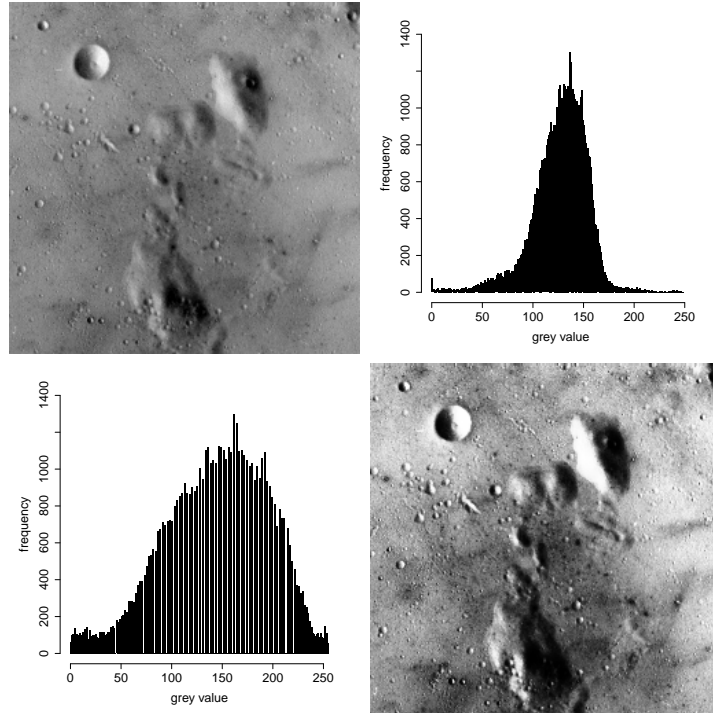


Figure 3.4: Swarm-based contrast enhancement. **(a) Top left:** Moon surface image from [Sig15],  $256 \times 256$  pixels. **(b) Top right:** Its histogram. **(c) Bottom left:** After swarm-based histogram enhancement with  $c = 1$ , and  $2 \cdot 10^6$  iterations with step size  $\alpha = 10^{-3}$ . **(d) Bottom right:** Enhanced image using the grey values from the transformed histogram.

### 3.3.2 Contrast Enhancement

The contrast of an image is characterised by the modulus of the difference between the greyvalues of neighbouring pixels. For recognising interesting image structures, their contrast should be sufficiently high. This may require some preprocessing that enhances the image contrast.

Let us now adapt our model of swarming to this application. To this end, it is sufficient to find a mapping of the greyvalues that yields a better contrast. As before, we consider the histogram  $h[f]$  of the image  $f$ , and we assign  $c_u$  agents to a grey value  $u$  if  $h[f](u) = c_u$ . However, since we want to increase the global contrast this time, we use the global explicit scheme (3.20), and we equip it with the repulsion forces from (3.16). Again, we set the weighting function  $w$  to a constant value of 1. Moreover, we employ reflecting boundary conditions to prevent that agents leave the admissible greyscale range  $[0, 255]$ . For  $t \rightarrow \infty$ , the swarm converges to a steady state distribution, where the grade of contrast enhancement grows with the repulsion parameter  $c$ . Once the histogram is enhanced, one simply replaces the grey values of the image by their transformed values.

Figure 3.4 illustrates this procedure, where the evolution reaches a steady state. We observe a clear visual contrast improvement of the test image. This is also confirmed quantitatively by its standard deviation, which has increased from 27.74 to 56.86.

### 3.3.3 Line Detection

Our third application scenario for models of swarming is concerned with another important image processing problem, the detection of lines. Our goal is to improve a classical method which is based on the so-called Hough transform [DH72].

The basic idea behind line detection with the Hough transform is as follows. For some greyscale image  $f$ , one searches for locations that may lie on significant lines by computing the gradient magnitude  $|\nabla f|$ . For a digital image, this requires finite difference approximations. A location is significant if its gradient magnitude exceeds a certain threshold  $t_g$ . In a next step, the line candidate pixels vote for all lines that pass through them. All lines through a pixel  $(x, y)$  satisfy the normal representation

$$\rho = x \cdot \cos \theta + y \cdot \sin \theta, \quad (3.24)$$

where  $\theta$  denotes the angle between the line normal and the  $x$ -axis, and  $\rho$  is the distance to the origin. Thus, a candidate point is mapped to a trigonometric curve  $\rho(\theta)$  in the Hough space  $(\theta, \rho)$ . If  $n$  candidate points lie on a line with parameters  $(\theta, \tilde{\rho})$ , then their corresponding  $n$  trigonometric curves in Hough space intersect in  $(\theta, \tilde{\rho})$ . Therefore, one can find lines in the input image  $f$  by searching for clustering points in its Hough space: One discretises the Hough space  $(\theta, \rho)$ , and each trigonometric curve votes for all cells that it crosses. The cells with the most votes characterise the most significant lines in the original image. Typically one finds these clustering points by applying a threshold  $t_a$  on the votes in Hough space.

While this sounds nice in theory, in practice it is not easy to find appropriate thresholds that avoid false negatives and false positives. Also the bin size of the discrete Hough space is problematic: If the discretisation is too fine, it is unlikely that many votes will fall in the same cell. If it is too coarse, the line parameters are prone to imprecisions.

As a remedy, we propose the following procedure. First, we consider a relatively fine discretisation in Hough space and threshold the votes. Afterwards we process the surviving votes with a swarm-based clustering. To this end, we set up  $n$  agents at every position  $(\rho, \theta)$  that received  $n$  votes. Note that in contrast to our clustering for quantisation – which took place in the one-dimensional histogram space – this is a two-dimensional clustering. In analogy to the quantisation setting, we use the linear attraction force (3.15) and  $w = 1 = \text{const.}$  within the localised update scheme (3.21), and compute its steady state.

Figure 3.5 shows how this works in a real-world setting. We observe that the classical Hough transform suffers from the fact that lines in the image cluster in several adjacent cells in Hough space. As a consequence, we obtain a bundle of almost parallel lines instead of a single line. Our swarm-based clustering in Hough space is well-suited to solve this problem, since votes from the neighbours move towards the local centroids. In this way they sharpen the clusters and avoid multiple almost parallel lines.

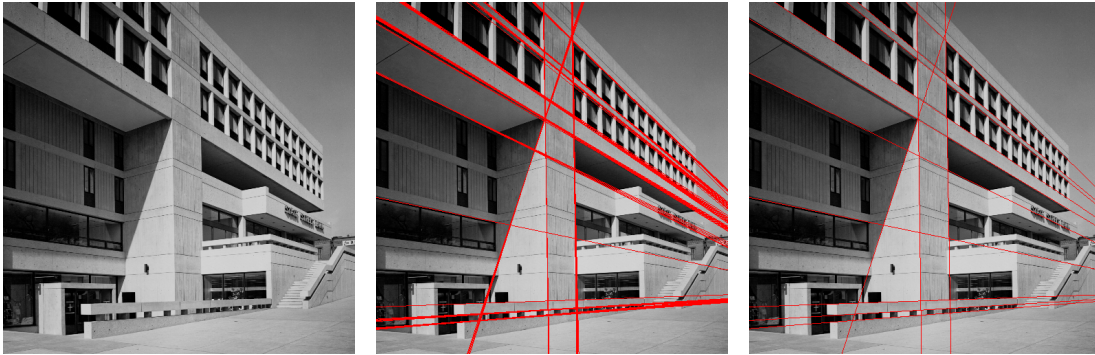


Figure 3.5: Swarm-based line detection. **(a) Left:** Test image,  $512 \times 512$  pixels. **(b) Middle:** 71 lines detected with the Hough transform. ( $t_g = 19$ ,  $t_a = 244$ ). **(c) Right:** 13 lines detected with the Hough transform with swarm-based post-processing ( $t_g = 19$ ,  $t_a = 244$ ,  $\delta = 5$ ,  $\alpha = 10^{-4}$ , 300 iterations).

### 3.3.4 Coherence Enhancement

With our last application, we demonstrate how the theory of *swarm models* can be used to construct a coherence enhancing image filter. The aim of the latter is the filling-in of missing information in a given image while preserving its original structure (cf. Figure 3.6). Challenging conditions during image acquisition often lead to noise or gaps in the image structure. In those cases coherence enhancing filters represent an important and necessary preprocessing step for many computer vision applications. Filters which fulfill the previously mentioned criteria



Figure 3.6: Idea of coherence enhancing image filtering. **Left:** input image showing an interrupted curve. **Right:** filtered image with closed gap.

can e.g. be found in [Wei99], [Wei03], and [LPZ12]. All of these approaches have in common that they employ the structure tensor [FG87] to estimate the so-called coherence orientation in a first step. Afterwards, this information is used in the main filtering phase.

We present a novel two-step approach for coherence enhancing image filtering which is purely based on our theory of swarm models. Therein, we follow the idea that grey values should propagate orthogonally to the gradient direction in order to increase coherence. For this purpose, we apply a swarm model and estimate the *coherence orientation* in the image first. Secondly, we define a *grey value evolution* which makes the grey values move in latter orientation. In both steps, we employ our localised update scheme (3.21) in combination with a time



step size  $\alpha$  that fulfils

$$0 \leq \alpha \leq \frac{1}{\max_i \sum_{j \in \mathcal{N}_{i,\delta}} w(\mathbf{x}_i^k, \mathbf{x}_j^k)}. \quad (3.25)$$

These conditions guarantee stability of our numerical scheme and have been derived in accordance with the theory presented in [Cár18].

### Step 1: Coherence Orientation

The first swarm model describes the evolution of the gradient vector field

$$\nabla f_\sigma : \{1, \dots, mn\} \rightarrow [-1, 1]^2, \quad (3.26)$$

which maps a pixel  $i$  to its corresponding gradient vector  $\nabla f_{\sigma,i} = \nabla f_\sigma(i)$ . We estimate the initial  $\nabla f_\sigma$  from a pre-smoothed version of the input image  $f$ . For pre-smoothing  $f$ , we use a Gaussian kernel with standard deviation  $\sigma$ .

Based on this, we consider the two-dimensional evolution

$$\partial_t \mathbf{x}_i = - \sum_{\substack{\pm \mathbf{x}_j \\ j \in \mathcal{N}_i}} w(\mathbf{x}_i, \mathbf{x}_j) k_3(|\mathbf{x}_j - \mathbf{x}_i|) (\mathbf{x}_i - \mathbf{x}_j) \quad (3.27)$$

$$\mathbf{x}_i(0) = \nabla f_{\sigma,i} \quad (3.28)$$

which describes the local alignment of the gradient vector  $\nabla f_{\sigma,i}$  in each pixel  $i$ , being described by the corresponding particle position  $\mathbf{x}_i$ . The particle velocity  $\dot{\mathbf{x}}_i$  depends on all gradient directions  $\mathbf{x}_j$ , as well as their negative counterparts  $-\mathbf{x}_j$ , of all mates  $j$  that lie within a disk-shaped neighbourhood  $\mathcal{N}_i$  of radius  $d_1$  around pixel  $i$  in the image plane. The weights

$$w(\mathbf{x}_i, \mathbf{x}_j) = \begin{cases} 0, & \text{if } \mathbf{x}_j^T \mathbf{x}_i \leq 0, \\ \frac{|\mathbf{x}_j|^2 + \varepsilon}{\varepsilon} \cdot \mathbf{x}_j^T \mathbf{x}_i, & \text{if } 0 < \mathbf{x}_j^T \mathbf{x}_i \leq \varepsilon, \\ |\mathbf{x}_j|^2 + \varepsilon, & \text{else,} \end{cases} \quad (3.29)$$

ensure that  $\mathbf{x}_i$  only aligns with those vectors  $\pm \mathbf{x}_j$ , which deviate from its own orientation by less than  $\pi/2$ . As a consequence, at most one of both vectors  $\mathbf{x}_j$  and  $-\mathbf{x}_j$  can influence the behaviour of  $\mathbf{x}_i$  at a time. Additionally, the weights  $w(\mathbf{x}_i, \mathbf{x}_j)$  approximate the idea that vectors with large magnitude should guide those with small modulus, but not the other way round. This asymmetry is implemented by multiplication with the term  $|\mathbf{x}_j|^2 + \varepsilon$ . By applying (3.13) as kernel function, small differences  $|\mathbf{x}_j - \mathbf{x}_i|$  lead to higher kernel weights, whereas large ones induce small kernel weights (see Figure 3.2).

Using the steady state  $\mathbf{x}^*$  of the initial value problem (3.27)-(3.28), we define the local dominant gradient orientation of pixel  $i$  as

$$\nabla \tilde{f}_i := (\nabla \tilde{f}_{i,1}, \nabla \tilde{f}_{i,2})^T = \begin{cases} (0, 0)^T, & \text{if } |\mathbf{x}^*| = 0, \\ \frac{\mathbf{x}^*}{|\mathbf{x}^*|}, & \text{else.} \end{cases} \quad (3.30)$$

Then, the desired coherence orientation is given by the local dominant tangent orientation

$$\nabla^\perp \tilde{f}_i := (-\nabla \tilde{f}_{i,2}, \nabla \tilde{f}_{i,1})^T. \quad (3.31)$$

### Step 2: Grey Value Evolution

In the second step, a swarm model is used to let the grey values of the input image  $f$  propagate in direction of  $\nabla^\perp \tilde{f}$  (as given in (3.31)). We define a one-dimensional evolution in which every particle  $i$  is connected to a pixel  $i$  in the image plane and moves in the tonal domain:

$$\partial_t u_i = - \sum_{j \in \mathcal{N}_i} w_{i,j} \cdot k_4(a \cdot |u_i - u_j|) \cdot (u_i - u_j) \quad (3.32)$$

$$u_i(0) = f_i. \quad (3.33)$$

Accordingly,  $u_i = u_i(t)$  denotes the enhanced grey value of pixel  $i$  at time  $t$ . Similar to the first step, the neighbourhood  $\mathcal{N}_i$  contains all neighbours of particle  $i$  that lie within euclidean distance  $d_2$  in the image plane. Within (3.32), we represent the weighting function  $w$  from (3.9) as non-negative constants

$$w_{i,j} = \left| \nabla^\perp \tilde{f}_i^T \mathbf{n}_{i,j} \right|^b \cdot \left| \nabla^\perp \tilde{f}_j^T \mathbf{n}_{j,i} \right|^b, \quad (3.34)$$

where  $\mathbf{n}_{i,j}$  denotes the normal vector pointing from pixel  $i$  to pixel  $j$  in the image plane. Consequently, the weights implement the idea, that – during grey value evolution – directions similar to the coherence orientation should be favoured. The parameter  $b > 0$  offers the possibility to control the allowed directional offset, which determines the anisotropy of the evolution.

In order to enforce the evolution across edges, the value of the kernel function  $k_4$  (as defined in (3.14)) increases with larger grey value differences  $|u_i - u_j|$ . In the context of (3.32), the scalar value  $a \geq 0$  steers the non-linearity of the model (large values of  $a$  approximate a linear model), and by this the influence of local grey value differences. The steady state  $\mathbf{u}^*$  of our grey value evolution denotes the coherence enhanced image.

### Experiments

Subsequently, we demonstrate the efficacy of our coherence enhancing image filter, and apply the filter to the greyscale fingerprint image in Figure 3.7.

**Colour Coding.** However, before discussing the results, let us briefly introduce the colour coding shown in Figure 3.7. In the following, we use it to illustrate the orientation of vectors, where we assume w.l.o.g. that the latter can be understood as an angle between 0 and  $\pi$ . This makes sense because we are not interested in the sign of a vector and thus don't distinguish between polar angles  $\theta \in [0, \pi)$  and  $\theta + n \cdot \pi$  ( $n \in \mathbb{Z}$ ).

**Time Step Size.** For all our experiments we choose – in accordance with (3.25) – constant time step sizes  $\alpha$ , namely

$$\alpha = \frac{1}{2 \cdot |\mathcal{N}_i| \cdot (1 + \varepsilon)} \quad (3.35)$$

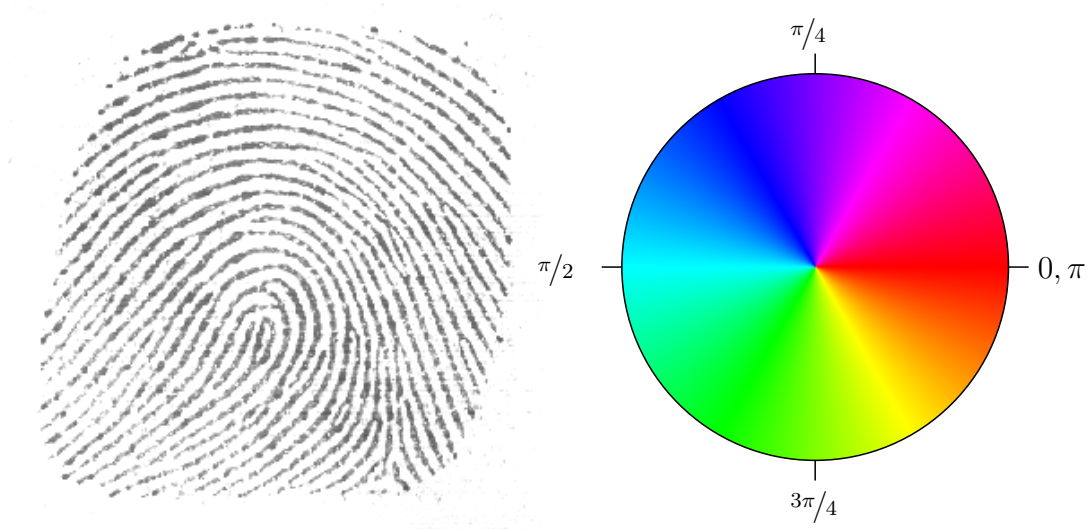


Figure 3.7: **Left:** Greyscale image of a fingerprint with  $300 \times 300$  pixels (high resolution version of the image used in [WWS06]). **Right:** Colour coding as used for illustrating the local dominant gradient orientation.

during the estimation of the local dominant gradient direction and

$$\alpha = \frac{1}{2 \cdot \max_i \sum_{j \in \mathcal{N}_i} w_{i,j}} \quad (3.36)$$

in the second step for grey value evolution. Note, that the weights  $w_{i,j}$  are constants (cf. (3.34)) and the maximum term in (3.36) can be precomputed easily.

**Stopping Criterion.** Our stopping criterion for the explicit scheme reads in case of the local dominant gradient direction

$$\sum_{i=1}^{mn} |\dot{\mathbf{x}}_i^k|^2 < 10^{-5}. \quad (3.37)$$

For the grey value evolution we end up iterating, if it holds that

$$\sum_{i=1}^{mn} |\dot{u}_i^k|^2 < 10^{-4}. \quad (3.38)$$

**Computational Efficiency.** Each of both steps of our algorithm has a complexity of  $\mathcal{O}(N \cdot |\mathcal{N}_i| \cdot k)$ , where  $N$  denotes the number of particles – in our case this is equivalent to the number of pixels of  $f$  –, and  $|\mathcal{N}_i|$  represents the size of the neighbourhood of an particle (which is the same for all particles in our setup). The number of needed iterations is given by  $k$ .

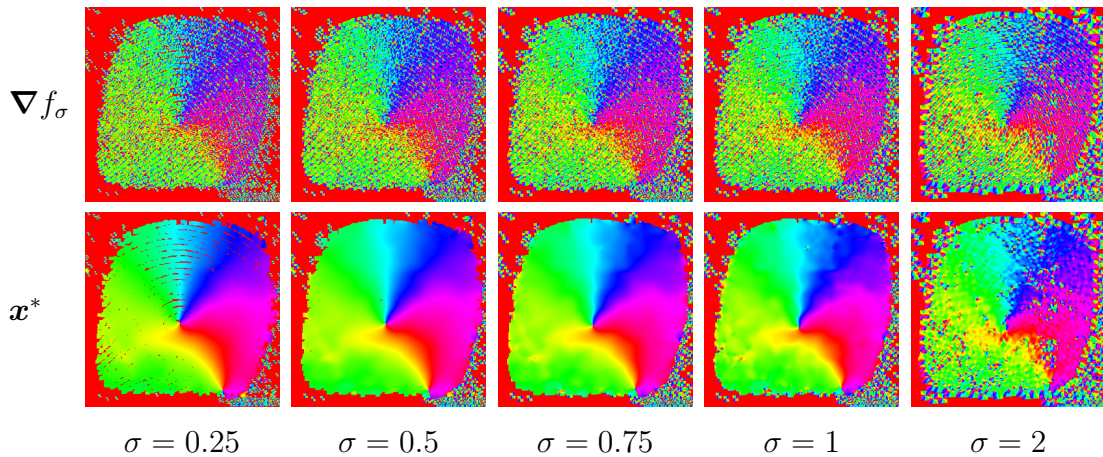


Figure 3.8: Influence of parameter  $\sigma$  on the orientation of the pre-smoothed gradient field  $\nabla f_\sigma$  and the steady state  $\mathbf{x}^*$  of the initial value problem (3.27)-(3.28).

**Parameter Selection and Influence.** During our experiments we make use of a grid size  $h = 1$ . Furthermore, we set  $d_1 = \sqrt{2}$ . Thus, the eight nearest neighbours of every pixel  $i$  in the image plane are considered in the first step. The value of  $\varepsilon$  (cf. (3.29)) is fixed to  $10^{-11}$ .

**Local Dominant Gradient Orientation.** In dependence on parameter  $\sigma$ , we present different results for the first step in Figure 3.8. The pictures show the orientation of the pre-smoothed gradient vectors, as well as the orientation of the local dominant gradient (steady state of the first evolution (3.27)-(3.28)). The illustrations make use of the previously mentioned colour coding (cf. Figure 3.7). Note, that at all positions with zero gradient magnitude the direction was set to zero (see e.g. the red areas in the surrounding of the fingerprint).

Our results show, that the original gradient field of  $f$  is quite noisy. This justifies the necessity of the first step of our algorithm, since a smooth vector field – describing the local dominant gradient – is essential for the subsequent grey value evolution.

From Figure 3.8 it also becomes clear, that an appropriate choice of  $\sigma$  allows to fill in directional information in regions with small or no gradient magnitude (see e.g. the differences between the results for  $\sigma = 0.25$  and  $\sigma = 0.5$ ). This is an important property of our model, since this allows us to smooth not only directly along the edges of an image, but also in between. On the other hand – and in accordance with the findings in [Wei99] – one can also observe, that with increasing value of  $\sigma$  cancellation artefacts for  $\nabla f_\sigma$  appear, which induce an irregular vector field  $\mathbf{x}^*$  (see e.g. the results for  $\sigma = 2$ ). This means that the value of  $\sigma$  should be chosen carefully and be as small as possible.

In the latter case, the steady state of (3.27)-(3.28) represents a smooth vector field (cf. Figure 3.8 for  $\sigma = 0.5$ ). However, note that close to the boundaries of large homogeneous regions irregularities will always appear (e.g. surroundings of the fingerprint). This is related to the fact, that the pre-smoothing might only affect

the region boundaries and might not influence its inner parts. By definition of the evolution in (3.27)-(3.29), the particle velocities within those areas are always zero. Consequently, the original gradient orientation – which is undetermined at those positions – can be arbitrary, and will not change (we set the angle to zero as mentioned before). This is highly likely to be in conflict with the orientation of surrounding regions. Fortunately, these artefacts are negligible since the interior of large homogeneous regions plays no role in the subsequent grey value evolution. From (3.32)-(3.33) it is clear, that the velocity in those areas is always zero and no transport of grey values takes place.

**Grey Value Evolution.** Based on our previous findings, we choose  $\sigma = 0.5$  to estimate a smooth local dominant gradient field (cf. Figure 3.8). In accordance with [Wei94], we select a sufficiently large neighbourhood radius  $d_2$  to approximate rotational invariance well. For all further experiments we fix  $d_2 = 5$ . Given this setup, we apply our grey value evolution for varying values of the parameters  $a$  and  $b$  as used in (3.32) and (3.34). We present the filtered images in Figure 3.9.

Parameter  $a$ , which steers the general amount of attractive forces between the particles, can be seen as the counterpart to the diffusion time in diffusion filters. As one can also see from Figure 3.9, higher values of  $a$  go along with an increased smoothing of the image.

On the other hand, parameter  $b$  allows to control how strict deviations from the local dominant tangent direction should be punished. In the end, this describes how much smoothing should be done in off-tangent direction. Consequently, low values of  $b$  induce a more blurred image, while with increasing value of  $b$  the anisotropy of the filter rises, leading to clearer structures in the image (cf. Figure 3.9).

Consequently, one can say that – depending on the input image – an adequate weighting of both parameters is important. In our case, we think the usage  $a = 0.2$  and  $b = 6$  offers a good compromise of smoothing and strictness about the smoothing direction.

When comparing the input image (cf. Figure 3.7) and our results in Figure 3.9 one can clearly see the efficacy of our suggested filter. Apparent gaps (e.g. at the bottom left or the top of the fingerprint) are closed and overall all lines in the image are smoothed in tangent direction.

## 3.4 Conclusions and Outlook

In this chapter, we show that discrete first-order models of swarming have a high potential in image processing that goes far beyond classical applications as tools for difficult *optimisation* tasks: By means of four proof-of-concept applications we have demonstrated their usefulness as powerful *modelling* methods. The fact that these applications serve fairly different goals underlines the genericity of the swarm-based paradigm: It is a highly versatile framework that can be adapted in an intuitive way to a broad spectrum of problems. Especially the application

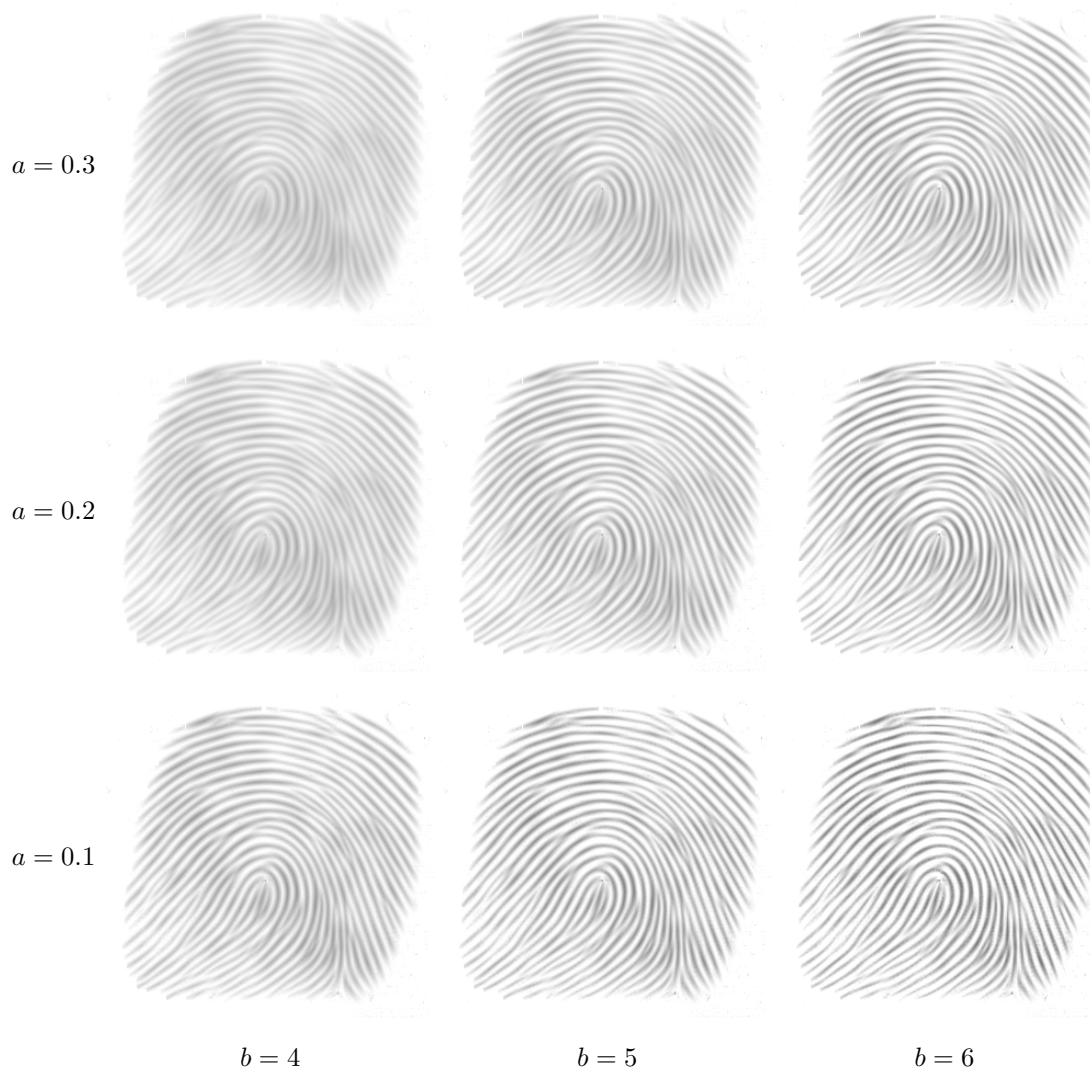


Figure 3.9: Influence of parameters  $a$  and  $b$  on the steady state  $\mathbf{u}^*$ .

as a coherence enhancing image filter illustrates well the potential of swarming models as a solution strategy for more complex tasks.

A possible starting point for future research might include the study of more efficient numerical algorithms, the evaluation of different ways to incorporate neighbourhood information (e.g. in terms of the weighting function  $w$  or the kernel functions  $k_a$  and  $k_r$ ), the equipment of our models with more problem-specific features, and a more extensive comparison to non-swarm based approaches. On top of that it makes also sense to study other models of swarming and apply them to further problems in the broad area of visual computing.

---

# Chapter 4

## Purely Repulsive Models and Backward Diffusion

“Alles ist Wechselwirkung.” [Eng.: Everything is interaction.]

---

Alexander von Humboldt, *Tagebücher der Amerikanische Reise*

### Contents

---

<b>4.1</b>	<b>Introduction</b>	<b>40</b>
<b>4.2</b>	<b>Model</b>	<b>43</b>
4.2.1	Motivation from Swarm Dynamics	43
4.2.2	Discrete Variational Model	44
<b>4.3</b>	<b>Theory</b>	<b>47</b>
4.3.1	General Results	47
4.3.2	Global Model	51
4.3.3	Relation to Variational Signal and Image Filtering	54
<b>4.4</b>	<b>Explicit Time Discretisation</b>	<b>56</b>
<b>4.5</b>	<b>Application to Image Enhancement</b>	<b>60</b>
4.5.1	Greyscale Images	60
4.5.2	Colour Images	62
4.5.3	Parameters	64
4.5.4	Related Work from an Application Perspective	65
<b>4.6</b>	<b>Conclusions and Outlook</b>	<b>68</b>
<b>4.A</b>	<b>Supplementary Material</b>	<b>69</b>
4.A.1	Derivations	69
4.A.2	Parameters for Local Contrast Enhancement	73

---

Main parts of this chapter base on our journal paper [BCWW20a], our conference publication [BCWW18], and our preprint [BCWW20b].

The inverse problem of backward diffusion is known to be ill-posed and highly unstable. Backward diffusion processes appear naturally in image enhancement and deblurring applications. It is therefore greatly desirable to establish a backward diffusion model which implements a smart stabilisation approach that can be used in combination with an easy to handle numerical scheme. So far, existing stabilisation strategies in literature require sophisticated numerics to solve the underlying initial value problem. We derive a class of space-discrete one-dimensional backward diffusion as gradient descent of energies where we gain stability by imposing range constraints. Interestingly, these energies are even convex. Furthermore, we establish a comprehensive theory for the time-continuous evolution and we show that stability carries over to a simple explicit time discretisation of our model. Finally, we confirm the stability and usefulness of our technique in experiments in which we enhance the contrast of digital greyscale and colour images.

## 4.1 Introduction

Forward diffusion processes are well-suited to describe the smoothing of a given signal or image. This process of blurring implies a loss of high frequencies or details in the original data. As a result, the inverse problem, backward diffusion, suffers from deficient information which are needed to uniquely reconstruct the original data. The introduction of noise due to measured data increases this difficulty even further. Consequently, a solution to the inverse problem – if it exists at all – is highly sensitive and heavily depends on the input data: Even the smallest perturbation in the initial data can have a large impact on the evolution and may cause large deviations. Therefore, it becomes clear that backward diffusion processes necessitate further stabilisation.

**Previous Work on Backward Diffusion.** Already more than 60 years ago, John [Joh55] discussed the quality of a numerical solution to the inverse diffusion problem given that a solution exists, and that it is bounded and non-negative. Since then, a large number of different regularisation methods have evolved which achieve stability by bounding the noise of the measured and the unperturbed data [TS96], by operator splitting [KW02], by Fourier regularisation [FXQ07], or by a modified Tikhonov regulariser [ZM11]. Hào and Duc [HD09] suggest a mollification method where stability for the inverse diffusion problem follows from a convolution with the Dirichlet kernel. In [HD11] the same authors provide a regularisation method for backward parabolic equations with time-dependent coefficients. Ternat et al. [TOD11] suggest low-pass filters and fourth-order regularisation terms for stabilisation.

Backward parabolic differential equations also enjoy high popularity in the image analysis community where they have e.g. been used for image restoration and image deblurring respectively. The first contribution to backward diffusion dates back to 1955 when Kovásznyai and Joseph [KJ55] proposed to use the scaled negative Laplacian for contour enhancement. Gabor [Gab65] observed that the isotropy of the Laplacian operator leads to amplification of accidental noise at



contour lines at the same time it enhances the contour lines. As a remedy, he proposed to restrict the contrast enhancement to the orthogonal contour direction and – in a second step – suggested additional smoothing in tangent direction. Lindenbaum et al. [LFB94] make use of averaged derivatives in order to improve the directional sensitive filter by Gabor. However, the authors point out that smoothing in only one direction favours the emergence of artefacts in nearly isotropic image regions. They recommend to use the Perona-Malik filter [PM90] instead. Forces of Perona-Malik type are also used by Pollak et al. [PWK00] who specify a family of evolution equations to sharpen edges and suppress noise in the context of image segmentation. In [tFd<sup>+</sup>94], ter Haar Romeny et al. stress the influence of higher order time derivatives on the Gaussian deblurred image. Referring to the heat equation, the authors express the time derivatives in the spatial domain and approximate them using Gaussian derivatives. Steiner et al. [SKB98] highlight how backward diffusion can be used for feature enhancement in planar curves.

In the field of image processing, a frequently used stabilisation technique constrains the extrema in order to enforce a maximum-minimum principle. This is e.g. implemented in the inverse diffusion filter of Osher and Rudin [OR91]. It imposes zero diffusivities at extrema and applies backward diffusion everywhere else. The so-called forward-and-backward (FAB) diffusion of Gilboa et al. [GSZ02] follows a slightly different approach. Closely related to the Perona-Malik filter [PM90] it uses negative diffusivities for a specific range of gradient magnitudes. On the other hand, it imposes forward diffusion for values of low and zero gradient magnitude. By doing so, the filter prevents the output values from exploding at extrema. However, it is worth mentioning that – so far – all adequate implementations of inverse diffusion processes with forward or zero diffusivities at extrema require sophisticated numerical schemes. They use e.g. minmod discretisations of the Laplacian [OR91], nonstandard finite difference approximations of the squared gradient magnitude [WGW09], and splittings into two-pixel interactions [WWG18].

Another, less popular stabilisation approach implies the application of a fidelity term and has been used to penalise deviations from the input image [SZ98, CSH78] or from the average grey value of the desired range [SC97]. Consequently, both the weights of the fidelity and the diffusion term control the range of the filtered image.

Further methods achieve stabilisation using a regularisation strategy built on FFT-based operators [Car14, Car16, Car17] and by the restriction to polynomials of fixed finite degree [HKZ87]. Mair et al. [MWR96] discuss the well-posedness of deblurring Gaussian blur in the discrete image domain based on analytic number theory.

In summary, the presented methods offer an insight into the challenge of handling backward diffusion in practice and underline the demand for careful stabilisation strategies and sophisticated numerical methods.

In this chapter we are going to present an alternative approach to deal with backward diffusion problems. It prefers smarter modelling over smarter numerics. To understand it better, it is useful to recapitulate some relations between diffusion

and energy minimisation.

**Diffusion and Energy Minimisation.** For the sake of convenience we assume a one-dimensional evolution that smoothes an initial signal  $f : [a, b] \rightarrow \mathbb{R}$ . In this context, the original signal  $f$  serves as initial state of the diffusion equation

$$\partial_t u = \partial_x (g(u_x^2) u_x) \quad (4.1)$$

where  $u = u(x, t)$  represents the filtered outcome with  $u(x, 0) = f(x)$ . Additionally, let  $u_x = \partial_x u$  and assume reflecting boundary conditions at  $x = a$  and  $x = b$ . Given a nonnegative diffusivity function  $g$ , growing diffusion times  $t$  lead to simpler representations of the input signal. From Perona and Malik's work [PM90] we know that the smoothing effect at signal edges can be reduced if  $g$  is a decreasing function of the contrast  $u_x^2$ . As long as the flux function  $\Phi(u_x) := g(u_x^2) u_x$  is strictly increasing in  $u_x$  the corresponding forward diffusion process  $\partial_t u = \Phi'(u_x) u_{xx}$  involves no edge sharpening. This diffusion can be regarded as the gradient descent evolution which minimises the energy

$$E[u] = \int_a^b \Psi(u_x^2) dx. \quad (4.2)$$

The potential function  $\tilde{\Psi}(u_x) = \Psi(u_x^2)$  is strictly convex in  $u_x$ , increasing in  $u_x^2$ , and fulfils  $\tilde{\Psi}'(u_x^2) = g(u_x^2)$ . Furthermore, the energy functional has a flat minimiser which is – due to the strict convexity of the energy functional – unique. The gradient descent / diffusion evolution is well-posed and converges towards this minimiser for  $t \rightarrow \infty$ . Due to this classical emergence of well-posed forward diffusion from strictly convex energies it seems natural to believe that backward diffusion processes are necessarily associated with non-convex energies. However, as we will see, this conjecture is wrong.

**Contributions of this Chapter.** In this chapter, we show that a specific class of backward diffusion processes are gradient descent evolutions of energies that have one unexpected property: They are convex! Our second innovation is the incorporation of a specific constraint: We impose reflecting boundary conditions in the diffusion *co-domain*. This means that in case of greyscale images with an allowed grey value range of  $(0, 255)$  the occurring values are mirrored at the boundary positions 0 and 255. While such range constraints have shown their usefulness in some other context (see e.g. [NS14a]), to our knowledge they have never been used for stabilising backward diffusions. For our novel backward diffusion models, we show also a surprising numerical fact: A simple explicit scheme turns out to be stable and convergent. Last but not least, we apply our models to the contrast enhancement of greyscale and colour images.

This chapter is based on our journal paper [BCWW20a] and extends our conference contribution [BCWW18] in several aspects. First, we enhance our model for convex backward diffusion to support not only a global and weighted setting but also a localised variant. We analyse this extended model in terms of stability and convergence towards a unique minimiser. Furthermore, we formulate a

simple explicit scheme for our newly proposed approach which shares all important properties with the time-continuous evolution. In this context, we provide a detailed discussion on the selection of suitable time step sizes. Additionally, we suggest two new applications: global contrast enhancement of digital colour images and local contrast enhancement of digital grey and colour images.

**Structure of the Chapter.** In Section 4.2, we present our model for convex backward diffusion with range constraints. We describe a general approach which allows to formulate weighted local and global evolutions. Section 4.3 includes proofs for model properties such as range and rank-order preservation as well as convergence analysis and the derivation of explicit steady-state solutions. Section 4.4 provides a simple explicit scheme which can be used to solve the occurring initial value problem. In Section 4.5, we explain how to enhance the global and local contrast of digital images using the proposed model. Furthermore, we discuss the relation to existing literature on contrast enhancement. In Section 4.6, we draw conclusions from our findings and give an outlook on future research. Section 4.A contains supplementary material, including derivations and further experiments.

## 4.2 Model

Let us now explore the roots of our model and derive – in a second step – the particle evolution which forms the heart of our method and which is given by the gradient descent of a convex energy.

### 4.2.1 Motivation from Swarm Dynamics

The idea behind our model goes back to the scenario of describing a one-dimensional evolution of particles within a closed system. Recent literature on mathematical swarm models employs a pairwise potential  $U : \mathbb{R}^n \rightarrow \mathbb{R}$  to characterise the behaviour of individual particles (see e.g. [DCBC06, CHDB07, GP04, GF07, CFTV10] and the references therein). The potential function allows to steer attractive and repulsive forces among swarm mates. Physically simplified models like [GP03] neglect inertia and describe the individual particle velocity  $\partial_t \mathbf{v}_i$  within a swarm of size  $N$  directly as

$$\partial_t \mathbf{v}_i = - \sum_{\substack{j=1 \\ j \neq i}}^N \nabla U(\mathbf{v}_i - \mathbf{v}_j), \quad i = 1, \dots, N, \quad (4.3)$$

where  $\mathbf{v}_i$  and  $\mathbf{v}_j$  denote particle positions in  $\mathbb{R}^n$ . These models are also referred to as first order models and we provide a more detailed introduction in Chapter 3. Often they are inspired by biology and describe long-range attractive and short-range repulsive behaviour between swarm members. The interplay of attractive and repulsive forces leads to flocking and allows to gain stability for the whole swarm. Inverting this behaviour – resulting in short-range attractive and long-range repulsive forces – leads to a highly unstable scenario in which the swarm

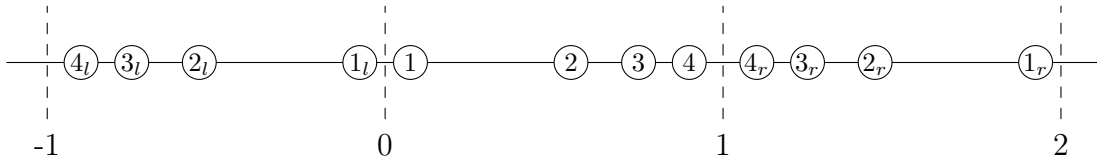


Figure 4.1: Four particles with positions in  $(0, 1)$  and their reflections at the left and right domain boundary (labelled  $l$  and  $r$  accordingly). Particle 2, for example, gets repelled by the particles  $1, 3, 4, 1_l, 2_r, 3_r, 4_r$ .

splits up into small separating groups which might never reach a point where they stand still. One would expect that a restriction to repulsive forces only will increase this instability even further. However, we will present a model which copes well with exactly this situation. In our set-up every particle moves within the open interval  $(0, 1)$  and has an interaction radius of size 1. Keeping this in mind, let us briefly examine the two main assumptions of the evolution. First, there exist reflections for all particles at the left and right domain boundary. Secondly, the particles repel each other and – furthermore – get repelled by the reflections. However, due to the limited viewing range, only one of the two reflections of a certain particle is considered at any given time, namely the one which is closer to the reference particle (see Figure 4.1). A special case occurs if the reference particle is located at position 0.5: the repulsive forces of both of its own reflections equal out.

## 4.2.2 Discrete Variational Model

We propose a dynamical system which has its roots in a spatial discretisation of the energy functional (4.2). Furthermore, we make use of a decreasing energy function  $\Psi : \mathbb{R}_0^+ \rightarrow \mathbb{R}$  and a global range constraint on  $u$ . The corresponding flux function  $\Phi$  is defined as  $\Phi(s) := \Psi'(s^2)s$ .

Our goal is to describe the evolution of one-dimensional – not necessarily distinct – particle positions  $v_i \in (0, 1)$ , where  $i = 1, \dots, N$ . Therefore, we extend the position vector  $\mathbf{v} = (v_1, \dots, v_N)^\top$  with the additional coordinates  $v_{N+1}, \dots, v_{2N}$  defined as  $v_{2N+1-i} := 2 - v_i \in (1, 2)$ . This extended position vector  $\mathbf{v} \in (0, 2)^{2N}$  allows to evaluate the energy function

$$E(\mathbf{v}, \mathbf{W}) = \frac{1}{4} \cdot \sum_{i=1}^{2N} \sum_{j=1}^{2N} w_{i,j} \cdot \Psi((v_j - v_i)^2), \quad (4.4)$$

which models the repulsion potential between all positions  $v_i$  and  $v_j$ . The coefficient  $w_{i,j}$  denotes entry  $j$  in row  $i$  of a constant non-negative weight matrix  $\mathbf{W} = (w_{i,j}) \in (\mathbb{R}_0^+)^{2N \times 2N}$ . It models the importance of the interaction between position  $v_i$  and  $v_j$ . All diagonal elements of the weight matrix are positive, i.e.  $w_{i,i} > 0, \forall i \in \{1, 2, \dots, 2N\}$ . In addition, we assume that the weights for all extended positions are the same as those for the original ones. Namely, we have

$$w_{i,j} = w_{i,2N+1-j} = w_{2N+1-i,j} = w_{2N+1-i,2N+1-j} \quad (4.5)$$

$\Psi_{a,n}(s^2)$	$\Psi'_{a,n}(s^2)$	$\Phi_{a,n}(s)$
$a \cdot ((s-1)^{2n} - 1)$	$\frac{a \cdot n}{s} \cdot (s-1)^{2n-1}$	$a \cdot n \cdot (s-1)^{2n-1}$

Table 4.1: One exemplary class of penaliser functions  $\Psi(s^2)$  for  $s \in [0, 1]$  with  $n \in \mathbb{N}$ ,  $a > 0$  and corresponding diffusivity  $\Psi'(s^2)$  and flux  $\Phi(s)$  functions.

for  $1 \leq i, j \leq N$ .

