

# Splitting Methods for Non-smooth Optimization on Manifolds

Ronny Bergmann

joint work with

R. Herzog, M. Silva Louzeiro, J. Persch, G. Steidl, D. Tenbrinck,  
J. Vidal-Núñez.

DNA Seminar, NTNU,

April 4, 2022



# Splitting Methods in Optimization

When solving an nonsmooth, high-dimensional optimisation problem

$$\arg \min_{p \in \mathcal{M}} f(p)$$

we want to use that our  $f: \mathcal{M} \rightarrow \overline{\mathbb{R}}$  can be written as

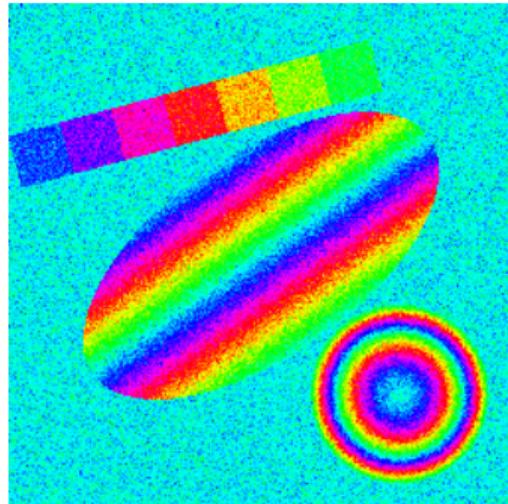
$$\arg \min_{p \in \mathcal{M}} \sum_{i=1}^N f_i(p)$$

for optimisation problems on a [Riemannian manifold  \$\mathcal{M}\$](#) .

# Manifold-valued Signal & Image Processing

Tasks in [Image Processing](#) is phrased as an optimisation problem.  
**Here.** The pixel take values on a manifold

- ▶ phase-valued data ( $\mathbb{S}^1$ )
- ▶ wind-fields, GPS ( $\mathbb{S}^2$ )
- ▶ DT-MRI ( $\mathcal{P}(3)$ )
- ▶ EBSD, (grain) orientations ( $\text{SO}(n)$ )



Artificial noisy phase-valued data.

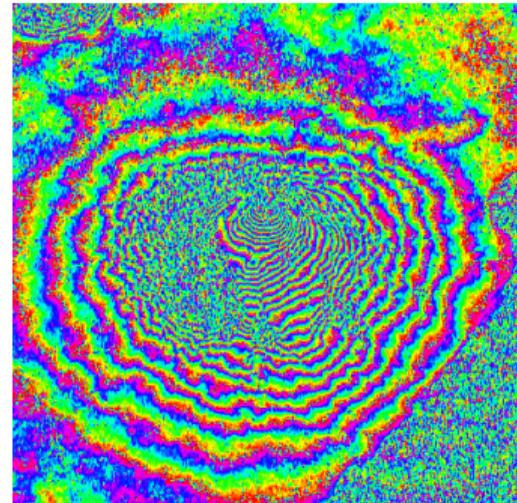
**Tasks.** Denoising, Inpainting, labeling (classification), deblurring,...

# Manifold-valued Signal & Image Processing

Tasks in [Image Processing](#) is phrased as an optimisation problem.

**Here.** The pixel take values on a manifold

- ▶ phase-valued data ( $\mathbb{S}^1$ )
- ▶ wind-fields, GPS ( $\mathbb{S}^2$ )
- ▶ DT-MRI ( $\mathcal{P}(3)$ )
- ▶ EBSD, (grain) orientations ( $\text{SO}(n)$ )



InSAR-Data of Mt. Vesuvius.

