UNIVERSITÄT ZU LÜBECK
INSTITUTE OF MATHEMATICS AND IMAGE COMPUTING

Non-smooth Higher-order Optimization on Manifolds

Master thesis

im Rahmen des Studiengangs
Scientific Computing und Applied Mathematics
der Technische Universität Berlin, Technische Universiteit Delft

Vorgelegt von
Willem Diepeveen

Ausgegeben und betreut von
Prof. Dr. Jan Lellmann
Institute of Mathematics and Image Computing

September 12, 2020

MIC

IM FOCUS DAS LEBEN

Kurzfassung

In dieser Arbeit wird ein Optimierungsverfahren höherer Ordnung zur Lösung nichtglatter Variationsprobleme auf Riemannschen Mannigfaltigkeiten vorgestellt. Dazu wird die Riemannsche Semismooth Newton (RSSN)-Methode auf ein nichtglattes nichtlineares Optimalitätssystem angewandt, das auf kürzlich veröffentlichten Arbeiten basiert. Insbesondere wird ein ein neues lokales Konvergenzresultat für eine inexakte Variante des Riemannschen Semismooth Newton-Verfahrens gezeigt. Die experimentellen Ergebnisse zeigen die Leistungsfähigkeit des Verfahrens zur Lösung mehrerer ℓ^2-TV-Probleme in Mannigfaltigkeiten mit positiver und negativer Krümmung.

Abstract

This thesis introduces a higher-order optimization method for solving non-smooth variational problems on Riemannian manifolds. In this work, we apply the Riemannian Semismooth Newton (RSSN) method to a non-smooth non-linear optimality system derived in recent advances in manifold duality theory. In particular we will show a novel local convergence result for an inexact version of the Riemannian Semismooth Newton method and show state-of-the-art performance in numerical experiments by solving several ℓ^2-TV-like problems on manifolds with positive and negative curvature.

Acknowledgment

I would like to thank Ronny Bergmann for fruitful discussions regarding several differential geometric interpretations of earlier work and his Julia library Manopt.jl. Despite being extremely busy, Ronny managed to respond with enthusiasm after every email I sent. Without his suggestions, this work would not have been what is has become.

Furthermore, I would like to thank Jan Lellmann for weekly online discussions. Especially, in these times of the pandemic, without these meetings this work would not have seen the light of day.

Contents

List of Notation and Symbols

Manifolds

$\mathcal{P}(d)$	Symmetric positive definite $d \times d$ matrices
$\mathbb{R}^d$	d-dimensional Euclidean space
S^d	d-dimensional sphere
$SO(3)$	Rotational group of $\mathbb{R}^3$
$\mathcal{TM}$	Tangent bundle $(= \cup_{p\in\mathcal{M}} T_p\mathcal{M})$
$\mathcal{T}_p\mathcal{M}$	Tangent space at p to $\mathcal{M}$
(U, φ)	Chart

Mappings

∇	Levi-Civita connection
$\flat$	Flat isomorphisim $\mathcal{T}_p\mathcal{M} \ni X \mapsto X^\flat \in \mathcal{T}_p^*$
$\sharp$	Sharp isomorphisim $\mathcal{T}_p^*\mathcal{M} \ni \xi \mapsto \xi^\sharp \in \mathcal{T}_p\mathcal{M}$
$\partial_C X(x)$	Subdifferential of a function X at a point $x \in \mathbb{R}^d$
$d(p, q)$	Riemannian distance between p and q
$D_C F(p)$	Clarke generalized differential of F at $p \in \mathcal{M}$
$D_p F[v]$	Differential of F at $p \in \mathcal{M}$ applied to $v \in \mathcal{T}_p\mathcal{M}$
Exp	Matrix exponential mapping
$\partial f(x)$	Subdifferential of a function f at a point $x \in \mathbb{R}^d$
$g(\cdot,\cdot)_p$	Riemmannian metric tensor at $p \in \mathcal{M}$
Log	Matrix logarihmic mapping
$\partial_{\mathcal{M},C} X(p)$	Clarke generalized covariant derivative of a vector field X at a point $p \in \mathcal{M}$
$\partial_{\mathcal{M}} F(p)$	Subdifferential of a function F at a point $p \in \mathcal{M}$
$(\cdot,\cdot)_p$	Riemmannian metric at p $(= g(\cdot,\cdot)_p)$
$\log_p q$	Logarihmic mapping at p applied to q
$\exp_p(v)$	Exponential mapping at p applied to v
$P_{p\to q} v$	Parallel transport of v from $\mathcal{T}_p\mathcal{M}$ to $\mathcal{T}_q\mathcal{M}$
$R(\cdot,\cdot)\cdot$	Riemannian curvature tensor
$\gamma_{p,q}$	Geodesic between p and q
$\gamma_{p;v}$	Geodesic starting from p in direction v

Abbreviations

ADMM	Alternating Directions Method of Multipliers
ASSN	Adaptive Semismooth Newton Method
CPPA	Cyclic Proximal Point Algorithm
eRCPA	Exact Riemannian Chambolle Pock Algorithm

eRSSN	Exact Riemannian Semismooth Newton Method
IRSSN	Inexact Riemannian Semismooth Newton Method
lRCPA	Linearized Riemannian Chambolle Pock Algorithm
lRSSN	Linearized Riemannian Semismooth Newton Method
PDHG	Primal Dual Hybrid Gradient Algorithm
PPA	Proximal Point Algorithm
RCPA	Riemannian Chambolle Pock Algorithm
ROF	The Rudin-Osher-Fatemi model
RSSN	Riemannian Semismooth Newton Method
SSN	Semismooth Newton Method
TV	Total Variation regularization

Chapter 1: Introduction

Although the field of image processing covers a wide variety of topics and problems, models for image processing will often come down to the minimization of some cost functional for the retrieval of an image. Take for example the *Rudin-Osher-Fatemi* (ROF) image denoising model [ROF92]. In its discrete anisotropic form, it can be written as the minimization problem

$$\min_{x \in \mathbb{R}^{N \times M}} \left\{ \frac{1}{2} \|x - h\|_2^2 + \alpha \|\nabla x\|_1 \right\}, \tag{1}$$

where $h \in \mathbb{R}^{N \times M}$ is a noisy image and ∇ is the discrete (horizontal and vertical) first-order gradient operator. Intuitively, this variational model says that we want the solutions to be reasonably close to the data h, but we say additionally (from prior knowledge) that its derivative in every direction should be small. Adding the second term as prior information is called regularization and is crucial in many image processing tasks to get well-behaved solutions that are robust with respect to noise. The particular approach (1) to regularizing is derived from so-called *Total Variation* (TV) regularization, which is one of the key models for image processing. The ROF (or ℓ^2-TV) model (1) has become the prototype of modern day variational image processing. Algorithms for solving this problem and its generalizations continue to be an active topic of research.

The class of methods for solving non-smooth convex optimization problems like (1) using only generalizations of the gradient of f and g, is often referred to as the *first-order methods*. Although these algorithms work reasonably well, the major drawback is their slow tail convergence: typically, these algorithms achieve ϵ-suboptimality within $\mathcal{O}(\frac{1}{\epsilon})$ iterations. Using acceleration, the amount of steps can be reduced to $\mathcal{O}(\frac{1}{\sqrt{\epsilon}})$ [HYY14]. For high accuracy solutions first-order methods are often not the best choice and instead *higher-order methods* are used. These algorithms use higher-order derivatives and come with linear or superlinear convergence.

1.1 Motivation

This brings us to the motivation of this thesis: second-order methods for manifold-valued image processing. In recent years statistical [PFA06, Pen06, FJ07, LNPS17] and PDE approaches [KS02, CTDF04] to data processing on manifolds have been researched extensively. From 2010 onwards, non-smooth variational approaches on Riemannian manifolds have been gaining momentum as well.

Manifold-valued Images

These types of images do not take a real value on every pixel as assumed in (1), but are “manifold-valued”. The most common manifolds for imaging (besides the classical $\mathbb{R}^d$) are the sphere S^d, the space of positive definite matrices $\mathcal{P}(d)$ and the special orthogonal group $SO(3)$. Examples from applications include, but are not limited to:

Non-linear Colour Spaces (CB and HSV) Whereas the standard RGB colour model is very convenient in application, non-linear colour models have been proposed as well. These models tend to give a better representation in terms of human colour perception.

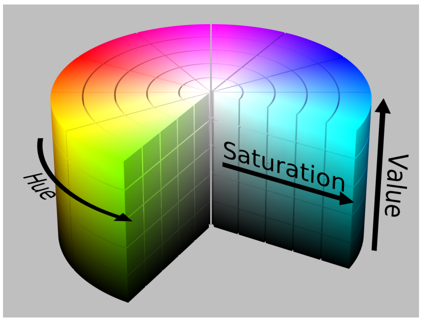


Figure 1: HSV model [Com18]

In the *Chromaticity Brightness* colour space [CKS01], the first octant is given a non-linear structure by viewing $\mathbb{R}^3$ as $S^2 \times \mathbb{R}$, restricted to the first octant. The chromaticity is given by the normalized RGB vector, which gives a value S^2, and the brightness by the length of the vector, giving a value in $\mathbb{R}$.

Another popular non-linear colour space is the *Hue Saturation Value* (HSV) colour space. Here, we look at the $S^1 \times \mathbb{R}^2$ manifold. This model can be interpreted as arranging the pure colors around a full circle and then select the saturation and the brightness (value).

Radar Interferometry (InSAR) InSAR images are obtained as the phase differences of two *Synthetic Aperture Radar* (SAR) images. These are images of an area taken from different positions or different times [MF98]. The technique is applied to detect millimeter changes to landscapes over days or even years. InSAR imaging has applications in monitoring earthquakes, volcanoes and landslides. The computed phase signals can be seen as data on the circle, i.e., the S^1 manifold. Such an S^1 image is typically visualized by giving each pixel the corresponding hue as discussed in the previous application.

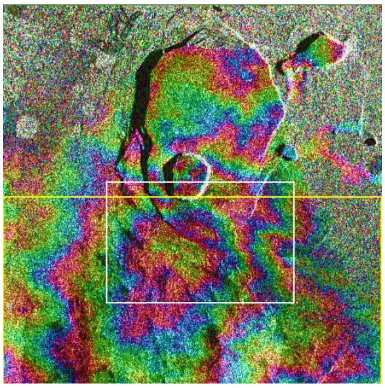


Figure 2: InSAR image [Com19]

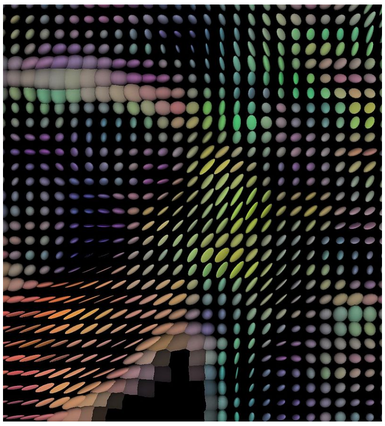


Figure 3: DTI image [Com20]

Diffusion Tensor Imaging (DTI) In Diffusion Tensor MRI we are interested in capturing information on the diffusion of water molecules in different tissues [BML94]. Taking at least six measurements with different magnetic fields, it is possible to capture this information in a diffusion tensor field, which can be represented as a symmetric positive definite matrix at every pixel. In other words we find our data in the space $\mathcal{P}(d)$. We can visualize such a positive definite matrix as an ellipsoid that indicates the amount of diffusion occurring in all spatial directions.

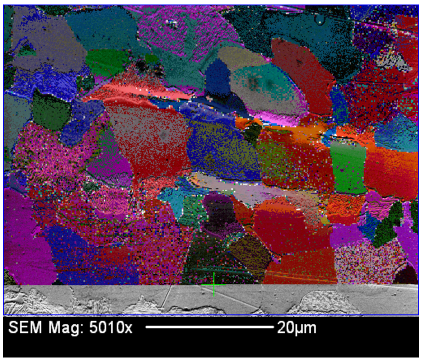


Figure 4: EBSD image [Com13]

Electron Backscatter Diffraction (EBSD) EBSD is a technique for exploring and mapping microstructures in crystalline materials such as metals and minerals [AWK93]. Backscattered electrons give information on the orientation of these microstructures. This orientation can be modelled by the rotation group $SO(3)$. Regions with the same orientation are called *grains*. Mapping this grain structure is the end goal and gives information on macroscopic material properties such as conductivity and lifetime. The grains can be assigned a certain colour[1] so that the orientation information can be captured in a single image in the end.

Manifold-valued Models and Solvers
In order to process these types of data, both the models and the solvers from image processing in $\mathbb{R}^d$ have to be generalized. So far, most of the attention has gone to the generalization of successful models and first-order algorithms. However, there are a lot of open questions regarding higher-order methods. In this work, we will be investigating how to extend the work of [RLV17] on the *Semismooth Newton* method for Total Variation regularization to the manifold setting. Our goal will be to develop a general fast second-order algorithm for non-smooth variational models on manifold-valued data.

1.2 Related work

Whereas applications of TV regularization on manifolds gained momentum at the start of the 2010s, the analysis of TV in the manifold setting can be traced back to the 1990s [GMS93]. Initially starting off with the case of the S^1 manifold, Total Cyclic Variation is proposed and existence of its minimizers was shown. It was only until 2006 that this result got a follow up. The more general result of functions $u : \Omega \to \mathcal{Y}$ on manifold domains Ω to a manifold $\mathcal{Y}$ has been studied in [GM06, GM07] using the theory of Cartesian currents. Here the space $BV(\Omega, \mathcal{Y})$ was rigorously defined and some properties such as lower semi-continuity were established.

Starting Up Manifold Valued Imaging
Although early (numerical) attempts for solving TV on manifolds originate from the early 2000s, e.g., with [CKS01] who proposed a method for denoising cyclic colour data, the majority of the contributions in TV regularization on manifolds were proposed in the 2010s. Here [SC11] tried to bridge the gap between the theoretical results mentioned before and applications by introducing TV regularization on S^1 and providing a numerical method to solve it. This attempt quickly got a follow up in [CS13], whose authors presented several novelties among which extensions to more general regularizers such as

[1] This scheme is somewhat more complex and will not be discussed further. A clear explanation can be found in [Per18].

Hubert-TV. In [VBK13] the authors focus on the $\mathcal{P}(3)$ manifold of symmetric positive definite 3×3 matrices[2] while using a TGV approach.

The more general case for solving TV on arbitrary Riemannian manifolds was proposed in [LSKC13], who reformulated the variational problem into a multi label optimization problem. Nevertheless, the key work of [WDS14] became in the end most popular by going with the full *intrinsic* approach. Contrary to *extrinsic* approaches, the methods no longer focussed on the embedding of the manifold into a higher dimensional (linear) space, but used manifold mappings. They proposed the ℓ^2-TV model

$$\min_{p \in \mathcal{M}^{N \times M}} \left\{ \frac{1}{2} \sum_{i,j=1}^{N,M} d\left(p_{ij}, h_{ij}\right)^2 + \alpha \sum_{i,j=1}^{N-1,M} d\left(p_{ij}, p_{i+1,j}\right) + \alpha \sum_{i,j=1}^{N,M-1} d\left(p_{ij}, p_{i,j+1}\right) \right\}, \tag{2}$$

as generalized Total Variation denoising for manifold valued images. Here $d(p,q)$ is the distance between $p, q \in \mathcal{M}$ on the manifold.

Generalized Models

When entering the second half of the 2010s we see more general models emerging. So was a second-order model for cyclic data proposed in [BLSW14], but soon enough popular models in the Euclidean case also got their generalized manifold case counterpart: a general second-order method [BBSW16], infimal convolution models [BFPS17, BFPS18] and TGV for manifold-valued imaging [BHSW18, BFPS18] were introduced. Moreover, at the same time specialized models got extended to applications such as inpainting [BW15], segmentation [WDS16], or manifold valued inverse problems [BWW$^+$16, SW18]. Moreover, new problem settings arose with the emergence TV for manifold-valued data on graphs [BT18].

Generalized Solvers

As TV and variants got extended to manifolds, so did the solvers. The proposed algorithms were given in terms of *Hadamard manifolds*: Riemannian manifolds that are complete, simply connected and have negative sectional curvature. The author of [Bac14] proposed the Cyclic Proximal Point Algorithm (CPPA) on Hadamard manifolds as an extension the proximal point method [Ban14]. The latter was again an extension onto the manifold case of the algorithm proposed in [Ber11] for the Euclidean case. Iteratively Reweighted Least Squares (IRLS) [BCH$^+$15] was a generalization of the version for spheres in [GS14]. The IRLS was then again adapted in [GS16] by adding a quasi-Newton step. Furthermore, [BPS16] proposed the Parallel Douglas-Racheford Algorithm (PDRA) as a generalization of the Douglas-Racheford algorithm for symmetric Hadamard manifolds, [SWU16] proposed an exact solutions for ℓ^1-TV with spherical data and [VSCL19] proposed a functional lifting approach.

[2] Better: 3×3 tensors.

An Important Development in Manifold-valued Image Processing
However, there were still issues with these algorithms regarding application to TV-based energies. As in the linear case, the proximal map of the TV term cannot be written in a closed form. Hence the maps typically have to be approximated, which takes much time. In the Euclidean case the difference operator was dealt with by using techniques such as Bregman splitting, ADMM or a Fenchel duality-based primal-dual algorithm as mentioned in the previous section. The latter has been realized in the most recent contribution. Fenchel duality theory for Hadamard manifolds was in the end introduced in [BHTVN19]. The authors proposed the *exact Riemannian Chambolle Pock Algorithm* (eRCPA) and the *linearized Riemannian Chambolle Pock Algorithm* (lRCPA) as an application of the developed theory. The approach shows competitive runtimes compared to other algorithms, made it easier to handle the proximity operators and was compatible with isotropic TV, which was still impossible up until then. This development will be the stepping stone towards our goal of realizing higher-order methods for manifold valued imaging.

1.3 Contribution

In this work sections containing new contributions are denoted by an asterix (*) in the section header. We present four main contributions to the field: one for the Semismooth Newton method in $\mathbb{R}^d$ and three related to the Riemannian Semismooth Newton method.

Contributions to the Semismooth Newton Related Topics

1. Resolving Known Issues with Semismooth Newton for TV in Euclidean Space In prior results [RLV17] SSN was applied to TV already. However, the method suffered from ill-posedness of the Newton matrix. The direct cause of the non-invertibility was not entirely clear, but it was shown that the matrix was positive semi-definite. Hence, the proposed solution was adding βI to the Newton matrix, where I the identity matrix and $\beta \ll 1$, in order to make the matrix positive definite and thus invertible. From a theoretical point of view it was not clear whether we could still expect superlinear convergence. Nevertheless, in practice it did not seem to be a problem.

In this work we will show explicitly where the ill-posedness comes from and present a more targeted solution that only resolves the equations in the Newton system causing the ill-posedness (contrary to the prior approach which used perturbed the whole Newton matrix). Furthermore, we investigate its performance in numerical experiments.

Contributions to Riemannian Semismooth Newton Related Topics

2. Being the First to Apply and Implement Riemannian Semismooth Newton The main contribution of this work comes from merging the ideas of [BHTVN19] [RLV17] and [OF18]. That is, using the theoretical framework of Riemannian Semismooth Newton [OF18] and the Fenchel duality theory [BHTVN19], we generalize the ideas presented in [RLV17] and develop a duality-based higher-order method for non-smooth variational problems on manifolds. We apply the new approach to several 1D and 2D problems and obtain state-of-the-art performance.

3. Expanding the Theoretical Framework of Riemannian Semismooth Newton For larger-scale problems, it is often not feasible to solve the Newton matrix. Hence, iterative methods for solving large-scale linear systems will be the next step towards making the method numerically feasible.

We build upon the theory of the Riemannian Semismooth Newton method [OF18] towards an *inexact* version and provide several convergence proofs (Thm. 6.12). We still get linear convergence despite the inexact solution of the Newton system. In numerical experiments, we validate our theoretical results.

4. Gaining Novel Insights Into Open Questions in Previous Work From the Fenchel duality theory on manifolds [BHTVN19], the authors derived two non-linear optimality systems: the so-called exact system and the linearized system. Whereas the names suggest that only the latter is an approximation, the former is too. However, the linearized system had more theoretical motivation, i.e., the resulting fixed point algorithm, the lRCPA, is shown to converge.

This thesis provides new insights into the workings of the exact optimality system: in particular, due to the high accuracies we can reach with our Riemannian Semismooth Newton method, we find hints that the exact optimality system does not have the same minimizer as ℓ^2-TV and therefore is not a good way of approaching non-smooth optimization on manifolds.

1.4 Outline

This work thesis is organized in two parts: one focused on Semismooth Newton (SSN) for Euclidean space (Chapter 2 and 3) and the other focused on generalizing the method to manifolds (Chapter 4,5 and 6). The first part continues directly upon the work in [RLV17]. The main focus here is resolving issues and gather information and intuition useful for the manifold case. The second part contains mostly new content.

Chapter 2
In this chapter our goal is to understand how to minimize a general non-smooth convex problem given by the sum of two convex energies. Through non-smooth convex analysis we will learn how classical smoothness restrictions can be replaced by convexity. We will discuss Fenchel duality theory which provides the tools to rewrite our optimization problem in a computationally feasible form. Finally, we will discuss the so-called Primal Dual Hybrid Gradient (PDHG) method, which will be the basis for the following.

Chapter 3
In this chapter we discuss the limitations of methods such as PDHG for solving non-smooth variational problems and our goal is to develop a faster alternative. We will look into the Semismooth Newton (SSN) method as a second-order-acceleration of PDHG. In particular, we will rewrite the PDHG iteration into a form that is suitable for SSN. Next, we will look into an application: SSN for solving ℓ^2-TV. We will continue the work in [RLV17] and resolve some remaining issues. With numerical experiments we show that

our novel approach to SSN for ℓ^2-TV does no longer suffer from non-invertibility and we obtain superlinear convergence.

Chapter 4
In order to generalize the notions from Euclidean space to manifolds, we need generalizations of our notions on Riemannian manifolds. In this chapter we will start with defining smooth manifolds and discussing important general notions such as differentials, vector fields, vector bundles and discuss the important result that every Riemannian manifold admits a metric tensor. From that we discuss Riemannian geometry, which enables us to talk about important manifolds mappings that will become important later when discussing the Riemannian Semismooth Newton method. We will also look into an important application of Riemannian geometry: Jacobi fields. These fields provide us with useful expressions that are key for later implementation.

In this chapter we will also take a closer look at the S^d and $\mathcal{P}(3)$ manifold. In particular, we will provide the relevant mappings that become important for implementing algorithms on these manifolds.

Chapter 5
This chapter is a direct generalization of the contents in chapter 2 to manifolds and relies heavily on the notions developed in the work in [BHTVN19]. The main goal is working towards a generalization to PDHG on manifolds. As it turns out the non-linearity of manifolds does not allow a direct generalization. Instead we must resort to algorithms solving for approximate solutions. The proposed algorithms, i.e., the exact and linearized Riemannian Chambolle Pock algorithms, are discussed along with open questions of the choices made here.

Chapter 6
Finally, we are ready to develop the Riemannian Semismooth Newton method. This chapter will be a direct generalization to chapter 3 and is structured in a similar fashion. We will discuss different ways of generalizing SSN to manifolds and proceed by focusing on vector fields. Subsequently, we will discuss the theory behind RSSN as formulated in [OF18] and prove a new result by looking at solving the Newton system inexactly. After discussing how to construct the Newton matrix, we will focus on TV denoising as an application for our method. Finally, in the numerical experiments we investigate both the RSSN and its inexact counterpart and we compare the former's performance to other state-of-the-art algorithms.

Chapter 7
We conclude the thesis by discussing the key issues of RSSN and provide suggestions for future work.

Chapter 2: Preliminaries I: Non-smooth Optimization

Before understanding how to solve non-smooth optimization problems on a manifold, we first consider how to solve them in a vector space. We will focus on solving

$$\inf_{x\in\mathbb{R}^n} \{f(x) + g(Ax)\}\,, \tag{3}$$

where $A : \mathbb{R}^n \to \mathbb{R}^m$ a linear mapping, $f : \mathbb{R}^n \to \bar{\mathbb{R}}$, $g : \mathbb{R}^m \to \bar{\mathbb{R}}$ are non-smooth functions and $\bar{\mathbb{R}}$ is the extended real line as defined below.

In this chapter we will build up towards solving this general problem by taking a step by step approach and introduce tools from convex analysis in $\mathbb{R}^n$ in Sect. 2.1 and a general way to solve the problem in Sect. 2.2. More details can be found in works such as [RW09, CV20].

2.1 Non-smooth Analysis

In the following we will look can also look at a certain class of functions: functions mapping into the extended real line.

Definition 2.1 (extended real line)**.** *Let* $\bar{\mathbb{R}} := \mathbb{R} \cup \{+\infty, -\infty\}$ *be the* extended real line *with the rules:*

(i) $\infty + c = \infty$ *and* $-\infty + c = -\infty \quad$ *for all* $c \in \mathbb{R}$*,*

(ii) $0 \cdot \infty = 0$ *and* $0 \cdot (-\infty) = 0$*,*

(iii) $\inf \mathbb{R} = \sup \emptyset = -\infty$ *and* $\inf \emptyset = \sup \mathbb{R} = +\infty$*,*

(iv) $+\infty - \infty = -\infty + \infty = +\infty$*.*

The first goal is generalizing the gradient of a function. For a useful generalized gradient, we need some additional information on f that provide a minimum of regularity. These properties turn out to be properness, convexity and lower semi-continuity. For that consider the following definitions.

Definition 2.2 (proper)**.** *A function* $f : \mathbb{R}^n \to \bar{\mathbb{R}}$ *is* proper *if* $\operatorname{dom} f := \{x \in \mathbb{R}^n | f(x) < \infty\} \neq \emptyset$ *and* $f(x) > -\infty$ *holds for all* $x \in \mathbb{R}^n$*.*

Definition 2.3 (convex)**.** *Let* C *be a convex set in* $\mathbb{R}^n$*. A function* $f : C \to \bar{\mathbb{R}}$ *is* convex *if*

$$f\,(tx_1 + (1-t)x_2) \leq tf\,(x_1) + (1-t)f\,(x_2) \qquad \forall x_1, x_2 \in C, t \in [0, 1]. \tag{4}$$

Definition 2.4 (epigraph)**.** *Let* $U \subset \mathbb{R}^n$ *be open. The* epigraph *of a function* $f : U \to \bar{\mathbb{R}}$ *is defined as*

$$\operatorname{epi} f := \{(x, \alpha) \in U \times \mathbb{R} | f(x) \leqslant \alpha\}\,. \tag{5}$$

Definition 2.5 (lower semi-continuous)**.** *Let $U \subset \mathbb{R}^n$ be open. A proper function $f : U \to \bar{\mathbb{R}}$ is called* lower semi-continuous *(lsc) if* $\operatorname{epi} f$ *is closed.*

Having these definitions, we are ready to generalize the gradient. The idea is that we can try to find hyperplanes tangent to and completely below the graph of the function.

Definition 2.6 (subgradient)**.** *Let $U \subset \mathbb{R}^n$ be open. For every $f : U \to \bar{\mathbb{R}}$ and $x \in U$,*

$$\partial f(x) := \{v \in \mathbb{R}^n | f(y) \geq f(x) + \langle v, y - x \rangle \ \forall y \in U\} \tag{6}$$

is the subdifferential *or the set of* subgradients *of f at x.*

Remark 2.7. *Note that in the case that the function is smooth and convex, we actually get that the set of sudgradients is a singleton containing the gradient if x is in the domain (otherwise it is empty).*

A generalized backward step

Consider some smooth function $f : \mathbb{R}^n \to \bar{\mathbb{R}}$. In order to find the minimum of this function, a gradient descent scheme would be a reasonable first approach, i.e.,

$$u^{k+1} = u^k - \tau \nabla f(u^k). \tag{7}$$

This equation can be viewed as a discretization of the ODE

$$u_t = -\tau \nabla f(u). \tag{8}$$

Now, instead of solving (8) we can try to discretize

$$u_t \in -\tau \partial f(u) \tag{9}$$

in the more general setting of a proper, convex, lsc function. Previously a forward Euler discretization was used. Remember that we can use backward Euler as well. Then, for the updated u^k we can choose

$$u^{k+1} \in F_{\tau f}(u^k) := (I - \tau \partial f) u^k, \tag{10}$$

$$u^{k+1} \in B_{\tau f}(u^k) := (I + \tau \partial f)^{-1} u^k. \tag{11}$$

The latter expression is closely related to the *proximal map.*

Definition 2.8 (proximal map)**.** *Let $f : \mathbb{R}^n \to \bar{\mathbb{R}}$ be proper, lsc, convex and let $\tau > 0$. The* proximal map *of f is defined as*

$$\operatorname{prox}_{\tau f}(x) := \arg\min_y \left\{ \frac{1}{2} \|y - x\|_2^2 + \tau f(y) \right\}. \tag{12}$$

A well-known result gives us an equivalent form for the backward step:

Proposition 2.9 ([RW09, Prop. 12.19])**.** *If $f : \mathbb{R}^n \to \bar{\mathbb{R}}$ is proper, lsc, convex with $\tau > 0$, then the backward step is uniquely given by*

$$B_{\tau f}(x) = \operatorname{prox}_{\tau f}(x). \tag{13}$$

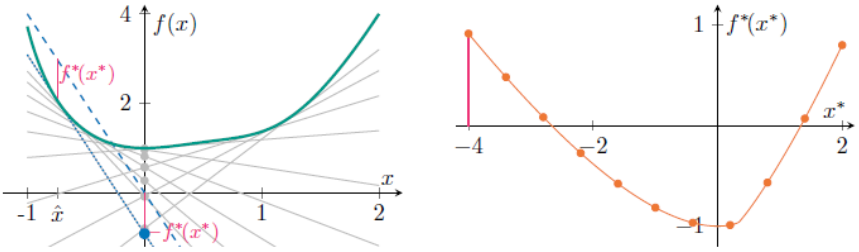


Figure 5: A visual explanation of the Fenchel conjugate of some function f. [BHTVN19]

Finally, with this we have the machinery to solve for

$$\inf_{x \in \mathbb{R}^n} f(x), \tag{14}$$

by simply iterating the proximal mapping. This is known as the *proximal point algorithm* (PPA). However, computing a single proximal step on the full problem is generally as hard as solving the original problem. Therefore, we consider problems of the form

$$\inf_{x \in \mathbb{R}^n} \left\{ f(x) + g(Ax) \right\}. \tag{15}$$

In this work we will focus on the primal-dual approach.

2.2 The Primal-Dual Hybrid Gradient Algorithm

For tackling the general problem in (15) we will use the following general strategy: introduce a variable y and rewrite the problem into

$$\inf_{x \in \mathbb{R}^n, y \in \mathbb{R}^m} \left\{ f(x) + g(y) \right\}, \qquad \text{s.t.} \;\; Ax = y \tag{16}$$

and try to optimize f and g alternatingly while trying to fulfil the constraint. We will see that duality theory comes to rescue and provides a clever way to do this splitting by considering the Lagrangian.

Definition 2.10 (Fenchel conjugate)**.** *Let $f : \mathbb{R}^n \to \bar{\mathbb{R}}$ be a function. The* Fenchel conjugate *of f is defined as the function $f^* : \mathbb{R}^n \to \bar{\mathbb{R}}$ such that*

$$f^*\left(x^*\right) := \sup_{x \in \mathbb{R}^n} \left\{ \langle x^*, x \rangle - f(x) \right\}. \tag{17}$$

For an interpretation of this function consider Fig. 5. The Fenchel conjugate at some point x^* can be seen as the (negative) distance we have to translate the hyperplane induced by the vector $[x^*, -1]^\top$ down (up) until it is tangent to the graph of f.

Subsequently, we can find the Fenchel conjugate of the Fenchel conjugate.

Definition 2.11 (Fenchel biconjugate)**.** *Let $f : \mathbb{R}^n \to \bar{\mathbb{R}}$ be a function. The* Fenchel biconjugate *of f is defined as the function $f^{**} : \mathbb{R}^n \to \bar{\mathbb{R}}$ such that*

$$f^{**}\left(x^{**}\right) := \sup_{x^* \in \mathbb{R}^n} \left\{ \langle x^{**}, x^* \rangle - f^*(x^*) \right\}. \tag{18}$$

The following theorem is the final piece we need to solve the problem (15).

Theorem 2.12 (Fenchel–Moreau-Rockafellar, [CV20, Thm. 5.1])**.** *Given a proper function $f : \mathbb{R}^n \to \bar{\mathbb{R}}$, the equality $f^{**}(x) = f(x)$ holds if and only if f is lower semi-continuous and convex.*

The Lagrangian approach

Using the previous result, we see that if f, g are proper lsc convex, we can rewrite

$$\inf_{x\in\mathbb{R}^n} \{f(x) + g(Ax)\} \tag{19}$$

into

$$\inf_{x\in\mathbb{R}^n} \sup_{y\in\mathbb{R}^m} \{f(x) - g^*(y) + \langle Ax, y\rangle\}, \tag{20}$$

where we call $l(x, y) := f(x) - g^*(y) + \langle Ax, y\rangle$ the *Lagrangian.* If we were now able to exchange inf and sup, we could write

$$\inf_{x\in\mathbb{R}^n} \sup_{y\in\mathbb{R}^m} \{f(x) + \langle y, Ax\rangle - g^*(y)\} = \sup_{y\in\mathbb{R}^m} \inf_{x\in\mathbb{R}^n} \{f(x) + \langle y, Ax\rangle - g^*(y)\} \tag{21}$$

$$= \sup_{y\in\mathbb{R}^m} \left\{ - \left\{ \sup_{x\in\mathbb{R}^n} -f(x) + \left\langle -A^\top y, x \right\rangle \right\} - g^*(y) \right\}. \tag{22}$$

From the definition of f^*, we obtain the *dual problem*

$$\sup_{y\in\mathbb{R}^m} \{-g^*(y) - f^*(-A^\top y)\}. \tag{23}$$

The following result provides a sufficient condition for (21) to hold and an alternative optimality condition.

Theorem 2.13 ([CV20, Thm. 5.9])**.** *Let $f : \mathbb{R}^n \to \bar{\mathbb{R}}$ and $g : \mathbb{R}^m \to \mathbb{R}$ be proper, convex, and lower semi-continuous, and $A : \mathbb{R}^n \to \mathbb{R}^m$ be a linear mapping. Assume furthermore that*

(i) the primal problem (19) admits a solution $\hat{x} \in \mathbb{R}^n$

(ii) there exists an $x_0 \in \operatorname{dom}(g \circ A) \cap \operatorname{dom} f$ with $Ax_0 \in \operatorname{int}(\operatorname{dom} g)$

Then, the dual problem (23) admits a solution $\hat{y} \in \mathbb{R}^m$ and

$$\min_{x\in\mathbb{R}^n} f(x) + g(Ax) = \max_{y\in\mathbb{R}^m} -g^*(y) - f^*(-A^\top y). \tag{24}$$

Furthermore, $\hat{x}$ and $\hat{y}$ are solutions to (19) and (23), respectively, if and only if

$$-A^\top \hat{y} \in \partial f(\hat{x}), \tag{25}$$

$$A\hat{x} \in \partial g^*(\hat{y}). \tag{26}$$

Assuming that a solution exists, we can try to solve the *primal-dual* optimality system

$$\begin{aligned} -A^\top y &\in \partial f(x), && (27)\\ Ax &\in \partial g^*(y), && (28) \end{aligned}$$

by rewriting it. For $\sigma > 0$ (27) can be rewritten into

$$\begin{aligned} -A^\top y \in \partial f(x) &\Leftrightarrow -\sigma A^\top y \in \sigma \partial f(x) && (29)\\ &\Leftrightarrow x - \sigma A^\top y \in x + \sigma \partial f(x) && (30)\\ &\overset{(13)}{\Leftrightarrow} x = \operatorname{prox}_{\sigma f}(x - \sigma A^\top y). && (31) \end{aligned}$$

Similarly, for $\tau > 0$ (28) can be rewritten into

$$Ax \in \partial g^*(y) \Leftrightarrow y = \operatorname{prox}_{\tau g^*}(y + \tau Ax). \tag{32}$$

We find the equivalent form of the primal-dual optimality system

$$\begin{aligned} x &= \operatorname{prox}_{\sigma f}(x - \sigma A^\top y), && (33)\\ y &= \operatorname{prox}_{\tau g^*}(y + \tau Ax). && (34) \end{aligned}$$

These proximal maps also play a role in constructing an iterative algorithm for solving (20):

- find y^{k+1} by only considering $-g^*(y) + \langle Ax^k, y\rangle$,
- find x^{k+1} by only considering $f(x) + \langle Ax, y^{k+1}\rangle$,
- repeat until converged.

If we take a backward step for y we get the update

$$\begin{aligned} y^{k+1} &= \arg\min_v \left\{ \tau g^*(v) - \tau\langle Ax^k, v\rangle + \frac{1}{2}\|v - y^k\|^2 \right\} && (35)\\ &= \arg\min_v \left\{ \tau g^*(v) - \tau\langle Ax^k, v\rangle + \frac{1}{2}\|v - y^k\|^2 + \frac{1}{2}\|\tau Ax^k\|^2 + \tau\langle Ax^k, y^k\rangle \right\} && (36)\\ &= \arg\min_v \left\{ \tau g^*(v) + \frac{1}{2}\|v - (y^k + \tau Ax^k)\|^2 \right\} = \operatorname{prox}_{\tau g^*}(y^k + \tau Ax^k). && (37) \end{aligned}$$

Similarly, for x we get the update

$$x^{k+1} = \operatorname{prox}_{\sigma f}(x^k - \sigma A^\top y^{k+1}). \tag{38}$$

These ideas were combined in the Primal-Dual Hybrid Gradient (PDHG) Algorithm [CP11, Alg. 2].

Algorithm 1 (modified) PDHG

```
Initialization: Let $L := \|A\|$ and $\tau_0, \sigma_0 > 0$ such that $\tau_0\sigma_0 L^2 \leq 1, \gamma \geq 0, x^0 \in \mathbb{R}^n, y^0 \in \mathbb{R}^m$ and set $\bar{x}^0 := x^0$
while not converged do
    $y^{k+1} := \operatorname{prox}_{\tau_k g^*}(y^k + \tau_k A\bar{x}^k)$
    $x^{k+1} := \operatorname{prox}_{\sigma_k f}(x^k - \sigma_k A^\top y^{k+1})$
    $\theta_k := 1/\sqrt{1+2\gamma\tau_k},\ \tau_{k+1} := \theta_k\tau_k,\ \sigma_{k+1} := \sigma_k/\theta_k$
    $\bar{x}^{k+1} := x^{k+1} + \theta_k(x^{k+1} - x^k)$
end while
```

Moreover, we have the following convergence result.

Theorem 2.14 ([CP11, Thm. 2]). *Let $f : \mathbb{R}^n \to \bar{\mathbb{R}}$ and $g : \mathbb{R}^m \to \mathbb{R}$ be proper, convex, and lower semi-continuous, let $A : \mathbb{R}^n \to \mathbb{R}^m$ be a linear mapping, let $\tau_0 > 0$, $\sigma_0 := 1/(\tau_0 L^2)$, and let $(x^k, y^k)_{k\geq 1}$ be defined by Alg. 1. Assume the existence of $\gamma > 0$ such that for any $x \in \operatorname{dom}\partial f$*

$$f\left(x'\right) \geq f(x) + \langle p, x' - x\rangle + \frac{\gamma}{2}\left\|x - x'\right\|^2, \quad \forall p \in \partial f(x), x' \in \mathbb{R}^n. \tag{39}$$

Then for all $\varepsilon > 0$ there exists N_0 (depending on ε and $\gamma\tau_0$) such that for every $N \geq N_0$,

$$\left\|\hat{x} - x^N\right\|^2 \leq \frac{1+\varepsilon}{N^2}\left(\frac{\left\|\hat{x} - x^0\right\|^2}{\gamma^2\tau_0^2} + \frac{L^2}{\gamma^2}\left\|\hat{y} - y^0\right\|^2\right), \tag{40}$$

where $(\hat{x}, \hat{y})$ is the solution of (20).

Chapter 3: The Semismooth Newton Method

In this chapter we will explore a the possibilities and limitations of the *Semismooth Newton method* (SSN) as a higher order method for non-smooth variational problems. In this part we will continue the research in [RLV17], whose author already researched the Semismooth Newton method in her master thesis. The overarching goal of this chapter is to clarify all issues related to the method and see what we do and do not want to generalize to the manifold case.

To this end, Sect. 3.1 will provide some additional motivation and context for the use and the development of the method. In Sect. 3.2 the theoretical background around the Semismooth Newton method will be discussed. In Sect. 3.3 we will elaborate on using the Semismooth Newton method as a higher order primal-dual method. In Sect. 3.4 the method will be applied for minimizing the ℓ^2-TV functional. Moreover, we make our contribution here by discussing known issues and resolving them. In Sect. 3.5 we investigate performance with numerical experiments. Finally, in Sect. 3.6 we will give some concluding remarks and provide an outlook and some suggestions to other possibilities for higher order methods, that went beyond the scope of this thesis.

3.1 Introduction

The class of methods for solving convex optimization problems of the form

$$\inf_{x\in\mathbb{R}^n} \left\{ f(x) + g(Ax) \right\}, \tag{41}$$

using only first-order information of f and g independently, i.e., one way or another using subgradients or proximal mappings, is often referred to as *first-order splitting methods.* Besides TV-based regularizing, also ℓ^1 regularization [DDDM04] can be written as (41). The latter gained popularity in applications such as compressed sensing [Don06].

These methods became an active field of research from around 2005 onwards. Examples of methods include but are not limited to Forward backward splitting [CW05], Douglas-Racheford splitting [CP07], Bregman splitting [YOGD08], the Alternating Method of Multipliers (ADMM) [WYYZ08] and Primal-Dual Splitting [CP11]. For a clear overview of the splitting methods and their relation to each other see [EZC10].

Although these algorithms worked reasonably well, the major drawback was slow tail convergence. Typically, the algorithms achieved ϵ-suboptimality within $\mathcal{O}(\frac{1}{\epsilon})$ iterations. Using acceleration, the amount of steps could typically be reduced to $\mathcal{O}(\frac{1}{\sqrt{\epsilon}})$. Successful cases include the FISTA algorithm and the before mentioned primal-dual algorithm with an overrelaxation on the step size [HYY14]. Other approaches to acceleration include preconditioning [PC11] and continuation approaches [WNF09].

Towards Faster Algorithms

A possible solution to speeding up convergence is using second-order information. In general, one can do this by either passing to interior point methods [FGZ14] or using Newton-like methods. Since the start of the 2010s multiple Newton-like methods have been proposed for problems of the form as in (41). These algorithms include but are not limited to: Quasi Newton methods [YVGS10, Che14], Proximal Newton methods [BF12, PJL$^+$13, LSS14, BNO16, YZS19], Forward Backward Newton methods [PSB14] and Semismooth Newton methods [GL08, MU14, BCNO16b, XLWZ18, LST18].

As mentioned in the chapter opening, in this work we will focus on the latter. This method did not originate from the field of first-order methods, but came from optimal control in the early 2000s [HIK02]. Nevertheless, the idea of the method was already formulated almost 10 years before that [QS93].

A Brief History of SSN

The history of the SSN can be traced back starting to the end of the 1970s with [Mif77] who introduced the notion of semismoothness for real-valued functions in the finite-dimensional case, the authors of [QS93] extended this notion to maps between finite dimensional spaces, introduced SSN to solve semismooth non-linear equations and provided a proof of local superlinear convergence under a non-degeneracy assumption. In [Qi93] the damped-Newton was proposed as an adaptation to SSN and global convergence was established. Later [ZT05] showed local superlinear convergence in the case that the Jacobian is not necessarily nonsingular, but allows for a weaker assumption.

With the theory in place we see application of the method emerging within different fields. The authors of [DLFK96, SH97] were among the first to apply SSN. Their key idea was to apply SSN to non-linear complementary problem functions (NCP-functions). These are functions of the form $\phi : \mathbb{R}^2 \to \mathbb{R}$ with the property

$$\phi(x) = 0 \;\Leftrightarrow\; x_1 \leq 0,\; x_2 \leq 0,\; x_1 x_2 = 0, \tag{42}$$

that are applied component-wise to the NCP system. A popular choice was the Fischer-Burmeister function

$$\phi_{FB}(x) = \sqrt{x_1^2 + x_2^2} - x_1 - x_2. \tag{43}$$

Later this idea using NCP was also adapted in [NQYH07] and applied to TV-based energies. A general approach was proposed in [Ulb02], where SSN applied to an NCP system was extended to the infinite-dimensional case. In the end, the authors of [HIK02] were among the first to break the tradition of solving NCP equations with SSN and applied the method on the optimality conditions of optimal control problems, which could be rewritten into a suitable form.

The Revival of SSN

After [HIK02], the idea of using SSN for problems other than the NCP function continued and got picked up at the late 2000s by the community currently working on ℓ^1-regularized optimization. The algorithms obtained from first-order methods were typically fixed point iterations, i.e.,

$$x^{k+1} = G(x^k), \tag{44}$$

for some (non-smooth) function G. The core idea was to see that this system would converge to some x that would satisfy

$$x = G(x) \;\Leftrightarrow\; x - G(x) = 0, \tag{45}$$

where the right hand equation has the correct form for SSN. Starting from 2008 onwards we see that [GL08] was directly inspired by the example in [HIK02] and saw that the optimality conditions used in the fixed point iteration of the forward backward splitting (FBS) algorithm also had the desired form and properties for SSN. They applied SSN in the infinite-dimensional case to a sparse wavelet regularized inverse problem setting. Six years later in 2014 [MU14] followed up this idea in the more specific setting of ℓ^1 regularization in the finite dimensional case, again by applying SSN to one of the fixed point algorithms. Later, [BCNO16b] extended the latter's contribution by generalizing the framework into a one that could not only generate an active set framework (as was the main idea so far), but also orthant-based methods and a second-order iterative soft-thresholding method.

Recent Developments

The most recent contribution to higher order SSN-based methods for TV is still that in [RLV17], where SSN and variants were applied to problems that could be stated in a primal-dual fashion. Shortly after that, the authors of [XLWZ18] proposed the Adaptive Semismooth Newton Method (ASSN). Their main idea was combining a regularization approach and a hyperplane projection technique in order to establish a scheme that is globally convergent, but also maintains local superlinear convergence. Furthermore, whereas the earlier contributions showed their case for ℓ^1, this approach is formulated for general convex functions, although only applied to ℓ^1 regularization. For now we will focus on SSN, but at the end of this section we will discuss ASSN once more.

3.2 Newton's Method for Non-smooth Systems of Equations

The goal of the Newton method is to find a zero of a function $X : \mathbb{R}^n \to \mathbb{R}^m$, i.e.,

$$X(x) = 0. \tag{46}$$

The key idea is, when starting from a point x, to find a step d such that

$$0 = X(x + d) \approx X(x) + \nabla X(x)d, \tag{47}$$

which leads to the equation for the step d

$$\nabla X(x)d = -X(x). \tag{48}$$

If $n = m$ and $\nabla X(x)$ is invertible, then

$$d = \nabla X(x)^{-1}X(x) \tag{49}$$

and we can iterate

$$x^{k+1} = d^k + x^k. \tag{50}$$

The Newton iteration converges locally superlinearly for smooth X. In this section we will discuss the case that X is non-smooth.

3.2.1 Generalized Differentials and Semismoothness

In the subsequent sections and chapters, we will usually assume that $X : \mathbb{R}^n \to \mathbb{R}^m$ is a locally Lipschitz function. Consequently, by Rademacher's theorem, X is differentiable almost everywhere. This motivates the following generalized differential [Cla90, Def. 2.6.1].