For the penaliser function  $\Psi$  we impose several restrictions which we discuss subsequently. Table 4.1 shows one reasonable class of functions  $\Psi_{a,n}$  as well as the corresponding diffusivities  $\Psi'_{a,n}$  and flux functions  $\Phi_{a,n}$ . In Figure 4.2 we provide an illustration of three functions using  $a = 1$  and  $n = 1, 2, 3$ . The penaliser is constructed following a three step procedure. First, the function  $\Psi(s^2)$  is defined as a continuously differentiable, decreasing, and strictly convex function for  $s \in [0, 1]$  with  $\Psi(0) = 0$  and  $\Phi_-(1) = 0$  (left-sided derivative). Next,  $\Psi$  is extended to  $[-1, 1]$  by symmetry and to  $\mathbb{R}$  by periodicity  $\Psi((2+s)^2) = \Psi(s^2)$ . This results in a penaliser  $\Psi(s^2)$  which is continuously differentiable everywhere except at even integers, where it is still continuous. Note that  $\Psi(s^2)$  is increasing on  $[-1, 0]$  and  $[1, 2]$ . The flux  $\Phi$  is continuous and increasing in  $(0, 2)$  with jump discontinuities at 0 and 2 (see Figure 4.2). Furthermore, we have that  $\Phi(s) = -\Phi(-s)$  and  $\Phi(2+s) = \Phi(s)$ . Exploiting the properties of  $\Psi$  allows us to rewrite (4.4) without the redundant entries  $v_{N+1}, \dots, v_{2N}$  (for details see Section 4.A.1) as

$$E(\mathbf{v}, \mathbf{W}) = \frac{1}{2} \cdot \sum_{i=1}^N \sum_{j=1}^N w_{i,j} \cdot \left( \Psi((v_j - v_i)^2) + \Psi((v_j + v_i)^2) \right). \quad (4.6)$$

A gradient descent for (4.4) is given by

$$\partial_t v_i = -\partial_{v_i} E(\mathbf{v}, \mathbf{W}) = \sum_{j \in J_1^i} w_{i,j} \cdot \Phi(v_j - v_i), \quad i = 1, \dots, 2N, \quad (4.7)$$

where  $v_i$  now are functions of the time  $t$  and

$$J_1^i := \{j \in \{1, 2, \dots, 2N\} \mid v_j \neq v_i\}. \quad (4.8)$$

Note that for  $1 \leq i, j \leq N$ , thus  $|v_j - v_i| < 1$ , the flux  $\Phi(v_j - v_i)$  is negative for  $v_j > v_i$  and positive otherwise, thus driving  $v_i$  always away from  $v_j$ . This implies that we have negative diffusivities  $\Psi'$  for all  $|v_j - v_i| < 1$ . Due to the convexity of  $\Psi(s^2)$ , the absolute values of the repulsive forces  $\Phi$  are decreasing with the distance between  $v_i$  and  $v_j$ . We remark that the jumps of  $\Phi$  at 0 and 2 are not problematic here, as all positions  $v_i$  and  $v_j$  in the argument of  $\Phi$  are distinct by the definition of  $J_1^i$ .

Let us discuss shortly how the interval constraint for the  $v_i$ ,  $i = 1, \dots, N$ , is enforced in (4.4) and (4.7). First, notice that  $v_{2N+1-i}$  for  $i = 1, \dots, N$  is the reflection of  $v_i$  on the right interval boundary 1. For  $v_i$  and  $v_{2N+1-j}$  with  $1 \leq i, j \leq N$  and  $v_{2N+1-j} - v_i < 1$  there is a repulsive force due to  $\Phi(v_{2N+1-j} - v_i) < 0$  that drives  $v_i$  and  $v_{2N+1-j}$  away from the *right* interval boundary. The closer  $v_i$  and

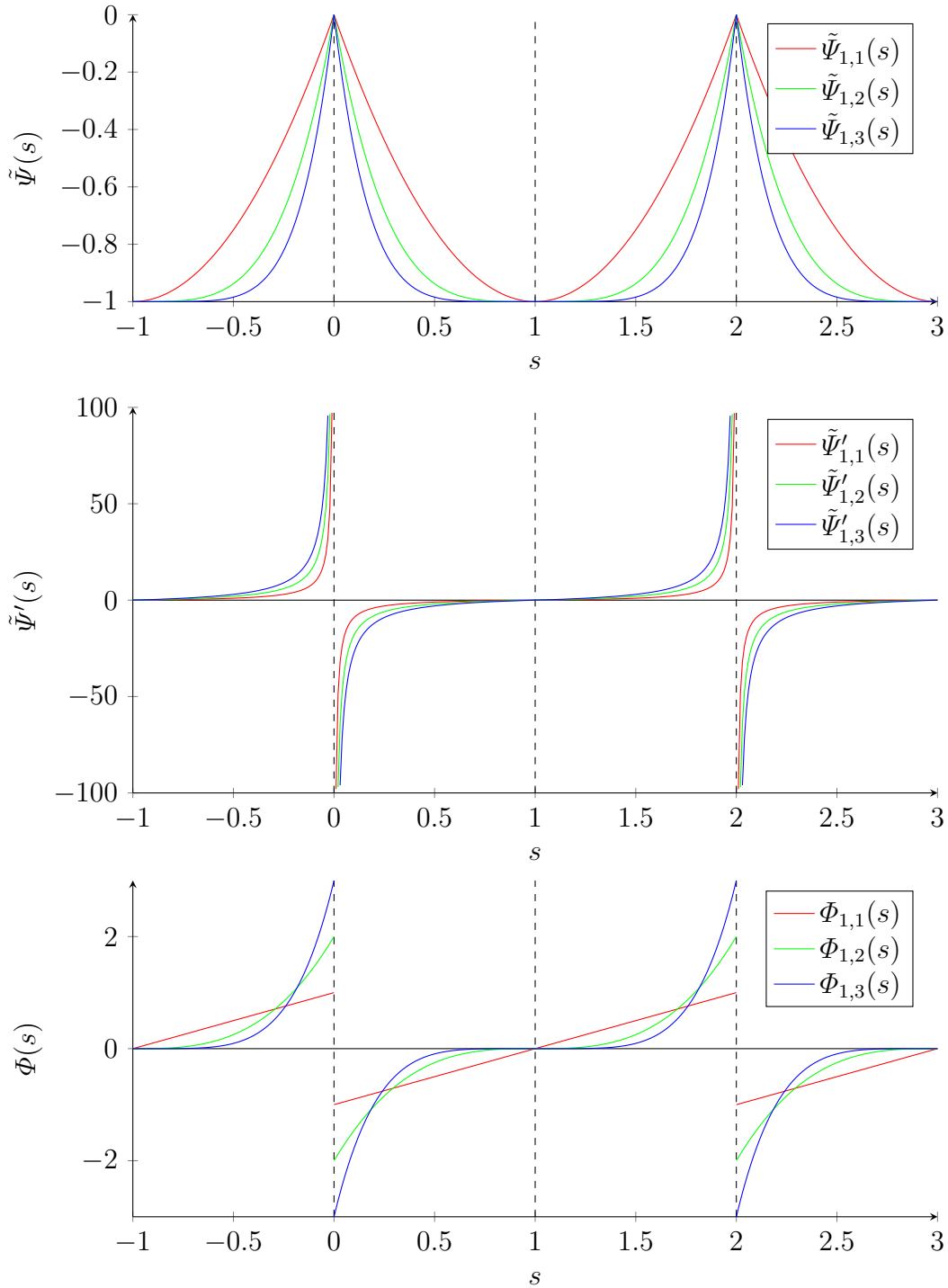


Figure 4.2: **Top:** Exemplary penaliser functions  $\tilde{\Psi}_{1,1}$ ,  $\tilde{\Psi}_{1,2}$ , and  $\tilde{\Psi}_{1,3}$  extended to the interval  $[-1, 3]$  by imposing *symmetry* and *periodicity* with  $\tilde{\Psi}_{a,n}(s) := \Psi_{a,n}(s^2)$ . **Middle:** Corresponding diffusivities  $\tilde{\Psi}'_{1,1}$ ,  $\tilde{\Psi}'_{1,2}$ , and  $\tilde{\Psi}'_{1,3}$  with  $\tilde{\Psi}'_{a,n}(s) := \Psi'_{a,n}(s^2)$ . **Bottom:** Corresponding flux functions  $\Phi_{1,1}$ ,  $\Phi_{1,2}$ , and  $\Phi_{1,3}$  with  $\Phi_{a,n}(s) = \Psi'_{a,n}(s^2)s$ .

$v_{2N+1-j}$  come to this boundary, the stronger is the repulsion. For  $v_{2N+1-j} - v_i > 1$ , we have  $\Phi(v_{2N+1-j} - v_i) > 0$ . By  $\Phi(v_{2N+1-j} - v_i) = \Phi((2 - v_j) - v_i) = \Phi((-v_j) - v_i)$ , this can equally be interpreted as a repulsion between  $v_i$  and  $-v_j$  where  $-v_j$  is the reflection of  $v_j$  at the left interval boundary 0. In this case the interaction between  $v_i$  and  $v_{2N+1-j}$  drives  $v_i$  and  $-v_j$  away from the *left* interval boundary. Recapitulating both possible cases, it becomes clear that every  $v_i$  is either repelled from the reflection of  $v_j$  at the left or at the right interval boundary, but never from both at the same time. As  $\partial_t v_{2N+1-i} = -\partial_t v_i$  holds in (4.7), the symmetry of  $\mathbf{v}$  is preserved. Dropping the redundant entries  $v_{N+1}, \dots, v_{2N}$ , Equation (4.7) can be rewritten as

$$\partial_t v_i = \sum_{j \in J_2^i} w_{i,j} \cdot \Phi(v_j - v_i) - \sum_{j=1}^N w_{i,j} \cdot \Phi(v_j + v_i), \quad (4.9)$$

for  $i = 1, \dots, N$ , where the second sum emphasises the repulsions between original and reflected coordinates in a symmetric way. The set  $J_2^i$  is defined as

$$J_2^i := \{j \in \{1, 2, \dots, N\} \mid v_j \neq v_i\}. \quad (4.10)$$

Equation (4.9) denotes a formulation for pure repulsion amongst  $N$  different positions  $v_i$  with stabilisation being achieved through the consideration of their reflections at the domain boundary. It is worth mentioning that within (4.6) and (4.9) we only make use of the first  $N \times N$  entries of  $\mathbf{W}$ . In the following, we denote this submatrix by  $\tilde{\mathbf{W}}$  and refer to its elements as  $\tilde{w}_{i,j}$ . Given an initial vector  $\mathbf{f} \in (0, 1)^N$  and initialising  $v_i(0) = f_i$ ,  $v_{2N+1-i}(0) = 2 - f_i$  for  $i = 1, \dots, N$ , the gradient descent (4.7) and (4.9) evolves  $\mathbf{v}$  towards a minimiser of  $E$ .

## 4.3 Theory

Below we provide a detailed analysis of the evolution and discuss its main properties. For this purpose we consider the Hessian of (4.6) whose entries for  $1 \leq i \leq N$  read

$$\partial_{v_i v_i} E(\mathbf{v}, \tilde{\mathbf{W}}) = \sum_{j \in J_2^i} \tilde{w}_{i,j} \cdot \Phi'(v_j - v_i) + \sum_{j=1}^N \tilde{w}_{i,j} \cdot \Phi'(v_j + v_i), \quad (4.11)$$

$$\partial_{v_i v_j} E(\mathbf{v}, \tilde{\mathbf{W}}) = \begin{cases} \tilde{w}_{i,j} \cdot (\Phi'(v_j + v_i) - \Phi'(v_j - v_i)), & \forall j \in J_2^i, \\ \tilde{w}_{i,j} \cdot \Phi'(v_j + v_i), & \forall j \in J_3^i, \end{cases} \quad (4.12)$$

where

$$J_3^i := \{j \in \{1, 2, \dots, N\} \mid v_i = v_j\}. \quad (4.13)$$

### 4.3.1 General Results

In a first step, let us investigate the well-posedness of the underlying initial value problem in the sense of Hadamard [Had02].

**Theorem 1** (Well-Posedness). *Let  $\Psi = \Psi_{a,n}$  as defined in Table 4.1. Then the initial value problem (4.9) is well-posed since*

- (a) *it has a solution,*
- (b) *the solution is unique, and*
- (c) *it depends continuously on the initial conditions.*

*Proof.* The initial value problem (4.9) can be written as

$$\dot{\mathbf{v}}(t) = \mathbf{f}(\mathbf{v}(t)) := -\nabla_{\mathbf{v}}E(\mathbf{v}(t), \mathbf{W}) \quad (4.14)$$

$$\mathbf{v}(0) = \mathbf{v}_0 \quad (4.15)$$

with  $\mathbf{v}(t) \in \mathbb{R}^{2N}$  and  $t \in \mathbb{R}_0^+$  where we make use of the fact that  $\mathbf{W}$  is a constant weight matrix.

In case  $\mathbf{f}(\mathbf{v}(t))$  is continuously differentiable and Lipschitz continuous all three conditions (a)–(c) hold. Existence and uniqueness directly follow from [Per01, chapter 3.1, Theorem 3]. Continuous dependence on the initial conditions is guaranteed due to [Per01, chapter 2.3, Theorem 1] which is based on Gronwall’s Lemma [Gro19]. Thus, let us now prove differentiability and Lipschitz continuity of  $\mathbf{f}(\mathbf{v}(t))$ .

**Differentiability:** Differentiability follows from the fact that all functions  $\Phi_{a,n}$  are continuously differentiable. As a consequence the partial derivatives of (8) w.r.t.  $v_i$  exist for  $i = 1, \dots, 2N$ .

**Lipschitz Continuity:** The Gershgorin circle theorem (cf. Chapter 2.1.5) allows to estimate a valid Lipschitz constant  $L$  as an upper bound of the spectral radius of the Jacobian of (4.9). For  $1 \leq i \leq N$  the entries read

$$\partial_{v_i}(\partial_t v_i) = - \sum_{j \in J_2^i} \tilde{w}_{i,j} \cdot (\Phi'(v_j - v_i) + \Phi'(v_j + v_i)) - 2 \cdot \sum_{j \in J_3^i} \tilde{w}_{i,j} \cdot \Phi'(v_j + v_i) \quad (4.16)$$

$$\partial_{v_j}(\partial_t v_i) = \begin{cases} \tilde{w}_{i,j} \cdot (\Phi'(v_j - v_i) - \Phi'(v_j + v_i)), & \forall j \in J_2^i, \\ -\tilde{w}_{i,j} \cdot \Phi'(v_j + v_i), & \forall j \in J_3^i. \end{cases} \quad (4.17)$$

The radii of the Gershgorin discs fulfil

$$\begin{aligned} r_i &= \sum_{\substack{j=1 \\ j \neq i}}^N |\partial_{v_j}(\partial_t v_i)| \\ &= \sum_{j \in J_2^i} \tilde{w}_{i,j} \cdot |\Phi'(v_j - v_i) - \Phi'(v_j + v_i)| + \sum_{\substack{j \in J_3^i \\ j \neq i}} \tilde{w}_{i,j} \cdot |\Phi'(v_j + v_i)| \\ &< \sum_{j \in J_2^i} \tilde{w}_{i,j} \cdot |\Phi'(v_j - v_i) - \Phi'(v_j + v_i)| + \sum_{j \in J_3^i} \tilde{w}_{i,j} \cdot |\Phi'(v_j + v_i)| \\ &=: \tilde{r}_i, \quad i = 1, \dots, N. \end{aligned} \quad (4.18)$$



Then we have  $|\lambda_i - \partial_{v_i}(\partial_t v_i)| < \tilde{r}_i$  for  $1 \leq i \leq N$  where  $\lambda_i$  denotes the  $i$ -th eigenvalue of the Jacobian of (4.9). This leads to the bounds

$$\begin{aligned}
 \lambda_i &< \sum_{j \in J_2^i} \tilde{w}_{i,j} \cdot (|\Phi'(v_j - v_i) - \Phi'(v_j + v_i)| - (\Phi'(v_j - v_i) + \Phi'(v_j + v_i))) \\
 &\quad + \sum_{j \in J_3^i} \tilde{w}_{i,j} \cdot (|\Phi'(v_j + v_i)| - 2 \cdot \Phi'(v_j + v_i)) \\
 &\leq \sum_{j \in J_2^i} \tilde{w}_{i,j} \cdot (|\Phi'(v_j - v_i)| + |\Phi'(v_j + v_i)| + |\Phi'(v_j - v_i)| + |\Phi'(v_j + v_i)|) \\
 &\quad + \sum_{j \in J_3^i} \tilde{w}_{i,j} \cdot (|\Phi'(v_j + v_i)| + 2 \cdot |\Phi'(v_j + v_i)|) \\
 &\leq 4 \cdot L_\Phi \cdot \sum_{j \in J_2^i} \tilde{w}_{i,j} + 3 \cdot L_\Phi \cdot \sum_{j \in J_3^i} \tilde{w}_{i,j} \\
 &< 4 \cdot L_\Phi \cdot \sum_{j=1}^N \tilde{w}_{i,j}, \quad i = 1, \dots, N,
 \end{aligned} \tag{4.19}$$

where  $L_\Phi$  represents the Lipschitz constant of the flux function  $\Phi$ . Using the same reasoning one can show that

$$\lambda_i > -4 \cdot L_\Phi \cdot \sum_{j=1}^N \tilde{w}_{i,j}, \quad i = 1, \dots, N. \tag{4.20}$$

Consequently, an upper bound for the spectral radius – and thus for the Lipschitz constant  $L$  of the gradient of (4.9) – reads

$$L \leq \max_{1 \leq i \leq N} |\lambda_i| < 4 \cdot L_\Phi \cdot \max_{1 \leq i \leq N} \sum_{j=1}^N \tilde{w}_{i,j} =: L_{\max}. \tag{4.21}$$

For our specific class of flux functions  $\Phi_{a,n}$  a valid Lipschitz constant  $L_\Phi$  is given by

$$L_\Phi = a \cdot n \cdot (2n - 1) \cdot 2^{2n-2} \tag{4.22}$$

such that we have

$$L < 4 \cdot a \cdot n \cdot (2n - 1) \cdot 2^{2n-2} \cdot \max_{1 \leq i \leq N} \sum_{j=1}^N \tilde{w}_{i,j}. \tag{4.23}$$

This concludes the proof.  $\square$

Next, let us show that no position  $v_i$  can ever reach or cross the interval boundaries 0 and 1.

**Theorem 2** (Avoidance of Range Interval Boundaries). *For any weighting matrix  $\tilde{\mathbf{W}} \in (\mathbb{R}_0^+)^{N \times N}$  all  $N$  positions  $v_i$  which evolve according to (4.9) and have an arbitrary initial value in  $(0, 1)$  do not reach the domain boundaries 0 and 1 for any time  $t \geq 0$ .*

*Proof.* Equation (4.9) can be written as

$$\partial_t v_i = \sum_{j \in J_2^i} \tilde{w}_{i,j} \cdot \left( \Phi(v_j - v_i) - \Phi(v_j + v_i) \right) - \sum_{j \in J_3^i} \tilde{w}_{i,j} \cdot \Phi(2v_i), \quad (4.24)$$

where  $1 \leq i \leq N$ . Notice that for  $j \in J_2^i$  we have

$$\lim_{v_i \rightarrow 0^+} \Phi(v_j - v_i) - \Phi(v_j + v_i) = 0, \quad (4.25)$$

$$\lim_{v_i \rightarrow 1^-} \Phi(v_j - v_i) - \Phi(v_j + v_i) = 0, \quad (4.26)$$

where the latter follows from the periodicity of  $\Phi$ . Consequently, any position  $v_i$  which gets arbitrarily close to one of the domain boundaries 0 or 1 experiences no impact by positions  $v_j$  with  $j \in J_2^i$ , and the first sum in (4.24) gets zero. The definition of  $\Psi(s^2)$  implies that

$$\Psi'(s^2) < 0, \quad \forall s \in (0, 1), \quad (4.27)$$

$$\Psi'(s^2) > 0, \quad \forall s \in (1, 2), \quad (4.28)$$

from which it follows for  $1 \leq i \leq N$  that

$$-\Phi(2v_i) > 0, \quad \forall v_i \in \left(0, \frac{1}{2}\right), \quad (4.29)$$

$$-\Phi(2v_i) < 0, \quad \forall v_i \in \left(\frac{1}{2}, 1\right). \quad (4.30)$$

Now remember that  $\tilde{\mathbf{W}} \in (\mathbb{R}_0^+)^{N \times N}$  and  $\tilde{w}_{i,i} > 0$ . In combination with (4.29) and (4.30) we get

$$\lim_{v_i \rightarrow 0^+} \partial_t v_i > 0 \quad \text{and} \quad \lim_{v_i \rightarrow 1^-} \partial_t v_i < 0, \quad (4.31)$$

which concludes the proof.  $\square$

Let us for a moment assume that the penaliser function is given by  $\Psi = \Psi_{a,n}$  from Table 4.1. Below, we prove that this implies convergence to the global minimum of the energy  $E(\mathbf{v}, \tilde{\mathbf{W}})$ .

**Theorem 3** (Convergence for  $\Psi = \Psi_{a,n=1}$ ). *For  $t \rightarrow \infty$ , given a penaliser  $\Psi_{a,1}$  with arbitrary  $a > 0$ , any initial configuration  $\mathbf{v} \in (0, 1)^N$  converges to a unique steady state  $\mathbf{v}^*$  which is the global minimiser of the energy given in (4.6).*

*Proof.* As a sum of convex functions, (4.6) is convex. Therefore, the function  $V(\mathbf{v}, \tilde{\mathbf{W}}) := E(\mathbf{v}, \tilde{\mathbf{W}}) - E(\mathbf{v}^*, \tilde{\mathbf{W}})$  (where  $\mathbf{v}^*$  is the equilibrium point) is a Lyapunov function with  $V(\mathbf{v}^*, \tilde{\mathbf{W}}) = 0$  and  $V(\mathbf{v}, \tilde{\mathbf{W}}) > 0$  for all  $\mathbf{v} \neq \mathbf{v}^*$ . Furthermore, we have

$$\partial_t V(\mathbf{v}, \tilde{\mathbf{W}}) = - \sum_{i=1}^N \left( \partial_{v_i} E(\mathbf{v}, \tilde{\mathbf{W}}) \right)^2 \leq 0. \quad (4.32)$$

According to Gershgorin's theorem [Ger31], one can show that the Hessian matrix of (4.6) is positive definite for  $\Psi = \Psi_{a,1}$  from which it follows that  $E(\mathbf{v}, \tilde{\mathbf{W}})$  has a strict (global) minimum. This implies that the inequality in (4.32) becomes strict except in case of  $\mathbf{v} = \mathbf{v}^*$ , and guarantees asymptotic Lyapunov stability [Lya92] of  $\mathbf{v}^*$ . Thus, we have convergence to  $\mathbf{v}^*$  for  $t \rightarrow \infty$ .  $\square$

*Remark 1.* Theorem 3 can be extended to the case of  $n = 2$  and – in a weaker formulation – to arbitrary  $n \in \mathbb{N}$ . The proofs for both cases are based on a straightforward application of the Gershgorin circle theorem. For details we refer to the supplementary material in Section 4.A.1.

(a) Given that  $\Psi = \Psi_{a,n=2}$ , let us assume that one of the following two conditions

- $v_i \neq \frac{1}{2}$ , or
- there exists  $j \in J_2^i$  for which  $v_j \neq 1 - v_i$  and  $\tilde{w}_{i,j} > 0$ ,

is fulfilled for every  $i \in [1, N]$  and  $t \geq 0$ . Then the Hessian matrix of (4.6) is positive definite and convergence to the strict global minimum of  $E(\mathbf{v}, \tilde{\mathbf{W}})$  follows.

(b) For all penaliser functions  $\Psi = \Psi_{a,n}$ , one can show that the Hessian matrix of (4.6) is positive semi-definite. This means that our method converges to a global minimum of  $E(\mathbf{v}, \tilde{\mathbf{W}})$ . However, this minimum does not have to be unique.

In general, the steady-state solution of (4.9) depends on the definition of the penaliser function  $\Psi$ . Based on (4.24), and assuming that  $\Psi = \Psi_{a,n}$ , a minimiser of  $E(\mathbf{v}, \tilde{\mathbf{W}})$  necessarily fulfils the equation

$$0 = \sum_{j \in J_2^i} \tilde{w}_{i,j} \cdot ((v_j^* - v_i^* - 1)^{2n-1} - (v_j^* + v_i^* - 1)^{2n-1}) - \sum_{j \in J_3^i} \tilde{w}_{i,j} \cdot (2v_i^* - 1)^{2n-1}, \quad (4.33)$$

where  $i = 1, \dots, N$ .

### 4.3.2 Global Model

If all positions  $v_i$  interact with each other during the evolution, i.e.  $\tilde{w}_{i,j} > 0$  for  $1 \leq i, j \leq N$ , we speak of our model as acting *globally*. Below, we prove the existence of weight matrices  $\tilde{\mathbf{W}}$  for which distinct positions  $v_i$  and  $v_j$  (with  $i \neq j$ ) can never become equal (assuming that the positions  $v_i$ ,  $i = 1, \dots, N$ , are distinct for  $t = 0$ ). This implies that the initial rank-order of  $v_i$  is preserved throughout the evolution.

**Theorem 4** (Distinctness of  $v_i$  and  $v_j$ ). *Among  $N$  initially distinct positions  $v_i \in (0, 1)$  evolving according to (4.9), no two ever become equal if  $\tilde{w}_{j,k} = \tilde{w}_{i,k} > 0$  for  $1 \leq i, j, k \leq N$ ,  $i \neq j$ .*

*Proof.* Given  $N$  distinct positions  $v_i \in (0, 1)$ , equation (4.9) can be written as

$$\partial_t v_i = \sum_{\substack{k=1 \\ k \neq i}}^N \tilde{w}_{i,k} \cdot \Phi(v_k - v_i) - \sum_{k=1}^N \tilde{w}_{i,k} \cdot \Phi(v_k + v_i), \quad (4.34)$$

for  $i = 1, \dots, N$ . We use this equation to derive the difference

$$\begin{aligned} \partial_t (v_j - v_i) &= (\tilde{w}_{j,i} + \tilde{w}_{i,j}) \cdot \Phi(v_i - v_j) \\ &\quad + \sum_{\substack{k=1 \\ k \neq i,j}}^N \left( \tilde{w}_{j,k} \cdot \Phi(v_k - v_j) - \tilde{w}_{i,k} \cdot \Phi(v_k - v_i) \right) \\ &\quad - \sum_{k=1}^N \left( \tilde{w}_{j,k} \cdot \Phi(v_k + v_j) - \tilde{w}_{i,k} \cdot \Phi(v_k + v_i) \right), \end{aligned} \quad (4.35)$$

where  $1 \leq i, j \leq N$ . Assume w.l.o.g. that  $v_j > v_i$  and consider (4.35) in the limit  $v_j - v_i \rightarrow 0$ . Then we have

$$\lim_{v_j - v_i \rightarrow 0} (\tilde{w}_{j,i} + \tilde{w}_{i,j}) \cdot \Phi(v_i - v_j) > 0, \quad (4.36)$$

if  $\tilde{w}_{j,i} + \tilde{w}_{i,j} > 0$ , which every global model fulfils by the assumption that  $\tilde{w}_{i,j} > 0$  for  $1 \leq i, j \leq N$ . This follows from the fact that  $\Phi(s) > 0$  for  $s \in (-1, 0)$ . Furthermore, we have

$$\begin{aligned} &\lim_{v_j - v_i \rightarrow 0} \sum_{\substack{k=1 \\ k \neq i,j}}^N \left( \tilde{w}_{j,k} \cdot \Phi(v_k - v_j) - \tilde{w}_{i,k} \cdot \Phi(v_k - v_i) \right) \\ &\quad - \sum_{k=1}^N \left( \tilde{w}_{j,k} \cdot \Phi(v_k + v_j) - \tilde{w}_{i,k} \cdot \Phi(v_k + v_i) \right) \\ &= \sum_{\substack{k=1 \\ k \neq i,j}}^N (\tilde{w}_{j,k} - \tilde{w}_{i,k}) \cdot \Phi(v_k - v_i) - \sum_{k=1}^N (\tilde{w}_{j,k} - \tilde{w}_{i,k}) \cdot \Phi(v_k + v_i) \\ &= 0 \quad \text{if } \tilde{w}_{j,k} = \tilde{w}_{i,k} \text{ for } 1 \leq k \leq N. \end{aligned} \quad (4.37)$$

In conclusion, we can guarantee for global models with distinct particle positions that

$$\lim_{v_j - v_i \rightarrow 0} \partial_t (v_j - v_i) > 0, \quad (4.38)$$

if  $\tilde{w}_{j,k} = \tilde{w}_{i,k}$  where  $1 \leq i, j, k \leq N$  and  $i \neq j$ . According to (4.38),  $v_j$  will always start moving away from  $v_i$  (and vice versa) when the difference between both gets sufficiently small. Since the initial positions are distinct, it follows that  $v_i \neq v_j$  for  $i \neq j$  for all times  $t$ .  $\square$

A special case occurs if all entries of the weight matrix  $\tilde{\mathbf{W}}$  are set to 1 – i.e.  $\tilde{\mathbf{W}} = \mathbf{1}\mathbf{1}^T$  with  $\mathbf{1} := (1, \dots, 1)^T \in \mathbb{R}^N$ . For this scenario, we obtain an analytic steady-state solution which is independent of the penaliser  $\Psi$ :

**Theorem 5** (Analytic Steady-State Solution for  $\tilde{\mathbf{W}} = \mathbf{1}\mathbf{1}^\top$ ). *Under the assumption that  $(v_i)$  is in increasing order,  $\tilde{\mathbf{W}} = \mathbf{1}\mathbf{1}^\top$ , and that  $\Psi(s^2)$  is twice continuously differentiable in  $(0, 2)$  the unique minimiser of (4.4) is given by  $\mathbf{v}^* = (v_1^*, \dots, v_{2N}^*)^\top$ ,  $v_i^* = (i - 1/2)/N$ ,  $i = 1, \dots, 2N$ .*

*Proof.* With  $\tilde{\mathbf{W}} = \mathbf{1}\mathbf{1}^\top$ , Equation (4.4) can be rewritten without the redundant entries of  $\mathbf{v}$  (for details see Section 4.A.1) as

$$E(\mathbf{v}) = \sum_{i=1}^{N-1} \sum_{j=i+1}^N \Psi((v_j - v_i)^2) + \frac{1}{2} \cdot \sum_{i=1}^N \Psi(4v_i^2) + \sum_{i=1}^{N-1} \sum_{j=i+1}^N \Psi((v_j + v_i)^2). \quad (4.39)$$

From this, one can verify by straightforward, albeit lengthy calculations that  $\nabla E(\mathbf{v}^*) = 0$ . Moreover, one finds that the Hessian of  $E$  at  $\mathbf{v}^*$  is

$$\mathrm{D}^2 E(\mathbf{v}^*) = \sum_{k=1}^N \mathbf{A}_k \Phi' \left( \frac{k}{N} \right) \quad (4.40)$$

Here,  $\mathbf{A}_k$  are sparse symmetric  $N \times N$ -matrices given by

$$\mathbf{A}_k = 2\mathbf{I} - \mathbf{T}_k - \mathbf{T}_{-k} + \mathbf{H}_{k+1} + \mathbf{H}_{2N-k+1}, \quad (4.41)$$

$$\mathbf{A}_N = \mathbf{I} + \mathbf{H}_{N+1}, \quad (4.42)$$

for  $k = 1, \dots, N - 1$ , where the unit matrix  $\mathbf{I}$ , the single-diagonal Toeplitz matrices  $\mathbf{T}_k$ , and the single-antidiagonal Hankel matrices  $\mathbf{H}_k$  are defined as

$$\mathbf{I} = (\delta_{i,j})_{i,j=1}^N, \quad (4.43)$$

$$\mathbf{T}_k = (\delta_{j-i,k})_{i,j=1}^N, \quad (4.44)$$

$$\mathbf{H}_k = (\delta_{i+j,k})_{i,j=1}^N. \quad (4.45)$$

Here,  $\delta_{i,j}$  denotes the Kronecker symbol,  $\delta_{i,j} = 1$  if  $i = j$ , and  $\delta_{i,j} = 0$  otherwise. All  $\mathbf{A}_k$ ,  $k = 1, \dots, N$  are weakly diagonally dominant with positive diagonal, thus positive semidefinite by Gershgorin's Theorem. Moreover, the tridiagonal matrix  $\mathbf{A}_1$  is of full rank, thus even positive definite. By strict convexity of  $\Psi(s^2)$ , all  $\Phi'(k/N)$  are positive, thus  $\mathrm{D}^2 E(\mathbf{v}^*)$  is positive definite.

As a consequence, the steady state of the gradient descent (4.9) for any initial data  $\mathbf{f}$  (with arbitrary rank-order) can – under the condition that  $\tilde{\mathbf{W}} = \mathbf{1}\mathbf{1}^\top$  – be computed directly by sorting the  $f_i$ : Let  $\sigma$  be the permutation of  $\{1, \dots, N\}$  for which  $(f_{\sigma^{-1}(i)})_{i=1, \dots, N}$  is increasing (this is what a sorting algorithm computes), the steady state is given by  $v_i^* = (\sigma(i) - 1/2)/N$  for  $i = 1, \dots, N$  (cf. Figure 4.3).  $\square$

Additionally, we present an analytic expression for the steady state of the global model given a penaliser function  $\Psi = \Psi_{a,n}$  (cf. Table 4.1) with  $n = 1$ .

**Theorem 6** (Analytic Steady-State Solution for  $\Psi = \Psi_{a,n=1}$ ). *Given  $N$  distinct positions  $v_i$  in increasing order and a penaliser function  $\Psi = \Psi_{a,n=1}$ , the unique minimiser of (4.4) is given by*

$$v_i^* = \frac{\sum_{j=1}^i \tilde{w}_{i,j} - \frac{1}{2} \tilde{w}_{i,i}}{\sum_{j=1}^N \tilde{w}_{i,j}}, \quad i = 1, \dots, N. \quad (4.46)$$

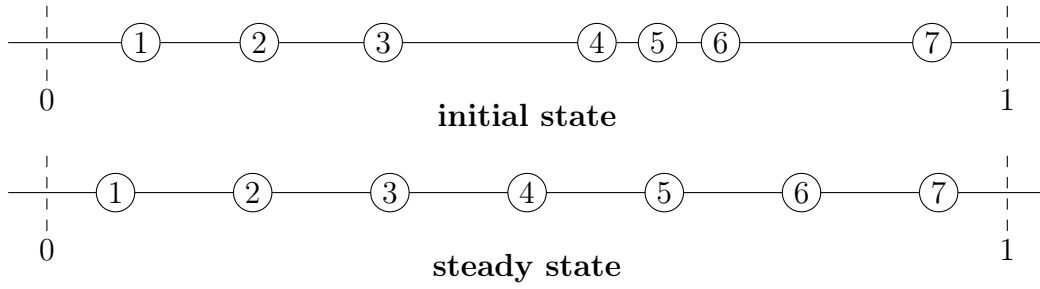


Figure 4.3: Application of the global model to a system of 7 particles with weight matrix  $\tilde{\mathbf{W}} = \mathbf{1}\mathbf{1}^T$ .

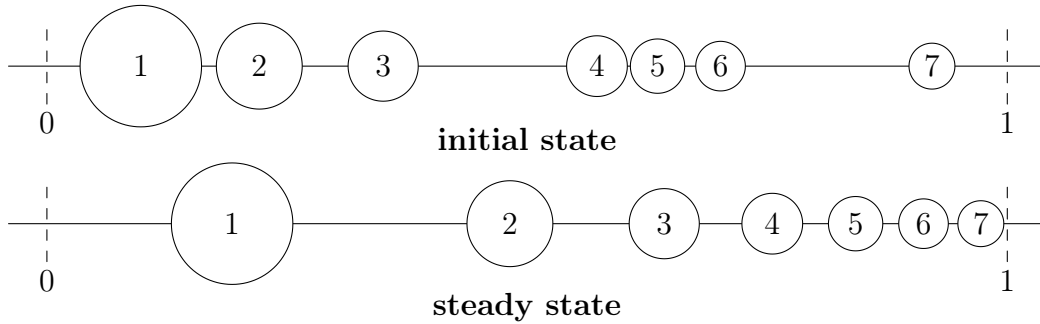


Figure 4.4: Application of the global model to a system of 7 particles with  $\tilde{w}_{i,k} = 1/k$  for  $1 \leq i, k \leq N$ .

*Proof.* The presented minimiser follows directly from (4.33). Figure 4.4 provides an illustration of the steady state.  $\square$

Finally, and in case all entries of the weight matrix  $\tilde{\mathbf{W}}$  are set to 1, we show that the global model converges – independently of  $\Psi$  – to a unique steady state:

**Theorem 7** (Convergence for  $\tilde{\mathbf{W}} = \mathbf{1}\mathbf{1}^T$ ). *Given that  $\tilde{\mathbf{W}} = \mathbf{1}\mathbf{1}^T$ , any initial configuration  $\mathbf{v} \in (0, 1)^N$  with distinct entries converges to a unique steady state  $\mathbf{v}^*$  for  $t \rightarrow \infty$ . This is the global minimiser of the energy given in (4.6).*

*Proof.* Using the same reasoning as in the proof for Theorem 3 we know that inequality (4.32) holds. Due to the positive definiteness of (4.40) it follows that  $E(\mathbf{v}, \tilde{\mathbf{W}})$  has a strict (global) minimum which implies that the inequality in (4.32) becomes strict except in case of  $\mathbf{v} = \mathbf{v}^*$ . This guarantees asymptotic Lyapunov stability of  $\mathbf{v}^*$  and thus convergence to  $\mathbf{v}^*$  for  $t \rightarrow \infty$ .  $\square$

### 4.3.3 Relation to Variational Signal and Image Filtering

Let us now interpret  $v_1, \dots, v_N$  as samples of a smooth 1D signal  $u : \Omega \rightarrow [0, 1]$  over an interval  $\Omega$  of the real axis, taken at sampling positions  $x_i = x_0 + i h$  with grid mesh size  $h > 0$ . We consider the model (4.4) with  $w_{i,j} := \gamma(|x_j - x_i|)$ , where  $\gamma : \mathbb{R}_0^+ \rightarrow [0, 1]$  is a non-increasing weighting function with compact support  $[0, \varrho]$ .

**Theorem 8** (Space-Continuous Energy). *Equation (4.4) can be considered as a discretisation of*

$$E[u] = \frac{1}{2} \int_{\Omega} (W(u_x^2) + B(u)) \, dx \quad (4.47)$$

with penaliser  $W(u_x^2) \approx C\Psi(u_x^2)$  and barrier function  $B(u) \approx D\Psi(4u^2)$ , where  $C$  and  $D$  are positive constants.

*Remark 2.*

- (a) The penaliser  $W$  is decreasing and convex in  $u_x$ . The barrier function  $B$  is convex and it enforces the interval constraint on  $u$  by favouring values  $u$  away from the interval boundaries. The discrete penaliser  $\Psi$  generates both the penaliser  $W$  for derivatives and the barrier function  $B$ .
- (b) Note that by construction of  $W$  the diffusivity  $g(u_x^2) := W'(u_x^2) \sim \Psi'(u_x^2)$  has a singularity at 0 with  $-\infty$  as limit.
- (c) The cut-off of  $\gamma$  at radius  $\varrho$  implies the locality of the functional (4.47) that can thereby be linked to a diffusion equation of type (4.1). Without a cut-off, a nonlocal diffusion equation would arise instead.

*Proof of Theorem 8.* We notice first that  $v_j - v_i$  and  $v_i + v_j$  for  $1 \leq i, j \leq N$  are first-order approximations of  $(j - i) h u_x(x_i)$  and  $2u(x_i)$ , respectively.

**Derivation of the Penaliser  $W$ .** Assume first for simplicity that  $\Psi(s^2) = -\kappa s$ ,  $\kappa > 0$  is linear in  $s$  on  $[0, 1]$  (thus not strictly convex). Then we have for a part of the inner sums of (4.4) corresponding to a fixed  $i$ :

$$\begin{aligned} & \frac{1}{2} \left( \sum_{j=1}^N \gamma(|x_j - x_i|) \cdot \Psi((v_j - v_i)^2) + \sum_{j=N+1}^{2N} \gamma(|x_j - x_{2N+1-i}|) \cdot \Psi((v_j - v_{2N+1-i})^2) \right) \\ &= \sum_{j=1}^N \gamma(|x_j - x_i|) \cdot \Psi((|v_j - v_i|)^2) \\ &\approx -\kappa h u_x(x_i) \sum_{j=1}^N \gamma(|j - i| h) \cdot |j - i| \\ &= h \Psi(u_x(x_i)^2) \sum_{k=1-i}^{N-i} |k| \gamma(|k| h) \\ &\approx h \Psi(u_x(x_i)^2) \cdot \frac{2}{h^2} \int_0^\varrho z \gamma(z) dz \\ &=: hC\Psi(u_x(x_i)^2), \end{aligned} \quad (4.48)$$

where in the last step the sum over  $k = 1 - i, \dots, N - i$  has been replaced with a sum over  $k = -\lfloor \varrho/h \rfloor, \dots, \lfloor \varrho/h \rfloor$ , thus introducing a cutoff error for those locations  $x_i$  that are within the distance  $\varrho$  from the interval ends. Summation

over  $i = 1, \dots, N$  approximates  $\int_{\Omega} C\Psi(u_x^2)dx$  from which we can read off  $W(u_x^2) \approx C\Psi(u_x^2)$ .

For  $\Psi(s^2)$  that are nonlinear in  $s$ ,  $\Psi(u_x(x_i)^2)$  in (4.48) is changed into a weighted sum of  $\Psi((ku_x(x_i))^2)$  for  $k = 1, \dots, N - 1$ , which still amounts to a decreasing function  $W(u_x^2)$  that is convex in  $u_x$ . Qualitatively,  $W'$  then behaves the same way as before.

**Derivation of the Barrier Function  $B$ .** Collecting the summands of (4.4) that were not used in (4.48), we have, again for fixed  $i$ ,

$$\begin{aligned} & \frac{1}{2} \left( \sum_{j=N+1}^{2N} \gamma(|x_j - x_i|) \cdot \Psi((v_j - v_i)^2) + \sum_{j=1}^N \gamma(|x_j - x_{2N+1-i}|) \cdot \Psi((v_j - v_{2N+1-i})^2) \right) \\ &= \sum_{j=1}^N \gamma(|x_j - x_i|) \cdot \Psi((v_i + v_j)^2) \\ &\approx \left( \frac{2}{h} \int_0^e \gamma(z)dz + 1 \right) \cdot \Psi(4u(x_i)^2) \\ &=: hD \cdot \Psi(4u(x_i)^2), \end{aligned} \tag{4.49}$$

and thus after summation over  $i$  analogous to the previous step  $\int_{\Omega} B(u) dx$  with  $B(u) \approx D\Psi(4u^2)$ .  $\square$

Similar derivations can be made for patches of 2D images. A point worth noticing is that the barrier function  $B$  is bounded. This differs from usual continuous models where such barrier functions tend to infinity at the interval boundaries. However, for each given sampling grid and patch size the barrier function is just strong enough to prevent  $W$  from pushing the values out of the interval.

## 4.4 Explicit Time Discretisation

Up to this point we have established a theory for the time-continuous evolution of particle positions. In order to be able to employ our model in simulations and applications we need to discretise (4.9) in time. Subsequently, we provide a simple yet powerful discretisation which preserves all important properties of the time-continuous model. An approximation of the time derivative in (4.9) by forward differences yields the explicit scheme

$$v_i^{k+1} = v_i^k + \alpha \cdot \sum_{\ell \in J_2^i} \tilde{w}_{i,\ell} \cdot \Phi(v_\ell^k - v_i^k) - \alpha \cdot \sum_{\ell=1}^N \tilde{w}_{i,\ell} \cdot \Phi(v_\ell^k + v_i^k), \tag{4.50}$$

for  $i = 1, \dots, N$ , where  $\alpha$  denotes the time step size and an upper index  $k$  refers to the time  $k\alpha$ . In the following, we derive necessary conditions for which the explicit scheme preserves the position range  $(0, 1)$  and the position ordering. Furthermore, we show convergence of (4.50) in dependence of  $\alpha$ .



**Theorem 9** (Avoidance of Range Interval Boundaries of the Explicit Scheme). *Let  $L_\Phi$  be the Lipschitz constant of  $\Phi$  restricted to the interval  $(0, 2)$ . Moreover, let  $0 < v_i^k < 1$ , for every  $1 \leq i \leq N$ , and assume that the time step size  $\alpha$  of the explicit scheme (4.50) satisfies*

$$0 < \alpha < \frac{1}{2 \cdot L_\Phi \cdot \max_{1 \leq i \leq N} \sum_{\ell=1}^N \tilde{w}_{i,\ell}}. \quad (4.51)$$

Then it follows that  $0 < v_i^{k+1} < 1$  for every  $1 \leq i \leq N$ .