[Rocca, Prati, and Guarnieri 1997]

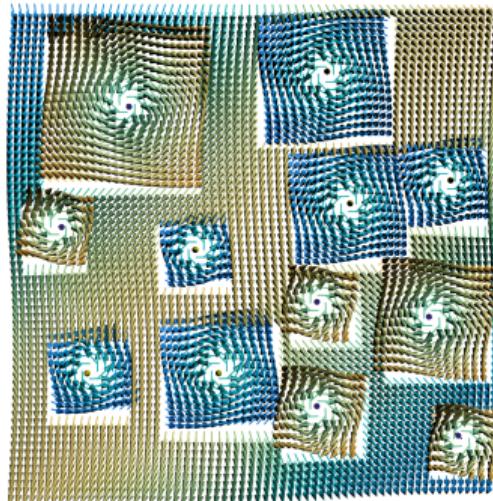
**Tasks.** Denoising, Inpainting, labeling (classification), deblurring,...

# Manifold-valued Signal & Image Processing

Tasks in [Image Processing](#) is phrased as an optimisation problem.

**Here.** The pixel take values on a manifold

- ▶ phase-valued data ( $\mathbb{S}^1$ )
- ▶ wind-fields, GPS ( $\mathbb{S}^2$ )
- ▶ DT-MRI ( $\mathcal{P}(3)$ )
- ▶ EBSD, (grain) orientations ( $\text{SO}(n)$ )



Artificial noisy data on the sphere  $\mathbb{S}^2$ .

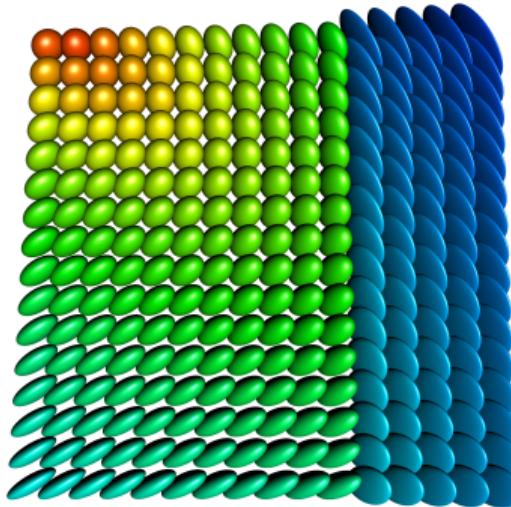
**Tasks.** Denoising, Inpainting, labeling (classification), deblurring,...

# Manifold-valued Signal & Image Processing

Tasks in [Image Processing](#) is phrased as an optimisation problem.

**Here.** The pixel take values on a manifold

- ▶ phase-valued data ( $\mathbb{S}^1$ )
- ▶ wind-fields, GPS ( $\mathbb{S}^2$ )
- ▶ DT-MRI ( $\mathcal{P}(3)$ )
- ▶ EBSD, (grain) orientations ( $\text{SO}(n)$ )



Artificial diffusion data,  
each pixel is a symmetric positive matrix.

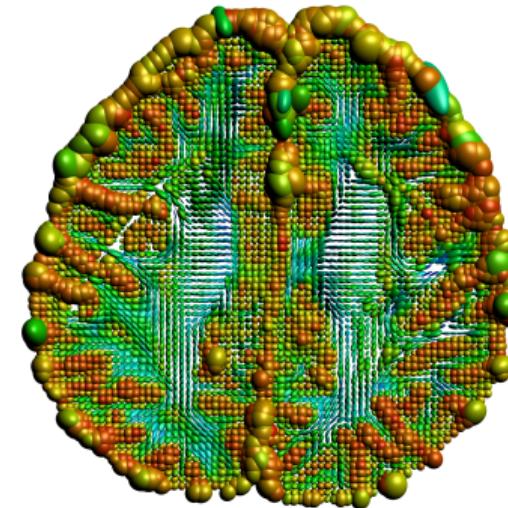
**Tasks.** Denoising, Inpainting, labeling (classification), deblurring,...

# Manifold-valued Signal & Image Processing

Tasks in [Image Processing](#) is phrased as an optimisation problem.

**Here.** The pixel take values on a manifold

- ▶ phase-valued data ( $\mathbb{S}^1$ )
- ▶ wind-fields, GPS ( $\mathbb{S}^2$ )
- ▶ DT-MRI ( $\mathcal{P}(3)$ )
- ▶ EBSD, (grain) orientations ( $\text{SO}(n)$ )



DT-MRI of the human brain.