Definition 3.1 (Clarke Generalized Differential)**.** *Let $X : U \subset \mathbb{R}^n \to \mathbb{R}^m$ be locally Lipschitz on the open set U. Let $D_X \subset U$ be the set on which X is differentiable. The* (Clarke) generalized differential *or* Clarke generalized Jacobian *of X at x, denoted by $\partial_C X(x)$, is defined as the convex hull of all $m \times n$ matrices V obtained as the limit of a sequence of $\nabla X(x_i)$ where $x_i \to x$ and $x_i \in D_X$:*

$$\partial_C X(x) := \operatorname{co}\left\{V : x_i \to x, \nabla X(x_i) \to V, x_i \in D_X\right\}. \tag{51}$$

It turns out that this notion will be closely related with the notion of the directional derivative [Hin10, Def. 2.3].

Definition 3.2 (Directional derivative)**.** *Let $X : U \subset \mathbb{R}^n \to \mathbb{R}^m$ be locally Lipschitz on the open set U. We call*

$$X'(x, d) := \lim_{t \searrow 0} \frac{X(x + td) - X(x)}{t} \tag{52}$$

the (one-sided) directional derivative *of X at x in direction d.*

Before relating Def. 3.1 and Def. 3.2, we note that not every locally Lipschitz function is amenable to a generalized Newton method. The notion of *semismoothness* turns out to be an suitable choice of regularity [Hin10, Def. 2.5].

Definition 3.3 (Semismoothness)**.** *Let $U \subset \mathbb{R}^n$ be non-empty and open. The function $X : U \to \mathbb{R}^m$ is* semismooth *at $x \in U$ if it is locally Lipschitz at x and if*

$$\lim_{\substack{V \in \partial_C X(x+td') \\ d' \to d, t \searrow 0}} \{Vd'\} \tag{53}$$

exists for every $d \in \mathbb{R}^n$. If X is semismooth at all $x \in U$, we call X semismooth (on U).

This notion is often hard to work with, but the following result comes in useful.

Theorem 3.4 ([Hin10, Thm. 2.9])**.** *Let $X : U \to \mathbb{R}^m$ be defined on the open set $U \subset \mathbb{R}^n$. Then, for every $x \in U$, the following statements are equivalent:*

(i) X is semismooth at x.

(ii) X is locally Lipschitz continuous at x, $X'(x; \cdot)$ exists, and, for every $V \in \partial_C X(x+d)$ it holds that

$$\|Vd - X'(x, d)\| = o(\|d\|) \qquad \text{as } d \to 0. \tag{54}$$

(iii) *F is locally Lipschitz continuous at $x, X'(x;\cdot)$ exists, and for every $V \in \partial_C X(x+d)$ it holds that*

$$\|X(x+d) - X(x) - Vd\| = o(\|d\|) \qquad \text{as } d \to 0. \tag{55}$$

Remark 3.5. *From (iii) it follows that C^1 functions are semismooth as well. In that case we have $\partial_C X(x) = \{\nabla X(x)\}$, which corresponds to the classical definition of the differential.*

Not only C^1 functions are semismooth. This also holds for piecewise C^1 functions, as stated by the following result. This result will be particularly useful later as an application to the non-smooth system we get for solving TV in both the real case (Sect. 3.4) and in the manifold case (Sect. 6.5).

Proposition 3.6 ([FP07, Prop. 7.4.6]). *Let $X : U \subset \mathbb{R}^n \to \mathbb{R}^m$, with U open, be piecewise semismooth near the $x \in U$. Then X is semismooth at x.*

Finally, there also exist stronger notions of semismoothness: *μ-order semismoothness*. For completeness, we will provide its definition as well [Hin10, Def. 2.6].

Definition 3.7. *Let $X : U \to \mathbb{R}^n$ be defined on the open set $U \subset \mathbb{R}^n$. Then, for $0 < \mu \leq 1$, X is called μ-*order semismooth *at $x \in U$ if X is locally Lipschitz at x, $X'(x,\cdot)$ exists, and, for every $V \in \partial_C X(x+d)$,*

$$\|Vd - X'(x,d)\| = \mathcal{O}\left(\|d\|^{1+\mu}\right) \qquad \text{as } d \to 0. \tag{56}$$

If X is μ-order semismooth at all $x \in U$, then we call X μ-order semismooth (on U*).*

For this stronger notion we have a similar characterization to Thm. 3.4.

Theorem 3.8 ([Hin10, Thm. 2.12]). *Let $X : U \to \mathbb{R}^m$ be defined on the open set $U \subset \mathbb{R}^n$. Then, for $x \in U$ and $0 < \mu \leq 1$, the following statements are equivalent:*

(i) *X is μ-order semismooth at x.*

(ii) *X is locally Lipschitz continuous at $x, X'(x,\cdot)$ exists, and, for every $V \in \partial_C X(x+d)$ it holds*

$$\|X(x+d) - X(x) - Vd\| = \mathcal{O}\left(\|d\|^{1+\mu}\right) \qquad \text{as } d \to 0. \tag{57}$$

3.2.2 Fast Local Convergence for Semismooth Functions

With the notion of semismoothness and the tools discussed in the previous section we can generalize the Newton algorithm by replacing the differential of the mapping X with the Clarke generalized differential. The resulting Semismooth Newton (SSN) algorithm is shown in Alg. 2. We obtain the following local convergence result:

Theorem 3.9 (Local convergence)**.** *Assume that x^* satisfies $X(x^*) = 0$, X is locally Lipschitz and semismooth at x^* and all $V \in \partial_C X(x^*)$ are nonsingular. Then the iteration in Alg. 2 is well-defined and converges superlinearly to x^* in a neighborhood of x^*. If in addition X is μ-order semismooth at x^*, then the convergence is of order $1 + \mu$.*

Proof. Local convergence follows from [QS93, Thm. 3.2]. The superlinearity is not explicitly stated in [QS93, Thm. 3.2], but is shown in its proof. □

Algorithm 2 Semismooth Newton

```
Initialization: $x^0 \in \mathbb{R}^n, k := 0$
while not converged do
    Choose any $V(x^k) \in \partial_C X(x^k)$
    Solve $V(x^k)d^k = -X(x^k)$
    $x^{k+1} := x^k + d^k$
    $k := k + 1$
end while
```

3.3 A Higher-order Primal-dual Method

In this section we will work towards a higher order method involving SSN for solving the problem

$$\inf_{x \in \mathbb{R}^n} \{f(x) + g(Ax)\}. \tag{58}$$

Assuming that the functions are proper, lower semi-continuous and convex we have seen in Sect. 2.2 that we can rewrite the equation into the saddle-point problem

$$\inf_{x \in \mathbb{R}^n} \sup_{y \in \mathbb{R}^m} \{f(x) + \langle Ax, y\rangle - g^*(y)\}, \tag{59}$$

and the equivalence to solving the optimality system

$$0 \in \partial f(x) + A^\top y, \tag{60}$$

$$0 \in \partial g^*(y) - Ax, \tag{61}$$

which can be rewritten in terms of proximal maps,

$$x = \mathrm{prox}_{\sigma f}(x - \sigma A^\top y), \tag{62}$$

$$y = \mathrm{prox}_{\tau g^*}(y + \tau Ax), \tag{63}$$

where $\sigma, \tau > 0$. In the following the variables $x \in \mathbb{R}^n$ will be referred to as the *primal variables* and $y \in \mathbb{R}^m$ as the *dual variables.*

The idea is to apply SSN to the generally non-smooth non-linear system of equations

$$X(x, y) = \begin{pmatrix} x - \mathrm{prox}_{\sigma f}(x - \sigma A^\top y) \\ y - \mathrm{prox}_{\tau g^*}(y + \tau Ax) \end{pmatrix} = 0 \tag{64}$$

in order to solve the original problem (58). For that, let

$$K \in \partial_C \operatorname{prox}_{\sigma f}\left(x - \sigma A^\top y\right), \tag{65}$$

$$H \in \partial_C \operatorname{prox}_{\tau g^*}(y + \tau A x). \tag{66}$$

By the chain rule, we find that

$$\begin{pmatrix} I-K & \sigma K A^\top \\ -\tau H A & I-H \end{pmatrix} \in \partial_C X(x,y) \tag{67}$$

and the Newton system becomes

$$\begin{pmatrix} I-K & \sigma K A^\top \\ -\tau H A & I-H \end{pmatrix} \begin{pmatrix} d_x \\ d_y \end{pmatrix} = - \begin{pmatrix} x - \operatorname{prox}_{\sigma f}(x - \sigma A^\top y) \\ y - \operatorname{prox}_{\tau g^*}(y + \tau A x) \end{pmatrix}. \tag{68}$$

It is not directly clear whether the non-smooth system in (64) satisfy the conditions in Thm. 3.9. It turns out that this is not trivial. In [RLV17, Sect. 4.2.3], Lipschitz continuity and positive semi-definiteness at the solution were shown. However, actual invertibility and additionally the semismoothness seems to be dependent on the application. This brings us to the next topic: a case study into ℓ^2-TV functionals.

3.4 Application to ℓ^2-TV-like Functionals*

In this section we will apply the Semismooth Newton method to the case of Total Variation image denoising.

3.4.1 The Newton System for TV

Let $d_1, d_2 \in \mathbb{N}$ be the dimensions of the image, let $h \in \mathbb{R}^{d_1 \times d_2}$ be the data. We are interested in solving the isotropic ($q = 2$) and anisotropic ($q = 1$) discrete ROF model

$$\inf_{x \in \mathbb{R}^{d_1 \times d_2}} \left\{ \frac{1}{2\alpha} \|x - h\|_2^2 + \|Tx\|_{q,1} \right\}, \tag{69}$$

where $\alpha > 0$ and $T : \mathbb{R}^{d_1 \times d_2} \to \mathbb{R}^{d_1 \times d_2 \times 2}$ is the (forward) finite difference operator with grid spacing 1 defined as

$$(Tx)_{i,j,k} = \begin{cases} 0 & \text{if } i = d_1 \text{ and } k = 1 \\ 0 & \text{if } j = d_2 \text{ and } k = 2 \\ x_{i+1,j} - x_{i,j} & \text{if } i < d_1 \text{ and } k = 1 \\ x_{i,j+1} - x_{i,j} & \text{if } j < d_2 \text{ and } k = 2 \end{cases} \tag{70}$$

and where

$$\|Tx\|_{q,1} = \sum_{i,j=1}^{d_1,d_2} \left(|(Tx)_{i,j,1}|^q + |(Tx)_{i,j,2}|^q\right)^{\frac{1}{q}}. \tag{71}$$

Using duality of the $\|\cdot\|_{q,1}$ norm we can rewrite the minimization problem as the saddle-point problem

$$\inf_{x\in\mathbb{R}^{d_1\times d_2}} \sup_{y\in\mathbb{R}^{d_1\times d_2\times 2}} \{\frac{1}{2\alpha}\|x-h\|_2^2 + \langle Tx, y\rangle - \iota_{B_{q^*}}(y)\}, \tag{72}$$

where q^* is such that $\frac{1}{q}+\frac{1}{q^*}=1$,

$$B_{q^*} := \left\{y\in\mathbb{R}^{d_1\times d_2\times 2} \mid \|y\|_{q^*,\infty}\leq 1\right\} = \left\{y\in\mathbb{R}^{d_1\times d_2\times 2} \mid \max\|y_{i,j,:}\|_{q^*}\leq 1\right\} \tag{73}$$

and

$$\iota_{B_{q^*}}(y) := \begin{cases} 0 & \text{if } y\in B_{q^*}, \\ \infty & \text{if } y\notin B_{q^*}. \end{cases} \tag{74}$$

This fits into the model (59) if we set

$$f(x) = \frac{1}{2\alpha}\|x-h\|_2^2 \quad \text{and} \quad g_q^*(y) = \iota_{B_{q^*}}(y). \tag{75}$$

Then we have for the data fidelity term

$$\operatorname{prox}_{\sigma f}(x) = \arg\min_u \left\{f(u) + \frac{1}{2\sigma}\|u-x\|_2^2\right\} \tag{76}$$

$$\Leftrightarrow \quad 0 = \nabla f(u) + \frac{1}{\sigma}(u-x) \tag{77}$$

$$\Leftrightarrow \quad 0 = \frac{1}{\alpha}(u-h) + \frac{1}{\sigma}(u-x) \tag{78}$$

$$\Leftrightarrow \quad \frac{\sigma+\alpha}{\sigma\alpha}u = \frac{1}{\sigma}x + \frac{1}{\alpha}h \tag{79}$$

and we find

$$\operatorname{prox}_{\sigma f}(x) = \frac{1}{\alpha+\sigma}(\alpha x + \sigma h) \quad \Rightarrow \quad \partial_C \operatorname{prox}_{\sigma f}(x) = \nabla \operatorname{prox}_{\sigma f}(x) = \frac{\alpha}{\alpha+\sigma}I, \tag{80}$$

where we used that $\operatorname{prox}_{\sigma f}(x)$ is smooth. Hence, we have to choose $K = \frac{\alpha}{\alpha+\sigma}I$ in the Newton matrix (67).

For the regularizer,

$$\operatorname{prox}_{\tau g_q^*}(y) = \arg\min_v \left\{g_q^*(v) + \frac{1}{2\tau}\|v-y\|_2^2\right\} \tag{81}$$

$$= \arg\min_v \left\{\iota_{B_{q^*}}(v) + \frac{1}{2\tau}\|v-y\|_2^2\right\} \tag{82}$$

$$\Leftrightarrow \quad v = \Pi_{B_{q^*}}(y), \tag{83}$$

where $\Pi_{B_{q^*}}$ is the projection onto the (convex) set B_{q^*}. Hence, we find

$$\operatorname{prox}_{\tau g_1^*}(y) = \left((\max\{1, |y_{i,j,k}|\})^{-1} y_{i,j,k}\right)_{i,j,k} \tag{84}$$

and

$$\operatorname{prox}_{\tau g_2^*}(y) = \left(\left(\max\left\{1, \|y_{i,j,:}\|_2\right\}\right)^{-1} y_{i,j,k} \right)_{i,j,k}. \tag{85}$$

We see that the components of $\operatorname{prox}_{\tau g_q^*}(y)$, i.e., $\|y_{i,j,:}\|_{q^*}$ smaller or larger than 1, are piecewise C^1. Therefore, we can use ordinary differentiation on each region. On the boundary we choose the value of the differential corresponding to the inner region. We obtain a map $H_q(v) \in \partial_C \operatorname{prox}_{\tau g_q^*}(x)$ with

$$(H_1(v)y)_{i,j,k} = \begin{cases} 0 & \text{if } i = d_1 \text{ and } k = 1 \\ 0 & \text{if } j = d_2 \text{ and } k = 2 \\ y_{i,j,k} & \text{if } |v_{i,j,k}| \le 1 \\ 0 & \text{if } |v_{i,j,k}| > 1 \end{cases} \tag{86}$$

and

$$(H_2(v)y)_{i,j,k} = \begin{cases} 0 & \text{if } i = d_1 \text{ and } k = 1 \\ 0 & \text{if } j = d_2 \text{ and } k = 2 \\ y_{i,j,k} & \text{if } \|v_{i,j,:}\|_2 \le 1 \\ \frac{1}{\|v_{i,j,:}\|_2} \left(y_{i,j,k} - \frac{v_{i,j,k} v_{i,j,1}}{\|v_{i,j,:}\|_2^2} y_{i,j,1} - \frac{v_{i,j,k} v_{i,j,2}}{\|v_{i,j,:}\|_2^2} y_{i,j,2} \right) & \text{if } \|v_{i,j,:}\|_2 > 1 \end{cases} \tag{87}$$

Here the first two conditions ensure the boundary conditions $y_{i,j,k} = 0$ for $i = d_1$ and $k = 1$ or $j = d_2$ and $k = 2$. The third and fourth options concern non-boundary points, i.e., $i < d_1$ and $k = 1$ and $j < d_2$ and $k = 2$. A proof (for general manifolds) can be found in appendix A.1

For the semismoothness we know that $\operatorname{prox}_{\sigma f}(x)$ is semismooth since it is smooth. The semismoothness of $\operatorname{prox}_{\tau g_q^*}(y)$ follows from Prop. 3.6. So we are indeed justified to use the Newton system in (67) with $A = T$.

For superlinear convergence in Thm. 3.9 we also need invertibility of the Newton matrix at the solution. For TV this is problematic, as will be discussed in the following.

3.4.2 Non-invertibility Caused by Loops

We will now consider a $d_1 \times d_2$ 2D image. We can construct the matrix representation of T using the Kronecker product,

$$T = \begin{bmatrix} I_{d_2} \otimes T_{d_1} \\ T_{d_2} \otimes I_{d_1} \end{bmatrix} \tag{88}$$

from the $n-1 \times n$ matrix representations T_n of the 1D forward finite difference operators with Neumann boundary conditions,

$$T_n = \begin{bmatrix} -1 & 1 & & & & \\ & -1 & 1 & & \varnothing & \\ & & \ddots & \ddots & & \\ & \varnothing & & -1 & 1 & \\ & & & & -1 & 1 \end{bmatrix}. \tag{89}$$

Note that the final (zero) row for the boundary points has been eliminated for convenience. As an example, consider the 3×3 grid in Fig. 6 as representation of a 3×3 image. Then, the adjoint $T^\top$ is

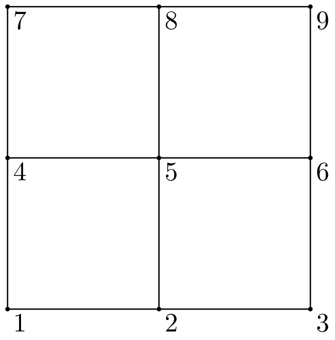


Figure 6: A 3×3 grid.

$$T^\top = \begin{bmatrix} -1 & 0 & 0 & 0 & 0 & 0 & -1 & 0 & 0 & 0 & 0 & 0 \\ 1 & -1 & 0 & 0 & 0 & 0 & 0 & -1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & 0 & 0 & 0 \\ 0 & 0 & -1 & 0 & 0 & 0 & 1 & 0 & 0 & -1 & 0 & 0 \\ 0 & 0 & 1 & -1 & 0 & 0 & 0 & 1 & 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & -1 \\ 0 & 0 & 0 & 0 & -1 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & -1 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}. \tag{90}$$

Now consider the vectors v_1 and v_2, both of which lie in the kernel of $T^\top$:

$$v_1 = \begin{bmatrix} 1 & 0 & -1 & 0 & 0 & 0 & -1 & 1 & 0 & 0 & 0 & 0 \end{bmatrix}^\top, \tag{91}$$

$$v_2 = \begin{bmatrix} 1 & 1 & -1 & 0 & 0 & -1 & -1 & 0 & 1 & 0 & -1 & 1 \end{bmatrix}^\top. \tag{92}$$

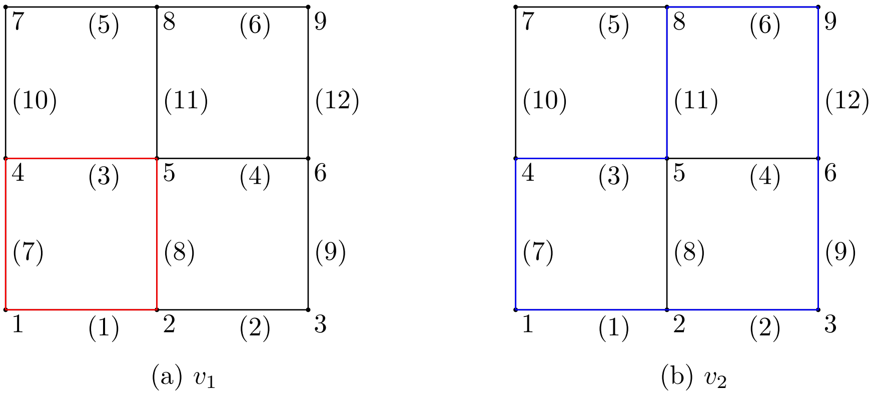
This means that in the case that $H = \mathrm{I}$ we have the Newton system

$$\begin{bmatrix} \frac{\sigma}{\alpha+\sigma} & \frac{\alpha\sigma}{\alpha+\sigma} T^\top \\ -\tau T & 0 \end{bmatrix}, \tag{93}$$

which has zero eigenvectors $v_1' = [0, v_1]^\top$ and $v_2' = [0, v_2]^\top$ and is therefore non-invertible.

Since we can associate the edges between grid-points as dual variables, we can visualize the eigenvectors (Fig. 7). The eigenvectors correspond to loops in the system coloured in red (v_1) and blue (v_2). One can actually check that every loop corresponds to an eigenvector with eigenvalue 0.

Remark 3.10. *This phenomenon does not occur in the 1D case, but emerges in higher dimensions.*



(a) v_1

(b) v_2

Figure 7: The corresponding zero eigenvectors represented as loops over the 3 × 3 grid.

Although this case might seem artificial, it is the cause of non-invertibility of the Newton matrix in real-world problems. Consider the case that somewhere in a $d_1 \times d_2$ grid we have 4 neighbouring grid-points, say $x_{i,j}$, $x_{i+1,j}$, $x_{i,j+1}$ and $x_{i+1,j+1}$, in a square that will have exactly the same value in the optimum. This is realistic, since total variation tends to give piecewise constant solutions. Now, let $l_1 = (i, j, 1)$, $l_2 = (i, j+1, 1)$, $l_3 = (i, j, 2)$ and $l_4 = (i+1, j, 2)$ be the indices of the corresponding dual variables. As the gradients are locally zero, we have $[Tx]_l = 0$. Since the dual variables must be in the unit ball we have $|y_l| \leq 1$ at the optimum. Then, at the solution for the ROF model we see

$$|y_l + \tau[Tx]_l| = |y_l| \leq 1 \qquad \Rightarrow \qquad H_{ll} = 1 \quad \text{for } l \in \{l_1, l_2, l_3, l_4\}. \tag{94}$$

In other words, we have four points that result in a loop and the system becomes singular in the same way as with the previous 3 × 3 example. As mentioned before, for TV we typically find piecewise constant solutions. So this prior knowledge in the form of TV regularization is actually the bottleneck for getting reasonable results with SSN. In other words, we need some extra help to work around this issue.

3.4.3 Invertibility by Dual Regularization

So far we tried to solve the saddle-point problem

$$\inf_{x \in \mathbb{R}^{d_1 \times d_2}} \sup_{y \in \mathbb{R}^{d_1 \times d_2 \times 2}} \left\{ \frac{1}{2\alpha} \|x - h\|_2^2 + \langle Tx, y \rangle - \iota_{B_{q^*}}(y) \right\}. \tag{95}$$

Instead we will now introduce a regularization term for the dual variable and solve

$$\inf_{x \in \mathbb{R}^{d_1 \times d_2}} \sup_{y \in \mathbb{R}^{d_1 \times d_2 \times 2}} \left\{ \frac{1}{2\alpha} \|x - h\|_2^2 + \langle Tx, y \rangle - \iota_{B_{q^*}}(y) - \frac{\beta}{2} \|y\|_2^2 \right\}. \tag{96}$$

The f part remains the same, but we have a new g_q^* we will call $\tilde{g}_q^*$ from now on. We need to know what $\operatorname{prox}_{\tau \tilde{g}^*}(x)$ is.

$$\begin{aligned}
\operatorname{prox}_{\tau\tilde{g}_q^*}(y) &= \arg\min_v \left\{ g_q^*(v) + \frac{1}{2\tau}\|v-y\|_2^2 \right\} && (97)\\
&= \arg\min_v \left\{ \iota_{B_{q^*}}(v) + \frac{\beta}{2}\|v\|_2^2 + \frac{1}{2\tau}\|v-y\|_2^2 \right\} && (98)\\
&= \arg\min_v \left\{ \iota_{B_{q^*}}(v) + \frac{\beta}{2}\|v\|_2^2 + \frac{1}{2\tau}\|v-y\|_2^2 - \frac{1}{2\tau}\left(1-\frac{1}{1+\beta\tau}\right)\|y\|_2^2 \right\} && (99)\\
&= \arg\min_v \left\{ \iota_{B_{q^*}}(v) + \frac{1+\beta\tau}{2\tau}\|v - \frac{1}{1+\beta\tau}y\|_2^2 \right\} && (100)\\
\Leftrightarrow \quad v &= \Pi_{B_{q^*}}\left(\frac{1}{1+\beta\tau}y\right), && (101)
\end{aligned}$$

where $\Pi_{B_{q^*}}$ is the projection onto B_{q^*}. Now, we find

$$\operatorname{prox}_{\tau\tilde{g}_1^*}(y) = \left(\left(\max\left\{1, \frac{|y_{i,j,k}|}{1+\beta\tau}\right\}\right)^{-1} \frac{y_{i,j,k}}{1+\beta\tau}\right)_{i,j,k} \tag{102}$$

and

$$\operatorname{prox}_{\tau\tilde{g}_2^*}(y) = \left(\left(\max\left\{1, \frac{\|y_{i,j,:}\|_2}{1+\beta\tau}\right\}\right)^{-1} \frac{y_{i,j,k}}{1+\beta\tau}\right)_{i,j,k}. \tag{103}$$

For the Clarke generalized differentials $\tilde{H}_1$ and $\tilde{H}_2$ we find

$$(\tilde{H}_1(v)y)_{i,j,k} = \begin{cases} 0 & \text{if } i = d_1 \text{ and } k = 1\\ 0 & \text{if } j = d_2 \text{ and } k = 2\\ \frac{1}{1+\beta\tau}y_{i,j,k} & \text{if } |v_{i,j,k}| \le 1+\beta\tau\\ 0 & \text{if } |v_{i,j,k}| > 1+\beta\tau \end{cases} \tag{104}$$

and

$$(\tilde{H}_2(v)y)_{i,j,k} = \begin{cases} 0 & \text{if } i = d_1 \text{ and } k = 1\\ 0 & \text{if } j = d_2 \text{ and } k = 2\\ \frac{1}{1+\beta\tau}y_{i,j,k} & \text{if } \|v_{i,j,:}\|_2 \le 1+\beta\tau\\ \frac{1}{\|v_{i,j,:}\|_2}\left(y_{i,j,k} - \frac{v_{i,j,k}v_{i,j,1}}{\|v_{i,j,:}\|_2^2}y_{i,j,1} - \frac{v_{i,j,k}v_{i,j,2}}{\|v_{i,j,:}\|_2^2}y_{i,j,2}\right) & \text{if } \|v_{i,j,:}\|_2 > 1+\beta\tau \end{cases} \tag{105}$$

We see that this resolves non-invertibility since the zero diagonal entries of $I - H$ in the Newton matrix (67) now become $1 - \frac{1}{1+\beta\tau} = \frac{\beta\tau}{1+\beta\tau}$.

Further, we see that as $\beta \to 0$ we approach the TV dual proximal map, but the problem becomes more and more ill-posed. For the condition number of the regularized Newton matrix V, we expect

$$\kappa(V) = \|V\|\|V^{-1}\| \approx C\frac{1+\beta\tau}{\beta} \approx \frac{C}{\beta} \quad \text{if } \beta \ll 1 \tag{106}$$

for some constant $C > 0$. As $\beta \to 0$ the condition number tends to increase. A large condition number has the major drawback that rounding errors in either the matrix or the vector can amplify and make it much more complicated to find an accurate result. This trade-off between accuracy and convergence will be topic of experiments in Sect. 3.5.

Remark 3.11. *One should note that there are other options to resolve the non-invertibility. The approach discussed above has the primary advantage that it is easy to implement and, as it turns out, is convenient to generalize to the manifold setting. Another option would be to use another discretization of the gradient [LLWS13] in the ROF model or use a pseudo-inverse for the Newton matrix. Due to lack of theoretical motivation of the latter two approaches to work with SSN, these have not been investigated in this work.*

3.5 Numerical Experiments*

In this section we will explore the behaviour of the Semismooth Newton (SSN) method through several numerical experiments with ℓ^2-TV:

$$\inf_{x\in\mathbb{R}^{d_1\times d_2}} \left\{ \frac{1}{2\alpha}\|x-h\|_2^2 + \|Tx\|_{q,1} \right\}. \tag{107}$$

The key questions we try to answer are

- Do we suffer from semi-definiteness in practice, i.e., is dual regularization necessary?
- Can we get quantitatively better performance using SSN than when using PDHG?
- Can we get a qualitatively better solution with SSN than with PDHG?

We will try to answer these questions through 3 experiments: a proof of concept for a 1D problem with known minimizer, runtime analysis and further exploration of the parameter β.

In the following sections our goal is to solve

$$X(x,y) := \begin{pmatrix} x - \operatorname{prox}_{\sigma f}(x-\sigma T^\top y) \\ y - \operatorname{prox}_{\tau \tilde{g}^*}(y+\tau Tx) \end{pmatrix} = 0 \tag{108}$$

for $f(x) = \frac{1}{2\alpha}\|x-h\|_2^2$ and $\tilde{g}_q^*(y) = \iota_{B_2}(y) + \frac{\beta}{2}\|y\|_2^2$ (so regularized isotropic TV) and $\sigma, \tau > 0$. Moreover, throughout the sections we will use the relative error

$$\epsilon_{rel}^k := \frac{\|X(x^k,y^k)\|_2}{\|X(x^0,y^0)\|_2} \tag{109}$$

as an measure for convergence.

All numerical experiments are implemented in Julia version 1.3.0 and run on a HP ZBook, 2.4 GHz Intel Core i7, 8 GB RAM.

3.5.1 Signal with known minimizers

For a proof of concept we will consider a 1-dimensional piecewise constant signal (Fig. 8)

$$h \in \mathbb{R}^{20}, \quad h_i := \begin{cases} a & \text{if } i \leq 10 \\ b & \text{if } i > 10 \end{cases}. \tag{110}$$

Remark 3.12. *For this signal we know the ℓ^2-TV minimizer in (69): for $\alpha > 0$ and $a > b$ the minimizer x^* is given by*

$$x_i^* = \begin{cases} a - \min\{\frac{1}{2}, \frac{\alpha}{10}\frac{1}{a-b}\}(a-b) & \textit{if } i \leq 10 \\ b + \min\{\frac{1}{2}, \frac{\alpha}{10}\frac{1}{a-b}\}(a-b) & \textit{if } i > 10 \end{cases} . \tag{111}$$

A proof (for general manifolds) can be found in appendix A.2.

Furthermore, note that $d_2 = 1$ and the operator T reduces to the 1-dimensional difference operator of (89) (i.e., we have $T : \mathbb{R}^{d_1} \to \mathbb{R}^{d_1 - 1} \times \{0\}$). This also means that the isotropic ($q = 2$) and anisotropic ($q = 1$) cases reduce to the same functional.

Moreover, in the one-dimensional case we do not have the loop issue as discussed in Sect. 3.4.2. Empirically this seems to resolve the non-invertibility, as we have no issues solving the Newton system for $\beta = 0$.

Choosing $a = 3$, $b = 1$ for the signal, we solve ℓ^2-TV with $\alpha = 1$ using SSN. We choose $\sigma = \tau = \frac{1}{2}$ and start from $x^0 = h$ and $y^0 = 0$.

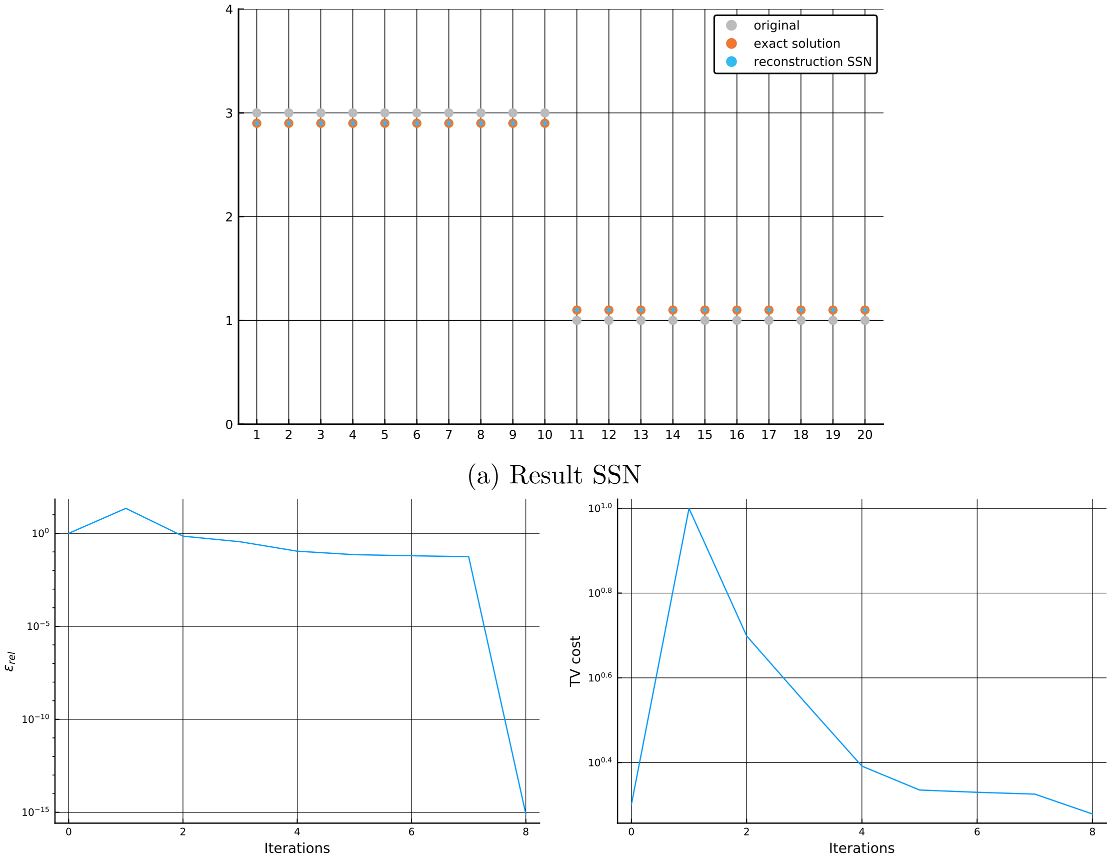
SSN converges after 8 iterations in 0.188 seconds superlinearly to a solution of $X(x, y) = 0$ with an accuracy of $\epsilon_{rel} = 10^{-16}$, which is in line with what we expected from theory.

3.5.2 Comparison of algorithms for solving regularized TV

For this experiment and the next we use a patch of the Lena image. The original image and the noisy image are shown in Fig. 9. Both in this and in the next section we added Gaussian noise with variance $\delta^2 = 0.04$. In the following we will denoise the image with regularized isotropic ℓ^2-TV with $\alpha = 0.2$ and $\sigma = \tau = \frac{1}{2}$. The dual regularization parameter β will differ from case to case. In both cases we start from the data for the primal, and from the zero vector for the dual variables.

For this part we are particularly interested in getting insight into the runtime performance of SSN. We will take two values for β, 10^{-3} and 10^{-6}, and compare performance of SSN with PDHG. We expect that PDHG will be faster at the start but will suffer from slow tail convergence. Hence, SSN should give better performance for higher accuracy solutions.

For our numerical experiment, we measure the (CPU) runtime until the algorithms reach $\epsilon_{rel} \in \{10^{-2}, 10^{-4}, 10^{-6}\}$. The runtimes for $\beta = 10^{-3}$ are shown in Tab. 1. The solutions of PDHG and SSN at $\epsilon_{rel} = 10^{-6}$ along with the progression of the relative error and the isotropic ℓ^2-TV-cost are shown in Fig. 10.

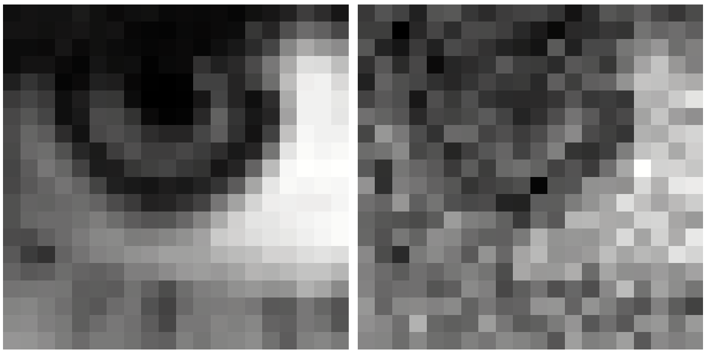


(a) Result SSN

(b) The progression of the relative error

(c) The progression of the ℓ^2-TV cost

Figure 8: The results of an experiment in which SSN is used to solve for the ℓ^2-TV minimizer of a signal with known minimizer. The SSN solution converges to the exact solution of the minimization problem and converges superlinearly in the vicinity of the minimizer.



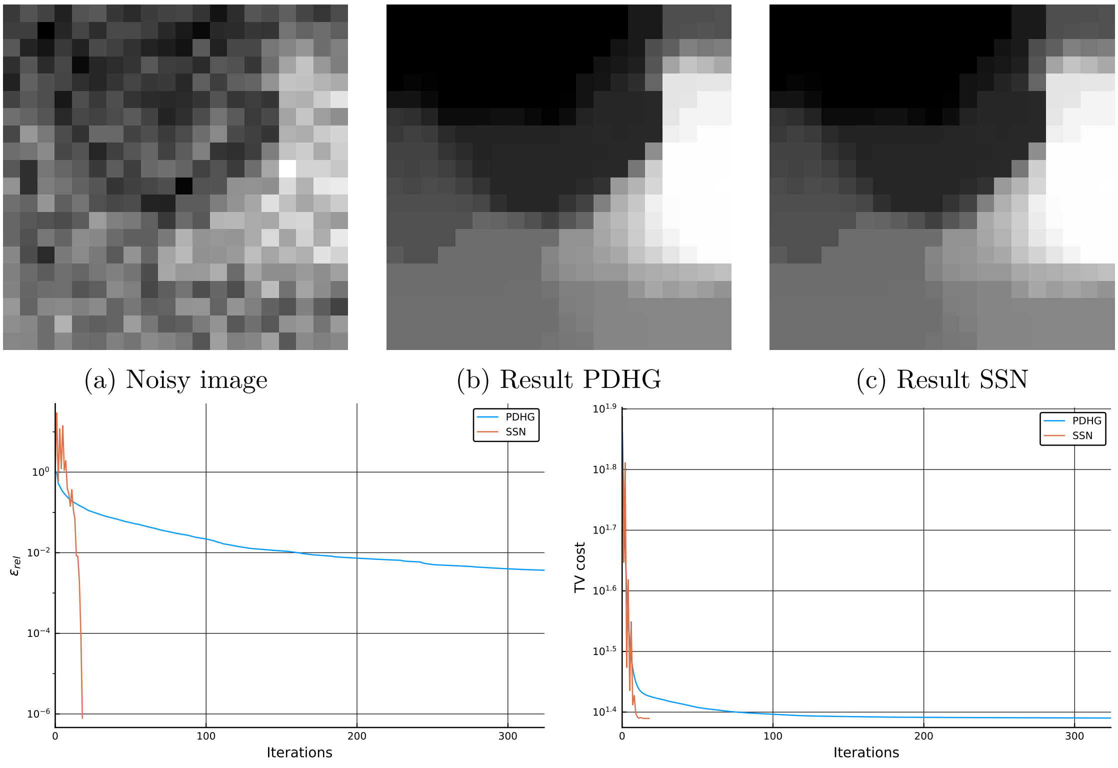
(a) Original Lena eye image

(b) Noisy Lena eye image

Figure 9: The original and the noisy eye of the Lena image.

$\beta = 10^{-3}$	$\epsilon_{rel} = 10^{-2}$		$\epsilon_{rel} = 10^{-4}$		$\epsilon_{rel} = 10^{-6}$	
Method	Time	# Iterations	Time	# Iterations	Time	# Iterations
PDHG	**1.188**	161	13.593	1792	61.282	7938
SSN	4.625	14	**5.578**	17	**5.843**	18

Table 1: The runtimes and iteration counts of SSN and PDHG with $\beta = 10^{-3}$ until $\epsilon_{rel} \leq \{10^{-2}, 10^{-4}, 10^{-6}\}$ is reached. SSN reaches higher accuracies faster than PDHG, but the latter is preferable for low accuracy solutions.



(a) Noisy image

(b) Result PDHG

(c) Result SSN

(d) The progression of the relative error

(e) The progression of the isotropic ℓ^2-TV cost

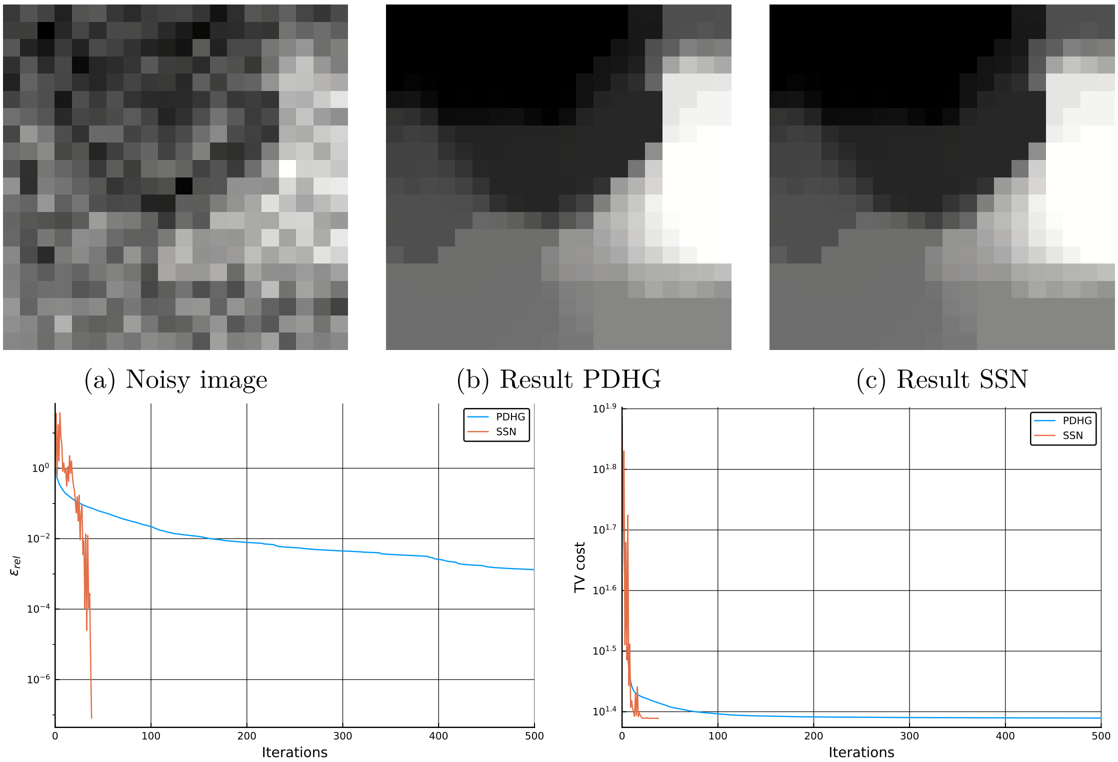
Figure 10: The results of a performance comparison experiment, in which the Lena eye is denoised using the dual regularized isotropic ℓ^2-TV model with $\beta = 10^{-3}$ and $\alpha = 0.2$. SSN converges superlinearly, whereas PDHG suffers from slow tail convergence.

For $\beta = 10^{-6}$ the runtimes are shown in Tab. 2. The solutions of PDHG and SSN at $\epsilon_{rel} = 10^{-6}$ along with the progression of the relative error and the (isotropic) ℓ^2-TV-cost are shown in Fig. 11.

The results confirm what we expect. If a relative error below approximately 10^{-2}-10^{-4} is requested, SSN performs better than PDHG for both values of β. Furthermore, we see the superlinear convergence more clearly than in the 1D case.

$\beta = 10^{-6}$	$\epsilon_{rel} = 10^{-2}$		$\epsilon_{rel} = 10^{-4}$		$\epsilon_{rel} = 10^{-6}$	
Method	Time	# Iterations	Time	# Iterations	Time	# Iterations
PDHG	**1.375**	162	27.563	3419	750.547	84156
SSN	8.547	26	**11.437**	33	**13.047**	38

Table 2: The runtimes and iteration counts of SSN and PDHG for $\beta = 10^{-6}$ until $\epsilon_{rel} \leq \{10^{-2}, 10^{-4}, 10^{-6}\}$ is reached. SSN reaches higher accuracies faster than PDHG, but the latter performs better for low accuracy solutions.



(a) Noisy image

(b) Result PDHG

(c) Result SSN

(d) The progression of the relative error

(e) The progression of the isotropic ℓ^2-TV cost

Figure 11: The results of a performance comparison experiment, in which the Lena eye is denoised using the dual regularized isotropic ℓ^2-TV model with $\beta = 10^{-6}$ and $\alpha = 0.2$. SSN converges superlinearly, whereas PDHG suffers from slow tail convergence.

3.5.3 Role of the dual regularization parameter β

In the previous experiment, we smoothed the TV term as in Sect. 3.4.3 in order to avoid a singular Newton matrix, i.e., we had to modify the energy. This leaves the question whether SSN can also perform well for the non-smoothed version.

For the final part our central question is whether we can get a qualitatively better solution with SSN than with PDHG. Our aim is to see whether it is possible to get a lower ℓ^2-TV energy with a small β and a very accurate SSN solution to the regularized problem than with a less accurate PDHG solution to the non-regularized problem. The main motivation for this is that PDHG has slow tail convergence and therefore might effectively stall at a sub-optimal energy.

In order to investigate the possibility of getting better results with SSN and a small β, the first step is to look into the role of β itself. In Sect. 3.4.3 we discussed that as $\beta \to 0$ we expect the solution to converge to that of (normal) ℓ^2-TV. However, we also expect the matrix to become more and more ill-conditioned causing trouble with convergence due to numerical underflow: rounding errors prohibit us from obtaining an accurate result. Once we can pick a good β, the second step will be to compare results for a SSN solving the regularized ℓ^2-TV problem and PDHG solving the (normal) ℓ^2-TV problem.

Finding a good β

To explore the trade-off between accuracy and convergence, we run the following experiment in order to find a good β. For different β we denoise the Lena eye by minimizing the regularized ℓ^2-TV functional with $\alpha = 0.2$ (as discussed in the previous section). The algorithm is assumed to have converged if $\epsilon_{rel} < 10^{-10}$. From observations in Tab. 1 and Tab. 2 from the previous experiment we can assume that the runtime will be about same order of magnitude as the runtime for solving the problem with PDHG until $\epsilon_{rel} = 10^{-4}$.

From the results in Tab. 3 we observe, in line with our expectations, the following:

(i) For $\beta \leq 10^{-7}$ SSN does not seem to converge, but until that point we find an decreasing ℓ^2-TV cost.

(ii) For lower β we eventually get a lower ℓ^2-TV energy than with PDHG.

To start off with (i) we see in Fig. 12 the behaviour of the relative error more clearly. For very small β the algorithm seems unable to reach a higher accuracy. Considering the non-amplifying behaviour of the relative error progression a possible cause could be rounding error taking. This would comply with our prediction that the condition number would cause numerical underflow for too small values for β. However, we note that an accuracy of about 10^{-7} is still achieved. Next, for (ii) our expectations also seem to be fulfilled. As β gets lower TV_{SSN^β} decreases and eventually finds a better solution with a lower energy than PDHG.

The remaining question now is whether we can not only find a lower cost, but also do this in less time than PDHG.

SSN outperforming PDHG

One might wonder if there is a moment PDHG performs better again. So for the next step we will use $\beta = 10^{-6}$ as a benchmark and run PDHG with decreasing relative error tolerance. The ℓ^2-TV cost will be compared with that of SSN along with the runtimes.

In Tab. 4 we see that indeed PDHG will eventually take the lead in terms of energy if we run the algorithm until $\epsilon_{rel} = 10^{-6}$. However, whereas SSN needs 12.751 seconds,

β	TV_{SSN^β}	$\mathrm{TV}_{SSN^\beta} - \mathrm{TV}_{PDHG}$	# SSN
10^{-1}	27.771273	3.2714496	7
10^{-2}	24.785553	0.2857292	11
10^{-3}	24.523937	0.02411314	20
10^{-4}	24.500692	0.0008689283	33
10^{-5}	24.498486	-0.0013390831	46
10^{-6}	24.498262	-0.0015609582	39
10^{-7}	24.498936	-0.0008878283	-
10^{-8}	24.498934	-0.0008907767	-

Table 3: The results for solving the dual regularized ℓ^2-TV problem for different values for β. For values of $\beta \geq 10^{-6}$ SSN converges. For $\beta = 10^{-7}$ and $\beta = 10^{-8}$ SSN was terminated after 1000 iterations. For small values of β, solving the regularized ℓ^2-TV problem with SSN can yield a lower energy than solving the non-smoothed problem with PDHG.