*Proof.* In accordance with (4.24) the explicit scheme (4.50) can be written as

$$v_i^{k+1} = v_i^k + \alpha \cdot \sum_{\ell \in J_2^i} \tilde{w}_{i,\ell} \cdot \left( \Phi(v_\ell^k - v_i^k) - \Phi(v_\ell^k + v_i^k) \right) - \alpha \cdot \sum_{\ell \in J_3^i} \tilde{w}_{i,\ell} \cdot \Phi(2v_i^k), \quad (4.52)$$

where  $i = 1, \dots, N$ . Now assume that  $0 < v_i^k, v_j^k < 1$  and let us examine the contribution of the two summation terms in (4.52). We need to distinguish the following five cases:

1. If  $v_i^k = v_j^k \leq \frac{1}{2}$  then  $2v_i^k \in (0, 1]$ . Thus,

$$0 \leq -\Phi(2v_i^k). \quad (4.53)$$

2. If  $\frac{1}{2} < v_i^k = v_j^k$  then  $2v_i^k \in (1, 2)$ . Thus, using  $\Phi(1) = 0$ ,

$$|\Phi(2v_i^k)| = |\Phi(2v_i^k) - \Phi(1)| \leq |2v_i^k - 1| \cdot L_\Phi < 2v_i^k \cdot L_\Phi. \quad (4.54)$$

3. If  $v_i^k < v_j^k$  then  $v_j^k - v_i^k, v_j^k + v_i^k \in (0, 2)$ . Thus,

$$|\Phi(v_j^k + v_i^k) - \Phi(v_j^k - v_i^k)| \leq L_\Phi \cdot 2v_i^k. \quad (4.55)$$

4. If  $v_j^k < v_i^k \leq \frac{1}{2}$  then  $v_j^k - v_i^k \in (-1, 0)$  and  $v_j^k + v_i^k \in (0, 1)$ . Thus,

$$0 \leq \Phi(v_j^k - v_i^k) - \Phi(v_j^k + v_i^k), \quad (4.56)$$

$$0 \leq -\Phi(2v_i^k). \quad (4.57)$$

5. Finally, if  $v_j^k < v_i^k$  and  $\frac{1}{2} < v_i^k$ , using the periodicity of  $\Phi$  we get

$$|\Phi(v_j^k - v_i^k) - \Phi(v_j^k + v_i^k)| = |\Phi(v_j^k + v_i^k) - \Phi(2 + v_j^k - v_i^k)| \leq 2v_i^k \cdot L_\Phi. \quad (4.58)$$

Combining (4.50) with (4.51) and (4.53)–(4.58) we obtain that

$$\begin{aligned} v_i^{k+1} - v_i^k &= -\alpha \cdot \sum_{\ell \in J_2^i} \tilde{w}_{i,\ell} \cdot \left( \Phi(v_\ell^k + v_i^k) - \Phi(v_\ell^k - v_i^k) \right) - \alpha \cdot \sum_{\ell \in J_3^i} \tilde{w}_{i,\ell} \cdot \Phi(2v_i^k) \\ &\geq -\alpha \cdot L_\Phi \cdot 2v_i^k \cdot \sum_{\ell=1}^N \tilde{w}_{i,\ell} \\ &> -v_i^k, \end{aligned} \quad (4.59)$$

from which it directly follows that  $v_i^{k+1} > 0$ , as claimed.

The proof for  $v_i^{k+1} < 1$  is straightforward. Assume w.l.o.g. that  $\tilde{v}_i^k := 1 - v_i^k$ . For the reasons given above, we obtain  $\tilde{v}_i^{k+1} > 0$ . Consequently,  $1 - v_i^{k+1} > 0$  and  $v_i^{k+1} < 1$  follows.  $\square$

**Theorem 10** (Rank-Order Preservation of the Explicit Scheme). *Let  $L_\Phi$  be the Lipschitz constant of  $\Phi$  restricted to the interval  $(0, 2)$ . Furthermore, let  $v_i^0$ , for  $i = 1, \dots, N$ , denote the initially distinct positions in  $(0, 1)$  and – in accordance with Theorem 4 – let the weight matrix  $\tilde{\mathbf{W}}$  have constant columns, i.e.  $\tilde{w}_{j,\ell} = \tilde{w}_{i,\ell}$  for  $1 \leq i, j, \ell \leq N$ . Moreover, let  $0 < v_i^k < v_j^k < 1$  and assume that the time step size  $\alpha$  used in the explicit scheme (4.50) satisfies*

$$0 < \alpha < \frac{1}{2 \cdot L_\Phi \cdot \max_{1 \leq i \leq N} \sum_{\ell=1}^N \tilde{w}_{i,\ell}}. \quad (4.60)$$

Then we have  $v_i^{k+1} < v_j^{k+1}$ .

*Proof.* For distinct positions, (4.50) reads

$$v_i^{k+1} = v_i^k + \alpha \cdot \sum_{\substack{\ell=1 \\ \ell \neq i}}^N \tilde{w}_{i,\ell} \cdot \Phi(v_\ell^k - v_i^k) - \alpha \cdot \sum_{\ell=1}^N \tilde{w}_{i,\ell} \cdot \Phi(v_\ell^k + v_i^k) \quad (4.61)$$

for  $i = 1, \dots, N$ . Considering this explicit discretisation for  $\partial_t v_i$  and  $\partial_t v_j$  we obtain for  $i, j \in \{1, 2, \dots, N\}$ :

$$\begin{aligned} v_j^{k+1} - v_i^{k+1} &= v_j^k - v_i^k + \alpha \cdot (\tilde{w}_{j,i} + \tilde{w}_{i,j}) \cdot \Phi(v_i^k - v_j^k) \\ &\quad + \alpha \cdot \sum_{\substack{\ell=1 \\ \ell \neq i,j}}^N \left( \tilde{w}_{j,\ell} \cdot \Phi(v_\ell^k - v_j^k) - \tilde{w}_{i,\ell} \cdot \Phi(v_\ell^k - v_i^k) \right) \\ &\quad - \alpha \cdot \sum_{\ell=1}^N \left( \tilde{w}_{j,\ell} \cdot \Phi(v_\ell^k + v_j^k) - \tilde{w}_{i,\ell} \cdot \Phi(v_\ell^k + v_i^k) \right). \end{aligned} \quad (4.62)$$

Now remember that  $v_i^k < v_j^k$  by assumption and that – as a consequence –

$$\alpha \cdot (\tilde{w}_{j,i} + \tilde{w}_{i,j}) \cdot \Phi(v_i^k - v_j^k) > 0. \quad (4.63)$$

Using the fact that  $\tilde{w}_{j,k} = \tilde{w}_{i,k}$  for  $1 \leq i, j, k \leq N$  and that  $\Phi$  is Lipschitz in  $(0, 2)$ ,

we also know that

$$\begin{aligned}
 T_1 &:= \alpha \cdot \sum_{\substack{\ell=1 \\ \ell \neq i, j}}^N \left| \tilde{w}_{j, \ell} \cdot \Phi(v_\ell^k - v_j^k) - \tilde{w}_{i, \ell} \cdot \Phi(v_\ell^k - v_i^k) \right| \\
 &= \alpha \cdot \sum_{\substack{\ell=1 \\ \ell \neq i, j}}^N \tilde{w}_{j, \ell} \cdot \left| \Phi(v_\ell^k - v_j^k) - \Phi(v_\ell^k - v_i^k) \right| \\
 &\leq \alpha \cdot L_\Phi \cdot |v_i^k - v_j^k| \cdot \sum_{\substack{\ell=1 \\ \ell \neq i, j}}^N \tilde{w}_{j, \ell} ,
 \end{aligned} \tag{4.64}$$

$$\begin{aligned}
 T_2 &:= \alpha \cdot \sum_{\ell=1}^N \left| \tilde{w}_{j, \ell} \cdot \Phi(v_\ell^k + v_j^k) - \tilde{w}_{i, \ell} \cdot \Phi(v_\ell^k + v_i^k) \right| \\
 &= \alpha \cdot \sum_{\ell=1}^N \tilde{w}_{j, \ell} \cdot \left| \Phi(v_\ell^k + v_j^k) - \Phi(v_\ell^k + v_i^k) \right| \\
 &\leq \alpha \cdot L_\Phi \cdot |v_j^k - v_i^k| \cdot \sum_{\ell=1}^N \tilde{w}_{j, \ell} .
 \end{aligned} \tag{4.65}$$

Let the time step size  $\alpha$  fulfil (4.60). Then we can write

$$T_1 + T_2 < 2 \cdot L_\Phi \cdot 2 \cdot |v_j^k - v_i^k| \cdot \sum_{\ell=1}^N \tilde{w}_{j, \ell} < v_j^k - v_i^k . \tag{4.66}$$

In combination with  $T_1, T_2 \geq 0$ , it follows that

$$T_2 - T_1 \geq -T_2 - T_1 > -(v_j^k - v_i^k), \tag{4.67}$$

and we immediately know that  $v_j^k - v_i^k - T_1 + T_2 > 0$ . Together with (4.62) and (4.63) we get  $0 < v_j^{k+1} - v_i^{k+1}$ , as claimed.  $\square$

**Theorem 11** (Convergence of the Explicit Scheme). *Let (4.6) be a twice continuously differentiable convex function. Then the explicit scheme (4.50) converges for time step sizes*

$$0 < \alpha \leq \frac{1}{2 \cdot L_\Phi \cdot \max_{1 \leq i \leq N} \sum_{j=1}^N \tilde{w}_{i, j}} < \frac{2}{L}, \tag{4.68}$$

where  $L_\Phi$  denotes the Lipschitz constant of  $\Phi$  restricted to the interval  $(0, 2)$  and  $L$  refers to the Lipschitz constant of the gradient of (4.6).

*Proof.* Convergence of the gradient method to the global minimum of  $E(\mathbf{v}, \tilde{\mathbf{W}})$  is well-known for continuously differentiable convex functions with Lipschitz continuous gradient and time step sizes  $0 < \alpha < 2/L$  (see e.g. [Nes04, Theorem 2.1.14]). A valid Lipschitz constant is given by  $L_{\max}$  as defined in (4.21). Consequently, the time step sizes  $\alpha$  need to fulfil (4.68) in order to ensure convergence of (4.50). The smaller or equal relation results from (4.21). The latter defines  $L_{\max} > L$  such that  $\tau = 2/L_{\max}$  represents a valid time step size.  $\square$

*Remark 3* (Optimal Time Step Size). The optimal time step size, i.e. the value of  $\alpha$  which leads to most rapid descent, is given by  $\alpha = 1/L$  (see e.g. [Nes04, 2.1.5]). Thus, we suggest to use  $\alpha = 1/L_{\max}$ .

## 4.5 Application to Image Enhancement

Now that we have presented a stable and convergent numerical scheme, we apply (4.50) to enhance the contrast of digital greyscale and colour images. Throughout all experiments we use  $\Psi = \Psi_{1,1}$  (cf. Table 4.1 and Figure 4.2).

### 4.5.1 Greyscale Images

The application of the proposed model to greyscale images follows the ideas presented in [BW16]. We define a digital greyscale image as a mapping  $f : \{1, \dots, n\} \times \{1, \dots, m\} \rightarrow [0, 1]$ . Note that all grey values are mapped to the interval  $(0, 1)$  to ensure the validity of our model before processing. The grid position of the  $i$ -th image pixel is given by the vector  $\mathbf{x}_i$  whereas  $v_i$  denotes the corresponding grey value. Subsequently, we will see that a well-considered choice of the weighting matrix  $\tilde{\mathbf{W}}$  allows either to enhance the *global* or the *local* contrast of a given image.

#### Global Contrast Enhancement

For global contrast enhancement we make use of the global model as discussed in Section 4.3.2. Only the  $N$  different occurring grey values  $v_i$  – and not their positions in the image – are considered. We let every entry  $\tilde{w}_{i,j}$  of the weighting matrix denote the frequency of grey value  $v_j$  in the image. Assuming an 8-bit greyscale image this leads to a weighting matrix of size  $256 \times 256$  which is independent of the image size. As illustrated in Figure 4.5, global contrast enhancement can be achieved in two ways: As a first option one can use the explicit scheme (4.50) to describe the evolution of all grey values  $v_i$  up to some time  $t$  (see column two of Figure 4.5). The amount of contrast enhancement grows with increasing values of  $t$ . In our experiments an image size of  $481 \times 321$  pixels and the application of the flux function  $\Phi_{1,1}$  with  $L_\Phi = 1$  imply an upper bound of  $1/(2 \cdot 481 \cdot 321)$  for  $\alpha$ . Thus, we can achieve the time  $t = 2 \cdot 10^{-6}$  in Figure 4.5 in a single iteration. If one is only interested in an enhanced version of the original image with maximum global contrast there is an alternative, namely the derived steady state solution for linear flux functions (4.46). The results are shown in the last column of Figure 4.5. This figure also confirms that the solution of the explicit scheme (4.50) converges to the steady-state solution (4.46) for  $t \rightarrow \infty$ . From (4.46) it is clear that this steady state is equivalent to histogram equalisation. In summary, this means that the application of our global model to greyscale images offers an evolution equation histogram equalisation which allows to control the amount of contrast enhancement in an intuitive way through the time parameter  $t$ .

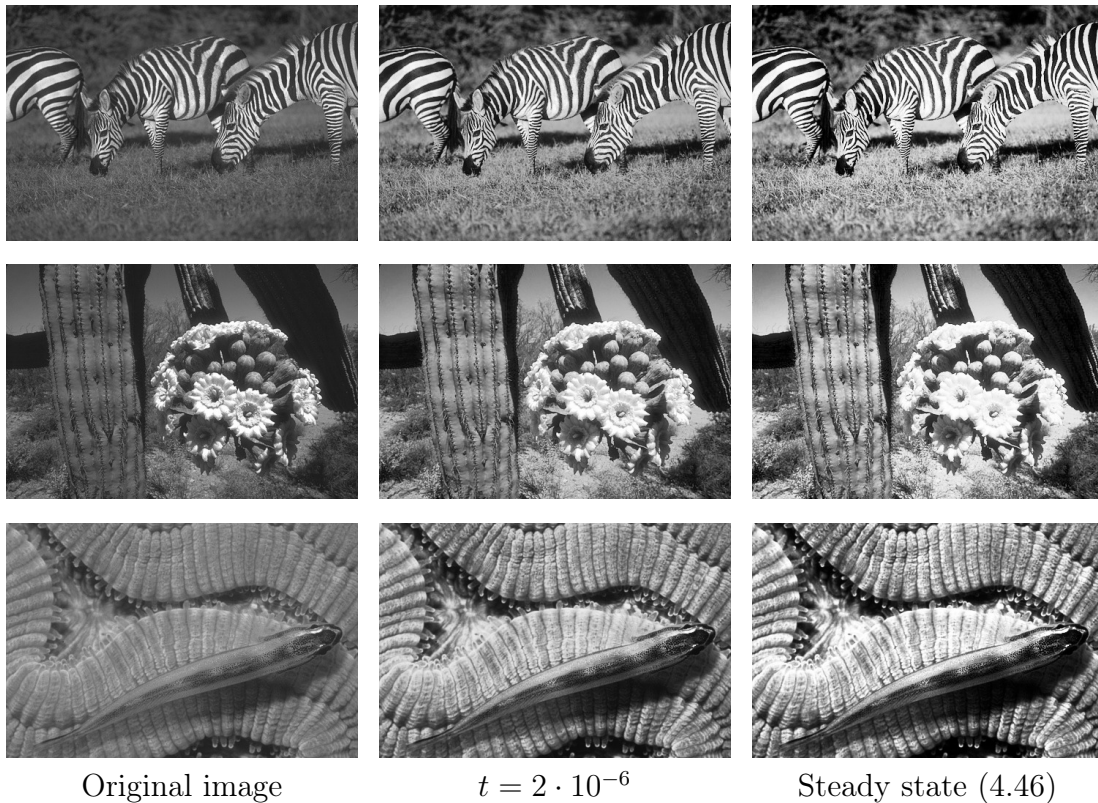


Figure 4.5: Global contrast enhancement using  $\Phi = \Phi_{1,1}$  and greyscale versions of images from the BSDS500 [AMFM11].

### Local Contrast Enhancement

In order to achieve local contrast enhancement we use our model to describe the evolution of grey values  $v_i$  at all  $n \cdot m$  image grid positions. The change of every grey value  $v_i$  depends on all grey values within a disk-shaped neighbourhood of radius  $\varrho$  around its grid position  $\mathbf{x}_i$ . We assume that

$$\tilde{w}_{i,j} := \gamma(|\mathbf{x}_j - \mathbf{x}_i|), \quad \forall i, j \in \{1, 2, \dots, N\}, \quad (4.69)$$

where we weight the spatial distance  $|\mathbf{x}_j - \mathbf{x}_i|$  by a function  $\gamma : \mathbb{R}_0^+ \rightarrow [0, 1]$  with compact support  $[0, \varrho)$  which fulfils

$$\begin{aligned} \gamma(x) &\in (0, 1], & \text{if } x < \varrho, \\ \gamma(x) &= 0, & \text{if } x \geq \varrho. \end{aligned} \quad (4.70)$$

The choice of  $\gamma$  is application dependent. However, it usually makes sense to define  $\gamma(x)$  as a non-increasing function in  $x$ . Possible choices are e.g.

$$\gamma_1(x) = \begin{cases} 1, & \text{if } x < \varrho, \\ 0, & \text{else,} \end{cases} \quad (4.71)$$

$$\gamma_2(x) = \begin{cases} 1 - 6\frac{x^2}{\varrho^2} + 6\frac{x^3}{\varrho^3}, & \text{if } 0 \leq x < \frac{\varrho}{2}, \\ 2 \cdot (1 - \frac{x}{\varrho})^3, & \text{if } \frac{\varrho}{2} \leq x < \varrho, \\ 0, & \text{else,} \end{cases} \quad (4.72)$$

which are both sketched in Figure 4.6. When applying this local model to images

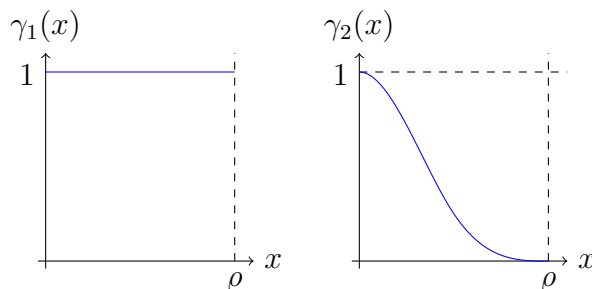


Figure 4.6: Box function  $\gamma_1$  and scaled cubic B-spline  $\gamma_2$ .

we make use of mirroring boundary conditions in order to avoid artefacts at the image boundaries. Figure 4.7 provides an example for local contrast enhancement of digital greyscale images. Again, we describe the grey value evolution with the explicit scheme (4.50). Furthermore, we use  $\gamma_1$  to model the influence of neighbouring grey values. As is evident from Figure 4.7, increasing the values for  $t$  goes along with enhanced local contrast.

## 4.5.2 Colour Images

Based on the assumption that our input data is given in sRGB colour space [SACM96, Int99] (in the following denoted by RGB) we represent a digital colour image by the mapping  $f : \{1, \dots, n\} \times \{1, \dots, m\} \rightarrow [0, 1]^3$ . Subsequently, our aim is the contrast enhancement of digital colour images without distorting the colour information. This means that we only want to adapt the *luminance* but not the *chromaticity* of a given image. For this purpose, we convert the given image data to YCbCr colour space [Pra01, Section 3.5] since this representation provides a separate luminance channel. Next, we perform contrast enhancement on the luminance channel only. Just as for greyscale images we map all Y-values to the interval  $(0, 1)$  to fulfil our model requirements. After enhancing the contrast, we transform the colour information of the image back to RGB colour space.

At this point it is important to mention that the colour gamut of the RGB colour space is a subset of the YCbCr colour gamut and during the conversion process of colour coordinates from YCbCr to RGB colour space the so-called colour gamut problem may occur: Colours from the YCbCr colour gamut may lie outside the RGB colour gamut and thus cannot be represented in RGB colour coordinates. Naik and Murthy [NM03] state that a simple clipping of the values to the bounds creates undesired shift of hue and may lead to colour artefacts. In order to avoid the colour gamut problem we adapt the ideas presented by Nikolova and Steidl [NS14a] which are based on the intensity representation of the HSI colour space [GW08, Section 6.2.3]. Using the original and enhanced intensities, they define an affine colour mapping and transform the original RGB values. This preserves the hue and results in an enhanced RGB image. It is straightforward to show that their algorithms are valid for any intensity  $\hat{f}$  of type

$$\hat{f} = c_r \cdot r + c_g \cdot g + c_b \cdot b, \quad (4.73)$$

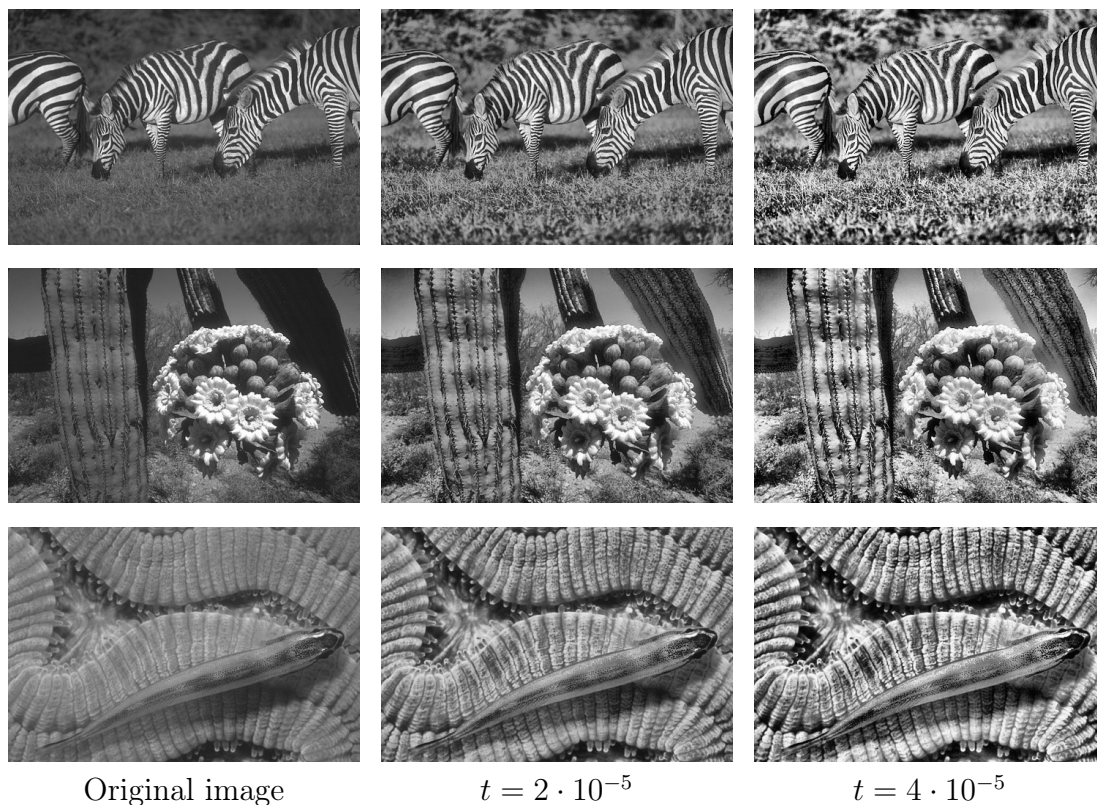


Figure 4.7: Local contrast enhancement using  $\Phi = \Phi_{1,1}$ ,  $\gamma = \gamma_1$ ,  $\varrho = 60$ , and greyscale versions of images from the BSDS500 [AMFM11].

with  $c_r + c_g + c_b = 1$  and  $c_r, c_g, c_b \in [0, 1]$ , where  $r, g$ , and  $b$  denote RGB colour coordinates. Thus, they are applicable to the luminance representation of the YCbCr colour space, too, i.e.  $c_r = 0.299$ ,  $c_g = 0.587$ ,  $c_b = 0.114$ . Tian and Cohen make use of the same idea in [TC17]. As in [NS14a], our result image is a convex combination of the outcomes of a multiplicative and an additive algorithm (see [NS14a, Algorithm 4 and 5]) with coefficients  $\lambda$  and  $1 - \lambda$  for  $\lambda \in [0, 1]$ . During our experiments we use a fixed value of  $\lambda = 0.5$  (for details on how to choose  $\lambda$  we refer to [NS14a]). An overview of our strategy for contrast enhancement of digital colour value images is given in Figure 4.8.

### Global Contrast Enhancement

Again, we apply the global model from Section 4.3.2 in order to achieve global contrast enhancement. As mentioned before, we consider the  $N$  different occurring Y-values of the YCbCr representation of the input image and denote them by  $v_i$  (similar to Section 4.5.1 we neglect their positions in the image). Every entry of the weighting matrix  $\tilde{w}_{i,j}$  contains the number of occurrences of the value  $v_j$  in the Y-channel of the image. It becomes clear that the application of our model – in this setting – basically comes down to histogram equalisation of the Y-channel. Figure 4.9 shows the resulting RGB images after global contrast enhancement. Similar to the greyscale scenario, we can either apply the explicit scheme (4.50) or – for  $\Phi = \Phi_{a,1}$  – estimate the steady state solution following

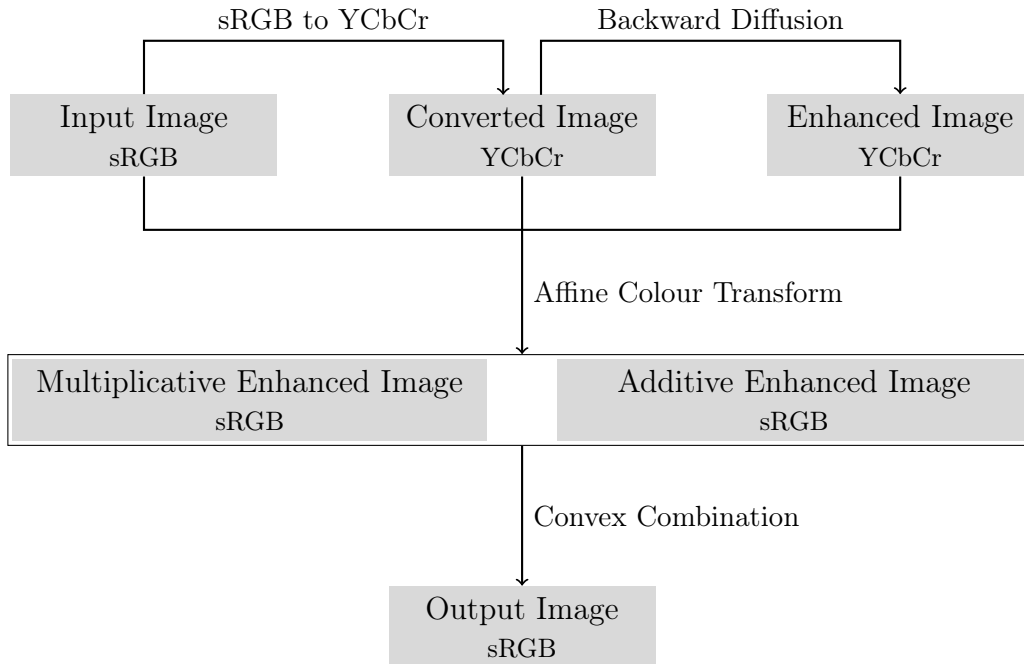


Figure 4.8: Procedure of contrast enhancement for digital colour images following [NS14a].

(4.46). For the first case the amount of contrast enhancement grows with the positive time parameter  $t$ . The second column of Figure 4.9 shows the results for  $\Phi = \Phi_{1,1}$  given time  $t$ . The corresponding steady state solutions are illustrated in the last column of Figure 4.9.

### Local Contrast Enhancement

In a similar manner – and adapting the ideas from Subsection 4.5.1 – we achieve local contrast enhancement in colour images. For this purpose we describe the evolution of Y-values  $v_i$  at all  $n \cdot m$  image grid positions using a disk-shaped neighbourhood of radius  $\varrho$  around the corresponding grid positions  $\mathbf{x}_i$ . The entries of the weighting matrix  $\tilde{\mathbf{W}}$  follow (4.69). In combination with mirrored boundary conditions the explicit scheme (4.50) allows to increase the local contrast of an image with growing  $t$ . Figure 4.10 shows exemplary results for  $\Phi = \Phi_{1,1}$  and  $\gamma = \gamma_1$  (cf. (4.71)). Note, how well – in comparison to the global set-up in Figure 4.9 – the structure of the door gets enhanced while the details of the door knob are preserved. The differences are even larger in the second image: For both the couple in the foreground and the background scenery, contrast increases which implies visibility also for larger times  $t$ .

### 4.5.3 Parameters

In total, our model has up to six parameters:  $\Phi$ ,  $\alpha$ ,  $t$ ,  $\lambda$ ,  $\varrho$ , and  $\gamma$ . During our experiments we have fixed  $\Phi(s)$  to the linear flux function  $\Phi_{1,1}(s)$  and  $\lambda$  to 0.5. Valid bounds for the time step size  $\alpha$  are given in Theorems 9–11. From the



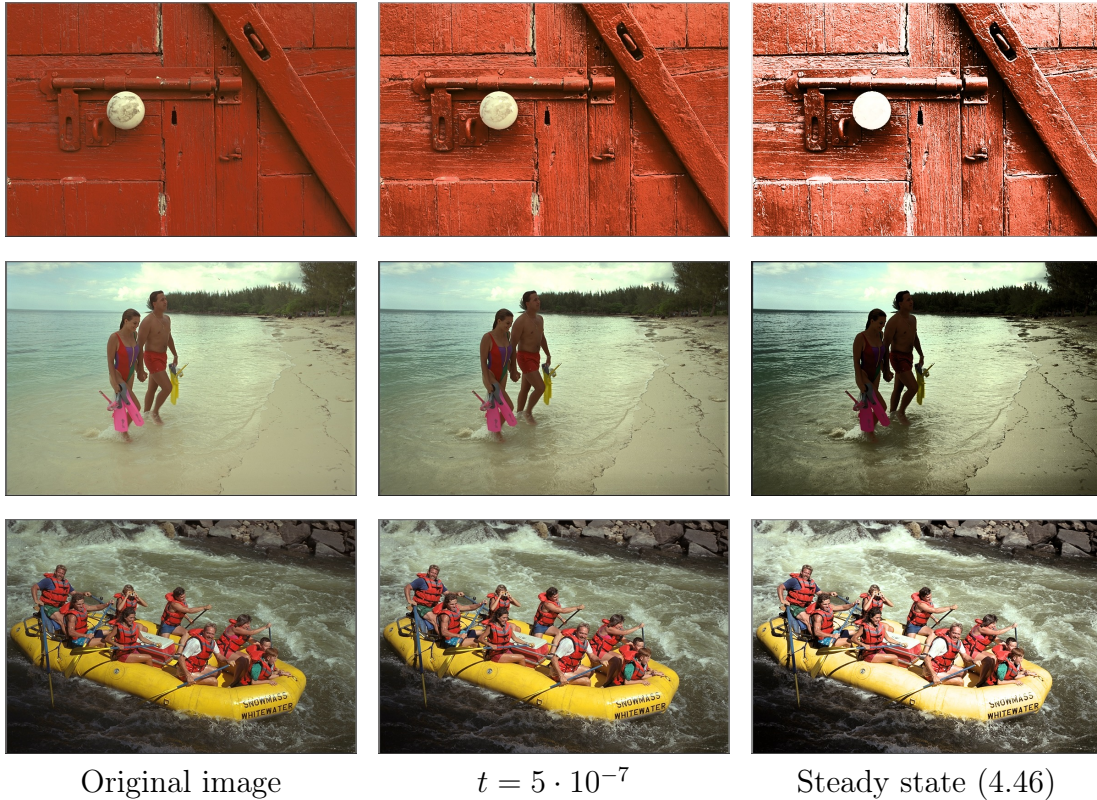


Figure 4.9: Global contrast enhancement using  $\Phi = \Phi_{1,1}$ ,  $\lambda = 0.5$ , and images from [Kod].

theory in Section 4.3 and the subsequent experiments on greyscale and colour images it becomes clear that the amount of contrast enhancement grows with the diffusion time. Thus, it remains to discuss the influence of the parameters  $\varrho$  and  $\gamma$ . We found out that the neighbourhood radius  $\varrho$  affects the diffusion time and controls the amount of perceived local contrast enhancement, i.e. it steers the localisation of the contrast enhancement process. Whereas small radii lead to high contrast in already small image areas, the size of image sections with high contrast increases with  $\varrho$ . For sufficiently large values of  $\varrho$  global histogram equalisation is approximated. Another interesting point is the choice of the weighting function  $\gamma$ . Overall, choosing  $\gamma = \gamma_1$  leads to more homogeneous contrast enhancement resulting in smoother perception. For  $\gamma = \gamma_2$  the focus always lies on the neighbourhood centre which implies even more enhancement of local structures than in the preceding case. We provide exemplary results in Figure 4.11. In summary,  $\gamma_2$  leads to more enhancement which, however, also creates undesired effects in smooth or noisy regions. Thus, we prefer  $\gamma_1$  over  $\gamma_2$ . Further experiments which visualise the effect of the parameters can be found in the supplementary material in Section 4.A.2.

#### 4.5.4 Related Work from an Application Perspective

Now that we have demonstrated the applicability of our model to digital images we want to discuss briefly its relation to other existing theories in the context of



Figure 4.10: Local contrast enhancement using  $\Phi = \Phi_{1,1}$ ,  $\gamma = \gamma_1$ ,  $\varrho = 60$ ,  $\lambda = 0.5$ , and images from [Kod].

image processing.

As mentioned in Section 4.5.1, applying the global model – the entries of  $\tilde{\mathbf{W}}$  representing the grey value frequencies – is identical to histogram equalisation (a common formulation can e.g. be found in [GW08]). Furthermore, there exist other closely related histogram specification techniques – such as [SC97, NS14b, NWC13] – which can have the same steady state. If we compare our evolution with the histogram modification flow introduced by Sapiro and Caselles [SC97], we see that their flow can also be translated into a combination of repulsion among grey-values and a barrier function. However, in [SC97] the repulsive force is constant, and the barrier function quadratic. Thus, they cannot be derived from the same kind of interaction between the  $v_i$  and their reflected counterparts as in our paper.

Referring to Section 4.5.1, there also exist well-known approaches which aim to enhance the local image contrast such as adaptive histogram equalisation – see [PAA<sup>+</sup>87] and the references therein – or contrast limited adaptive histogram equalisation [Zui94]. The latter technique tries to overcome the over-amplification of noise in mostly homogeneous image regions when using adaptive histogram equalisation. Both approaches share the basic idea with our approach in Section 4.5.1 and perform histogram equalisation for each pixel, i.e. the mapping function for every pixel is determined using a neighbourhood of predefined size and its corresponding histogram.



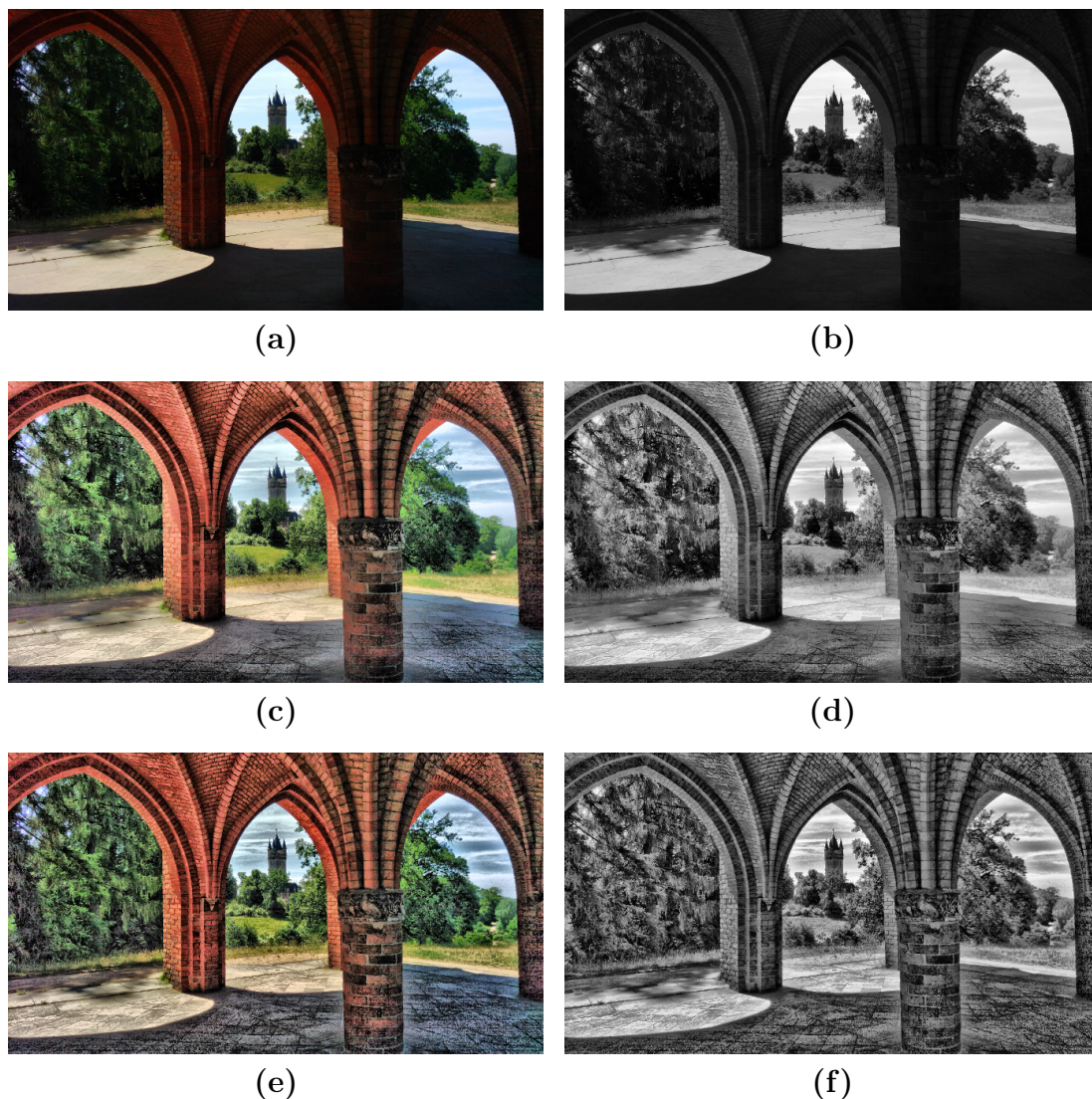


Figure 4.11: **(a)**: Original image ( $683 \times 384$  px) of Flatowturm (Potsdam, Germany) taken by the author. **(b)**: Greyscale version. **(c)**: Version with locally enhanced contrast using  $t = 3 \cdot 10^{-5}$ ,  $\rho = 60$ ,  $\gamma = \gamma_1$ . **(d)**: Greyscale version with locally enhanced contrast using  $t = 3 \cdot 10^{-5}$ ,  $\rho = 60$ ,  $\gamma = \gamma_1$ . **(e)**: Version with locally enhanced contrast using  $t = 20 \cdot 10^{-5}$ ,  $\rho = 60$ ,  $\gamma = \gamma_2$ . **(f)**: Greyscale version with locally enhanced contrast using  $t = 20 \cdot 10^{-5}$ ,  $\rho = 60$ ,  $\gamma = \gamma_2$ .

Another related research topic is the rich field of colour image enhancement which we broach in Section 4.5.2. A short review of existing methods – as well as two new ideas – is presented in [BK07]. Therein, Bassiou and Kotropoulos also mention the colour gamut problem for methods which perform contrast enhancement in a different colour space and transform colour coordinates to RGB afterwards. Of particular interest are the publications by Naik and Murthy [NM03] and Nikolova and Steidl [NS14a] whose ideas are used in Section 4.5.2. Both of them suggest – based on an affine colour transform – strategies to overcome the colour gamut problem while avoiding colour artefacts in the resulting image. A recent approach which also makes use of these ideas is presented by Tian and Cohen [TC17]. Ojo

et al. [OSA16] make use of the HSV colour space to avoid the colour gamut problem when enhancing the contrast of colour images. A variational approach for contrast enhancement which tries to approximate the hue of the input image was recently published by Pierre et al. [PAB<sup>+</sup>17].

## 4.6 Conclusions and Outlook

In this chapter we have presented a mathematical model which describes pure backward diffusion as gradient descent of strictly convex energies. The underlying evolution makes use of ideas from the area of collective behaviour and – in terms of the latter – our model can be understood as a fully repulsive discrete first order swarm model. Not only it is surprising that our model allows backward diffusion to be formulated as a convex optimisation problem but also that it is sufficient to impose reflecting boundary conditions in the diffusion co-domain in order to guarantee stability. This strategy is contrary to existing approaches which either assume forward or zero diffusion at extrema or add classical fidelity terms to avoid instabilities. Furthermore, discretisation of our model does not require sophisticated numerics. We have proven that a straightforward explicit scheme is sufficient to preserve the stability of the time-continuous evolution. In our experiments, we show that our model can directly be applied to contrast enhancement of digital greyscale and colour images.

We see our contribution mainly as an *example* of stable modelling of backward parabolic evolutions that create neither theoretical nor numerical problems. We are convinced that this concept has far more widespread applications in inverse problems, image processing, and computer vision. Exploring them will be part of our future research.

## 4.A Supplementary Material

### 4.A.1 Derivations

#### Derivation of Equation (4.6)

First, Equation (4.4) can be reformulated as

$$\begin{aligned}
E(\mathbf{v}, \mathbf{W}) &= \frac{1}{4} \cdot \sum_{i=1}^{2N} \sum_{j=1}^{2N} w_{i,j} \cdot \Psi((v_j - v_i)^2) \\
&= \frac{1}{4} \cdot \left( \sum_{i=1}^N \sum_{j=1}^{2N} w_{i,j} \cdot \Psi((v_j - v_i)^2) \right. \\
&\quad \left. + \sum_{i=1}^N \sum_{j=1}^{2N} w_{2N+1-i,j} \cdot \Psi((v_j - 2 + v_i)^2) \right) \\
&= \frac{1}{4} \cdot \sum_{i=1}^N \sum_{j=1}^N \left( w_{i,j} \cdot \Psi((v_j - v_i)^2) \right. \\
&\quad \left. + w_{i,2N+1-j} \cdot \Psi((2 - v_j - v_i)^2) \right. \\
&\quad \left. + w_{2N+1-i,j} \cdot \Psi((v_j - 2 + v_i)^2) \right. \\
&\quad \left. + w_{2N+1-i,2N+1-j} \cdot \Psi((2 - v_j - 2 + v_i)^2) \right).
\end{aligned}$$

Using

$$\Psi((2 + s)^2) = \Psi(s^2) = \Psi((-s)^2),$$

and

$$w_{i,j} = w_{2N+1-i,j} = w_{i,2N+1-j} = w_{2N+1-i,2N+1-j},$$

the energy simplifies to

$$E(\mathbf{v}, \mathbf{W}) = \frac{1}{2} \cdot \sum_{i=1}^N \sum_{j=1}^N w_{i,j} \cdot (\Psi((v_j - v_i)^2) + \Psi((v_j + v_i)^2)).$$

#### Positive (Semi-)Definiteness of the Hessian Matrix in Remark 1

Assuming a penaliser function  $\Psi(s^2) = \Psi_{a,n}(s^2)$  according to Table 4.1, the flux function and its derivative read

$$\begin{aligned}
\Phi(s) &= a \cdot n \cdot (s - 1)^{2n-1}, \\
\Phi'(s) &= a \cdot n \cdot (2n - 1) \cdot (s - 1)^{2n-2}.
\end{aligned}$$

Therefore, the entries of the Hessian (4.11) and (4.12) adapt to

$$\begin{aligned}\partial_{v_i v_i} E(\mathbf{v}, \tilde{\mathbf{W}}) &= a \cdot n \cdot (2n - 1) \cdot \\ &\quad \left( \sum_{j \in J_2^i} \tilde{w}_{i,j} \cdot \left( (v_j - v_i - 1)^{2n-2} + (v_j + v_i - 1)^{2n-2} \right) + \right. \\ &\quad \left. \sum_{j \in J_3^i} \tilde{w}_{i,j} \cdot (v_j + v_i - 1)^{2n-2} \right), \\ \partial_{v_i v_j} E(\mathbf{v}, \tilde{\mathbf{W}}) &= a \cdot n \cdot (2n - 1) \cdot \tilde{w}_{i,j} \cdot \\ &\quad \left( (v_j + v_i - 1)^{2n-2} - (v_j - v_i - 1)^{2n-2} \right), \quad \forall j \in J_2^i, \\ \partial_{v_i v_j} E(\mathbf{v}, \tilde{\mathbf{W}}) &= a \cdot n \cdot (2n - 1) \cdot \tilde{w}_{i,j} \cdot (v_j + v_i - 1)^{2n-2}, \quad \forall j \in J_3^i.\end{aligned}$$

Using the Gershgorin circle theorem it is now possible to derive the range of all eigenvalues of the Hessian matrix. The radius of the Gershgorin discs is given by

$$\begin{aligned}r_i &= \sum_{\substack{j=1 \\ j \neq i}}^N |\partial_{v_i v_j} E(\mathbf{v}, \tilde{\mathbf{W}})| \\ &= a \cdot n \cdot (2n - 1) \cdot \\ &\quad \left( \sum_{j \in J_2^i} \tilde{w}_{i,j} \cdot \left| (v_j + v_i - 1)^{2n-2} - (v_j - v_i - 1)^{2n-2} \right| + \right. \\ &\quad \left. \sum_{\substack{j \in J_3^i \\ j \neq i}} \tilde{w}_{i,j} \cdot (v_j + v_i - 1)^{2n-2} \right), \quad \forall i = 1, \dots, N.\end{aligned}$$

Note that the difference  $d_i := \partial_{v_i v_i} E(\mathbf{v}, \tilde{\mathbf{W}}) - r_i$  fulfils

$$\begin{aligned}d_i &= a \cdot n \cdot (2n - 1) \cdot \\ &\quad \left( \sum_{j \in J_2^i} \tilde{w}_{i,j} \cdot \left( (v_j - v_i - 1)^{2n-2} + (v_j + v_i - 1)^{2n-2} - \right. \right. \\ &\quad \left. \left. \left| (v_j + v_i - 1)^{2n-2} - (v_j - v_i - 1)^{2n-2} \right| \right) + \right. \\ &\quad \left. \tilde{w}_{i,i} \cdot (2v_i - 1)^{2n-2} \right) \\ &\geq a \cdot n \cdot (2n - 1) \cdot \tilde{w}_{i,i} \cdot (2v_i - 1)^{2n-2} \\ &\geq 0, \quad \forall i = 1, \dots, N,\end{aligned}$$

where we have used the triangle inequality and the fact that  $\tilde{w}_{i,i} > 0$  and  $v_i \in (0, 1)$ . From the theory of Gershgorin it is known that  $\lambda_i \geq d_i$  for  $1 \leq i \leq N$ . Therefore, the eigenvalues of the Hessian are non-negative and the Hessian is positive semi-definite.