Camino Project: [cmic.cs.ucl.ac.uk/camino](http://cmic.cs.ucl.ac.uk/camino)

**Tasks.** Denoising, Inpainting, labeling (classification), deblurring,...

# Manifold-valued Signal & Image Processing

Tasks in [Image Processing](#) is phrased as an optimisation problem.  
**Here.** The pixel take values on a manifold

- ▶ phase-valued data ( $\mathbb{S}^1$ )
- ▶ wind-fields, GPS ( $\mathbb{S}^2$ )
- ▶ DT-MRI ( $\mathcal{P}(3)$ )
- ▶ EBSD, (grain) orientations ( $\text{SO}(n)$ )



Grain orientations in EBSD data.

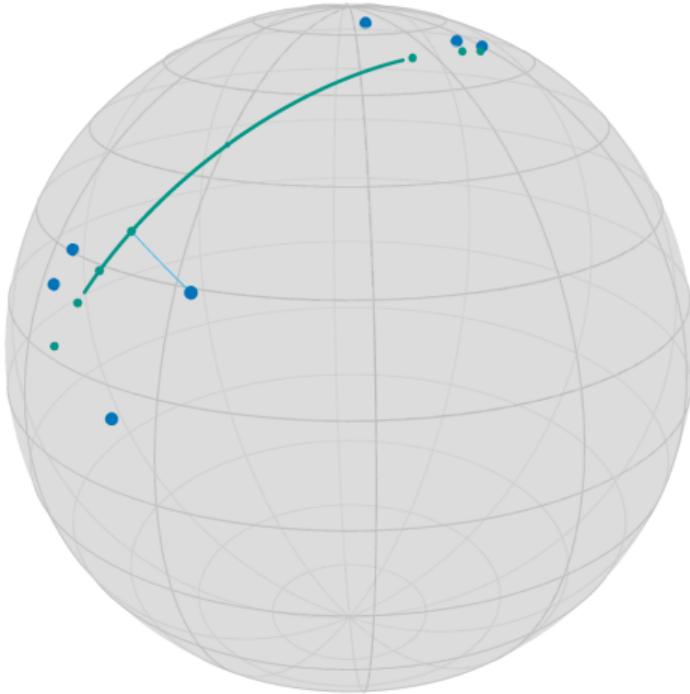
MTEX toolbox: [mTEX toolbox.github.io](https://github.com/mtex-toolbox/mtex-toolbox)

**Tasks.** Denoising, Inpainting, labeling (classification), deblurring,...

# Regression & Interpolation

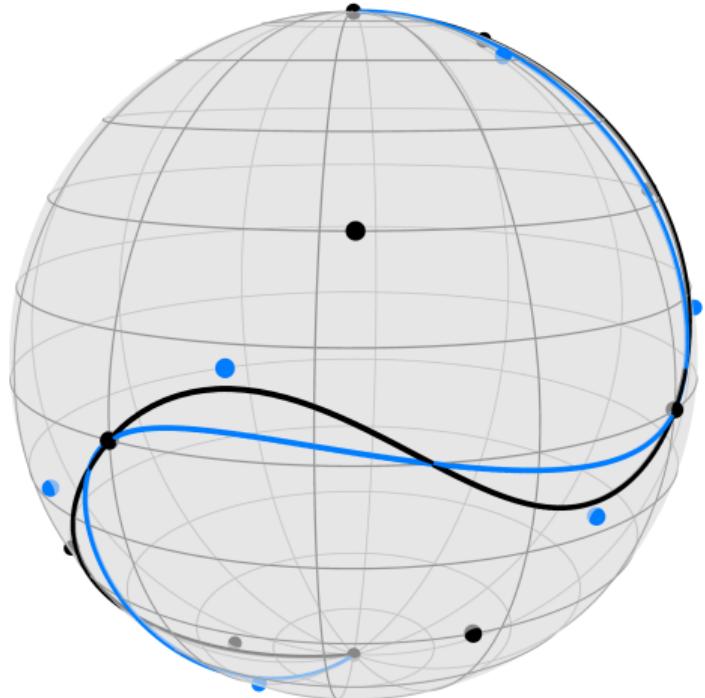


NTNU



**Regression.** Find a geodesic/curve  
“explaining the data best”

[Rentmeesters 2011; Fletcher 2013]