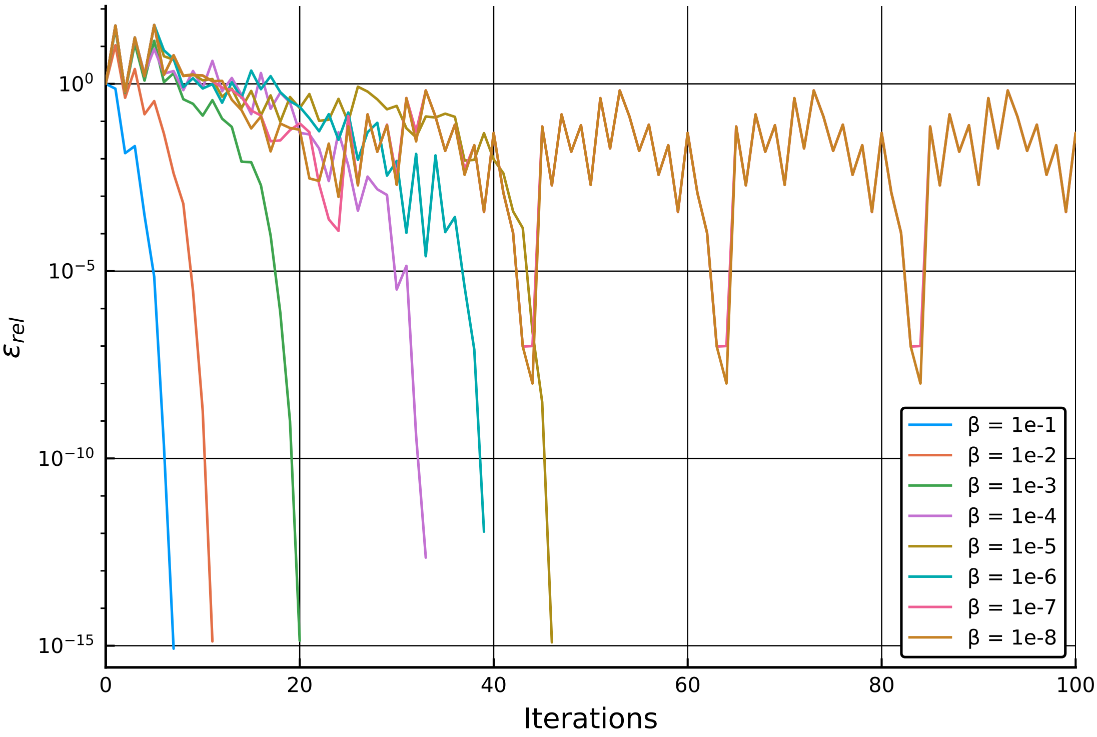


Figure 12: The results of an experiment, in which the relative error of SSN for dual regularized isotropic ℓ^2-TV will be compared for different values for β. For the two smallest values for β, numerical underflow prohibits high accuracy.

for PDHG it takes up to 53 times longer to do this. So indeed it could be a good idea to use SSN even in the case of ℓ^2-TV.

Although these experiments give a clear indication of the possibilities of SSN on its own and compared to PDHG, there is more to this when it comes to practical problems. These will be the topic of the following section.

ϵ_{rel}	TV_{PDHG}	$\mathrm{TV}_{SSN^\beta} - \mathrm{TV}_{PDHG}$	t_{PDHG}	$\frac{t_{PDHG}}{t_{SSN^\beta}}$
10^{-4}	24.499825	-0.0015609582	26.078	2.045173
10^{-5}	24.498367	-0.00010436704	92.204	7.231119
10^{-6}	24.498251	1.2367425e-5	684.124	53.652576

Table 4: The results of an experiment, in which we compare the solutions of ℓ^2-TV by PDHG to those of dual regularized ℓ^2-TV by SSN. In practical cases, i.e., PDHG reaching an accuracy of $\epsilon_{rel} \leq 10^{-5}$, SSN obtains a solution with a lower energy than PDHG even though it solves a smoothed version of the problem. In order for PDHG to find a lower energy than SSN, we need a much more accurate solution $\epsilon_{rel} = 10^{-6}$ and hence a longer runtime.

3.6 Towards SSN for Manifold-valued Data*

At this point, we would like to summarize the main conclusions that can be drawn in the Euclidean setting, in order to prepare for the extension to manifolds in the remainder of this thesis.

Best Practices
From the experiments it is clear that the introduction of a dual regularization term is a valuable asset to solving TV with SSN for 2D problems. Not only is it possible to achieve local superlinear convergence, but we are even able to get better results than PDHG when solving normal TV (i.e., with $\beta = 0$).

Full Potential of SSN
One should realize that these experiments do not show the potential of both algorithms to their full extent. PDHG is very easily parallellizable and SSN could benefit from a good iterative method when solving the Newton system. The latter is made possible by the introduction of the inexact Semismooth Newton method [MQ95]. The interesting case now would be to look into actual (large scale) real world problems such as denoising a 256×256 (or even 512×512) image.

However, in order to run a fair experiment we should realize both the parallel PDHG and the accelerated SSN. One option could be a GMRES extension to SSN. However, during preliminary experiments, we found developing an efficient preconditioner to be a major hurdle, therefore we did not follow this path further.

Comparison to Prior Work with SSN
As mentioned in the introduction of this work, there was another approach in [RLV17] to SSN by adding a small positive multiple of the identity matrix to the Newton matrix. In our experiments we did not focus on comparing the results of our dual regularized approach and the latter for the simple reason that we were interested in finding a mathematically solid idea that could be generalized to the manifold case. Whether our implementation or the one in [RLV17] is actually preferable in the Euclidean case remains open.

A Potential Globalization Scheme
We also considered a globalization scheme that might be feasible for the manifold case. We particularly looked into the Adaptive Semismooth Newton method (ASSN) from [XLWZ18]. Whereas from numerical results we indeed observed convergence, the theory of the algorithm partially relies on the fixed point map $X : \mathbb{R}^d \to \mathbb{R}^d$ to be α-averaged (see Sect. 1.1 of [XLWZ18] for a definition), which in our application remains to be shown. Reason to discard it, was its performance. Preliminary experiments showed that as SSN needs more iterations as $\beta \to 0$, so did ASSN. However, whereas SSN stayed below the 50 iterations for $\beta = 10^{-6}$, the globalized ASSN method did not enter the superlinear convergence region after 10.000 iterations. Therefore, we focus on local convergence aspects in the following.

Chapter 4: Preliminaries II: Manifolds and Riemannian Geometry

In this chapter we will discuss differential geometry and Riemannian geometry. All of the discussed topics in Sect. 4.1 and Sect. 4.2 can be found in [Lee13, Car92, CE08].

This chapter is organized as follows. In Sect. 4.1 basic notions from differential geometry will be discussed. In particular, the path towards Riemannian geometry will be paved and the application of Jacobi fields will be discussed. In Sect. 4.2 we will look into the S^d and $\mathcal{P}(d)$ manifold as examples and discuss the relevant mappings that are required for numerical purposes.

4.1 Differential Geometry and Riemannian Geometry

Whereas smooth manifolds are still just a step more general than the concept of a vector space, the nature of its notions are very different from the linear case. For the former we can oversee the whole space at all time, but for manifolds we are often limited to a local neighbourhood. Moreover, mappings and vector fields call for generalized notions.

4.1.1 Basic Notions in Differential Geometry

In this section we will develop the basic notions of differential geometry and work towards Riemannian geometry by defining the metric tensor as a first milestone.

Smooth Manifolds

In differential geometry we typically look at non-linear spaces. In these spaces, useful characteristics of vector spaces, in particular addition and scalar multiplication, are missing. In order to overcome this, we want the space to locally look like $\mathbb{R}^d$, i.e., we want it to be locally homeomorphic. For completeness, we give the following definition.

Definition 4.1 (locally homeomorphic)**.** *We say a set $\mathcal{M}$ is* locally homeomorphic *to $\mathbb{R}^d$ if each $p \in \mathcal{M}$ has an open neighbourhood U that is homeomorphic to some open subset of $\mathbb{R}^d$. In particular, there exists a mapping $\varphi : U \to \varphi(U) \subset \mathbb{R}^d$ such that φ is invertible and both φ as its inverse are C^0 maps.*

These homeomorphisms are often called charts and these are combined in an atlas, covering up the whole manifold (Fig. 13).

Definition 4.2 (coordinate chart)**.** *If $\varphi : U \to \varphi(U) \subset \mathbb{R}^d$ is a local homeomorphism on a set $\mathcal{M}$, then we say that (U, φ) is a* (coordinate) chart *for $\mathcal{M}$.*

Definition 4.3 (atlas)**.** *If $\varphi : U \to \varphi(U) \subset \mathbb{R}^d$ is a local homeomorphism on a set $\mathcal{M}$, then we define an* atlas *for $\mathcal{M}$ as a collection $\{(U_\alpha, \varphi_\alpha)\}$ of coordinate charts which covers $\mathcal{M}$, in the sense that $\cup U_\alpha = \mathcal{M}$.*

Next, we also need some geometrical regularity: we need the space to be second countable and Hausdorff[3].

Definition 4.4 (topological manifold)**.** *A* topological d-manifold *is a d-dimensional second countable Hausdorff space $\mathcal{M}$ that is locally homeomorphic to $\mathbb{R}^d$.*

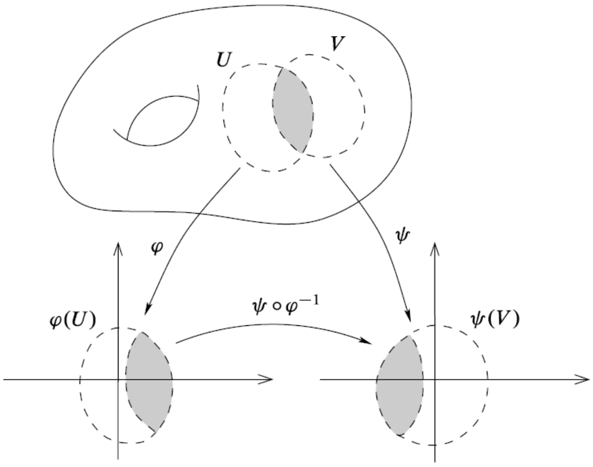


Figure 13: An illustration of coordinate charts on a manifold [Lee13]. In order to consider different parts of the manifolds, different charts are needed.

For differential geometry, there is more. We want a differentiable manifold. So far we have only considered the "geometry" part, but not the "differential" part. Now, the latter comes in to play when considering a certain smoothness of the coordinate charts.

Definition 4.5 (C^r-compatible coordinate charts, C^r-atlas)**.** *Fix $r \in \{0, 1, 2, \dots, \infty\}$. We say*

(i) Two charts (U, φ) and (V, ψ) for a manifold $\mathcal{M}$ are C^r-compatible *if the transition functions $\psi \circ \varphi^{-1}$ and $\varphi \circ \psi^{-1}$ are C^r maps.*

(ii) A C^r-atlas *for $\mathcal{M}$ is a collection of C^r-compatible coordinate charts which covers $\mathcal{M}$. A C^r-structure on $\mathcal{M}$ is a maximal C^r-atlas, that is an atlas $\mathcal{U} = \{(U_\alpha, \varphi_\alpha)\}$ such that every coordinate chart (V, ψ), which is compatible with all the $(U_\alpha, \varphi_\alpha)$ is already contained in $\mathcal{U}$.*

Finally, we are ready to define a smooth manifold.

Definition 4.6 (C^r-manifold)**.** *A* C^r-manifold *is a topological manifold $\mathcal{M}$ with a C^r-structure. A chart for a smooth manifold will mean a chart in the given smooth structure.*

From here on out, a smooth manifold will be referred to as a C^∞-manifold, i.e., we have C^∞-charts.

[3] These are technical notions that are often satisfied. In this work, we will leave out the details.

Mappings and Vector Fields on Manifolds
Smooth functions between manifolds can be interpreted in terms of coordinate charts (Fig. 14).

Definition 4.7 (smooth functions)**.** *Let $\mathcal{M}$ and $\mathcal{N}$ be smooth manifolds. The* function $F : \mathcal{M} \to \mathcal{N}$ *is* smooth *if for each $p \in \mathcal{M}$ we can find charts (U, φ) and (V, ψ) for $\mathcal{M}$ and $\mathcal{N}$ such that $p \in U$ and $\psi \circ F \circ \varphi^{-1}$ is smooth (as a map between Euclidean spaces).*

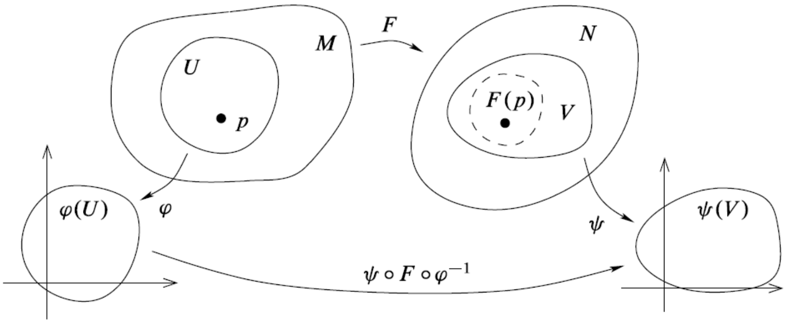


Figure 14: An illustration of the interpretation of smooth functions between manifolds [Lee13]. Smoothness of maps F can be interpreted in terms of classical smoothness between Euclidean spaces.

In particular, we can look at smooth functions $f : \mathcal{M} \to \mathbb{R}$, which we denote by $f \in C^\infty(\mathcal{M})$. If we now consider curves $\gamma : (-\varepsilon, \varepsilon) \to \mathcal{M}$ we have $(f \circ \gamma) : (-\varepsilon, \varepsilon) \subset \mathbb{R} \to \mathbb{R}$ and we can differentiate the real-valued function $f \circ \gamma$ as we are used to from the Euclidean case. This motivates us to define tangent vectors as derivatives of such smooth functions.

Definition 4.8 (tangent vector, tangent space)**.** *Let $\gamma : (-\varepsilon, \varepsilon) \to \mathcal{M}$ be a C^1-curve with $\gamma(0) = p$ and let $C^\infty(p)$ denote the set of smooth functions on a neighborhood of p. The mapping $\dot\gamma(0) : C^\infty(p) \to \mathbb{R}$ defined by*

$$\dot\gamma(0)f := \left.\frac{\mathrm{d}f(\gamma(t))}{\mathrm{d}t}\right|_{t=0}, \quad f \in C^\infty(p) \tag{112}$$

is called the tangent vector *to the curve γ at $t = 0$. The* tangent space $\mathcal{T}_p\mathcal{M}$ *at a point p is the collection of all tangent vectors of curves going through p.*

The tangent space admits a vector space structure of the same dimension as the manifold. In particular, for $u, v \in \mathcal{T}_p\mathcal{M}$ and $a, b \in \mathbb{R}$, we know that $au + bv \in \mathcal{T}_p\mathcal{M}$. The collection of all tangent spaces is called *tangent bundle*, i.e.,

$$\mathcal{T}\mathcal{M} := \bigcup_{p \in \mathcal{M}} \{p\} \times \mathcal{T}_p\mathcal{M} \tag{113}$$

and is a manifold of dimension $2d$, if $\mathcal{M}$ is a d-dimensional manifold [Lee13, Prop. 3.18]. For an interpretation of the tangent space and tangent bundle for the case of $\mathcal{M} = S^2$, see Fig. 15. With that we should note that, we can easily interpret a tangent vector as a vector in a plane tangent to the manifold even though the definition suggests that we look at differential operators.

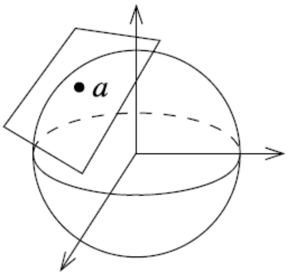


Figure 15: An illustration of the interpretation of the tangent space on a sphere [Lee13]. The union of all these planes and its base points forms the tangent bundle.

Also note that since the tangent space is a vector space it has a dual space, the *cotangent space* and is denoted by $\mathcal{T}_p^*\mathcal{M}$. We call $\xi \in \mathcal{T}_p^*\mathcal{M}$ a *covector* and the duality pairing is denoted by $\langle v, \xi \rangle_p$ for $v \in \mathcal{T}_p\mathcal{M}$ and $\xi \in \mathcal{T}_p^*\mathcal{M}$.

Going back to the more general case of $F : \mathcal{M} \to \mathcal{N}$ we can generalize the derivative of a function between manifolds as an operator that maps tangent vectors.

Definition 4.9 (differential). *Let $F : \mathcal{M} \to \mathcal{N}$ a smooth function, let $p \in \mathcal{M}$ and $v \in \mathcal{T}_p\mathcal{M}$. The mapping $D_pF[v] : C^\infty(\mathcal{N}) \to \mathbb{R}$ given by*

$$(D_pF[v])\, f := v(f \circ F), \quad f \in C^\infty(\mathcal{N}), \tag{114}$$

is a tangent vector at $F(p)$ of $\mathcal{N}$. The mapping

$$D_pF : \mathcal{T}_p\mathcal{M} \to \mathcal{T}_{F(p)}\mathcal{N}, \quad v \mapsto D_pF[v], \tag{115}$$

is called differential *of F.*

This operation is visualized in Fig. 16.

Moreover, the differential of the concatenation $G \circ F$ is given by the chain rule, i.e.,

$$D_p(G \circ F)[v] = D_{F(p)}G[D_pF[v]], \quad v \in \mathcal{T}_p\mathcal{M}. \tag{116}$$

Vector Bundles and Metrics

We have been looking at operations on single vectors, but typically we are interested in the behaviour at entire fields of vectors. However, before getting there we will define the notion of a *vector bundle.*

Definition 4.10 (vector bundle). *A* vector bundle *of rank k over a smooth manifold $\mathcal{M}$ is a manifold E with a surjective smooth projection map $\pi : E \to \mathcal{M}$ such that:*

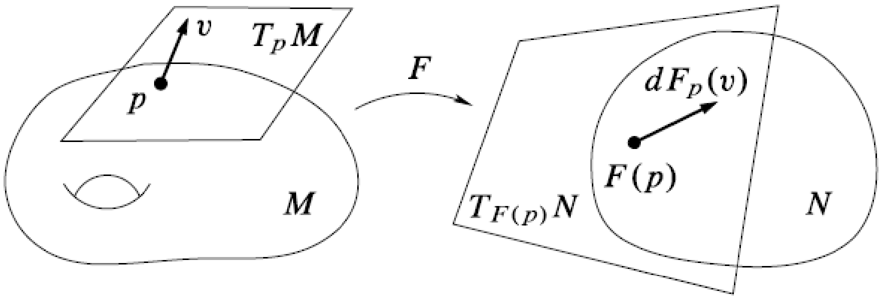


Figure 16: An illustration on the interpretation of the differential as mapping between tangent spaces [Lee13]. The differential maps a tangent vector in $\mathcal{T}_p\mathcal{M}$ to a tangent vector in $\mathcal{T}_{F(p)}\mathcal{N}$.

- *Each $E_p = \pi^{-1}(p)$ is a (real) vector space of dimension k.*
- *For each $p \in \mathcal{M}$, there exist a neighbourhood U and a homeomorphism $\Phi : \pi^{-1}(U) \to U \times \mathbb{R}^k$ (called a* local trivialization *of E over U), satisfying the following conditions:*

  (i) $\pi_U \circ \Phi = \pi$ (where $\pi_U : U \times \mathbb{R}^k \to U$ is the projection).

  (ii) For each $q \in U$, the restriction of Φ to E_q is a vector space isomorphism from E_q to $\{q\} \times \mathbb{R}^k \cong \mathbb{R}^k$

Then, a section can be defined as follows.

Definition 4.11 (section of a vector bundle)**.** *A* section *of a vector bundle E with projection map $\pi : E \to \mathcal{M}$ is a smooth map $\sigma : \mathcal{M} \to E$ such that $\pi \circ \sigma = \mathrm{id}_{\mathcal{M}}$, meaning $\sigma(p) \in E_p$ for each $p \in \mathcal{M}$. The space of all (smooth) sections is denoted $\Gamma(E)$.*

A vector field is an example of a section of the tangent bundle.

Definition 4.12 (vector field)**.** *A (smooth)* vector field *X on a manifold $\mathcal{M}$ is a smooth choice of a vector $X_p \in \mathcal{T}_p\mathcal{M}$ for each point $p \in \mathcal{M}$. That is, X is a (smooth) section of the bundle $\mathcal{TM}$ with projection $\pi : \mathcal{TM} \to \mathcal{M}$, meaning a smooth map $X : \mathcal{M} \to \mathcal{TM}$ such that $\pi \circ X = \mathrm{id}_{\mathcal{M}}$. We write $\mathcal{X} := \mathcal{X}(\mathcal{M}) := \Gamma(\mathcal{TM})$ for the set of all vector fields.*

Remark 4.13. *It is also possible to multiply a smooth function with a vector field. If $f \in C^\infty(\mathcal{M})$ and $X \in \mathcal{X}(\mathcal{M})$, we can define fX as the vector field such that $(fX)_p = f(p)X_p$. This should not be confused with $(Xf)_p = X_p(f)$.*

Similarly, $\Gamma(\mathcal{T}^*\mathcal{M})$ is the section of covector fields. Another example would be $\Gamma(Q(\mathcal{TM}))$, the section over the vector bundle of quadratic forms on $\mathcal{M}$, i.e., we have a quadratic form $q_p : \mathcal{T}_p\mathcal{M} \to \mathbb{R}$ for all $p \in \mathcal{M}$. There is a one-to-one relation between the quadratic forms and the symmetric bilinear forms: indeed for a vector space V, a bilinear form $b : V \times V \to \mathbb{R}$ and $v \in V$, we can define a quadratic form $q(v) := b(v, v)$ and we can recover b from q through

$$2b(v, w) := q(v + w) - q(v) - q(w). \tag{117}$$

If we restrict ourselves to positive definite quadratic forms, we can define a *positive definite section* as a smooth field of positive definite symmetric bilinear forms generated by a section of positive definite quadratic forms in $\Gamma(Q(\mathcal{TM}))$. Such a positive definite section corresponds to a section of inner products.

Definition 4.14 (Riemannian metric)**.** *A positive definite section $g \in \Gamma(Q(\mathcal{TM}))$ is called a* Riemannian metric *on $\mathcal{M}$.*

Given such a metric g, also referred to as the metric tensor, we denote it by $g(\cdot, \cdot)_p : \mathcal{T}_p\mathcal{M} \times \mathcal{T}_p\mathcal{M} \to \mathbb{R}$ (or simply by $(\cdot, \cdot)_p$) and denote the corresponding norm by $\| \cdot \|_p$.

Finally, it turns out that such a positive definite section generated by a section in $\Gamma(Q(TM))$ always exists for a smooth manifold.

Theorem 4.15 (existence of Riemannian metrics, [Lee13, Prop. 13.3])**.** *Every smooth manifold admits a Riemannian metric.*

The first result of the metric tensor now is that we can define a notion of length.

Definition 4.16 (length)**.** *Suppose $\gamma : [a, b] \to \mathcal{M}$ is a piecewise smooth curve. The* length *of γ (with respect to the Riemannian metric g) is*

$$\operatorname{len}(\gamma) := \int_a^b \|\dot{\gamma}(t)\|_{\gamma(t)} \, dt. \tag{118}$$

The distance is a direct consequence.

Definition 4.17 (distance)**.** *The* distance *between two points $p, q \in M$ is the infimal length*

$$d(p, q) := \inf_{\gamma} \operatorname{len}(\gamma) \tag{119}$$

taken over all piecewise smooth curves γ in $\mathcal{M}$ from p to q.

The existence of the metric tensor is the start towards Riemannian geometry.

4.1.2 Riemannian Geometry

As we have seen, having a metric provides us with a notion of distance. Moreover, a metric enables us to talk about even more general notions such as the covariant derivative, which in turn provides us with several manifold mappings and a notion of intrinsic curvature.

A Generalized Directional Derivative

The first step is defining the notion of an affine connection.

Definition 4.18 (affine connection)**.** *An* affine connection *is a mapping* $\nabla : \mathcal{X}(\mathcal{M}) \times \mathcal{X}(\mathcal{M}) \to \mathcal{X}(\mathcal{M})$ *between vector fields denoted by* $(X, Y) \mapsto \nabla_X Y$*, with the following properties:*

(i) linearity in first component over $C^\infty(\mathcal{M})$

$$\nabla_{f_1 X_1 + f_2 X_2} Y = f_1 \nabla_{X_1} Y + f_2 \nabla_{X_2} Y, \quad \forall f_1, f_2 \in C^\infty(\mathcal{M}), X_1, X_2, Y \in \mathcal{X}(\mathcal{M}), \tag{120}$$

(ii) linearity in second component over $\mathbb{R}$

$$\nabla_X (aY_1 + bY_2) = a\nabla_X Y_1 + b\nabla_X Y_2, \quad \forall a, b \in \mathbb{R}, X, Y_1, Y_2 \in \mathcal{X}(\mathcal{M}), \tag{121}$$

(iii) product rule

$$\nabla_X (fY) = f\nabla_X Y + (X(f))Y, \quad \forall f \in C^\infty(\mathcal{M}), X, Y \in \mathcal{X}(\mathcal{M}). \tag{122}$$

This connection can be seen as a generalization of the directional derivative. However, whereas in $\mathbb{R}^d$ the directional derivative is well-defined, this is not the case for manifolds. Nevertheless, there are very useful properties: symmetry and metric compatibility.

Definition 4.19 (metric compatibility)**.** *A connection* ∇ *is called* compatible with the metric g *if the Ricci identity holds*

$$X(g(Y, Z)) = g(\nabla_X Y, Z) + g(Y, \nabla_X Z) \tag{123}$$

for all $X, Y, Z \in \mathcal{X}(\mathcal{M})$.

Definition 4.20. *A connection is called* symmetric *if*

$$\nabla_X Y - \nabla_Y X = [X, Y], \tag{124}$$

where $[X, Y] := XY - YX$ *is the Lie bracket.*

Now we have the following result.

Theorem 4.21 (Levi-Civita connection, [Car92, Thm. 3.6])**.** *Given a Riemannian manifold* $\mathcal{M}$ *there exists a unique affine connection* ∇ *on* $\mathcal{M}$ *satisfying the conditions:*

(i) ∇ *is symmetric.*

(ii) ∇ *is compatible with the Riemannian metric.*

In the following, we will be using the Levi-Civita connection unless stated otherwise. This motivates us for the definition of the covariant derivative.

Definition 4.22 (covariant derivative)**.** *Let and $\mathcal{M}$ be a Riemannian manifold and let $\gamma : [0,1] \to \mathcal{M}$ be a curve. The operator $\frac{D}{\mathrm{d}t} : [0,1] \times \mathcal{X}(\gamma) \to \mathcal{X}(\gamma)$ is the* covariant derivative *along γ, is defined as*

$$\frac{D}{\mathrm{d}t} X(t) := \nabla_{\dot\gamma(t)} \tilde{X}, \tag{125}$$

where $\tilde{X} \in \mathcal{X}(\mathcal{M})$ is any vector field such that $X(t) = \tilde{X}(\gamma(t))$.

Remark 4.23. *Note that for a fixed X, $\frac{DX}{\mathrm{d}t} : [0,1] \to \mathcal{X}(\gamma)$ given by $t \mapsto \frac{DX}{\mathrm{d}t}(t)$ denotes a mapping into a vector field over γ.*

It is often useful to be able to have an explicit expression for the covariant derivative. For that we need a basis along a curve $\gamma : [0,1] \to \mathcal{M}$. That is a collection of linearly independent vector fields $\Theta_i \in \mathcal{X}(\gamma)$. Note that for some coordinate chart (U, φ) and $\gamma \subset U$ we can find this basis at $p = \gamma(t)$ through applying the differential $D_x \varphi^{-1}$ for $x = \varphi(p)$ to a linearly independent vector field on $\mathbb{R}^d$. The latter always exists. Then, for a vector field $X \in \mathcal{X}(\gamma)$ we can find $u^i \in C^\infty([0,1])$ and $x^i \in C^\infty([0,1])$ for $i = 1, ..., d$ and write

$$\dot\gamma = \sum_i u^i \Theta_i \quad \text{and} \quad X = \sum_i x^i \Theta_i. \tag{126}$$

Indeed, we have $u^i = \dot\gamma(\pi^i \circ \varphi)$ and $x^i = X(\pi^i \circ \varphi)$ where $\pi^i : \mathbb{R}^d \to \mathbb{R}$ is the projection onto the ith coordinate in $\mathbb{R}^d$. Then, we can write [Car92, Remark 2.3]

$$\frac{DX}{dt} = \sum_k \left(\frac{dx^k}{dt} + \sum_{i,j} \Gamma^k_{ij} x^i u^j \right) (\Theta_k)_{\gamma(t)}\,, \tag{127}$$

where $\Gamma^k_{ij} \in C^\infty([0,1])$ are the so-called Christoffel symbols representing an additional contribution due to the manifold structure and $(\Theta_k)_{\gamma(t)}$ is the vector in the vector field Θ_k evaluated at $\gamma(t)$.

Remark 4.24. *Instead of $(\Theta_k)_{\gamma(t)}$, often we also write $\Theta_k(t)$ or $(\Theta_k)_p$ if $p = \gamma(t)$ is clear.*

We will not go into detail how to compute the Christoffel symbols, because it is beyond the scope of this work. However, we do want to note that these are determined by the metric tensor and furthermore in the case of $\mathbb{R}^d$ we have $\Gamma^k_{ij} = 0$. So in other words, we see that the directional derivative corresponds to the covariant derivative and we have that the former is well-defined as we are used to.

Remark 4.25. *We can do a similar trick with the differential. For a map $F : \mathcal{M} \to \mathcal{N}$ and a curve $\gamma : [0,1] \to \mathcal{M}$ we can again find a basis $\{\Theta_i\}_i$ with $\Theta_i \in \mathcal{X}(\gamma)$. Then for a vector field $X \in \mathcal{X}(\gamma)$ we can find $x^i \in C^\infty([0,1])$ for $i = 1, ..., d.$ and write*

$$X = \sum_i x^i \Theta_i. \tag{128}$$

Similarly, for a chart (V, ψ) such that $F(U) \subset V \subset \mathcal{N}$ we can find a basis $\{\Psi_j\}_j$ on V with $\Psi_j \in \mathcal{X}(F(\gamma))$ and we can compute the vector field

$$Y = \sum_j \frac{df^j}{dt} \Psi_j, \tag{129}$$

where $f^j = \pi^j \circ \psi \circ F \in C^\infty(U)$, π^j is the projection onto the jth coordinate and $\frac{df^j}{dt} = X(f) = \sum_i x^i \Theta_i(f^j) \in C^\infty([0,1])$. Then we have for $t = t_0$ such that $\gamma(t_0) = p$ that

$$D_pF[X_p] = Y_{F(p)} = \sum_j \left.\frac{df^j}{dt}\right|_{t_0} (\Psi_j)_{F(p)}. \tag{130}$$

Note that we can see $(\Theta_i(f^j))_i^j$ is nothing more than the Jacobian in this context.

The covariant derivative has many useful applications. We start with the notion of parallellism.

Definition 4.26 (parallel vector field)**.** *A vector field $X \in \mathcal{X}(\gamma)$ is called* parallel *to $\gamma : [0,1] \to \mathcal{M}$ if*

$$\frac{D}{\mathrm{d}t} X = 0, \text{ for all } t \in [0,1]. \tag{131}$$

This gives us the notion of a *geodesic.*

Definition 4.27 (geodesic)**.** *A* geodesic *is defined as a curve that is parallel to itself, i.e.,*

$$\frac{D}{\mathrm{d}t}\dot{\gamma} = \nabla_{\dot{\gamma}}\dot{\gamma} = 0. \tag{132}$$

It also turns out that geodesics have constant speed and are locally distance minimizing, but more importantly, they are the foundation of a series of manifold mappings.

From Geodesics to Manifold Mappings
From the description of a geodesic as the solution to a non-linear second-order differential equation, it follows that a geodesic can also be characterized using a starting point $p \in \mathcal{M}$ and a direction $v \in \mathcal{T}_p\mathcal{M}$ using the boundary conditions

$$\gamma_{p;v}(0) = p, \quad \dot{\gamma}_{p;v}(0) = v. \tag{133}$$

Nevertheless, it is not guaranteed that geodesics are well-defined arbitrarily long. This will be the next topic of interest. We denote the subset of $\mathcal{T}_p\mathcal{M}$ for which these geodesics are well defined until $t = 1$ by $\mathcal{G}_p$. A Riemannian manifold $\mathcal{M}$ is said to be *complete* if $\mathcal{G}_p = \mathcal{T}_p\mathcal{M}$ holds for all $p \in \mathcal{M}$. A special type of manifolds with this property is a *Hadamard Manifold*: a simply connected complete Riemannian manifold with non-positive sectional curvature[4].

[4] We will discuss curvature at the end of this section

Definition 4.28 (exponential map)**.** *The* exponential map *is defined as the function* $\exp_p : \mathcal{G}_p \to \mathcal{M}$ *given by*

$$\exp_p(v) := \gamma_{p,v}(1). \tag{134}$$

Note that $\exp_p(tv) = \gamma_{p,v}(t)$ holds for every $t \in [0,1]$. Next, we introduce the set $\mathcal{G}'_p \subset \mathcal{T}_p\mathcal{M}$ as some open ball of radius $0 < r_p \leq \infty$ about the origin such that $\exp_p : \mathcal{G}'_p \to \exp_p\left(\mathcal{G}'_p\right)$ is a diffeomorphism. Then, we can also define its inverse. This radius r_p is often referred to as the *injectivity radius*.

Definition 4.29 (logarithmic map)**.** *The* logarithmic map *is defined as the inverse function of the exponential map, i.e.,* $\log_p : \exp_p\left(\mathcal{G}'_p\right) \to \mathcal{G}'_p \subset \mathcal{T}_p\mathcal{M}$.

If the logarithmic map is well defined, another way of characterizing geodesics would be through its end points. We write $\gamma_{p,q} : [0,1] \to \mathcal{M}$ to indicate a geodesic starting from p going to q. Indeed we can write

$$\gamma_{p,q}(t) = \exp_p\left(t \log_p(q)\right). \tag{135}$$

Furthermore, the exponential and logarithmic maps can be used as a coordinate chart. This choice of chart is often referred to as (geodesic) normal coordinates. In particular, if we on top of that use a set of orthonormal vectors at $p \in \mathcal{M}$ as basis vectors, we have $g_p = I$ and moreover, the Christoffel symbols at p vanish, i.e., $\Gamma^k_{ij}(p) = 0$. This is particularly useful for computing the covariant derivative at one point.

Also note that the Riemannian distance between $p, q \in \mathcal{M}$, for $q \in \mathcal{G}_p$, can be written as

$$d^2(p,q) = (\log_p q, \log_p q)_p = \| \log_p q\|_p^2. \tag{136}$$

Yet another important mapping is the parallel transport map, which allows us to transport information across the manifold.

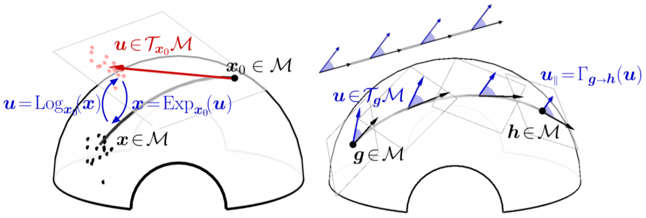

$\boldsymbol{u} \in \mathcal{T}_{\boldsymbol{x}_0}\mathcal{M}$
$\boldsymbol{x}_0 \in \mathcal{M}$
$\boldsymbol{u} = \mathrm{Log}_{\boldsymbol{x}_0}(\boldsymbol{x})$
$\boldsymbol{x} = \mathrm{Exp}_{\boldsymbol{x}_0}(\boldsymbol{u})$
$\boldsymbol{x} \in \mathcal{M}$
$\boldsymbol{u}_{\parallel} = \Gamma_{\boldsymbol{g}\to\boldsymbol{h}}(\boldsymbol{u})$
$\boldsymbol{u} \in \mathcal{T}_{\boldsymbol{g}}\mathcal{M}$
$\boldsymbol{h} \in \mathcal{M}$
$\boldsymbol{g} \in \mathcal{M}$

Figure 17: A visual representation of some manifold structures such as geodesics, the exponential map and the logarithmic map (left) and the parallel transport map (right) [CJ19].

Definition 4.30 (parallel transport)**.** *We define the* parallel transport *of a tangent vector* $v \in T_p\mathcal{M}$ *as a mapping* $P_{p\to q} : \mathcal{T}_p\mathcal{M} \to \mathcal{T}_q\mathcal{M}$ *into the tangent space at* $q \in \mathcal{M}$ *by*

$$P_{p\to q}v := X(1), \tag{137}$$

where $X \in \mathcal{X}(\gamma_{p,q})$ *is the vector field parallel to a minimizing geodesic* $\gamma_{p,q}$ *with* $X(0) = v$.

A summary of the discussed maps are shown in Fig. 17.

It is typically difficult to compute parallel transport and often we need to resort to approximations. A very popular choice is the pole ladder approximation as shown in Fig. 18.

Definition 4.31 (pole ladder)**.** *We define the* pole ladder *approximation to parallel transport by*

$$P^{\mathrm{P}}_{p\to q}(v) := -\log_q\left(\gamma\left(\exp_p(v), \gamma_{p,q}\left(\frac{1}{2}\right); 2\right)\right) \in T_p\mathcal{M}.$$

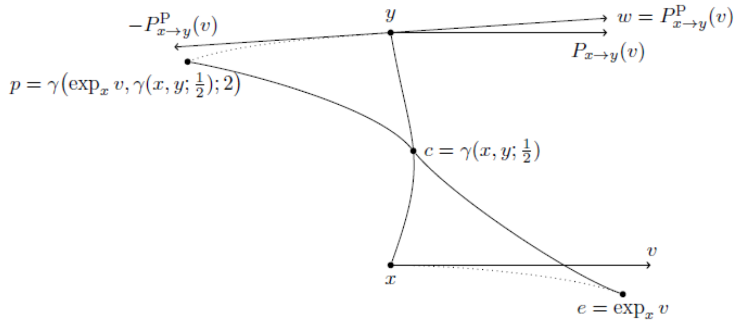

$-P^{\mathrm{P}}_{x\to y}(v)$
y
$w = P^{\mathrm{P}}_{x\to y}(v)$
$P_{x\to y}(v)$
$p = \gamma(\exp_x v, \gamma(x, y; \frac{1}{2}); 2)$
$c = \gamma(x, y; \frac{1}{2})$
v
x
$e = \exp_x v$

Figure 18: Illustration of the construction of the pole ladder, for given $p, q \in \mathcal{M}$ and $v \in T_p\mathcal{M}$ [Per18].

For a special class of manifolds this approximation is even exact. That is for symmetric manifolds. Before we can move on to this result we need the notion of the Riemannian reflection.

Definition 4.32 (reflection)**.** *A mapping $\mathcal{R}_p : \mathcal{M} \to \mathcal{M}$ on a Riemannian manifold $\mathcal{M}$ is called* (geodesic) reflection *at $p \in \mathcal{M}$ if*

$$\mathcal{R}_p(p) = p \quad \text{and} \quad D_p\mathcal{R}_p = -I. \tag{138}$$

For all $p, q \in \mathcal{M}$ with $q \in \mathcal{G}'_p$, we can write the reflection as

$$\mathcal{R}_p(q) = \exp_p\left(-\log_p q\right). \tag{139}$$

Now, we can define a symmetric manifold as follows.

Definition 4.33 (symmetric Riemannian manifold)**.** *A connected Riemannian manifold $\mathcal{M}$ is called (globally)* symmetric *if the geodesic reflection at every point $p \in \mathcal{M}$ is an isometry of $\mathcal{M}$, i.e., for all $x, y \in \mathcal{M}$ we have*

$$d\left(\mathcal{R}_p(x), \mathcal{R}_p(y)\right) = d(x, y). \tag{140}$$

Examples are spheres, Grassmannians, hyperbolic spaces and symmetric positive definite matrices, which are typically among the manifolds of interest for imaging purposes.

We find the following result:

Proposition 4.34 ([Pen18, Prop. 2.1])**.** *Let $\mathcal{M}$ be a connected, complete, and symmetric Riemannian manifold, then the pole ladder is exactly the parallel transport along geodesics, i.e., for all $p, q \in \mathcal{M}$ and $v \in T_p\mathcal{M}$ we have*

$$P_{p\to q}(v) = P^{\mathrm{P}}_{p\to q}(v). \tag{141}$$

The notion of symmetric space is not only useful for approximating parallel transport, but will also be of utmost important in Sect. 4.1.3. It turns out that we can find the differentials and/or covariant derivatives of geodesics, exponential and logarithmic maps on these kind of spaces.

However, before moving on to that topic through the notion of Riemannian curvature, we discuss a final set of manifold mappings that allows us to associate the cotangent space $\mathcal{T}_p^*\mathcal{M}$ at some point $p \in \mathcal{M}$ to the tangent space $\mathcal{T}_p\mathcal{M}$. The Riemannian metric furnishes a linear bijective correspondence between the tangent and cotangent spaces via the Riesz map and its inverse, the so-called musical isomorphisms.

Definition 4.35. *The* musical isomorphisms *are defined as*

$$\flat : \mathcal{T}_p\mathcal{M} \ni X \mapsto X^\flat \in \mathcal{T}_p^*\mathcal{M}, \tag{142}$$

satisfying

$$\langle X^\flat, Y\rangle_p = (X, Y)_p \textit{ for all } Y \in \mathcal{T}_p\mathcal{M} \tag{143}$$

and its inverse,

$$\sharp : \mathcal{T}_p^*\mathcal{M} \ni \xi \mapsto \xi^\sharp \in \mathcal{T}_p\mathcal{M}, \tag{144}$$

satisfying

$$(\xi^\sharp, Y)_p = \langle \xi, Y\rangle_p \textit{ for all } Y \in \mathcal{T}_p\mathcal{M}. \tag{145}$$

These isomorphisms also allow us to define parallel transport for cotangent vectors. That is for $\xi_p \in \mathcal{T}_p^*\mathcal{M}$

$$\mathcal{P}_{p\to q}\xi_p := \left(\mathcal{P}_{p\to q}\xi_p^\sharp\right)^\flat. \tag{146}$$

Curvature

The final topic is curvature. We will discuss the main notions briefly.

Definition 4.36 (Riemannian curvature tensor)**.** *The* Riemannian curvature tensor $R : \mathcal{X}(\mathcal{M}) \times \mathcal{X}(\mathcal{M}) \times \mathcal{X}(\mathcal{M}) \to \mathcal{X}(\mathcal{M})$ *is given by*

$$R(X, Y)Z := \nabla_X\nabla_Y Z - \nabla_Y\nabla_X Z - \nabla_{[X,Y]}Z. \tag{147}$$

This curvature tensor $R(\cdot,\cdot)\cdot$ can be interpreted as translating a vector Z_p along an infinitesimal loop in a plane spanned by the geodesics starting from p with velocities X_p and Y_p. The way Z_p rotates tells us about the nature of the curvature of the manifold. We distinguish positive, negative and zero curvature. Without going into detail here, the classical examples are a sphere S^d, hyperbolic space $\mathbb{H}^d$ and Euclidean space $\mathbb{R}^d$. We can make this rigorous using the following definition.

Definition 4.37 (sectional curvature)**.** *Given a 2 -dimensional subspace* $\Pi \subset T_p\mathcal{M}$*, the* sectional curvature *of* Π *is defined by*

$$K(\Pi) := K(v,u) := \frac{(R(u,v)v,u)_p}{\|u\|_p^2\|v\|_p^2 - (u,v)_p^2} \tag{148}$$

for every two linear independent vectors $v,u \in \Pi$.

If $K(\Pi) = c$ for all sections $\Pi \subset T_p\mathcal{M}$ and $p \in \mathcal{M}$ we say $\mathcal{M}$ has constant (sectional) curvature. A manifold $\mathcal{M}$ has non-positive (-negative) sectional curvature if $K(\Pi) \leq 0(K(\Pi) \geq 0)$ for all sections $\Pi \subset T_p\mathcal{M}$ and $p \in \mathcal{M}$, respectively.

4.1.3 Jacobi Fields

Finally, we are ready to look at an application of the discussed theory in this section. We will look at so-called Jacobi fields. These fields turn out to be particularly interesting for numerical implementation, since for symmetric spaces we can find the derivatives of the geodesic, exponential and logarithmic map.

Let $\Gamma : [0,1] \times (-\varepsilon,\varepsilon) \to \mathcal{M}$ be a function such that for every fixed s, $\Gamma(t,s)$ is a geodesic parametrized in t. Let $\frac{\partial}{\partial s}\Gamma(t,s) =: D_{(t,s)}\Gamma[\frac{\partial}{\partial s}]$ and $\frac{\partial}{\partial t}\Gamma(t,s) =: D_{(t,s)}\Gamma[\frac{\partial}{\partial t}]$[5].

The goal is finding a differential equation satisfied by $\frac{\partial}{\partial s}\Gamma(t,s)|_{s=0}$. Using the symmetry of the Levi-Civita connection we see that

$$\nabla_{\frac{\partial}{\partial t}\Gamma}\frac{\partial}{\partial s}\Gamma - \nabla_{\frac{\partial}{\partial s}\Gamma}\frac{\partial}{\partial t}\Gamma(t,s) = \left[\frac{\partial}{\partial t}\Gamma, \frac{\partial}{\partial s}\Gamma\right] = D_{(t,s)}\Gamma\left[\frac{\partial}{\partial t}, \frac{\partial}{\partial s}\right] = 0 \tag{149}$$

and hence we have $\nabla_{\frac{\partial}{\partial t}\Gamma}\frac{\partial}{\partial s}\Gamma = \nabla_{\frac{\partial}{\partial s}\Gamma}\frac{\partial}{\partial t}\Gamma$. Therefore, we also have

$$\nabla_{\frac{\partial}{\partial t}\Gamma}\nabla_{\frac{\partial}{\partial t}\Gamma}\frac{\partial}{\partial s}\Gamma = \nabla_{\frac{\partial}{\partial t}\Gamma}\nabla_{\frac{\partial}{\partial s}\Gamma}\frac{\partial}{\partial t}\Gamma. \tag{150}$$

Since $\nabla_{\frac{\partial}{\partial t}\Gamma}\frac{\partial}{\partial t}\Gamma = 0$ (because $\frac{\partial}{\partial t}\Gamma$ is a geodesic), we can further write

$$\nabla_{\frac{\partial}{\partial t}\Gamma}\nabla_{\frac{\partial}{\partial t}\Gamma}\frac{\partial}{\partial s}\Gamma = \nabla_{\frac{\partial}{\partial t}\Gamma}\nabla_{\frac{\partial}{\partial s}\Gamma}\frac{\partial}{\partial t}\Gamma = \nabla_{\frac{\partial}{\partial t}\Gamma}\nabla_{\frac{\partial}{\partial s}\Gamma}\frac{\partial}{\partial t}\Gamma - \nabla_{\frac{\partial}{\partial s}\Gamma}\nabla_{\frac{\partial}{\partial t}\Gamma}\frac{\partial}{\partial t}\Gamma. \tag{151}$$

Since we also have that $\left[\frac{\partial}{\partial t}\Gamma, \frac{\partial}{\partial s}\Gamma\right]$, we see that the right-hand-side equals the Riemannian curvature operator. We now define $J(t) = \frac{\partial}{\partial s}\Gamma(t,s)|_{s=0}$ and $\dot{\gamma}(t) = \frac{\partial}{\partial t}\Gamma(t,0)$. Using the antisymmetry in the first two coordinates of the curvature operator we find

[5] In literature on Jacobi fields, this notation is often used.

$$\frac{D^2}{\mathrm{d}t^2} J + R\left(J, \dot{\gamma}\right) \dot{\gamma} = 0. \tag{152}$$

This equation is called the Jacobi field equation and a vector field J satisfying it is called a *Jacobi field*.