**Case  $n = 1$ .** For  $n = 1$  the difference  $d_i$  satisfies

$$d_i = a \cdot \left( 2 \cdot \sum_{j \in J_2^i} \tilde{w}_{i,j} + \tilde{w}_{i,i} \right) > 0, \quad \forall i = 1, \dots, N,$$

since  $\tilde{w}_{i,i} > 0$  and as a consequence of  $\lambda_i \geq d_i$  for  $1 \leq i \leq N$  the Hessian matrix is positive definite.

**Case  $n = 2$ .** For  $n = 2$  the difference  $d_i$  reads

$$\begin{aligned} d_i &= 6 \cdot n \cdot \left( \sum_{j \in J_2^i} \tilde{w}_{i,j} \cdot \left( (v_j - v_i - 1)^2 + (v_j + v_i - 1)^2 \right. \right. \\ &\quad \left. \left. - |(v_j + v_i - 1)^2 - (v_j - v_i - 1)^2| \right) \right. \\ &\quad \left. + \tilde{w}_{i,i} \cdot (2v_i - 1)^2 \right) \\ &= 6 \cdot n \cdot \left( \sum_{j \in J_2^i} \tilde{w}_{i,j} \cdot (2v_j^2 + 2v_i^2 - 4v_j + 2 - 4v_i \cdot |v_j - 1|) \right. \\ &\quad \left. + \tilde{w}_{i,i} \cdot (2v_i - 1)^2 \right). \end{aligned}$$

Since  $v_j \in (0, 1)$  we know that  $|v_j - 1| = 1 - v_j$  and we get

$$\begin{aligned} d_i &= 6 \cdot n \cdot \left( 2 \cdot \sum_{j \in J_2^i} \tilde{w}_{i,j} \cdot (v_j^2 + 2v_j v_i + v_i^2 - 2v_j - 2v_i + 1) \right. \\ &\quad \left. + \tilde{w}_{i,i} \cdot (2v_i - 1)^2 \right) \\ &= 6 \cdot n \cdot \left( 2 \cdot \sum_{j \in J_2^i} \tilde{w}_{i,j} \cdot \left( (v_j + v_i)^2 - 2 \cdot (v_j + v_i) + 1 \right) \right. \\ &\quad \left. + \tilde{w}_{i,i} \cdot (2v_i - 1)^2 \right) \\ &= 6 \cdot n \cdot \left( 2 \cdot \sum_{j \in J_2^i} \tilde{w}_{i,j} \cdot (v_j + v_i - 1)^2 + \tilde{w}_{i,i} \cdot (2v_i - 1)^2 \right). \end{aligned}$$

Therefore, if for  $1 \leq i \leq N$  at least one of the two conditions

- $v_i \neq \frac{1}{2}$  (since  $\tilde{w}_{i,i} > 0$ ),
- $\exists j \in J_2^i$  with  $v_j \neq 1 - v_i$  and  $\tilde{w}_{i,j} > 0$ ,

holds, one can guarantee  $\lambda_i \geq d_i > 0$  and thus positive definiteness of the Hessian matrix.

**Derivation of Equation (4.39)**

Using  $\tilde{\mathbf{W}} = \mathbf{1}\mathbf{1}^\top$ , (4.4) adapts to

$$\begin{aligned}
 E(\mathbf{v}) &= \frac{1}{4} \cdot \sum_{i=1}^{2N} \sum_{j=1}^{2N} \Psi((v_j - v_i)^2) \\
 &= \frac{1}{4} \cdot \left( \sum_{i=1}^N \sum_{j=1}^N \Psi((v_j - v_i)^2) + \sum_{i=1}^N \sum_{j=1}^N \Psi((2 - v_j - v_i)^2) \right. \\
 &\quad \left. + \sum_{i=1}^N \sum_{j=1}^N \Psi((v_j - 2 + v_i)^2) + \sum_{i=1}^N \sum_{j=1}^N \Psi((2 - v_j - 2 + v_i)^2) \right) \\
 &= \frac{1}{2} \cdot \left( \sum_{i=1}^N \sum_{j=1}^N \Psi((v_j - v_i)^2) + \sum_{i=1}^N \sum_{j=1}^N \Psi((v_j + v_i)^2) \right).
 \end{aligned}$$

Splitting the sums into  $i < j$ ,  $i = j$ , and  $i > j$  we get

$$\begin{aligned}
 E(\mathbf{v}) &= \frac{1}{2} \cdot \left( \sum_{i=1}^{N-1} \sum_{j=i+1}^N \Psi((v_j - v_i)^2) + \sum_{i=1}^N \Psi(0) + \sum_{j=1}^{N-1} \sum_{i=j+1}^N \Psi((v_j - v_i)^2) \right. \\
 &\quad \left. + \sum_{i=1}^{N-1} \sum_{j=i+1}^N \Psi((v_j + v_i)^2) + \sum_{i=1}^N \Psi(4v_i^2) + \sum_{j=1}^{N-1} \sum_{i=j+1}^N \Psi((v_j + v_i)^2) \right).
 \end{aligned}$$

Finally, use  $\Phi(0) = 0$ , switch  $i$  and  $j$  in the third term of each row, and use  $(v_j - v_i)^2 = (v_i - v_j)^2$  to obtain

$$\begin{aligned}
 E(\mathbf{v}) &= \frac{1}{2} \cdot \left( 2 \cdot \sum_{i=1}^{N-1} \sum_{j=i+1}^N \Psi((v_j - v_i)^2) + \sum_{i=1}^N \Psi(4v_i^2) \right. \\
 &\quad \left. + 2 \cdot \sum_{i=1}^{N-1} \sum_{j=i+1}^N \Psi((v_j + v_i)^2) \right) \\
 &= \sum_{i=1}^{N-1} \sum_{j=i+1}^N \Psi((v_j - v_i)^2) + \frac{1}{2} \cdot \sum_{i=1}^N \Psi(4v_i^2) + \sum_{i=1}^{N-1} \sum_{j=i+1}^N \Psi((v_j + v_i)^2).
 \end{aligned}$$



### 4.A.2 Parameters for Local Contrast Enhancement

Subsequently, we illustrate the influence of the parameters  $t$ ,  $\varrho$ , and  $\gamma$  on the results of our local contrast enhancement model while keeping  $\Phi = \Phi_{1,1}$  and  $\lambda = 0.5$  fixed. In Figure 4.12, we illustrate the relation of  $t$  and  $\varrho$  and its effect on the resulting contrast enhanced image. In Figure 4.13, we show how the results differ between weighting function  $\gamma_1$  and  $\gamma_2$ .



Figure 4.12: Relation of  $t$  and  $\varrho$  when applying our model to greyscale images using  $\gamma = \gamma_1$ . Time increases from bottom to top. Radius increases from left to right.

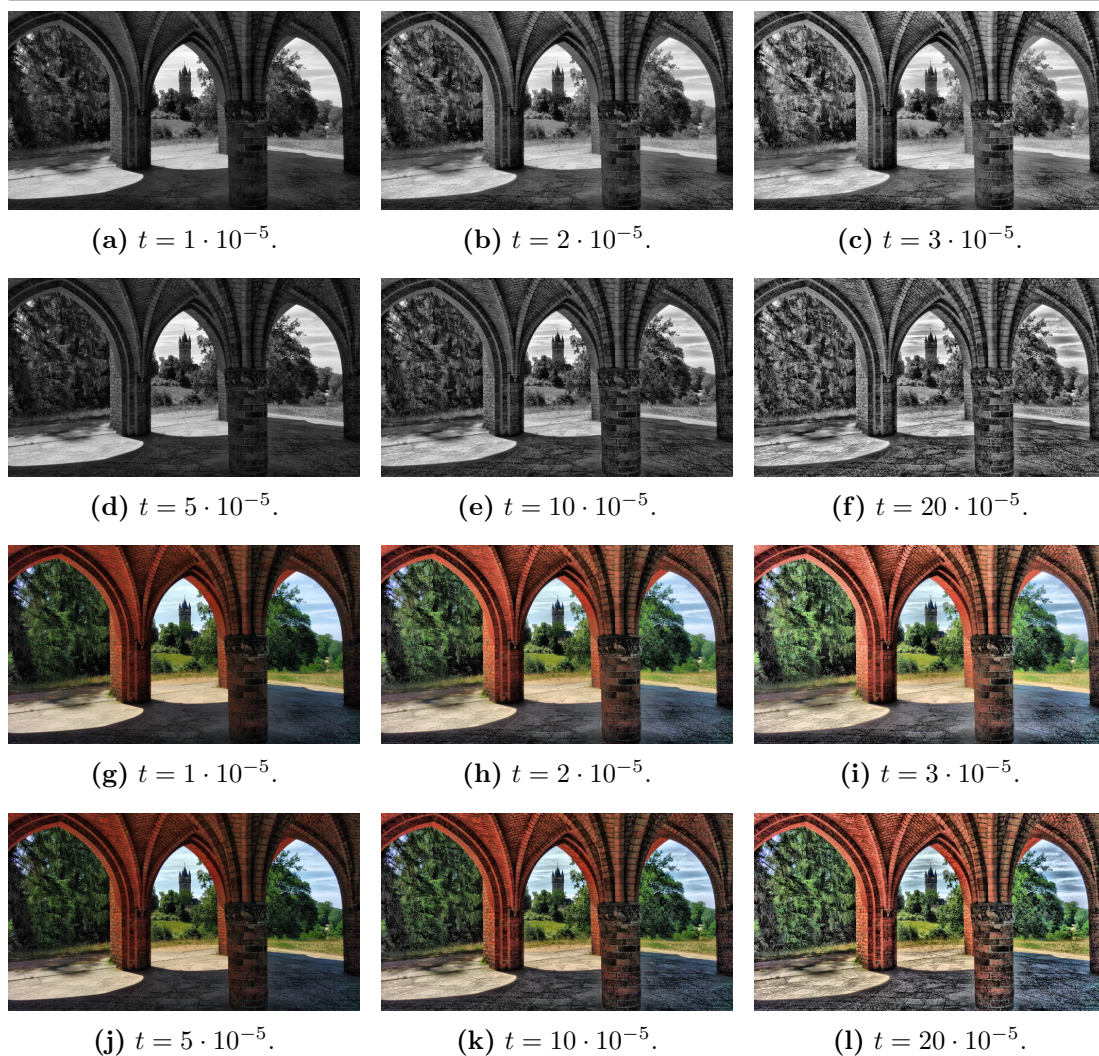


Figure 4.13: Influence of the weighting function  $\gamma$  when applying our model to greyscale and colour images using  $\varrho = 60$  and  $\lambda = 0.5$ . **(a)-(c)**: greyscale input data and  $\gamma = \gamma_1$ . **(d)-(f)**: greyscale input data and  $\gamma = \gamma_2$ . **(g)-(i)**: colour input data and  $\gamma = \gamma_1$ . **(j)-(l)**: colour input data and  $\gamma = \gamma_2$ .

---

# Chapter 5

## Evolutions for One-Dimensional Signal Approximation

“Although this may seem a paradox, all exact science is dominated by the idea of approximation. When a man tells you that he knows the exact truth about anything, you are safe in inferring that he is an inexact man.”

---

Bertrand Russell, *The Scientific Outlook*

### Contents

---

<b>5.1</b>	<b>Introduction</b>	<b>76</b>
<b>5.2</b>	<b>Modelling One-Dimensional Signal Approximation</b>	<b>78</b>
5.2.1	Problem Statement	78
5.2.2	The Approximation Functions $u$	78
5.2.3	Digital Input Signals $f$	84
<b>5.3</b>	<b>The Dar–Bruckstein Method</b>	<b>88</b>
5.3.1	Compact Reformulation of the Dar–Bruckstein Method	89
5.3.2	Limitations of the Dar–Bruckstein Method	90
<b>5.4</b>	<b>Direct Energy Optimisation</b>	<b>91</b>
5.4.1	Particle Swarm Optimisation (PSO)	91
5.4.2	First-Order Optimisation Methods	92
<b>5.5</b>	<b>Experiments</b>	<b>97</b>
5.5.1	Piecewise Constant Approximation Functions $u_c(x)$	98
5.5.2	Piecewise Linear Approximation Functions $u_\ell(x)$	116
<b>5.6</b>	<b>Conclusions and Outlook</b>	<b>120</b>

---

This chapter is dedicated to the problem of finding optimal piecewise constant and linear approximations for arbitrary one-dimensional signals and extends the conference publication [BWD19] which is joint work with Joachim Weickert and Yehuda Dar. Motivated from a compression context these approximations should

consist of a specified number of samples and minimise the mean squared error to the original signal. We formulate this approximation estimation task in terms of a discrete energy minimisation problem which turns out to be – in general – nonconvex. Besides the generic formulation we derive a specific energy for the case of piecewise constant and linear approximation signals.

Initially, we restrict ourselves to piecewise constant approximations and discuss suitable minimisation strategies. In this context we reformulate a recent adaptive sampling method by Dar and Bruckstein [DB19] in a compact and transparent way. This allows us to analyse its limitations when it comes to violations of its three key assumptions: signal smoothness, local linearity, and error balancing. As a remedy, we propose a direct energy optimisation approach which does not rely on any of these assumptions and employs a particle swarm optimisation algorithm. Furthermore, we investigate the applicability of first-order optimisation methods. Our experiments show that for nonsmooth signals or low sample numbers, the direct optimisation approach offers substantial qualitative advantages over the Dar–Bruckstein method. Additionally, we observe that the gradient descent algorithm and related methods represent a useful solution strategy for continuous piecewise linear input signals. As a more general contribution, we disprove the optimality of the principle of error balancing for optimising data in the  $\ell^2$ -norm.

In a next step, we discuss our energy-based solution strategy for piecewise linear approximation functions. The increased problem complexity requires to solve a two-staged convex-nonconvex optimisation problem for which we use a combination of the gradient descent and a particle swarm optimisation algorithm to achieve high-quality results. Corresponding experiments can be regarded as a proof of concept and advise to interpolate given discrete input data linearly.

## 5.1 Introduction

Sampling and reconstruction of continuous signals is one of the fundamental concepts in signal processing. On the one hand side there exists the classical sampling theory (see e.g. [Jer77] for a review) which relies on the idea of uniform sampling. It teaches us that signals with limited overall bandwidth can be reconstructed perfectly from discrete data. An extension and localisation of this theory proves that lossless signal reconstruction from nonuniformly sampled data is possible if the sampling rate gets adapted to the local signal bandwidth [Hor68, CPL85, BPP98, WO07, AD19, MA09, AG01]. Signal reconstruction in case of nonuniform samples also plays an important role for the purpose of noise removal. In particular, piecewise constant signals are considered in the literature, see e.g. [VB17, LJ11] and the references therein.

These ideas have heavily influenced the area of signal compression which represents another important and highly relevant application. Aiming at higher compression rates, in particular lossy signal representations become an attractive option. Recently, Dar and Bruckstein [DB19] have introduced a simple and efficient adaptive sampling strategy for approximating 1-D signals by piecewise constant functions. It involves three assumptions: smoothness, local linearity,

and error balancing. In practice, however, signals can be nonsmooth, they can violate local linearity, and the optimality of error balancing is unclear. Thus, finding an optimal approach for the general case remains an open problem.

**Contributions of this Chapter.** Subsequently, we investigate a new approach to one-dimensional function approximation which is closely connected to the fields of adaptive sampling, segmentation, and lossy signal approximation. In this context, we come up with an energy minimisation approach which favours globally optimal piecewise signal approximations that minimise the mean squared error (MSE) (see Chapter 2.1.4). We put special emphasis on digital input signals and adapt the energy to piecewise constant and piecewise linear output signals. In a compression sense our ansatz focusses on the idea of gaining the highest possible approximation quality for a specified and limited number of samples which is directly related to the required file size when storing the data.

For piecewise constant approximations we provide – based on the work by Belhachmi et al. [BBBW09] – an alternative and simpler derivation of the Dar–Bruckstein model. The latter allows us to quantify the effects of violating local linearity and to disprove the optimality of error balancing.

As a remedy, we propose an energy minimisation model which does neither rely on smoothness nor local linearity or error balancing. Furthermore, it is not restricted to a specific type of input and output signal. In this work, we analyse this model in detail for piecewise constant and piecewise linear input signals. With the help of a minimal example we illustrate adequate numerical optimisation strategies to solve the arising nonconvex minimisation problem. For all occurring scenarios, a particle swarm optimisation (PSO) algorithm performs well (see Chapter 2.2.5). Apart from that, we discuss the applicability of first-order methods like the gradient descent method (see Chapter 2.2.1).

With the help of comprehensive experiments on synthetic and real world data we validate our theoretical results for both types of output signals (piecewise constant and piecewise linear signals). Additionally, we find that the quality of our novel approach can exceed the one of the Dar–Bruckstein method.

**Structure of the Chapter.** In Section 5.2, we formulate the general problem statement for one-dimensional signal approximation. We discuss its most important characteristics and adaptation to piecewise constant and linear output signals. Furthermore, we focus on the interpretation of discrete input data in terms of piecewise constant and linear functions. In Section 5.3 we reformulate and analyse the Dar–Bruckstein model. The successive Section 5.4 is dedicated to the evaluation of suitable numerical minimisation techniques for our direct energy optimisation approach. In Section 5.5 we present experiments for smooth and nonsmooth input functions for which we evaluate the efficacy of our proposed energy minimisation approach. This includes – amongst others – a comparison to the Dar–Bruckstein model for piecewise constant approximation functions. We conclude with a summary and outlook in Section 5.6.

## 5.2 Modelling One-Dimensional Signal Approximation

In this section we formalise the approximation problem and highlight possible adaptations to the function type of the desired output signal. Additionally, we discuss the treatment of digital input signals.

### 5.2.1 Problem Statement

In the following we assume that we are given a signal domain  $[a, b] \subset \mathbb{R}$  and some integrable one-dimensional input signal  $f : [a, b] \rightarrow \mathbb{R}$ . Our aim is to approximate  $f$  by a piecewise-defined function  $u : [a, b] \rightarrow \mathbb{R}$  which minimises the mean squared error (MSE) w.r.t.  $f$ . We require that the function  $u$  consists of  $N$  segments. The vector  $\mathbf{x} := (x_0, x_1, \dots, x_N)^T$  contains the positions of all  $N + 1$  segment boundaries which fulfil

$$a =: x_0 < x_1 < \dots < x_{N-1} < x_N := b. \quad (5.1)$$

Based on this, our problem of finding the  $\ell^2$ -optimal approximation  $u$  of  $f$  comes down to minimising the discrete energy

$$E_f(\mathbf{x}, \mathbf{u}) = \frac{1}{b-a} \sum_{i=0}^{N-1} \int_{x_i}^{x_{i+1}} (f(y) - u(y))^2 dy, \quad (5.2)$$

where  $\mathbf{u} \in \mathbb{R}^P$  denotes a vector containing  $P$  samples of  $u$ . Depending on the application, this may be interpreted as function approximation, adaptive sampling, segmentation, or lossy signal compression.

Referring to the input signal  $f$  we define the functions

$$g(x) := f^2(x), \quad (5.3)$$

$$F(x) := \int_a^x f(y) dy, \quad (5.4)$$

$$G(x) := \int_a^x g(y) dy, \quad (5.5)$$

which we use throughout our analysis of the model.

In this work, we focus on two different model adaptations being discussed below: the first one dealing with piecewise constant approximation functions  $u_c(x)$ , another one considering piecewise linear functions  $u_\ell(x)$ .

### 5.2.2 The Approximation Functions $u$

#### Piecewise Constant Approximation Functions $u_c(x)$

In our first scenario, we approximate  $f$  by a piecewise constant function of type

$$u_c(x) := \begin{cases} u_i, & \text{if } x \in [x_i, x_{i+1}) \text{ and } 0 \leq i < N - 1, \\ u_{N-1}, & \text{if } x \in [x_{N-1}, x_N]. \end{cases} \quad (5.6)$$

Consequently, we set  $P = N$  and use  $\mathbf{u} := (u_0, u_1, \dots, u_{N-1})^\top$ . Once the segment boundaries  $\mathbf{x}$  are known, the  $\ell^2$ -optimal approximation of  $f$  is given by the mean value of each sampling interval:

$$u_i := \frac{1}{x_{i+1} - x_i} \int_{x_i}^{x_{i+1}} f(y) dy, \quad \text{for } 0 \leq i \leq N-1. \quad (5.7)$$

We observe that  $u_c$  is completely determined by  $f$  and  $\mathbf{x}$ . This allows us to rewrite the energy (5.2) as

$$E_f(\mathbf{x}) = \frac{1}{b-a} \sum_{i=0}^{N-1} \int_{x_i}^{x_{i+1}} (f(y) - u_i)^2 dy \quad (5.8)$$

$$= \frac{\int_a^b (f(y))^2 dy}{b-a} - \frac{1}{b-a} \sum_{i=0}^{N-1} \frac{\left( \int_{x_i}^{x_{i+1}} f(y) dy \right)^2}{x_{i+1} - x_i}. \quad (5.9)$$

Although (5.8) does not look very complicated, in general the energy is non-smooth, nonconvex, and may have many local minima.

### Piecewise Linear Approximation Functions $u_\ell(x)$

In our second setup we assume that – on each interval  $[x_i, x_{i+1}]$  – the function  $u_\ell(x)$  is given by linear interpolation

$$u_\ell(x) := \frac{x_{i+1} - x}{x_{i+1} - x_i} u_i + \frac{x - x_i}{x_{i+1} - x_i} u_{i+1}, \quad \text{for } x \in [x_i, x_{i+1}], \quad (5.10)$$

where  $i = 0, 1, \dots, N-1$ . This results in a piecewise linear and continuous function  $u_\ell(x)$  on the domain  $[a, b]$  which is based on  $P = N+1$  samples denoted by  $\mathbf{u} := (u_0, u_1, \dots, u_N)^\top$ . Using

$$u_\ell^2(x) = \left( \frac{x_{i+1}u_i - x_i u_{i+1} - x(u_i - u_{i+1})}{x_{i+1} - x_i} \right)^2, \quad \text{for } i = 0, 1, \dots, N-1, \quad (5.11)$$

in combination with (5.2) our approximation problem comes down to minimising the discrete energy

$$E_f(\mathbf{x}, \mathbf{u}) = \frac{\int_a^b f^2(y) dy}{b-a} - \frac{1}{b-a} \sum_{i=0}^{N-1} \frac{\mathcal{S}(f, u_i, u_{i+1}, x_i, x_{i+1})}{x_{i+1} - x_i}, \quad (5.12)$$

which again represents the MSE of  $u$  w.r.t.  $f$ . We denote the contribution of each segment  $[x_i, x_{i+1}]$  by

$$\begin{aligned}
 \mathcal{S}(f, u_i, u_{i+1}, x_i, x_{i+1}) &:= 2(x_{i+1}u_i - x_i u_{i+1}) \int_{x_i}^{x_{i+1}} f(y) dy \\
 &\quad - 2(u_i - u_{i+1}) \int_{x_i}^{x_{i+1}} y f(y) dy \\
 &\quad - (x_{i+1}u_i - x_i u_{i+1})^2 \\
 &\quad + (x_{i+1}u_i - x_i u_{i+1})(u_i - u_{i+1})(x_{i+1} + x_i) \\
 &\quad - \frac{1}{3}(u_i - u_{i+1})^2(x_{i+1}^2 + x_{i+1}x_i + x_i^2), \quad (5.13)
 \end{aligned}$$

where  $i = 0, 1, \dots, N - 1$ . As we have seen in the previous section about piecewise constant approximation functions  $u_c(x)$ , the estimation of ideal values  $\mathbf{u}$  is considerably easier than finding optimal boundary positions  $\mathbf{x}$ . This behaviour carries over to piecewise linear approximation functions  $u_\ell(x)$ . More specifically, let us now show that the corresponding energy (5.12) is strictly convex in  $\mathbf{u}$  which means that for every boundary configuration  $\mathbf{x}$  there exists a unique minimiser  $\mathbf{u}$  of  $E_f(\mathbf{x}, \mathbf{u})$ .

**Theorem 12** (Strict Convexity of  $E_f(\mathbf{x}, \mathbf{u})$  in  $\mathbf{u}$ ). *The energy  $E_f(\mathbf{x}, \mathbf{u})$  is strictly convex in  $\mathbf{u}$ .*

*Proof.* Note, that we make use of  $E := E_f(\mathbf{x}, \mathbf{u})$  throughout this proof for better readability. First, let us consider  $\nabla_{\mathbf{u}} E$ , the gradient of (5.12) w.r.t.  $\mathbf{u}$ . We have

$$\begin{aligned}
 \partial_{u_0} E &= -\frac{1}{(b-a)(x_1-a)} \left( 2x_1 \int_a^{x_1} f(y) dy - 2 \int_a^{x_1} y f(y) dy \right. \\
 &\quad - 2(x_1 u_0 - a u_1) x_1 \\
 &\quad + x_1(u_0 - u_1)(x_1 + a) \\
 &\quad + (x_1 u_0 - a u_1)(x_1 + a) \\
 &\quad \left. - \frac{2}{3}(u_0 - u_1)(x_1^2 + x_1 a + a^2) \right), \quad (5.14)
 \end{aligned}$$



$$\begin{aligned}
 \partial_{u_i} E = & -\frac{1}{(b-a)(x_i - x_{i-1})} \left( -2x_{i-1} \int_{x_{i-1}}^{x_i} f(y) dy + 2 \int_{x_{i-1}}^{x_i} yf(y) dy \right. \\
 & + 2(x_i u_{i-1} - x_{i-1} u_i) x_{i-1} \\
 & - x_{i-1} (u_{i-1} - u_i) (x_i + x_{i-1}) \\
 & - (x_i u_{i-1} - x_{i-1} u_i) (x_i + x_{i-1}) \\
 & \left. + \frac{2}{3} (u_{i-1} - u_i) (x_i^2 + x_i x_{i-1} + x_{i-1}^2) \right) \\
 & - \frac{1}{(b-a)(x_{i+1} - x_i)} \left( 2x_{i+1} \int_{x_i}^{x_{i+1}} f(y) dy - 2 \int_{x_i}^{x_{i+1}} yf(y) dy \right. \\
 & - 2(x_{i+1} u_i - x_i u_{i+1}) x_{i+1} \\
 & + x_{i+1} (u_i - u_{i+1}) (x_{i+1} + x_i) \\
 & + (x_{i+1} u_i - x_i u_{i+1}) (x_{i+1} + x_i) \\
 & \left. - \frac{2}{3} (u_i - u_{i+1}) (x_{i+1}^2 + x_{i+1} x_i + x_i^2) \right),
 \end{aligned} \tag{5.15}$$

$$\begin{aligned}
 \partial_{u_N} E = & -\frac{1}{(b-a)(b - x_{N-1})} \left( -2x_{N-1} \int_{x_{N-1}}^b f(y) dy + 2 \int_{x_{N-1}}^b yf(y) dy \right. \\
 & + 2(bu_{N-1} - x_{N-1} u_N) x_{N-1} \\
 & - x_{N-1} (u_{N-1} - u_N) (b + x_{N-1}) \\
 & - (bu_{N-1} - x_{N-1} u_N) (b + x_{N-1}) \\
 & \left. + \frac{2}{3} (u_{N-1} - u_N) (b^2 + bx_{N-1} + x_{N-1}^2) \right),
 \end{aligned} \tag{5.16}$$

for  $i = 1, 2, \dots, N - 1$ . Based on the elements of the gradient, we derive the Hessian of (5.12) w.r.t.  $\mathbf{u}$ :

$$\begin{aligned}
 \partial_{u_0 u_0} E = & -\frac{-2x_1 + x_1^2 + x_1 a + x_1^2 + x_1 a - \frac{2}{3}(x_1^2 + x_1 a + a^2)}{(b-a)(x_1 - a)} \\
 = & \frac{2(x_1^2 - 2x_1 a + a^2)}{3(b-a)(x_1 - a)} \\
 = & \frac{2(x_1 - a)^2}{3(b-a)(x_1 - a)} \\
 = & \frac{2(x_1 - a)}{3(b-a)},
 \end{aligned} \tag{5.17}$$

$$\begin{aligned}
\partial_{u_0 u_1} E &= -\frac{2ax_1 - x_1^2 - ax_1 - ax_1 - a^2 + \frac{2}{3}(x_1^2 + x_1a + a^2)}{(b-a)(x_1 - a)} \\
&= \frac{x_1^2 - 2x_1a + a^2}{3(b-a)(x_1 - a)} \\
&= \frac{(x_1 - a)^2}{3(b-a)(x_1 - a)} \\
&= \frac{x_1 - a}{3(b-a)}, \tag{5.18}
\end{aligned}$$

$$\begin{aligned}
\partial_{u_i u_{i-1}} E &= -\frac{2x_i x_{i-1} - x_{i-1} x_i - x_{i-1}^2 - x_i^2 - x_i x_{i-1} + \frac{2}{3}(x_i^2 + x_i x_{i-1} + x_{i-1}^2)}{(b-a)(x_i - x_{i-1})} \\
&= \frac{x_{i-1}^2 - 2x_i x_{i-1} + x_i^2}{3(b-a)(x_i - x_{i-1})} \\
&= \frac{(x_i - x_{i-1})^2}{3(b-a)(x_i - x_{i-1})} \\
&= \frac{x_i - x_{i-1}}{3(b-a)}, \tag{5.19}
\end{aligned}$$

$$\begin{aligned}
\partial_{u_i u_i} E &= -\frac{-2x_{i-1}^2 + x_{i-1} x_i + x_{i-1}^2 + x_{i-1} x_i + x_{i-1}^2 - \frac{2}{3}(x_i^2 + x_i x_{i-1} + x_{i-1}^2)}{(b-a)(x_i - x_{i-1})} \\
&\quad -\frac{-2x_{i+1}^2 + x_{i+1}^2 + x_{i+1} x_i + x_{i+1}^2 + x_{i+1} x_i - \frac{2}{3}(x_{i+1}^2 + x_{i+1} x_i + x_i^2)}{(b-a)(x_{i+1} - x_i)} \\
&= \frac{2(x_i^2 - 2x_i x_{i-1} + x_{i-1}^2)}{3(b-a)(x_i - x_{i-1})} + \frac{2(x_{i+1}^2 - 2x_{i+1} x_i + x_i^2)}{3(b-a)(x_{i+1} - x_i)} \\
&= \frac{2(x_i - x_{i-1})^2}{3(b-a)(x_i - x_{i-1})} + \frac{2(x_{i+1} - x_i)^2}{3(b-a)(x_{i+1} - x_i)} \\
&= \frac{2(x_i - x_{i-1} + x_{i+1} - x_i)}{3(b-a)} \\
&= \frac{2(x_{i+1} - x_{i-1})}{3(b-a)}, \tag{5.20}
\end{aligned}$$

$$\begin{aligned}
\partial_{u_i u_{i+1}} E &= -\frac{2x_i x_{i+1} - x_{i+1}^2 - x_{i+1} x_i - x_i x_{i+1} - x_i^2 + \frac{2}{3}(x_{i+1}^2 + x_{i+1} x_i + x_i^2)}{(b-a)(x_{i+1} - x_i)} \\
&= \frac{x_{i+1} - 2x_{i+1} x_i + x_i^2}{3(b-a)(x_{i+1} - x_i)} \\
&= \frac{(x_{i+1} - x_i)^2}{3(b-a)(x_{i+1} - x_i)} \\
&= \frac{x_{i+1} - x_i}{3(b-a)}, \tag{5.21}
\end{aligned}$$

$$\begin{aligned}
 \partial_{u_N u_{N-1}} E &= - \frac{2bx_{N-1} - x_{N-1}b - x_{N-1}^2 - b^2 - bx_{N-1}}{(b-a)(b-x_{N-1})} \\
 &\quad - \frac{\frac{2}{3}(b^2 + bx_{N-1} + x_{N-1}^2)}{(b-a)(b-x_{N-1})} \\
 &= \frac{b^2 - 2bx_{N-1} + x_{N-1}^2}{3(b-a)(b-x_{N-1})} \\
 &= \frac{b - x_{N-1}}{3(b-a)}, \tag{5.22}
 \end{aligned}$$

$$\begin{aligned}
 \partial_{u_N u_N} E &= - \frac{-2x_{N-1}^2 + x_{N-1}b + x_{N-1}^2 + x_{N-1}b + x_{N-1}^2}{(b-a)(b-x_{N-1})} \\
 &\quad + \frac{\frac{2}{3}(b^2 + bx_{N-1} + x_{N-1}^2)}{(b-a)(b-x_{N-1})} \\
 &= \frac{2(b^2 - 2bx_{N-1} + x_{N-1}^2)}{3(b-a)(b-x_{N-1})} \\
 &= \frac{2(b-x_{N-1})^2}{3(b-a)(b-x_{N-1})} \\
 &= \frac{2(b-x_{N-1})}{3(b-a)}, \tag{5.23}
 \end{aligned}$$

where we again assume  $i = 1, 2, \dots, N-1$ . All other entries of the Hessian vanish. Due to the fact that the Hessian is symmetric, all of its eigenvalues are real-valued. Using (2.10), we estimate the Gershgorin radii of the Hessian:

$$r_0 = \frac{x_1 - a}{3(b-a)}, \tag{5.24}$$

$$r_i = \frac{x_{i+1} - x_{i-1}}{3(b-a)}, \quad \text{for } i = 1, 2, \dots, N-1, \tag{5.25}$$

$$r_N = \frac{b - x_{N-1}}{3(b-a)}. \tag{5.26}$$

We make use of the fact that the segment boundaries  $x_0, x_1, \dots, x_N$  are – according to (5.1) – sorted in ascending order. Based on the Gershgorin Circle Theorem (see Chapter 2.1.5) we conclude that every eigenvalue  $\lambda$  of the Hessian fulfils at least one of the following three conditions:

$$0 < \frac{x_1 - a}{3(b-a)} \leq \lambda \leq \frac{x_1 - a}{b-a} \leq 1, \tag{5.27}$$

$$0 < \frac{x_{i+1} - x_{i-1}}{3(b-a)} \leq \lambda \leq \frac{x_{i+1} - x_{i-1}}{b-a} \leq 1, \quad \text{for } i = 1, 2, \dots, N-1, \tag{5.28}$$

$$0 < \frac{b - x_{N-1}}{3(b-a)} \leq \lambda \leq \frac{b - x_{N-1}}{b-a} \leq 1. \tag{5.29}$$

Consequently, all eigenvalues are positive such that the Hessian is positive definite and  $E$  is strictly convex in  $\mathbf{u}$ . This concludes the proof.  $\square$

In order to find the optimal tonal values  $\mathbf{u}$  for a given boundary configuration  $\mathbf{x}$  it suffices to solve the initial value problem

$$\dot{\mathbf{u}}(t) = -\nabla_{\mathbf{u}} E_f(\mathbf{x}, \mathbf{u}) \quad (5.30)$$

$$\mathbf{u}(t) = \mathbf{u}_0 \quad (5.31)$$

which describes a gradient descent process on the energy with arbitrary initial tonal values  $\mathbf{u}_0 \in \mathbb{R}^P$ . As a consequence of Theorem 12, the steady-state of this process

$$\mathbf{u}^* := \lim_{t \rightarrow \infty} \mathbf{u}(t) \quad (5.32)$$

represents a unique minimiser of  $E_f(\mathbf{x}, \mathbf{u})$  for fixed boundary positions  $\mathbf{x}$ . An appropriate technique to estimate  $\mathbf{u}^*$  is e.g. the gradient descent method discussed in Chapter 2.2.1. One can use (5.27)–(5.29) to derive the Lipschitz estimate

$$\tilde{L} := \frac{\max\{x_1 - a, \{x_{i+1} - x_{i-1} \mid i \in [1, N-1]\}, b - x_{N-1}\}}{b - a} \leq 1, \quad (5.33)$$

and to set – according to (2.18) – a time step size  $\alpha$  which guarantees convergence of the gradient descent method. Note, that in this case even the comparatively large time step size  $\alpha = 1$  is being considered stable.

It becomes clear that the estimation of the optimal function  $u_\ell$  – as in case of a piecewise constant function  $u_c$  – only depends on  $f$  and  $\mathbf{x}$ . As a consequence, also the approximation problem for  $u_\ell$  reduces to the task of finding the boundary positions  $\mathbf{x}$  which minimise the energy (5.12). However, this remains a difficult problem since the energy might still be a nonsmooth and nonconvex function with numerous local minima as we will see in Chapter 5.4.

### 5.2.3 Digital Input Signals $f$

Our approximation model covers – amongst others – the important class of digital input signals. In the context of our model, we expect digital input data to be given in terms of a uniformly sampled discrete real-valued input signal with samples  $\mathbf{f} := (f_1, f_2, \dots, f_n)^T \in \mathbb{R}^n$ . Furthermore, we assume that each  $f_i$  represents either a sample of a piecewise constant or piecewise linear function  $f : [a, b] \rightarrow \mathbb{R}$  taken at position

$$p_i = a + \frac{h}{2} + (i-1)h, \quad \text{for } i = 1, 2, \dots, n, \quad (5.34)$$

where  $h$  denotes the sampling distance

$$h := \frac{b-a}{n}. \quad (5.35)$$

For notational convenience we introduce the left and right boundary positions

$$\ell_i := p_i - \frac{h}{2} \quad \text{and} \quad r_i := p_i + \frac{h}{2}, \quad \text{for } i = 1, 2, \dots, n, \quad (5.36)$$

of an interval of width  $h$  with centre  $p_i$ .

Let us now take a look at the specific functions  $f$ ,  $g$ ,  $F$ , and  $G$  for the piecewise constant and piecewise linear case.

### Piecewise Constant Input Signals

The functions  $f(x)$  and  $g(x)$  are given by

$$f(x) = \begin{cases} f_i, & \text{if } x \in [\ell_i, r_i) \text{ and } i = 1, 2, \dots, n-1, \\ f_n, & \text{if } x \in [\ell_n, r_n], \end{cases} \quad (5.37)$$

$$g(x) = \begin{cases} f_i^2, & \text{if } x \in [\ell_i, r_i) \text{ and } i = 1, 2, \dots, n-1, \\ f_n^2, & \text{if } x \in [\ell_n, r_n]. \end{cases} \quad (5.38)$$

For  $y_1, y_2 \in [\ell_i, r_i]$  and  $i = 1, 2, \dots, n$  the integrals of  $f(x)$  and  $g(x)$  simplify to

$$\int_{y_1}^{y_2} f(z) dz = f_i(y_2 - y_1), \quad (5.39)$$

$$\int_{y_1}^{y_2} g(z) dz = f_i^2(y_2 - y_1). \quad (5.40)$$

As a consequence, the functions  $F(x)$  and  $G(x)$  read

$$F(x) = \begin{cases} f_1(x - \ell_1), & \text{if } x \in [\ell_1, r_1), \\ \sum_{j=1}^{i-1} f_j h + f_i(x - \ell_i), & \text{if } x \in [\ell_i, r_i) \text{ and } i = 2, 3, \dots, n-1, \\ \sum_{j=1}^{n-1} f_j h + f_n(x - \ell_n), & \text{if } x \in [\ell_n, r_n], \end{cases} \quad (5.41)$$

$$G(x) = \begin{cases} f_1^2(x - \ell_1), & \text{if } x \in [\ell_1, r_1), \\ \sum_{j=1}^{i-1} f_j^2 h + f_i^2(x - \ell_i), & \text{if } x \in [\ell_i, r_i) \text{ and } i = 2, 3, \dots, n-1, \\ \sum_{j=1}^{n-1} f_j^2 h + f_n^2(x - \ell_n), & \text{if } x \in [\ell_n, r_n]. \end{cases} \quad (5.42)$$

From (5.37) and (5.38) it becomes clear that – in general –  $f$  and  $g$  represent non-continuous functions on the domain  $[a, b]$ . Furthermore, we have  $F \in C^0([a, b])$  and  $G \in C^0([a, b])$ .

**Example Function.** In order to illustrate the properties of our model and to explain reasonable solution strategies we present a simple example which we use throughout the theoretical part of this chapter. We consider the discrete input signal

$$\mathbf{f}_1 := (8, 5.5, 2, 3)^T \quad (5.43)$$

on a domain  $[a, b]$  with  $a := 0$  and  $b := 4$ . The signal consists of  $n = 4$  samples taken at the positions

$$\mathbf{p}_1 := \left( \frac{1}{2}, \frac{3}{2}, \frac{5}{2}, \frac{7}{2} \right)^T \quad (5.44)$$

with sampling distance  $h = 1$ . Using (5.37), (5.38), (5.41), and (5.42) we get

$$f_1(x) := \begin{cases} 8, & \text{if } 0 \leq x < 1, \\ 5.5, & \text{if } 1 \leq x < 2, \\ 2, & \text{if } 2 \leq x < 3, \\ 3, & \text{if } 3 \leq x \leq 4, \end{cases} \quad (5.45)$$

$$g_1(x) := \begin{cases} 64, & \text{if } 0 \leq x < 1, \\ 30.25, & \text{if } 1 \leq x < 2, \\ 4, & \text{if } 2 \leq x < 3, \\ 9, & \text{if } 3 \leq x \leq 4, \end{cases} \quad (5.46)$$

$$F_1(x) := \begin{cases} 8x, & \text{if } 0 \leq x < 1, \\ \frac{11x+5}{2}, & \text{if } 1 \leq x < 2, \\ \frac{4x+19}{2}, & \text{if } 2 \leq x < 3, \\ \frac{6x+13}{2}, & \text{if } 3 \leq x \leq 4, \end{cases} \quad (5.47)$$

$$G_1(x) := \begin{cases} 64x, & \text{if } 0 \leq x < 1, \\ \frac{121x+135}{4}, & \text{if } 1 \leq x < 2, \\ \frac{16x+345}{4}, & \text{if } 2 \leq x < 3, \\ \frac{36x+285}{4}, & \text{if } 3 \leq x \leq 4. \end{cases} \quad (5.48)$$

Plots of all four functions can be found in Figure 5.1.