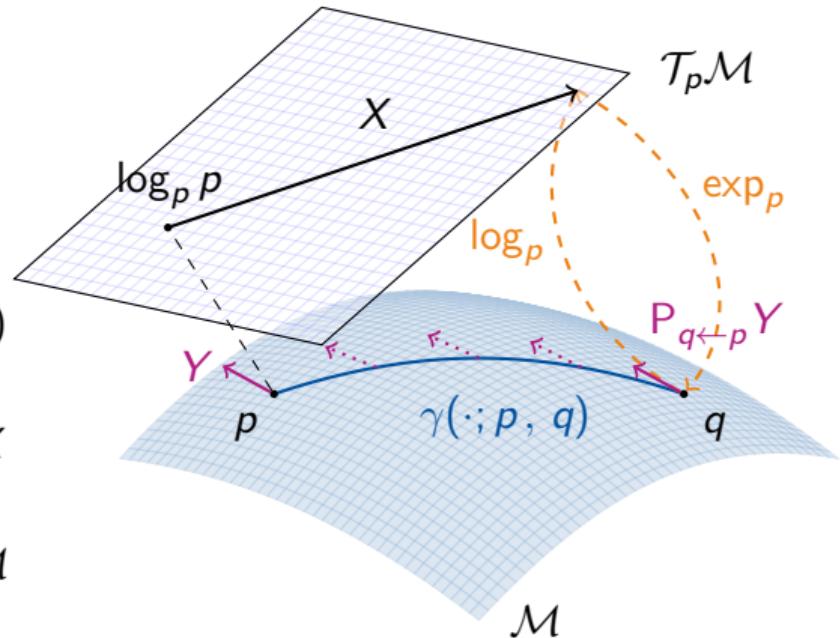
**Interpolation.** Interpolate data with a  
(Bézier) curve of min. acceleration.

[RB and Gousenbourger 2018]

# A $d$ -dimensional Riemannian manifold $\mathcal{M}$

## Notation.

- ▶ Geodesic  $\gamma(\cdot; p, q)$
- ▶ Tangent space  $T_p\mathcal{M}$
- ▶ inner product  $(\cdot, \cdot)_p$
- ▶ Logarithmic map  $\log_p q = \dot{\gamma}(0; p, q)$
- ▶ Exponential map  $\exp_p X = \gamma_{p,X}(1)$   
where  $\gamma_{p,X}(0) = p$  and  $\dot{\gamma}_{p,X}(0) = X$
- ▶ Parallel transport  $P_{q \leftarrow p} Y$  “move”  
tangent vectors from  $T_p\mathcal{M}$  to  $T_q\mathcal{M}$





# The Proximal Map

NTNU

For  $\varphi: \mathcal{M} \rightarrow (-\infty, +\infty]$  and  $\lambda > 0$  the **Proximum** is defined by

[Moreau 1965; Rockafellar 1976; Ferreira and Oliveira 2002]

$$\text{prox}_{\lambda\varphi}(p) := \arg \min_{q \in \mathcal{M}} \frac{1}{2} d_{\mathcal{M}}(q, p)^2 + \lambda\varphi(q).$$

- ▶  $\text{prox}_{\lambda\varphi}$  is **well defined** for convex, lsc.  $\varphi$
- ▶ their proximal map is **non-expansive**
- ▶ a minimizer  $q^*$  of  $\varphi$  is a **fix point** of the prox:  $\text{prox}_{\lambda\varphi}(q^*) = q^*$
- ▶ starting with some  $p_0 \in \mathcal{M}$  the **proximal point algorithm** (PPA)

$$p_{k+1} = \text{prox}_{\lambda_k\varphi}(p_k)$$

converges (weakly) if  $\{\lambda_k\}_k \notin \ell_1(\mathbb{N})$  on Hadamard manifolds.