For symmetric spaces we can simplify the Jacobi equation using the following result.

Proposition 4.38. *Let $\mathcal{M}$ be a symmetric space. Let $\gamma : [0,1] \to \mathcal{M}$ be geodesic and $\{\Theta_1 = \Theta_1(t), \ldots, \Theta_n = \Theta_n(t)\}$ a parallel transported orthonormal frame along γ. Let $J(t) = \sum_{i=1}^n a_i(t)\Theta_i(t)$ be a Jacobi field of a variation through γ Set $a := (a_1 \ldots, a_n)^{\mathrm{T}}$. Then the following relations hold true:*

(i) The Jacobi equation (152) can be written as

$$a''(t) + Ga(t) = 0, \tag{153}$$

with the constant coefficient matrix $G := \left(\langle R\left(\Theta_i, \dot{\gamma}\right) \dot{\gamma}, \Theta_j \rangle_\gamma\right)_{i,j=1}^n$

(ii) Let $\{\theta_1, \ldots, \theta_n\}$ be chosen as the initial orthonormal basis which diagonalizes the operator $\Theta \mapsto R(\Theta, \dot{\gamma})\dot{\gamma}$ at $t = 0$ with corresponding eigenvalues $\kappa_i, i = 1, \ldots, n$, and let $\{\Theta_1, \ldots, \Theta_n\}$ be the corresponding parallel transported frame along γ. Then the matrix G becomes diagonal and (153) decomposes into the n ordinary linear differential equations

$$a_i''(t) + \kappa_i a_i(t) = 0, \quad i = 1, \ldots, n. \tag{154}$$

(iii) The Jacobi fields

$$J_k(t) := \begin{cases} c^i \sinh(\sqrt{-\kappa_k} t)\Theta_k(t) + d^i \cosh(\sqrt{-\kappa_k} t)\Theta_k(t) & \text{if } \kappa_k < 0 \\ c^i t\Theta_k(t) + d^i\Theta_k(t) & \text{if } \kappa_k = 0 \\ c^i \sin(\sqrt{\kappa_k} t)\Theta_k(t) + d^i \cos(\sqrt{k} t)\Theta_k(t) & \text{if } \kappa_k > 0 \end{cases} \tag{155}$$

$k = 1, \ldots, n$ form a basis of the $2n$-dimensional linear space of Jacobi fields of a variation through γ.

Proof. (i) and (ii) can be found in [BBSW16, Prop. 3.5]. For (iii) we know that the solution space to n second-order differential equations as in (154) is $2n$ dimensional and has of the form as described in (155). □

In recent literature, solutions to the Jacobi field equations have been extensively used. That is because Jacobi fields can be used to calculate the differential and covariant derivatives of several important manifold mappings ([Per18, Lemma 2.3]).

Proposition 4.39. *Let $\mathcal{M}$ be a symmetric Riemannian manifold and let $\{\Theta_k\}_{k=1}^n$ be a parallel transported orthogonal frame along the geodesic $\gamma : [0,1] \to \mathcal{M}$ as defined below. Further, the frame diagonalizes the Riemannian curvature tensor $R(\cdot, \dot{\gamma})\dot{\gamma}$ at $\gamma(0)$ with respective eigenvalues $\kappa_k, k = 1, \ldots, d$.*

(i) For $\gamma(0) = p$, $\gamma(1) = q$, $\tau \in [0,1]$ *and*

$$\alpha(\kappa) := \begin{cases} \frac{\sinh(\sqrt{-\kappa}(1-\tau))}{\sinh(\sqrt{-\kappa})} & \kappa < 0 \\ 1-\tau & \kappa = 0 \\ \frac{\sin(\sqrt{\kappa}(1-\tau))}{\sin(\sqrt{\kappa})} & \kappa > 0 \end{cases} \tag{156}$$

we have $D_p\gamma_{(\cdot),q}(\tau)[\xi] = \sum_{k=1}^{d} \langle \xi, \Theta_k(0)\rangle_p \, \alpha(\kappa_k)\, \Theta_k(\tau)$,

(ii) For $\gamma(0) = p$, $\gamma(1) = q$, $\tau \in [0,1]$ *and*

$$\alpha(\kappa) := \begin{cases} \frac{\sinh(\sqrt{-\kappa}\tau)}{\sinh(\sqrt{-\kappa})} & \kappa < 0 \\ \tau & \kappa = 0 \\ \frac{\sin(\sqrt{\kappa}\tau)}{\sin(\sqrt{\kappa})} & \kappa > 0 \end{cases} \tag{157}$$

we have $D_p\gamma_{q,(\cdot)}(\tau) = \sum_{k=1}^{d} \langle \xi, \Theta_k(0)\rangle_p \, \alpha(\kappa_k)\, \Theta_k(1-\tau)$,

(iii) For $\gamma(0) = p$, $\gamma(1) = \exp_p(u)$ *and*

$$\alpha(\kappa) := \begin{cases} \cosh(\sqrt{-\kappa}) & \kappa < 0 \\ 1 & \kappa = 0 \\ \cos(\sqrt{\kappa}) & \kappa > 0 \end{cases} \tag{158}$$

we have $D_p \exp_{(\cdot)}(u)[\xi] = \sum_{k=1}^{d} \langle \xi, \Theta_k(0)\rangle_p \, \alpha(\kappa_k)\, \Theta_k(1)$,

(iv) For $\gamma(0) = p$, $\gamma(1) = \exp_p(u)$ *and*

$$\alpha(\kappa) = \begin{cases} \frac{\sinh(\sqrt{-\kappa})}{\sqrt{-\kappa}} & \kappa < 0 \\ 1 & \kappa = 0 \\ \frac{\sin(\sqrt{\kappa})}{\sqrt{\kappa}} & \kappa > 0 \end{cases} \tag{159}$$

we have $D_u \exp_p(\cdot)[\xi] = \sum_{k=1}^{d} \langle \xi, \Theta_k(0)\rangle_p \, \alpha(\kappa_k)\, \Theta_k(1)$,

(v) For $\gamma(0) = p$, $\gamma(1) = q$ *and*

$$\alpha(\kappa) := \begin{cases} -\sqrt{-\kappa}\frac{\cosh(\sqrt{-\kappa})}{\sinh(\sqrt{-\kappa})} & \kappa < 0 \\ -1 & \kappa = 0 \\ -\sqrt{\kappa}\frac{\cos(\sqrt{\kappa})}{\sin(\sqrt{\kappa})} & \kappa > 0 \end{cases} \tag{160}$$

we have $\nabla_{\xi_p} \log_{(\cdot)}(q) = \sum_{k=1}^{d} \langle \xi, \Theta_k(0)\rangle_p \, \alpha(\kappa_k)\, \Theta_k(0)$,

(vi) For $\gamma(0) = p$, $\gamma(1) = q$ *and*

$$\alpha(\kappa) := \begin{cases} \frac{\sqrt{-\kappa}}{\sinh(\sqrt{-\kappa})} & \kappa < 0 \\ 1 & \kappa = 0 \\ \frac{\sqrt{\kappa}}{\sin(\sqrt{\kappa})} & \kappa > 0 \end{cases} \tag{161}$$

we have $D_p \log_q(\cdot)[\xi] = \sum_{k=1}^{d} \langle \xi, \Theta_k(0)\rangle_p \, \alpha(\kappa_k)\, \Theta_k(1)$.

Proof. The proofs are given in [Per18, Lemma 2.3]. However, there is a slight misunderstanding with (*v*). In [Per18] the authors claim to have computed $D_p \log_{(\cdot)}(q)$, while they show the result as given here. Whereas the two can be identified, equality is not entirely true. We will discuss this misunderstanding next. □

First, note that $\mathcal{TM}$ is a $2d$-dimensional manifold if $\mathcal{M}$ is d-dimensional and that we should actually write $\mathcal{T}_{(p,\log_p(q))}\mathcal{TM}$. This brings us to the notion of the point and the vector part of the tangent space of a tangent bundle. That is, we can decompose the tangent space into two d-dimensional tangent spaces

$$\mathcal{T}_{(p,\log_p(q))}\mathcal{TM} \cong \mathcal{T}_p\mathcal{M} \times \mathcal{T}_p\mathcal{M} \tag{162}$$

and in particular it is easy to see that

$$D_p \log_{(\cdot)}(q)[v] = (v, \nabla_v \log_{(\cdot)}(q)), \quad v \in \mathcal{T}_p\mathcal{M}. \tag{163}$$

Typically, we look at the vector part (second part) of the tangent space in actual computations. Therefore, $D_p \log_{(\cdot)}(q)[v]$ is often identified with the covariant derivative part (as done in [Per18], but also in other work such as [BLPS18]). We could write $(D_p \log_{(\cdot)}(q)[v])^v$ with v for vector part, but this is often omitted.

Remark 4.40. *Note that the length of γ plays an important role for the eigenvalues in the previous results. If $\gamma'_{p,q}$ is a unit speed geodesic and $\kappa'_k, k = 1, \ldots, d$ are the respective eigenvalues of $R(\cdot, \dot\gamma')\dot\gamma'$ at $\gamma'(0)$, then in the previous result we find for the eigenvalues κ_k in the case of a $[0,1]$ parametrized geodesic $\gamma : [0,1] \to \mathcal{M}$ we have that*

$$\kappa_k = \kappa'_k \ell^2, \quad \textit{for all} \quad k = 1, \ldots, d. \tag{164}$$

Finally, we can also find adjoint operator $(D_pF)^* : T_{F(p)}\mathcal{M} \to T_p\mathcal{M}$ of these special mappings. These are called *adjoint Jacobi fields* and are given by

$$(D_pF)^*[w] = \sum_{k=1}^{d} \langle w, \Xi_k \rangle_{F(p)} \, \alpha(\kappa_k)\Theta_k(0), \quad w \in T_{F(p)}\mathcal{M}, \tag{165}$$

where $\{\Xi_k\}_k$ is the orthonormal frame in $T_{F(p)}\mathcal{M}$ as a result of transporting $\{\Theta_k(0)\}$ along γ as before. A similar expression holds for the covariant derivative in (*v*) of the previous result.

4.2 Specific Manifolds

In this section, we will discuss several manifolds that will be used in the numerical experiments later on in this work. We note that the geodesic $\gamma_{x,y} : [0,1] \to \mathcal{M}$ between two points $x, y \in \mathcal{M}$ is always given by

$$\gamma_{x,y}(t) = \exp_x\left(t \log_x(y)\right) \tag{166}$$

and hence won't be discussed seperately in the following. Unless stated otherwise, these results can be found in [Per18].

Note that this notion of geodesic and the notions following do not rely on charts, but are maps onto the manifold. Approaches relying on these types of maps are often referred to as intrinsic approaches. We will take a closer look into the differences between intrinsic and oppositely extrinsic approaches and justify our choices in the next chapter. First, we look into S^d and $\mathcal{P}(d)$.

4.2.1 The Sphere S^d

The sphere S^d embedded into $\mathbb{R}^{d+1}$ is given by

$$\mathbb{S}^d := \left\{ x \in \mathbb{R}^{d+1} \mid \|x\|_2 = 1 \right\}. \tag{167}$$

It has dimension d. The tangent space at $x \in \mathrm{S}^d$ is given by

$$\mathcal{T}_x \mathrm{S}^d = \left\{ v \in \mathbb{R}^{d+1} \mid \langle x, v \rangle = 0 \right\}, \tag{168}$$

with the Riemannian metric given by the Euclidean inner product. It has constant curvature $K = 1$.

We use the following functions in our computations:

Geodesic Distance
The distance between two points $x, y \in S^d$ is given by

$$d_{S^d}(x, y) = \arccos(\langle x, y \rangle). \tag{169}$$

Exponential Map
The exponential map $\exp_x : \mathcal{T}_x S^d \to S^d$ at a point $x \in S^d$ is given by

$$\exp_x(v) = \cos\left(\|v\|_2\right) x + \frac{\sin\left(\|v\|_2\right)}{\|v\|_2} v. \tag{170}$$

Logarithmic Map
The logarithmic map $\log_x : S^d \backslash \{x\} \to \mathcal{T}_x S^d$ at a point $x \in S^d$ is given by

$$\log_x(y) = d_{S^d}(x, y) \frac{y - \langle x, y \rangle x}{\|y - \langle x, y \rangle x\|}, \quad x \neq -y. \tag{171}$$

Parallel Transport Map
The parallel transport map $P_{x \to y} : \mathcal{T}_x S^d \to \mathcal{T}_y S^d$ along the geodesic from x to y is given by

$$P_{x \to y}(v) = v - \frac{\langle \log_x(y), v \rangle}{d^2_{S^d}(x, y)} \left(\log_x(y) + \log_y(x) \right). \tag{172}$$

Eigen Decomposition of the Curvature Operator

Let $v \in \mathcal{T}_x S^d$. We define $\xi_1 = \frac{v}{\|v\|_2}$ and ξ_i for $i = 2, ..., d$ unit vectors in $T_x \mathrm{S}^d$ orthogonal to ξ_1 w.r.t. the Euclidean inner product.

Then, the eigenvalues of the curvature operator along $\gamma_{x;v}$ are $\kappa_1 = 0$ for the eigenvector ξ_1 and $\kappa_i = 1$ for the other ξ_i [BBSW16].

4.2.2 The $\mathcal{P}(d)$ Manifold of Symmetric Positive Definite Matrices

The manifold $(\mathcal{P}(d), \langle\cdot,\cdot\rangle_{\mathcal{P}(d)})$ of symmetric positive definite $d \times d$ matrices $P(d)$ is given by

$$\mathcal{P}(d) := \{x \in \mathrm{Sym}(d) \mid a^\top x a > 0 \text{ for all } a \in \mathbb{R}^d\}, \tag{173}$$

where $\mathrm{Sym}(d)$ denotes the space of symmetric $d \times d$ matrices. The dimension of $\mathcal{P}(d)$ is $\frac{d(d+1)}{2}$. The tangent space of $P(d)$ at $x \in \mathcal{P}(d)$ is given by

$$\mathcal{T}_x\mathcal{P}(d) := \{x^{\frac{1}{2}} \eta x^{\frac{1}{2}} \mid \eta \in \mathrm{Sym}(d)\}. \tag{174}$$

A Riemannian metric on $\mathcal{P}(d)$ at x is given by the affine invariant metric

$$\langle u, v\rangle_{x,\mathcal{P}(d)} = \mathrm{tr}(x^{-1} u x^{-1} v) \tag{175}$$

We use the following functions in our computations:

Geodesic Distance

The distance between two points $x, y \in \mathcal{P}(d)$ is given by

$$d_{\mathcal{P}(d)}(x, y) = \| \operatorname{Log}\left(x^{-\frac{1}{2}} y x^{-\frac{1}{2}}\right) \|_F, \tag{176}$$

where $\| \cdot \|_F$ is the Frobenius norm and Log is the matrix logarithm.

Exponential Map

The exponential map $\exp_x : \mathcal{T}_x\mathcal{P}(d) \to \mathcal{P}(d)$ at a point $x \in \mathcal{P}(d)$ is given by

$$\exp_x(v) = x^{\frac{1}{2}} \operatorname{Exp}\left(x^{-\frac{1}{2}} v x^{-\frac{1}{2}}\right) x^{\frac{1}{2}}, \tag{177}$$

where Exp is the matrix exponential.

Logarithmic Map

The logarithmic map $\log_x : \mathcal{P}(d) \to \mathcal{T}_x\mathcal{P}(d)$ at a point $x \in \mathcal{P}(d)$ is given by

$$\log_x(y) = x^{\frac{1}{2}} \operatorname{Log}\left(x^{-\frac{1}{2}} v x^{-\frac{1}{2}}\right) x^{\frac{1}{2}}. \tag{178}$$

Parallel Transport Map

The parallel transport $P_{x\to y} : \mathcal{T}_x\mathcal{P}(d) \to \mathcal{T}_y\mathcal{P}(d)$ along the geodesic from x to y is given by

$$P_{x\to y}(v) = x^{\frac{1}{2}} \operatorname{Exp}\left(\frac{1}{2} x^{-\frac{1}{2}} \log_x(y) x^{-\frac{1}{2}}\right) x^{-\frac{1}{2}} v x^{-\frac{1}{2}} \operatorname{Exp}\left(\frac{1}{2} x^{-\frac{1}{2}} \log_x(y) x^{-\frac{1}{2}}\right) x^{\frac{1}{2}}. \tag{179}$$

Eigen Decomposition of the Curvature Operator

Let the matrix v, such that $\tilde{v} = x^{\frac{1}{2}} v x^{\frac{1}{2}} \in T_x\mathcal{P}$, have the eigenvalues $\lambda_1, \dots, \lambda_d$ with a corresponding orthonormal basis of eigenvectors $v_1, \dots, v_d$ in $\mathbb{R}^d$, i.e.,

$$v = \sum_{i=1}^{d} \lambda_i v_i v_i^\top, \tag{180}$$

Then using a more appropriate index system for the frame, namely,

$$\mathcal{I} := \{(i,j) : i = 1, \dots, d; j = i, \dots, d\}, \tag{181}$$

the matrices

$$\xi_{ij} := \begin{cases} \frac{1}{2}\left(v_i v_j^\top + v_j v_i^\top\right), & (i,j) \in \mathcal{I} \quad \text{if } i = j \\ \frac{1}{\sqrt{2}}\left(v_i v_j^\top + v_j v_i^\top\right), & (i,j) \in \mathcal{I} \quad \text{if } i \neq j \end{cases} \tag{182}$$

generate an orthonormal basis of $T_x\mathcal{P}(r)$. That is $x^{\frac{1}{2}} \xi_{ij} x^{\frac{1}{2}}$ form an orthonormal basis of $T_x\mathcal{P}(r)$.

Then, the eigenvalues of the curvature operator along $\gamma_{x;\tilde{v}}(t)$ are

$$\kappa_{ij} = -\frac{1}{4}\left(\lambda_i - \lambda_j\right)^2, \quad (i,j) \in \mathcal{I}, \tag{183}$$

with corresponding eigenvectors $x^{\frac{1}{2}} \xi_{ij} x^{\frac{1}{2}}$ [BBSW16].

Chapter 5: Towards Optimization on Manifolds

In this section we will discuss how to solve

$$\inf_{p\in\mathcal{M}} \{F(p) + G(\Lambda(p))\}, \tag{184}$$

using duality theory. Throughout this chapter, $\Lambda : \mathcal{M} \to \mathcal{N}$ is a non-linear mapping, $F : \mathcal{M} \to \bar{\mathbb{R}}$, $G : \mathcal{N} \to \bar{\mathbb{R}}$ are non-smooth functions and $\mathcal{M}, \mathcal{N}$ are smooth manifolds.

In Sect. 5.1 we will discuss two general approaches to optimization on manifolds in: intrinsic and extrinsic. The remainder of this chapter is entirely mirrored to chapter 2: we will generalize the notions from convex analysis in Sect. 5.2 and after that we move on to a general duality based framework [BHTVN19] to solve (184) in Sect. 5.3.

5.1 Two Approaches: Extrinsic vs. Intrinsic

Around 2014 the mathematical data science community working on topics such as sparse principal component analysis, compressed mode analysis in physics, unsupervised feature selection and sparse blind convolution (see [XLWZ18] for a clear overview) grew interested in the problem in (184). The main issue was that the currently existing algorithms (subgradient descent and PPA) did not optimally respect the structure of the optimization problem or were just not that practical in general. Hence, this community set out to develop new algorithms, which turned out to be independently of the image processing community.

A Series of Extrinsic Attempts

In the absence of specialized algorithms for solving (184), a logical backup was working from the setting of first-order methods. The orthogonality constraints could be taken care of by adding an additional penalty term to the model, which could be easily done in an augmented Lagrangian setting and solved using ADMM-like methods. These algorithms include the Method of Splitting Orthogonality Constraints (SOC) [LO14], the Proximal Alternating Minimization method (PALM) [BST14], the Proximal Alternating Minimization Augmented Lagrangian method (PAMAL) [CJY16], Manifold ADMM (MADMM) [KGB16] and the Extended Proximal Alternating Linearized Minimization method (EPALM) [ZZCL17] as an extension of [BST14].

One might wonder why the algorithms used by the image processing community did not go hand in hand with these developments. The reason is threefold. Fist, we see that by this time (±2014) the image processing group had just realized a workable way of applying Total Variation to manifold data through a generalized ROF model [WDS14]. More importantly, the geometry of the problems were entirely different. Whereas image processing looked at manifolds such as $S^d, \mathcal{P}(d)$ and $SO(3)$, the manifold of interest by the data science community was the Stiefel manifold, because these problems were mostly coping with orthogonality constraints. But the most important reason was that the image processing community passed to intrinsic methods.

Towards Intrinsic Optimization

The additional penalty approach behind the algorithms of the data science group concerned with the Stiefel manifold are so-called *extrinsic* approaches to optimization on manifolds. The main idea of an extrinsic approach is to get some kind of linearity so that results from the linear case can be used for manifolds as well. There are two main lines of doing this. One way is to embed the manifold in $\mathbb{R}^N$ for some N, so that we can work in a linear space around it. Otherwise, since the manifold locally looks like a linear space we can look at different charts of the manifold and solve our problems there as long as we stay in this localized domain. Many first attempts can be traced back to using extrinsic approaches (even back to the 1970s with [Lue72]). However, soon enough extrinsic approaches ran into some issues:

Embedding issues:

- The N in the embedding space $\mathbb{R}^N$ can get very large.
- Numerical errors when trying to project the next iterate onto the manifold.

Localization issues:

- Symmetries of the underlying Riemannian manifold are in general not respected by such algorithms.
- Often we do not have convenient canonical maps for the manifold[6].
- Localizing to a chart leads to distortions in the metric which will in turn lead to slow convergence.
- Talking about global convergence is hard to establish if the entire approach relies on localization.

For these reasons, research into *intrinsic* algorithms grew popular and soon became the standard. As mentioned, in the end it was also the approach picked up by the image analysis community. With an intrinsic approach, one does not rely on some kind of embedding or mapping into a linear space, but one rather uses mappings from and to the manifold, incorporating the intrinsic geometric structure of the manifold. One should note that the downside is that the mathematics becomes increasingly difficult. In the following we will use continue in line with the intrinsic approach.

An Outlook to Non-smooth Optimization

In a recent contribution [CMMCSZ20] the authors used an intrinsic method for ℓ^1 regularization optimization on the Stiefel manifold. This attempt can be seen as an important step towards the merger of the tradition of extrinsic approaches to optimization over the Stiefel manifold with intrinsic non-smooth optimization on manifolds.

[6] e.g., in the case of the Grassmanian.

However, this goes wildly beyond the scope of this work. In the following, we will continue focusing on an intrinsic approach to non-smooth analysis, and in particular into generalizing the Fenchel duality theory.

5.2 Non-smooth Analysis on Manifolds

In the linear case we used the dual variables in $\mathbb{R}^d$ for some d. In general Banach spaces we can move on to dual space for Fenchel duality theory; since we have $(\mathbb{R}^d)^* \cong \mathbb{R}^d$ this is a generalization. For manifolds we will also use the more general approach and look for a dual space for duality theory.

The goal is to find a manifold version of (20) of the form

$$\text{“}\inf_{p\in\mathcal{M}} \sup_{\xi\in Z^*} \{F(p) + \langle \Lambda(p), \xi\rangle - G^*(\xi)\}\text{”}$$

where $\xi \in Z^*$ a dual vector in a dual space Z^*. However, we run into a few issues:

- A general manifold does not necessarily have a dual space, in other words: what should Z^* be?
- If we can define a dual space, how should $\langle \Lambda(p), \xi\rangle$ be read? The point $\Lambda(p) \in \mathcal{N}$ does not allow for a duality pairing due non-linearity of $\mathcal{N}$ and $\Lambda : \mathcal{M} \to \mathcal{N}$ being a non-linear mapping.
- How do we go from a dual space and a duality pairing to a generalized conjugate function $G^*(\xi)$?
- Finally, with these definitions, does the constructed optimization problem have a solution?

Whereas these questions are rather advanced, the answers steadily come as we generalize the notions from convex analysis. The key will be finding some sort of linearity that allows for a dual space. The following follows from [BHTVN19].

Convex Analysis on Manifolds
Initially, in order to talk about convexity we need to pass to strongly convex subsets of Riemannian manifolds [BHTVN19, Def. 2.9].

Definition 5.1 (strongly convex set)**.** *A subset $\mathcal{C} \subset \mathcal{M}$ of a Riemannian manifold $\mathcal{M}$ is said to be* strongly convex *if, for all $p, q \in \mathcal{C}$, a minimal geodesic $\gamma_{p,q}$ between p and q exists, is unique and lies completely in $\mathcal{C}$.*

Next, we can generalize the well-known notions of properness, convexity and lower semi-continuity [BHTVN19, Def. 2.11.i-iv].

Definition 5.2 (proper)**.** *A function $F : \mathcal{M} \to \bar{\mathbb{R}}$ is* proper *if* $\operatorname{dom} F := \{x \in \mathcal{M} | F(x) < \infty\} \neq \emptyset$ *and $F(x) > -\infty$ holds for all $x \in \mathcal{M}$.*

Definition 5.3 (convex)**.** *Suppose that $\mathcal{C} \subset \mathcal{M}$ is strongly convex. A proper function $F : \mathcal{M} \to \bar{\mathbb{R}}$ is called (geodesically)* convex *on $\mathcal{C} \subset \mathcal{M}$ if for all $p, q \in \mathcal{C}$ the composition $F \circ \gamma_{p,q}(t)$ is a convex function on $[0, 1]$ in the classical sense.*

Definition 5.4 (epigraph)**.** *Suppose that $\mathcal{A} \subset \mathcal{M}$. The* epigraph *of a function $F : \mathcal{A} \to \bar{\mathbb{R}}$ is defined as*

$$\operatorname{epi} F := \{(x, \alpha) \in \mathcal{A} \times \mathbb{R} | F(x) \leq \alpha\}. \tag{185}$$

Definition 5.5 (lower semi-continuous)**.** *Suppose that $\mathcal{A} \subset \mathcal{M}$. A proper function $F : \mathcal{A} \to \bar{\mathbb{R}}$ is called* lower semi-continuous *(lsc) if* $\operatorname{epi} F$ *is closed.*

For generalizing the subdifferential, remember that in the $\mathbb{R}^n$ case we used the following

$$\partial f(x) := \{v \in \mathbb{R}^n | f(y) \geq f(x) + \langle v, y - x\rangle \ \forall y \in U\},$$

where $f : U \to \bar{\mathbb{R}}$ and $x \in U \subset \mathbb{R}^n$. This notion relies on the subtraction of two points in $\mathbb{R}^n$, which is not possible on manifolds. Now, the key idea is that the logarithmic map generalizes subtraction and subsequently gives us a tangent vector. The latter lives in a vector space and thus can be paired with a cotangent vector. This motivates the following definition [BHTVN19, Def. 2.12].

Definition 5.6 (subdifferential)**.** *Suppose that $\mathcal{C} \subset \mathcal{M}$ is strongly convex. The* subdifferential *$\partial_{\mathcal{M}} F$ on $\mathcal{C}$ at a point $p \in \mathcal{C}$ of a proper, convex function $F : \mathcal{C} \to \bar{\mathbb{R}}$ is given by*

$$\partial_{\mathcal{M}} F(p) := \left\{\xi \in \mathcal{T}_p^* \mathcal{M} | F(q) \geq F(p) + \langle \xi, \log_p q\rangle_p \text{ for all } q \in \mathcal{C}\right\}. \tag{186}$$

The definition of the proximal map closely mirrors the linear case:

Definition 5.7 (proximal mapping)**.** *Let $\mathcal{M}$ be a Riemannian manifold, $F : \mathcal{M} \to \bar{\mathbb{R}}$ be proper, and $\lambda > 0$. The* proximal map *of F is defined as*

$$\operatorname{prox}_{\lambda F}(p) := \arg\min_{q \in \mathcal{M}} \left\{\frac{1}{2\lambda} d_{\mathcal{M}}^2(p, q) + F(q)\right\}. \tag{187}$$

Fenchel conjugate functions

Next, the idea of using tangent and cotangent spaces as dual space is used as well to generalize the Fenchel dual functions. For that we need the exponential and the logarithmic map to be well-defined [BHTVN19, Def 2.10].

Definition 5.8. *Let $\mathcal{C} \subset \mathcal{M}$ and $p \in \mathcal{C}$. Define the tangent subset $\mathcal{L}_{\mathcal{C},p} \subset \mathcal{T}_p\mathcal{M}$ as*

$$\mathcal{L}_{\mathcal{C},p} := \left\{X \in \mathcal{T}_p\mathcal{M} \mid \exp_p X \in \mathcal{C} \text{ and } \|X\|_p = d_{\mathcal{M}}(\exp_p X, p)\right\}, \tag{188}$$

a localized variant of the pre-image of the exponential map.

Then we can define the generalization of the Fenchel conjugate [BHTVN19, Def. 3.1]. Note that a manifold does not have a clear origin: we should choose a base point m.

Definition 5.9 (m-Fenchel conjugate)**.** *Suppose that $F : \mathcal{C} \to \bar{\mathbb{R}}$ and $m \in \mathcal{C}$. The* m-Fenchel conjugate *of F is defined as the function $F_m^* : \mathcal{T}_p^*\mathcal{M} \to \bar{\mathbb{R}}$ such that*

$$F_m^* \left(\xi_m\right) := \sup_{X \in \mathcal{L}_{C,m}} \left\{\langle \xi_m, X \rangle - F\left(\exp_m X\right)\right\}, \quad \xi_m \in \mathcal{T}_m^*\mathcal{M}. \tag{189}$$

For the Fenchel biconjugate we can then define the following [BHTVN19, Def. 3.5].

Definition 5.10 ((mm')-Fenchel biconjugate)**.** *Suppose that $F : \mathcal{C} \to \bar{\mathbb{R}}$ and $m, m' \in \mathcal{C}$. Then the* (mm')-Fenchel biconjugate function *$F_{mm'} : \mathcal{C} \to \bar{\mathbb{R}}$ is defined as*

$$F_{mm'}^{**}(p) := \sup_{\xi_{m'} \in \mathcal{T}_{m'}^*\mathcal{M}} \left\{\langle \xi_{m'}, \log_{m'} p \rangle - F_m^* \left(\mathcal{P}_{m' \to m} \xi_{m'}\right)\right\}, \quad p \in \mathcal{C}. \tag{190}$$

Now, if we choose $m' = m$ and we get

$$F_{mm}^{**}(p) = \sup_{\xi_m \in \mathcal{T}_m^*\mathcal{M}} \left\{\langle \xi_m, \log_m p \rangle - F_m^* \left(\xi_m\right)\right\}, \quad p \in \mathcal{C}, \tag{191}$$

we have the generalization to the Fenchel-Moroe-Rockafellar theorem:

Theorem 5.11 ([BHTVN19, Thm. 3.11])**.** *Let $F : \mathcal{C} \to \bar{\mathbb{R}}$ be a proper, convex function and $m \in \mathcal{C}$. Then $F(p) = F_{mm}^{**}(p)$ for all $p \in \mathcal{C}$ if and only if F is lsc on $\mathcal{C}$.*

In order to get back to the issues stated at the beginning of this section, a proper saddle-point formulation can be constructed using these definitions. Since $G : \mathcal{N} \to \bar{\mathbb{R}}$, the dual space becomes $Z^* = \mathcal{T}_n^*\mathcal{N}$ with some base point $n \in \mathcal{N}$. The duality pairing must be $\langle \log_n \Lambda(p), \xi_n \rangle$, rather than $\langle \Lambda(p), \xi \rangle$ and $G_n^*(\xi_n)$ is the proper notion of the Fenchel conjugate of G. Hence, we are left with

$$\inf_{p \in \mathcal{M}} \sup_{\xi \in T_n^*\mathcal{N}} \{F(p) + \langle \log_n \Lambda(p), \xi_n \rangle - G_n^*(\xi_n)\}. \tag{192}$$

The last issue brought up at the start of this section was concerned with existence of a solution. This remains an open question. Besides existence, strong duality, which would allow to swap inf and sup, remains open as well. Developing the theory incorporating the strong duality would be a valuable asset towards providing criteria for existence.

5.3 The Riemannian Chambolle-Pock Algorithms

Next, we want to generalize the PDHG algorithm. The resulting algorithms proposed in [BHTVN19] are the so-called *exact* and *linearized Riemannian Chambolle Pock algorithms* (eRCPA and lRCPA). First the exact version will be considered.

eRCPA

Assuming that an optimal saddle-point solution (p, ξ_n) exists, the optimality condition for the sup can be found by differentiating (192) with respect to ξ_n [BHTVN19]:

$$\log_n \Lambda(p) \in \partial G_n^* \left(\xi_n\right). \tag{193}$$

For the optimality condition of the inf term, an expression of the form "$F(p)+\langle p, [\log_n \Lambda]^* \xi_n\rangle$" must be differentiated. This expression does not make much sense for two reasons. Clearly, p is not a tangent vector and hence it is by no means clear what the duality pairing would mean for this case, but more importantly: $\log_n \Lambda$ is a non-linear operator and the notion of an adjoint operator is canonically given for linear operators.

In [BHTVN19] is chosen for a linearization of the non-linear mapping Λ in order to overcome the issue of the adjoint. The following approximation is used

$$\Lambda(p) \approx \exp_{\Lambda(m)} D_m\Lambda\left[\log_m p\right]. \tag{194}$$

Subsequently, if $n := \Lambda(m)$ is the base point for (192)

$$\langle \log_n \Lambda(p), \xi_n\rangle \approx \langle \log_{\Lambda(m)} \exp_{\Lambda(m)} D_m\Lambda\left[\log_m p\right], \xi_{\Lambda(m)}\rangle = \langle D_m\Lambda\left[\log_m p\right], \xi_{\Lambda(m)}\rangle \tag{195}$$

and now

$$\langle \log_m p, (D_m\Lambda)^*[\xi_{\Lambda(m)}]\rangle \tag{196}$$

does make sense, where $D_m\Lambda^* : \mathcal{T}^*_{\Lambda(m)}\mathcal{N} \to \mathcal{T}^*_m\mathcal{M}$ is the adjoint operator of $D_m\Lambda$. At this point [BHTVN19] initially take

$$\mathcal{P}_{m\to p}\left(-(D_m\Lambda)^*[\xi_{\Lambda(m)}]\right) \in \partial_{\mathcal{M}} F(p) \tag{197}$$

as second optimality condition. Here parallel transport is used in order to get to the correct tangent space. Subsequently, the restriction of $n := \Lambda(m)$ is dropped and the following optimality system is proposed [BHTVN19]:

$$\mathcal{P}_{m\to p}\left(-(D_m\Lambda)^*\left[\mathcal{P}_{n\to\Lambda(m)}\xi_n\right]\right) \in \partial_{\mathcal{M}} F(p), \tag{198}$$

$$\log_n \Lambda(p) \in \partial G^*_n\left(\xi_n\right), \tag{199}$$

which the authors of [BHTVN19] rewrite into

$$p = \operatorname{prox}_{\sigma F}\left(\exp_p\left(\mathcal{P}_{m\to p}\left(-\sigma(D_m\Lambda)^*\left[\mathcal{P}_{n\to\Lambda(m)}\xi_n\right]\right)^\sharp\right)\right), \tag{200}$$

$$\xi_n = \operatorname{prox}_{\tau G^*_n}\left(\xi_n + \tau\left(\log_n \Lambda\left(p\right)\right)^\flat\right). \tag{201}$$

The exact Riemannian Chambolle Pock (eRCPA) scheme (Alg. 3) is used to solve it.

The major drawback of the eRCPA algorithm is, that any theoretical guarantee is missing. To get this guarantee, we need more linearity. In the exact scheme we only used the linearization for one of the optimality conditions, but we can also do it for both cases.

lRCPA

By adopting the approximation in (194) into the saddle-point problem we find

$$\inf_{p\in\mathcal{M}} \sup_{\xi_n\in\mathcal{T}^*_n\mathcal{N}} \{F(p) + \langle D_m\Lambda\left[\log_m p\right], \xi_n\rangle - G^*_n\left(\xi_n\right)\}, \tag{202}$$

Algorithm 3 Exact Riemannian Chambolle-Pock [BHTVN19] (eRCPA)

```
  Initialization: $m \in \mathcal{C}, n \in \mathcal{D}, p^{(0)} \in \mathcal{C}$, $\xi_n^{(0)} \in \mathcal{T}_n^*\mathcal{N}$, and parameters $\sigma_0, \tau_0, \theta_0, \gamma$
  $k := 0, \quad \bar{p}^{(0)} := p^{(0)}$
  while not converged do
      $\xi_n^{(k+1)} := \operatorname{prox}_{\tau_k G_n^*}\left(\xi_n^{(k)} + \tau_k \left(\log_n \Lambda\left(\bar{p}^{(k)}\right)\right)^\flat\right)$
      $p^{(k+1)} := \operatorname{prox}_{\sigma_k F}\left(\exp_{p^{(k)}}\left(\mathcal{P}_{m\to p^{(k)}}\left(-\sigma_k (D_m\Lambda)^*\left[\mathcal{P}_{n\to\Lambda(m)}\xi_n^{(k+1)}\right]\right)^\sharp\right)\right)$
      $\theta_k := (1 + 2\gamma\sigma_k)^{-\frac{1}{2}}, \quad \sigma_{k+1} := \sigma_k\theta_k, \quad \tau_{k+1} := \tau_k/\theta_k$
      $\bar{p}^{(k+1)} := \exp_{p^{(k+1)}}\left(-\theta_k \log_{p^{(k+1)}} p^{(k)}\right)$
      $k := k + 1$
  end while
```

corresponding to the *linearized primal problem*

$$\inf_{p\in\mathcal{M}} \left\{F(p) + G\left(\exp_{\Lambda(m)} D_m\Lambda\left[\log_m p\right]\right)\right\}. \tag{203}$$

The problem in (202) will be referred to as the *linearized saddle-point problem* from now on (contrary to the *exact saddle-point problem* as discussed before). Now, assuming that a solution exists, the following optimality conditions are proposed [BHTVN19]:

$$\mathcal{P}_{m\to p}\left(-(D_m\Lambda)^*\left[\mathcal{P}_{n\to\Lambda(m)}\xi_n\right]\right) \in \partial_{\mathcal{M}} F(p), \tag{204}$$

$$D_m\Lambda\left[\log_m p\right] \in \partial G_n^*\left(\xi_n\right), \tag{205}$$

which the authors of [BHTVN19] rewrite into

$$p = \operatorname{prox}_{\sigma F}\left(\exp_p\left(\mathcal{P}_{m\to p}\left(-\sigma\left(D_m\Lambda\right)^*\left[\mathcal{P}_{n\to\Lambda(m)}\xi_n\right]\right)^\sharp\right)\right), \tag{206}$$

$$\xi_n = \operatorname{prox}_{\tau G_n^*}\left(\xi_n + \tau\left(\mathcal{P}_{\Lambda(m)\to n} D_m\Lambda\left[\log_m p\right]\right)^\flat\right). \tag{207}$$

The following weak duality result holds:

Theorem 5.12 ([BHTVN19, Thm. 4.2])**.** *Let $n := \Lambda(m)$. The dual problem of (203) is given by*

$$\sup_{\xi_n\in\mathcal{T}_n^*\mathcal{N}} F_m^*\left(-(D_m\Lambda)^*\left[\xi_n\right]\right) - G_n^*\left(\xi_n\right) \tag{208}$$

and weak duality holds, i.e.

$$\inf_{p\in\mathcal{M}} F(p) + G\left(\exp_{\Lambda(m)} D_m\Lambda\left[\log_m p\right]\right) \geq \sup_{\xi_n\in\mathcal{T}_n^*,\mathcal{N}} -F_m^*\left(-(D_m\Lambda)^*\left[\xi_n\right]\right) - G_n^*\left(\xi_n\right). \tag{209}$$

The *linearized* Riemannian Chambolle Pock Algorithm (lRCPA), derived from these conditions, is shown in Alg. 4.

Algorithm 4 Linearized Riemannian Chambolle-Pock [BHTVN19] (lRCPA)

```
Initialization: $m^{(k)} \in \mathcal{C}, n^{(k)} \in \mathcal{D}, p^{(0)} \in \mathcal{C}$, $\xi^{(0)}_{n^(0)} \in \mathcal{T}^*_{n^(0)}\mathcal{N}$, and parameters $\sigma_0, \tau_0, \theta_0, \gamma$

$k := 0, \quad \bar{\xi}^{(0)}_{n^{(0)}} := \xi^{(0)}_{n^(0)}$
**while** not converged **do**
    $p^{(k+1)} := \operatorname{prox}_{\sigma_n F}\left(\exp_{p^{(k)}}\left(\mathcal{P}_{m^{(k)}\to p^{(k)}}\left(-\sigma_k\left(D_{m^K}\Lambda\right)^*\left[\mathcal{P}_{n^{(k)}\to\Lambda\left(m^{(k)}\right)}\xi^{(k)}_{n^{(k)}}\right]\right)^\sharp\right)\right)$
    $\xi^{(k+1)}_{n^{(k)}} := \operatorname{prox}_{\tau_k G^*_{n(k)}}\left(\xi^{(k)}_{n^{(k)}} + \tau_k\left(\mathcal{P}_{\Lambda\left(m^{(k)}\right)\to n^{(k)}} D_{m^k}\Lambda\left[\log_{m^{(k)}} p^{(k+1)}\right]\right)^\flat\right)$
    $\theta_k := (1+2\gamma\sigma_k)^{-\frac{1}{2}}, \quad \sigma_{k+1} := \sigma_k\theta_k, \quad \tau_{k+1} := \tau_k/\theta_k$
    $\bar{\xi}^{(k+1)}_{n(k+1)} := \mathcal{P}_{n(k)\to n(k+1)}\left(\xi^{(k+1)}_{n(k)} + \theta_k\left(\xi^{(k+1)}_{n(k)} - \xi^{(k)}_{n(k)}\right)\right)$
    $\xi^{(k+1)}_{n(k+1)} := \mathcal{P}_{n(k)\to n(k+1)}\xi^{(k+1)}_{n^{(k)}}$
    $k := k+1$
**end while**
```

Next, let

$$L := \|D_m\Lambda\|_{\Lambda(m)} \tag{210}$$

be the operator norm of $D_m\Lambda : \mathcal{T}_m\mathcal{M} \to \mathcal{T}_{\Lambda(m)}\mathcal{N}$, then we get the following convergence result for Hadamard manifolds.

Theorem 5.13 (Convergence lRCPA, [BHTVN19, Thm. 4.3])**.** *Let $\mathcal{M}$ and $\mathcal{N}$ be two Hadamard manifolds and $F : \mathcal{M} \to \bar{\mathbb{R}}$, $G : \mathcal{N} \to \bar{\mathbb{R}}$ be proper, convex, lsc, and $\Lambda : \mathcal{M} \to \mathcal{N}$. Fix $m \in M$ and $n := \Lambda(m) \in \mathcal{N}$. Suppose that the linearized saddle-point problem in (202) has a solution $\left(\hat{p}, \hat{\xi}^*_n\right)$. Choose σ, τ such that $\sigma\tau L^2 < 1$, with L defined in (210), and let the iterates $\left(\xi^{(k)}_n, p^{(k)}, \bar{\xi}^{(k)}_n\right)$ be given by Alg. 4. Suppose that there exists $K \in \mathbb{N}$ such that for all $k \geq K$, the following holds:*

$$C(k) := \frac{1}{\sigma} d^2_{\mathcal{M}}\left(p^{(k)}, \tilde{p}^{(k)}\right) + \left\langle \bar{\xi}^{(k)}_n, D_m\Lambda\left[\zeta_k\right]\right\rangle_n \geq 0, \tag{211}$$

where $\tilde{\mathrm{p}}^{(k)}$ is defined by

$$\tilde{p}^{(k)} := \exp_{p^{(k)}}\left(\mathcal{P}_{m\to p^{(k)}} - \left(\sigma(D_m\Lambda)^*\left[\mathcal{P}_{n\to\Lambda(m)}\xi^{(k+1)}_n\right]\right)^\sharp\right), \tag{212}$$

$$\zeta_k := \mathcal{P}_{p(k)\to m}\left(\log_{p(k)} p^{(k+1)} - \mathcal{P}_{\bar{p}(k)\to p(t)} \log_{\bar{p}(k)} \hat{p}\right) - \log_m p^{(k+1)} + \log_m \hat{p}, \tag{213}$$

with $\bar{\xi} n^{(k)} = 2\xi^{(k)}_n - \xi^{(k-1)}_n$. Then the following statements are true.

(i) The sequence $\left(p^{(k)}, \xi^{(k)}_n\right)$ remains bounded, i.e.,

$$\frac{1}{2\tau}\left\|\widehat{\xi}_n - \xi^{(k)}_n\right\|^2_n + \frac{1}{2\sigma} d^2_{\mathcal{M}}\left(p^{(k)}, \widehat{p}\right) \leq \frac{1}{2\tau}\left\|\hat{\xi}_n - \xi^{(0)}_n\right\|^2_n + \frac{1}{2\sigma} d^2_{\mathcal{M}}\left(p^{(0)}, \widehat{p}\right). \tag{214}$$

(ii) There exists a saddle-point (p^*, ξ_n^*) *such that* $p^{(k)} \to p^*$ *and* $\xi_n^{(k)} \to \xi_n^*$.

Remark 5.14. *For the authors of [BHTVN19] it was not clear how good the linearization of* Λ *in (194) is. It is also unclear whether a single approximation as in eRCPA makes sense. While lRCPA has a theoretical backbone, the convergence of eRCPA is currently unknown.*

Chapter 6: The Riemannian Semismooth Newton Method

In this chapter we will investigate a *Riemannian Semismooth Newton method* (RSSN) with the goal of realizing a duality-based higher-order method for non-smooth optimization on manifolds.

The structure is similar to that of chapter 3. Sect. 6.1 will provide some additional motivation and context for the use and the development of the method. In Sect. 6.2 the theoretical background around the Riemannian Semismooth Newton method will be discussed. In Sect. 6.3 we will expand this theory by proving a convergence result for the case that the Newton system is solved inexactly. In Sect. 6.4 we will elaborate on using the Riemannian Semismooth Newton method as a higher order primal-dual method. In Sect. 6.5 the method will be applied for solving the ROF model. Finally, in Sect. 6.6 we investigate the numerical performance.

6.1 Introduction

The theory and ideas of the current manifold-valued image processing community originate from the field of smooth optimization on Riemannian manifolds. Starting around 1994, contributions to intrinsic methods for smooth optimization formed the basis for the current paradigm for manifold-valued image processing. Pioneering work was done in [Smi94, Udr94], whose authors formulated several algorithms such as gradient descent, Newton's method and conjugate gradient to Riemannian manifolds. From then on, the community started working on generalizing algorithms to Riemannian manifolds [ABG07, BFFY18, CDGS17], specializing algorithms for better results [EAS98, AEK08, HWY13] or applying the obtained literature to real-world problems [ADM^{+}02, ATV13]. In the end, much of the obtained literature and and many of the foundational ideas were bundled in [AMS09], who provided an extensive overview of first-and second-order methods for optimization on matrix manifolds.

The Rise of Non-smooth Optimization

For non-smooth problem, the pioneering works were [FO98], in which the subgradient was extended to Riemannian manifolds and was shown to convergence on Hadamard manifolds, and [FO02], whose authors extended the proximal map and proved convergence on Hadamard Manifolds. Further development in non-smooth optimization gained a boost in the late 2000s when [AF05] introduced the proximal subdifferential on Riemannian manifolds, [HP11] proposed another framework and generalized the notion of Clarke subdifferential for Riemannian manifolds, and [KA10] proposed a framework for duality on $CAT(0)$ metric spaces.The latter became one of the sources of inspiration for the Fenchel duality theory in [BHTVN19] as discussed in the previous chapter.