### Piecewise Linear Input Signals

Similar derivations can be made for a piecewise linear input signal. We get

$$f(x) = \begin{cases} f_1, & \text{if } x \in [a, p_1), \\ m_i x + t_i, & \text{if } x \in [p_i, p_{i+1}) \text{ and } i = 1, 2, \dots, n-1, \\ f_n, & \text{if } x \in [p_n, b], \end{cases} \quad (5.49)$$

$$g(x) = \begin{cases} f_1^2, & \text{if } x \in [a, p_1), \\ m_i^2 x^2 + 2m_i t_i x + t_i^2, & \text{if } x \in [p_i, p_{i+1}) \text{ and } i = 1, 2, \dots, n-1, \\ f_n^2, & \text{if } x \in [p_n, b], \end{cases} \quad (5.50)$$

with

$$m_i := \frac{f_{i+1} - f_i}{h}, \quad \text{and} \quad t_i := \frac{p_{i+1} f_i - p_i f_{i+1}}{h}. \quad (5.51)$$

For  $y_1, y_2 \in [p_i, p_{i+1}]$  and  $i = 1, 2, \dots, n-1$ , the integrals of  $f(x)$  and  $g(x)$  read

$$\int_{y_1}^{y_2} f(z) dz = \frac{m_i}{2} (y_2^2 - y_1^2) + t_i (y_2 - y_1), \quad (5.52)$$

$$\int_{y_1}^{y_2} g(z) dz = \frac{m_i^2}{3} (y_2^3 - y_1^3) + m_i t_i (y_2^2 - y_1^2) + t_i^2 (y_2 - y_1). \quad (5.53)$$

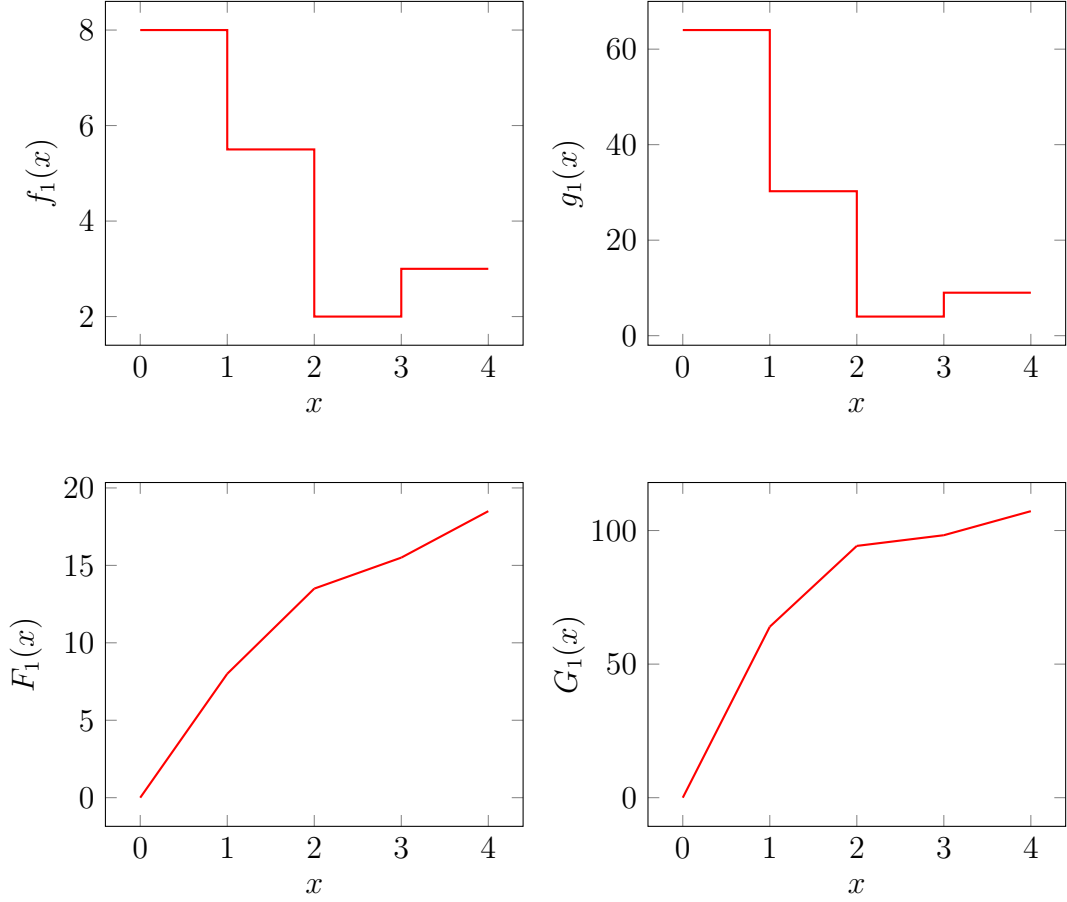


Figure 5.1: The functions  $f_1(x)$ ,  $g_1(x)$ ,  $F_1(x)$ , and  $G_1(x)$  as defined in (5.45), (5.46), (5.47), and (5.48).

Consequently, the functions  $F(x)$  and  $G(x)$  are given by

$$F(x) = \begin{cases} f_1(x - x_0), & \text{if } x \in [a, p_1), \\ \frac{f_1 h}{2} + \sum_{j=1}^{i-1} \int_{p_j}^{p_{j+1}} f(y) dy + \int_{p_i}^x f(y) dy, & \text{if } x \in [p_i, p_{i+1}), \\ \frac{f_1 h}{2} + \sum_{j=1}^{n-1} \int_{p_j}^{p_{j+1}} f(y) dy + f_n(h - b + x), & \text{if } x \in [p_n, b], \end{cases} \quad (5.54)$$

$$G(x) = \begin{cases} f_1^2(x - a), & \text{if } x \in [a, p_1), \\ \frac{f_1^2 h}{2} + \sum_{j=1}^{i-1} \int_{p_j}^{p_{j+1}} g(y) dy + \int_{p_i}^x g(y) dy, & \text{if } x \in [p_i, p_{i+1}), \\ \frac{f_1^2 h}{2} + \sum_{j=1}^{n-1} \int_{p_j}^{p_{j+1}} g(y) dy + f_n^2(\frac{h}{2} - b + x), & \text{if } x \in [p_n, b], \end{cases} \quad (5.55)$$

where  $i = 2, 3, \dots, n - 1$ .

Let us now take a brief look at the smoothness properties of the input signal and its related functions, too. Referring to (5.49) and (5.50), we observe that – by construction –  $f, g \in C^0([a, b])$ . As a consequence, we have  $F, G \in C^1([a, b])$ .

**Example Function.** Also in case of a piecewise linear input signal  $f$  we use the samples  $\mathbf{f}_1$  (see (5.43)) and their sampling positions  $\mathbf{p}_1$  (see (5.44)) to define an exemplary function on the domain  $[a, b]$  with  $a := 0$  and  $b := 4$ . In combination with (5.49), (5.50), (5.54), and (5.55) this results in

$$f_2(x) := \begin{cases} 8, & \text{if } 0 \leq x < \frac{1}{2}, \\ -\frac{5}{2}x + \frac{37}{4}, & \text{if } \frac{1}{2} \leq x < \frac{3}{2}, \\ -\frac{7}{2}x + \frac{43}{4}, & \text{if } \frac{3}{2} \leq x < \frac{5}{2}, \\ x - \frac{1}{2}, & \text{if } \frac{5}{2} \leq x < \frac{7}{2}, \\ 3, & \text{if } \frac{7}{2} \leq x \leq 4, \end{cases} \quad (5.56)$$

$$g_2(x) := \begin{cases} 64, & \text{if } 0 \leq x < \frac{1}{2}, \\ \frac{25}{4}x^2 - \frac{185}{4}x + \frac{1369}{16}, & \text{if } \frac{1}{2} \leq x < \frac{3}{2}, \\ \frac{49}{4}x^2 - \frac{301}{4}x + \frac{1849}{16}, & \text{if } \frac{3}{2} \leq x < \frac{5}{2}, \\ x^2 - x + \frac{1}{4}, & \text{if } \frac{5}{2} \leq x < \frac{7}{2}, \\ 9, & \text{if } \frac{7}{2} \leq x \leq 4, \end{cases} \quad (5.57)$$

$$F_2(x) := \begin{cases} 8x, & \text{if } 0 \leq x < \frac{1}{2}, \\ -\frac{5}{4}x^2 + \frac{37}{4}x - \frac{5}{16}, & \text{if } \frac{1}{2} \leq x < \frac{3}{2}, \\ -\frac{7}{4}x^2 + \frac{43}{4}x - \frac{23}{16}, & \text{if } \frac{3}{2} \leq x < \frac{5}{2}, \\ \frac{1}{2}x^2 - \frac{1}{2}x + \frac{101}{8}, & \text{if } \frac{5}{2} \leq x < \frac{7}{2}, \\ 3x + \frac{13}{2}, & \text{if } \frac{7}{2} \leq x \leq 4, \end{cases} \quad (5.58)$$

$$G_2(x) := \begin{cases} 64x, & \text{if } 0 \leq x < \frac{1}{2}, \\ -\frac{2}{15} \left( -\frac{5}{2}x + \frac{37}{4} \right)^3 + \frac{1504}{15}, & \text{if } \frac{1}{2} \leq x < \frac{3}{2}, \\ -\frac{2}{21} \left( -\frac{7}{2}x + \frac{43}{4} \right)^3 + \frac{1315}{14}, & \text{if } \frac{3}{2} \leq x < \frac{5}{2}, \\ \frac{1}{3} \left( x - \frac{1}{2} \right)^3 + \frac{181}{2}, & \text{if } \frac{5}{2} \leq x < \frac{7}{2}, \\ 9x + 68, & \text{if } \frac{7}{2} \leq x \leq 4. \end{cases} \quad (5.59)$$

See Figure 5.2 for the corresponding function plots.

### 5.3 The Dar–Bruckstein Method

Recently Dar and Bruckstein [DB19] have proposed an approach to solve the approximation problem for piecewise constant functions  $u_c(x)$  (as presented in Chapter 5.2.2) very efficiently.



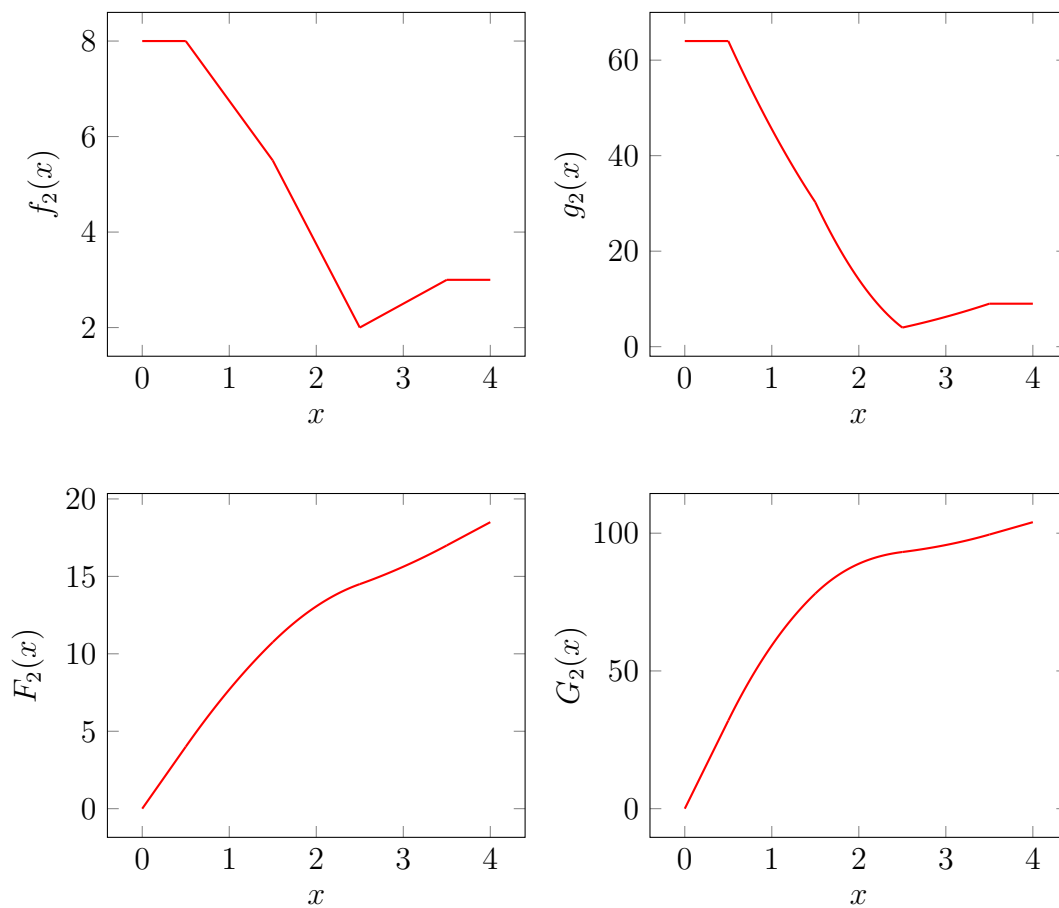


Figure 5.2: The functions  $f_2(x)$ ,  $g_2(x)$ ,  $F_2(x)$ , and  $G_2(x)$  as given in (5.56), (5.57), (5.58), and (5.59).

### 5.3.1 Compact Reformulation of the Dar–Bruckstein Method

To explain the underlying ideas and assumptions in a simple and transparent way, we reformulate its derivation. This reformulation is inspired by work of Belhachmi et al. [BBBW09, Section 6].

Denoting the squared error in the interval  $[x_i, x_{i+1}]$  by

$$e_i := \int_{x_i}^{x_{i+1}} (f(y) - u_i)^2 dy, \quad (5.60)$$

we can write the energy function (5.8) as

$$E_f(\mathbf{x}) = \frac{1}{b-a} \sum_{i=0}^{N-1} e_i. \quad (5.61)$$

Dar and Bruckstein assume that the input signal  $f$  is a continuously differentiable ( $C^1$ ) function and that  $N$  is large enough such that  $f$  can be approximated well by a linear function within each interval  $[x_i, x_{i+1}]$  for  $i = 0, 1, \dots, N-1$ . Thus, in

$(x_i, x_{i+1})$  we have

$$f'(x) = \frac{f(x_{i+1}) - f(x_i)}{x_{i+1} - x_i} =: f'_i, \quad (5.62)$$

$$f(x) = f(x_i) + (x - x_i) f'_i, \quad (5.63)$$

$$u_i = \frac{1}{2} (f(x_i) + f(x_{i+1})). \quad (5.64)$$

Using this in (5.60) and applying some simple calculations yields

$$e_i = \frac{1}{12} h_i^3 f_i'^2 \quad (5.65)$$

where  $h_i := x_{i+1} - x_i$  denotes the interval width.

As a heuristics for minimising the global energy (5.61), one may assume that  $\mathbf{x}$  is optimal if all local errors  $e_i$  are balanced. Using  $e_0 = e_1 = \dots = e_{N-1} = \text{const.}$  with (5.65) gives the following proportionalities:

$$f_i'^2 \sim \frac{1}{h_i^3} \implies \frac{1}{h_i} \sim \sqrt[3]{f_i'^2}. \quad (5.66)$$

Since  $1/h_i$  can be seen as a measure for the local density of the sampling points, one should choose the interval boundaries for optimal sampling proportional to  $\sqrt[3]{f_i'^2}$ . Consequently, Dar and Bruckstein select  $\mathbf{x}$  such that every segment  $[x_i, x_{i+1}]$  contains the same amount of the cube root of the squared signal derivative. More precisely:

$$\int_{x_i}^{x_{i+1}} \sqrt[3]{(f'(y))^2} dy = \frac{1}{N} \int_a^b \sqrt[3]{(f'(y))^2} dy =: T_{opt} \quad (5.67)$$

for  $i = 0, 1, \dots, N-1$ . The threshold  $T_{opt}$  is computed a priori. Thus, the analytical formula (5.67) allows to estimate the interval boundaries  $\mathbf{x}$  in a simple and efficient way.

### 5.3.2 Limitations of the Dar–Bruckstein Method

We have seen that the Dar–Bruckstein approach relies on three assumptions:  $C^1$ -smoothness, local linearity, and error balancing. Let us now analyse the impact of these assumptions on the optimality of the method in detail.

- Obviously the smoothness assumption on  $f$  is violated if the signal is nondifferentiable or noisy.
- To quantify inaccuracies caused by violations of the local linearity assumption, we derive a formula for  $e_i$  that does not use this assumption. We can rewrite (5.60) as

$$e_i = \int_{x_i}^{x_{i+1}} (f(y) - f(\xi_i))^2 dy, \quad (5.68)$$

where we have used the continuity of  $f$ , which guarantees that there exists a  $\xi_i \in [x_i, x_{i+1}]$  with  $f(\xi_i) = u_i$ . With the mean value theorem, Equation (5.68) becomes

$$e_i = \int_{x_i}^{x_{i+1}} (\xi_i - y)^2 (f'(\theta_i))^2 dy \quad (5.69)$$

$$= \frac{1}{3} ((x_{i+1} - \xi_i)^3 + (\xi_i - x_i)^3) (f'(\theta_i))^2 \quad (5.70)$$

for a suitable  $\theta_i \in [x_i, x_{i+1}]$ . Using  $h_i = x_{i+1} - x_i$  and defining  $\eta_i := (x_{i+1} - \xi_i)/h_i$  allows to rewrite (5.70) as

$$e_i = \frac{1}{3} (1 - 3\eta_i + 3\eta_i^2) h_i^3 (f'(\theta_i))^2. \quad (5.71)$$

Comparing the exact error (5.71) with the error (5.65) that exploits local linearity shows the following: Since  $\xi_i \in [x_i, x_{i+1}]$ , we know that  $\eta_i \in [0, 1]$ . However, only for  $\eta_i = \frac{1}{2}$ , we obtain  $\frac{1}{3} (1 - 3\eta_i + 3\eta_i^2) = \frac{1}{12}$ . In the worst case with  $\eta_i = 0$  or  $1$ , this factor becomes  $\frac{1}{3}$ . Moreover, since  $f \in C^1[a, b]$ , there exist constants  $m := \min_{[a,b]} f'$  and  $M := \max_{[a,b]} f'$ . Thus,  $(f'(\theta_i))^2$  can attain any value between  $m^2$  and  $M^2$ , which can differ substantially from  $f_i'^2$ . This shows that without local linearity, (5.65) can be violated severely. Moreover, (5.67) does no longer balance the errors then.

- While the principle of error balancing sounds plausible, one cannot prove that it is fulfilled for the globally optimal  $u$  which minimises the MSE. We provide a simple counterexample in Section 5.4.2 (*Approximation of  $f_1(x)$  using  $N = 2$* , Figure 5.3).

## 5.4 Direct Energy Optimisation

The preceding discussion shows that it can be desirable to renounce all three assumptions of the Dar–Bruckstein model. Interestingly, there is a surprisingly simple solution: We can rely directly on the discrete model  $E_f(\mathbf{x}, \mathbf{u})$  as given in (5.2), which is perfect from a modelling viewpoint. This ansatz also allows to estimate approximation functions of arbitrary type while the Dar–Bruckstein method restricts itself to piecewise constant  $u$ . However, we have to deal with a challenging optimisation problem (cf. Chapter 5.2).

### 5.4.1 Particle Swarm Optimisation (PSO)

Since we cannot expect to find an efficient algorithm with formal convergence guarantees to a global minimum, we use a nature-inspired metaheuristic that ends up in a good local minimum. Based on our tests, we recommend to minimise (5.2) by a Particle Swarm Optimisation (PSO) approach. This means that – on the one hand – we abandon formal convergence guarantees, while – on the other hand – we benefit from a versatile and flexible algorithm which does not impose

any additional restrictions on our problem. For this reason, we can use PSO in all our occurring scenarios. We refer to Section 2.2.5 for details on the algorithm itself.

### 5.4.2 First-Order Optimisation Methods

Nonetheless, we want to examine the usability of first-order optimisation methods for the minimisation of  $E_f(\mathbf{x}, \mathbf{u})$ , too. Without any doubt, these techniques can only represent a serious alternative in case of sufficiently simple and smooth energy landscapes for which a reasonable initialisation exists. This would allow to benefit from the convergence of first-order methods to a local energy minimum which is either equal or close to the global energy minimum.

Especially, the idea of backtracking line search (see Section 2.2.4) turns out to be an effective tool when dealing with unpleasant energies. Below, we employ two minimal examples to sketch the idea of how first-order methods allow to solve the approximation problem with piecewise constant functions  $u_c$ . In Chapter 5.5, we prove experimentally that these ideas carry over to real world scenarios.

#### Minimal Examples for Piecewise Constant Functions $u_c(x)$

We consider a minimal setup using the input functions  $f_1$  and  $f_2$  – as defined in (5.45) and (5.56) – for which we try to find the corresponding optimal approximation function  $u_c$ . The latter shall consist of  $N = 2$  segments with corresponding boundaries  $\mathbf{x} = (x_0, x_1, x_2)^T$ . According to (5.1) we have  $x_0 = a = 0$  and  $x_2 = b = 4$ . Consequently, our task reduces to the estimation of the remaining segment boundary  $x_1$  which minimises – referring to (5.9) – the energy

$$E_f(x_1) = \frac{G(4)}{4} - \frac{(8F(4)F(x_1) - 4F^2(4))x_1 - 16F^2(x_1)}{16x_1(-4 + x_1)}. \quad (5.72)$$

**Approximation of  $f_1(x)$  using  $N = 2$ .** Using (5.45)-(5.48) in combination with (5.72) leads to the energy

$$\begin{aligned} E_{f_1}(x_1) &:= \frac{429}{16} - \frac{(148F_1(x_1) - 1369)x_1 - 16F_1^2(x_1)}{16x_1(-4 + x_1)} & (5.73) \\ &= \frac{429}{16} - \begin{cases} \frac{160x_1 - 1369}{16x_1 - 64}, & \text{if } 0 \leq x_1 < 1, \\ \frac{330x_1^2 - 1439x_1 - 100}{16x_1(-4 + x_1)}, & \text{if } 1 \leq x_1 < 2, \\ \frac{232x_1^2 - 571x_1 - 1444}{16x_1(-4 + x_1)}, & \text{if } 2 \leq x_1 < 3, \\ \frac{300x_1 + 169}{16x_1}, & \text{if } 3 \leq x_1 \leq 4, \end{cases} & (5.74) \end{aligned}$$

which we visualise in Figure 5.3. As one can see, this energy has a unique global minimum at  $x_1 = 2$  and kinks at the jump positions of the input function  $f_1$ :  $x_1 = 1$ ,  $x_1 = 2$ , and  $x_1 = 3$ . Consequently, it is not differentiable there. We also illustrate  $f_1$  and the approximation  $u_c$  resulting in the lowest MSE (using  $x_1 = 2$ ) for  $N = 2$  in Figure 5.3. It is remarkable that the idea of error balancing

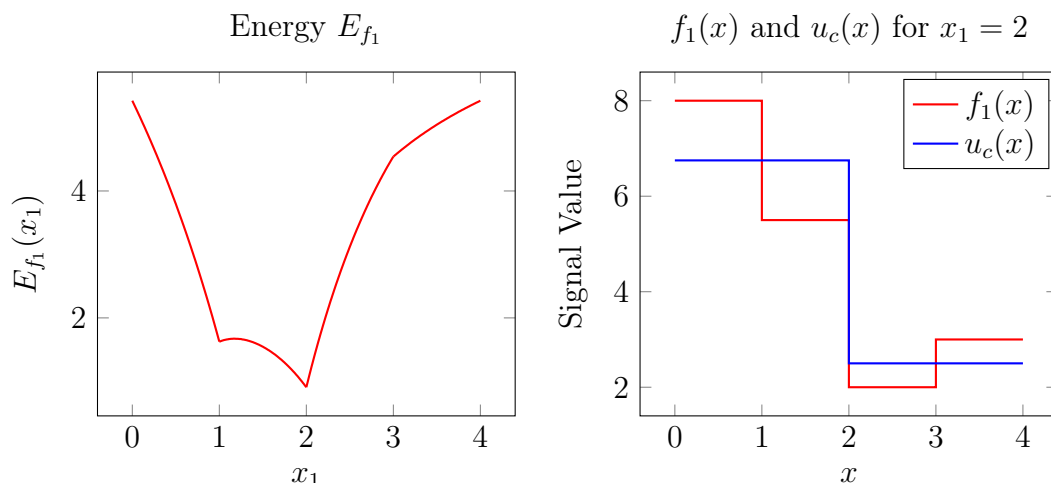


Figure 5.3: Energy  $E_{f_1}$  and the optimal approximation function  $u_c(x)$  for the piecewise constant  $f_1(x)$  in case of  $N = 2$ .

– which is used in the Dar–Bruckstein model – is not even appropriate for this simple example: The error in the left segment of  $u_c$  is clearly larger than in the right segment.

Now, let us consider the first- and second-order derivative of  $E_{f_1}$

$$E'_{f_1}(x_1) = \begin{cases} -\frac{729}{16(-4+x_1)^2}, & \text{if } 0 < x_1 < 1, \\ \frac{-119x_1^2 - 200x_1 + 400}{16x_1^2(-4+x_1)^2}, & \text{if } 1 < x_1 < 2, \\ \frac{357x_1^2 - 2888x_1 + 5776}{16x_1^2(-4+x_1)^2}, & \text{if } 2 < x_1 < 3, \\ \frac{169}{16x_1^2}, & \text{if } 3 < x_1 < 4, \end{cases} \quad (5.75)$$

$$E''_{f_1}(x_1) = \begin{cases} \frac{729}{8(-4+x_1)^3}, & \text{if } 0 < x_1 < 1, \\ \frac{119x_1^3 + 300x_1^2 - 1200x_1 + 1600}{8x_1^3(-4+x_1)^3}, & \text{if } 1 < x_1 < 2, \\ \frac{-357x_1^3 + 4332x_1^2 - 17328x_1 + 23104}{8x_1^3(-4+x_1)^3}, & \text{if } 2 < x_1 < 3, \\ -\frac{169}{8x_1^3}, & \text{if } 3 < x_1 < 4, \end{cases} \quad (5.76)$$

in order to better understand how to employ first-order optimisation methods for the minimisation of (5.74). We sketch both functions in Figure 5.4 and observe, that

$$\max_{x_1 \in X} |E''_{f_1}(x_1)| < \frac{365}{64}, \quad \text{where } X = \{(0, 1) \cup (1, 2) \cup (2, 3) \cup (3, 4)\}. \quad (5.77)$$

With the help of this information, we investigate the feasibility of applying the gradient descent method (see Chapter 2.2.1) without and with backtracking line search (see Chapter 2.2.4) to our approximation problem. Subsequently, we use the abbreviations GD (gradient descent) and GD–BTLS (gradient descent with backtracking line search) to refer to both algorithms.

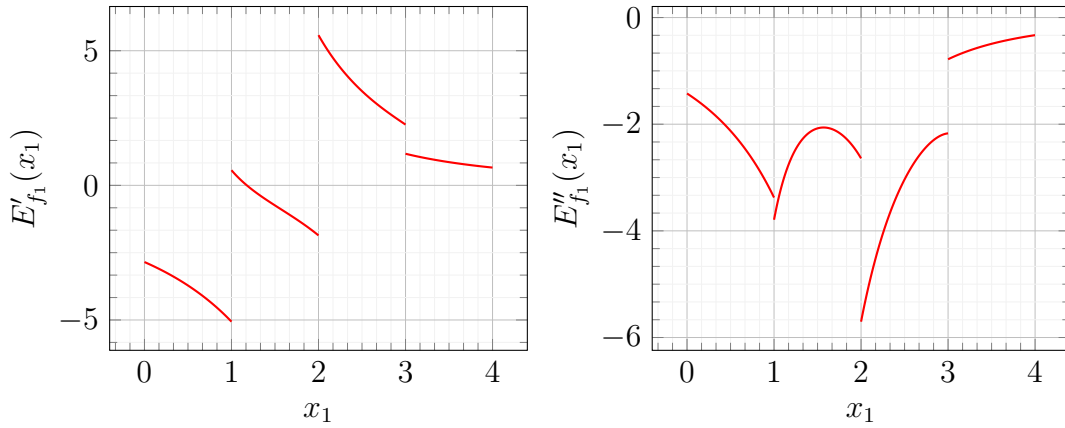


Figure 5.4: First- and second-order derivative of  $E_{f_1}$ , where  $x_1 \in X$ .

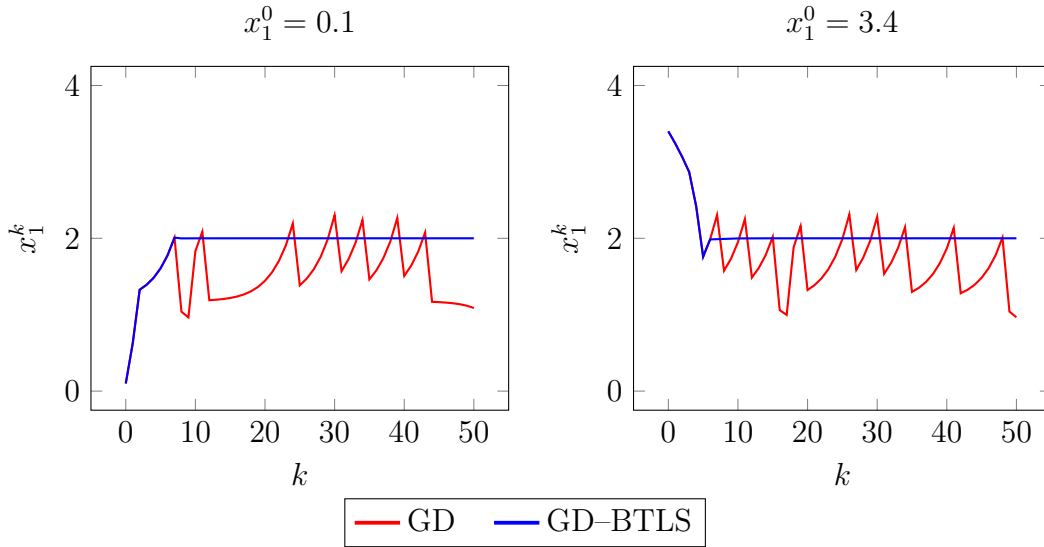


Figure 5.5: Different behaviour of the gradient descent method without and with backtracking line search when minimising  $E_{f_1}$ .

For a moment we ignore the kinks of the energy function and estimate the optimal time step size for GD based on (5.77) and according to (2.19) as

$$\alpha = \frac{64}{365}. \quad (5.78)$$

Below, we use this step size for GD and as initial step size for GD-BTLS. Furthermore, we choose two arbitrary initial values:  $x_1^0 = 0.1$  and  $x_1^0 = 3.4$ . For both, we apply GD and GD-BTLS with the aim to minimise (5.74) and to detect its global minimiser  $x_1^* = 2$ . Figure 5.5 shows the estimated boundary position  $x_1^k$  in dependence of the number of iterations  $k$ .

As one can see, GD fails in both cases: The estimate  $x_1^k$  oscillates around the optimal value  $x_1^* = 2$  for  $k \geq 7$  (in case of  $x_1^0 = 0.1$ ) and  $k \geq 5$  (in case of  $x_1^0 = 3.4$ ). This was to be expected due to the fact that the energy function is – because of its kinks – not differentiable. This violates the assumptions of

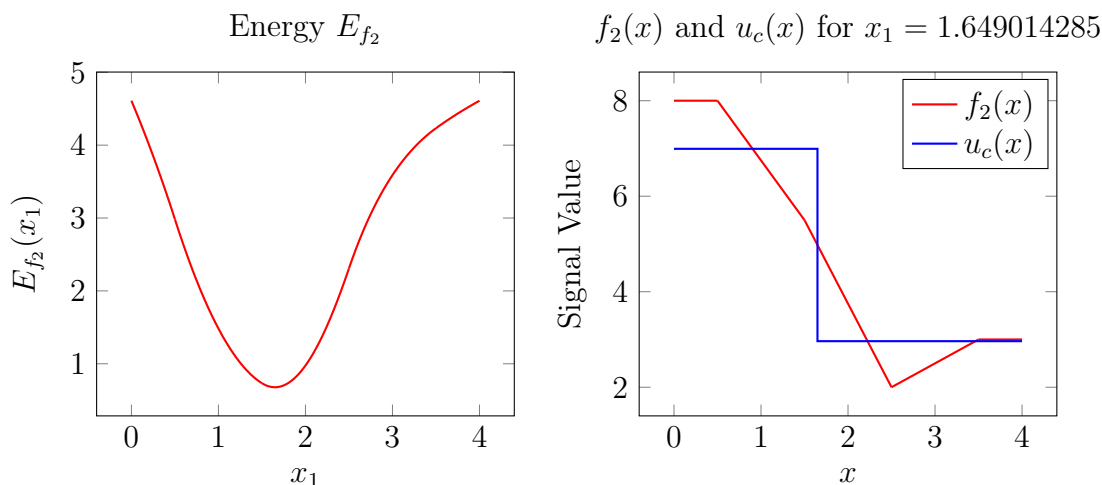


Figure 5.6: Energy  $E_{f_2}$  and the optimal approximation function  $u_c(x)$  for the piecewise linear  $f_2(x)$  in case of  $N = 2$ .

GD stated in Chapter 2.2.1 such that classical convergence results do not apply anymore.

In contrast, the GD–BTLS algorithm performs well. It converges – in both cases – to the desired boundary value  $x_1^* = 2$ . More precisely, we get an error of  $|x_1^k - x_1^*| < 10^{-4}$  for  $k \geq 15$  (in case of  $x_1^0 = 0.1$ ) and  $k \geq 16$  (in case of  $x_1^0 = 3.4$ ).

**Approximation of  $f_2(x)$  using  $N = 2$ .** For the piecewise linear input function  $f_2$  we use (5.56)-(5.59) in combination with (5.72) to derive the energy

$$\begin{aligned}
 E_{f_2}(x_1) &= 26 - \frac{(148F(x_1) - 1369)x_1 - 16(F(x_1))^2}{16x_1(-4 + x_1)} \quad (5.79) \\
 &= 26 - \begin{cases} \frac{160x_1 - 1369}{16x_1 - 64}, & \text{if } 0 \leq x_1 < \frac{1}{2}, \\ \frac{-400x_1^4 + 2960x_1^3 - 200x_1^2 - 21164x_1 - 25}{256x_1(-4 + x_1)}, & \text{if } \frac{1}{2} \leq x_1 < \frac{3}{2}, \\ \frac{-784x_1^4 + 5488x_1^3 - 5416x_1^2 - 17396x_1 - 529}{256x_1(-4 + x_1)}, & \text{if } \frac{3}{2} \leq x_1 < \frac{5}{2}, \\ \frac{-16x_1^4 + 328x_1^3 - 1120x_1^2 + 2806x_1 - 10201}{64x_1(-4 + x_1)}, & \text{if } \frac{5}{2} \leq x_1 < \frac{7}{2}, \\ \frac{300x_1 + 169}{16x_1}, & \text{if } \frac{7}{2} \leq x_1 \leq 4. \end{cases} \quad (5.80)
 \end{aligned}$$

A plot of the energy is given in Figure 5.6. The energy function appears to be a smooth function and we have  $E_{f_2} \in C_{L_{f_2}}^{1,1}((0, 4))$  with Lipschitz constant

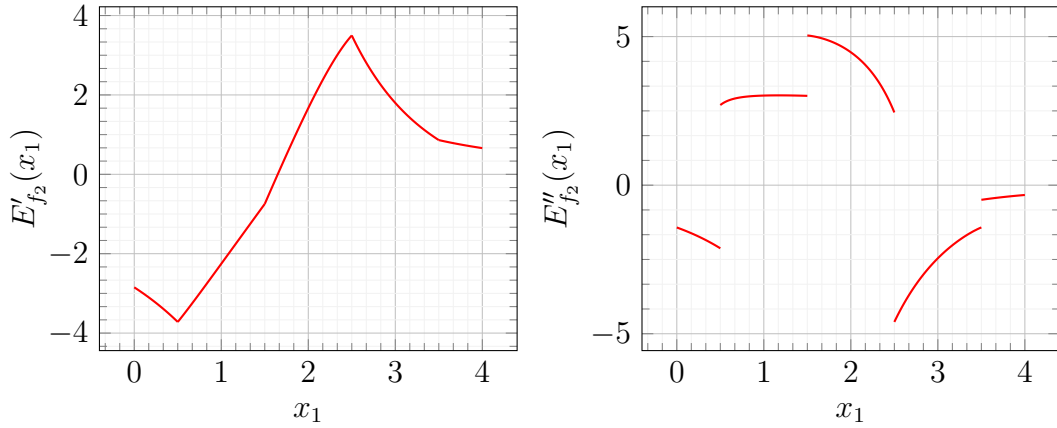


Figure 5.7: Optimal approximation of piecewise linear  $f_2(x)$  using piecewise constant  $u_c(x)$  for  $N = 2$ .

$L_{f_2} = \frac{68023}{13500}$  which we derive below. The energy's first derivative is given by

$$E'_{f_2}(x_1) = \begin{cases} -\frac{729}{16(-4+x_1)^2}, & \text{if } 0 < x_1 \leq \frac{1}{2}, \\ \frac{400x_1^5 - 3880x_1^4 + 11840x_1^3 - 10982x_1^2 - 25x_1 + 50}{128x_1^2(-4+x_1)^2}, & \text{if } \frac{1}{2} < x_1 \leq \frac{3}{2}, \\ \frac{784x_1^5 - 7448x_1^4 + 21952x_1^3 - 19530x_1^2 - 529x_1 + 1058}{128x_1^2(-4+x_1)^2}, & \text{if } \frac{3}{2} < x_1 \leq \frac{5}{2}, \\ \frac{16x_1^5 - 260x_1^4 + 1312x_1^3 - 837x_1^2 - 10201x_1 + 20402}{32x_1^2(-4+x_1)^2}, & \text{if } \frac{5}{2} < x_1 \leq \frac{7}{2}, \\ \frac{169}{16x_1^2}, & \text{if } \frac{7}{2} < x_1 < 4, \end{cases} \quad (5.81)$$

and plotted in Figure 5.7. As one can see, the energy  $E_{f_2}$  has a unique global minimum of approximately 0.67793632 at the only root of  $E'_{f_2}$  which is  $x_1 \approx 1.649014285$ . The corresponding optimal approximation  $u_c$  of  $f_2$  is shown on the right hand side of Figure 5.6. Without considering the kinks of  $f_2$  the second derivative of  $E_{f_2}$  reads

$$E''_{f_2}(x_1) = \begin{cases} \frac{729}{8(-4+x_1)^3}, & \text{if } 0 < x_1 < \frac{1}{2}, \\ \frac{400x_1^6 - 4800x_1^5 + 19200x_1^4 - 25396x_1^3 + 75x_1^2 - 300x_1 + 400}{128x_1^3(-4+x_1)^3}, & \text{if } \frac{1}{2} < x_1 < \frac{3}{2}, \\ \frac{784x_1^6 - 9408x_1^5 + 37632x_1^4 - 48748x_1^3 + 1587x_1^2 - 6348x_1 + 8464}{128x_1^3(-4+x_1)^3}, & \text{if } \frac{3}{2} < x_1 < \frac{5}{2}, \\ \frac{16x_1^6 - 192x_1^5 + 768x_1^4 - 3574x_1^3 + 30603x_1^2 - 122412x_1 + 163216}{32x_1^3(-4+x_1)^3}, & \text{if } \frac{5}{2} < x_1 < \frac{7}{2}, \\ -\frac{169}{8x_1^3}, & \text{if } \frac{7}{2} < x_1 < 4, \end{cases} \quad (5.82)$$

and is shown in Figure 5.7. We observe that

$$\max_{x_1 \in X} |E''_{f_2}(x_1)| < \frac{68023}{13500}, \quad \text{with } X = (0, 4) \setminus \left\{ \frac{1}{2}, \frac{3}{2}, \frac{5}{2}, \frac{7}{2} \right\}, \quad (5.83)$$

which corresponds to the previously stated Lipschitz estimate  $L_{f_2}$ . In contrast to our first example considering  $E_{f_1}$ , our current energy function  $E_{f_2}$  fulfils all



requirements of the gradient descent method (see Chapter 2.2.1). This allows us to apply the gradient descent algorithm using the optimal time step size

$$\alpha = \frac{1}{L_{f_2}} = \frac{13500}{68023} \quad (5.84)$$

and to benefit from the method's local convergence guarantees. For our specific scenario we validate the convergence to the global energy minimum  $x_1^* = 1.649014285$  using the same initial values as in our previous example:  $x_1^0 = 0.1$  and  $x_1^0 = 3.4$ . This results in an error  $|x_1^k - x_1^*| < 10^{-4}$  for  $k \geq 5$  (in case of  $x_1^0 = 0.1$ ) and  $k \geq 7$  iterations (in case of  $x_1^0 = 3.4$ ).

## 5.5 Experiments

Let us now evaluate the approximation quality of the previously studied approaches. We examine the approximation problem for piecewise constant functions  $u_c$  and the approximation problem for piecewise linear functions  $u_\ell$  separately. Throughout our experiments we assume a number of segments  $N \in [2, 100]$ . For the scenario of piecewise constant approximation functions  $u_c$  we compare the Dar–Bruckstein method, uniform (re-)sampling, and our direct energy optimisation strategy (using the numerical optimisation techniques introduced in Chapter 2.2). As a naive approach, uniform (re-)sampling provides a lower quality threshold which can be reached with minimal effort. It is used as a reference for cost-benefit analysis and should be excelled by all other approaches.

In case of piecewise linear approximation functions  $u_\ell$ , we assess the performance of our direct optimisation approach using the SPSO algorithm (see Chapter 2.2.5). In all experiments we employ own implementations of the algorithms, i.e.:

- We have implemented the Dar–Bruckstein model as is proposed in [DB19, Subsection 2.1] using (5.67).
- We have implemented the Standard Particle Swarm Optimisation 2011 algorithm which we introduce in Chapter 2.2.5. We adhere to [ZCR13].
- As first-order optimisation techniques, we have implemented the gradient descent method (see Chapter 2.2, Algorithm 1), the heavy ball method (see Chapter 2.2, Algorithm 2), and Adaptive FSI schemes (see Chapter 2.2, Algorithm 3).
- Our implementation of the heavy ball method involves an optional adaptive time step size strategy. The latter makes use of the descent property of the gradient descent method for differentiable functions [Nes04, (1.2.12)] and ensures that the energy values decrease at least as good as specified in this inequality. In order to adapt the time step size we scale the Lipschitz estimate by a factor of 1.05 every time the descent condition is not fulfilled and update  $\alpha$  based on (2.22).
- Additionally, we have implemented backtracking line search (see Algorithm 4 in Chapter 2.2) as an automatic time step size selection strategy.

If not stated otherwise, we run the SPSO algorithm 50 times for each specified value of  $N$  and report the minimum of our MSE computations (also referred to as  $\min_{\text{MSE}}$ ). This is done because the SPSO algorithm involves randomisation and the quality of multiple program runs with identical parameters may differ somewhat. Additionally, we use a maximum number of

- 10000 iterations ( $k \leq 10000$ ) in case of piecewise constant approximation functions, and
- 1000 iterations ( $k \leq 1000$ ) in case of piecewise linear approximation functions,

as a stopping criterion for SPSO.

Whenever we use the symbol  $\odot$  to display results, this refers to an experiment with random initialisation of the segment boundaries  $\mathbf{x}$  (following a uniform distribution) which we terminate after 60 minutes. This is done to compare the performance of SPSO to a brute force approach. To ensure comparability of the results we run all of these experiments on a single core of the same machine: Intel<sup>®</sup> Core<sup>™</sup> i7-6700 CPU (3.40GHz), 32 GB RAM, Debian 9.11.

### 5.5.1 Piecewise Constant Approximation Functions $u_c(x)$

In our experiments for piecewise constant approximation functions  $u_c$  we consider three different types of input functions  $f$ :

1. smooth input signals  $f$ ,
2. piecewise constant input signals  $f$ ,
3. piecewise linear input signals  $f$ .

#### Smooth Input Signals

In our first experiment we want to solve the approximation problem for the smooth chirp signal

$$f_3(x) := 255 \cos(2\pi x(1 + 5x)), \quad \text{for } x \in [0, 1], \quad (5.85)$$

which is illustrated in Figure 5.8. It was also studied in [DB19]. It constitutes a prototype for a smooth signal, which is nevertheless challenging in its high frequent part. We try to find its optimal piecewise constant approximation function  $u_c$  in dependency of its number of segments  $N$  as discussed in Chapter 5.2.2. Subsequently, we compare the following techniques with regard to their efficacy and effectiveness to minimise the MSE (5.8):

- Uniform (Re-)Sampling (US),
- the Dar–Bruckstein method (DB),
- our direct energy optimisation approach using

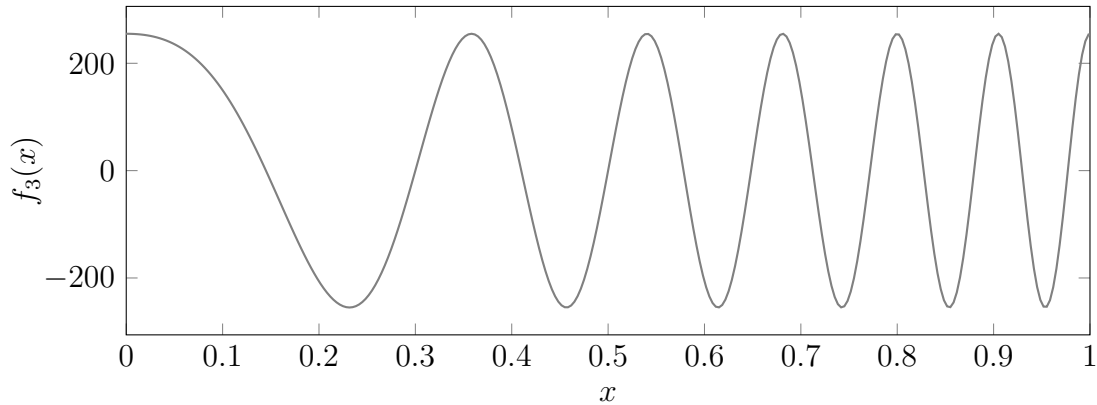


Figure 5.8: Smooth chirp signal  $f_3(x)$  as defined in (5.85).

- Standard Particle Optimisation 2011 (SPSO) with a swarm size of  $n = 1000$ , as well as
- Adaptive FSI schemes (AFSI) where the results from US, DB, and SPSO (using  $\min_{\text{MSE}}$ ) serve as initialisation (in the following denoted by AFSI [US], AFSI [DB], and AFSI [SPSO]).

Let us now discuss the results of the corresponding experiments which we present in Figure 5.9 and Table 5.1.

In general, SPSO – amongst the three elementary solution strategies US, DB, and SPSO – performs best followed by DB and US performing worst. Only for  $N \in \{95, 100\}$ , DB performs better than SPSO and for  $N \in \{2, 5, 6, 7, 8, 9, 10, 14\}$  even US gives better results than DB. This supports our discussion of the Dar–Bruckstein method in Chapter 5.3.2: For low values of  $N$  the local linearity assumption is violated. This leads to significant inaccuracies and DB performs worse than US. The effect caused by this violation vanishes with increasing  $N$ . When a sufficiently high number of segments  $N$  is reached the original signal resembles a linear function in each section and DB can beat SPSO.