**But.** computing one step (numerically) might be quite expensive.



# Cyclic Proximal Point Algorithm

NTNU

**Idea.** Split  $f = \sum_{i=1}^N f_i$  and apply the

Cyclic Proximal Point-Algorithmus (CPPA):

[Bertsekas 2011; Bačák 2014]

$$p_{k+\frac{i+1}{N}} = \text{prox}_{\lambda_k f_i}\left(p_{k+\frac{i}{N}}\right), \quad i = 0, \dots, N-1, \quad k = 0, 1, \dots$$

This converges on to a minimizer of  $f$  on a Hadamard manifold  $\mathcal{M}$  if

- ▶ all  $f_i$  proper, convex, lsc.
- ▶  $\{\lambda_k\}_{k \in \mathbb{N}} \in \ell_2(\mathbb{N}) \setminus \ell_1(\mathbb{N})$ .

# Applications of CPPA

The algorithm works well also

- ▶ with inexact/approximate evaluations of the proximal maps  
[Bačák, RB, Steidl, and Weinmann 2016]
- ▶ works numerically on non-Hadamard manifolds
- ▶ a lot of simple proximal maps available in closed form:
  1. distance  $\varphi(p) = \frac{1}{n} d_{\mathcal{M}}(p, q)^n, n \in \{1, 2\}, p \in \mathcal{M}$
  2. finite difference  $\varphi(p) = \frac{1}{n} d_{\mathcal{M}}(p_1, p_2)^n, n \in \{1, 2\}, p \in \mathcal{M}^2$
- ▶ second order difference  $\varphi(p) = d_{2,\mathcal{M}}(p_1, p_2, p_3), p \in \mathcal{M}^3$

**Example.**  $\ell^2$ -TV for a given signal  $f \in \mathcal{M}^N$

$$\arg \min_{p \in \mathcal{M}^N} \frac{1}{2} d_{\mathcal{M}^N}(f, p)^2 + \sum_{i=1}^{N-1} d_{\mathcal{M}}(p_i, p_{i+1})$$



# The Reflection

NTNU

A map  $\mathcal{R}_p$  is called **Reflection** on  $\mathcal{M}$ , if

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

Analogously: **Reflection at the prox** we denote by

$$\mathcal{R}_{\lambda\varphi}(x) = \mathcal{R}_{\text{prox}_{\lambda\varphi}(x)}(x)$$

**Example.** On  $\mathbb{R}^n$  we have  $\mathcal{R}_p(x) = 2p - x = p - (x - p)$ .



# The Douglas-Rachford Algorithm (DRA)

NTNU

**Goal.** Find a minimizer of two proper, convex, lsc. functions

$$\arg \min_{p \in \mathcal{M}} F(p) + G(p)$$

**Iteration:** For  $p_0 \in \mathcal{M}$  compute the Krasnoselskii-Mann-iteration, i.e.,

[RB, Persch, and Steidl 2016]

$$\begin{aligned} q_k &= \mathcal{R}_{\lambda F}(\mathcal{R}_{\lambda G}(p_k)) \\ p_{k+1} &= \gamma(\beta_k; p_k, q_k) \end{aligned}$$

with  $\beta_k \in (0, 1)$  and  $\sum_{k \in \mathbb{N}} \beta_k(1 - \beta_k) = \infty$



# Convergence of the Douglas–Rachford Algorithm

NTNU

## Theorem

[Kakavandi 2013]

Let  $\mathcal{R}_{\lambda F}, \mathcal{R}_{\lambda G}$  be non-expansive and hence  $T = \mathcal{R}_{\lambda F} \circ \mathcal{R}_{\lambda G}$  is nonexpansive. Let  $T$  possess a fix point  $\hat{q}$ .

Then the sequence  $\{q_k\}$  in the DRA converges for every start point  $p_0 \in \mathcal{M}$  to a fix point  $\hat{q}$  of  $T$  (in the  $q_k$ ).