A Different Focus

The image processing community was heading into a different direction than the non-smooth optimization community, which focused on theoretical performance of algorithms, such as global convergence, often studying problems of the form

$$\inf_{p\in\mathcal{M}} F(p), \tag{215}$$

where $F : \mathcal{M} \to \mathbb{R}$ is a Lipschitz function. The specific case that F was the sum of two functions, giving more structure and therefore more potential for developing fast algorithms, was initially irrelevant. With the introduction of CPPA as the first splitting algorithm on (Hadamard) manifolds, the image processing community split off from the theoretical developments of the non-smooth optimization community.

However, by the start of the 2010s, theoretical foundations of non-smooth optimization on Hadamard spaces became more and more complete. For the non-Hadamard case, there were many open problems. Therefore, we see from 2010 onwards the following developments: [BNO11] present a non-smooth version of the Kurdyka-Lojasiewicz (KL) inequality and with that shows the convergence of PPA on general Riemannian manifolds, [Hos15] showed that subgradient descent applied to locally Lipschitz functions on Riemannian manifolds satisfying the KL inequality converges to a singular critical point, and [BCNO16a] proposed a new approach to the convergence of PPA that extends previous results to a broader class of functions.

Merging Fields

Around 2015, the first numerical implementations of these and new algorithms got attention. Fist, [GH16a] introduced a non-smooth trust region method for Riemannian manifolds and showed global convergence, [GH16b] used an approximate subdifferential and proposed a descent method with global convergence, [HU17] proposed a gradient sampling algorithm and showed its global convergence and finally [HHY18] proposed a line search al algorithm and generalized the Wolfe conditions for Riemannian manifolds.

Recently in 2018 a higher-order method was introduced: the Riemannian Semismooth Newton Method (RSSN) [OF18]. This is a promising approach, because of its very general applicability to manifolds with both positive and negative sectional curvature. Therefore, we will consider the method in the remaining sections of this thesis.

6.2 Newton's Method for Finding Zeros of Non-smooth Vector Fields

There is no straightforward generalization of the Newton method to manifolds. Remember from the Euclidean case that the goal of the Newton method will be to find a zero of some mapping by linearizing it in each step. On manifolds, there is no such thing as a zero. To resolve this, an option would be to derive some non-linear optimality condition $X : \mathcal{M} \to \mathbb{R}^n$ and to use the differential and the exponential map to take Newton steps:

$$p^{k+1} = \exp_{p^k}(-[D_{p^k}X(\cdot)]^{-1}X(p^k)). \tag{216}$$

However, we have also seen that we can define vector fields over a manifold as sections of the tangent bundle $\mathcal{TM}$. While the tangent bundle is not a linear space in general, it provides everything we need: there is such a thing as a zero tangent vector we can look for and the covariant derivative can be our tool to find a Newton operator. In other words we can also look at mappings $X : \mathcal{M} \to \mathcal{TM}$ and define the Newton iteration through

$$p^{k+1} = \exp_{p^k}(-[\nabla X(\cdot)]^{-1}_{p^k} X(p^k)). \tag{217}$$

Whereas the first option, a map into a vector space, could be useful as well, finding a zero of a vector field will be more useful for our purposes. In section Sect. 6.4 we will elaborate on our choice further, but for now we will focus on a generalized covariant derivative approach for finding zeros of semismooth vector fields. Throughout this section we will use the notions and results developed in [OF18].

6.2.1 Generalized Covariant Derivatives and Semismooth Vector Fields

As with the linear case, we will need local Lipschitzness. For vector fields we can define the following [OF18, Def. 6]:

Definition 6.1 ((locally) Lipschitz). *A vector field X on $\mathcal{M}$ is said to be* Lipschitz continuous *on $\Omega \subset \mathcal{M}$ if there exists a constant $L > 0$ such that for all $p, q \in \Omega$ and all γ geodesics joining p to q, there holds*

$$\left\| P^{\gamma}_{p \to q} X(p) - X(q) \right\|_q \leq L \operatorname{len}(\gamma), \quad \forall p, q \in \Omega \tag{218}$$

Moreover, given $p \in \mathcal{M}$, if there exists $\delta > 0$ such that X is Lipschitz continuous on the open ball $B_\delta(p)$, then X is said to be Lipschitz continuous at p. Moreover, if for all $p \in \mathcal{M}$, X is Lipschitz continuous at p, then X is said to be locally Lipschitz continuous *on $\mathcal{M}$.*

Subsequently, we can generalize Rademacher's theorem to Lipschitz vector fields.

Theorem 6.2 ([OF18, Thm. 10]). *If X is a locally Lipschitz continuous vector field on $\mathcal{M}$, then X is almost everywhere differentiable on $\mathcal{M}$.*

Hence, it makes sense to define the generalized covariant derivative [OF18, Def. 11].

Definition 6.3 (Clarke generalized covariant derivative). *The* Clarke generalized covariant derivative *$\partial_{\mathcal{M},C} X$ of a locally Lipschitz continuous vector field X is the set-valued mapping on $\mathcal{M}$ defined as*

$$\partial_{\mathcal{M},C} X(p) := \operatorname{co} \left\{ V \in \mathcal{L}\left(\mathcal{T}_p\mathcal{M}\right) : \exists \{p_k\} \subset \mathcal{D}_X, \lim_{k \to +\infty} p_k = p, \, V = \lim_{k \to +\infty} P_{p_k p} \nabla X\left(p_k\right) \right\}, \tag{219}$$

where $\mathcal{D}_X \subset \mathcal{M}$ is the set on which X is differentiable, "co" represents the convex hull and $\mathcal{L}\left(\mathcal{T}_p\mathcal{M}\right)$ denotes the vector space consisting of all bounded linear operators from $\mathcal{T}_p\mathcal{M}$ to $\mathcal{T}_p\mathcal{M}$.

As with the Euclidean case, we will also need the directional derivative for the notion of semismoothness.

Definition 6.4 (directional derivative)**.** *The* directional derivative *of a vector field X on $\mathcal{M}$ at $p \in \mathcal{M}$ in the direction $v \in \mathcal{T}_p\mathcal{M}$ is defined by*

$$X'(p, v) := \lim_{t \searrow 0} \frac{1}{t} \left[P_{\exp_p(tv) \to p} X \left(\exp_p(tv) \right) - X(p) \right] \in \mathcal{T}_p\mathcal{M}, \tag{220}$$

whenever the limit exists. If this directional derivative exists for every v, then X is said to be directionally differentiable *at p.*

Finally, we are able to generalize the notion of semismoothness to vector fields [OF18, Def. 18]. In [OF18] the equivalent notions of semismoothness as in Thm. 3.4 are used directly instead of the one in Def. 3.3. We will follow [OF18], since this definition is more convenient to work with.

Definition 6.5 (semismooth vector field)**.** *A vector field X on $\mathcal{M}$ that is Lipschitz continuous at $p \in \mathcal{M}$ and directionally differentiable at $q \in B_\delta(p)$ for all directions in $\mathcal{T}_p\mathcal{M}$, is said to be* semismooth *at p iff for every $\epsilon > 0$ there exists $0 < \delta < r_p$, where r_p is the injectivity radius, such that*

$$\|X(p) - P_{q \to p} \left[X(q) + V_q \log_q p \right] \|_p \leq \epsilon d(p, q), \quad \forall q \in B_\delta(p), \quad \forall V_q \in \partial_{\mathcal{M},C} X(q). \tag{221}$$

*The vector field X is said to be μ-*order semismooth *at p for $0 < \mu \leq 1$ iff there exist $\epsilon > 0$ and $0 < \delta < r_p$ such that*

$$\|X(p) - P_{q \to p} \left[X(q) + V_q \log_q p \right] \|_p \leq \epsilon d(p, q)^{1+\mu}, \quad \forall q \in B_\delta(p), \quad \forall V_q \in \partial_{\mathcal{M},C} X(q). \tag{222}$$

Remark 6.6. *The expression $P_{q \to p} \left[X(q) + V_q \log_q p \right]$ can be seen as the linear approximation of X around q, evaluated at p.*

6.2.2 Fast Local Convergence for Semismooth Vector Fields

The Riemannian Semismooth Newton (RSSN) method for finding a zero of a vector field, i.e., $X(p) = 0$, is now a straightforward generalization of the Euclidean case. The method is shown in Alg. 5.

For convergence we also get a the following result, similar result to the linear case.

Theorem 6.7 ([OF18, Thm. 19])**.** *Let X be a locally Lipschitz continuous vector field on $\mathcal{M}$ and $p^* \in \mathcal{M}$ be a solution of problem $X(p) = 0$. Assume that X is semismooth at p^* and all $V_{p^*} \in \partial_{\mathcal{M},C} X(p^*)$ are invertible. Then, there exists a $\delta > 0$ such that for each $p^0 \in B_\delta(p^*) \setminus \{p^*\}$, $(p^k)_{k \geq 0}$ generated by Alg. 5 is well-defined, belongs to $B_\delta(p^*)$ and converges superlinearly to p^* Additionally, if X is μ-order semismooth at p^*, then the convergence of $(p^k)_{k \geq 0}$ to p^* is of order $1 + \mu$.*

Algorithm 5 Riemannian Semismooth Newton

```
Initialization: $p^0 \in \mathcal{M}, k := 0$
while not converged do
    Choose any $V(p^k) \in \partial_{\mathcal{M},C} X(p^k)$
    Solve $V(p^k)d^k = -X(p^k)$ in the vector space $\mathcal{T}_{p^k}\mathcal{M}$
    $p^{k+1} := \exp_{p^k}(d^k)$
    $k := k + 1$
end while
```

6.3 The Inexact Riemannian Semismooth Newton Method*

Before continuing to applications, we will consider a generalization of RSSN, the *Inexact Riemannian Semismooth Newton* (IRSSN) method in Alg. 6. The main motivation for this method is that solving the Newton system with high precision can be very expensive. This is especially the case for higher-dimensional manifolds where the size of the matrix, i.e., the representation of the generalized covariant derivative, scales with the manifold dimension. For example, for a d-dimensional array containing S^2 signals the Newton matrix is already 2^d times larger than for $\mathbb{R}$ and for $\mathcal{P}(3)$ this is already 6^d times larger. Solving the matrix inexactly using an iterative method can ameliorate this problem.

Algorithm 6 Inexact Semismooth Newton

```
Initialization: $p^0 \in \mathcal{M}, a^0 \geq 0, k := 0$
while not converged do
    Choose $V_k(p^k) \in \partial_{\mathcal{M},C} X(p^k)$
    Solve $V_k(p^k)d^k = -X(p^k) + r^k$ in $\mathcal{T}_{p^k}\mathcal{M}$ where $\|r^k\|_{(p^k)} \leq a^k \|X(p^k)\|_{(p^k)}$
    $p^{k+1} := \exp_{p^k}(d^k)$
    Choose $a^{k+1} \geq 0$
    $k := k + 1$
end while
```

In this section we will focus on proving Thm. 6.12: a local convergence result for Riemannian manifolds. The proof presented will be based on the ideas of the Inexact Semismooth Newton methods in $\mathbb{R}^n$ as discussed in [MQ95, FFK96]. The technicalities are inspired by the approach of the convergence proof of Riemannian Semismooth Newton [OF18] as already discussed in the previous section.

6.3.1 Towards a Convergence Proof for Inexact Riemannian Semismooth Newton

To start of with the technicalities, we first need to account for curvature. In particular, we need to account for how geodesics spread. The idea for approaching this originates from [OF18, Def. 2]. That is, we can summarize this information in a number that will take care of all issues regarding curvature in the proof.

Definition 6.8. *Let $p \in \mathcal{M}$ and r_p be the radius of injectivity of $\mathcal{M}$ at p. Define the quantity*

$$K_p := \sup\left\{ \frac{d\left(\exp_q u, \exp_q v\right)}{\|u-v\|_q} : q \in B_{r_p}(p), u, v \in \mathcal{T}_q\mathcal{M}, u \neq v, \|v\|_q \leq r_p, \|u-v\|_q \leq r_p \right\}. \tag{223}$$

The following remark from [OF18] should be considered.

Remark 6.9. *This number K_p measures how fast the geodesics spread apart in $\mathcal{M}$. In particular, when $u = 0 \in \mathcal{T}_q\mathcal{M}$ or more generally when u and v are on the same line through* 0, *$d\left(\exp_q u, \exp_q v\right) = \|u-v\|_q$. Hence, $K_p \geq 1$, for all $p \in \mathcal{M}$. When $\mathcal{M}$ has non-negative sectional curvature (see Def. 4.37), the geodesics spread apart less than the rays, i.e., $d\left(\exp_p u, \exp_p v\right) \leq \|u-v\|_q$ and, in this case, $K_p = 1$ for all $p \in \mathcal{M}$.*

Next, remember the definition of an operator norm.

Definition 6.10. *Let $p \in \mathcal{M}$. The* norm of a linear map *$A : \mathcal{T}_p\mathcal{M} \to \mathcal{T}_p\mathcal{M}$ is defined by*

$$\|A\|_p := \sup\left\{\|Av\|_p : v \in \mathcal{T}_p\mathcal{M}, \|v\|_p \leq 1\right\}. \tag{224}$$

We have the following result.

Lemma 6.11 ([OF18, Lemma 17])**.** *Let X be a locally Lipschitz continuous vector field on $\mathcal{M}$. Assume that all $V_p \in \partial_{\mathcal{M},C}X(p)$ are invertible at $p \in \mathcal{M}$ and let $\lambda_p \geq \max\left\{\|V_p^{-1}\|_p : V_p \in \partial_{\mathcal{M},C}X(p)\right\}$. Then, for every $\epsilon > 0$ satisfying $\epsilon\lambda_p < 1$, there exists $0 < \delta < r_p$ such that all $V_q \in \partial_{\mathcal{M},C}X(q)$ are invertible on $B_\delta(p)$ and*

$$\|V_q^{-1}\|_q \leq \frac{\lambda_p}{1-\epsilon\lambda_p}, \quad \forall q \in B_\delta(p), \quad \forall V_q \in \partial_{\mathcal{M},C}X(q). \tag{225}$$

6.3.2 Fast Local Convergence for Semismooth Vector Fields

With these tools we can move on to the main result of this section.

Theorem 6.12. *Let X be locally Lipschitz continuous vector field on $\mathcal{M}$ and $p^* \in \mathcal{M}$ be a solution of problem the $X(p) = 0$. Assume that X is semismooth at p^* and that all $V_{p^*} \in \partial_{\mathcal{M},C}X(p^*)$ are invertible. Then the following statements hold:*

- *(i) There exist $a > 0$ and $\delta > 0$ such that for every $p^0 \in B_\delta(p^*)$ and $a^k \leq a$, the sequence $(p^k)_{k\geq 0}$ generated by Alg. 6 is well-defined, is contained in $B_\delta(p^*)$ and converges Q-linearly to the solution p^*.*
- *(ii) If the sequence $(p^k)_{k\geq 0}$ generated by Alg. 6 converges to the solution p^* and further $\|r^k\|_{(p^k)} \in o\left(\|X(p^k)\|_{(p^k)}\right)$, then the rate of convergence is Q-superlinear.*

(iii) If the sequence $(p^k)_{k\geq 0}$ generated by Alg. 6 converges to the solution p^*, X is μ-order semismooth at p^*, and $\|r^k\|_{(p^k)} \in O\left(\|X(p^k)\|_{(p^k)}^{1+\mu}\right)$, then the rate of convergence is Q-order $1+\mu$.

Proof. (i) Let K_{p^*} be as defined in Def. 6.8 and let r_{p^*} be the injectivity radius. Since X is locally Lipschitz, there exist constants $\hat{\delta} > 0$ and L such that for all $p \in B_{\hat{\delta}}(p^*)$

$$\|X(p)\|_p = \|P_{p^* \to p} X(p^*) - X(p)\|_p \leq L d(p, p^*). \tag{226}$$

The equality holds since $X(p^*) = 0$ and parallel transport is linear.

Now, since all $V_{p^*} \in \partial_{\mathcal{M},C} X(p^*)$ are invertible at $p^* \in \mathcal{M}$ by assumption, we can take $\lambda_{p^*} \geq \max\{\|V_{p^*}^{-1}\|_{p^*} : V_{p^*} \in \partial_{\mathcal{M},C} X(p^*)\}$. Furthermore, take $a < \frac{1}{\lambda_{p^*} L K_{p^*}}$, choose $a^k \leq a$ $\forall k \in \mathbb{N}$ and ϵ satisfying $\epsilon \lambda_{p^*}(1 + K_{p^*}) < 1 - a\lambda_{p^*} L K_{p^*}$. As $\epsilon\lambda_{p^*} < 1$, by Lemma 6.11 we can find a $0 < \delta < \min\{\hat{\delta}, r_{p^*}\}$ such that for all $p \in B_\delta(p^*)$ and $V_p \in \partial_{\mathcal{M},C} X(p)$

$$\|V_p^{-1}\|_p \leq \frac{\lambda_{p^*}}{1 - \epsilon\lambda_{p^*}}. \tag{227}$$

From the semismoothness of X,

$$\|X(p^*) - P_{p \to p^*}\left[X(p) + V_p \log_p p^*\right]\|_{(p^*)} \leq \epsilon d(p, p^*) \tag{228}$$

by (221).

We now show that for this δ the Newton iteration is well-defined. Let $k \in \mathbb{N}$ and assume that $p^k \in B_\delta(p^*)$. Let d^k be such that

$$\|V_{p^k} d^k + X(p^k)\|_{(p^k)} \leq a^k \|X(p^k)\|_{(p^k)}. \tag{229}$$

Then

$$\begin{aligned}
\|\log_{p^k} p^* - d^k\|_{(p^k)} &= \|\log_{p^k} p^* + V_{p^k}^{-1} X(p^k) - V_{p^k}^{-1}(V_{p^k} d^k + X(p^k))\|_{(p^k)} && (230)\\
&\leq \|\log_{p^k} p^* + V_{p^k}^{-1} X(p^k)\|_{(p^k)} + \|V_{p^k}^{-1}\|_{(p^k)} \|V_{p^k} d^k + X(p^k)\|_{(p^k)} && (231)\\
&\overset{(229)}{\leq} \|\log_{p^k} p^* + V_{p^k}^{-1} X(p^k)\|_{(p^k)} + a^k \|V_{p^k}^{-1}\|_{(p^k)} \|X(p^k)\|_{(p^k)}. && (232)
\end{aligned}$$

Since $X(p^*) = 0$ and parallel transport is an isometry we see that

$$\begin{aligned}
\|\log_{p^k} p^* + V_{p^k}^{-1} X(p^k)\|_{(p^k)} &= \|V_{p^k}^{-1}\left(V_{p^k} \log_{p^k} p^* + X(p^k)\right)\|_{(p^k)} && (233)\\
&\leq \|V_{p^k}^{-1}\|_{(p^k)} \|P_{p^k \to p^*}\left(X(p^k) + V_{p^k} \log_{p^k} p^*\right)\|_{(p^*)} && (234)\\
&\leq \|V_{p^k}^{-1}\|_{(p^k)} \|X(p^*) - P_{p^k \to p^*}\left(X(p^k) + V_{p^k} \log_{p^k} p^*\right)\|_{(p^*)}. && (235)
\end{aligned}$$

Substituting (235) back into (232) we find

$$\|\log_{p^k} p^* - d^k\|_{(p^k)} \leq \|V_{p^k}^{-1}\|_{(p^k)} \left(\|X\left(p^*\right) - P_{p^k \to p^*}\left(X(p^k) + V_{p^k} \log_{p^k} p^*\right)\|_{(p^*)} + a^k \|X(p^k)\|_{(p^k)}\right) \tag{236}$$

$$\overset{(227),(228),(226)}{\leq} \frac{\lambda_{p^*}}{1-\epsilon\lambda_{p^*}}(\epsilon d(p^k,p^*) + a^k L d(p^k,p^*)) \tag{237}$$

$$\overset{a^k \leq a}{\leq} \frac{\lambda_{p^*}}{1-\epsilon\lambda_{p^*}}(\epsilon + aL)d(p^k,p^*). \tag{238}$$

Now note that since $K_{p^*} \geq 1$ (see Remark 6.9) we have

$$\frac{\lambda_{p^*}}{1-\epsilon\lambda_{p^*}}(\epsilon + aL) \leq \frac{\lambda_{p^*}K_{p^*}}{1-\epsilon\lambda_{p^*}}(\epsilon + aL). \tag{239}$$

For our choice of a and ϵ we find

$$\epsilon\lambda_{p^*}\left(1 + K_{p^*}\right) < 1 - a\lambda_{p^*}LK_{p^*} \tag{240}$$

$$\Leftrightarrow \quad \epsilon\lambda_{p^*} + \epsilon\lambda_{p^*}K_{p^*} + a\lambda_{p^*}LK_{p^*} < 1 \tag{241}$$

$$\Leftrightarrow \quad \lambda_{p^*}K_{p^*}(\epsilon + aL) < 1 - \epsilon\lambda_{p^*} \tag{242}$$

$$\Leftrightarrow \quad \frac{\lambda_{p^*}K_{p^*}}{1-\epsilon\lambda_{p^*}}(\epsilon + aL) < 1. \tag{243}$$

Since $d(p^k,p^*) < \delta$, we obtain from combining (238), (239) and (243)

$$\|\log_{p^k} p^* + V_{p^k}^{-1}X(p^k)\|_{(p^k)} < d(p^k,p^*) < \delta \leq r_{p^*}. \tag{244}$$

Moreover, we have $\|\log_{p^k} p^*\|_{(p^k)} = d(p^k,p^*) \leq r_{p^*}$. Hence, we find (see Def. 6.8)

$$d(\exp_{p^k}(d^k),p^*) \leq K_{p^*}\|\log_{p^k} p^* - d^k\|_{(p^k)}. \tag{245}$$

Combining this result we find

$$\frac{d(p^{k+1},p^*)}{d(p^k,p^*)} \overset{Alg.\ 6}{=} \frac{d(\exp_{p^k}(d^k),p^*)}{d(p^k,p^*)} \overset{(245)}{\leq} \frac{K_{p^*}\|\log_{p^k} p^* - d^k\|_{(p^k)}}{d(p^k,p^*)} \tag{246}$$

$$\overset{(238)}{\leq} \frac{\lambda_{p^*}K_{p^*}}{1-\epsilon\lambda_{p^*}}(\epsilon + aL) \overset{(243)}{<} 1. \tag{247}$$

From this result we conclude by induction that if we choose $p^0 \in B_\delta(p^*)$ as in the assumption, we have $p^k \in B_\delta(p^*)\ \forall k \in \mathbb{N}$ and convergence is Q-linear.

(ii) The second part is very similar. Let K_{p^*}, r_{p^*} and the $\hat{\delta}$ with corresponding L as before. Choose $\epsilon > 0$ such that $\epsilon\lambda_{p^*}(1 + 2K_{p^*}) < 1$ and take $0 < \delta < \min\{\hat{\delta}, r_{p^*}\}$ such that (227) and (228) hold. Due to the assumption $\|r^k\|_{(p^k)} \in o\left(\|X(p^k)\|_{(p^k)}\right)$, the assumption that $p^k \to p^*$, and that X is continuous, we have that $\|X(p^k)\|_{(p^k)} \to 0$ and moreover for large enough k we have

$$\|r^k\|_{(p^k)} < \epsilon\delta. \tag{248}$$

Because of the convergence assumption $p^k \to p^*$, we also have for large k that $d(p^k, p^*) < \delta$. Consequently, using the similar steps that lead to (238) from (i) we see

$$\begin{aligned} \|\log_{p^k} p^* - d^k\|_{(p^k)} &\leq \frac{\lambda_{p^*}}{1-\epsilon\lambda_{p^*}}(\epsilon d(p^k, p^*) + \|r^k\|_{(p^k)}) &(249)\\ &\leq \frac{2\epsilon\lambda_{p^*}}{1-\epsilon\lambda_{p^*}}\delta \leq \frac{2\epsilon\lambda_{p^*}K_{p^*}}{1-\epsilon\lambda_{p^*}}\delta. &(250)\end{aligned}$$

For our choice of ϵ we find

$$\begin{aligned} &\epsilon\lambda_{p^*}\,(1+2K_{p^*}) < 1 &(251)\\ \Leftrightarrow\quad &2\epsilon\lambda_{p^*}K_{p^*} < 1-\epsilon\lambda_{p^*} &(252)\\ \Leftrightarrow\quad &\frac{2\epsilon\lambda_{p^*}K_{p^*}}{1-\epsilon\lambda_{p^*}} < 1. &(253)\end{aligned}$$

Since $d(p^k, p^*) < \delta$, we obtain from combining (250) and (253)

$$\|\log_{p^k} p^* + V_{p^k}^{-1}X(p^k)\|_{(p^k)} < \delta \leq r_{p^*}. \tag{254}$$

Again, we have $\|\log_{p^k} p^*\|_{(p^k)} = d(p^k, p^*) \leq r_{p^*}$ and we find (see Def. 6.8)

$$d(\exp_{p^k}(d^k), p^*) \leq K_{p^*}\|\log_{p^k} p^* - d^k\|_{(p^k)}. \tag{255}$$

Finally we see that for large k

$$\begin{aligned} \frac{d(p^{k+1}, p^*)}{d(p^k, p^*)} &\overset{Alg.\ 6}{=} \frac{d(\exp_{p^k}(d^k), p^*)}{d(p^k, p^*)} \overset{(255)}{\leq} \frac{K_{p^*}\|\log_{p^k} p^* - d^k\|_{(p^k)}}{d(p^k, p^*)} &(256)\\ &\overset{(249)}{\leq} \frac{\epsilon\lambda_{p^*}K_{p^*}}{1-\epsilon\lambda_{p^*}} + \frac{\lambda_{p^*}K_{p^*}}{1-\epsilon\lambda_{p^*}}\frac{\|r^k\|_{(p^k)}}{d(p^k, p^*)} &(257)\end{aligned}$$

and by $\|X(p^k)\|_{(p^k)} \leq Ld(p^k, p^*)$

$$\leq \frac{\epsilon\lambda_{p^*}K_{p^*}}{1-\epsilon\lambda_{p^*}} + \frac{1}{L}\frac{\lambda_{p^*}K_{p^*}}{1-\epsilon\lambda_{p^*}}\frac{\|r^k\|_{(p^k)}}{\|X(p^k)\|_{(p^k)}} \tag{258}$$

and note that this expression holds for all (arbitrarily small) $\epsilon > 0$ such that $\epsilon\lambda_{p^*}(1 + 2K_{p^*}) < 1$. Hence, we can focus solely on the residual term and see by our assumption $\|r^k\|_{(p^k)} \in o(\|X(p^k)\|_{(p^k)})$ that

$$\lim_{k\to\infty}\frac{d(p^{k+1}, p^*)}{d(p^k, p^*)} \leq \lim_{k\to\infty}\frac{\lambda_{p^*}K_{p^*}}{L(1-\epsilon\lambda_{p^*})}\frac{\|r^k\|_{(p^k)}}{\|X(p^k)\|_{(p^k)}} = 0 \tag{259}$$

and conclude that the convergence is superlinear.

(iii) Again this is very similar to the previous cases. Let μ denote the order of the semismoothness and let K_{p^*}, r_{p^*} and $\hat{\delta}$ with corresponding L as before. For $\epsilon > 0$ such that $\epsilon\lambda_{p^*} < 1$, choose $0 < \delta < \min\{\hat{\delta}, r_{p^*}\}$ satisfying $\epsilon\lambda_{p^*}(1 + 2\delta^\mu K_{p^*}) < 1$ and such that (227) and

$$\left\|X(p^*) - P_{p\to p^*}\left[X(p) + V_p \exp_p^{-1} p^*\right]\right\|_{(p^*)} \leq \epsilon d(p, p^*)^{1+\mu} \tag{260}$$

hold. Then for large enough k we have by the same reasoning as for establishing (248), that

$$\|r^k\|_{(p^k)} < \epsilon\delta^{1+\mu}. \tag{261}$$

Similarly as in (253)

$$\begin{aligned} \|\log_{p^k} p^* - d^k\|_{(p^k)} &\leq \frac{\lambda_{p^*}}{1 - \epsilon\lambda_{p^*}}(\epsilon d(p^k, p^*)^{1+\mu} + \|r^k\|_{(p^k)}) && (262)\\ &\leq \frac{2\epsilon\lambda_{p^*}\delta^\mu}{1 - \epsilon\lambda_{p^*}}\delta \leq \frac{2\epsilon\lambda_{p^*}\delta^\mu K_{p^*}}{1 - \epsilon\lambda_{p^*}}\delta && (263)\end{aligned}$$

follows and for our choice of ϵ and δ we find

$$\epsilon\lambda_{p^*}\left(1 + 2\delta^\mu K_{p^*}\right) < 1 \Leftrightarrow \frac{2\epsilon\lambda_{p^*}\delta^\mu K_{p^*}}{1 - \epsilon\lambda_{p^*}} < 1. \tag{264}$$

Using similar arguments as for establishing (258) in (ii), we obtain

$$\frac{d(p^{k+1}, p^*)}{d(p^k, p^*)} \leq \frac{\epsilon\lambda_{p^*}K_{p^*}}{1 - \epsilon\lambda_{p^*}} + \frac{1}{L}\frac{\lambda_{p^*}K_{p^*}}{1 - \epsilon\lambda_{p^*}}\frac{\|r^k\|_{(p^k)}}{\|X(p^k)\|_{(p^k)}^{1+\mu}}. \tag{265}$$

Because ϵ can be arbitrarily small, we can again focus on the second term as in (ii). Finally, we see by our assumption $\|r^k\|_{(p^k)} = \mathcal{O}(\|X(p^k)\|_{(p^k)}^{1+\mu})$ that

$$\lim_{k\to\infty} \frac{d(p^{k+1}, p^*)}{d(p^k, p^*)^{1+\mu}} \leq \lim_{k\to\infty} \frac{\lambda_{p^*}K_{p^*}}{L(1 - \epsilon\lambda_{p^*})}\frac{\|r^k\|_{(p^k)}}{\|X(p^k)\|_{(p^k)}^{1+\mu}} =: M \tag{266}$$

for some $M > 0$ and conclude that the convergence is Q-order $1 + \mu$. □

In particular, from (ii) we obtain the following result.

Corollary 6.12.1. *If the sequence $(p^k)_{k\geq 0}$ generated by Alg. 6 converges to the solution p^* and sequence $\{a^k\}$ converges to zero, then the rate of convergence is Q-superlinear.*

Remark 6.13. *We also like to note that in the linear case (ii) and (iii) are formulated stronger: if a convergent sequence exists, the converse statements in (ii) and (iii) also hold. However, the proof in [FFK96] relies heavily on the linearity of $\mathbb{R}^d$ and is therefore much harder to translate to the manifold case. This remains an open problem.*

6.3.3 Observations

Before moving on to applications, we will elaborate on some key observations in the theory of the RSSN method. Although the proof of the RSSN convergence is skipped in this work, we would like to mention that it is very similar to that of the inexact variant. In particular, we can make the following observations regarding the convergence behaviour of both algorithms. By a continuity argument, we see that the minimal radius of convergence is determined by ϵ. This ϵ must satisfy the inequality (that is for $a^k = 0$)

$$\epsilon\lambda_{p^*}\left(1 + K_{p^*}\right) < 1. \tag{267}$$

In other words, a too small ϵ can give trouble in getting the RSSN to work.

We see that we run into trouble in two cases: a large λ_{p^*} and a large K_{p^*} correspondence. This comes down to the following scenarios.

The generalized covariant derivative is close to singular: Here a large λ_{p^*} only admits a very small ϵ, which in turn results in a small convergence region.

The manifold has negative curvature: For negatively curved manifolds we do not have an a priori estimate for K_{p^*}. If this value becomes arbitrarily large, we cannot expect a large region of convergence.

6.4 A Higher-order Primal-dual Method for Manifolds*

After having considered the Semismooth Newton methods for finding zeros $X(p) = 0$, we now come back to the original problem

$$\inf_{p\in\mathcal{M}}\{F(p) + G(\Lambda(p))\}. \tag{268}$$

Assuming that the functions are proper, lower semi-continuous and convex we rewrite this problem as a saddle-point problem

$$\inf_{p\in\mathcal{M}}\sup_{\xi_n\in\mathcal{T}_n^*\mathcal{N}}\{F(p) + \langle\log_n\Lambda(p),\xi_n\rangle - G_n^*\left(\xi_n\right)\}. \tag{269}$$

As we saw in Sect. 5.3 for the manifold case we could not solve this problem directly, but had to pass to approximations which gave two different optimality systems: the exact and the linearized optimality system.

Exact optimality

The exact optimality system (Sect. 5.3)

$$\mathcal{P}_{m\to p}\left(-(D_m\Lambda)^*\left[\mathcal{P}_{n\to\Lambda(m)}\xi_n\right]\right) \in \partial_{\mathcal{M}}F(p), \tag{270}$$

$$\log_n\Lambda(p) \in \partial G_n^*\left(\xi_n\right), \tag{271}$$

can be rewritten into the system

$$p = \operatorname{prox}_{\sigma F}\left(\exp_p\left(\mathcal{P}_{m\to p}\left(-\sigma(D_m\Lambda)^*\left[\mathcal{P}_{n\to\Lambda(m)}\xi_n\right]\right)^\sharp\right)\right), \tag{272}$$

$$\xi_n = \operatorname{prox}_{\tau G_n^*}\left(\xi_n + \tau\left(\log_n\Lambda(p)\right)^\flat\right), \tag{273}$$

where $\sigma, \tau > 0$.

Linearized optimality
For the linearized optimality system we approximated the saddle-point problem even further as in (Sect. 5.3) and instead solve

$$\inf_{p\in\mathcal{M}}\inf_{\xi_n\in\mathcal{T}_n^*\mathcal{N}}\{F(p) + \langle D_m\Lambda\left[\log_m p\right], \xi_n\rangle - G_n^*\left(\xi_n\right)\}. \tag{274}$$

Then the linearized optimality system

$$\mathcal{P}_{m\to p}\left(-(D_m\Lambda)^*\left[\xi_n\right]\right) \in \partial_{\mathcal{M}}F(p), \tag{275}$$

$$D_m\Lambda\left[\log_m p\right] \in \partial G_n^*\left(\xi_n\right), \tag{276}$$

can be rewritten into

$$p = \operatorname{prox}_{\sigma F}\left(\exp_p\left(\mathcal{P}_{m\to p}\left(-\sigma\left(D_m\Lambda\right)^*\left[\mathcal{P}_{n\to\Lambda(m)}\xi_n\right]\right)^\sharp\right)\right), \tag{277}$$

$$\xi_n = \operatorname{prox}_{\tau G_n^*}\left(\xi_n + \tau\left(\mathcal{P}_{\Lambda(m)\to n}D_m\Lambda\left[\log_m p\right]\right)^\flat\right), \tag{278}$$

where $\sigma, \tau > 0$.

The next step is to find a way to rewrite both of these systems of non-linear equations into something that can be solved by the Riemannian Semismooth Newton method.

6.4.1 Choosing a Type of Newton Method

Whereas the dual variable ξ_n lives in a vector space and both

$$\xi_n - \operatorname{prox}_{\tau G_n^*}\left(\xi_n + \tau\left(\log_n\Lambda(p)\right)^\flat\right) = 0 \tag{279}$$

for the exact dual optimality condition and

$$\xi_n - \operatorname{prox}_{\tau G_n^*}\left(\xi_n + \tau\left(\mathcal{P}_{\Lambda(m)\to n}D_m\Lambda\left[\log_m p\right]\right)^\flat\right) = 0 \tag{280}$$

for the linearized dual optimality condition make sense, this is unfortunately not necessarily the case for the primal variable. Here, we need a more general approach. Two approaches can be chosen:

Constructing a Vector Space
For $\ell \in \mathcal{M}$, the tangent space $\mathcal{T}_\ell\mathcal{M}$ is fixed and (272), (277) can be rewritten as

$$\log_\ell p - \log_\ell \operatorname{prox}_{\sigma F}\left(\exp_p\left(\mathcal{P}_{m\to p}\left(-\sigma\left(D_m\Lambda\right)^*\left[\mathcal{P}_{n\to\Lambda(m)}\xi_n\right]\right)^\sharp\right)\right) = 0. \tag{281}$$

In the case of the exact optimality conditions, the mapping $X : \mathcal{M}\times\mathcal{T}_n^*\mathcal{N} \to \mathcal{T}_\ell\mathcal{M}\times\mathcal{T}_n^*\mathcal{N}$ is defined as

$$X(p,\xi_n) := \begin{pmatrix} \log_\ell p - \log_\ell \operatorname{prox}_{\sigma F}\left(\exp_p\left(\mathcal{P}_{m\to p}\left(-\sigma\left(D_m\Lambda\right)^*\left[\mathcal{P}_{n\to\Lambda(m)}\xi_n\right]\right)^\sharp\right)\right) \\ \xi_n - \operatorname{prox}_{\tau G_n^*}\left(\xi_n + \tau\left(\log_n \Lambda\left(p\right)\right)^\flat\right) \end{pmatrix} \tag{282}$$

and the resulting non-linear system of equations is $X(p,\xi_n) = 0$. However, this approach has a major drawback. Although in the case of flat and negatively curved spaces we find

$$\log_\ell p = \log_\ell \operatorname{prox}_{\sigma F}\left(\exp_p\left(\mathcal{P}_{m\to p}\left(-\sigma\left(D_m\Lambda\right)^*\left[\mathcal{P}_{n\to\Lambda(m)}\xi_n\right]\right)^\sharp\right)\right) \tag{283}$$

$$\Leftrightarrow \quad p = \operatorname{prox}_{\sigma F}\left(\exp_p\left(\mathcal{P}_{m\to p}\left(-\sigma\left(D_m\Lambda\right)^*\left[\mathcal{P}_{n\to\Lambda(m)}\xi_n\right]\right)^\sharp\right)\right), \tag{284}$$

by the uniqueness of geodesics, this is not necessarily the case for manifolds with positive curvature. For example, in the case of S^2 we can see the arising issue very clearly. Imagine that ℓ lives on the south pole, but the optimal p lies on the north pole of the sphere. Now, given that at some point in the RSSN process both p and the point resulting from the proximal mapping lie close to the north pole as well (i.e., the algorithm has almost converged), it could be that the $\log_\ell \cdot$ operator gives two very different, even opposite directed, tangent vectors. In that case its difference does not vanish and it would seem like we are not converged at all.

Constructing a Vector Field
The other approach would be to construct a vector field for the primal variables and (272), (277) are rewritten as

$$-\log_p \operatorname{prox}_{\sigma F}\left(\exp_p\left(\mathcal{P}_{m\to p}\left(-\sigma\left(D_m\Lambda\right)^*\left[\mathcal{P}_{n\to\Lambda(m)}\xi_n\right]\right)^\sharp\right)\right) = 0. \tag{285}$$

In the case of the exact optimality conditions, the mapping $X : \mathcal{M}\times\mathcal{T}_n^*\mathcal{N} \to \mathcal{T}\mathcal{M}\times\mathcal{T}_n^*\mathcal{N}$ is now defined as

$$X(p,\xi_n) := \begin{pmatrix} -\log_p \operatorname{prox}_{\sigma F}\left(\exp_p\left(\mathcal{P}_{m\to p}\left(-\sigma\left(D_m\Lambda\right)^*\left[\mathcal{P}_{n\to\Lambda(m)}\xi_n\right]\right)^\sharp\right)\right) \\ \xi_n - \operatorname{prox}_{\tau G_n^*}\left(\xi_n + \tau\left(\log_n \Lambda\left(p\right)\right)^\flat\right) \end{pmatrix} \tag{286}$$

and the resulting non-linear system of equations is $X(p,\xi_n) = 0$.

The major drawback of the vector space approach does not occur when using (285). Finding a zero tangent vector always corresponds to (272), (277). Therefore, in the following we will focus on developing the Newton systems for the exact and the linearized optimality system following the vector field approach.

6.4.2 Generalized Differentials and a General Newton Matrix

Now that the vector fields

$$X_e(p,\xi_n) := \begin{pmatrix} -\log_p \operatorname{prox}_{\sigma F}\left(\exp_p\left(\mathcal{P}_{m\to p}\left(-\sigma\left(D_m\Lambda\right)^*\left[\mathcal{P}_{n\to\Lambda(m)}\xi_n\right]\right)^\sharp\right)\right) \\ \xi_n - \operatorname{prox}_{\tau G_n^*}\left(\xi_n + \tau\left(\log_n \Lambda\left(p\right)\right)^\flat\right) \end{pmatrix} \tag{287}$$

and

$$X_l(p,\xi_n) := \begin{pmatrix} -\log_p \operatorname{prox}_{\sigma F}\left(\exp_p\left(\mathcal{P}_{m\to p}\left(-\sigma\left(D_m\Lambda\right)^*\left[\mathcal{P}_{n\to\Lambda(m)}\xi_n\right]\right)^\sharp\right)\right) \\ \xi_n - \operatorname{prox}_{\tau G_n^*}\left(\xi_n + \tau\left(\mathcal{P}_{\Lambda(m)\to n} D_m\Lambda\left[\log_m p\right]\right)^\flat\right) \end{pmatrix} \tag{288}$$

have been constructed, we pass to a general

$$X(p,\xi_n) := (X_1(p,\xi_n), X_2(p,\xi_n)) := (-\log_p f_1(p,\xi_n), \xi_n - f_2(p,\xi_n)), \tag{289}$$

where $f_1 : \mathcal{M}\times\mathcal{T}_n^*\mathcal{N} \to \mathcal{M}$ and $f_2 : \mathcal{M}\times\mathcal{T}_n^*\mathcal{N} \to \mathcal{T}_n^*\mathcal{N}$ are the mappings corresponding to either the exact or the linearized system. In order to proceed with RSSN, the generalized covariant derivative of X must be constructed. The main idea of finding the Newton operator, both for the exact and the linearized system, follows from rewriting the covariant derivative along $\gamma := (\gamma_1,\gamma_2) : (-\epsilon,\epsilon) \to \mathcal{M}\times\mathcal{T}_n^*\mathcal{N}$ as separate contributions from $\gamma_1 : (-\epsilon,\epsilon)\to\mathcal{M}$ and from $\gamma_2 : (-\epsilon,\epsilon)\to\mathcal{T}_n^*\mathcal{N}$. Let M be the dimension of $\mathcal{M}$ and N the dimension of $\mathcal{T}_n^*\mathcal{N}$. Then, by passing to normal coordinates (U,ϕ) on $\mathcal{M}$ and (V,ψ) on $\mathcal{T}_n^*\mathcal{N}$ originating at (p,ξ_n)[7] we can find basis $\{\Theta\}_i^M$ and $\{\Xi\}_j^N$ and coordinates $u_1^i\in C^\infty((-\epsilon,\epsilon))$, $x_1^i\in C^\infty((-\epsilon,\epsilon))$ for $i=1,...,M$ and $u_2^j\in C^\infty((-\epsilon,\epsilon))$, $x_2^j\in C^\infty((-\epsilon,\epsilon))$ for $j=1,...,N$ and write

$$\gamma_1 = \sum_i^M u_1^i\Theta_i \quad\text{and}\quad X_1 = \sum_i^N x_1^i\Theta_i \tag{290}$$

and

$$\gamma_2 = \sum_j^M u_2^j\Xi_j \quad\text{and}\quad X_2 = \sum_j^N x_2^j\Xi_j. \tag{291}$$

If we now first assume that X is smooth. Then, bringing the latter expressions this into the covariant derivative as in (127) and denoting $\Psi_k\in\{\Theta\}_i^M\cup\{\Xi\}_j^N$ as either basis vector we can write

[7] Note that we might as well say $(p,0)$ since the second component is a vector space and hence we have $\mathcal{T}_{\xi_n}\mathcal{T}_n^*\mathcal{N}\cong\mathcal{T}_n^*\mathcal{N}$,i.e., we might as well take the origin here.

$$\frac{DX}{dt} = \sum_{k}^{M+N} \left(\frac{dx^k}{dt} + \sum_{i,j}^{M+N} \Gamma_{ij}^k x^i u^j \right) \Psi_k \tag{292}$$

$$= \sum_{i}^{M} \left(\frac{dx_1^k}{dt} \right) \Theta_i + \sum_{j}^{N} \left(\frac{dx_2^k}{dt} \right) \Xi_j + \sum_{k}^{M+N} \left(\sum_{i,j}^{M+M} \Gamma_{ij}^k x^i u^j \right) \Psi_k \tag{293}$$

$$= \sum_{i}^{M} \left(\frac{dx_1^k}{dt_1} + \frac{dx_1^k}{dt_2} \right) \Theta_i + \sum_{j}^{N} \left(\frac{dx_2^k}{dt_1} + \frac{dx_2^k}{dt_2} \right) \Xi_j + \sum_{k}^{M+N} \left(\sum_{i,j}^{M+M} \Gamma_{ij}^k x^i u^j \right) \Psi_k \tag{294}$$

$$= \frac{DX_1}{dt_1} + \sum_{i}^{M} \left(\frac{dx_1^k}{dt_2} \right) \Theta_i + \sum_{j}^{N} \left(\frac{dx_2^k}{dt_1} \right) \Xi_j + \frac{DX_2}{dt_2} + \sum_{k}^{M+N} \left(\sum_{i,j}^{M+M} \Gamma_{ij}^k x^i u^j \right) \Psi_k. \tag{295}$$

We need the covariant derivative at $t = 0$, i.e., at (p, ξ_n). So let $\eta_p = \dot{\gamma}_1(0)$ and $\eta_\xi = \dot{\gamma}_2(0)$. Also note that $\Gamma_{ij}^k|_{t=0} = 0$ since we work in normal coordinates. Then

$$\nabla_{\eta_p + \eta_{\xi_n}} X = \left. \frac{DX}{dt} \right|_{t=0} \tag{296}$$

$$= \left. \frac{DX_1}{dt_1} \right|_{t=0} + \sum_{i}^{M} \left(\frac{dx_1^k}{dt_2} \right) \Bigg|_{t=0} \Theta_i + \sum_{j}^{N} \left(\frac{dx_2^k}{dt_1} \right) \Bigg|_{t=0} \Xi_j + \left. \frac{DX_2}{dt_2} \right|_{t=0} \tag{297}$$

$$= \left. \frac{DX_1}{dt_1} \right|_{t=0} + D_{\xi_n} X_1[\eta_{\xi_n}] + D_p X_2[\eta_p] + \left. \frac{DX_2}{dt_2} \right|_{t=0} \tag{298}$$

$$= -\frac{D}{dt_1} \left(\log_{(\cdot)} f_1((\cdot), \xi_n) \right) \Bigg|_{t=0} + D_{\xi_n} X_1[\eta_{\xi_n}] + D_p X_2[\eta_p] + \frac{D}{dt_2} \left((\cdot) - f_2(p, \cdot) \right) \Bigg|_{t=0}. \tag{299}$$

Using the chain rule for the first term and $\frac{D}{dt_2}(\cdot)|_{t=0} = I[\eta_{\xi_n}] = \eta_{\xi_n}$ for the last one we get

$$= -\frac{D}{dt_1} \left(\log_{(\cdot)} f_1(p, \xi_n) \right) \Bigg|_{t=0} - \frac{D}{dt_1} \left(\log_p f_1(\cdot, \xi_n) \right) \Bigg|_{t=0} + D_{\xi_n} X_1[\eta_{\xi_n}]$$

$$+ D_p X_2[\eta_p] + \eta_{\xi_n} - \frac{D}{dt_2} \left(f_2(p, \cdot) \right) \Bigg|_{t=0} \tag{300}$$

$$= -\nabla_{\eta_p} \left(\log_{(\cdot)} f_1(p, \xi_n) \right) - D_p \left(\log_p f_1((\cdot), \xi_n) \right) [\eta_p] + D_{\xi_n} X_1[\eta_{\xi_n}]$$

$$+ D_p X_2[\eta_p] + \eta_{\xi_n} - \nabla_{\eta_{\xi_n}} f(p, \cdot). \tag{301}$$

For our purposes, we do not have a smooth X, but as for now we have only discussed generalized covariant derivatives, whereas we see from the expression above that we also need a generalization of the differential. First we need another generalization of Rademacher's theorem. The result in Sect. 6.2 helps us to formulate the following result.