Next, we observe that SPSO is the only (elementary) method which is able to detect a local energy minimum at all. We conclude this from the fact that AFSI is guaranteed to converge to a local minimum for  $f_3$  and that SPSO leads to the same MSE as AFSI [SPSO] for  $N = 5$  and  $N = 10$ . Furthermore, the results for SPSO and AFSI [SPSO] differ by less than one for  $N \in \{20, 30, 40\}$  which means that the approximation provided by SPSO is at least close to a local optimum.

Overall, AFSI can improve US, DB, and SPSO for all investigated values of  $N$ . While we get no or only small improvements with AFSI [SPSO], the MSE can be reduced significantly when initialising AFSI with the results from US and DB. For low values of  $N$ , AFSI [SPSO] is the method of choice. Starting from  $N \geq 24$  AFSI [SPSO] and AFSI [DB] give results of similar quality and for  $N \geq 37$  AFSI [DB] is superior to AFSI [SPSO]. AFSI [US] is always inferior to both other approaches.

Additionally, the improvements gained with AFSI prove that we have to deal with a complex energy landscape: AFSI never converges to the same local minimum no matter which one of the three different initialisations we use.

N	US	AFSI [US]	DB	AFSI [DB]	SPSO				AFSI [SPSO]
					$\mu_{\text{MSE}}$	$\sigma_{\text{MSE}}$	min <sub>MSE</sub>	max <sub>MSE</sub>	
5	31008.99	22831.95	32166.87	31614.25	19055.49	0.00	19055.49	19055.49	19055.49
10	18212.44	14356.27	23655.93	14474.85	12014.77	459.47	11131.75	12431.60	11131.75
20	12062.19	5157.29	5661.00	3673.77	4556.37	632.89	3403.23	5998.27	3403.16
30	4760.98	2188.78	2590.86	1823.67	2037.90	67.90	1906.95	2285.76	1906.73
40	2796.54	1431.34	1477.48	1132.80	1218.49	37.17	1177.55	1357.26	1177.17
50	1827.80	946.87	975.92	768.76	854.12	20.76	823.13	907.48	821.86
60	1284.03	694.65	686.30	550.15	624.44	13.37	601.65	665.50	598.79
70	949.98	519.00	510.96	421.75	479.90	9.26	460.10	513.00	451.85
80	730.64	418.87	377.20	327.71	383.08	6.95	367.62	400.07	361.57
90	579.10	334.26	307.64	265.66	311.04	5.11	301.63	328.83	294.40
100	470.12	280.56	247.84	217.79	258.05	4.25	249.73	269.60	241.98

Table 5.1: Mean squared error of  $u_c$  w.r.t.  $f_3$ .

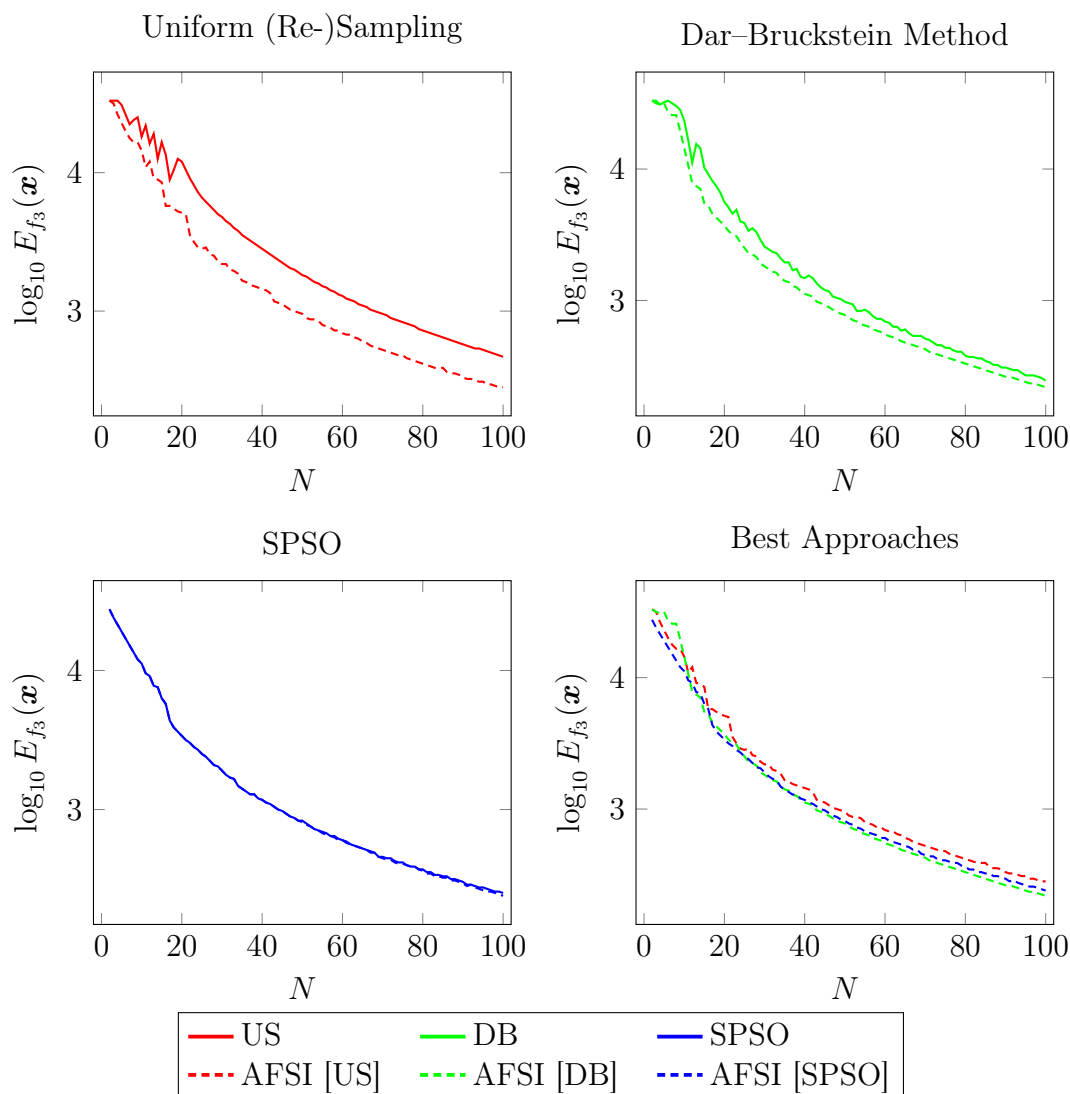


Figure 5.9: Approximation quality of  $u_c$  for the smooth chirp signal  $f_3$ .

In Figure 5.10 we see the best approximation functions for  $N = 11$  using AFSI [US], AFSI [DB], and AFSI [SPSO]. At first glance the three solutions seem to be of similar visual quality and it is hard to identify the best. In order of decreasing MSE we have: AFSI [US] (MSE: 11078.56), AFSI [DB] (MSE: 10733.56), AFSI [SPSO] (MSE: 9655.13). A more detailed view at Figure 5.10 allows to understand why AFSI [SPSO] is superior to the other two methods: The approximation error concentrates on high signal frequencies. Using AFSI [SPSO] we have small deviations from the input signal for low and average frequencies while the approximation  $u$  cannot reproduce high frequencies of  $f$  (e.g. for  $x \in [0.8, 1]$ ). For limited  $N$  this sounds like a reasonable strategy to optimise the corresponding MSE.

On the other hand, AFSI [US] and AFSI [DB] don't follow this principle and deviate from the original signal already for lower frequencies:  $x \in [0.7, 0.85]$  for AFSI [US];  $x \in [0.65, 0.75]$  for AFSI [DB]. Consequently, this results in a higher MSE.

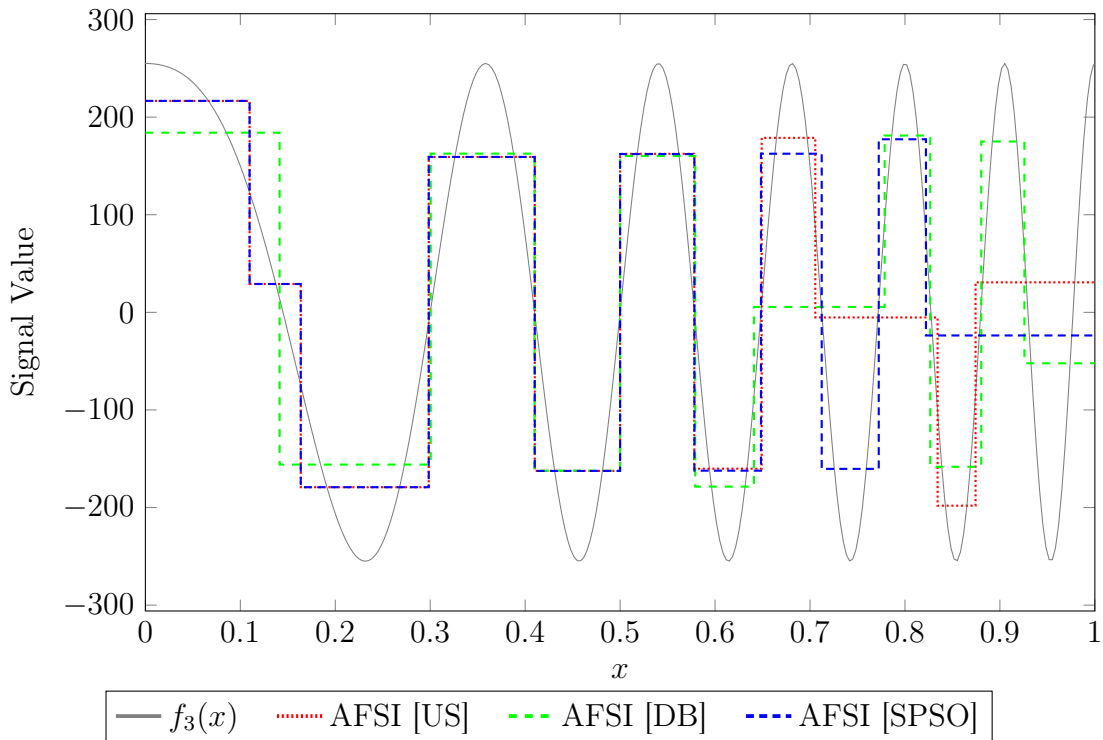


Figure 5.10: Best piecewise constant approximations of  $f_3$  for  $N = 11$ .

### Piecewise Constant Input Signal

In our next experiments we consider two piecewise constant input signals. The first one is based on the uniformly sampled version of (5.85) using  $n = 100$  samples. In the following we refer to this signal as the piecewise constant chirp signal. The second signal represents line 51 of the 8-bit test image *trui* (cf. Figure 5.11). We assume that within each of its 256 pixels the function values are constant. Subsequently, we refer to the latter as the *trui* 51 signal. Both signals are illustrated in Figure 5.12. Again, we try to find the optimal piecewise constant approximation function  $u_c$  for a given number of segments  $N$ . Since the input function  $f$  is a discrete and non-differentiable function we employ the



Figure 5.11: Test image *trui*,  $256 \times 256$  pixels.

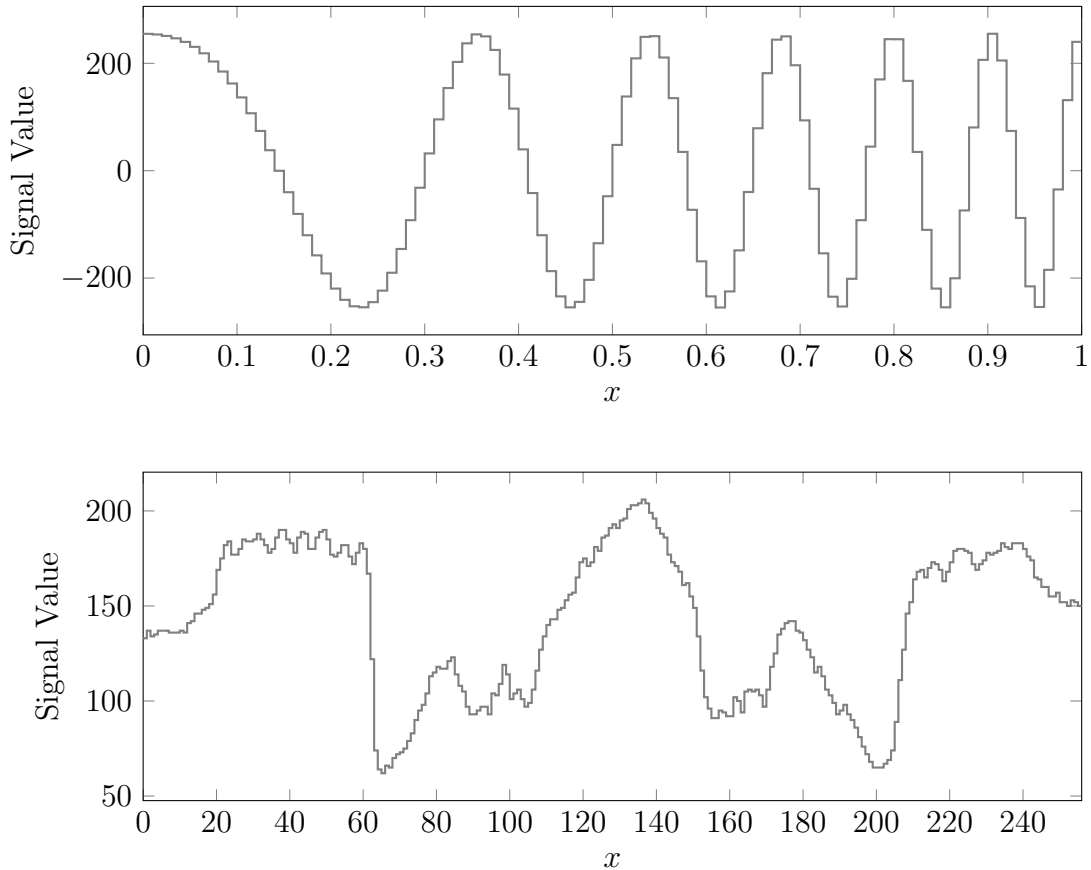


Figure 5.12: Uniformly sampled chirp signal with  $n = 100$  samples and the true signal with  $n = 256$  samples. Both interpreted as a piecewise constant function.

following techniques and compare the quality of the resulting approximations:

- Uniform (Re-)Sampling (US),
- the Dar–Bruckstein method (DB) using central differences and mirrored boundaries to estimate the signal derivative  $f'$  and the threshold (5.67),
- our direct energy optimisation approach using
  - Standard Particle Optimisation 2011 (SPSO) with a swarm size of  $n = 1000$ ,
  - the gradient descent method with backtracking line search (GD) using the results from US and DB as initialisation (subsequently referred to as GD [US] and GD [DB]),
  - the heavy ball method with adaptive time step size (HB) also using the results from US and DB as initialisation (referred to as HB [US] and HB [DB]), and
  - SPSO, GD, and HB based on a random initialisation in the previously mentioned 60 minute brute force experiment denoted by SPSO 🌀, GD 🌀, and HB 🌀.

N	US	GD [US]	HB [US]	DB	GD [DB]	HB [DB]	SPSO			
							$\mu_{\text{MSE}}$	$\sigma_{\text{MSE}}$	$\text{min}_{\text{MSE}}$	$\text{max}_{\text{MSE}}$
5	31005.72	22849.46	22804.62	32573.10	31883.23	31838.19	18997.25	0.00	18997.25	18997.25
10	18133.28	14607.57	18133.28	21956.83	14205.83	13872.63	11768.84	1211.69	10301.46	16273.41
20	11887.54	6000.42	11887.54	7102.99	4451.00	3780.55	6166.77	1393.68	3148.08	9080.05
30	5101.76	3620.53	2532.83	3726.56	2615.70	2172.81	2994.74	1233.68	1659.25	5417.21
40	3251.78	3251.78	3251.78	2484.97	2011.63	1744.69	1999.79	1002.70	1053.20	4949.92
50	1390.41	1390.41	1390.41	1652.36	1444.29	976.55	1072.10	398.42	569.29	3184.06
60	1768.76	1115.15	1768.76	1549.78	824.38	540.61	677.57	374.52	359.58	2900.50
70	1382.97	1180.53	1382.97	1277.81	692.37	453.00	473.85	245.41	202.86	1298.63
80	1292.10	1292.10	1292.10	1043.04	493.71	352.77	288.89	138.21	151.29	766.58
90	1018.28	477.85	1018.28	910.60	509.37	348.82	210.09	162.04	95.56	960.48
100	0.00	0.00	0.00	819.19	346.46	216.89	144.59	82.57	72.36	617.80

Table 5.2: Mean squared error of  $u_c$  w.r.t. the piecewise constant chirp signal.



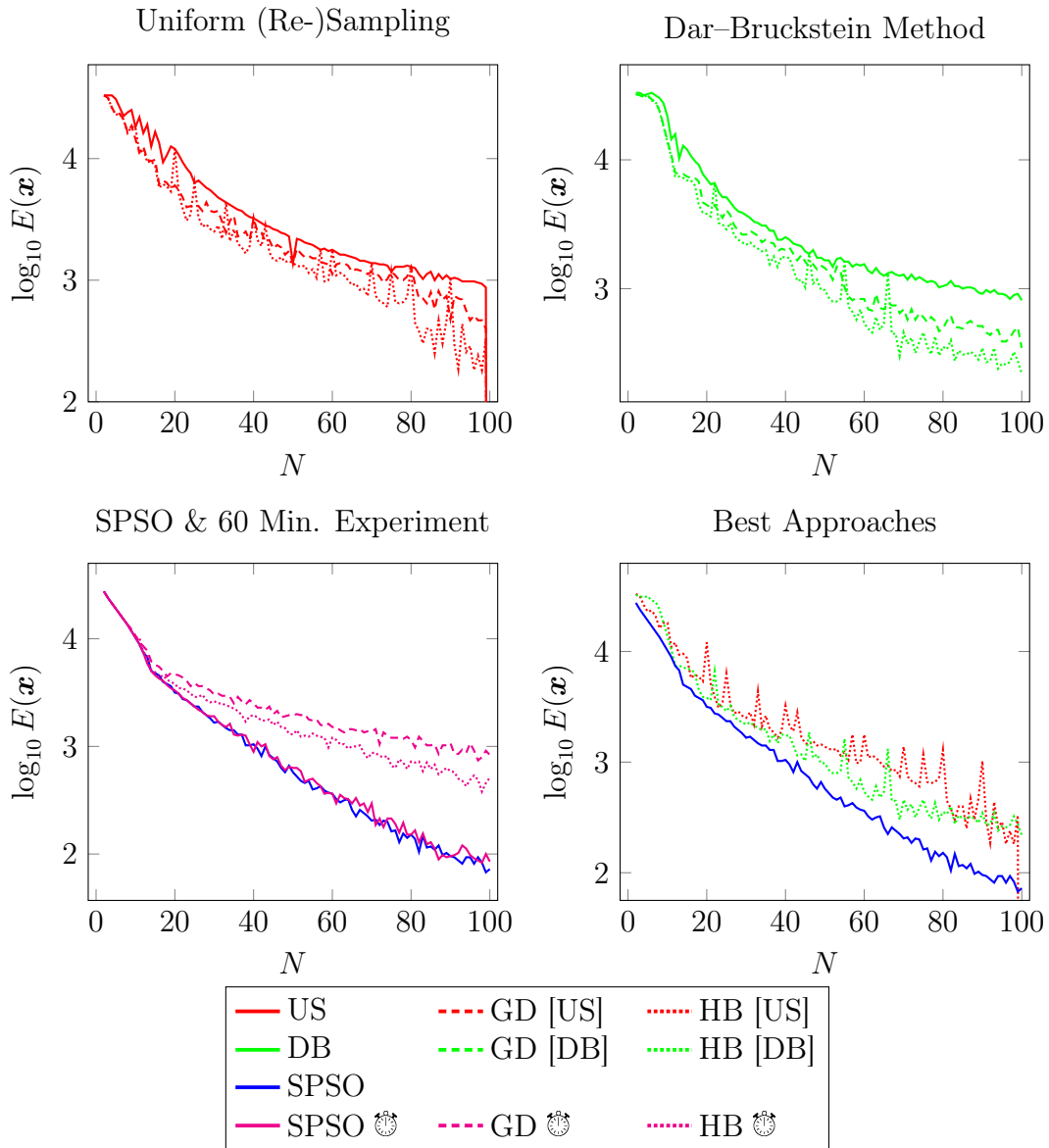


Figure 5.13: Approximation quality of  $u_c$  for the piecewise constant chirp signal.

**Piecewise Constant Chirp Signal.** First, we take a look at the results for the piecewise constant chirp signal which we provide in Table 5.2 and Table 5.3. A visualisation of the corresponding approximation quality is given in Figure 5.13. Like for the smooth input signal the results for SPSO are better than for DB and US. Only for small values of  $N$ , DB performs worse than US. Again, we trace this back to the violation of the linearity assumption of the Dar-Bruckstein method. Furthermore, we have a special case for  $N = 100$ . In this setting US leads to the global optimal solution since our input signal consists of 100 equally distributed samples. Note, that any of the methods which are not based on US comes close to the global optimum in this specific case.

Considering all applied methods, SPSO gives the best results. Especially with growing  $N$  the difference to the other approaches becomes quite large. For  $N = 80$  the MSE of the second best technique HB [DB] is e.g. 2.3 times higher. This

N	SPSO 🐼	GD 🐼	HB 🐼	N	SPSO 🐼	GD 🐼	HB 🐼
5	18997.25	18997.79	18997.25	60	366.62	1509.40	1206.36
10	10301.46	10632.16	10337.38	70	292.39	1372.26	863.67
20	3299.23	4659.68	3769.24	80	151.75	1152.21	719.53
30	1919.83	3024.63	2626.71	90	95.33	1061.00	594.46
40	887.85	2515.75	1939.97	100	85.69	824.72	508.56
50	626.41	1937.49	1294.22				

Table 5.3: Mean squared error of  $u_c$  w.r.t. the piecewise constant chirp signal.

is also in contrast to the smooth input signal experiments where the differences among all approaches are much smaller.

Overall, the best two-step approach appears to be HB which – in most cases – beats GD for both initialisations. Besides, it turns out that DB represents a better initialisation for HB than US. This can e.g. be seen for  $N = 50$  where the MSE for HB [US] is 1.4 times higher than for HB [DB].

From our brute force experiment lasting 60 minutes we learn that also in this case SPSO is superior to the other two techniques (see Table 5.3). For  $N \geq 20$  SPSO 🐼 beats GD 🐼 and HB 🐼 significantly: e.g. for  $N = 70$  the MSE of the second best method HB 🐼 is almost 3 times higher than for SPSO 🐼.

If we take a deeper look at the best approximations  $u_c$  of the piecewise constant chirp signal we observe the same behaviour as for the input signal  $f_3$ . In Figure 5.14 we present the outcome for  $N = 11$ . Again, the SPSO approach ensures a good signal approximation for low and average signal frequencies ( $x \in [0, 0.85]$ ) and shifts errors to the high frequencies ( $x > 0.85$ ). The quality of both other methods already suffers at lower frequencies ( $x \in [0.7, 0.85]$ ) which leads to a higher MSE. Consequently, SPSO gives the best results (MSE: 8914.82), followed by HB [DB] (MSE: 10488.75) and HB [US] (MSE: 11060.02).

**Piecewise Constant trui 51 Signal.** The next scenario which we discuss is the piecewise constant approximation of the piecewise constant trui 51 signal. We state the MSE for the applied methods in Table 5.4 and provide the corresponding visualisation in Figure 5.15. Please note that we do not list and plot the results for the heavy ball method (HB) in this case. The implemented adaptive time step size selection strategy turned out to be inappropriate such that HB could not improve any of the given initialisations. Consequently, the usage of HB was of no benefit such that we skip a discussion of its results here. Apart from that, we find very similar behaviour as for the approximation of the piecewise constant chirp signal. Overall, we get the following ranking of the applied techniques (in order of increasing MSE): SPSO, GD [DB], DB, GD [US], US. Only for  $N \leq 20$ , GD [US] gives better results than DB. In the same range of  $N$ , GD [US] and GD [DB] lead to almost identical MSE values. Furthermore, we observe that the application of a first-order optimisation technique can improve the results of both, US and DB, to some extent. However, we benefit only little in case of GD [DB] while GD [US] leads to remarkable improvements over US. Again, the 60 minute experiment shows that SPSO 🐼 is superior to other techniques like –



N	US	GD [US]	DB	GD [DB]	SPSO				SPSO 	GD 
					$\mu_{\text{MSE}}$	$\sigma_{\text{MSE}}$	min <sub>MSE</sub>	max <sub>MSE</sub>		
5	725.41	349.46	697.78	356.26	347.36	0.00	347.36	347.36	347.36	347.40
10	590.66	196.38	445.53	218.13	126.37	3.51	123.47	130.96	123.47	150.29
20	197.74	112.72	120.01	101.13	47.83	3.75	43.05	60.15	44.88	84.09
30	162.87	75.89	58.74	53.74	26.28	1.28	24.03	29.48	24.69	62.84
40	96.53	56.64	38.77	36.14	17.39	0.79	15.15	18.83	16.95	48.21
50	66.91	49.00	25.32	24.32	12.63	0.60	11.41	14.28	12.56	41.35
60	63.00	38.06	23.31	19.04	10.23	0.42	8.88	11.20	10.07	36.79
70	36.85	33.82	16.82	15.38	8.34	0.40	7.54	9.28	7.83	30.36
80	40.37	22.50	15.37	13.27	7.01	0.43	6.09	8.29	6.93	26.65
90	27.37	22.89	13.86	12.95	5.87	0.31	5.21	6.67	5.37	23.95
100	27.75	18.80	10.93	10.16	5.09	0.28	4.33	5.64	5.27	21.39

Table 5.4: Mean squared error of  $u_c$  w.r.t. the piecewise constant  $u_{trui}$  51 signal.

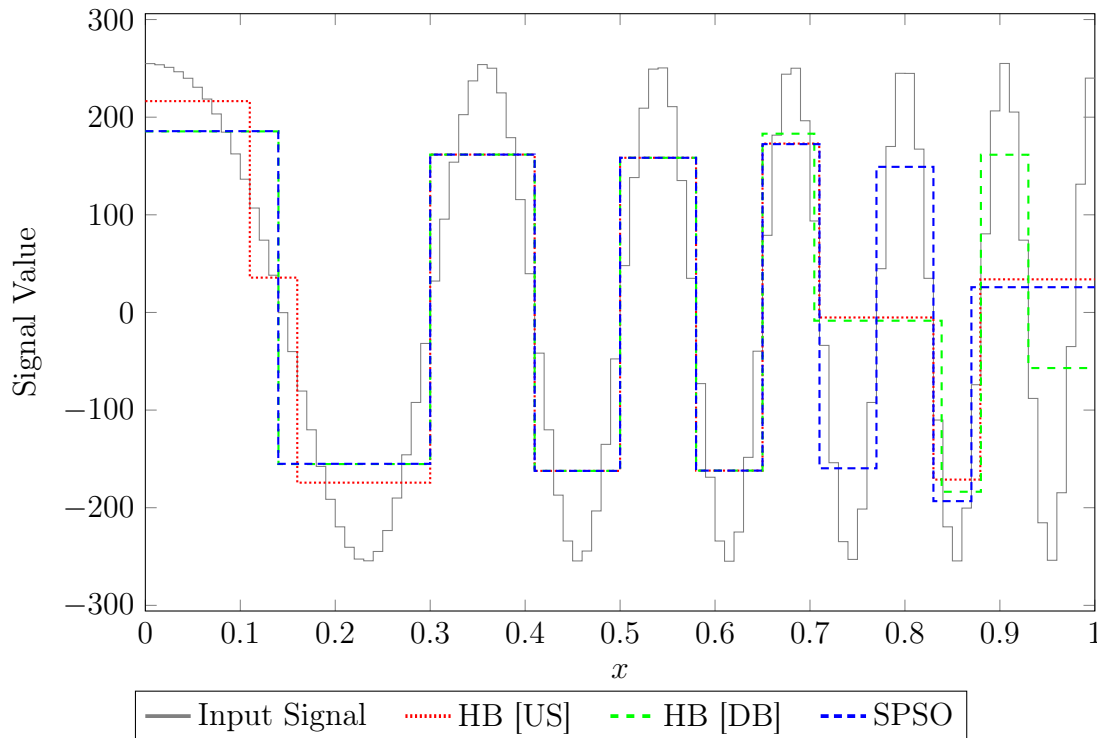


Figure 5.14: Best piecewise constant approximations of the piecewise constant chirp signal for  $N = 11$ .

in this case – GD .

Next, we take a look at the best piecewise constant signal approximations for  $N = 10$ . From Table 5.4 we get that the MSE for GD [DB] is about 1.7 times larger than for SPSO which gives the best results for  $N = 10$ . GD [US] with a MSE of 196.38 lies in between of both methods. We show the corresponding approximation functions  $u_c$  in Figure 5.16. It becomes clear that also visually SPSO adapts best to the input signal, e.g. for  $x \in [0, 60]$  or  $x \in [150, 190]$ . In the same regions GD [DB] can only provide a rough approximation of the input signal.

### Piecewise Linear Input Signal

In our next experiments we make use of piecewise linear input functions. According to the minimal example discussed in Chapter 5.4.2 this can lead to a smooth energy function which allows the usage of more pleasant optimisation techniques than for piecewise constant input functions. For this scenario we consider the same input data as in the previous case, i.e.

- 100 uniformly taken samples of  $f_3$  (as defined in (5.85)), and
- the 256 grey values of line 51 of the 8-bit test image *trui* (cf. Figure 5.11) which we assume to be located at the corresponding pixel centres.

This time – however – we interpolate linearly between the samples and refer to the resulting signals as the piecewise linear chirp signal and the piecewise linear *trui*

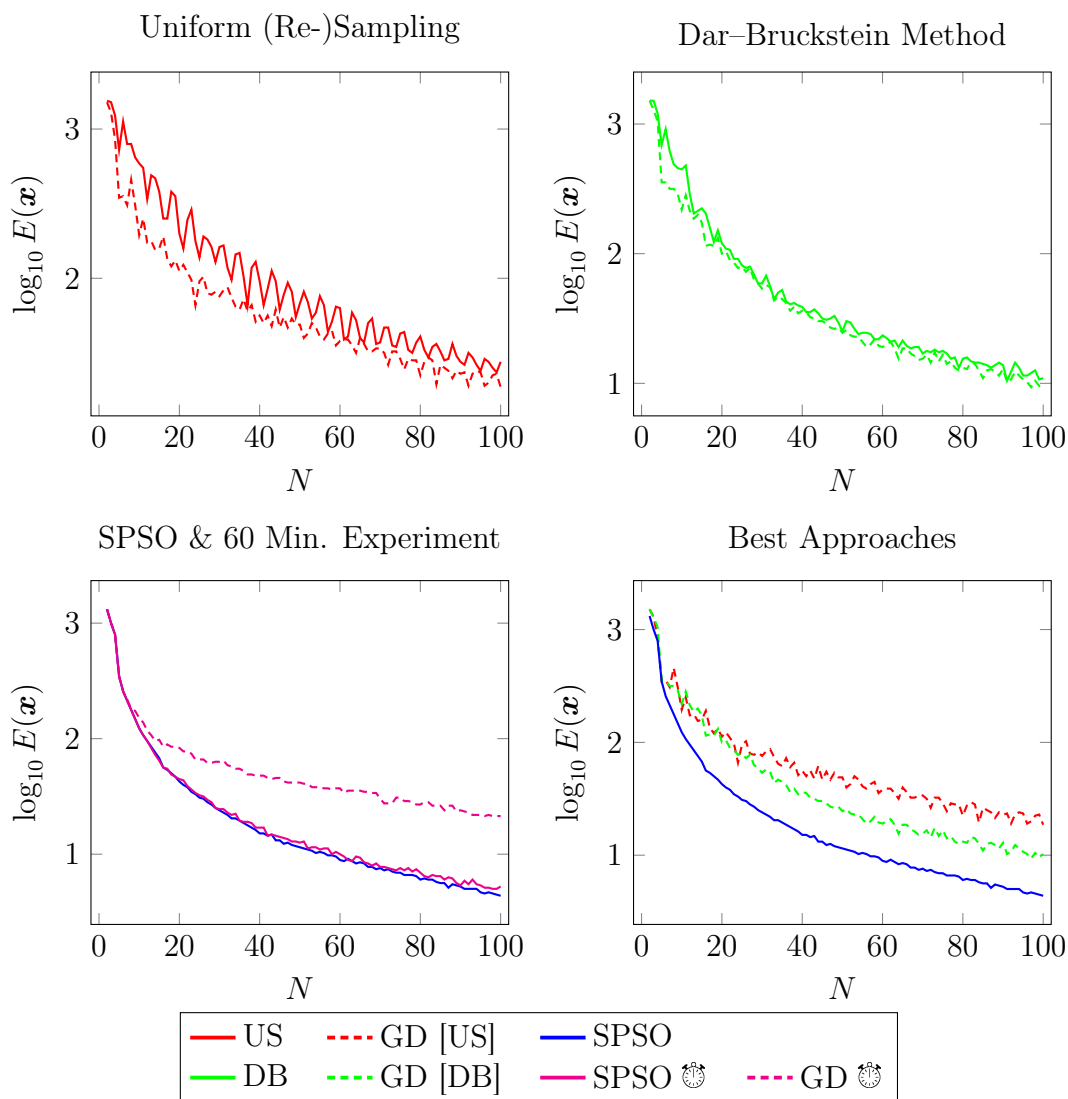


Figure 5.15: Approximation quality of  $u_c$  for the piecewise constant true 51 signal.

51 signal. The corresponding input functions  $f$  are illustrated in Figure 5.17. For both signals we want to estimate the optimal piecewise constant approximation function  $u_c$  given a desired number of segments  $N$ . Based on our findings in Chapter 5.4.2, we do this using

- Uniform (Re-)Sampling (US),
- the Dar-Bruckstein method (DB), using (5.51) to estimate  $f'$  and the threshold (5.67), and
- our direct energy optimisation approach with
  - Standard Particle Optimisation 2011 (SPSO) with a swarm size of  $n = 1000$ ,
  - Adaptive FSI schemes (AFSI) where the results from US, DB, and SPSO serve as initialisation (referred to as AFSI [US], AFSI [DB], and

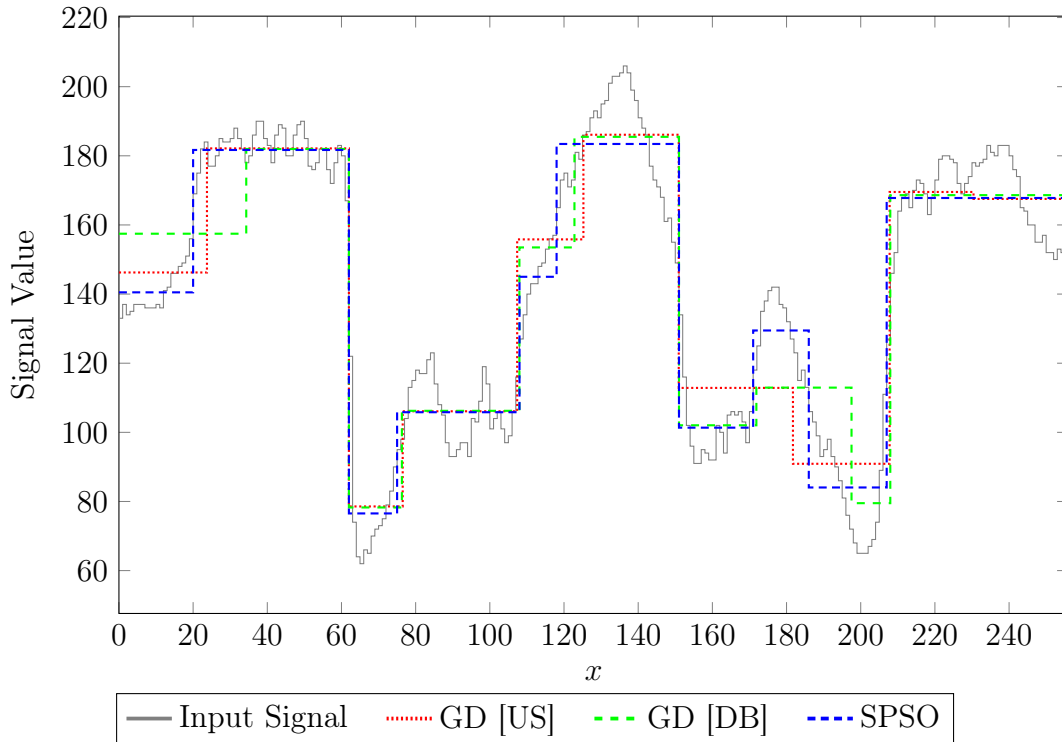


Figure 5.16: Best piecewise constant approximations of the piecewise constant trui 51 signal for  $N = 10$ .

AFSI [SPSO]), and

- randomly initialised SPSO and GD in a 60 minute brute force experiment denoted by SPSO 🕒 and GD 🕒.

**Piecewise Linear Chirp Signal.** Initially, we want to study the approximation results for the piecewise linear chirp signal provided in Table 5.5 and Figure 5.18.

In general, the model ranking in terms of increasing MSE is given by: AFSI [DB], AFSI [SPSO], SPSO, AFSI [US], DB, US. Apart from that, AFSI [SPSO] gives the lowest approximation error for  $N \leq 24$  and AFSI [US] performs better than AFSI [DB] for  $N \leq 10$ . We observe that AFSI can improve the results of US a lot in this range: e.g. the MSE decreases by almost 27% for  $N = 5$ . Another reason why AFSI [US] can beat AFSI [DB] might be the weak performance of DB for low  $N$  where it sometimes even results in higher errors than US (e.g. for  $N = 10$ ). Although  $f$  is guaranteed to be a piecewise linear function, the local linearity assumption of DB is violated for low  $N$ : the input signal  $f$  cannot be represented by a linear function within each segment  $[x_i, x_{i+1}]$ .

It is also worth mentioning that US gives better results than DB for  $N > 88$ . In our opinion this is related to the fact that the piecewise linear chirp signal reduces to 100 uniformly distributed samples. By model definition US can give a somewhat good – though not perfect – approximation when coming close to this number of signal samples.

N	US	AFSI [US]	DB	AFSI [DB]	SPSO	AFSI [SPSO]	SPSO	AFSI	SPSO	AFSI
5	30088.53	21986.48	31440.80	30737.39	18257.93	18257.93	18257.93	18257.93	18257.93	18257.93
10	17440.95	13704.24	21908.89	13945.56	9997.18	9997.18	9997.18	9997.18	9997.18	9997.18
20	11474.42	4898.13	5882.66	3548.81	3263.07	3263.07	3263.07	3263.07	3263.07	3263.07
30	4535.44	2093.27	2956.65	1767.66	1751.58	1751.58	1751.58	1751.58	1751.58	1751.58
40	2676.79	1373.18	1843.80	1088.42	1142.07	1142.07	1142.07	1142.07	1142.07	1142.07
50	1735.23	899.67	1509.19	765.96	764.16	764.16	764.16	764.16	764.16	764.16
60	1216.94	658.06	976.02	538.27	541.56	541.56	541.56	541.56	541.56	541.56
70	908.72	493.42	724.19	403.99	413.97	413.97	413.97	413.97	413.97	413.97
80	714.55	395.46	640.87	318.39	331.36	331.36	331.36	331.36	331.36	331.36
90	550.63	316.99	575.74	259.57	266.14	266.14	266.14	266.14	266.14	266.14
100	439.31	267.22	546.96	213.14	228.31	228.31	228.31	228.31	228.31	228.31

Table 5.5: Mean squared error of  $u_c$  w.r.t. the piecewise linear chirp signal.

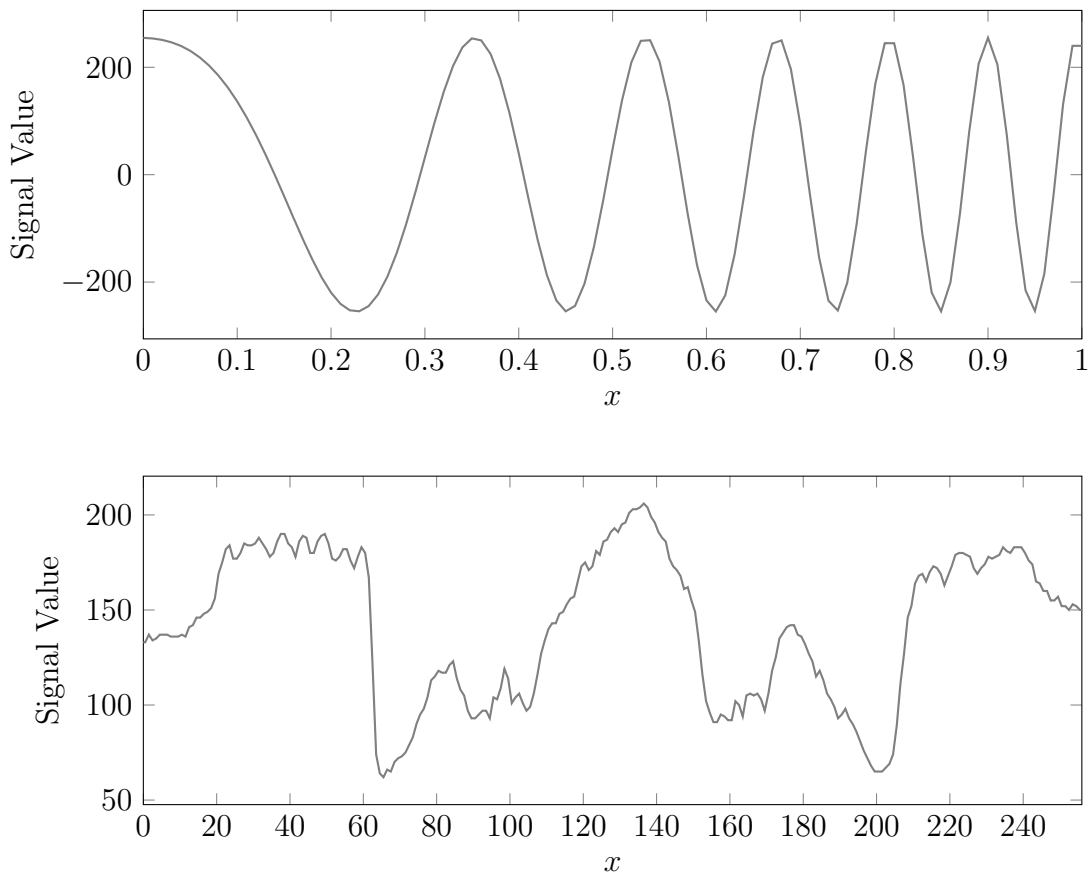


Figure 5.17: The piecewise linear chirp signal at the top and the piecewise linear trui 51 signal below.

Let us now judge the overall performance of SPSO. Considering all values of  $N$ , SPSO on its own already gives good results. For higher values –  $N \geq 54$  – AFSI leads to minor improvements such that we conclude that applying AFSI to SPSO results is not worth the effort. For  $N = 54$  both MSE values differ for the first time by more or equal than  $5 \cdot 10^{-3}$ . This difference increases with growing  $N$  and approaches 3.5 for  $N = 100$ .

Things look different for US and DB: The application of AFSI can greatly improve the results such that the MSE sometimes even decreases by more than 50 percent (see e.g. US for  $N = 30$ ). On average, the MSE of US and DB is 1.8 times higher than for AFSI [US] and AFSI [DB].

Due to the fact that AFSI always converges to a local energy minimum, our experiments again prove the complicated shape of our energy function: no matter which initialisation we choose – US, DB, or SPSO – we always end up in a different minimiser. Based on our findings we conclude that one should use SPSO or AFSI [SPSO] for  $N \leq 24$  whereas for higher  $N$  we suggest AFSI [DB].