## Theorem

[RB, Persch, and Steidl 2016]

Let  $F, G$  be proper, convex, lsc., let there be a minimizer  $p^*$  of  $F + G$ , and let  $T = \mathcal{R}_{\lambda F} \circ \mathcal{R}_{\lambda G}$  be non-expansive.

Then there exists for every  $p^*$  a fix point  $\hat{q}$  of  $T$ , such that

$$p^* = \text{prox}_{\lambda\psi}(\hat{q})$$

holds. Further, for every  $\hat{q}$ , the point  $\text{prox}_{\lambda\psi}(\hat{q})$  is a minimizer of  $F + G$ .



## Imaging: Parallel (or consensus) Douglas–Rachford.

For a sum  $f = \sum_{i=1}^N f_i$  vectorize the objective

$$G(\mathbf{x}) = \sum_{i=1}^N f_i(x_{\textcolor{blue}{i}}), \quad \mathbf{x} \in \mathcal{M}^N$$

$\Rightarrow \text{prox}_{\lambda G}$  is element-wise easy proxes

And

$$F(\mathbf{x}) = \iota_D(\mathbf{x}), \quad D := \{\mathbf{x} \in \mathcal{M}^N : x_1 = x_2 = \dots = x_N\}$$

$\Rightarrow \text{prox}_{\lambda F}$  is the Riemannian center of mass (mean).

We obtain convergence on Hadamard manifolds of **constant curvature** and numerically works fine on Hadamard manifolds.



# The Riemannian $m$ -Fenchel Conjugate

[RB, Herzog, Silva Louzeiro, Tenbrinck, and Vidal-Núñez 2021]

alternative approaches: [Ahmadi Kakavandi and Amini 2010; RB, Herzog, and Silva Louzeiro 2021]

Let  $m \in \mathcal{C} \subset \mathcal{M}$  be given and  $F: \mathcal{C} \rightarrow \overline{\mathbb{R}}$ .

The  $m$ -Fenchel conjugate  $F_m^*: \mathcal{T}_m^*\mathcal{M} \rightarrow \overline{\mathbb{R}}$  is defined by

$$F_m^*(\xi_m) := \sup_{X \in \mathcal{L}_{\mathcal{C},m}} \{ \langle \xi_m, X \rangle - F(\exp_m X) \},$$

where  $\mathcal{L}_{\mathcal{C},m} := \{X \in \mathcal{T}_m\mathcal{M} \mid q = \exp_m X \in \mathcal{C} \text{ and } \|X\|_p = d(q, p)\}$ .