Theorem 6.14. *Let $\mathcal{M}$ and $\mathcal{N}$ be smooth manifolds. If $F : \mathcal{M} \to \mathcal{N}$ is a locally Lipschitz continuous function, then F is almost everywhere differentiable on $\mathcal{M}$.*

Proof. The proof is the same as [OF18, Thm. 10] only with a general manifold $\mathcal{N}$ instead of $\mathcal{TM}$. □

The generalized differential can now be defined as follows.

Definition 6.15 (Clarke generalized differential)**.** *The* Clarke generalized differential *$D_C F$ of a locally Lipschitz continuous function $F : \mathcal{M} \to \mathcal{N}$ is the set-valued mapping defined as*

$$D_C F(p) := \operatorname{co}\left\{V \in \mathcal{L}\left(\mathcal{T}_p\mathcal{M}, \mathcal{T}_{F(p)}\mathcal{N}\right) : \exists \{p_k\} \subset \mathcal{D}_F, \lim_{k\to+\infty} p_k = p,\ V = \lim_{k\to+\infty} D_{p^k}F\right\} \tag{302}$$

where $\mathcal{D}_F \subset \mathcal{M}$ is the set on which F is differentiable, "co" represents the convex hull and $\mathcal{L}\left(\mathcal{T}_p\mathcal{M}, \mathcal{T}_{F(p)}\mathcal{N}\right)$ denotes the vector space consisting of all bounded linear operator from $\mathcal{T}_p\mathcal{M}$ to $\mathcal{T}_{F(p)}\mathcal{N}$.

Finally, the generalized covariant derivative will have the form

$$\partial_{\mathcal{M},C}X(p,\xi_n) = \begin{bmatrix} \partial_{\mathcal{M},C,p}X_1 & D_{C,\xi_n}X_1 \\ D_{C,p}X_2 & \partial_{C,\xi_n}X_2 \end{bmatrix}, \tag{303}$$

where

$$\partial_{\mathcal{M},C,p}X_1 = -\partial_{C,\mathcal{M},p}\left(\log_{(\cdot)} f_1(p,\xi_n)\right) - D_{C,p}\left(\log_p f_1((\cdot),\xi_n)\right), \tag{304}$$

$$\partial_{C,\xi_n}X_2 = I - \partial_{C,\xi_n} f(p,\cdot). \tag{305}$$

Actually computing the generalized covariant derivative (303) for a general manifold can be complicated. In this work we will focus on symmetric manifolds as discussed in Sect. 4.1.2. In Sect. 4.1.3 we saw that for symmetric manifolds we know the differentials and/or covariant derivatives of geodesics, exponential and logarithmic maps. If we also use that in symmetric spaces the parallel transport map is exactly the pole ladder [Pen18] - which is also formulated in terms of logarithmic, exponential and geodesic maps - we have exact expressions for the full generalized covariant derivative, with the exception of the differentials of the proximal maps. As we shall see in Sect. 6.5, for ℓ^2-TV we even have exact expressions for the proxes on top of that.

6.4.3 The Generalized Covariant Derivative for the Exact Newton System

Let $\mathcal{M}$ and $\mathcal{N}$ be symmetric manifolds. In the case of the exact optimality conditions (272) and (273) we constructed the vector field $X_e : \mathcal{M} \times \mathcal{T}_n^*\mathcal{N} \to \mathcal{TM} \times \mathcal{T}_n^*\mathcal{N}$ given by

$$X_e(p,\xi) = \begin{pmatrix} -\log_p \operatorname{prox}_{\sigma F}\left(\exp_p\left(\mathcal{P}_{m\to p}\left(-\sigma(D_m\Lambda)^*\left[\mathcal{P}_{n\to\Lambda(m)}\xi_n\right]\right)^\sharp\right)\right) \\ \xi_n - \operatorname{prox}_{\tau G_n^*}\left(\xi_n + \tau\left(\log_n \Lambda(p)\right)^\flat\right) \end{pmatrix}. \tag{306}$$

In the following the four components of the operator $V_e \in \mathcal{L}(\mathcal{T}_p\mathcal{M} \times \mathcal{T}_n^*\mathcal{N})$ will be constructed so that

$$V_e = \begin{bmatrix} V_{e,1,p} & V_{e,1,\xi_n} \\ V_{e,2,p} & V_{e,2,\xi_n} \end{bmatrix} \in \partial_{\mathcal{M},C} X_e(p, \xi_n). \tag{307}$$

Compute $V_{e,1,p}$
For $\partial_{\mathcal{M},C,p}X_{e,1}$ we need to compute

$$\begin{aligned} \partial_{\mathcal{M},C,p}X_{e,1} = &-\partial_{C,p} \log_{(\cdot)} \operatorname{prox}_{\sigma F}\left(\exp_p\left(\mathcal{P}_{m\to p}\left(-\sigma(D_m\Lambda)^*\left[\mathcal{P}_{n\to\Lambda(m)}\xi_n\right]\right)^\sharp\right)\right) \\ &- D_{C,p}\log_p \operatorname{prox}_{\sigma F}\left(\exp_{(\cdot)}\left(\mathcal{P}_{m\to(\cdot)}\left(-\sigma(D_m\Lambda)^*\left[\mathcal{P}_{n\to\Lambda(m)}\xi_n\right]\right)^\sharp\right)\right). \end{aligned} \tag{308}$$

In order to compute the second term, we will use the chain rule. For that we will pass to the the pole ladder

$$\begin{aligned} P^{\mathrm{P}}_{m\to p}(\xi) &= -\log_p\left(\gamma\left(\exp_m(\xi), \gamma\left(m, p; \frac{1}{2}\right); 2\right)\right) \qquad (309) \\ &= -\log_p\left(\exp_{\exp_m(\xi)}\left(2\log_{\exp_m(\xi)}\gamma\left(m, p; \frac{1}{2}\right)\right)\right) \qquad (310) \end{aligned}$$

as substitution for the parallel transport map. Remember that this is exact for symmetric manifolds. If we rewrite the pole ladder as in (310) then we get for the differential

$$\begin{aligned} \partial_{\mathcal{M},C,p}X_{e,1} = &-\partial_{C,p} \log_{(\cdot)} \operatorname{prox}_{\sigma F}\left(\exp_p\left(\mathcal{P}_{m\to p}\left(-\sigma(D_m\Lambda)^*\left[\mathcal{P}_{n\to\Lambda(m)}\xi_n\right]\right)^\sharp\right)\right) \\ &- D_{C,p}\log_p \operatorname{prox}_{\sigma F}\left(\exp_{(\cdot)}\left(-\log_{(\cdot)}\left(\exp_{\exp_m(q_\xi(\xi))}\left(2\log_{\exp_m(q_\xi(\xi))}\gamma\left(m, (\cdot); \frac{1}{2}\right)\right)\right)\right)\right). \end{aligned} \tag{311}$$

Now, let

$$\begin{aligned} q_0(p,\xi) &:= \operatorname{prox}_{\sigma F}\left(\exp_p\left(\mathcal{P}_{m\to p}\left(-\sigma(D_m\Lambda)^*\left[\mathcal{P}_{n\to\Lambda(m)}\xi_n\right]\right)^\sharp\right)\right), && (312a) \\ q_1(p,\xi) &:= \operatorname{prox}_{\sigma F}\left(q_2(p,\xi)\right), && (312b) \\ q_2(p,\xi) &:= \exp_p\left(q_3(p,\xi)\right), && (312c) \\ q_3(p,\xi) &:= -\log_p(q_4(p,\xi)), && (312d) \\ q_4(p,\xi) &:= \exp_{\exp_m(q_\xi(\xi))}(q_5(p,\xi)), && (312e) \\ q_5(p,\xi) &:= 2\log_{\exp_m(q_\xi(\xi))}(q_p(p)), && (312f) \\ q_\xi(\xi) &:= \left(-\sigma(D_m\Lambda)^*\left[\mathcal{P}_{n\to\Lambda(m)}\xi_n\right]\right)^\sharp, && (312g) \\ q_p(p) &:= \gamma\left(m, p; 1/2\right). && (312h) \end{aligned}$$

Then

$$V_{e,1,p} := A_1 + A_2 B \left[C_1 + C_2 \left[D_1 + D_2 EFG\right]\right] \in \partial_{\mathcal{M},C,p} X_{e,1}, \tag{313}$$

where

$$\begin{aligned}
A_1 &:= -\nabla \log_{(\cdot)} q_0(p,\xi)\big|_p, && (314a)\\
A_2 &:= -D_{q_1(p,\xi)} \log_p(\cdot), && (314b)\\
B &\in D_{C,q_2(p,\xi)} \operatorname{prox}_{\sigma F}(\cdot), && (314c)\\
C_1 &:= D_p \exp_{(\cdot)} q_3(p,\xi), && (314d)\\
C_2 &:= D_{q_3(p,\xi)} \exp_p(\cdot), && (314e)\\
D_1 &:= -(D_p \log_{(\cdot)} q_4(p,\xi))^v = (\nabla \log_{(\cdot)} q_4(p,\xi))\big|_p, && (314f)\\
D_2 &:= -D_{q_4(p,\xi)} \log_p(\cdot), && (314g)\\
E &:= D_{q_5(p,\xi)} \exp_{\exp_m(q_\xi(\xi))}(\cdot), && (314h)\\
F &:= 2D_{q_p(p)} \log_{\exp_m(q_\xi(\xi))}(\cdot), && (314i)\\
G &:= D_p \gamma\left(m,(\cdot);1/2\right). && (314j)
\end{aligned}$$

Note that the only non-smoothness can come from the proximal mapping.

Compute $V_{e,1,\xi_n}$
For $D_{C,\xi_n} X_{e,1}$ we need to compute

$$D_{C,\xi_n} X_{e,1} = -D_{C,\xi_n} \log_p \operatorname{prox}_{\sigma F}\left(\exp_p\left(\mathcal{P}_{m\to p}\left(-\sigma(D_m\Lambda)^*\left[\mathcal{P}_{n\to\Lambda(m)}(\cdot)\right]\right)^\sharp\right)\right). \tag{315}$$

Now note that $H : \mathcal{T}_n^*\mathcal{N} \to \mathcal{T}_p\mathcal{M}$ given by

$$\xi \mapsto \mathcal{P}_{m\to p}\left(-\sigma(D_m\Lambda)^*\left[\mathcal{P}_{n\to\Lambda(m)}\xi\right]\right)^\sharp \tag{316}$$

is a linear map. Hence the differential will be

$$V_{e,1,\xi_n} := A_2 B C_2 H \in D_{C,\xi_n} X_{e,1}, \tag{317}$$

where A_2, B and C_2 as described before.

Compute $V_{e,2,p}$
For $D_{C,p} X_{e,2}$ we need to compute

$$\begin{aligned}
D_{C,p} X_{e,2} &= D_p \xi_n - D_{C,p} \operatorname{prox}_{\tau G_n^*}\left(\xi_n + \tau\left(\log_n \Lambda(p)\right)^\flat\right) && (318)\\
&= -D_{C,p} \operatorname{prox}_{\tau G_n^*}\left(\xi_n + \tau\left(\log_n \Lambda(p)\right)^\flat\right). && (319)
\end{aligned}$$

Let

$$\begin{aligned}
\tilde{\eta}_1(p,\xi) &:= \xi_n + \tau\left(\log_n \Lambda(p)\right)^\flat, && (320a)\\
\eta_p(p) &:= \Lambda(p). && (320b)
\end{aligned}$$

Then

$$V_{e,2,p} := -JKLM \in D_{C,p}X_{e,2}, \tag{321}$$

where

$$\begin{aligned} J &\in D_{\eta_1(p,\xi)} \operatorname{prox}_{\tau G_n^*}(\cdot), && (322a)\\ K &:= \tau\flat, && (322b)\\ L &:= D_{\eta_p(p)} \log_n(\cdot), && (322c)\\ M &:= D_p\Lambda(\cdot). && (322d) \end{aligned}$$

Compute $V_{e,2,\xi_n}$
Finally, for $\partial_{C,\xi}X_{e,2}$ we find

$$\begin{aligned} &\partial_{C,\xi_n}X_{e,2} = I - \partial_{C,\xi_n} \operatorname{prox}_{\tau G_n^*}\left(\xi_n + \tau\left(\log_n \Lambda\left(p\right)\right)^\flat\right) && (323)\\ \Rightarrow\quad &V_{e,2,\xi_n} := I - J \in \partial_{C,\xi_n}X_{e,2}, && (324) \end{aligned}$$

where I is the identity and J as described in the previous step.

6.4.4 The Generalized Covariant Derivative for the Linearized Newton System

Again let $\mathcal{M}$ and $\mathcal{N}$ be symmetric manifolds. In the case of the linearized optimality conditions (277) and (278) we constructed the vector field $X_l : \mathcal{M}\times\mathcal{T}_n^*\mathcal{N} \to \mathcal{T}\mathcal{M}\times\mathcal{T}_n^*\mathcal{N}$ given by

$$X_l(p,\xi) = \begin{pmatrix} -\log_p \operatorname{prox}_{\sigma F}\left(\exp_p\left(\mathcal{P}_{m\to p}\left(-\sigma(D_m\Lambda)^*\left[\mathcal{P}_{n\to\Lambda(m)}\xi_n\right]\right)^\sharp\right)\right) \\ \xi_n - \operatorname{prox}_{\tau G_n^*}\left(\xi_n + \tau\left(\mathcal{P}_{\Lambda(m)\to n}D_m\Lambda\left[\log_m p\right]\right)^\flat\right) \end{pmatrix}. \tag{325}$$

For computing the components of the operator $V_l \in \mathcal{L}(\mathcal{T}_p\mathcal{M}\times\mathcal{T}_n^*\mathcal{N})$, note that only the dual component has changed compared to (306). Hence step 1 and 2 of the previous part are the same: $V_{l,1,p} = V_{e,1,p}$ and $V_{l,1,\xi_n} = V_{e,1,\xi_n}$. For the derivative $\partial_{C,\xi}X_{l,2}$ we will not get anything different as well: $V_{l,2,\xi_n} = V_{l,2,\xi_n}$. So only $V_{l,2,p}$ will be different.

For $D_{C,p}X_{l,2}$ we need to compute

$$\begin{aligned} D_{C,p}X_{l,2} &= D_p\xi_n - D_{C,p} \operatorname{prox}_{\tau G_n^*} \operatorname{prox}_{\tau G_n^*}\left(\xi_n + \tau\left(\mathcal{P}_{\Lambda(m)\to n}D_m\Lambda\left[\log_m p\right]\right)^\flat\right) && (326)\\ &= -D_{C,p} \operatorname{prox}_{\tau G_n^*}\left(\xi_n + \tau\left(\mathcal{P}_{\Lambda(m)\to n}D_m\Lambda\left[\log_m p\right]\right)^\flat\right). && (327) \end{aligned}$$

Let

$$\tilde{\eta}_1(p,\xi) := \xi_n + \tau\mathcal{P}_{\Lambda(m)\to n}D_m\Lambda\left[\log_m p\right]. \tag{328}$$

Then

$$V_{l,2,p} := -\tilde{J}K\tilde{L}\tilde{M} \in D_{C,p}X_{l,2}, \tag{329}$$

where K as before and

$$\begin{align} \tilde{J} &\in D_{\tilde{\eta}_1(p,\xi)} \operatorname{prox}_{\tau G_n^*}(\cdot), \tag{330a}\\ \tilde{L} &:= \mathcal{P}_{\Lambda(m)\to n} D_m\Lambda, \tag{330b}\\ \tilde{M} &:= D_p \log_m(\cdot). \tag{330c} \end{align}$$

6.4.5 Building the Newton Matrix

In the previous we constructed two linear operators: the generalized covariant derivative for the exact and the linearized optimality system. In practice a matrix is preferable. To construct a matrix representation of an operator

$$V = \begin{bmatrix} V_{1,p} & V_{1,\xi_n} \\ V_{2,p} & V_{2,\xi_n} \end{bmatrix} \in \partial_{C,\mathcal{M}} X(p,\xi_n), \tag{331}$$

we will need a basis. Let $\{\Theta_j\}_j$ be an orthonormal basis for $\mathcal{T}_p\mathcal{M}$ and let $\{\Xi_j\}_j$ be a basis for $\mathcal{T}_n^*\mathcal{N}$. Then, we want to find of U and W in this basis, i.e, find $\{u^j\}_j$ and $\{w^j\}_j$ such that

$$U = \sum_{j=1}^{M} u^j \Theta_j \quad \text{and} \quad W = \sum_{j=1}^{N} w^j \Xi_j \tag{332}$$

solve the system

$$\sum_j^M \langle \Theta_i, V_{1,p}[\Theta_j]\rangle u^j + \sum_j^N \langle \Theta_i, V_{1,\xi}[\Xi_j]\rangle w^j = -\langle \Theta_i, X_1\rangle \qquad i = 1,2,...,M, \tag{333}$$

$$\sum_j^M \langle \Xi_i, V_{2,p}[\Theta_j]\rangle u^j + \sum_j^N \langle \Xi_i, V_{2,\xi}[\Xi_j]\rangle w^j = -\langle \Xi_i, X_2\rangle \qquad i = 1,2,...,N, \tag{334}$$

where M is the dimension of $\mathcal{M}$ (and hence also of $\mathcal{T}_p\mathcal{M}$) and N that of $\mathcal{N}$ (and again also that of $\mathcal{T}_n^*\mathcal{N}$). In matrix notation that would be

$$\begin{pmatrix} \langle \Theta_1, V_{1,p}[\Theta_1]\rangle & \cdots & \langle \Theta_1, V_{1,p}[\Theta_M]\rangle & \langle \Theta_1, V_{1,\xi}[\Xi_1]\rangle & \cdots & \langle \Theta_1, V_{1,\xi}[\Xi_N]\rangle \\ \vdots & \ddots & \vdots & \vdots & \ddots & \vdots \\ \langle \Theta_M, V_{1,p}[\Theta_1]\rangle & \cdots & \langle \Theta_M, V_{1,p}[\Theta_M]\rangle & \langle \Theta_M, V_{1,\xi}[\Xi_1]\rangle & \cdots & \langle \Theta_M, V_{1,\xi}[\Xi_N]\rangle \\ \langle \Xi_1, V_{2,p}[\Theta_1]\rangle & \cdots & \langle \Xi_1, V_{2,p}[\Theta M]\rangle & \langle \Xi_1, V_{2,\xi}[\Xi_1]\rangle & \cdots & \langle \Xi_1, V_{2,\xi}[\Xi_N]\rangle \\ \vdots & \ddots & \vdots & \vdots & \ddots & \vdots \\ \langle \Xi_N, V_{2,p}[\Theta_1]\rangle & \cdots & \langle \Xi_N, V_{2,p}[\Theta_M]\rangle & \langle \Xi_N, V_{2,\xi}[\Xi_1]\rangle & \cdots & \langle \Xi_N, V_{2,\xi}[\Xi_N]\rangle \end{pmatrix} \begin{pmatrix} u^1 \\ \vdots \\ u^M \\ w^1 \\ \vdots \\ w^N \end{pmatrix} = - \begin{pmatrix} \langle \Theta_1, X_1\rangle \\ \vdots \\ \langle \Theta_M, X_1\rangle \\ \langle \Xi_1, X_2\rangle \\ \vdots \\ \langle \Xi_N, X_2\rangle \end{pmatrix} \tag{335}$$

Now we are finally ready for applying the algorithms to look at a case study.

6.5 Application to ℓ^2-TV-like Functionals*

Let $\mathcal{M}$ be a Riemannian manifold, $d_1, d_2 \in \mathbb{N}$ be the dimensions of the image, let $h \in \mathcal{M}^{d_1 \times d_2}$ be our data. We are interested in solving the isotropic $(q = 2)$ and anisotropic $(q = 1)$ discrete ROF model

$$\inf_{p \in \mathcal{M}^{d_1 \times d_2}} \frac{1}{2\alpha} \sum_{i,j=1}^{d_1,d_2} d^2_{\mathcal{M}}\left(p_{i,j}, h_{i,j}\right) + \|T(p)\|_{p,q,1}, \tag{336}$$

where $\alpha > 0$ and $T : \mathcal{M}^{d_1 \times d_2} \to \mathcal{T}\mathcal{M}^{d_1 \times d_2 \times 2}$ is the non-linear finite difference operator defined as

$$(T(p))_{i,j,k} := \begin{cases} 0 \in \mathcal{T}_{p_{i,j}}\mathcal{M} & \text{if } i = d_1 \text{ and } k = 1 \\ 0 \in \mathcal{T}_{p_{i,j}}\mathcal{M} & \text{if } j = d_2 \text{ and } k = 2 \\ \log_{p_{i,j}} p_{i+1,j} \in \mathcal{T}_{p_{i,j}}\mathcal{M} & \text{if } i < d_1 \text{ and } k = 1 \\ \log_{p_{i,j}} p_{i,j+1} \in \mathcal{T}_{p_{i,j}}\mathcal{M} & \text{if } j < d_2 \text{ and } k = 2 \end{cases} \tag{337}$$

and where

$$\|T(p)\|_{p,q,1} ;= \sum_{i,j=1}^{d_1,d_2} (\|(T(p))_{i,j,1}\|^q_{p_{i,j}} + \|(T(p))_{i,j,2}\|^q_{p_{i,j}})^{\frac{1}{q}}. \tag{338}$$

Note that, whereas $T(p) \in \mathcal{T}_p\mathcal{M}^{d_1 \times d_2} \times \mathcal{T}_p\mathcal{M}^{d_1 \times d_2} = \mathcal{T}_{p,p}\mathcal{M}^{d_1 \times d_2 \times 2}$ we refer to p as the base-point of $T(p)$ instead of (p, p) and hence we write $\mathcal{T}_p\mathcal{M}^{d_1 \times d_2 \times 2}$ as well.

Using duality of the $\| \cdot \|_{p,q,1}$ norm and parallel transport being an isometry, we can choose $m \in \mathcal{M}$ and write

$$\|T(p)\|_{p,q,1} = \|P_{p \to m} T(p)\|_{m,q,1} = \sup_{\eta_m \in T^*_m \mathcal{M}^{d_1 \times d_2 \times 2}} \langle P_{p \to m} T(p), \eta_m \rangle - \iota_{B_{q^*}}(\eta_m), \tag{339}$$

where q^* such that $\frac{1}{q} + \frac{1}{q^*} = 1$,

$$\begin{aligned} B_{q^*} &:= \left\{ \nu_m \in \mathcal{T}^*_m \mathcal{M}^{d_1 \times d_2 \times 2} \mid \|\nu_m\|_{m,q^*,\infty} \leq 1 \right\} && (340) \\ &= \left\{ \nu_m \in \mathcal{T}^*_m \mathcal{M}^{d_1 \times d_2 \times 2} \mid \max \|(\nu_m)_{i,j,:}\|_{m,q^*} \leq 1 \right\}. && (341) \end{aligned}$$

and

$$\iota_{B_{q^*}}(\eta_m) := \begin{cases} 0 & \text{if } \eta_m \in B_{q^*} \\ \infty & \text{if } \eta_m \notin B_{q^*} \end{cases}. \tag{342}$$

Then, let $n := 0 \in \mathcal{T}_m \mathcal{M}^{d_1 \times d_2 \times 2}$ be the zero vector. We can write

$$\langle P_{p \to m} T(p), \eta_m \rangle = \langle P_{p \to m} T(p) - n, \eta_m \rangle. \tag{343}$$

Finally, remember that we can decompose the tangent bundle tangent space into a vector part η_m and a point part ζ_m. Let $\xi_n = (\zeta_m, \eta_m) \in \mathcal{T}^*_m \mathcal{M}^{d_1 \times d_2 \times 2} \times \mathcal{T}^*_m \mathcal{M}^{d_1 \times d_2 \times 2} \cong$

$\mathcal{T}_n^*\mathcal{T}\mathcal{M}^{d_1\times d_2\times 2}$, then

$$\langle P_{p\to m}T(p)-n,\eta_m\rangle = \sup_{\zeta_m\in T_m^*\mathcal{M}^{d_1\times d_2\times 2}} \langle(\log_m p, P_{p\to m}T(p)-n),(\zeta_m,\eta_m)\rangle - \iota_{\{0\}}(\zeta_m) \tag{344}$$

$$= \sup_{\zeta_m\in T_m^*\mathcal{M}^{d_1\times d_2\times 2}} \langle\log_n T(p),(\zeta_m,\eta_m)\rangle - \iota_{\{0\}}(\zeta_m), \tag{345}$$

where we used the definition of the logarithmic map on the tangent bundle in the last step.

Bringing everything together we find a the saddle-point problem in the form we are looking for

$$\inf_{p\in\mathcal{M}^{d_1\times d_2}} \sup_{\xi_n\in\mathcal{T}_n^*\mathcal{T}\mathcal{M}^{d_1\times d_2\times 2}} \frac{1}{2\alpha}\sum_{i,j=1}^{d_1,d_2} d^2_{\mathcal{M}}\left(p_{i,j},h_{i,j}\right) + \langle\log_n T(p),\xi_n\rangle_n - \iota_{\{0\}\times B_{q^*}}(\xi_n). \tag{346}$$

Remark 6.16. *Note that for $q=1$ (336) (and thus also the following result) reduces to minimizing the canonical (anisotropic) ℓ^2-TV (or ROF)on manifolds as in (2), i.e.,*

$$\inf_{p\in\mathcal{M}^{d_1\times d_2}} \frac{1}{2\alpha}\sum_{i,j=1}^{d_1,d_2} d^2_{\mathcal{M}}\left(p_{i,j},h_{i,j}\right) + \sum_{i,j=1}^{d_1-1,d_2} d_{\mathcal{M}}\left(p_{i,j},p_{i+1,j}\right) + \sum_{i,j=1}^{d_1,d_2-1} d_{\mathcal{M}}\left(p_{i,j},p_{i,j+1}\right). \tag{347}$$

Proximal Maps and Generalized Differentials

First, we need the proximal maps of F and G_n^* and the operators B and J for constructing the Newton operator as discussed in Sect. 6.4.

We will continue to use the notation

$$F(p) := \sum_{i,j=1}^{d_1,d_2} d_{\mathcal{M}}\left(p_{i,j},h_{i,j}\right)^2 \quad\text{and}\quad G^*_{n,q}(\xi_n) := \iota_{\{0\}\times B_{q^*}}(\xi_n). \tag{348}$$

Then we have for the data fidelity term [BLPS18, Prop. 4.4]

$$\operatorname{prox}_{\sigma F}(p) = \gamma_{p,h}\left(\frac{\sigma}{\alpha+\sigma}\right) \tag{349}$$

and we find

$$D_{C,p}\operatorname{prox}_{\sigma F}(p) = D_p\operatorname{prox}_{\sigma F}(\cdot) = D_p\gamma_{(\cdot),h}\left(\frac{\sigma}{\alpha+\sigma}\right), \tag{350}$$

where we used that $\operatorname{prox}_{\sigma F}(x)$ is smooth. Hence, we can choose $B = D_p\gamma_{(\cdot),h}\left(\frac{\sigma}{\alpha+\sigma}\right)$ for the Newton operator.

For the dual variable we live in a vector space. The proximal mapping comes down to the same as we have already computed in Sect. 3.4, but with a modification for the point part, i.e, for $\xi_n = (\xi_m^1,\xi_m^2)\in\mathcal{T}_m^*\mathcal{M}^{d_1\times d_2\times 2}\times\mathcal{T}_m^*\mathcal{M}^{d_1\times d_2\times 2}$

$$\operatorname{prox}_{\tau G^*_{n,1}}(\xi_n) = \left(0,\left(\max\left\{1,\|(\xi_m^2)_{i,j,k}\|_m\right\}\right)^{-1}(\xi_m^2)_{i,j,k}\right)_{i,j,k} \tag{351}$$

and

$$\operatorname{prox}_{\tau G^*_{n,2}}(\xi_n) = \left(0, \left(\max\left\{1, \|(\xi^2_m)_{i,j,:}\|_{m,2}\right\}\right)^{-1} (\xi^2_m)_{i,j,k}\right)_{i,j,k}. \tag{352}$$

Then, the generalized covariant derivative applied to $\eta_n = (\eta^1_m, \eta^2_m)$ is given by

$$(J_1(\xi_n)\eta_n)_{i,j,k} = \begin{cases} (0,0) & \text{if } i = d_1 \text{ and } k = 1 \\ (0,0) & \text{if } j = d_2 \text{ and } k = 2 \\ (0, (\eta^2_m)_{i,j,k}) & \text{if } \|(\xi^2_m)_{i,j,:}\|_m \leq 1 \\ \left(0, \frac{1}{\|(\xi^2_m)_{i,j,k}\|_m}\left((\eta^2_m)_{i,j,k} - \frac{\langle(\xi^2_m)_{i,j,k},(\eta^2_m)_{i,j,k}\rangle_m}{\|(\xi^2_m)_{i,j,k}\|^2_m}(\xi^2_m)_{i,j,k}\right)\right) & \text{if } \|(\xi^2_m)_{i,j,k}\|_m > 1 \end{cases} \tag{353}$$

and

$$(J_2(\xi_n)\eta_n)_{i,j,k} = \begin{cases} (0,0) & \text{if } i = d_1 \text{ and } k = \\ (0,0) & \text{if } j = d_2 \text{ and } k = \\ (0, (\eta^2_m)_{i,j,k}) & \text{if } \|(\xi^2_m)_{i,j,:}\|_{m,2} \leq \\ \left(0, \frac{1}{\|(\xi^2_m)_{i,j,:}\|_{m,2}}\left((\eta^2_m)_{i,j,k} - \sum_{\kappa=1,2} \frac{\langle(\xi^2_m)_{i,j,\kappa},(\eta^2_m)_{i,j,\kappa}\rangle_m}{\|(\xi^2_m)_{i,j,:}\|^2_{m,2}}(\xi^2_m)_{i,j,k}\right)\right) & \text{if } \|(\xi^2_m)_{i,j,:}\|_{m,2} > \end{cases} \tag{354}$$

where the operator J_1 corresponds to anisotropic TV and J_2 to isotropic. Here the first two conditions ensure the boundary conditions $y_{i,j,k} = 0$ for $i = d_1$ and $k = 1$ or $j = d_2$ and $k = 2$. The third and fourth options should be understood as the case of not being a boundary point, i.e., $i < d_1$ and $k = 1$ and $j < d_2$ and $k = 2$. A proof can be found in appendix A.1.

For the semismoothness we already have that $\operatorname{prox}_{\sigma F}(x)$ is semismooth since it is smooth: by the generalized Taylor series in [DPM03] we find semismoothness according to Def. 6.5. The semismoothness of $\operatorname{prox}_{\tau G^*_q}(y)$ follows from invoking Prop. 3.6 as in the non-manifold case. Indeed we can do this, since the dual variable just lives in a finite dimensional vector space, which is isomorphic with $\mathbb{R}^d$. So we are indeed justified to use RSSN for ℓ^2-TV.

Remark 6.17. *As we see from the formulation of the optimization problem to computing the dual proximal maps, the point part does not play an important role in the optimization problem: it is always zero. Therefore we can focus solely on the vector part in the following. To start of, we can ignore the point part for the proximal map in implementations and also drop them for computing J. This gives as an extra advantage that the Newton matrix becomes smaller.*

Linearization of the Operator T

Next, to derive $D_m T$ and its adjoint, let $p \in \mathcal{M}^{d_1 \times d_2}$ and $v \in \mathcal{T}_p\mathcal{M}^{d_1 \times d_2}$. We follow the approach in [BHTVN19, Sect. 5]. First, applying the chain rule we find

$$(D_p T[v])_{i,j,k} = D_{p_{i,j}} \log_{(\cdot)}(p_{i,j+e_k})[v_{i,j}] + D_{p_{i,j+e_k}} \log_{p_{i,j}}(\cdot)[v_{i,j+e_k}], \tag{355}$$

with the obvious modifications at the boundary. In the above formula e_k represents either the vector (0,1) or (1,0) used to reach either the neighbour to the right ($k = 1$) or below ($k = 2$).

Again, we use the decomposition of the tangent bundle tangent space. As noted in the previous remark, the point part does not bring new information. Using the result in (163), (355) reduces to

$$(D_pT[v])^v_{i,j,k} = \nabla_{v_{p_{i,j}}} \log_{(\cdot)}(p_{i,j+e_k}) + D_{p_{i,j+e_k}} \log_{p_{i,j}}(\cdot)\,[v_{i,j+e_k}] \in \mathcal{T}_{p_{i,j}}\mathcal{M}, \tag{356}$$

where the superscript v denotes the vector part of the tangent space. We used that that the second term only gave a contribution to the vector part of the tangent bundle tangent space in the first place. We can compute these maps using Jacobi fields as discussed in Prop. 4.39. With a slight abuse of notation we will drop the superscript v in the following since we know that we only use the vector part of the tangent space. Then, we have[8] $(D_pT) : \mathcal{T}_p\mathcal{M}^{d_1\times d_2} \to \mathcal{T}_p\mathcal{M}^{d_1\times d_2\times 2}(= \mathcal{T}_p\mathcal{M}^{d_1\times d_2} \times \mathcal{T}_p\mathcal{M}^{d_1\times d_2})$ given by Jacobi fields.

Subsequently, its adjoint can be computed using the adjoint Jacobi fields as given in (165). Following the same line of reasoning, note that we should also focus on the vector part of the dual space, i.e., $\mathcal{T}^*_p\mathcal{M}^{d_1\times d_2\times 2}$ instead of looking at the entire space $\mathcal{T}^*_{T(p)}\mathcal{T}\mathcal{M}^{d_1\times d_2\times 2}$. Define $N_{i,j}$ to be the set of neighbours of the pixel $p_{i,j}$ and let $\eta \in \mathcal{T}^*_{(p,p)}\mathcal{M}^{d_1\times d_2\times 2}$ then we can find [BHTVN19, Sect. 5]

$$(D_pT[\eta])^*_{i,j} = \sum_k \Big(\nabla \log_{(\cdot)}(p_{i,j+e_k})\Big)^* [\eta_{i,j,k}] \; + \sum_{(i',j')\in N_{i,j}} \Big(D_{p_{i,j}} \log_{p_{i',j'}}(\cdot)\Big)^* [\eta_{i'j'k}]\,. \tag{357}$$

Regularization of the Dual

In the case of $\mathbb{R}^d$ we saw in Sect. 3.4.2 that the Newton matrix became non-invertible. For general manifolds we are now motivated to look to the dual regularized ℓ^2-TV saddle-point problem

$$\inf_{p\in\mathcal{M}^{d_1\times d_2}} \sup_{\xi_n\in\mathcal{T}^*_n\mathcal{T}\mathcal{M}^{d_1\times d_2\times 2}} \frac{1}{2\alpha}\sum_{i,j=1}^{d_1,d_2} d^2_{\mathcal{M}}(p_{i,j},h_{i,j}) + \langle \log_n T(p),\xi_n\rangle_n - \iota_{\{0\}\times B_{q^*}}(\xi_n) - \frac{\beta}{2}\|\xi_n\|_n^2 \tag{358}$$

and hence consider $\tilde{G}^*_{n,1} = \iota_{\{0\}\times B_{q^*}}(\xi_n) + \frac{\beta}{2}\|\xi_n\|^2_n$. Then, as with the linear case it is not hard to see that the proximal maps become

$$\operatorname{prox}_{\tau\tilde{G}^*_{n,1}}(\xi_n) = \left(0, \left(\max\left\{1, \frac{\|(\xi^2_m)_{i,j,k}\|_m}{1+\beta\tau}\right\}\right)^{-1} \frac{(\xi^2_m)_{i,j,k}}{1+\beta\tau}\right)_{i,j,k} \tag{359}$$

and

$$\operatorname{prox}_{\tau\tilde{G}^*_{n,2}}(\xi_n) = \left(0, \left(\max\left\{1, \frac{\|(\xi^2_m)_{i,j,:}\|_{m,2}}{1+\beta\tau}\right\}\right)^{-1} \frac{(\xi^2_m)_{i,j,k}}{1+\beta\tau}\right)_{i,j,k} \tag{360}$$

[8] Here the decomposition originates from the two components represented by $k = 1$ and $k = 2$. This should not be confused with the point and vector part of the tangent space of the tangent bundle.

and as with the regular proxes we find for the covariant derivatives

$$(J_1(\xi_n)\eta_n)_{i,j,k} = \begin{cases} (0,0) & \text{if } i = d_1 \text{ and } k = 1 \\ (0,0) & \text{if } j = d_2 \text{ and } k = 2 \\ (0,(\eta_m^2)_{i,j,k}) & \text{if } \|(\xi_m^2)_{i,j,:}\|_m \leq 1 + \beta\tau \\ \left(0, \frac{1}{\|(\xi_m^2)_{i,j,k}\|_m} \left((\eta_m^2)_{i,j,k} - \frac{\langle(\xi_m^2)_{i,j,k},(\eta_m^2)_{i,j,k}\rangle_m}{\|(\xi_m^2)_{i,j,k}\|_m^2}(\xi_m^2)_{i,j,k}\right)\right) & \text{if } \|(\xi_m^2)_{i,j,k}\|_m > 1 + \beta\tau \end{cases} \tag{361}$$

and

$$(J_2(\xi_n)\eta_n)_{i,j,k} = \begin{cases} (0,0) & \text{if } i = d_1 \text{ and } k = \\ (0,0) & \text{if } j = d_2 \text{ and } k = \\ (0,(\eta_m^2)_{i,j,k}) & \text{if } \|(\xi_m^2)_{i,j,:}\|_{m,2} \leq \\ \left(0, \frac{1}{\|(\xi_m^2)_{i,j,:}\|_{m,2}} \left((\eta_m^2)_{i,j,k} - \sum_{\kappa=1,2} \frac{\langle(\xi_m^2)_{i,j,\kappa},(\eta_m^2)_{i,j,\kappa}\rangle_m}{\|(\xi_m^2)_{i,j,:}\|_{m,2}^2}(\xi_m^2)_{i,j,k}\right)\right) & \text{if } \|(\xi_m^2)_{i,j,:}\|_{m,2} > \end{cases} \tag{362}$$

Remark 6.18. *For a general manifold we did not show that we run into ill-posedness issues. However, from numerical observations it seems to be the case for 2D problems (as in the $\mathbb{R}^d$ case). Actually showing this remains an open problem.*

6.6 Numerical Experiments*

In this section we will explore the behaviour of the Riemannian Semismooth Newton method through several numerical experiments with Total Variation on the S^2 and $\mathcal{P}(3)$ manifold. In chapter 3 we have already seen how (R)SSN works for the flat manifold $\mathbb{R}$. In this part S^2 and $\mathcal{P}(3)$ are chosen so we can see the behaviour on positively respectively negatively curved spaces.

The key questions we try to answer are

- Does RSSN for the exact or linearized system work?
- Can we get quantitatively better performance using RSSN than when using RCPA?
- Does inexact RSSN behave as predicted in Thm. 6.12?

We will try to answer these questions through three experiments: a proof of concept for a 1D problem with known minimizer, runtime analysis, and a proof of concept for IRSSN.

Remember that in the following sections our goal is to solve the exact and the linearized optimality system. We will refer to RSSN for the exact system as eRSSN and for the linearized system as lRSSN. Both algorithms and additionally eRCPA and lRCPA have been implemented using Manopt.jl [Ber19].

Throughout the sections we will use the relative error

$$\epsilon_{rel}^k := \frac{\|X(p^k,\xi_n^k)\|_{(p^k,\xi_n^k)}}{\|X(p^k,\xi_n^0)\|_{(p^0,\xi_n^0)}} \tag{363}$$

as measure for convergence.

Again, all numerical experiments are implemented in Julia version 1.3.0 and run on a HP ZBook, 2.4 GHz Intel Core i7, 8 GB RAM.

6.6.1 Signal with Known Minimizers

In this first experiment we want to investigate whether either the eRSSN or lRSSN work by investigating the progression of the relative errors and the distances to the exact solution to a problem with known minimizer. For a proof of concept we will consider a 1-dimensional piecewise constant signal

$$h \in \mathcal{M}^{2\ell} \quad h_i := \begin{cases} \hat{p}_1 & \text{if } i \leq \ell \\ \hat{p}_2 & \text{if } i > \ell \end{cases} . \tag{364}$$

For this signal we know the exact ℓ^2-TV minimizer in (336): for $\alpha > 0$ and $\hat{p}_1, \hat{p}_2 \in \mathcal{M}$ the minimizer p^* is given by

$$p_i^* := \begin{cases} p_1^* & \text{if } i \leq \ell \\ p_2^* & \text{if } i > \ell \end{cases} , \tag{365}$$

where

$$p_1^* = \gamma_{\hat{p}_1,\hat{p}_2}(\delta) \quad \text{and} \quad p_2^* = \gamma_{\hat{p}_2,\hat{p}_2}(\delta) \quad \text{where} \quad \delta = \min\left\{\frac{1}{2}, \frac{\alpha}{\ell}\frac{1}{d_{\mathcal{M}}(\hat{p}_1,\hat{p}_2)}\right\}. \tag{366}$$

A proof can be found in appendix A.2.

Furthermore, note that $d_2 = 1$ and the operator T reduces to the 1-dimensional nonlinear difference operator (i.e., we have $T : \mathcal{M}^{2\ell} \to \mathcal{TM}^{2\ell}$). This also means that the isotropic ($q = 2$) and anisotropic ($q = 1$) cases reduce to the same functional.

Further we note that as with $\mathcal{M} = \mathbb{R}$, we do not know for certain that the Newton matrix is invertible. Again, empirically this seems to be the case, as we had no issues solving the Newton system for $\beta = 0$.

For the following two cases we use $\ell = 10$ $\alpha = 5$ and $\sigma = \tau = \frac{1}{2}$. We use a tolerance of $\epsilon_{rel} = 10^{-10}$ and perform a maximum of 50 iterations for the S^2 problem and 15 for the $\mathcal{P}(3)$ problem. Furthermore, for a given primal base point $m \in \mathcal{M}$ we will choose $n = T(m) \in \mathcal{TM}^{2\ell}$ for the dual base point. As we saw we need n to be the zero vector. By choosing m_i the same in every grid point, this condition is satisfied.

In the following we will refer to p_e as the solution obtained from eRSSN and p_l as the solution from lRSSN. The results are summarized with respect to the distance to the exact solution in Tab. 5.

Case 1: $\mathcal{M} = S^2$
We choose

$$\hat{p}_1 := \frac{1}{\sqrt{2}}(1,1,0)^\top, \quad \hat{p}_2 := \frac{1}{\sqrt{2}}(1,-1,0)^\top \tag{367}$$

and

$$m_i := (1,0,0)^\top \quad \text{for all } i = 1,...,2\ell \tag{368}$$

	Cold start		Warm start	
$\mathcal{M}$	$d_{\mathcal{M}}(p^*, p_e)$	$d_{\mathcal{M}}(p^*, p_l)$	$d_{\mathcal{M}}(p^*, p_e)$	$d_{\mathcal{M}}(p^*, p_l)$
S^2	0.018284421	3.2953753	0.9582239	**0.0**
$\mathcal{P}(3)$	2.4091654	2.1286297e-13	1.643346	**1.8381813e-13**

Table 5: The distances between the exact minimizers of the 1D piecewise constant signal to the results of eRSSN and lRSSN on the S^2 and the $\mathcal{P}(3)$ manifold. In the case of a warm start, i.e., a good prior estimate for both primal and dual variable, the lRSSN solution is very close to the ℓ^2-TV minimizer. For eRSSN the solution does not correspond to the ℓ^2-TV minimizer.

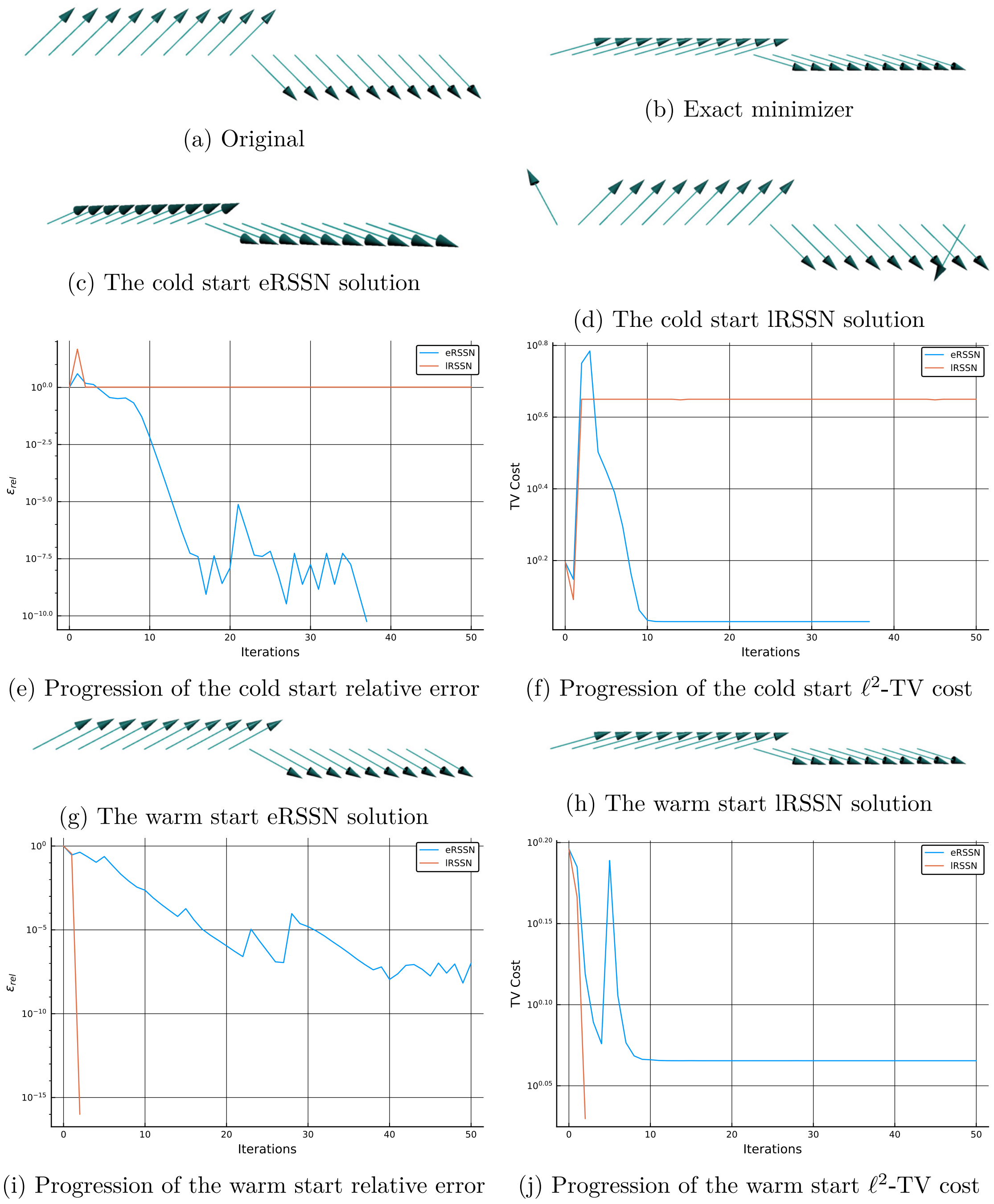
and find that n is the zero tangent vector with base m.

We run the experiment for two starting points. In both the cases the primal variable starts from the data, i.e., $p^0 = h$. For the dual we start from the zero vector in the first case, i.e., $\xi_n^0 = 0$ (cold start) and for the second case we start from the dual as the result of one RCPA step (warm start). That is eRSSN starts with the dual result after one eRCPA step and lRSSN starts with the dual vector after one lRCPA step. The reason to include the warm start as well comes from the theory of RSSN. Since RSSN only converges locally (superlinearly), a warm start might be necessary. The results are shown in Fig. 19.