From our 60 minute experiment it becomes clear that SPSO 🌀 and AFSI 🌀 with random initialisation yield identical results for  $N \leq 12$ . Furthermore, AFSI 🌀 gives slightly better results than SPSO 🌀 for  $N \geq 60$ . We trace this back to the difficulty of SPSO to deal with high dimensional problems as mentioned by



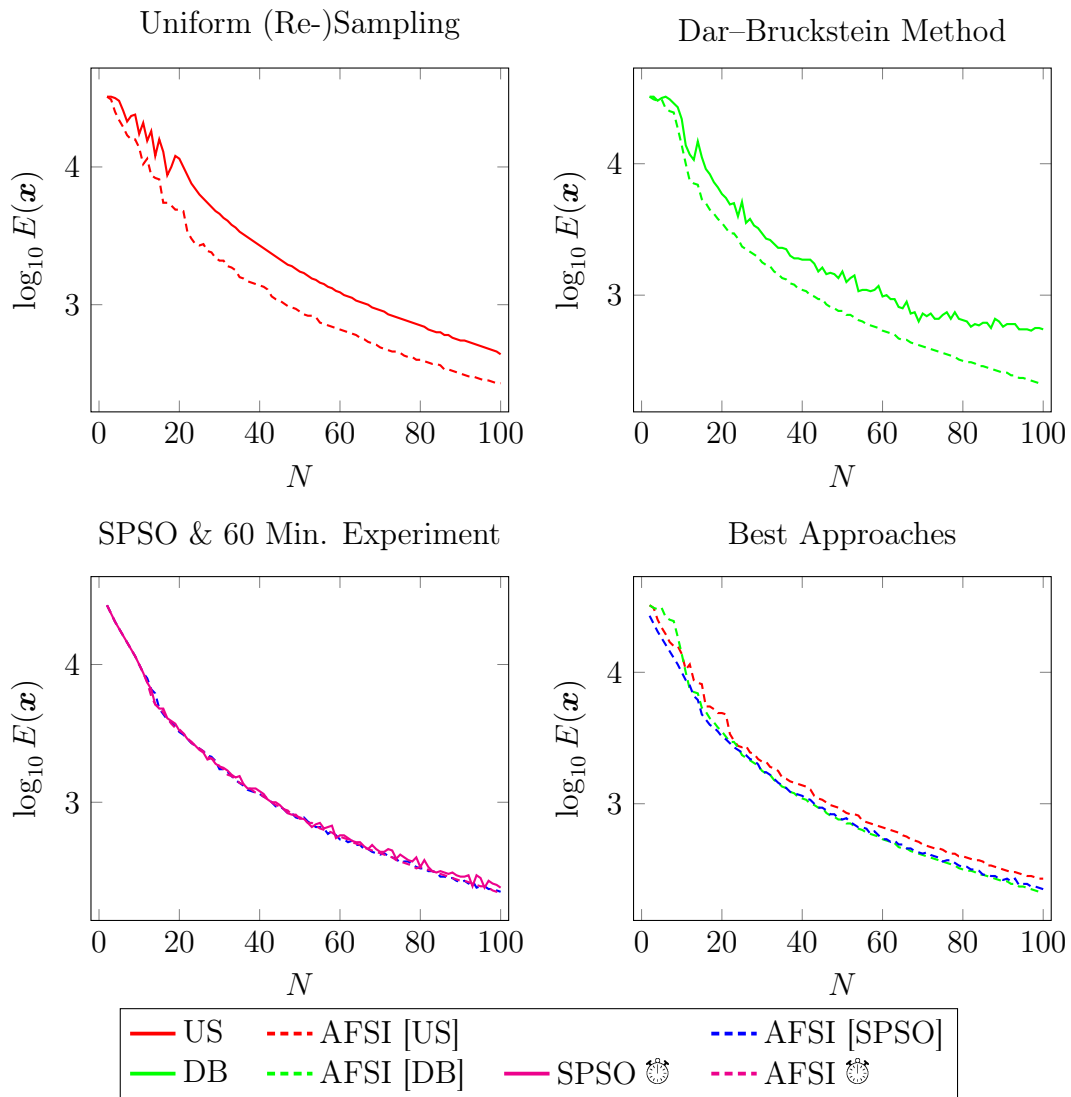


Figure 5.18: Approximation quality of  $u_c$  for the piecewise linear chirp signal.

[ZCR13].

In Figure 5.19 we show the best piecewise constant approximations for  $N = 11$  estimated using AFSI [US], AFSI [DB], and SPSO. Note that AFSI cannot improve the results of SPSO in this setting such that we don't discuss it here. Overall, the outcome resembles our results for the smooth and the piecewise constant chirp signal: Only when using SPSO, the approximation signal  $u_c$  can deal well with low and average input signal frequencies and shifts the error to the high frequent parts ( $x \in [0.86, 1]$ ). Both other methods already fail at the approximation of lower signal frequencies ( $x \in [0.7, 0.82]$ ).

**Piecewise Linear trui 51 Signal.** In our last experiment, we aim at finding piecewise constant approximation functions  $u_c$  which resemble the piecewise linear trui 51 signal. The relationship between the occurring MSE and the corresponding segment number  $N$  becomes clear from the results presented in Table 5.6 and

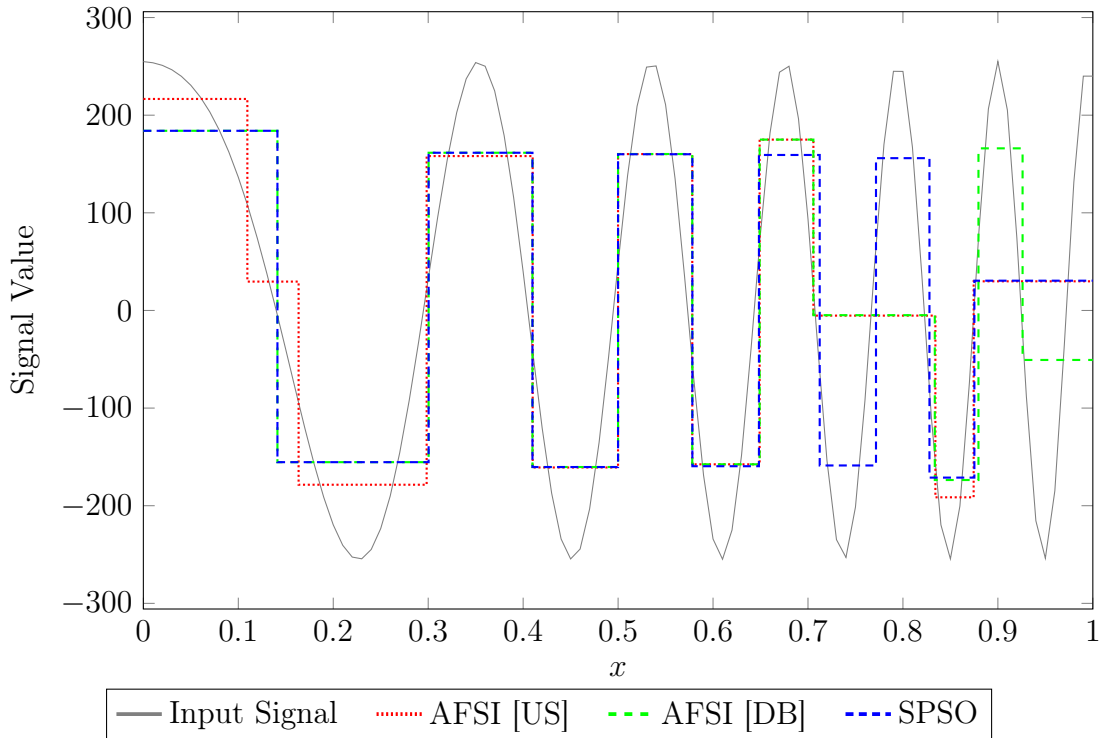


Figure 5.19: Best piecewise constant approximations of the piecewise linear chirp signal for  $N = 11$ .

Figure 5.20. We refrain from plotting the results of AFSI [SPSO] and SPSO  $\odot$  since they are identical to those of SPSO: the MSE differs by less than  $10^{-2}$ .

We observe that for every  $N \in [2, 100]$  it is always SPSO which provides the boundary configuration  $\mathbf{x}$  resulting in the lowest MSE. Only in case of  $N = 5$  other methods like AFSI [US] and AFSI [DB] yield the same MSE value. For  $N \geq 18$ , AFSI [DB] yields the second lowest MSE values, followed by AFSI [US], DB, and US. Like for the piecewise linear chirp signal, AFSI [US] beats AFSI [DB] for low values of  $N$ . Furthermore, US gives better results than DB for very small  $N$ , i.e.  $N < 5$ . The approximation error of AFSI [DB] comes close to the MSE of SPSO with growing  $N$ : While for  $N = 10$  there is a significant gap (the MSE of AFSI [DB] is 1.5 times higher than for SPSO) both methods yield results of almost identical quality for  $N = 100$ .

Although AFSI cannot improve the SPSO results, it causes dramatic enhancement when initialised with the outcome of US and DB: the MSE for US is on average 3.5 time higher than for AFSI [US], the MSE values for DB are on average 2.0 times higher than for AFSI [DB].

When considering the results of our 60 minute brute force random experiment we see that AFSI  $\odot$  is not able to beat SPSO  $\odot$ .

Let us now take a look at Figure 5.21 which shows the best approximations of the piecewise linear trui 51 signal for  $N = 10$  using the results of AFSI [US], AFSI [DB], and SPSO. Altogether, the three methods provide a good approximation of the input signal. However, SPSO clearly outperforms the other two techniques for  $x \in [150, 185]$  where its approximation adapts well to the input

N	US	AFSI [US]	DB	AFSI [DB]	SPSO	AFSI [SPSO]	SPSO	AFSI [SPSO]	SPSO	AFSI
5	718.24	344.69	669.21	344.69	344.69	344.69	344.69	344.69	344.69	344.69
10	583.60	131.97	435.23	182.77	120.65	120.65	120.65	120.65	120.65	120.65
20	191.91	54.01	111.62	58.33	43.33	43.33	43.33	43.33	43.33	47.19
30	155.99	36.77	52.28	31.32	25.31	25.31	25.31	25.31	25.31	29.79
40	90.73	32.62	33.20	20.42	16.31	16.31	16.31	16.31	16.31	18.94
50	60.33	17.29	25.60	13.43	11.66	11.66	11.66	11.66	11.66	14.09
60	57.08	12.97	19.37	10.45	9.00	9.00	9.00	9.00	9.00	10.49
70	32.17	9.89	14.00	7.97	7.47	7.47	7.47	7.47	7.47	9.11
80	35.31	8.96	13.39	6.50	6.11	6.11	6.11	6.11	6.11	6.89
90	21.14	6.85	10.84	5.22	5.04	5.04	5.04	5.04	5.04	6.22
100	22.11	6.02	10.85	4.28	4.26	4.26	4.26	4.26	4.26	5.26

Table 5.6: Mean squared error of  $u_c$  w.r.t. the piecewise linear true 51 signal.

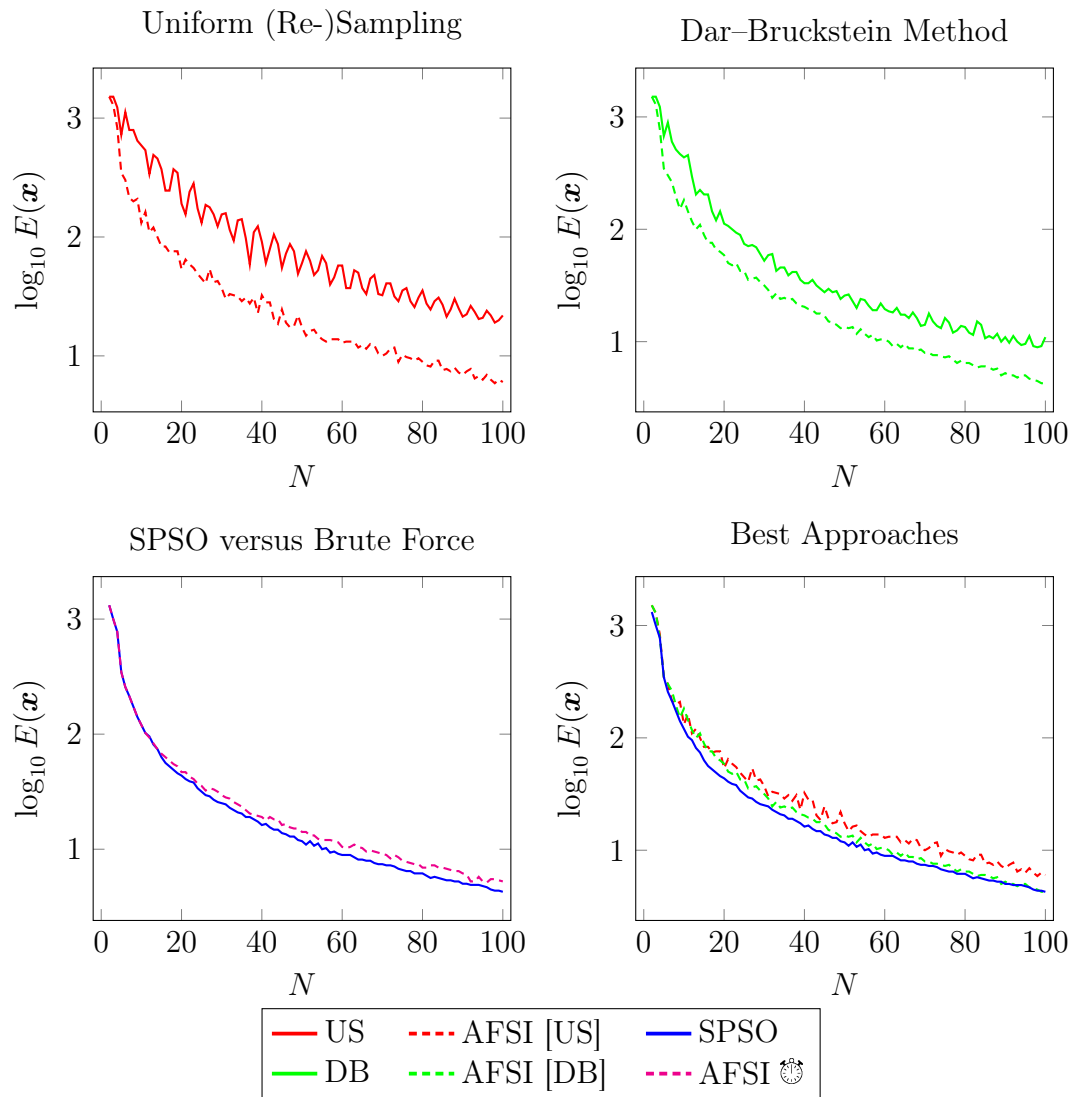


Figure 5.20: Approximation quality of  $u_c$  for the piecewise linear trui 51 signal.

signal structure. In comparison, AFSI [US] and AFSI [DB] only provide a flat approximation there. Considering the whole domain, AFSI [DB] gives the crudest result and – accordingly – the highest MSE. Our findings are similar to those for  $N = 10$  for the piecewise constant trui 51 signal shown in Figure 5.16. However, for the piecewise linear input signal the applied techniques seem to have less difficulties to adapt to the input signal.

### 5.5.2 Piecewise Linear Approximation Functions $u_\ell(x)$

The second part of our experiments is dedicated to optimal piecewise linear approximation functions  $u_\ell$  for piecewise constant and piecewise linear input functions  $f$ . More precisely, we want to find the function  $u_\ell$  which minimises the MSE w.r.t. the input signals  $f_1, f_2$ , the piecewise constant trui 51 signal, and the piecewise linear trui 51 signal. Based on our findings in Chapter 5.2.2, we follow our direct energy optimisation strategy and employ a combination of the

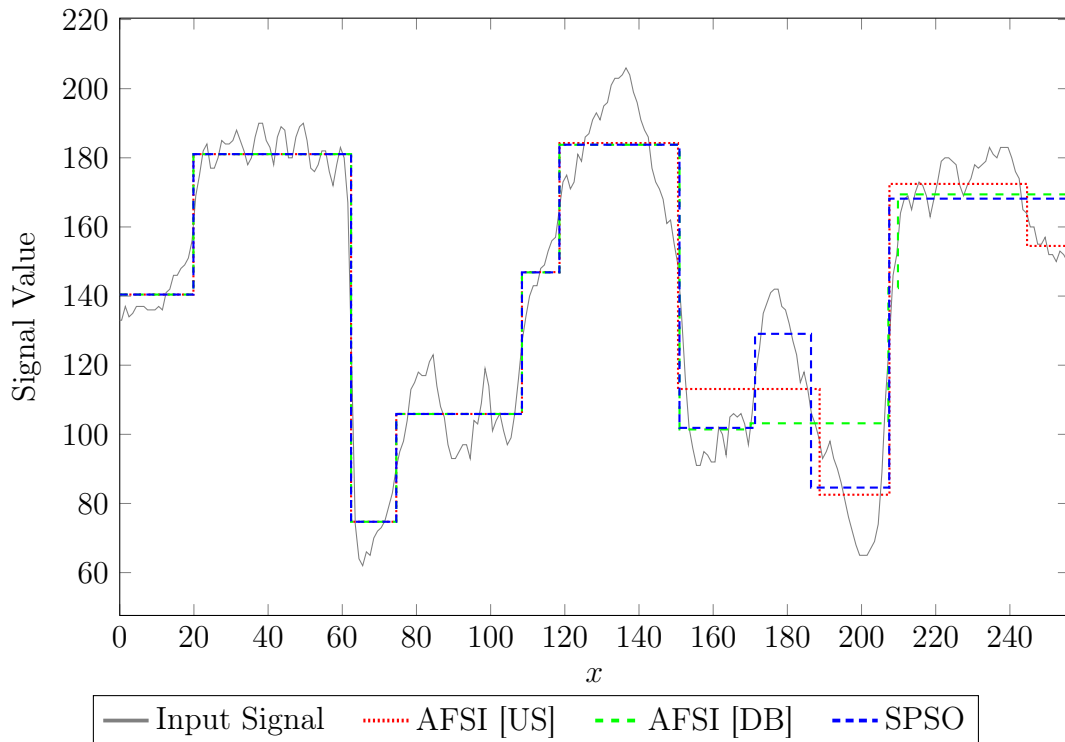


Figure 5.21: Best piecewise constant approximations of the piecewise linear true signal for  $N = 10$ .

Standard Particle Optimisation 2011 (SPSO) algorithm (see Chapter 2.2.5) and the gradient descent (GD) method (see Chapter 2.2.1) to estimate a minimiser of the energy (5.12) for a given number of segments  $N$ . We use GD to calculate the unique minimiser  $\mathbf{u}$  for given segment boundaries  $\mathbf{x}$ . The strict convexity of  $E_f(\mathbf{x}, \mathbf{u})$  in  $\mathbf{u}$  (cf. Theorem 12) allows us to do this efficiently. Accordingly, we utilise SPSO to find the boundary configuration  $\mathbf{x}$  which results in the lowest MSE.

For  $f_1$  and  $f_2$  we use a swarm size of  $n = 100$ , when working with the true signal data we set  $n = 1000$ . In every experiment we initialise the segment boundaries  $\mathbf{x}$  randomly on the signal domain following a uniform distribution. Nevertheless, the boundary positions are sorted and fulfil (5.1). As mentioned before, we estimate the corresponding initial optimal tonal configuration  $\mathbf{u}$  with the help of the gradient descent algorithm.

**Minimal Example Using  $f_1$  and  $f_2$ .** In order to get a better understanding of the estimation process for piecewise linear approximation functions  $u_\ell$  we begin our experiments with an investigation of the approximation problem for the input functions  $f_1$  (as defined in (5.45)) and  $f_2$  (as defined in (5.56)). Both functions can be regarded as a minimalistic representative of a piecewise constant and a piecewise linear function and are shown in Figure 5.1 and Figure 5.2. In our experiments, we consider  $N = 2, 3, \dots, 7$  segments for both input functions. We provide the resulting MSE values in Table 5.7 and illustrate them together with the corresponding functions  $u_\ell$  for  $N = 4$  and  $N = 7$  in Figure 5.22.

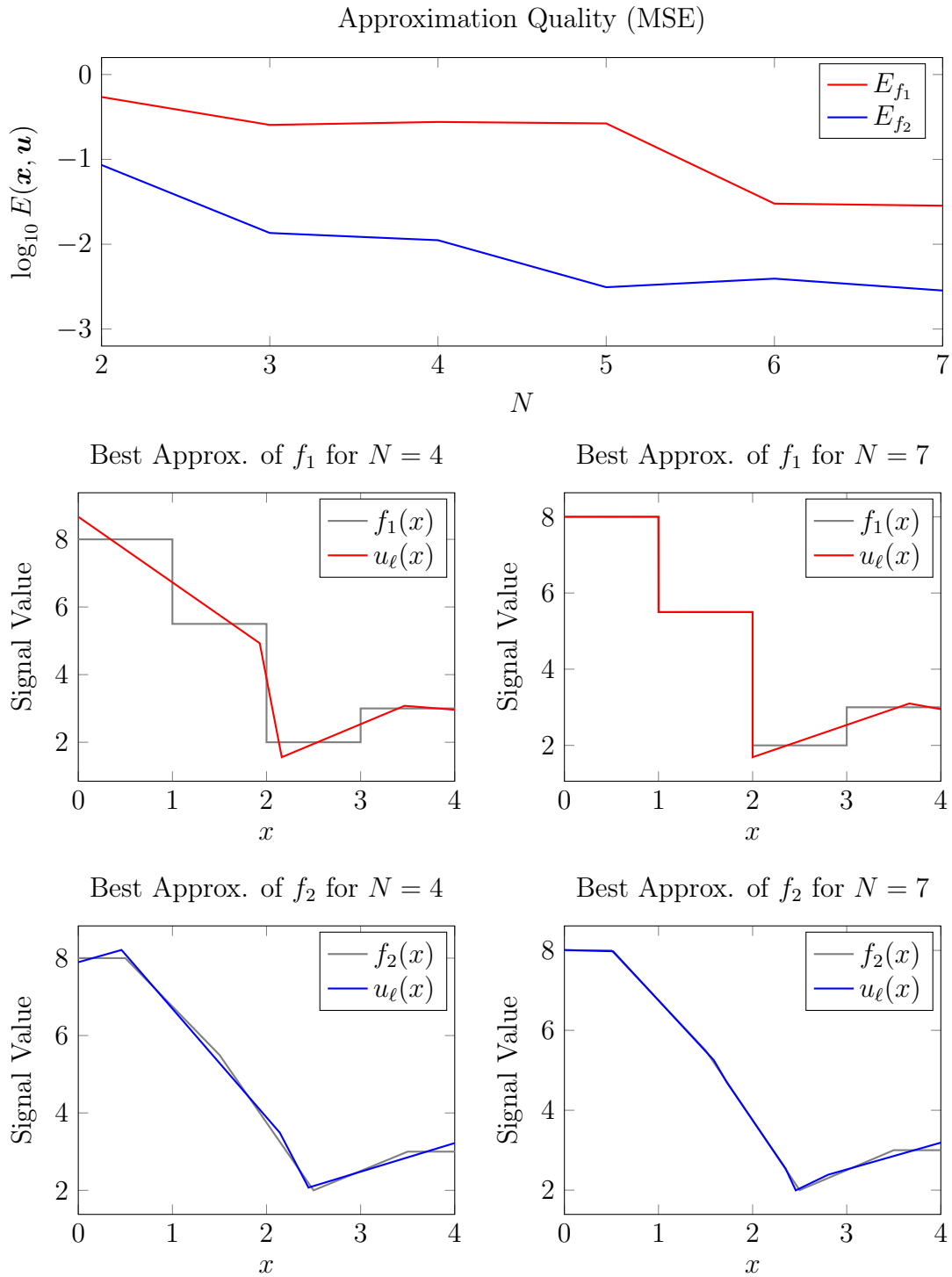


Figure 5.22: Piecewise linear approximation of  $f_1$  and  $f_2$ .

$N$	SPSO $f_1$	SPSO $f_2$	$N$	SPSO $f_1$	SPSO $f_2$
2	0.54314	0.085843	5	0.26523	0.003112
3	0.25468	0.013562	6	0.03008	0.003927
4	0.27604	0.011146	7	0.02845	0.002846

Table 5.7: Mean squared error of  $u_\ell$  w.r.t.  $f_1$  and  $f_2$ .

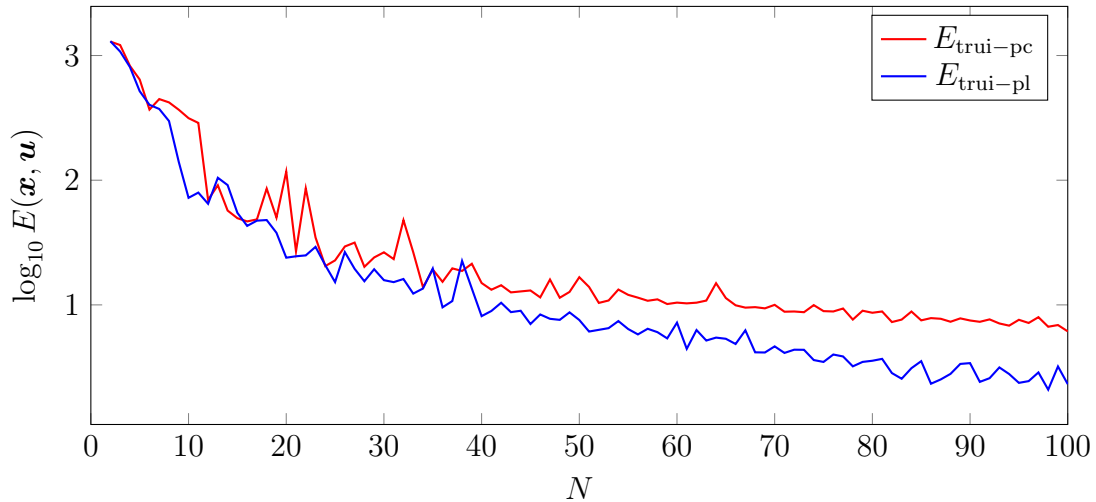


Figure 5.23: Approximation quality of  $u_\ell$  for the piecewise constant (trui-pc) and piecewise linear (trui-pl) trui 51 signal.

As one can see, the MSE values for the continuous linear input  $f_2$  are lower than for  $f_1$  for all values of  $N$ : On average the MSE for  $f_1$  is 25 times higher. This makes sense since there are no jumps in the function  $f_2$  (for  $f_1$  there are jumps) and according to (5.10) we do not allow any jumps in the approximation function  $u_\ell$ . Thus, by definition  $u_\ell$  is better suited to approximate  $f_2$  than  $f_1$ . Nevertheless, the approximation of  $f_1$  works – from this point of view – surprisingly well. For  $N = 7$  we observe only small difficulties for  $x \in [2, 4]$  while for  $N = 4$  we only get a crude approximation of  $f_1$ . This is different when using the input signal  $f_2$ . Already for  $N = 2$  the approximation function  $u_\ell$  fits  $f_2$  well: The MSE is below  $10^{-1}$  and significantly lower than for the piecewise constant input function. As shown in Figure 5.22, we get a good signal approximation for  $N = 4$  and an almost perfect fit for  $N = 7$ . Apart from that, the latter result again emphasises the difficulty of our minimisation problem: The input function  $f_2$  consists – by definition – of 5 linear segments such that for  $N \geq 5$  we know that there exist multiple functions  $u_\ell$  which imply an energy value of 0. SPSO allows us to find a good approximation function but it fails to estimate one of the global energy minimisers.

**Piecewise Constant and Linear trui 51 Signal.** Keeping these results in mind, let us now continue with experiments on real world data. For this purpose we consider again the piecewise constant and piecewise linear trui 51 input signal which we refer to as trui-pc and trui-pl below. Both signals are introduced in Chapter 5.5.1 and we visualise them in Figure 5.12 and Figure 5.17. In our experiments, we try to estimate the best possible piecewise linear approximation function  $u_\ell$  for trui-pc and trui-pl using  $N = 2, 3, \dots, 100$ . The achieved MSE values for both input functions are given in Table 5.8 and sketched in Figure 5.23. As before, we obtain – in general – better results for the piecewise linear input function. Again, we trace this back to our model design and refer to the fact that continuous piecewise linear approximation functions cannot deal well with

N	SPSO trui-pc	SPSO trui-pl	N	SPSO trui-pc	SPSO trui-pl
5	641.86	517.45	60	10.43	7.21
10	314.51	72.21	70	10.01	4.65
20	117.16	23.91	80	8.65	3.56
30	26.40	15.79	90	7.50	3.41
40	14.98	8.12	100	6.13	2.31
50	16.67	7.56			

Table 5.8: Mean squared error of  $u_\ell$  w.r.t. the piecewise constant (trui-pc) and piecewise linear (trui-pl) trui 51 signal.

jump discontinuities in the input function. In the specific case of trui-pc and trui-pl, the MSE for the piecewise constant input is lower only in 8 of 99 cases, i.e. for  $N \in [2, 6, 13, 14, 15, 24, 35, 38]$ . The superiority of trui-pc for some particular values of  $N$  can e.g. be explained by the shape of the input signal or the random and non-deterministic behaviour of SPSO. We observe that on average the MSE for the input function trui-pc is 2.5 times higher than for trui-pl. As one can also see from Figure 5.23 the gap between the MSE for trui-pc and trui-pl becomes larger with a growing number of segments  $N$ .

On top of that, we want to compare the estimated approximation functions with lowest MSE for both input functions, trui-pc and trui-pl, in case of  $N \in \{5, 10, 20\}$ . For this purpose, we again consider Table 5.8 and the corresponding plots in Figure 5.24. First of all, we notice that the main characteristics of the input signals are captured in both settings. However, not only in terms of MSE but also visually, the achieved approximation of trui-pl excels the one of trui-pc. For example, the central signal bump –  $x \in [110, 160]$  – is only resembled properly for  $N = 20$  in case of trui-pl. For  $N = 10$ , the estimate  $u_\ell$  fails to approximate trui-pc for  $x \in [130, 200]$ . Obviously, the approximation error in case of trui-pl is much lower for this region of  $x$ . In general, we observe that for the input signal trui-pl there exist less “undershoots” than for trui-pc. Overall, the achieved functions tend to cross the input signal more often and represent a more crude approximation when using trui-pc.

## 5.6 Conclusions and Outlook

Within this chapter we investigated the problem of attaining  $\ell^2$ -optimal approximations of one-dimensional signals for a fixed number of samples. In particular, we focussed on piecewise constant and piecewise linear functions which approximate interpolated discrete input data.

In the beginning, we provided a general energy-based model for one-dimensional signal approximation. On top, we introduced the corresponding nonconvex minimisation problem which aims at finding the global minimiser of the MSE w.r.t. the original signal. This also included adapted energy formulations for piecewise constant and piecewise linear output signals. In this context, we showed that the problem of finding optimal sample positions is in general nonconvex while the estimation of the corresponding optimal sample values is convex and can be solved



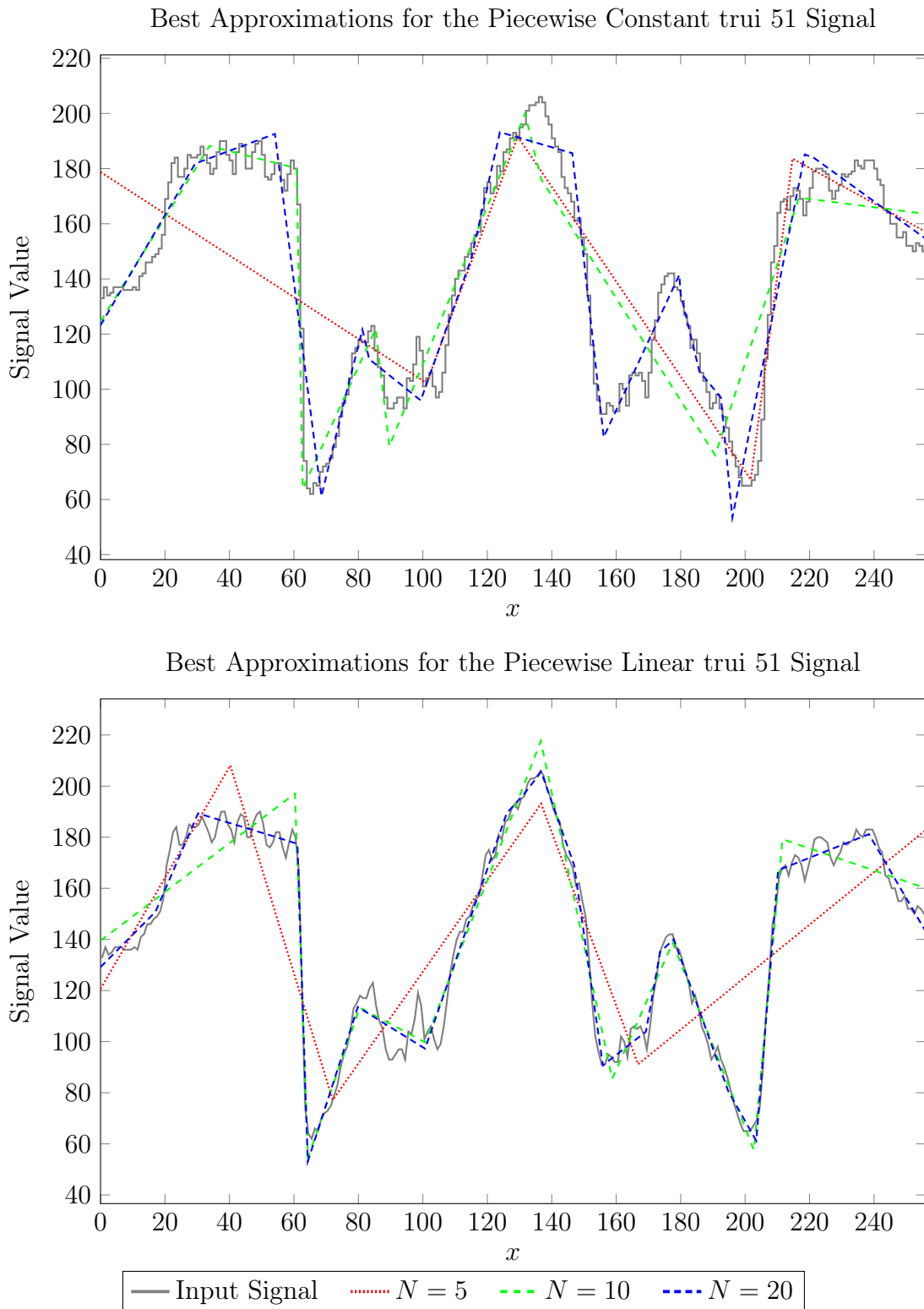


Figure 5.24: Best piecewise linear approximations of the piecewise constant and the piecewise linear trui 51 signal for  $N \in \{5, 10, 20\}$ .

with minimal effort. This coincides with previous research from the area of homogeneous diffusion [MHW<sup>+</sup>12] which studied the spatial and tonal optimisation of inpainting masks.

In a second step, we have supplied a simple alternative derivation of the recent Dar–Bruckstein model. As part of its analysis, we analysed the limitations of their approach and pointed out that the approximation quality suffers in case of violations of the signal smoothness or local linearity. Furthermore, we proved that also error balancing can induce suboptimal approximations.

Motivated by these findings, we pursued the idea of direct minimisation of the previously introduced energy. In this way, we renounced all limiting assumptions of the Dar–Bruckstein model and provided an approach which leads to a globally optimal solution in case the corresponding nonconvex optimisation problem can be solved exactly. For this purpose, we suggested two fundamentally different solution strategies. Independent of the signal characteristics, a particle swarm optimisation approach achieves high quality approximations. However, due to its random nature it lacks of any convergence guarantees. We showed that numerical first-order optimisation methods – which provide such guarantees – represent a reasonable alternative for Lipschitz continuous input signals. Furthermore, an adaptation to less regular signals involving jumps turned out to be possible with the help of automatic time step size selection via backtracking line search.

In our experiments for piecewise constant approximations, we evaluated the quality of our direct energy minimisation approach in comparison to the Dar–Bruckstein method and uniform resampling. The latter served as a lowest tolerable quality reference. When considering smooth input signals, we observed that particle swarm optimisation and first-order optimisation techniques perform almost equally well. For low samples numbers  $N$ , Adaptive FSI (AFSI) schemes using Standard Particle Swarm Optimisation 2011 (SPSO) results as initialisation gained the best quality. For higher  $N$ , AFSI initialised with the Dar–Bruckstein (DB) method works best. With increased problem complexity – in case of piecewise constant input signals – the best approximation quality results from SPSO. It performed clearly better than its followers, the heavy ball (HB) or the gradient descent (GD) algorithm using DB results for initialisation. In time limited experiments we also validated that SPSO is superior to a random brute force method (GD and HB with random initialisation). For piecewise linear input we got similar behaviour as for smooth input functions: AFSI based approaches and SPSO attained nearly the same quality. However, SPSO performed constantly better than the other methods on real world data. In general, we observed weak performance of DB for low sample numbers or signals lacking smoothness. Particle swarm optimisation represents the method of choice in these cases. Apart from that, first-order minimisation methods initialised with DB results offer a simple and efficient alternative to SPSO for piecewise linear or smooth input signals. Additionally, the experiments showed the eligibility of using SPSO in practice since applying GD, HB, or AFSI (with SPSO initialisation) lead to no or only small improvements of the MSE value. This means that SPSO was able to detect a local minimum or at least provided a solution close to one.

The second part of our experiments dealt with piecewise linear approximations

and verified our theoretical findings. We found out that also in this setting, SPSO is a suitable tool to estimate a good signal approximation.

From all experiments, we learned that piecewise linear input signals are more pleasant to handle than piecewise constant ones. This becomes, for example, relevant when considering alternatives to particle swarm optimisation in the context of direct energy minimisation. When doing so, we suggest to interpolate samples linearly in case of discrete input data. In summary, the application of numerical first-order optimisation methods pays off when initialised with DB or US results. However, the gained MSE reduction is higher for piecewise linear input functions. Another interesting fact is that the resulting approximations from SPSO for the chirp signal – and its discrete variants – show that a MSE minimisation for a fixed number of samples basically acts as a low pass filter. While the approximation signal adapts well to the lower frequencies, the error is shifted to high signal frequencies. This sounds reasonable and implements a well-known practice from the field of compression.

This chapter sheds light on the possibilities and benefits of direct energy minimisation for the purpose of one-dimensional signal approximation. Future research in this area might e.g. concentrate on parameter tuning and speeding-up for particle swarm optimisation techniques. We found a general parameter setting that worked well in our experiments. However, we believe that the SPSO model parameters can be adapted in dependency of the input signal characteristics and the desired number of samples. Doing so could – amongst others – result in a speed-up of the minimisation process and increase the competitiveness of the SPSO algorithm. Also, the extension of our theory to a multi-dimensional setting represents another interesting challenge. Highly relevant applications exist, e.g. in the field of image processing. Establishing a link to work dealing with compression, non-uniform sampling, or piecewise constant approximation [Sau98, KBPW18, Dav11] could be a first step in this direction.



---

# Chapter 6

## Conclusions and Outlook

“We are all apprentices in a craft where no one ever becomes a master.”

---

Ernest Hemingway, *New York Journal-American*

### Contents

---

6.1	Summary and Conclusions . . . . .	125
6.2	Outlook . . . . .	127

---

### 6.1 Summary and Conclusions

The focus of this thesis was on evolutionary mathematical models for the purpose of signal enhancement and approximation. In Chapter 1, we stated three goals for this present work. First, we wanted to show that – and elaborate how – the mathematical modelling of swarm dynamics represents an appropriate strategy to solve image processing tasks. As our second goal, we defined the derivation of a smart and stable mathematical model for pure backward diffusion. The last aim we formulated was to provide new insights into the optimal adaptive sampling of arbitrary one-dimensional signals under the constraint of limited resources. To achieve these goals, we designed and analysed domain-specific gradient descent processes. Subsequently, we summarise our contributions in the different areas.

**Attractive-Repulsive Swarming Models for Image Processing.** In Chapter 3, we built on the idea of attractive-repulsive discrete first-order models of swarming. They describe the movement of individual swarm members with the help of potential forces, a principle which is well-known in literature for modelling swarm dynamics. We extended the time evolution of discrete first-order models with an additional weighting function and obtained a model which is covered by the theory of nonsymmetric nonlocal evolutions. In experiments, we proved the usefulness of our model as an intuitive modelling tool for image processing tasks. We used our model to formulate grey scale quantisation, contrast enhancement,

line detection, and coherence enhancement as a swarm evolution which solves the corresponding problem in its steady state.

**Purely Repulsive Models and Backward Diffusion.** Motivated by the concept of a purely repulsive discrete first-order model of swarming, we presented a new and intelligent model for a specific class of backward diffusion in Chapter 4. Our model is characterised by remarkable properties: First of all, we designed the model in such a way that it implements globally negative diffusivities. Furthermore, we equipped our model with reflecting boundary conditions in the diffusion co-domain in order to stabilise the underlying ill-posed backward diffusion problem. This was the first time that this type of constraint is used within the context of backward diffusion. In our model analysis, we pointed out that our backward diffusion process describes a gradient descent evolution on a convex energy. This was not to be expected and allowed us to prove convergence of our model to a unique minimiser. Closely connected, we discussed an important numerical benefit of our model: In contrast to existing approaches, already a simple explicit scheme inherits the stability and convergence properties of the time-continuous evolution. In our experiments, we demonstrated the applicability of our backward diffusion model for the purpose of global and local contrast enhancement of greyscale and colour images.

**Evolutions for One-Dimensional Signal Approximation.** Our study of  $\ell^2$ -optimal one-dimensional signal approximation was the topic of Chapter 5. Therein, we discussed the unsolved problem of estimating a piecewise-defined function for a fixed number of samples that represents an approximation to an arbitrary input signal which is optimal in a least-squares sense. For the benefit of minimal model restrictions, we followed a direct minimisation approach of an energy which is nonconvex in general. In the context of piecewise constant and piecewise linear approximation functions, we showed that our task reduces to an estimation of optimal interval boundaries. This was based on our finding that – for given interval boundaries – there exists a unique set of optimal sample values which can be determined easily. For piecewise constant approximation functions, we provided a concise reformulation of the Dar–Bruckstein model that allowed us to prove the ineligibility of error balancing as a criterion for  $\ell^2$ -optimality. In our experiments for piecewise constant approximation functions, we compared the approximation quality of our approach using a particle swarm optimisation strategy and numerical first-order methods with the results of the Dar–Bruckstein model. We achieved results of high quality and could outperform the method of Dar and Bruckstein. Additionally, we illustrated the high potential of our model in experiments for piecewise linear approximation functions.

**General Conclusions.** Not only this PhD thesis has highlighted the value of looking into a subject from its roots but also the rich potential and chances this strategy has to offer. We began our journey with an exploration of the highly interdisciplinary field of swarm dynamics. A deeper understanding of existing

models and the embedding into a well-established mathematical framework enabled us to express swarm evolutions in terms of differential equations. In a next step, we successfully used this achievement to reformulate image processing tasks as swarming processes that allow an intuitive understanding of abstract methods. This new possibility to express and reinterpret classical problems offered us – in combination with the flexibility of our evolution equations – the chance to derive a novel and ground-breaking model for solving the backward diffusion equation with standard numerics. In case of our signal approximation studies, we could disprove the suitability of error balancing as an optimality criterion and provided new insights along with state-of-the-art results. Hence, we see how important it is to question even basic assumptions or apparently solved problems. For us this was the key to progress and success.

## 6.2 Outlook

In addition to the proposals made at the end of the individual chapters, let us briefly discuss a few more potential future research directions.

**Purely Repulsive Models and Backward Diffusion.** As proposed, our backward diffusion model from Chapter 4 allows to describe the evolution of one-dimensional data like – in our case – grey or intensity values of digital images. A reasonable extension of the suggested evolution represents the development of a theory for the  $n$ -dimensional case. Amongst others, this requires to solve potential issues regarding the non-uniqueness of the corresponding minimiser and the smart implementation of the reflecting boundary conditions in a multi-dimensional setup. On the other hand, this could pave the road for further application scenarios. Connected to our experiments, such an  $n$ -dimensional model could be directly applied to colour images. Other interesting use cases might be the processing of depth maps or three-dimensional models as used in computer vision or the navigation of autonomous vehicles in robotics.

**Evolutions for One-Dimensional Signal Approximation.** Our approach in Chapter 5 reveals the idea of signal approximation with tonal optimisation. This is a common approach in image processing and – more specifically – in image compression. Inspired by this, and based on our findings, we suggest to study an evolutionary process which models the idea of density approximation in a two-dimensional setup. This could be useful for inpainting and compression of digital images. One possible starting point for research could be the implementation of a minimisation process on the inpainting or reconstruction error of the image. In this context, powerful tools to achieve this goal could be Voronoi diagrams or Delaunay triangulations of the image domain. Both concepts would allow to define a model that acts on well-defined subsets of the image domain. Helpful ideas for modelling could e.g. be found in works on stippling [Sec02], hierarchical data representation [SBHJ00], or centroidal Voronoi tessellations [DFG99].

Overall, we believe in the future of nature-inspired evolution models that rely on a well-founded mathematical background as such approaches represent solid, powerful, and intuitive ways to solve difficult problems. In this work, we could just scratch the surface of evolution equations which are inspired by studies on swarm behaviour. At the same time, we were able to discover the rich potential of this idea and appreciated the interdisciplinarity in our work. The understanding of connections to other research areas and well-known mathematical models was of great importance and motivated us to study the – amongst others – occurring complex dynamics. In conclusion, we think that our contributions provide a good basis for future research and hope that they will serve as an inspiration for further studies.