**A new model.** This can be used to minimize

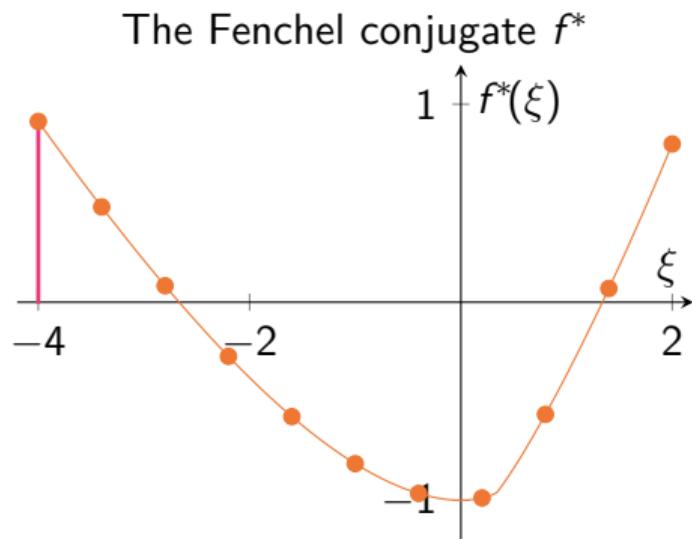
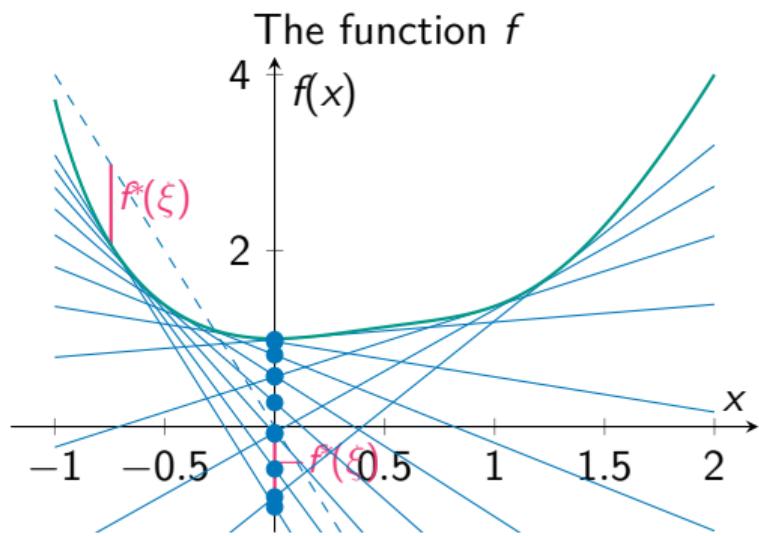
$$\arg \min_{p \in \mathcal{C}} F(p) + G(\Lambda(p))$$

where we have to linearize  $\Lambda: \mathcal{M} \rightarrow \mathcal{N}$  as  $\Lambda(p) \approx \exp_{\Lambda(m)} D\Lambda(m)[\log_m p]$

[Valkonen 2014]

# Illustration of the Fenchel Conjugate

For the case  $\mathcal{M} = \mathbb{R}$  the base point  $m$  is irrelevant, so we set  $m = 0$ .





# The Exact Riemannian Chambolle–Pock Algorithm (eRCPA)

[RB, Herzog, Silva Louzeiro, Tenbrinck, and Vidal-Núñez 2021; Chambolle and Pock 2011]

**Goal.** Minimize  $F(p) + G(\Lambda(p))$  with an arbitrary map  $\Lambda\mathcal{M} \rightarrow \mathcal{N}$ .

**Input:**  $m, p^{(0)} \in \mathbb{R}^d$ ,  $n = \Lambda(m)$ ,  $\xi_n^{(0)} \in \mathbb{R}^d$ , and parameters  $\sigma, \tau, \theta > 0$

1:  $k \leftarrow 0$

2:  $\bar{p}^{(0)} \leftarrow p^{(0)}$

3: **while** not converged **do**

4:  $\xi_n^{(k+1)} \leftarrow \text{prox}_{\tau G_n^*}(\xi_n^{(k)} + \tau (\log_n \Lambda(\bar{p}^{(k)}))^{\flat})$

5:  $p^{(k+1)} \leftarrow \text{prox}_{\sigma F}\left(p^{(k)} + P_{p^{(k)} \leftarrow m}(-\sigma D\Lambda(m)^*[\xi_n^{(k+1)}])^\sharp\right)$

6:  $\bar{p}^{(k+1)} \leftarrow p^{(k+1)} + \theta(p^{(k+1)} - p^{(k)})$

7:  $k \leftarrow k + 1$

8: **end while**

**Output:**  $p^{(k)}$



# Manifolds.jl & Manopt.jl

NTNU

[RB 2022]

The presented algorithms are implemented within the Julia package **Manopt.jl**.  
The Julia package provides general framework to implement optimisation  
algorithms on Manifolds (similar to Manopt, pymanopt)

[Boumal, Mishra, Absil, and Sepulchre 2014; Townsend, Koep, and Weichwald 2016]

The algorithms are implemented using on **ManifoldsBase.jl**, which is an  
interface for manifolds. A corresponding Library of manifolds is provided in  
**Manifolds.jl**.