Cold start eRSSN converges in 3.547 seconds to a solution, whereas lRSSN gets stuck in a local minimum and is terminated after 50 iterations (after 4.234 seconds). Although eRSSN seems to have superior performance here, the solution is not equal to the exact minimizer. We even have a distance of $d_{\mathcal{M}}(p^*, p_e) = 0.01828$ away from the exact solution (see Tab. 5).

In both cases we solve approximate optimality systems and until now it is not yet known how good the approximations are. This first experiment indicates that the solution to the exact optimality system does not give the ℓ^2-TV minimizer, because even though eRSSN found a very accurate solution, it does not correspond the correct minimizer.

Warm start Now eRSSN performs worse than without a warm start. The algorithm is terminated after 50 iterations (at 3.251 seconds). A possible explanation could be that rounding errors prohibit us from getting a better solution. For the cold start we also observed similar behaviour. For lRSSN we see right away that a warm start gives better performance. In 0.469 seconds (only two iterations) we converge with 0 error to the solution of the linearized system. As it turns out this is the same point as the ROF minimizer, i.e., $d_{\mathcal{M}}(p^*, p_l) = 0$ as well (again see Tab. 5).



(a) Original

(b) Exact minimizer

(c) The cold start eRSSN solution

(d) The cold start lRSSN solution

(e) Progression of the cold start relative error

(f) Progression of the cold start ℓ^2-TV cost

(g) The warm start eRSSN solution

(h) The warm start lRSSN solution

(i) Progression of the warm start relative error

(j) Progression of the warm start ℓ^2-TV cost

Figure 19: The results along with the progression of the relative error and the ℓ^2-TV cost for eRSSN and lRSSN applied to a S^2 problem with known minimizer. With a cold start eRSSN converges, but does not converge to the exact ℓ^2-TV minimizer. With a warm start eRSSN seems to suffer from rounding errors and does not converge to the exact ℓ^2-TV minimizer. lRSSN gets stuck in a local minimum in the case of a cold start, but for a warm start the algorithm converges superlinearly to the exact minimizer.

Case 2: $\mathcal{M} = \mathcal{P}(3)$
Next, we choose

$$\hat{p}_1 := \exp_I\left(\frac{2}{\|X\|_I}X\right), \quad \hat{p}_2 := \exp_I\left(-\frac{2}{\|X\|_I}X\right), \quad \text{with} \quad X := \begin{pmatrix} 1 & 2 & 2 \\ 2 & 2 & 0 \\ 2 & 0 & 6 \end{pmatrix}, \tag{369}$$

where $X \in \mathcal{T}_I\mathcal{P}(3)$ and I is the identity matrix and pick

$$m_i := I \quad \text{for all } i = 1, ..., 2\ell \tag{370}$$

and once again find that n is the zero tangent vector, i.e., the zero matrix, at base m.

We will distinguish between a warm start and a cold start again. The results are shown in Fig. 20.

Cold start Now, lRSSN is the better method. It converges superlinearly in 4.61 seconds. eRSSN on the other hand seems to get a worse result after iteration 10. It is terminated after 76.531 seconds. However, whereas the results look very similar, eRSSN end up a distance $d_{\mathcal{M}}(p^*, p_e) = 2.4091654$ away from the exact solution, while $d_{\mathcal{M}}(p^*, p_l) = 2.13 \cdot 10^{-13}$ for lRSSN.

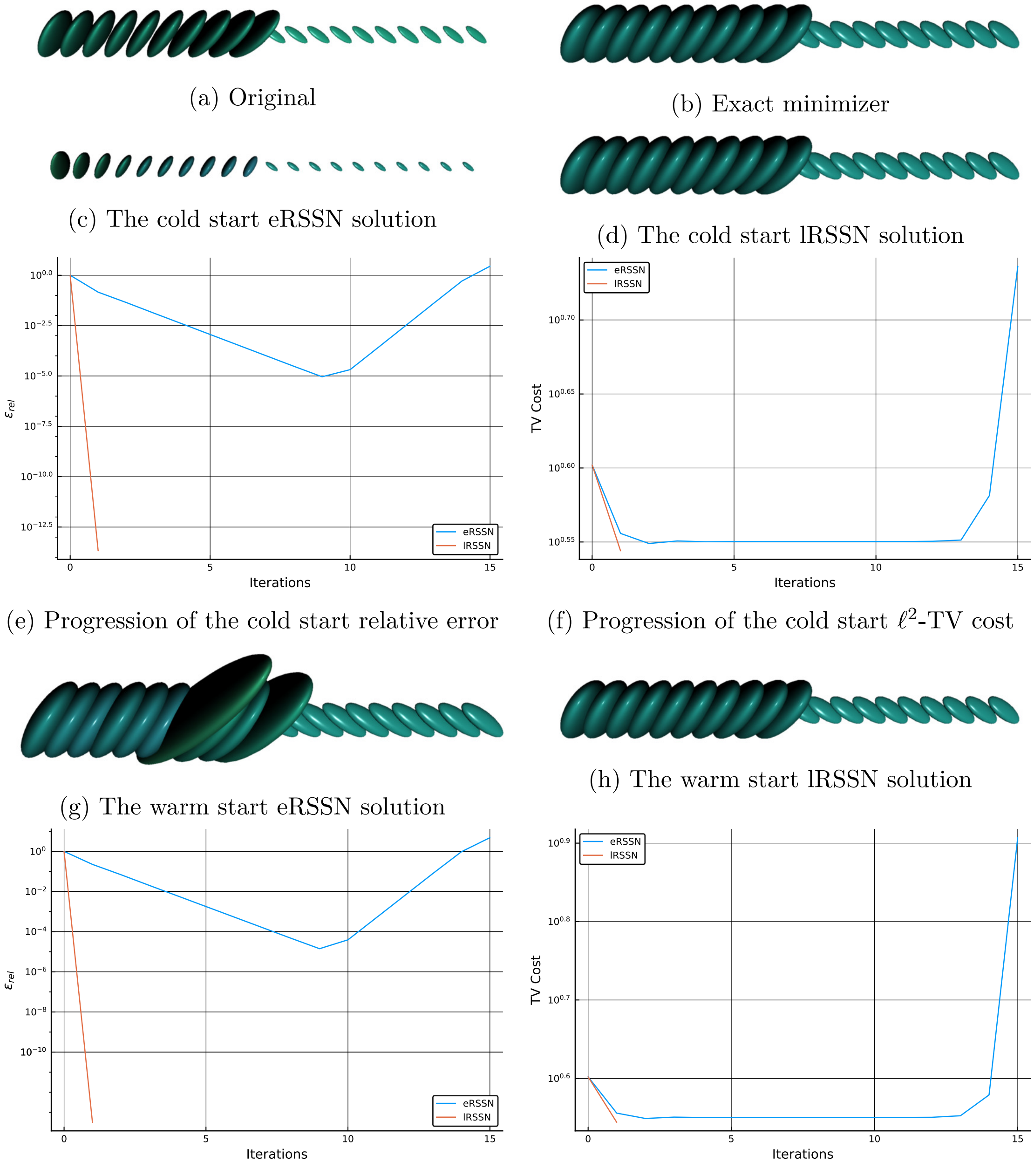
Warm start For the warm start we observe that eRSSN performs slightly better than before, but in the end once again diverges. The method was terminated after 73.625 seconds at a distance $d_{\mathcal{M}}(p^*, p_e) = 1.643346$ away from the exact solution. lRSSN on the other hand converges after 1 iteration in 4.202 seconds and ends up at distance $d_{\mathcal{M}}(p^*, p_l) = 1.84 \cdot 10^{-13}$ away from the exact solution.

Observations
Whereas lRSSN results are superb, eRSSN performs much worse: in the case of the S^2 manifold we do not seem to converge to the minimizer of ℓ^2-TV, but more concerning is the behaviour for the $\mathcal{P}(3)$ manifold. The method diverges after some point.

A first explanation would lie in the nature of the problem we are solving. We foresaw that every RSSN-based method could run into trouble for negatively curved manifolds (i.e., a large $K_{(p^*,\xi_n^*)}$). However, since $K_{(p^*,\xi_n^*)}$ is independent of the method[9], the discrepancy between eRSSN and lRSSN might also be caused by something else. As we discussed in Sect. 6.3.3, it is also possible to have a very ill-conditioned matrix at the optimum (i.e., a large $\lambda_{(p^*,\xi_n^*)}$). At this point we are not close enough to the solution to make such a claim.

Based on the much better performance of the lRSSN method, we restrict ourselves to the lRSSN methods in the following sections.

[9] Indeed for Hadamard manifolds it is solely dependent on the manifold.



(a) Original

(b) Exact minimizer

(c) The cold start eRSSN solution

(d) The cold start lRSSN solution

(e) Progression of the cold start relative error

(f) Progression of the cold start ℓ^2-TV cost

(g) The warm start eRSSN solution

(h) The warm start lRSSN solution

(i) Progression of the warm start relative error

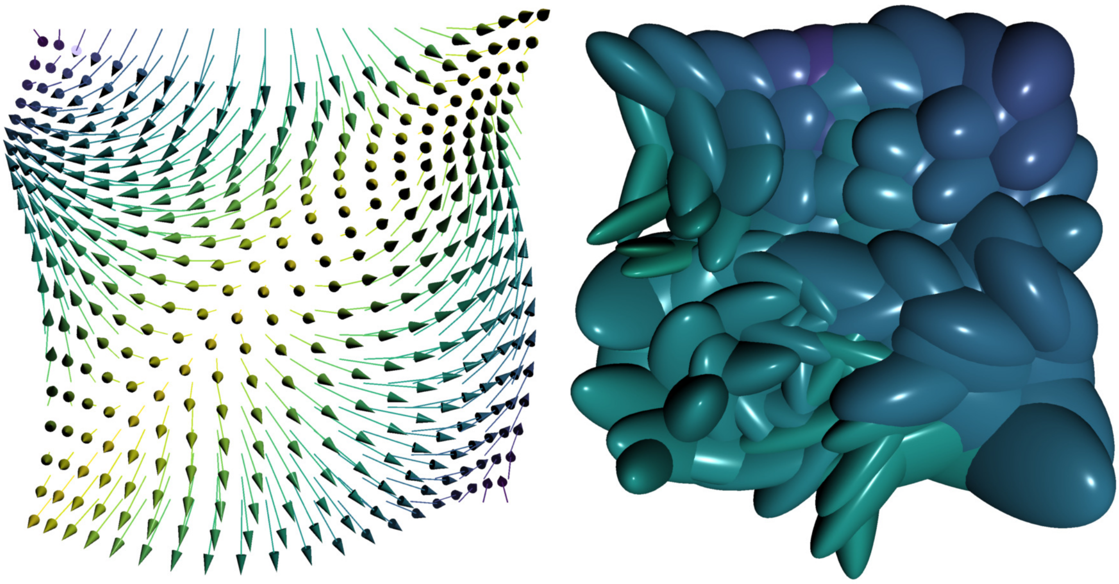
(j) Progression of the warm start ℓ^2-TV cost

Figure 20: The results along with the progression of the relative error and the ℓ^2-TV cost for eRSSN and lRSSN applied to a $\mathcal{P}(3)$ problem with known minimizer. With or without a warm start, eRSSN diverges and the ℓ^2-TV energy amplifies. lRSSN converges superlinearly to the exact minimizer for both cold and warm start.

6.6.2 Comparison of Algorithms for Solving Regularized TV

For this experiment we investigate the runtime performance for reaching different accuracies with the lRSSN method and compare it to the lRCPA algorithm. In this experiment we will focus on 2D problems. From numerical observation we saw that the matrices for both S^2 and $\mathcal{P}(3)$ became singular if we did not use dual regularization. So in the following we will resort to solving regularized ℓ^2-TV with $\beta > 0$. In particular, we will use $\beta = 10^{-6}$ motivated by the numerical experiments in chapter 3.

For the S^2 problem of this experiment we use a 20×20 artificial S^2 rotations image from Manopt.jl with a 0.5 rotation around each axis. For the $\mathcal{P}(3)$ problem we will use a 10×10 artificial $\mathcal{P}(3)$ image also from Manopt.jl. The original images are shown in Fig. 21.



(a) Original S^2 data

(b) Original $\mathcal{P}(3)$ data

Figure 21: Original 2D manifold-valued S^2 (left) and $\mathcal{P}(3)$ (right) images.

For this part we are interested in getting insight into the runtime performance of lRSSN. We compare performance of lRSSN with lRCPA. Although lRCPA is not guaranteed to converge on positively curved manifolds, it performed well in recent work [BHTVN19]. We expect that lRCPA will be faster at the start but will suffer from slow tail convergence. Hence, lRSSN should give better performance for higher accuracy solutions.

Initialization

For our numerical experiment, we measure the (CPU) runtime until the algorithms reach $\epsilon_{rel} \in \{10^{-2}, 10^{-4}, 10^{-6}\}$. lRSSN diverges if we use a cold start and even a warm dual start is not sufficient. Therefore, we run lRCPA until $\epsilon_{rel} = 1/2$ and use the resulting primal and dual iterates as p^0 and ξ_n^0 as starting points for the experiment. The relative errors mentioned before include the initial error loss due to the pre-steps with lRCPA.

Case 1: $\mathcal{M} = S^2$

In the following we will compute regularized isotropic ℓ^2-TV solution on this data with $\alpha = 1.5$. For m we will choose

$$m_{i,j} := (0,0,1)^\top, \quad \text{for } i,j = 1,\ldots,20, \tag{371}$$

so that we have n as the zero vector.

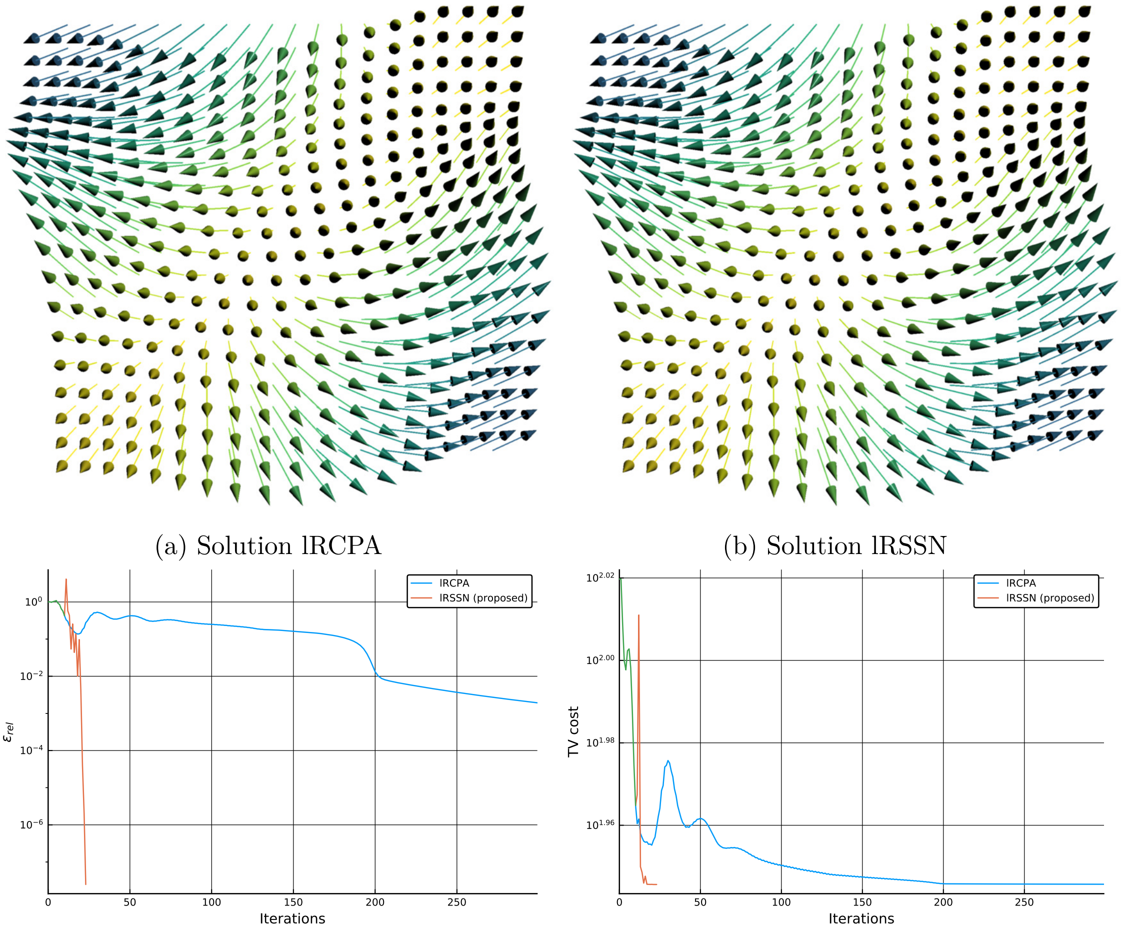
We initialized both lRSSN and lRCPA with $\sigma = \tau = 0.35$. Furthermore, for lRCPA we used $\gamma = 0.2$ to controlled the primal and dual step size.

The pre-steps took 2.437 seconds. The resulting runtimes are shown in Tab. 6. The solutions of lRCPA and lRSSN at $\epsilon_{rel} = 10^{-6}$ along with the development of the relative error and the (isotropic) ℓ^2-TV-cost are shown in Fig. 22.

$\beta = 10^{-6}$	$\epsilon_{rel} = 10^{-2}$		$\epsilon_{rel} = 10^{-4}$		$\epsilon_{rel} = 10^{-6}$	
Method	Time	# Iterations	Time	# Iterations	Time	# Iterations
lRCPA	**46.515**	193	**159.11**	697	608.781	2886
lRSSN	330.422	10	346.187	11	**410.875**	13

Table 6: The runtimes and number of iterations of iterations for lRCPA and lRSSN to converge to the three accuracies for the S^2 problem. For high-accuracy solutions the proposed lRSSN algorithm outperforms lRCPA with respect to runtime on a manifold with positive curvature.

The results confirm what we expect: for accuracies larger accuracies, errors smaller than 10^{-4}, lRSSN beats lRCPA. Furthermore, we see the superlinear convergence more clearly than in the 1D case.



(a) Solution lRCPA

(b) Solution lRSSN

(c) Progression of the relative error

(d) Progression of the isotropic ℓ^2-TV cost

Figure 22: The results of lRCPA and lRSSN for an artificial S^2 problem and the progression of the relative errors and the ℓ^2-TV costs. The initial error and cost drop due to the pre-steps is shown in green. The proposed lRSSN method converges within 13 iterations after the pre-steps superlinearly to an optimal solution on a manifold with positive curvature, while lRCPA suffers from slow tail convergence.

Case 2: $\mathcal{M} = \mathcal{P}(3)$
Next, we will compute regularized isotropic ℓ^2-TV solution on this data with $\alpha = 0.5$. For m we will choose

$$m_{i,j} := I, \qquad \text{for } i, j = 1, \ldots, 10, \tag{372}$$

so that we have n as the zero vector.

We initialize both lRSSN and lRCPA with $\sigma = \tau = 0.4$. Furthermore, for lRCPA we use $\gamma = 0.2$ to control the primal and dual step size.

The pre-steps took 9.718 seconds. The resulting runtimes are shown in Tab. 7. The solutions of lRCPA and lRSSN at $\epsilon_{rel} = 10^{-6}$ along with the development of the relative error and the (isotropic) ℓ^2-TV cost are shown in Fig. 23. We note that lRCPA stalled before reaching the relative error of 10^{-6} and was terminated after 2500 iterations.

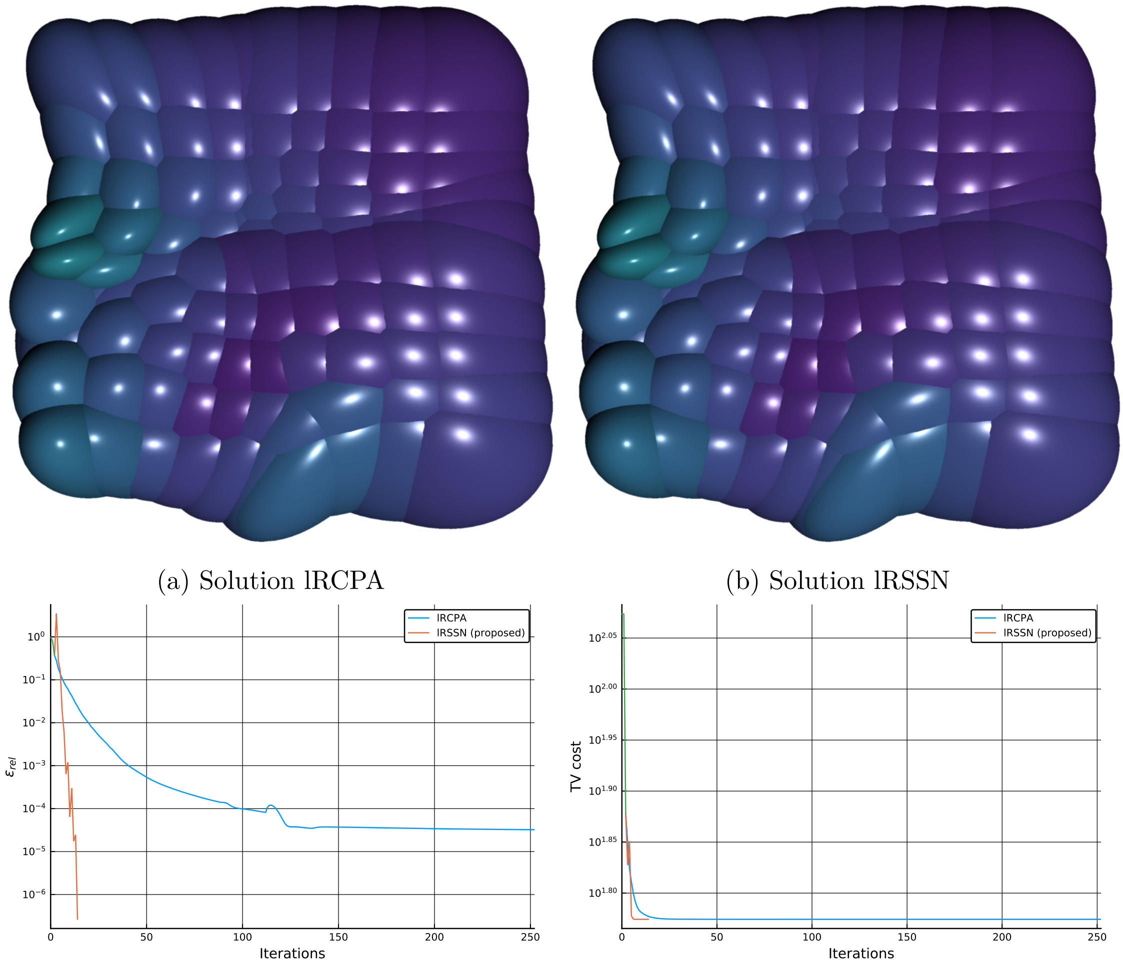
$\beta = 10^{-6}$	$\epsilon_{rel} = 10^{-2}$		$\epsilon_{rel} = 10^{-4}$		$\epsilon_{rel} = 10^{-6}$	
Method	Time	# Iterations	Time	# Iterations	Time	# Iterations
lRCPA	**9.922**	18	**48.36**	97	1213.327	≥2500
lRSSN	237.062	5	380.047	8	**556.328**	12

Table 7: The runtimes and number of iterations of iterations for lRCPA and lRSSN to converge to the three accuracies for the $\mathcal{P}(3)$ problem. For high-accuracy solutions the proposed lRSSN algorithm outperforms lRCPA with respect to runtime on a manifold with negative curvature. lRCPA was terminated after 2500 iterations before being able to reach $\epsilon_{rel} = 10^{-6}$.

As for th S^2 example previously, the results confirm what we expect. For higher accuracies, with errors smaller than 10^{-4}, lRSSN beats lRCPA. The latter even seems unable to reach such high accuracies in reasonable time. We observe superlinear convergence for lRSSN with this example as well.

Observations
Even though lRSSN performs very well, one might wonder whether our examples represent real-life situations. Instead of using the relative error as convergence criterion, we could also use the change in ℓ^2-TV cost. As we see in both figures for the ℓ^2-TV cost, lRCPA reaches a terminal cost rather quickly as well. One might wonder whether the same results would yield when passing to a different (more practical) convergence criterion based on changes in the cost functional.

For our purposes, we were interested in a proof of concept for lRSSN and showing the superlinear convergence with respect to the relative error measure. So this question will be left for future research.



(a) Solution lRCPA

(b) Solution lRSSN

(c) Progression of the relative error

(d) Progression of the isotropic ℓ^2-TV cost

Figure 23: The results of lRCPA and lRSSN for an artificial $\mathcal{P}(3)$ problem and the progression of the relative errors and the ℓ^2-TV costs. The initial error and cost drop due to the pre-steps is shown in green. The proposed lRSSN method converges within 12 iterations after the pre-steps superlinearly to an optimal solution on a manifold with negative curvature, while lRCPA suffers from slow tail convergence.

6.6.3 An Outlook to Inexact Semismooth Newton

In this final experiment we will try to validate our predictions for inexact Riemannian Semismooth Newton. In particular, we want to verify linear convergence for a^k = constant and superlinear convergence for $a^k \to 0$ as $k \to \infty$. We will also look into an application: denoising with ℓ^2-TV. In the following we will restrict ourselves to the inexact variant of lRSSN. Given the similar behaviour of the methods on positively and negatively curved manifolds in previous experiments, we will only consider an S^2 example for this experiment.

For our problem we will choose denoising Bernoulli's Lemniscate, which is a figure 8 on the 2-sphere. Note that this is once again a 1D problem. We will consider 128 S^2-valued points on the Lemniscate curve and distort them with Gaussian noise with variance $\delta^2 = 0.01$, in the sense that for each point p we will pick a tangent vector v_p drawn from Gaussian distribution over the tangent space[10], and our data h will be such that

$$h_i := \exp_{p_i}(v_{p_i}). \tag{373}$$

We use ℓ^2-TV with $\alpha = 0.5$ and m as the intersection point of the Lemniscate:

$$m_i := \frac{1}{\sqrt{2}}(1,0,1)^\top, \qquad \text{for } i = 1,\dots,128. \tag{374}$$

For this case n will be the zero vector with base m. We will not use dual regularization, i.e., $\beta = 0$.

We need a warm start by lRCPA and use $\sigma = \tau = 0.35$ and $\gamma = 0.2$. We take pre-steps until $\epsilon_{rel} = 10^{-1}$ and continue with IRSSN after that. Then, we will run three experiments with

$$a_1^k := 0, \qquad a_2^k := \frac{1}{5}, \qquad a_3^k := \frac{1}{5k}, \tag{375}$$

and we define a residual

$$r_i^k := a_i^k \|X(p^k,\xi_n^k)\|_{(p^k,\xi_n^k)} U_{(p^k,\xi_n^k)}, \qquad \text{for } i = 1,2,3, \tag{376}$$

where $U_{(p^k,\xi_n^k)}$ is a tangent vector drawn from a normal distribution with unit covariance matrix. Subsequently, after the pre-steps we solve d^k through solving

$$V_{(p^k,\xi_n^k)} d^k = X(p^k,\xi_n^k) + r_i^k. \tag{377}$$

So in other words, by simulating residual and solving the problem exactly with lRSSN we simulate the behaviour of IRSSN.[11]

[10] Note that we can use our classical Gaussian distribution here, because the tangent space is a linear space.

[11] Due to the lack of a proper preconditioner, we needed to resort to this way of testing instead of using an iterative method.

The convergence rate q will be approximated by

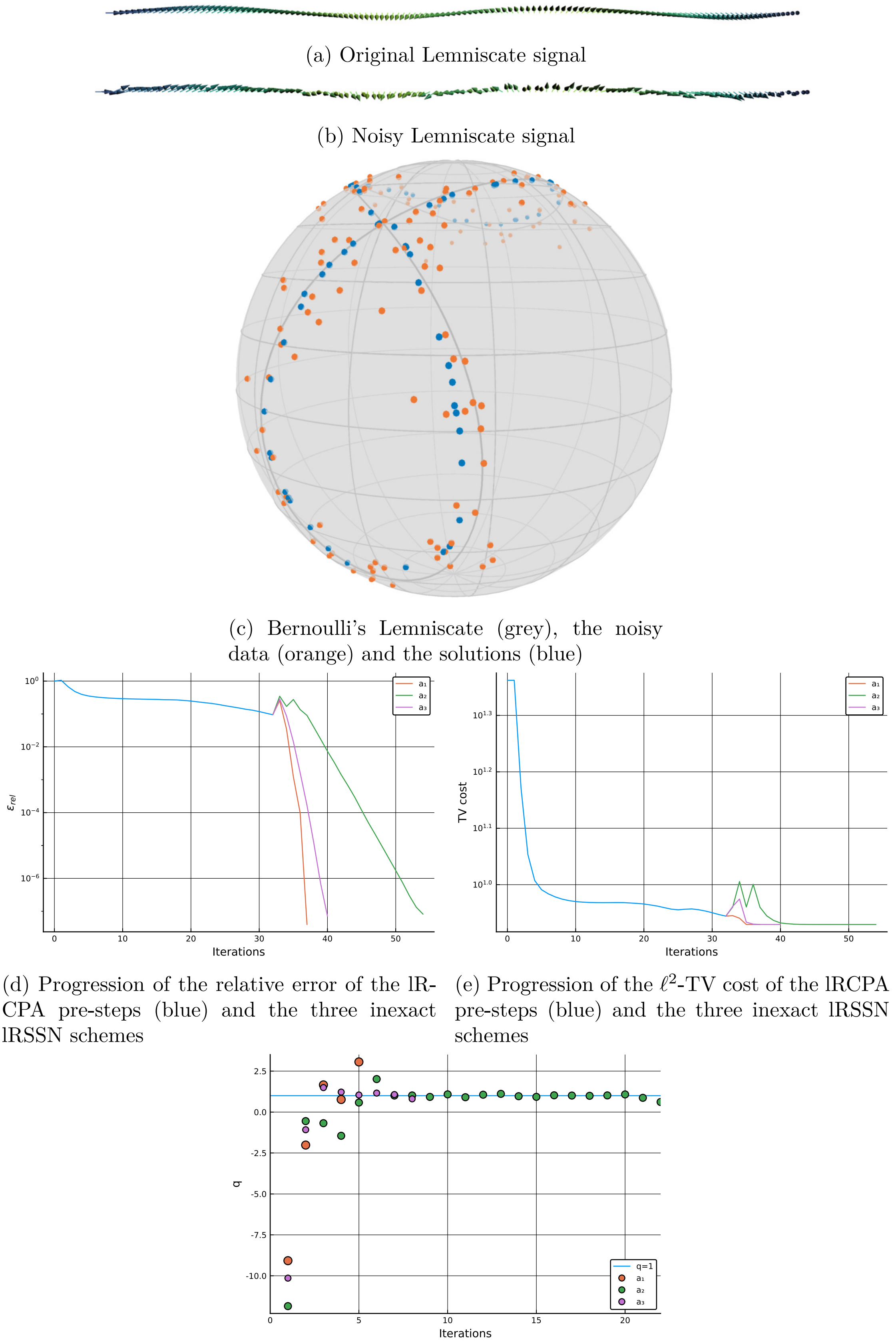
$$q^k := \frac{\log\left(\frac{\|X(p^k,\xi_n^k)\|}{\|X(p^{k-1},\xi_n^{k-1})\|}\right)}{\log\left(\frac{\|X(p^{k-1},\xi_n^{k-1})\|}{\|X(p^{k-2},\xi_n^{k-2})\|}\right)}. \tag{378}$$

The results are shown in Fig. 24. As expected we observe that the scheme of a_2^k converges linearly, i.e., with rate $q = 1$. For a_3^k we observe the relative error progression closely follows that of normal lRSSN, i.e., using a_1^k, and we observe that q is mainly slightly larger than 1, which indicates superlinear convergence. This is also in line with our expectations.

6.6.4 Final Remarks

With these numerical experiments we set out to confirm the theoretical findings and show a proof of concept for the developed algorithms. In particular, we wanted to show local superlinear convergence on positively and negatively curved manifolds for eRSSN and lRSSN and establish (at least) local linear convergence for an inexact version of either one of the algorithms.

Especially lRSSN has shown to be very promising. For 1D signals and 2D images with S^2 and $\mathcal{P}(3)$ data, superlinear convergence was obtained, when initialized at a proper starting point. In particular, the convergence behaviour of lRSSN can be used to overcome the slow tail convergence, which is holding back several first-order methods such as lRCPA. Additionally, lRSSN combined with an inexact scheme for solving the Newton matrix has the potential to result in a very successful algorithm for efficiently solving larger-scale problems.



(a) Original Lemniscate signal

(b) Noisy Lemniscate signal

(c) Bernoulli's Lemniscate (grey), the noisy data (orange) and the solutions (blue)

(d) Progression of the relative error of the lRCPA pre-steps (blue) and the three inexact lRSSN schemes

(e) Progression of the ℓ^2-TV cost of the lRCPA pre-steps (blue) and the three inexact lRSSN schemes

(f) Progression of the convergence rates after the pre-steps

Figure 24: The results of inexact lRSSN for denoising Bernoulli's Lemniscate for different degrees of inexactness denoted by a_1, a_2 and a_3, where a_1 corresponds to normal lRSSN. Only the a_2 solution is shown in (c). The a_1 and a_3 solutions are indistinguishable from a_2. After the prep-steps a_1 and a_3 converge superlinearly and a_2 linearly.

Chapter 7: Conclusions

In this work we have presented the exact and linearized Riemannian Semismooth Newton method (eRSSN and lRSSN) as higher-order optimization methods for solving nonsmooth variational problems. By transferring best practices from the linear case to the manifold case we were able to apply both algorithms to solving isotropic and anisotropic ℓ^2-TV problems for manifolds of positive and negative curvature. In particular, we used the dual regularization approach that solved ill-posedness in the linear case as a strategy for handling manifold-valued images.

From numerical experiments with 1D signals and 2D images we see promising results for lRSSN: in particular, we obtain superlinear convergence towards the ℓ^2-TV minimizer and establish state-of-the-art runtimes for high-accuracy solutions on the S^2 and the $\mathcal{P}(3)$ manifold.

The numerical results of eRSSN also provide novel insights: due to the high accuracies we can reach with eRSSN, we find hints that the exact optimality system derived from the ℓ^2-TV functional has a different solution than ℓ^2-TV itself. Whether the two optimizers were actually equal was still an open question from previous work [BHTVN19].

Moreover, the first step towards making general RSSN-based methods feasible for solving large scale manifold valued imaging problems has been set by proving a local convergence result for inexact RSSN. This result is also backed-up by numerical experiments with inexact lRSSN, in which we observed, as expected, local linear convergence.

Suggestions for Future Research
This work paved the way for several interesting follow up projects:

The Ill-posedness of the lRSSN Matrix for ℓ^2-TV-like problems For the case of solving ℓ^2-TV, in the $\mathbb{R}^d$ case we could show that the ill-posedness of the Newton matrix was caused by cycles of pixels with the same value at the optimum. In the manifold case the obtained matrix has a less transparent structure than before. Numerical experiments suggest that the matrix can become non-invertible without dual regularization. A full theoretical analysis is left for future work.

The role of σ and τ So far our choice for σ and τ was made on the basis of whether these values would work for RCPA. A proper exploration of the precise effects could give us new insights into the properties of the Newton matrix. As a starting point, we would suggest to focus on the $\mathbb{R}^d$ case, for there is still a lot unclear as well.

lRSSN for Large Scale Problems Another suggestion would be to look for large scale applications. We experimented with GMRES approach to test our inexact lRSSN, but without a good preconditioner this attempt was pointless. Especially because in realistic cases we choose a small β for the dual regularization, resulting in a large condition number. It is well known that the condition number directly affects the convergence speed for the worse. We would suggest to look into preconditioners for the Newton matrix as the first step towards solving large scale problems.

lRSSN Variations We used the exact and linearized optimality system solving the ℓ^2-TV problem. For this we needed to choose a base point $m \in \mathcal{M}$ and $n \in \mathcal{N}$. In this thesis, these were fixed. In [BHTVN19] it was suggested to look into changing these per iteration when using lRCPA.

In this work we claimed that we should choose a particular n in the case of a Total Variation regularizer, i.e., n being a zero vector with base point m. Controlling m would still be an interesting approach and could yield better results: we could make less of an error when linearizing. This would not only be interesting for lRCPA, but also for lRSSN.

lRSSN for Other Models So far, only TV has been used as an application for lRSSN (as with lRCPA). Total Generalized Variation also allows for the dual representation and could be an interesting next candidate to apply lRSSN on.

Expanding the Theory of Inexact RSSN Thm. 6.12 in the linear case could be formulated much stronger than for manifolds. Showing the converses of (ii) and (iii) in our theorem would provide more even more insight into the limitations of IRSSN.

lRSSN Globalization Finally, a globalization strategy of RSSN would be an extremely valuable result. Recently, a scheme was proposed for SSN that is an outstanding candidate to be made compatible with our types of problem: Adaptive Semismooth Newton (ASSN).

If we want to extend this scheme to manifolds, there are some obstacles. The method relies on monotonicity of the vector field. Whereas there exist generalizations of classical monotonicity to vector fields, we can only do this for spaces with negative curvature. To be more complete, we would like the space to be a Hadamard manifold.

Given our results for the $\mathcal{P}(3)$ manifold, globalization might not be far away and would definitely be a worthwhile topic to look into.

Chapter A: Appendix

A.1 Covariant Derivatives for the ℓ^2-TV-like Dual Proximal Maps

In the following we will focus on the vector part ξ_m of the dual variable $\xi_n = (\zeta_m, \xi_m)$ for the proximal map, since the point part will give zero anyways.

Now, as for the linear case we see that the components of (the vector part of) $\operatorname{prox}_{\tau G^*_{n,q}}(\eta_n)$, i.e., $\|(\eta_m)_{i,j,:}\|_{m,q^*}$ smaller or larger than 1, are piecewise C^1 in the $\mathcal{T}^*_m\mathcal{M}^{d_1\times d_2\times 2}$ vector space. So by considering the two regions seperately we can use ordinary (covariant) derivative on each region and choose the value corresponding to the inner region for the generalized (covariant) derivative on the boundary. We will compute the (covariant) derivative for the isotropic case and als give the result of the anisotropic case. The calculations for the anisotropic case are similar and therefore omited.

For finding the (covariant) derivative, we need to pass to coordinates. Choose an orthonormal basis $\{\Xi_{i,j,k,l}\}_l$ in $\mathcal{T}_{m_{i,j,1}}\mathcal{M}(= (\mathcal{T}_m\mathcal{M}^{d_1\times d_2\times 2})_{i,j})$, then we will work in the normal coordinates of this basis and we can write $(\xi_m)_{i,j,k} = \sum_{l=1}^{L} \xi^{k,l}\Xi_{i,j,k,l}$, where L is the dimension of $\mathcal{M}$. Then, the lth component of $(\xi_m)_{i,j,k}$ after applying the proximal map is

$$(\max(1, \|(\xi_m)_{i,j,:}\|_{m,2}))^{-1}\xi^{k,l}\Xi_{i,j,k,l} = \left(\max\left(1, \sqrt{\sum_{\alpha=1}^{K}\sum_{\mu=1}^{L}(\xi^{\alpha,\mu})^2}\right)\right)^{-1}\xi^{k,l}\Xi_{i,j,k,l} \tag{379}$$

Now we note that in normal coordinates the Christoffel symbols vanish and we can focus on the derivatives of the coordinates. We can now distinguish cases in order to find our operator J_2 (with J_1 corresponding to anisotropic).

If $\|(\xi_m)_{i,j,:}\|_{m,2} \leq 1$ we have

$$(\max(1, \|(\xi_m)_{i,j,:}\|_{m,2}))^{-1}\xi^{k,l}\Xi_{i,j,k,l} = \xi^{k,l}\Xi_{i,j,k,l} \tag{380}$$

and hence we find

$$\partial_{\xi^{\kappa,\ell}}\xi^{k,l}\Xi_{i,j,k,l} = \delta^l_\ell\delta^k_\kappa\Xi_{i,j,k,l}. \tag{381}$$

Then, we have for $\eta_m = \sum_{a,b=1}^{d_1,d_2}\sum_{\kappa=1}^{K}\sum_{\ell=1}^{L}\eta^{\kappa,\ell}\Xi_{a,b,\kappa,\ell} \in \mathcal{T}_n\mathcal{T}\mathcal{M}^{d_1\times d_2\times 2}$

$$(J_2\eta_m)_{i,j,k} = \sum_{l=1}^{L}(J\eta_m)_{i,j,k,l} = \sum_{l=1}^{L}\sum_{\kappa=1}^{K}\sum_{\ell=1}^{L}\eta^{\kappa,\ell}(J\sum_{a,b=1}^{d_1,d_2}\Xi_{a,b,\kappa,\ell})_{i,j,k,l} \tag{382}$$

$$= \sum_{l=1}^{L}\sum_{\kappa=1}^{K}\sum_{\ell=1}^{L}\eta^{\kappa,\ell}\delta^l_\ell\delta^k_\kappa\Xi_{i,j,k,l} = \sum_{l=1}^{L}\eta^{k,l}\Xi_{i,j,k,l} = (\eta_m)_{i,j,k} \tag{383}$$

Otherwise, we have

$$(\max(1, \|(\xi_m)_{i,j,:}\|_{m,2}))^{-1}\xi^{l,k}\Xi_{i,j,k,l} = \frac{\eta^{k,l}}{\sqrt{\sum_{\alpha=1}^{K}\sum_{\mu=1}^{L}(\eta^{\alpha,\mu})^2}}\Xi_{i,j,k,l} \tag{384}$$

In order to calculate the derivative, we start with calculating the derivative of the numerator

$$\partial_{\xi^{\kappa,\ell}}\sqrt{\sum_{\alpha=1}^{K}\sum_{\mu=1}^{L}(\xi^{\alpha,\mu})^2} = \frac{1}{2}\frac{1}{\|(\xi_m)_{i,j,:}\|_{m,2}}(2\xi^{\kappa,\ell}) = \frac{\xi^{\kappa,\ell}}{\|(\xi_m)_{i,j,:}\|_{m,2}} \tag{385}$$

and hence for the total expression we get

$$\partial_{\xi^{\kappa,\ell}}\frac{\xi^{k,l}}{\sqrt{\sum_{\alpha=1}^{K}\sum_{\mu=1}^{L}(\xi^{\alpha,\mu})^2}}\Xi_{i,j,k,l} = \frac{1}{\|(\xi_m)_{i,j,:}\|_{m,2}^2}\left(\|(\xi_m)_{i,j,:}\|_{m,2}\delta_\ell^l\delta_\kappa^k - \frac{\xi^{k,l}\xi^{\kappa,\ell}}{\|(\xi_m)_{i,j,:}\|_{m,2}}\right)\Xi_{i,j,k,l} \tag{386}$$

$$= \frac{1}{\|(\xi_m)_{i,j,:}\|_{m,2}}\left(\delta_\ell^l\delta_\kappa^k - \frac{\xi^{k,l}\xi^{\kappa,\ell}}{\|(\xi_m)_{i,j,:}\|_{m,2}^2}\right)\Xi_{i,j,k,l} \tag{387}$$

Then, we have for $\eta_m = \sum_{a,b=1}^{d_1,d_2}\sum_{\kappa=1}^{K}\sum_{\ell=1}^{L}\eta^{\kappa,\ell}\Xi_{a,b,\kappa,\ell} \in \mathcal{T}_n\mathcal{TM}^{d_1\times d_2\times 2}$

$$(J_2\eta_m)_{i,j,k} = \sum_{l=1}^{L}(J\eta_m)_{i,j,k,l} = \sum_{l=1}^{L}\sum_{\kappa=1}^{K}\sum_{\ell=1}^{L}\eta^{\kappa,\ell}(J\sum_{a,b=1}^{d_1,d_2}\Xi_{a,b,\kappa,\ell})_{i,j,k,l} \tag{388}$$

$$= \sum_{l=1}^{L}\sum_{\kappa=1}^{K}\sum_{\ell=1}^{L}\eta^{\kappa,\ell}\frac{1}{\|(\xi_m)_{i,j,:}\|_{m,2}}\left(\delta_\ell^l\delta_\kappa^k - \frac{\xi^{k,l}\xi^{\kappa,\ell}}{\|(\xi_m)_{i,j,:}\|_{m,2}^2}\right)\Xi_{i,j,k,l} \tag{389}$$

$$= \sum_{l=1}^{L}\frac{1}{\|(\xi_m)_{i,j,:}\|_{m,2}}\left(\eta^{k,l} - \sum_{\kappa=1}^{K}\sum_{\ell=1}^{L}\eta^{\kappa,\ell}\frac{\langle(\xi_m)_{i,j,k},\Xi_{i,j,k,l}\rangle\langle(\xi_m)_{i,j,\kappa},\Xi_{i,j,\kappa,\ell}\rangle}{\|(\xi_m)_{i,j,:}\|_{m,2}^2}\right)\Xi_{i,j,k,l} \tag{390}$$

$$= \sum_{l=1}^{L}\frac{1}{\|(\xi_m)_{i,j,:}\|_{m,2}}\left(\eta^{k,l} - \sum_{\kappa=1}^{K}\frac{\langle(\xi_m)_{i,j,k},\Xi_{i,j,k,l}\rangle\langle(\xi_m)_{i,j,\kappa},(\eta_m)_{i,j,\kappa}\rangle}{\|(\xi_m)_{i,j,:}\|_{m,2}^2}\right)\Xi_{i,j,k,l} \tag{391}$$

$$= \frac{1}{\|(\xi_m)_{i,j,:}\|_{m,2}}\left(\sum_{l=1}^{L}\eta^{k,l}\Xi_{i,j,k,l} - \sum_{\kappa=1}^{K}\frac{\langle(\xi_m)_{i,j,\kappa},(\eta_m)_{i,j,\kappa}\rangle}{\|(\xi_m)_{i,j,:}\|_{m,2}^2}\sum_{l=1}^{L}\langle(\xi_m)_{i,j,k},\Xi_{i,j,k,l}\rangle\Xi_{i,j,k,l}\right) \tag{392}$$

$$= \frac{1}{\|(\xi_m)_{i,j,:}\|_{m,2}}\left((\eta_m)_{i,j,k} - \sum_{\kappa=1}^{K}\frac{\langle(\xi_m)_{i,j,\kappa},(\eta_m)_{i,j,\kappa}\rangle}{\|(\xi_m)_{i,j,:}\|_{m,2}^2}(\xi_m)_{i,j,k}\right) \tag{393}$$

Similarly we can find for the anisotropic case that if $\|(\xi_m)_{i,j,k}\|_m \leq 1$ we have

$$(J_1\eta_m)_{i,j,k} = (\eta_m)_{i,j,k} \tag{394}$$

and else

$$(J_1\eta_m)_{i,j,k} = \frac{1}{\|(\xi_m)_{i,j,k}\|_m}\left((\eta_m)_{i,j,k} - \frac{\langle(\xi_m)_{i,j,k},(\eta_m)_{i,j,k}\rangle}{\|(\xi_m)_{i,j,k}\|_m^2}(\xi_m)_{i,j,k}\right) \tag{395}$$

If we also include the boundary conditions in the operators J_q we find for the vector parts of our dual vector $\xi_n = (\zeta_m, \xi_m)$

$$(J_1(\xi_m)\eta_m)_{i,j,k} = \begin{cases} 0 & \text{if } i = d_1 \text{ and } k = 1 \\ 0 & \text{if } j = d_2 \text{ and } k = 2 \\ (\eta_m)_{i,j,k} & \text{if } \|(\xi_m)_{i,j,:}\|_m \leq 1 \\ \frac{1}{\|(\xi_m)_{i,j,k}\|_m} \left((\eta_m)_{i,j,k} - \frac{\langle (\xi_m)_{i,j,k}, (\eta_m)_{i,j,k} \rangle}{\|(\xi_m)_{i,j,k}\|_m^2} (\xi_m)_{i,j,k} \right) & \text{if } \|(\xi_m)_{i,j,k}\|_m > 1 \end{cases} \tag{396}$$

and

$$(J_2(\xi_m)\eta_m)_{i,j,k} = \begin{cases} 0 & \text{if } i = d_1 \text{ and } k = 1 \\ 0 & \text{if } j = d_2 \text{ and } k = 2 \\ (\eta_m)_{i,j,k} & \text{if } \|(\xi_m)_{i,j,:}\|_{m,2} \leq 1 \\ \frac{1}{\|(\xi_m)_{i,j,:}\|_{m,2}} \left((\eta_m)_{i,j,k} - \sum_{\kappa=1}^{K} \frac{\langle (\xi_m)_{i,j,\kappa}, (\eta_m)_{i,j,\kappa} \rangle}{\|(\xi_m)_{i,j,:}\|_{m,2}^2} (\xi_m)_{i,j,k} \right) & \text{if } \|(\xi_m)_{i,j,:}\|_{m,2} > 1 \end{cases} \tag{397}$$

A.2 Exact Solutions to 1D ℓ^2-TV for 2n gridpoints on manifolds

For this argument we will modify the idea used in [WDS14, Theorem 2]. Let $\mathcal{M}$ be a complete connected Riemannian manifold. Consider the following TV minimization problem on $2n$ gridpoints:

$$\min_{p \in \mathcal{M}^{2n}} \frac{1}{2\alpha} \sum_{i=1}^{n} d_{\mathcal{M}}(p_i, f_1)^2 + \frac{1}{2\alpha} \sum_{i=n+1}^{2n} d_{\mathcal{M}}(p_i, f_2)^2 + \sum_{i=1}^{2n-1} d_{\mathcal{M}}(p_i, p_{i+1}) \tag{398}$$

For finding a solution we will split up the proof in the following steps:

- For an optimal $\hat{p}$ we must have $d_{\mathcal{M}}(\hat{p}_i, f_1) \leq d_{\mathcal{M}}(f_1, f_2)$ for all $i \leq n$ and $d_{\mathcal{M}}(\hat{p}_i, f_2) \leq d_{\mathcal{M}}(f_1, f_2)$ for all $i > n$
- There is a minimizer where all components lie on a minimizing geodesic connecting b and a
- For that case the first n gridpoints have the same value and so do the second n gridpoints
- Using these results we reduce the problem a 1 dimensional problem in an Euclidean setting and solve this.