---

# Appendix A

## Bibliography

- [AD19] R. Alexandru and P. L. Dragotti. Time-based Sampling and Reconstruction of Non-bandlimited Signals. In *IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP 2019, Brighton, United Kingdom*, pages 7948–7952. IEEE, May 2019.
- [AG01] A. Aldroubi and K. Gröchenig. Nonuniform Sampling and Reconstruction in Shift-Invariant Spaces. *SIAM Review*, 43(4):585–620, 2001.
- [AMFM11] P. Arbelaez, M. Maire, C. Fowlkes, and J. Malik. Contour Detection and Hierarchical Image Segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 33(5):898–916, 2011.
- [ARRM14] R. Aranda, M. Rivera, and A. Ramirez-Manzanares. A flocking based method for brain tractography. *Medical Image Analysis*, 18(3):515–530, April 2014.
- [BBBW09] Z. Belhachmi, D. Bucur, B. Burgeth, and J. Weickert. How to Choose Interpolation Data in Images. *SIAM Journal of Applied Mathematics*, 70(1):333–352, 2009.
- [BCWW18] L. Bergerhoff, M. Cárdenas, J. Weickert, and M. Welk. Modelling Stable Backward Diffusion and Repulsive Swarms with Convex Energies and Range Constraints. In M. Pelillo and E. Hancock, editors, *Energy Minimization Methods in Computer Vision and Pattern Recognition, 11th International Conference, EMMCVPR 2017, Venice, Italy*, volume 10746 of *Lecture Notes in Computer Science*, pages 409–423, Cham, March 2018. Springer.
- [BCWW20a] L. Bergerhoff, M. Cárdenas, J. Weickert, and M. Welk. Stable Backward Diffusion Models that Minimise Convex Energies. *Journal of Mathematical Imaging and Vision*, 62(6):941–960, July 2020.
- [BCWW20b] L. Bergerhoff, M. Cárdenas, J. Weickert, and M. Welk. Stable Backward Diffusion Models that Minimise Convex Energies. arXiv:1903.03491v2 [math.NA], June 2020.

- [BK07] N. Bassiou and C. Kotropoulos. Color image histogram equalization by absolute discounting back-off. *Computer Vision and Image Understanding*, 107(1):108–122, July 2007.
- [BM08] C. Blum and D. Merkle, editors. *Swarm Intelligence: Introduction and Applications*. Natural Computing Series. Springer, Berlin, 2008.
- [BM17] M. R. Bonyadi and Z. Michalewicz. Particle Swarm Optimization for Single Objective Continuous Space Problems: A Review. *Evolutionary Computation*, 25(1):1–54, 2017.
- [BPP98] N. N. Brueller, N. Peterfreund, and M. Porat. Non-stationary signals: optimal sampling and instantaneous bandwidth estimation. In *Proc. IEEE-SP International Symposium on Time-Frequency and Time-Scale Analysis*, pages 113–115, Pittsburgh, PA, USA, October 1998.
- [BV04] S. Boyd and L. Vandenberghe. *Convex Optimization*. Cambridge University Press, Cambridge, March 2004. <https://web.stanford.edu/~boyd/cvxbook/>.
- [BW16] L. Bergerhoff and J. Weickert. Modelling Image Processing with Discrete First-Order Swarms. In N. Pillay, P. A. Engelbrecht, A. Abraham, C. M. du Plessis, V. Snášel, and K. A. Muda, editors, *Advances in Nature and Biologically Inspired Computing*, volume 419, pages 261–270. Springer, Cham, 2016.
- [BWD19] L. Bergerhoff, J. Weickert, and Y. Dar. Algorithms for Piecewise Constant Signal Approximations. In *2019 27th European Signal Processing Conference (EUSIPCO 2019)*. IEEE, 2019.
- [Car14] A. S. Carasso. Compensating operators and stable backward in time marching in nonlinear parabolic equations. *GEM - International Journal on Geomathematics*, 5(1):1–16, April 2014.
- [Car16] A. S. Carasso. Stable explicit time marching in well-posed or ill-posed nonlinear parabolic equations. *Inverse Problems in Science and Engineering*, 24(8):1364–1384, 2016.
- [Car17] A. S. Carasso. Stabilized Richardson leapfrog scheme in explicit stepwise computation of forward or backward nonlinear parabolic equations. *Inverse Problems in Science and Engineering*, 25(12):1719–1742, 2017.
- [Cár18] Giovanni Marcelo Cárdenas. *Nonlocal Evolutions in Image Processing*. PhD thesis, Mathematical Image Analysis Group, Saarland University, Saarbrücken, Germany, 2018.

- 
- [CDM91] A. Colorni, M. Dorigo, and V. Maniezzo. Distributed optimization by ant colonies. In *Proceedings of the First European Conference On Artificial Life*, pages 134–142, Paris, France, December 1991.
- [CE54] P. J. Clark and F. C. Evans. Distance to Nearest Neighbor as a Measure of Spatial Relationships in Populations. *Ecology*, 35(4):445–453, October 1954.
- [CFTV10] J. A. Carrillo, M. Fornasier, G. Toscani, and F. Vecil. Particle, kinetic, and hydrodynamic models of swarming. In G. Naldi, L. Pareschi, and G. Toscani, editors, *Mathematical Modeling of Collective Behavior in Socio-Economic and Life Sciences*, Modeling and Simulation in Science, Engineering and Technology, pages 297–336. Birkhäuser, Boston, June 2010.
- [CHDB07] Y. L. Chuang, Y. R. Huang, M. R. D’Orsogna, and A. L. Bertozzi. Multi-Vehicle Flocking: Scalability of Cooperative Control Algorithms using Pairwise Potentials. In *2007 IEEE International Conference on Robotics and Automation, ICRA 2007, 10-14 April 2007, Roma, Italy*, pages 2292–2299, 2007.
- [CKJ<sup>+</sup>02] I. D. Couzin, J. Krause, R. James, G. D. Ruxton, and N. R. Franks. Collective Memory and Spatial Sorting in Animal Groups. *Journal of Theoretical Biology*, 218(1):1–11, September 2002.
- [CPL85] J. J. Clark, M. R. Palmer, and P. D. Lawrence. A transformation method for the reconstruction of functions from nonuniformly spaced samples. *IEEE Transactions on Acoustics, Speech, and Signal Processing*, 33(5):1151–1165, October 1985.
- [CS07] F. Cucker and S. Smale. Emergent Behavior in Flocks. *IEEE Transactions on Automatic Control*, 52(5):852–862, May 2007.
- [CSH78] A. Carasso, J. Sanderson, and J. Hyman. Digital Removal of Random Media Image Degradations by Solving the Diffusion Equation Backwards in Time. *SIAM Journal on Numerical Analysis*, 15(2):344–367, April 1978.
- [Dav11] O. Davydov. Algorithms and Error Bounds for Multivariate Piecewise Constant Approximation. In E. H. Georgoulis, A. Iske, and J. Levesley, editors, *Approximation Algorithms for Complex Systems*, pages 27–45, Berlin, 2011. Springer.
- [DB19] Y. Dar and A. M. Bruckstein. On High-Resolution Adaptive Sampling of Deterministic Signals. *Journal of Mathematical Imaging and Vision*, 61(7):944–966, 2019.
- [DCBC06] M. R. D’Orsogna, Y. L. Chuang, A. L. Bertozzi, and L. S. Chayes. Self-Propelled Particles with Soft-Core Interactions: Patterns, Stability, and Collapse. *Physical Review Letters*, 96:104302, March 2006.

- [DFG99] Q. Du, V. Faber, and M. D. Gunzburger. Centroidal Voronoi Tessellations: Applications and Algorithms. *SIAM Review*, 41(4):637–676, 1999.
- [DH72] R. O. Duda and P. E. Hart. Use of the Hough Transformation to Detect Lines and Curves in Pictures. *Communications of the ACM*, 15(1):11–15, January 1972.
- [FG87] W. Förstner and E. Gülch. A fast operator for detection and precise location of distinct points, corners and centres of circular features. In *Proc. ISPRS Intercommission Conference on Fast Processing of Photogrammetric Data*, pages 281–305, Interlaken, Switzerland, June 1987.
- [FS13] B. T. Fine and D. A. Shell. Unifying microscopic flocking motion models for virtual, robotic, and biological flock members. *Autonomous Robots*, 35(2-3):195–219, October 2013.
- [FXQ07] C.-L. Fu, X.-T. Xiong, and Z. Qian. Fourier regularization for a backward heat equation. *Journal of Mathematical Analysis and Applications*, 331(1):472–480, 2007.
- [Gab65] D. Gabor. Information theory in electron microscopy. *Laboratory Investigation*, 14:801–807, June 1965.
- [Gaz13] V. Gazi. On Lagrangian dynamics based modeling of swarm behavior. *Physica D: Nonlinear Phenomena*, 260:159–175, October 2013.
- [Ger31] S. Gerschgorin. Über die Abgrenzung der Eigenwerte einer Matrix. *Bulletin de l’Académie des Sciences de l’URSS. Classe des sciences mathématiques et naturelles*, pages 749–754, 1931.
- [GF07] V. Gazi and B. Fidan. Coordination and control of multi-agent dynamic systems: Models and approaches. In E. Sahin, W. M. Spears, and A. F. T. Winfield, editors, *Swarm Robotics*, volume 4433 of *Lecture Notes in Computer Science*, pages 71–102. Springer, Berlin, 2007.
- [GP03] V. Gazi and K. M. Passino. Stability Analysis of Swarms. *IEEE Transactions on Automatic Control*, 48(4):692–697, April 2003.
- [GP04] V. Gazi and K. M. Passino. Stability analysis of social foraging swarms. *IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics*, 34(1):539–557, February 2004.
- [GPS01] H. Goldstein, C. P. Poole, and J. L. Safko. *Classical Mechanics*. Pearson, third edition, 2001.

- 
- [Gro19] T. H. Gronwall. Note on the Derivatives with Respect to a Parameter of the Solutions of a System of Differential Equations. *Annals of Mathematics*, 20(4):292–296, July 1919.
- [GSZ02] G. Gilboa, N. A. Sochen, and Y. Y. Zeevi. Forward-and-backward diffusion processes for adaptive image enhancement and denoising. *IEEE Transactions on Image Processing*, 11(7):689–703, 2002.
- [GW08] R. C. Gonzalez and R. E. Woods. *Digital Image Processing*. Prentice-Hall, Upper Saddle River, NJ, USA, third edition, 2008.
- [Had02] J. Hadamard. Sur les problèmes aux dérivées partielles et leur signification physique. *Princeton University Bulletin*, 13:49–52, 1902.
- [HD09] D. N. Hào and N. V. Duc. Stability results for the heat equation backward in time. *Journal of Mathematical Analysis and Applications*, 353(2):627–641, May 2009.
- [HD11] D. N. Hào and N. V. Duc. Stability results for backward parabolic equations with time-dependent coefficients. *Inverse Problems*, 27(2):025003, January 2011.
- [Hen19] C. Henry. SpaceX submits paperwork for 30,000 more Starlink satellites. <https://spacenews.com/spacex-submits-paperwork-for-30000-more-starlink-satellites/>, 2019. Last visited March 7, 2020.
- [HKZ87] R. A. Hummel, B. B. Kimia, and S. W. Zucker. Deblurring Gaussian blur. *Computer Vision, Graphics, and Image Processing*, 38(1):66–80, April 1987.
- [Hor68] K. Horiuchi. Sampling Principle for Continuous Signals with Time-Varying Bands. *Information and Control*, 13(1):53–61, July 1968.
- [HTF09] T. Hastie, R. Tibshirani, and J. H. Friedman. *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*. Springer Series in Statistics. Springer, second edition, 2009.
- [Int99] International Electrotechnical Commission. IEC 61966-2-1:1999 - Multimedia systems and equipment - Colour measurement and management - Part 2-1: Colour management - Default RGB colour space - sRGB, 1999. <https://webstore.iec.ch/publication/6169>.
- [Jer77] A. J. Jerri. The Shannon sampling theorem – Its various extensions and applications: A tutorial review. *Proceedings of the IEEE*, 65(11):1565–1596, November 1977.
- [Joh55] F. John. Numerical solution of the equation of heat conduction for preceding times. *Annali di Matematica Pura ed Applicata*, 40(1):129–142, December 1955.
-

- [Jon06] R. Jones. *Myths and Legends of Britain and Ireland*. New Holland, 2006.
- [KBPW18] L. Karos, P. Bheed, P. Peter, and J. Weickert. Optimising Data for Exemplar-Based Inpainting. In J. Blanc-Talon, D. Helbert, W. Philips, D. C. Popescu, and P. Scheunders, editors, *Advanced Concepts for Intelligent Vision Systems*, volume 11182 of *Lecture Notes in Computer Science*, pages 547–558, Cham, 2018. Springer.
- [KE95] J. Kennedy and R. Eberhart. Particle swarm optimization. In *Proceedings of ICNN'95 - International Conference on Neural Networks*, volume 4, pages 1942–1948, Perth, WA, Australia, November 1995. IEEE.
- [KHD13] U. Kirchmaier, S. Hawe, and K. Diepold. A Swarm Intelligence inspired algorithm for contour detection in images. *Applied Soft Computing*, 13(6):3118–3129, June 2013.
- [KJ55] L. S. G. Kovásznyai and H. M. Joseph. Image Processing. *Proceedings of the IRE*, 43(5):560–570, May 1955.
- [Kod] Kodak Lossless True Color Image Suite. <http://www.r0k.us/graphics/kodak/>. Last visited August 31, 2018.
- [KW02] S.M. Kirkup and M. Wadsworth. Solution of inverse diffusion problems by operator-splitting methods. *Applied Mathematical Modelling*, 26(10):1003–1018, 2002.
- [LeV07] R. J. LeVeque. *Finite Difference Methods for Ordinary and Partial Differential Equations*. SIAM, Philadelphia, 2007.
- [LFB94] M. Lindenbaum, M. Fischer, and A. M. Bruckstein. On Gabor's contribution to image enhancement. *Pattern Recognition*, 27(1):1–8, January 1994.
- [LJ11] M. A. Little and N. S. Jones. Generalized methods and solvers for noise removal from piecewise constant signals. I. Background theory. *Proceedings of the Royal Society A*, 467(2135):3088–3114, June 2011.
- [Llo82] S. P. Lloyd. Least squares quantization in PCM. *IEEE Transactions on Information Theory*, 28(2):129–136, 1982.
- [LPZ12] F. Li, L. Pi, and T. Zeng. Explicit coherence enhancing filter with spatial adaptive elliptical kernel. *IEEE Signal Processing Letters*, 19(9):555–558, September 2012.
- [LT99] J. Liu and Y. Y. Tang. Adaptive image segmentation with distributed behavior-based agents. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 21(6):544–551, June 1999.

- 
- [Lya92] A. M. Lyapunov. The general problem of the stability of motion. *International Journal of Control*, 55(3):531–534, 1992.
- [MA09] G. R. Murthy and N. Ahuja. Non-uniform sampling: A novel approach. In *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing, ICASSP 2009, 19-24 April 2009, Taipei, Taiwan*, pages 3229–3232. IEEE, April 2009.
- [Mac67] J. MacQueen. Some methods for classification and analysis of multivariate observations. In *Proceedings of the Fifth Berkeley Symposium on Mathematical Statistics and Probability, Volume 1: Statistics*, pages 281–297, Berkeley, California, 1967. University of California Press.
- [Map18] Maplesoft. *Maple 2018.2*. Waterloo Maple Inc., November 2018. <https://www.maplesoft.com/>.
- [MB04] D. Marthaler and A. L. Bertozzi. Tracking Environmental Level Sets with Autonomous Vehicles. In *Recent Developments in Cooperative Control and Optimization*, volume 3 of *Cooperative Systems*, pages 317–332. Springer, New York, 2004.
- [MHW<sup>+</sup>12] 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. In A. M. Bruckstein, B. M. ter Haar Romeny, A. M. Bronstein, and M. M. Bronstein, editors, *Scale Space and Variational Methods in Computer Vision*, volume 6667 of *Lecture Notes in Computer Science*, pages 26–37. Springer, 2012.
- [MWR96] B. A. Mair, D. C. Wilson, and Z. Reti. Deblurring the Discrete Gaussian Blur. In *Proceedings of the Workshop on Mathematical Methods in Biomedical Image Analysis*, pages 273–277, Los Alamitos, June 1996. IEEE.
- [Nes04] Y. Nesterov. *Introductory Lectures On Convex Optimization*. Springer, New York, 2004.
- [NM03] S. K. Naik and C. A. Murthy. Hue-Preserving Color Image Enhancement Without Gamut Problem. *IEEE Transactions on Image Processing*, 12(12):1591–1598, December 2003.
- [NS14a] M. Nikolova and G. Steidl. Fast Hue and Range Preserving Histogram Specification: Theory and New Algorithms for Color Image Enhancement. *IEEE Transactions on Image Processing*, 23(9):4087–4100, September 2014.
- [NS14b] M. Nikolova and G. Steidl. Fast Ordering Algorithm for Exact Histogram Specification. *IEEE Transactions on Image Processing*, 23(12):5274–5283, December 2014.
-

- [NW99] J. Nocedal and S. J. Wright. *Numerical Optimization*. Springer, New York, 1999.
- [NWC13] M. Nikolova, Y.-W. Wen, and R. Chan. Exact Histogram Specification for Digital Images Using a Variational Approach. *Journal of Mathematical Imaging and Vision*, 46(3):309–325, July 2013.
- [OR91] S. Osher and L. Rudin. Shocks and other nonlinear filtering applied to image processing. In A. G. Tescher, editor, *Applications of Digital Image Processing XIV*, volume 1567 of *Proceedings of SPIE*, pages 414–431. SPIE Press, Bellingham, 1991.
- [OSA16] J. A. Ojo, I. D. Solomon, and S. A. Adeniran. Colour-Preserving Contrast Enhancement Algorithm for Images. In L. Chen, S. Kapoor, and R. Bhatia, editors, *Emerging Trends and Advanced Technologies for Computational Intelligence: Extended and Selected Results from the Science and Information Conference 2015*, pages 207–222, Cham, 2016. Springer.
- [PAA<sup>+</sup>87] S. M. Pizer, E. P. Amburn, J. D. Austin, R. Cromartie, A. Geselowitz, T. Greer, B. M. ter Haar Romeny, J. B. Zimmerman, and K. Zuiderveld. Adaptive histogram equalization and its variations. *Computer Vision, Graphics, and Image Processing*, 39(3):355–368, 1987.
- [PAB<sup>+</sup>17] F. Pierre, J.-F. Aujol, A. Bugeau, G. Steidl, and V.-T. Ta. Variational Contrast Enhancement of Gray-Scale and RGB Images. *Journal of Mathematical Imaging and Vision*, 57(1):99–116, January 2017.
- [Per01] L. Perko. *Differential Equations and Dynamical Systems*. Number 7 in Texts in Applied Mathematics. Springer, third edition, 2001.
- [PM90] P. Perona and J. Malik. Scale space and edge detection using anisotropic diffusion. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 12:629–639, 1990.
- [PM92] W. B. Pennebaker and J. L. Mitchell. *JPEG: Still image data compression standard*. Springer Science & Business Media, 1992.
- [Pol64] B. T. Polyak. Some methods of speeding up the convergence of iteration methods. *USSR Computational Mathematics and Mathematical Physics*, 4(5):1–17, 1964.
- [Pol87] B. T. Polyak. *Introduction to Optimization*. Optimization Software, New York, 1987.
- [Pra01] W. K. Pratt. *Digital Image Processing*. Wiley & Sons, New York, third edition, 2001.



- 
- [PWK00] I. Pollak, A. S. Willsky, and H. Krim. Image segmentation and edge enhancement with stabilized inverse diffusion equations. *IEEE Transactions on Image Processing*, 9(2):256–266, February 2000.
- [Rey87] C. W. Reynolds. Flocks, Herds and Schools: A Distributed Behavioral Model. *ACM SIGGRAPH Computer Graphics*, 21(4):25–34, August 1987.
- [Rey07] C. W. Reynolds. Boids (Flocks, Herds, and Schools: a Distributed Behavioral Model). <http://www.red3d.com/cwr/boids/>, July 2007. Last visited April 5, 2020.
- [RS09] M. Rubenstein and W. M. Shen. Scalable self-assembly and self-repair in a collective of robots. In *2009 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 1484–1489, St. Louis, Missouri, USA, October 2009. IEEE.
- [SACM96] M. Stokes, M. Anderson, S. Chandrasekar, and R. Motta. A Standard Default Color Space for the Internet - sRGB (Version 1.10). <https://www.w3.org/Graphics/Color/sRGB>, November 1996. Last visited August 31, 2018.
- [SAF<sup>+</sup>14] G. Simone, G. Audino, I. Farup, F. Albrechtsen, and A. Rizzi. Termite Retinex: a new implementation based on a colony of intelligent agents. *Journal of Electronic Imaging*, 23(1):013006, 2014.
- [Sau98] D. Saupe. Optimal piecewise linear image coding. In S. A. Rajala and M. Rabbani, editors, *Visual Communications and Image Processing '98*, volume 3309, pages 747–760. International Society for Optics and Photonics, SPIE, 1998.
- [SBHJ00] G. L. Schussman, M. Bertram, B. Hamann, and K. I. Joy. Hierarchical Data Representations Based on Planar Voronoi Diagrams. In *Proceedings of the 2000 Joint Eurographics and IEEE TCVG Symposium on Visualization, VisSym 2000, Amsterdam, The Netherlands, May 29-30, 2000*, pages 63–72, 2000.
- [SC97] G. Sapiro and V. Caselles. Histogram modification via differential equations. *Journal of Differential Equations*, 135:238–268, 1997.
- [Sec02] A. Secord. Weighted Voronoi stippling. In A. Finkelstein, editor, *Proceedings of the Second International Symposium on Non-Photorealistic Animation and Rendering, NPAR 2002, Annecy, France, June 3-5, 2002*, pages 37–43. ACM, 2002.
- [SGBW10] C. Schmaltz, P. Gwosdek, A. Bruhn, and J. Weickert. Electrostatic Halftoning. *Computer Graphics Forum*, 29(8):2313–2327, December 2010.
-

- [Sig15] Signal and Image Processing Institute of the University of Southern California. The USC-SIPI image database. <http://sipi.usc.edu/database/>, 2015. Last visited August 16, 2015.
- [SK04] H.-R. Schwarz and N. Köckler. *Numerische Mathematik*. Vieweg+Teubner, Wiesbaden, eighth edition, 2004.
- [SKB98] A. Steiner, R. Kimmel, and A. M. Bruckstein. Planar Shape Enhancement and Exaggeration. *Graphical Models and Image Processing*, 60(2):112–124, March 1998.
- [SP15] D. Shishika and D. A. Paley. Lyapunov stability analysis of a mosquito-inspired swarm model. In *2015 54th IEEE Conference on Decision and Control (CDC)*, pages 482–488, Osaka, Japan, December 2015. IEEE.
- [Sum05] D. J. T. Sumpter. The principles of collective animal behaviour. *Philosophical Transactions of the Royal Society B: Biological Sciences*, 361(1465):5–22, November 2005.
- [SZ98] N. A. Sochen and Y. Y. Zeevi. Resolution enhancement of colored images by inverse diffusion processes. In *Proc. 1998 IEEE International Conference on Acoustics, Speech and Signal Processing*, pages 2853–2856, Seattle, WA, May 1998.
- [TC17] Q.-C. Tian and L. D. Cohen. Color Consistency for Photo Collections Without Gamut Problems. In L. Amsaleg, G. Guðmundsson, C. Gurrin, B. Jónsson, and S. Satoh, editors, *MultiMedia Modeling*, volume 10132 of *Lecture Notes in Computer Science*, pages 90–101, Cham, January 2017. Springer.
- [Tea20] The GIMP Development Team. *GNU Image Manipulation Program 2.10.16*. February 2020. <https://www.gimp.org/>.
- [tFd+94] B. M. ter Haar Romeny, L. M. J. Florack, M. de Swart, J. Wilting, and M. A. Viergever. Deblurring Gaussian blur. In F. L. Bookstein, J. S. Duncan, N. Lange, and D. C. Wilson, editors, *Mathematical Methods in Medical Imaging III*, volume 2299 of *Proceedings of SPIE*, pages 139–148. SPIE Press, Bellingham, July 1994.
- [TOD11] F. Ternat, O. Orellana, and P. Daripa. Two stable methods with numerical experiments for solving the backward heat equation. *Applied Numerical Mathematics*, 61(2):266–284, February 2011.
- [TOW19] J. A. Tómasson, P. Ochs, and J. Weickert. AFSI: Adaptive Restart for Fast Semi-Iterative Schemes for Convex Optimisation. In T. Brox, A. Bruhn, and M. Fritz, editors, *Pattern Recognition*, volume 11269 of *Lecture Notes in Computer Science*, pages 669–681, Cham, 2019. Springer.

- 
- [TS96] U. Tautenhahn and T. Schröter. On Optimal Regularization Methods for the Backward Heat Equation. *Zeitschrift für Analysis und ihre Anwendungen*, 15(2):475–493, 1996.
- [TS05] I. Triandaf and I. B. Schwartz. A collective motion algorithm for tracking time-dependent boundaries. *Mathematics and Computers in Simulation*, 70(4):187–202, December 2005.
- [UNE20] UNESCO World Heritage Centre. Giant’s Causeway and Causeway Coast. <https://whc.unesco.org/en/list/369>, 2020. Last visited March 5, 2020.
- [VB17] S. M. Vovk and V. F. Borulko. Determination of amplitude levels of the piecewise constant signal by using polynomial approximation. *Radioelectronics and Communications Systems*, 60(3):113–122, March 2017.
- [VCBJ+95] T. Vicsek, A. Czirók, E. Ben-Jacob, I. Cohen, and O. Shochet. Novel Type of Phase Transition in a System of Self-Driven Particles. *Physical Review Letters*, 75:1226–1229, August 1995.
- [VZ12] T. Vicsek and A. Zafeiris. Collective motion. *Physics Reports*, 517(3–4):71–140, August 2012.
- [Wei94] J. Weickert. Anisotropic diffusion filters for image processing based quality control. In A. Fasano and M. Primicerio, editors, *Proc. Seventh European Conference on Mathematics in Industry*, pages 355–362. Teubner, Stuttgart, 1994.
- [Wei98] J. Weickert. *Anisotropic Diffusion in Image Processing*. Teubner, Stuttgart, 1998.
- [Wei99] J. Weickert. Coherence-Enhancing Diffusion Filtering. *International Journal of Computer Vision*, 31(2/3):111–127, April 1999.
- [Wei03] J. Weickert. Coherence-Enhancing Shock Filters. In B. Michaelis and G. Krell, editors, *Pattern Recognition*, volume 2781 of *Lecture Notes in Computer Science*, pages 1–8. Springer, Berlin, 2003.
- [WGW09] M. Welk, G. Gilboa, and J. Weickert. Theoretical foundations for discrete forward-and-backward diffusion filtering. In X-C. Tai, K. Mørken, M. Lysaker, and K.-A. Lie, editors, *Scale Space and Variational Methods in Computer Vision*, volume 5567 of *Lecture Notes in Computer Science*, pages 527–538. Springer, Berlin, 2009.
- [WO07] D. Wei and A. V. Oppenheim. Sampling based on local bandwidth. In *2007 Conference Record of the Forty-First Asilomar Conference on Signals, Systems and Computers*, pages 1103–1107, November 2007.
-

- [WWG18] M. Welk, J. Weickert, and G. Gilboa. A Discrete Theory and Efficient Algorithms for Forward-and-Backward Diffusion Filtering. *Journal of Mathematical Imaging and Vision*, 60(9):1399–1426, 2018.
- [WWS06] M. Welk, J. Weickert, and G. Steidl. From Tensor-Driven Diffusion to Anisotropic Wavelet Shrinkage. In H. Bischof, A. Leonardis, and A. Pinz, editors, *Computer Vision – ECCV 2006, Part I*, volume 3951 of *Lecture Notes in Computer Science*, pages 391–403. Springer, Berlin, 2006.
- [Yan10a] X.-S. Yang. *Nature Inspired Cooperative Strategies for Optimization (NICSO 2010)*, chapter A New Metaheuristic Bat-Inspired Algorithm, pages 65–74. Springer, Berlin, Heidelberg, 2010.
- [Yan10b] X.-S. Yang. *Nature-Inspired Metaheuristic Algorithms*, chapter Firefly Algorithm, pages 81–96. Luniver Press, second edition, 2010.
- [YBEM10] C. A. Yates, R. E. Baker, R. Erban, and P. K. Maini. Refining self-propelled particle models for collective behaviour. *Canadian Applied Maths Quarterly*, 18(3):299–350, 2010.
- [ZCR13] M. Zambrano-Bigiarini, M. Clerc, and R. Rojas-Mujica. Standard Particle Swarm Optimisation 2011 at CEC-2013: A baseline for future PSO improvements. In *Proceedings of the IEEE Congress on Evolutionary Computation, CEC 2013, Cancun, Mexico, June 20-23, 2013*, pages 2337–2344, Cancun, Mexico, June 2013. IEEE.
- [ZM11] Z. Zhao and Z. Meng. A modified Tikhonov regularization method for a backward heat equation. *Inverse Problems in Science and Engineering*, 19(8):1175–1182, 2011.
- [Zui94] K. Zuiderveld. Contrast Limited Adaptive Histogram Equalization. In P. S. Heckbert, editor, *Graphics Gems IV*, pages 474–485. Academic Press Professional, Inc., San Diego, CA, USA, 1994.

---

# Appendix B

## Own Publications

### Journal Papers

1. L. Bergerhoff, M. Cárdenas, J. Weickert, and M. Welk. Stable Backward Diffusion Models that Minimise Convex Energies. *Journal of Mathematical Imaging and Vision*, 62(6):941–960, July 2020. Springer.

### Conference Papers

2. L. Bergerhoff and J. Weickert. Modelling Image Processing with Discrete First-Order Swarms. In N. Pillay, P. A. Engelbrecht, A. Abraham, C. M. du Plessis, V. Snášel, and K. A. Muda, editors, *Advances in Nature and Biologically Inspired Computing*, volume 419 of *Advances in Intelligent Systems and Computing*, pages 261–270, Cham, 2016. Springer.
3. L. Bergerhoff, M. Cárdenas, J. Weickert, and M. Welk. Modelling Stable Backward Diffusion and Repulsive Swarms with Convex Energies and Range Constraints. In M. Pelillo and E. Hancock, editors, *Energy Minimization Methods in Computer Vision and Pattern Recognition*, volume 10746 of *Lecture Notes in Computer Science*, pages 409–423, Cham, March 2018. Springer.
4. L. Bergerhoff, J. Weickert, and Y. Dar. Algorithms for Piecewise Constant Signal Approximations. *2019 27th European Signal Processing Conference (EUSIPCO 2019)*, 2019. IEEE.

### Technical Reports

5. L. Bergerhoff, J. Weickert, and Y. Dar. Algorithms for Piecewise Constant Signal Approximations. arXiv:1903.01320v3 [eess.SP], June 2019.
6. L. Bergerhoff, M. Cárdenas, J. Weickert, and M. Welk. Stable Backward Diffusion Models that Minimise Convex Energies. arXiv:1903.03491v2 [math.NA], June 2020.



---

# Appendix C

## Glossary

<b>AFSI</b> . . . . .	adaptive fast semi-iterative
<b>BTLS</b> . . . . .	backtracking line search
<b>DB</b> . . . . .	Dar–Bruckstein method
<b>GD</b> . . . . .	gradient descent
<b>HB</b> . . . . .	heavy ball method
<b>JPEG</b> . . . . .	Joint Photographic Experts Group
<b>MSE</b> . . . . .	mean squared error
<b>ODE</b> . . . . .	ordinary differential equation
<b>PSO</b> . . . . .	Particle Swarm Optimisation
<b>SPSO</b> . . . . .	Standard Particle Optimisation 2011
<b>SPSO-2011</b> . . . . .	Standard Particle Swarm Optimisation 2011
<b>US</b> . . . . .	uniform (re-)sampling





---

# Appendix D

## List of Symbols

$\mathbf{1}$	$n$ -dimensional column vector containing ones
$a$	scalar value, non-linearity parameter, domain boundary
$a_{i,j}$	element of the real-valued matrix $\mathbf{A}$ in row $i$ and column $j$
$\mathbf{A}$	real-valued matrix
$\mathbf{A}_1$	symmetric real-valued $4 \times 4$ matrix, defined in (2.11)
$\mathbf{A}_k$	real-valued matrix
$b$	scalar value, directional offset parameter, domain boundary
$B$	barrier function
$c$	spatial scale, repulsion parameter, introduced in (3.12)
$c_1$	cognitive acceleration coefficient
$c_2$	social acceleration coefficient
$c_r, c_g, c_b$	scalar values, introduced in (4.73)
$c_u$	scalar value, frequency of grey value $g$
$C^k(Q)$	class of $k$ -times differentiable functions on $Q$
$C_L^{k,p}(Q)$	class of $k$ -times differentiable functions on $Q$ whose $p$ -th derivative is Lipschitz continuous with constant $L$
$C$	positive constant
$\mathbb{C}$	set of complex numbers
$d$	minimum desired decrease parameter of backtracking line search
$d_1$	scalar value, radius
$d_2$	scalar value, radius

$D$	positive constant
$e_i$	squared error of the $i$ -th interval, defined in (5.60)
$E$	energy function
$E_f$	energy function for approximation of the signal $f$
$E_i$	total potential energy of particle $i$ , defined in (3.5)
$E'$	first-order derivative of $E$
$E''$	second-order derivative of $E$
$\nabla E$	energy gradient
$D^2 E$	Hessian of the energy function $E$
$f$	real-valued function
$f_1, f_2, f_3$	real-valued functions defined in (5.45), (5.56), and (5.85)
$f_i$	$i$ -th element of the column vector $\mathbf{f}$ , or example function defined in (2.4)
$f_\sigma$	image $f$ , pre-smoothed with a Gaussian kernel with standard deviation $\sigma$
$f'$	derivative of the function $f$
$f'_i$	derivative of the function $f$ in the $i$ -th interval
$\hat{f}$	intensity value, introduced in (4.73)
$\nabla \tilde{f}_i$	local dominant gradient orientation in pixel $i$
$\nabla^\perp \tilde{f}_i$	local dominant tangent orientation in pixel $i$ , defined in (3.31)
$\mathbf{f}$	column vector or vector-valued function
$\mathbf{f}_1$	vector containing sampled one-dimensional signal data
$F$	integral of the function $f$ , defined in (5.4)
$F_1, F_2$	integrals of the functions $f_1$ and $f_2$ , defined in (5.47) and (5.58)
$\mathbf{F}$	matrix containing samples of a two-dimensional signal
$\mathbf{F}_2$	sampled two-dimensional signal data, defined in (2.8)
$g$	diffusivity function in Chapter 4, squared function $f$ in Chapter 5
$g_1, g_2$	real-valued functions defined in (5.46) and (5.57)
$\mathbf{g}_1$	sampled one-dimensional signal data, defined in (2.7)

---

$\mathbf{g}_i^k$ . . . . .	centre of gravity, defined in (2.34)
$\bar{\mathbf{g}}^k$ . . . . .	best minimiser found up to iteration $k$ , defined in (2.31)
$G$ . . . . .	integral of the function $g$ , defined in (5.5)
$G_1, G_2$ . . . . .	integrals of the functions $g_1$ and $g_2$ , defined in (5.48) and (5.59)
$\mathbf{G}$ . . . . .	matrix containing samples of a two-dimensional signal
$\mathbf{G}_2$ . . . . .	sampled two-dimensional signal data, defined in (2.8)
$h$ . . . . .	grid size, sampling distance
$h[f]$ . . . . .	histogram of the discrete grey scale image $f$
$h[f](g)$ . . . . .	frequency of the grey value $g$ in the discrete grey scale image $f$
$h_i$ . . . . .	width of the $i$ -th interval
$\mathbf{H}_k$ . . . . .	Hankel matrix
$\mathcal{H}_i(\mathbf{g}_i^k,  \mathbf{g}_i^k - \mathbf{x}_i^k )$ . . . . .	random point from the hypersphere around $\mathbf{g}_i^k$ with radius $ \mathbf{g}_i^k - \mathbf{x}_i^k $
$i$ . . . . .	index variable
$j$ . . . . .	index variable
$J_1^i, J_2^i, J_3^i$ . . . . .	sets of natural numbers
$k$ . . . . .	time level, iteration number, or index
$k_1, k_2, k_3, k_4$ . . . . .	kernel functions, defined in (3.11)-(3.14)
$k_a$ . . . . .	kernel function steering attractive behaviour
$k_r$ . . . . .	kernel function steering repulsive behaviour
$K_i$ . . . . .	$i$ -th Gershgorin disc, defined in (2.10)
$\mathbf{l}_i^k$ . . . . .	position between the previous best position among the neighbours of particle $i$ and its current position in iteration $k$ , defined in (2.33)
$\bar{\mathbf{l}}_i^k$ . . . . .	previous best position among the neighbours of particle $i$ in iteration $k$
$\ell$ . . . . .	index variable
$\ell_i$ . . . . .	left boundary position, defined in (5.36)
$L$ . . . . .	Lipschitz constant
$L_{f_2}$ . . . . .	Lipschitz estimate of function $f_2$
$L_{\max}$ . . . . .	upper bound for the Lipschitz constant $L$

---

$L_\Phi$ . . . . .	Lipschitz constant of the flux function $\Phi$
$\tilde{L}$ . . . . .	Lipschitz estimate
$m$ . . . . .	dimensionality, length, number of samples, real-valued constant in Chapter 5
$m_i$ . . . . .	mass of particle $i$ in Chapter 3, parameter of a linear function, defined in (5.51)
$M$ . . . . .	real-valued constant in Chapter 5
$n$ . . . . .	dimensionality, length, number of samples, scalar value
$n_{neigh}$ . . . . .	number of neighbours
$\mathbf{n}_{i,j}$ . . . . .	normal vector pointing from pixel $i$ to pixel $j$
$N$ . . . . .	number of samples
$\mathcal{N}_i$ . . . . .	disk-shaped neighbourhood of pixel $i$ in the image plane
$\mathcal{N}_{i,\delta}(t)$ . . . . .	disk-shaped neighbourhood of the $i$ -th agent with radius $\delta$ at time $t$ , defined in (3.4)
$\mathcal{N}_{i,\delta}^k$ . . . . .	disk-shaped neighbourhood of the $i$ -th agent with radius $\delta$ in iteration $k$
$\mathbb{N}$ . . . . .	set of natural numbers
$\mathcal{O}$ . . . . .	Landau symbol
$p$ . . . . .	index variable
$p_i$ . . . . .	real-valued position, defined in (5.34)
$\mathbf{p}_1$ . . . . .	vector containing one-dimensional position data
$\mathbf{p}^k$ . . . . .	descent direction in iteration $k$
$\mathbf{p}_i^k$ . . . . .	position between the previous best position of particle $i$ and its current position in iteration $k$ , defined in (2.32)
$\bar{\mathbf{p}}_i^k$ . . . . .	previous best position of particle $i$ in iteration $k$ , defined in (2.30)
$P$ . . . . .	number of samples
$q$ . . . . .	scalar value, number of quantisation levels
$Q$ . . . . .	subset of $\mathbb{R}^n$
$r_i$ . . . . .	$i$ -th Gershgorin radius, defined in (2.10), right boundary position, defined in (5.36)
$\mathbb{R}$ . . . . .	set of real numbers

---

$s$ . . . . .	scalar value
$S$ . . . . .	swarm, subset of $\mathbb{N}$ , defined in (3.1)
$\mathcal{S}$ . . . . .	segment contribution, defined in (5.13)
$t$ . . . . .	scalar value, time value
$t_a$ . . . . .	threshold for accumulation scores in Hough space
$t_g$ . . . . .	threshold for gradient magnitude
$t_i$ . . . . .	parameter of a linear function, defined in (5.51)
$T_{opt}$ . . . . .	threshold, defined in (5.67)
$\mathbf{T}_k$ . . . . .	Toeplitz matrix
$u$ . . . . .	scalar value, grey value, scalar-valued function as used in Chapter 4 and 5
$u_c$ . . . . .	piecewise constant approximation function
$u_i$ . . . . .	grey value of pixel $i$
$u_i(t)$ . . . . .	grey value of pixel $i$ at time $t$
$u_t$ . . . . .	abbreviation of $\partial_t u$
$u_x$ . . . . .	abbreviation of $\partial_x u$
$u_\ell$ . . . . .	piecewise linear approximation function
$\dot{u}_i$ . . . . .	grey value change of pixel $i$ over time
$\dot{u}_i^k$ . . . . .	grey value change of pixel $i$ in iteration $k$
$\mathbf{u}$ . . . . .	column vector or vector-valued function
$\mathbf{u}_1^k, \mathbf{u}_2^k$ . . . . .	independent and uniformly distributed random vectors in iteration $k$ with components in $[0, 1]$
$\mathbf{u}(t)$ . . . . .	state of dynamical system at time $t$
$\mathbf{u}^*$ . . . . .	steady state of the dynamical system $\mathbf{u}(t)$ , grey value distribution
$U$ . . . . .	potential energy, potential function
$-\nabla_{\mathbf{x}_i} U$ . . . . .	potential force acting on individual $i$
$v_i$ . . . . .	state / position of the $i$ -th element
$v_i^k$ . . . . .	state / position of the $i$ -th element in iteration $k$
$v_i^*$ . . . . .	state / position of the $i$ -th element in the steady state
$\mathbf{v}$ . . . . .	state / position vector

---

$\mathbf{v}_0$	initial state / position vector
$\mathbf{v}_i$	state / position vector of the $i$ -th element
$\mathbf{v}(t)$	state of dynamical system at time $t$
$\dot{\mathbf{v}}(t)$	time derivative of the dynamical system $\mathbf{v}(t)$
$\mathbf{v}^*$	steady state of the dynamical system $\mathbf{v}(t)$
$V$	Lyapunov function
$w$	non-negative weighting function, introduced in (3.9)
$w_{i,j}$	scalar value, non-negative weight
$\tilde{w}_{i,j}$	scalar value, non-negative weight
$W$	penaliser function
$\mathbf{W}$	non-negative weight matrix
$\tilde{\mathbf{W}}$	non-negative weight matrix, submatrix of $\mathbf{W}$
$x$	state / position
$x_i^k$	state / position after $k$ iterations
$x_i^*$	optimal state / position
$\mathbf{x}$	state / position vector
$\mathbf{x}^k$	state / position vector after $k$ iterations
$\mathbf{x}_0$	initial state / position vector
$\mathbf{x}_i$	state / position vector of the $i$ -th element
$\mathbf{x}_i^k$	state / position vector of the $i$ -th element in iteration $k$
$\mathbf{x}(t)$	state of dynamical system at time $t$
$\mathbf{x}_i(t)$	state / position vector of the $i$ -th element at time $t$
$\dot{\mathbf{x}}(t)$	time derivative of the dynamical system $\mathbf{x}(t)$
$\mathbf{x}^*$	steady state of the dynamical system $\mathbf{x}(t)$
$X$	subset of $\mathbb{R}^+$
$y_1, y_2$	real-valued positions
$\mathbf{y}$	state / position vector
$z$	real-valued position
$\alpha$	step size used in first-order minimisation methods
$\alpha_k$	extrapolation parameter of AFSI
$\alpha^*$	optimal step size, defined in (2.19)

---

$\bar{\alpha}$ . . . . .	initial step size
$\beta$ . . . . .	inertia parameter, defined in (2.22)
$\gamma$ . . . . .	scalar value, non-increasing weighting function with compact support in Chapter 4.3.3
$\gamma_1, \gamma_2$ . . . . .	non-increasing weighting functions defined in (4.71) and (4.72)
$\delta$ . . . . .	scalar value, radius
$\delta_{i,j}$ . . . . .	Kronecker symbol
$\varepsilon$ . . . . .	small and positive constant
$\eta_i$ . . . . .	real-valued position
$\theta, \tilde{\theta}$ . . . . .	scalar value, angle
$\theta_i$ . . . . .	real-valued position
$\kappa$ . . . . .	positive constant
$\lambda$ . . . . .	scalar value
$\lambda_i$ . . . . .	$i$ -th eigenvalue
$\xi_i$ . . . . .	real-valued position
$\rho$ . . . . .	spectral radius, defined in (2.9), contraction parameter of backtracking line search, radius, distance to origin
$\tilde{\rho}$ . . . . .	radius, distance to origin
$\varrho$ . . . . .	positive scalar value, radius
$\sigma$ . . . . .	standard deviation of a Gaussian, permutation in Chapter 4
$\Phi$ . . . . .	flux function
$\Phi_{a,n}$ . . . . .	flux function defined in Table 4.1
$\Phi_-$ . . . . .	left-sided derivative of the flux function
$\Phi'$ . . . . .	derivative of the flux function $\Phi$
$\Psi$ . . . . .	energy function, penaliser function
$\Psi_{a,n}$ . . . . .	penaliser function defined in Table 4.1
$\Psi'$ . . . . .	derivative of the energy function $\Psi$ , diffusivity

---

$\Psi'_{a,n}$ . . . . .	derivative of penaliser function defined in Table 4.1
$\tilde{\Psi}$ . . . . .	potential function
$\omega$ . . . . .	step size parameter for AFSI, inertia parameter in the context of SPSO-2011
$\Omega$ . . . . .	interval of the real axis
$\odot$ . . . . .	element-wise vector multiplication operator
$\nabla$ . . . . .	gradient operator
$\nabla_{\mathbf{v}}$ . . . . .	gradient w.r.t. $\mathbf{v}$
$\partial_t$ . . . . .	first order partial derivative w.r.t. $t$
$\partial_{v_i}$ . . . . .	first order partial derivative w.r.t. $v_i$
$\partial_{v_i v_j}$ . . . . .	second order partial derivative w.r.t. $v_i$ and $v_j$
$\partial_x$ . . . . .	first order partial derivative w.r.t. $x$
$\cup$ . . . . .	union of sets operator
$ \cdot $ . . . . .	Euclidean norm
$\ \cdot\ _p$ . . . . .	p-norm, $\ell_p$ -norm
$\ \cdot\ _F$ . . . . .	Frobenius norm



---

---