We want to note that if we have a Hadamard manifold, the minimizer is the only one.

Step 1
Let $\hat{p}$ be a minimizer. Now, if for all n gridpoints we have $d_{\mathcal{M}}(\hat{p}_i, f_1) > d_{\mathcal{M}}(f_1, f_2)$ then $\hat{p}_1^* = \hat{p}_2^* = ... = \hat{p}_{2n-1}^* = \hat{p}_{2n}^* = a$ would be a solution with a strictly smaller cost. This gives a contradiction.

Now, if among these first n points there are ℓ gridpoints, say i_j for $j = 1, .., \ell$, such that $d_{\mathcal{M}}(\hat{p}_{i_j}, f_1) \leq d_{\mathcal{M}}(f_1, f_2)$. Then $\hat{p}^*$ defined as

$$\hat{p}_i^* = \begin{cases} \hat{p}_{i_1} & 1 \leq i \leq i_1 \\ \hat{p}_{i_j} & i_j \leq i < i_{j+1} \\ \hat{p}_{i_\ell} & i_\ell \leq i \leq n \\ \hat{p}_i & n+1 \leq i \end{cases} \tag{399}$$

has a lower cost by the triangle inequality, which again gives a contradiction.

A similar claim follows for the second n gridpoints.

So we conclude that for an optimal $\hat{p}$ we must have $d_{\mathcal{M}}(\hat{p}_i, f_1) \leq d_{\mathcal{M}}(f_1, f_2)$ for all $i \leq n$ and $d_{\mathcal{M}}(\hat{p}_i, f_2) \leq d_{\mathcal{M}}(f_1, f_2)$ for all $i > n$.

Step 2

Next, we define $z \in \mathcal{M}^{2n}$ having components on the same minimizing geodesic, i.e,

$$z_i = \begin{cases} \gamma^*_{f_1,f_2}(d_{\mathcal{M}}(f_1, \hat{p}_i)) & i \leq n \\ \gamma^*_{f_2,f_1}(d_{\mathcal{M}}(f_2, \hat{p}_i)) & i > n \end{cases} \tag{400}$$

where $\gamma^*_{x,y}(t)$ is a *unit speed* minimizing geodesic from x to y at time $t \in [0, d_{\mathcal{M}}(x,y)]$. We will prove that z is also a minimizer. Now consider the case that we the components of z lie in the following order on the geodesic: $f_1, z_1, z_2, ..., z_{2n-1}, z_{2n}, f_2$. Then we have

$$d_{\mathcal{M}}(f_1, z_1) + \sum_{i=1}^{2n-1} d_{\mathcal{M}}(z_i, z_{i+1}) + d_{\mathcal{M}}(f_2, z_{2n}) = d_{\mathcal{M}}(f_1, f_2) \leq d_{\mathcal{M}}(f_1, \hat{p}_1) + \sum_{i=1}^{2n-1} d_{\mathcal{M}}(\hat{p}_i, \hat{p}_{i+1}) + d_{\mathcal{M}}(f_2, \hat{p}_{2n}) \tag{401}$$

Since we have $d_{\mathcal{M}}(f_1, z_1) = d_{\mathcal{M}}(f_1, \hat{p}_1)$ and $d_{\mathcal{M}}(f_2, z_{2n}) = d_{\mathcal{M}}(f_2, \hat{p}_{2n})$

$$\sum_{i=1}^{2n-1} d_{\mathcal{M}}(z_i, z_{i+1}) \leq \sum_{i=1}^{2n-1} d_{\mathcal{M}}(\hat{p}_i, \hat{p}_{i+1}) \tag{402}$$

must hold. Then we also have

$$\frac{1}{2\alpha} \sum_{i=1}^{n} d_{\mathcal{M}}(z_i, f_1)^2 + \frac{1}{2\alpha} \sum_{i=n+1}^{2n} d_{\mathcal{M}}(z_i, f_2)^2 + \sum_{i=1}^{2n-1} d_{\mathcal{M}}(z_i, z_{i+1}) \tag{403}$$

$$\leq \frac{1}{2\alpha} \sum_{i=1}^{n} d_{\mathcal{M}}(\hat{p}_i, f_1)^2 + \frac{1}{2\alpha} \sum_{i=n+1}^{2n} d_{\mathcal{M}}(\hat{p}_i, f_2)^2 + \sum_{i=1}^{2n-1} d_{\mathcal{M}}(\hat{p}_i, \hat{p}_{i+1}) \tag{404}$$

and we see that z is indeed also a minimizer.

By symmetry, for the remaining cases we can focus on the first n gridpoints. Without loss of generality, we also may assume that only one gridpoint in the first n points is out of order. This comes down to the case that we have the following ordering along the geodesic: $f_1, z_1, ..., z_j, z_i, z_{j+1}, ..., z_{2n}, f_2$ where $i < j$, $i \in \{1, .., n\}$ and $j \in \{1, .., 2n\}$. Now let $z' \in \mathcal{M}^{2n}$ be such that

$$z'_k = \begin{cases} z_k & k \neq i \\ z_j & k = i \end{cases} \tag{405}$$

Then by following the same reasoning as before we can show that z is a minimizer, but now z' would give a solution with a lower cost than we had, which we cannot have.

So we conclude that we can find a minmizer z such that $f_1, z_1, z_2, ..., z_{2n-1}, z_{2n}, f_2$ lie on a minmizing geodesic in this ordering.

Step 3
Finally, we will show that we in particular have $z_1 = z_2 = ... = z_n$ and $z_{n+1} = z_{n+2} = ... = z_{2n}$. Again by symmetry we only have to show that there are no jumps between the first n points. Now assume there are jumps and the first jump is at point i. We have two cases: $d_{\mathcal{M}}(z_i, f_1) > d_{\mathcal{M}}(z_{i+1}, f_1)$ and $d_{\mathcal{M}}(z_i, f_1) < d_{\mathcal{M}}(z_{i+1}, f_1)$. The former has already been eliminated in the previous step, so we can focus on showing the latter cannot be true either.

If we have $d_{\mathcal{M}}(z_i, f_1) < d_{\mathcal{M}}(z_{i+1}, f_1)$, then we can move z_{i+1} towards z_i and move all points $z_{i+2}, ..., z_n$ over the same distance along the geodesic, i.e, without changing distances between the moved points. Now we see that the loss of $d_{\mathcal{M}}(z_i, z_{i+1})$ will be cancelled equally by the gain of $d_{\mathcal{M}}(z_n, z_{n+1})$. However, the distance to b has strictly decreased for all moved points. So we again have found a solution with lower cost than the optimal solution. Again we have a contradiction and we also find that $d_{\mathcal{M}}(z_i, f_1) = d_{\mathcal{M}}(z_{i+1}, f_1)$.

From this we conclude that we must have $z_1 = z_2 = ... = z_n$ and $z_{n+1} = z_{n+2} = ... = z_{2n}$.

Step 4
Now we are ready to calculate the actual solution, which we will also call $\hat{p}$. By our previously established results, we now know tha the solution lies on the minimizing geodesic connecting f_1 and f_2. We write

$$\hat{p}_1 = ... = \hat{p}_n = \gamma_{f_1, f_2}(\hat{t_1}) \quad \hat{p}_{n+1} = ... = \hat{p}_{2n} = \gamma_{f_2, f_1}(\hat{t_2}) \tag{406}$$

where $\gamma_{x,y}(t)$ is a minimizing geodesic such that $\gamma_{x,y}(0) = x$ and $\gamma_{x,y}(1) = y$.

By symmetry we must have $\hat{t} = \hat{t_1} = \hat{t_2}$. Then we can rewrite our optimization problem (398) into a 1 dimensional problem

$$\min_t \frac{n}{2\alpha}(d_{\mathcal{M}}(f_1, f_2)t)^2 + \frac{n}{2\alpha}(d_{\mathcal{M}}(f_1, f_2)t)^2 + d_{\mathcal{M}}(f_1, f_2)|1 - 2t| \tag{407}$$

$$\Leftrightarrow \min_t \sup_g \frac{n}{\alpha} d_{\mathcal{M}}(f_1, f_2)t^2 + (1 - 2t)g - \iota_{B_1}(g), \tag{408}$$

where $B_1 = \{y \in \mathbb{R} | \, |y| \leq 1\}$. We may solve the minimization problem first. Differentiating to t gives

$$\frac{2n}{\alpha} d_{\mathcal{M}}(f_1, f_2)t - 2g = 0 \quad \Rightarrow \quad t = \frac{\alpha}{n} \frac{1}{d_{\mathcal{M}}(f_1, f_2)} g. \tag{409}$$

Filling this back into the optimization problem gives

$$\sup_g \frac{\alpha}{n}\frac{1}{d_{\mathcal{M}}(f_1,f_2)}g^2 + (1-2\frac{\alpha}{n}\frac{1}{d_{\mathcal{M}}(f_1,f_2)}g)g - \iota_{B_1}(g), \tag{410}$$

$$\Leftrightarrow \quad \sup_g -\frac{\alpha}{n}\frac{1}{d_{\mathcal{M}}(f_1,f_2)}g^2 + g - \iota_{B_1}(g) \tag{411}$$

$$\Leftrightarrow \quad \sup_g -\frac{\alpha}{n}\frac{1}{d_{\mathcal{M}}(f_1,f_2)}(g-\frac{n}{2\alpha}d_{\mathcal{M}}(f_1,f_2))^2 - \iota_{B_1}(g) \tag{412}$$

$$\Leftrightarrow \quad \inf_g \frac{\alpha}{n}\frac{1}{d_{\mathcal{M}}(f_1,f_2)}(g-\frac{n}{2\alpha}d_{\mathcal{M}}(f_1,f_2))^2 + \iota_{B_1}(g) \tag{413}$$

$$\Leftrightarrow \quad \hat{g} = \operatorname{prox}_{\frac{n}{\alpha}d_{\mathcal{M}}(f_1,f_2)\iota_{B_1}(\cdot)}\left(\frac{n}{2\alpha}d_{\mathcal{M}}(f_1,f_2)\right) = \frac{1}{\max(1,\frac{n}{2\alpha}d_{\mathcal{M}}(f_1,f_2))}\frac{n}{2\alpha}d_{\mathcal{M}}(f_1,f_2) \tag{414}$$

$$\Leftrightarrow \quad \hat{g} = \min\left(1,\frac{1}{n}\frac{2\alpha}{d_{\mathcal{M}}(f_1,f_2)}\right)\frac{n}{2\alpha}d_{\mathcal{M}}(f_1,f_2) = \min\left(\frac{1}{2},\frac{1}{n}\frac{\alpha}{d_{\mathcal{M}}(f_1,f_2)}\right)\frac{n}{\alpha}d_{\mathcal{M}}(f_1,f_2) \tag{415}$$

and we see that we get

$$\hat{t} = \frac{\alpha}{n}\frac{1}{d_{\mathcal{M}}(f_1,f_2)}\hat{g} = \min\left(\frac{1}{2},\frac{1}{n}\frac{\alpha}{d_{\mathcal{M}}(f_1,f_2)}\right) \tag{416}$$

References

[ABG07] Absil, P-A ; Baker, Christopher G. ; Gallivan, Kyle A.: Trust-region methods on Riemannian manifolds. In: *Foundations of Computational Mathematics* 7 (2007), Nr. 3, S. 303–330. http://dx.doi.org/10.1007/s10208-005-0179-9.

[ADM+02] Adler, Roy L. ; Dedieu, Jean-Pierre ; Margulies, Joseph Y. ; Martens, Marco ; Shub, Mike: Newton's method on Riemannian manifolds and a geometric model for the human spine. In: *IMA Journal of Numerical Analysis* 22 (2002), Nr. 3, S. 359–390. http://dx.doi.org/10.1093/imanum/22.3.359.

[AEK08] Abrudan, Traian E. ; Eriksson, Jan ; Koivunen, Visa: Steepest descent algorithms for optimization under unitary matrix constraint. In: *IEEE Transactions on Signal Processing* 56 (2008), Nr. 3, S. 1134–1147. http://dx.doi.org/10.1109/TSP.2007.908999.

[AF05] Azagra, Daniel ; Ferrera, Juan: Proximal calculus on Riemannian manifolds. In: *Mediterranean Journal of Mathematics* 2 (2005), Nr. 4, S. 437–450. http://dx.doi.org/10.1007/s00009-005-0056-4.

[AMS09] Absil, P-A ; Mahony, Robert ; Sepulchre, Rodolphe: *Optimization algorithms on matrix manifolds*. Princeton University Press, 2009. http://dx.doi.org/10.1515/9781400830244.

[ATV13] Afsari, Bijan ; Tron, Roberto ; Vidal, René: On the convergence of gradient descent for finding the Riemannian center of mass. In: *SIAM Journal on Control and Optimization* 51 (2013), Nr. 3, S. 2230–2260. http://dx.doi.org/10.1137/12086282X.

[AWK93] Adams, Brent L. ; Wright, Stuart I. ; Kunze, Karsten: Orientation imaging: the emergence of a new microscopy. In: *Metallurgical Transactions A* 24 (1993), Nr. 4, S. 819–831. http://dx.doi.org/10.1007/BF02656503.

[Bac14] Bacák, Miroslav: Computing medians and means in Hadamard spaces. In: *SIAM Journal on Optimization* 24 (2014), Nr. 3, S. 1542–1566. http://dx.doi.org/10.1137/140953393.

[Ban14] Banert, Sebastian: Backward–backward splitting in Hadamard spaces. In: *Journal of Mathematical Analysis and Applications* 414 (2014), Nr. 2, S. 656–665. http://dx.doi.org/10.1016/j.jmaa.2014.01.054.

[BBSW16] Bacák, Miroslav ; Bergmann, Ronny ; Steidl, Gabriele ; Weinmann, Andreas: A second order nonsmooth variational model for restoring manifold-valued images. In: *SIAM Journal on Scientific Computing* 38 (2016), Nr. 1, S. A567–A597. http://dx.doi.org/10.1137/15M101988X.

[BCH+15] Bergmann, Ronny ; Chan, Raymond H. ; Hielscher, Ralf ; Persch, Johannes ; Steidl, Gabriele: Restoration of manifold-valued images by half-quadratic minimization. In: *arXiv preprint arXiv:1505.07029* (2015). http://dx.doi.org/10.3934/ipi.2016001.

[BCNO16a] Bento, Glaydston de C. ; Cruz Neto, Joã. X. ; Oliveira, Paulo R.: A new approach to the proximal point method: convergence on general Riemannian manifolds. In: *Journal of Optimization Theory and Applications* 168 (2016), Nr. 3, S. 743–755. http://dx.doi.org/10.1007/s10957-015-0861-2.

[BCNO16b] Byrd, Richard H. ; Chin, Gillian M. ; Nocedal, Jorge ; Oztoprak, Figen: A family of second-order methods for convex ℓ^1-regularized optimization. In: *Mathematical Programming* 159 (2016), Nr. 1-2, S. 435–467. http://dx.doi.org/10.1007/s10107-015-0965-3.

[Ber11] Bertsekas, Dimitri P.: Incremental proximal methods for large scale convex optimization. In: *Mathematical programming* 129 (2011), Nr. 2, S. 163. http://dx.doi.org/10.1007/s10107-011-0472-0.

[Ber19] Bergmann, Ronny: manopt.jl. In: *Optimization on Manifolds in Julia.* (2019)

[BF12] Becker, Stephen ; Fadili, Jalal: A quasi-Newton proximal splitting method. In: *Advances in neural information processing systems*, 2012, S. 2618–2626

[BFFY18] Bortoloti, MAA ; Fernandes, TA ; Ferreira, OP ; Yuan, Jinyun: Damped Newton's Method on Riemannian Manifolds. In: *arXiv preprint arXiv:1803.05126* (2018)

[BFPS17] Bergmann, Ronny ; Fitschen, Jan H. ; Persch, Johannes ; Steidl, Gabriele: Infimal convolution coupling of first and second order differences on manifold-valued images. In: *International Conference on Scale Space and Variational Methods in Computer Vision* Springer, 2017, S. 447–459

[BFPS18] Bergmann, Ronny ; Fitschen, Jan H. ; Persch, Johannes ; Steidl, Gabriele: Priors with coupled first and second order differences for manifold-valued image processing. In: *Journal of mathematical imaging and vision* 60 (2018), Nr. 9, S. 1459–1481. http://dx.doi.org/10.1007/s10851-018-0840-y.

[BHSW18] Bredies, Kristian ; Holler, Martin ; Storath, Martin ; Weinmann, Andreas: Total generalized variation for manifold-valued data. In: *SIAM Journal on Imaging Sciences* 11 (2018), Nr. 3, S. 1785–1848. http://dx.doi.org/10.1137/17M1147597.

[BHTVN19] Bergmann, Ronny ; Herzog, Roland ; Tenbrinck, Daniel ; Vidal-Núñez, José: Fenchel Duality Theory and A Primal-Dual Algorithm on Riemannian Manifolds. (2019)

[BLPS18] Bergmann, Ronny ; Laus, Friederike ; Persch, Johannes ; Steidl, Gabriele: Recent Advances in Denoising of Manifold-Valued Images. In: *arXiv preprint arXiv:1812.08540* (2018)

[BLSW14] Bergmann, Ronny ; Laus, Friederike ; Steidl, Gabriele ; Weinmann, Andreas: Second order differences of cyclic data and applications in variational denoising. In: *SIAM Journal on Imaging Sciences* 7 (2014), Nr. 4, S. 2916–2953. http://dx.doi.org/10.1137/140969993.

[BML94] Basser, Peter J. ; Mattiello, James ; LeBihan, Denis: MR diffusion tensor spectroscopy and imaging. In: *Biophysical journal* 66 (1994), Nr. 1, S. 259–267. http://dx.doi.org/10.1016/S0006-3495(94)80775-1.

[BNO11] Bento, GC ; Neto, JX ; Oliveira, PR: Convergence of inexact descent methods for nonconvex optimization on Riemannian manifolds. In: *arXiv preprint arXiv:1103.4828* (2011)

[BNO16] Byrd, Richard H. ; Nocedal, Jorge ; Oztoprak, Figen: An inexact successive quadratic approximation method for L-1 regularized optimization. In: *Mathematical Programming* 157 (2016), Nr. 2, S. 375–396. http://dx.doi.org/10.1007/s10107-015-0941-y.

[BPS16] Bergmann, Ronny ; Persch, Johannes ; Steidl, Gabriele: A parallel Douglas–Rachford algorithm for minimizing ROF-like functionals on images with values in symmetric Hadamard manifolds. In: *SIAM Journal on Imaging Sciences* 9 (2016), Nr. 3, S. 901–937. http://dx.doi.org/10.1137/15M1052858.

[BST14] Bolte, Jérôme ; Sabach, Shoham ; Teboulle, Marc: Proximal alternating linearized minimization for nonconvex and nonsmooth problems. In: *Mathematical Programming* 146 (2014), Nr. 1-2, S. 459–494. http://dx.doi.org/10.1007/s10107-013-0701-9.

[BT18] Bergmann, Ronny ; Tenbrinck, Daniel: A graph framework for manifold-valued data. In: *SIAM Journal on Imaging Sciences* 11 (2018), Nr. 1, S. 325–360. http://dx.doi.org/10.1137/17M1118567.

[BW15] Bergmann, Ronny ; Weinmann, Andreas: Inpainting of cyclic data using first and second order differences. In: *International Workshop on Energy Minimization Methods in Computer Vision and Pattern Recognition* Springer, 2015, S. 155–168

[BWW$^+$16] Baust, Maximilian ; Weinmann, Andreas ; Wieczorek, Matthias ; Lasser, Tobias ; Storath, Martin ; Navab, Nassir: Combined tensor fitting and TV regularization in diffusion tensor imaging based on a Riemannian manifold approach. In: *IEEE transactions on medical imaging* 35 (2016), Nr. 8, S. 1972–1989. `http://dx.doi.org/10.1109/TMI.2016.2528820`.

[Car92] Carmo, Manfredo Perdigao d.: *Riemannian geometry.* Birkhäuser, 1992

[CDGS17] Castro, Rodrigo ; Di Giorgi, Gustavo ; Sierra, Willy: Secant Method on Riemannian Manifolds. In: *arXiv preprint arXiv:1712.02655* (2017)

[CE08] Cheeger, Jeff ; Ebin, David G.: *Comparison theorems in Riemannian geometry.* Bd. 365. American Mathematical Soc., 2008

[Che14] Chen, Dai-Qiang: *Fixed Point Algorithm Based on Quasi-Newton Method for Convex Minimization Problem with Application to Image Deblurring.* 2014

[CJ19] Calinon, Sylvain ; Jaquier, Noémie: *Gaussians on Riemannian Manifolds for Robot Learning and Adaptive Control.* 2019

[CJY16] Chen, Weiqiang ; Ji, Hui ; You, Yanfei: An augmented lagrangian method for ℓ^1-regularized optimization problems with orthogonality constraints. In: *SIAM Journal on Scientific Computing* 38 (2016), Nr. 4, S. B570–B592. `http://dx.doi.org/10.1137/140988875`.

[CKS01] Chan, Tony F. ; Kang, Sung H. ; Shen, Jianhong: Total variation denoising and enhancement of color images based on the CB and HSV color models. In: *Journal of Visual Communication and Image Representation* 12 (2001), Nr. 4, S. 422–435. `http://dx.doi.org/10.1006/jvci.2001.0491`.

[Cla90] Clarke, Frank H.: *Optimization and nonsmooth analysis.* Bd. 5. Siam, 1990. `http://dx.doi.org/10.1137/1.9781611971309`.

[CMMCSZ20] Chen, Shixiang ; Ma, Shiqian ; Man-Cho So, Anthony ; Zhang, Tong: Proximal gradient method for nonsmooth optimization over the Stiefel manifold. In: *SIAM Journal on Optimization* 30 (2020), Nr. 1, S. 210–239. `http://dx.doi.org/10.1137/18M122457X`.

[Com13] Commons, Wikimedia: *File:EBSD in process.png — Wikimedia Commons, the free media repository.* https://commons.wikimedia.org/w/index.php?title=File:EBSD_in_process.png&oldid=92599066. Version: 2013. – [Online; accessed 10-September-2020]

[Com18] Commons, Wikimedia: *File:HSV color solid cylinder.png — Wikimedia Commons, the free media repository.* https://commons.wikimedia.org/w/index.php?title=File:HSV_color_solid_cylinder.png&oldid=314486272. Version: 2018. – [Online; accessed 10-September-2020]

[Com19] Commons, Wikimedia: *File:SAR Kilauea topo interferogram.jpg — Wikimedia Commons, the free media repository.* https://commons.wikimedia.org/w/index.php?title=File:SAR_Kilauea_topo_interferogram.jpg&oldid=377560242. Version: 2019. – [Online; accessed 10-September-2020]

[Com20] Commons, Wikimedia: *File:DTI-axial-ellipsoids.jpg — Wikimedia Commons, the free media repository.* https://commons.wikimedia.org/w/index.php?title=File:DTI-axial-ellipsoids.jpg&oldid=453175503. Version: 2020. – [Online; accessed 10-September-2020]

[CP07] Combettes, Patrick L. ; Pesquet, Jean-Christophe: A Douglas–Rachford splitting approach to nonsmooth convex variational signal recovery. In: *IEEE Journal of Selected Topics in Signal Processing* 1 (2007), Nr. 4, S. 564–574. http://dx.doi.org/10.1109/JSTSP.2007.910264.

[CP11] Chambolle, Antonin ; Pock, Thomas: A first-order primal-dual algorithm for convex problems with applications to imaging. In: *Journal of mathematical imaging and vision* 40 (2011), Nr. 1, S. 120–145. http://dx.doi.org/10.1007/s10851-010-0251-1.

[CS13] Cremers, Daniel ; Strekalovskiy, Evgeny: Total cyclic variation and generalizations. In: *Journal of mathematical imaging and vision* 47 (2013), Nr. 3, S. 258–277. http://dx.doi.org/10.1007/s10851-012-0396-1.

[CTDF04] Chefd'Hotel, Christophe ; Tschumperlé, David ; Deriche, Rachid ; Faugeras, O: Regularizing flows for constrained matrix-valued images. In: *Journal of Mathematical Imaging and Vision* 20 (2004), Nr. 1-2, S. 147–162. http://dx.doi.org/10.1023/B:JMIV.0000011920.58935.9c.

[CV20] Clason, Christian ; Valkonen, Tuomo: Introduction to Nonsmooth Analysis and Optimization. In: *arXiv preprint arXiv:2001.00216* (2020)

[CW05] Combettes, Patrick L. ; Wajs, Valérie R: Signal recovery by proximal forward-backward splitting. In: *Multiscale Modeling & Simulation* 4 (2005), Nr. 4, S. 1168–1200. http://dx.doi.org/10.1137/050626090.

[DDDM04] Daubechies, Ingrid ; Defrise, Michel ; De Mol, Christine: An iterative thresholding algorithm for linear inverse problems with a sparsity constraint. In: *Communications on Pure and Applied Mathematics: A Journal Issued by the Courant Institute of Mathematical Sciences* 57 (2004), Nr. 11, S. 1413–1457. http://dx.doi.org/10.1002/cpa.20042.

[DLFK96] De Luca, Tecla ; Facchinei, Francisco ; Kanzow, Christian: A semismooth equation approach to the solution of nonlinear complementarity problems. In: *Mathematical programming* 75 (1996), Nr. 3, S. 407–439. http://dx.doi.org/10.1007/BF02592192.

[Don06] Donoho, David L.: Compressed sensing. In: *IEEE Transactions on information theory* 52 (2006), Nr. 4, S. 1289–1306. http://dx.doi.org/10.1109/TIT.2006.871582.

[DPM03] Dedieu, Jean-Pierre ; Priouret, Pierre ; Malajovich, Gregorio: Newton's method on Riemannian manifolds: covariant alpha theory. In: *IMA Journal of Numerical Analysis* 23 (2003), Nr. 3, S. 395–419. http://dx.doi.org/10.1093/imanum/23.3.395.

[EAS98] Edelman, Alan ; Arias, Tomás A ; Smith, Steven T.: The geometry of algorithms with orthogonality constraints. In: *SIAM journal on Matrix Analysis and Applications* 20 (1998), Nr. 2, S. 303–353. http://dx.doi.org/10.1137/S0895479895290954.

[EZC10] Esser, Ernie ; Zhang, Xiaoqun ; Chan, Tony F.: A general framework for a class of first order primal-dual algorithms for convex optimization in imaging science. In: *SIAM Journal on Imaging Sciences* 3 (2010), Nr. 4, S. 1015–1046. http://dx.doi.org/10.1137/09076934X.

[FFK96] Facchinei, Francisco ; Fischer, Andreas ; Kanzow, Christian: Inexact Newton methods for semismooth equations with applications to variational inequality problems. In: *Nonlinear Optimization and Applications.* Springer, 1996, S. 125–139

[FGZ14] Fountoulakis, Kimon ; Gondzio, Jacek ; Zhlobich, Pavel: Matrix-free interior point method for compressed sensing problems. In: *Mathematical Programming Computation* 6 (2014), Nr. 1, S. 1–31. http://dx.doi.org/10.1007/s12532-013-0063-6.

[FJ07] Fletcher, P T. ; Joshi, Sarang: Riemannian geometry for the statistical analysis of diffusion tensor data. In: *Signal Processing* 87 (2007), Nr. 2, S. 250–262. http://dx.doi.org/10.1016/j.sigpro.2005.12.018.

[FO98] Ferreira, OP ; Oliveira, PR: Subgradient algorithm on Riemannian manifolds. In: *Journal of Optimization Theory and Applications* 97 (1998), Nr. 1, S. 93–104. http://dx.doi.org/10.1023/A:1022675100677.

[FO02] Ferreira, OP ; Oliveira, PR: Proximal point algorithm on Riemannian manifolds. In: *Optimization* 51 (2002), Nr. 2, S. 257–270. http://dx.doi.org/10.1080/02331930290019413.

[FP07] Facchinei, Francisco ; Pang, Jong-Shi: *Finite-dimensional variational inequalities and complementarity problems*. Springer Science & Business Media, 2007

[GH16a] Grohs, Philipp ; Hosseini, Seyedehsomayeh: Nonsmooth trust region algorithms for locally Lipschitz functions on Riemannian manifolds. In: *IMA Journal of Numerical Analysis* 36 (2016), Nr. 3, S. 1167–1192. http://dx.doi.org/10.1093/imanum/drv043.

[GH16b] Grohs, Philipp ; Hosseini, Seyedehsomayeh: ε-subgradient algorithms for locally Lipschitz functions on Riemannian manifolds. In: *Advances in Computational Mathematics* 42 (2016), Nr. 2, S. 333–360. http://dx.doi.org/10.1007/s10444-015-9426-z.

[GL08] Griesse, Roland ; Lorenz, Dirk A.: A semismooth Newton method for Tikhonov functionals with sparsity constraints. In: *Inverse Problems* 24 (2008), Nr. 3, S. 035007. http://dx.doi.org/10.1088/0266-5611/24/3/035007.

[GM06] Giaquinta, Mariano ; Mucci, Domenico: The BV-energy of maps into a manifold: Relaxation and density results. In: *Annali della Scuola Normale Superiore di Pisa-Classe di Scienze* 5 (2006), Nr. 4, S. 483–548

[GM07] Giaquinta, Mariano ; Mucci, Domenico: Maps of bounded variation with values into a manifold: total variation and relaxed energy. In: *Pure and Applied Mathematics Quarterly* 3 (2007), Nr. 2, S. 513–538. http://dx.doi.org/10.4310/PAMQ.2007.v3.n2.a6.

[GMS93] Giaquinta, Mariano ; Modica, Giuseppe ; Souček, Jaroslav: Variational problems for maps of bounded variation with values in S^1. In: *Calculus of Variations and Partial Differential Equations* 1 (1993), Nr. 1, S. 87–121. http://dx.doi.org/10.1007/BF02163266.

[GS14] Grohs, Philipp ; Sprecher, Markus: Total variation regularization by iteratively reweighted least squares on Hadamard spaces and the sphere. In: *preprint* 39 (2014)

[GS16] Grohs, Philipp ; Sprecher, Markus: Total variation regularization on Riemannian manifolds by iteratively reweighted minimization. In: *Information and Inference: A Journal of the IMA* 5 (2016), Nr. 4, S. 353–378. http://dx.doi.org/10.1093/imaiai/iaw011.

[HHY18] Hosseini, Seyedehsomayeh ; Huang, Wen ; Yousefpour, Rohollah: Line search algorithms for locally Lipschitz functions on Riemannian manifolds. In: *SIAM Journal on Optimization* 28 (2018), Nr. 1, S. 596–619. http://dx.doi.org/10.1137/16M1108145.

[HIK02] Hintermüller, M. ; Ito, K. ; Kunisch, K.: The Primal-Dual Active Set Strategy as a Semismooth Newton Method. In: *SIAM Journal on Optimization* 13 (2002), jan, Nr. 3, S. 865–888. http://dx.doi.org/10.1137/s1052623401383558.

[Hin10] Hintermüller, Michael: Semismooth Newton methods and applications. In: *Department of Mathematics, Humboldt-University of Berlin* (2010)

[Hos15] Hosseini, S: Convergence of nonsmooth descent methods via Kurdyka-Lojasiewicz inequality on Riemannian manifolds. In: *Hausdorff Center for Mathematics and Institute for Numerical Simulation, University of Bonn (2015,(INS Preprint No. 1523))* (2015)

[HP11] Hosseini, S ; Pouryayevali, MR: Generalized gradients and characterization of epi-Lipschitz sets in Riemannian manifolds. In: *Nonlinear Analysis: Theory, Methods & Applications* 74 (2011), Nr. 12, S. 3884–3895. http://dx.doi.org/10.1016/j.na.2011.02.023.

[HU17] Hosseini, Seyedehsomayeh ; Uschmajew, André: A Riemannian gradient sampling algorithm for nonsmooth optimization on manifolds. In: *SIAM Journal on Optimization* 27 (2017), Nr. 1, S. 173–189. http://dx.doi.org/10.1137/16M1069298.

[HWY13] He, Jinsu ; Wang, Jinhua ; Yao, Jen-Chih: Convergence criteria of Newton's method on Lie groups. In: *Fixed Point Theory and Applications* 2013 (2013), Nr. 1, S. 293. http://dx.doi.org/10.1186/1687-1812-2013-293.

[HYY14] He, Bingsheng ; You, Yanfei ; Yuan, Xiaoming: On the convergence of primal-dual hybrid gradient algorithm. In: *SIAM Journal on Imaging Sciences* 7 (2014), Nr. 4, S. 2526–2537. http://dx.doi.org/10.1137/140963467.

[KA10] Kakavandi, Bijan A. ; Amini, Massoud: Duality and subdifferential for convex functions on complete CAT (0) metric spaces. In: *Nonlinear Analysis: Theory, Methods & Applications* 73 (2010), Nr. 10, S. 3450–3455. http://dx.doi.org/10.1016/j.na.2010.07.033.

[KGB16] Kovnatsky, Artiom ; Glashoff, Klaus ; Bronstein, Michael M.: MADMM: a generic algorithm for non-smooth optimization on manifolds. In: *European Conference on Computer Vision* Springer, 2016, S. 680–696

[KS02] Kimmel, Ron ; Sochen, Nir: Orientation diffusion or how to comb a porcupine. In: *Journal of Visual Communication and Image Representation* 13 (2002), Nr. 1-2, S. 238–248. `http://dx.doi.org/10.1006/jvci.2001.0501`.

[Lee13] Lee, John M.: Smooth manifolds. Version: 2013. `http://dx.doi.org/10.1007/978-1-4419-9982-5`. In: *Introduction to Smooth Manifolds*. Springer, 2013.

[LLWS13] Lellmann, Jan ; Lellmann, Björn ; Widmann, Florian ; Schnörr, Christoph: Discrete and continuous models for partitioning problems. In: *International journal of computer vision* 104 (2013), Nr. 3, S. 241–269

[LNPS17] Laus, Friederike ; Nikolova, Mila ; Persch, Johannes ; Steidl, Gabriele: A nonlocal denoising algorithm for manifold-valued images using second order statistics. In: *SIAM Journal on Imaging Sciences* 10 (2017), Nr. 1, S. 416–448. `http://dx.doi.org/10.1137/16M1087114`.

[LO14] Lai, Rongjie ; Osher, Stanley: A splitting method for orthogonality constrained problems. In: *Journal of Scientific Computing* 58 (2014), Nr. 2, S. 431–449. `http://dx.doi.org/10.1007/s10915-013-9740-x`.

[LSKC13] Lellmann, Jan ; Strekalovskiy, Evgeny ; Koetter, Sabrina ; Cremers, Daniel: Total variation regularization for functions with values in a manifold. In: *Proceedings of the IEEE International Conference on Computer Vision*, 2013, S. 2944–2951

[LSS14] Lee, Jason D. ; Sun, Yuekai ; Saunders, Michael A.: Proximal Newton-type methods for minimizing composite functions. In: *SIAM Journal on Optimization* 24 (2014), Nr. 3, S. 1420–1443. `http://dx.doi.org/10.1137/130921428`.

[LST18] Li, Xudong ; Sun, Defeng ; Toh, Kim-Chuan: A highly efficient semismooth Newton augmented Lagrangian method for solving Lasso problems. In: *SIAM Journal on Optimization* 28 (2018), Nr. 1, S. 433–458. `http://dx.doi.org/10.1137/16M1097572`.

[Lue72] Luenberger, David G.: The gradient projection method along geodesics. In: *Management Science* 18 (1972), Nr. 11, S. 620–631. `http://dx.doi.org/10.1287/mnsc.18.11.620`.

[MF98] Massonnet, Didier ; Feigl, Kurt L.: Radar interferometry and its application to changes in the Earth's surface. In: *Reviews of geophysics* 36 (1998), Nr. 4, S. 441–500. `http://dx.doi.org/10.1029/97RG03139`.

[Mif77] Mifflin, Robert: Semismooth and semiconvex functions in constrained optimization. In: *SIAM Journal on Control and Optimization* 15 (1977), Nr. 6, S. 959–972. `http://dx.doi.org/10.1137/0315061`.

[MQ95] Martínez, JoséMario ; Qi, Liqun: Inexact Newton methods for solving nonsmooth equations. In: *Journal of Computational and Applied Mathematics* 60 (1995), Nr. 1-2, S. 127–145. `http://dx.doi.org/10.1016/0377-0427(94)00088-I`.

[MU14] Milzarek, Andre ; Ulbrich, Michael: A semismooth Newton method with multidimensional filter globalization for ℓ^1-optimization. In: *SIAM Journal on Optimization* 24 (2014), Nr. 1, S. 298–333. `http://dx.doi.org/10.1137/120892167`.

[NQYH07] Ng, Michael K. ; Qi, Liqun ; Yang, Yu-Fei ; Huang, Yu-Mei: On semismooth Newton's methods for total variation minimization. In: *Journal of Mathematical Imaging and Vision* 27 (2007), Nr. 3, S. 265–276. `http://dx.doi.org/10.1007/s10851-007-0650-0`.

[OF18] Oliveira, Fabiana R. ; Ferreira, Orizon P.: Newton method for finding a singularity of a special class of locally Lipschitz continuous vector fields on Riemannian manifolds. In: *arXiv preprint arXiv:1810.11636* (2018)

[PC11] Pock, Thomas ; Chambolle, Antonin: Diagonal preconditioning for first order primal-dual algorithms in convex optimization. In: *2011 International Conference on Computer Vision* IEEE, 2011, S. 1762–1769

[Pen06] Pennec, Xavier: Intrinsic statistics on Riemannian manifolds: Basic tools for geometric measurements. In: *Journal of Mathematical Imaging and Vision* 25 (2006), Nr. 1, S. 127. `http://dx.doi.org/10.1007/s10851-006-6228-4`.

[Pen18] Pennec, Xavier: Parallel transport with pole ladder: a third order scheme in affine connection spaces which is exact in affine symmetric spaces. In: *arXiv preprint arXiv:1805.11436* (2018)

[Per18] Persch, Johannes: *Optimization Methods for Manifold-valued Image Processing.* Verlag Dr. Hut, 2018

[PFA06] Pennec, Xavier ; Fillard, Pierre ; Ayache, Nicholas: A Riemannian framework for tensor computing. In: *International Journal of computer vision* 66 (2006), Nr. 1, S. 41–66. `http://dx.doi.org/10.1007/s11263-005-3222-z`.

[PJL$^+$13] Pan, Han ; Jing, Zhongliang ; Lei, Ming ; Liu, Rongli ; Jin, Bo ; Zhang, Canlong: A sparse proximal Newton splitting method for constrained image deblurring. In: *Neurocomputing* 122 (2013), S. 245–257. `http://dx.doi.org/10.1016/j.neucom.2013.06.027`.

[PSB14] Patrinos, Panagiotis ; Stella, Lorenzo ; Bemporad, Alberto: Forward-backward truncated Newton methods for convex composite optimization. In: *arXiv preprint arXiv:1402.6655* (2014)

[Qi93] Qi, Liqun: Convergence analysis of some algorithms for solving nonsmooth equations. In: *Mathematics of operations research* 18 (1993), Nr. 1, S. 227–244. `http://dx.doi.org/10.1287/moor.18.1.227`.

[QS93] Qi, Liqun ; Sun, Jie: A nonsmooth version of Newton's method. In: *Mathematical programming* 58 (1993), Nr. 1-3, S. 353–367. `http://dx.doi.org/10.1007/BF01581275`.

[RLV17] Rust, Caterina ; Lellmann, Jan ; Vogt, Thomas: *Semiglatte Optimierungsverfahren zweiter Ordnung in der Bildverarbeitung*, University of Lübeck, Diplomarbeit, 2017

[ROF92] Rudin, Leonid I. ; Osher, Stanley ; Fatemi, Emad: Nonlinear total variation based noise removal algorithms. In: *Physica D: nonlinear phenomena* 60 (1992), Nr. 1-4, S. 259–268. `http://dx.doi.org/10.1016/0167-2789(92)90242-F`.

[RW09] Rockafellar, R T. ; Wets, Roger J-B: *Variational analysis*. Bd. 317. Springer Science & Business Media, 2009. `http://dx.doi.org/10.1007/978-3-642-02431-3`.

[SC11] Strekalovskiy, Evgeny ; Cremers, Daniel: Total variation for cyclic structures: Convex relaxation and efficient minimization. In: *CVPR 2011* IEEE, 2011, S. 1905–1911

[SH97] Sun, Defeng ; Han, Jiye: Newton and quasi-Newton methods for a class of nonsmooth equations and related problems. In: *SIAM Journal on Optimization* 7 (1997), Nr. 2, S. 463–480. `http://dx.doi.org/10.1137/S1052623494274970`.

[Smi94] Smith, Steven T.: Optimization techniques on Riemannian manifolds. In: *Fields institute communications* 3 (1994), Nr. 3, S. 113–135

[SW18] Storath, Martin ; Weinmann, Andreas: Wavelet sparse regularization for manifold-valued data. In: *arXiv preprint arXiv:1808.00505* (2018)

[SWU16] Storath, Martin ; Weinmann, Andreas ; Unser, Michael: Exact algorithms for L^1-TV regularization of real-valued or circle-valued signals. In: *SIAM Journal on Scientific Computing* 38 (2016), Nr. 1, S. A614–A630. `http://dx.doi.org/10.1137/15M101796X`.

[Udr94] Udriste, Constantin: *Convex functions and optimization methods on Riemannian manifolds*. Bd. 297. Springer Science & Business Media, 1994

[Ulb02] Ulbrich, Michael: Semismooth Newton methods for operator equations in function spaces. In: *SIAM Journal on Optimization* 13 (2002), Nr. 3, S. 805–841. http://dx.doi.org/10.1137/S1052623400371569.

[VBK13] Valkonen, Tuomo ; Bredies, Kristian ; Knoll, Florian: Total generalized variation in diffusion tensor imaging. In: *SIAM Journal on Imaging Sciences* 6 (2013), Nr. 1, S. 487–525. http://dx.doi.org/10.1137/120867172.

[VSCL19] Vogt, Thomas ; Strekalovskiy, Evgeny ; Cremers, Daniel ; Lellmann, Jan: Lifting methods for manifold-valued variational problems. In: *arXiv preprint arXiv:1908.03776* (2019)

[WDS14] Weinmann, Andreas ; Demaret, Laurent ; Storath, Martin: Total variation regularization for manifold-valued data. In: *SIAM Journal on Imaging Sciences* 7 (2014), Nr. 4, S. 2226–2257. http://dx.doi.org/10.1137/130951075.

[WDS16] Weinmann, Andreas ; Demaret, Laurent ; Storath, Martin: Mumford–Shah and Potts regularization for manifold-valued data. In: *Journal of Mathematical Imaging and Vision* 55 (2016), Nr. 3, S. 428–445. http://dx.doi.org/10.1007/s10851-015-0628-2.

[WNF09] Wright, Stephen J. ; Nowak, Robert D. ; Figueiredo, Mário AT: Sparse reconstruction by separable approximation. In: *IEEE Transactions on Signal Processing* 57 (2009), Nr. 7, S. 2479–2493. http://dx.doi.org/10.1109/TSP.2009.2016892.

[WYYZ08] Wang, Yilun ; Yang, Junfeng ; Yin, Wotao ; Zhang, Yin: A new alternating minimization algorithm for total variation image reconstruction. In: *SIAM Journal on Imaging Sciences* 1 (2008), Nr. 3, S. 248–272. http://dx.doi.org/10.1137/080724265.

[XLWZ18] Xiao, Xiantao ; Li, Yongfeng ; Wen, Zaiwen ; Zhang, Liwei: A regularized semi-smooth Newton method with projection steps for composite convex programs. In: *Journal of Scientific Computing* 76 (2018), Nr. 1, S. 364–389. http://dx.doi.org/10.1007/s10915-017-0624-3.

[YOGD08] Yin, Wotao ; Osher, Stanley ; Goldfarb, Donald ; Darbon, Jerome: Bregman iterative algorithms for ℓ^1-minimization with applications to compressed sensing. In: *SIAM Journal on Imaging sciences* 1 (2008), Nr. 1, S. 143–168. http://dx.doi.org/10.1137/070703983.

[YVGS10] Yu, Jin ; Vishwanathan, SVN ; Günter, Simon ; Schraudolph, Nicol N.: A quasi-Newton approach to nonsmooth convex optimization problems in machine learning. In: *Journal of Machine Learning Research* 11 (2010), Nr. Mar, S. 1145–1200. http://dx.doi.org/10.1145/1390156.1390309.

[YZS19] Yue, Man-Chung ; Zhou, Zirui ; So, Anthony Man-Cho: A family of inexact SQA methods for non-smooth convex minimization with provable convergence guarantees based on the Luo–Tseng error bound property. In: *Mathematical Programming* 174 (2019), Nr. 1-2, S. 327–358. http://dx.doi.org/10.1007/s10107-018-1280-6.

[ZT05] Zhou, Guanglu ; Toh, Kim-Chuan: Superlinear convergence of a Newton-type algorithm for monotone equations. In: *Journal of optimization theory and applications* 125 (2005), Nr. 1, S. 205–221. http://dx.doi.org/10.1007/s10957-004-1721-7.

[ZZCL17] Zhu, Hong ; Zhang, Xiaowei ; Chu, Delin ; Liao, Li-Zhi: Nonconvex and nonsmooth optimization with generalized orthogonality constraints: An approximate augmented Lagrangian method. In: *Journal of Scientific Computing* 72 (2017), Nr. 1, S. 331–372. http://dx.doi.org/10.1007/s10915-017-0359-1.