[Axen, Baran, RB, and Rzecki 2021]

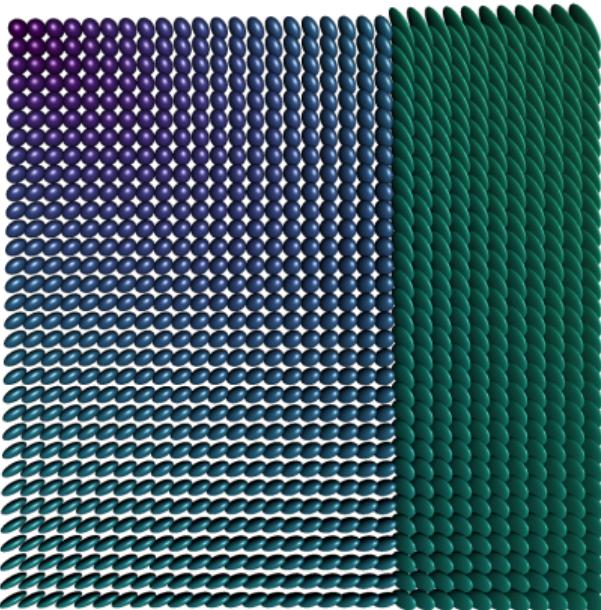
## Motivation.

Provide an efficient, well-tested, well-documented Library of Riemannian  
manifolds.

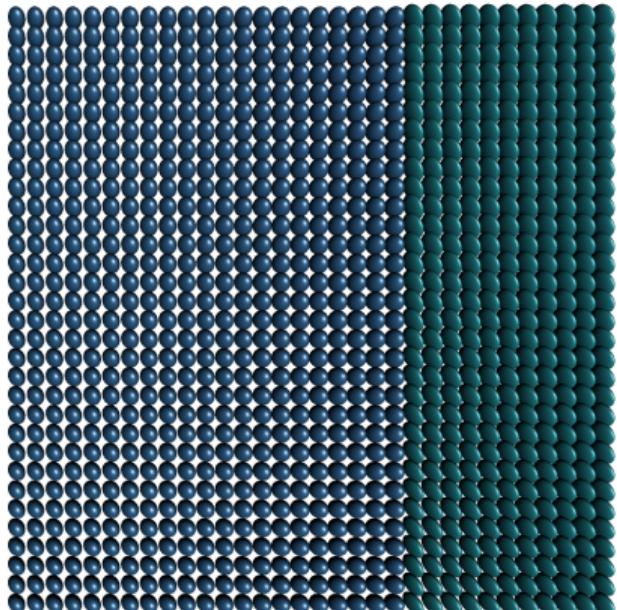
# Numerical Example for a $\mathcal{P}(3)$ -valued Image



NTNU



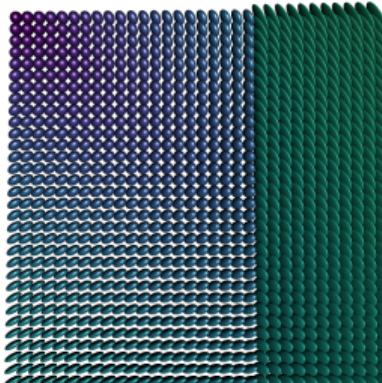
$\mathcal{P}(3)$ -valued data.



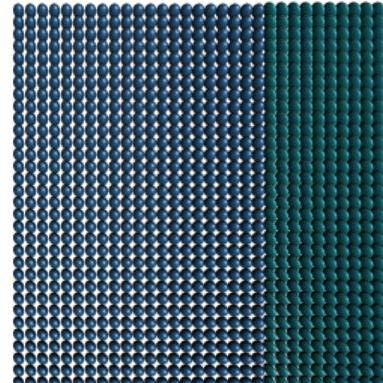
anisotropic TV,  $\alpha = 6$ .

- ▶ in each **pixel** we have a symmetric positive definite matrix
- ▶ Applications: denoising/inpainting e.g. of DT-MRI data

# Numerical Example for a $\mathcal{P}(3)$ -valued Image



$\mathcal{P}(3)$ -valued data.



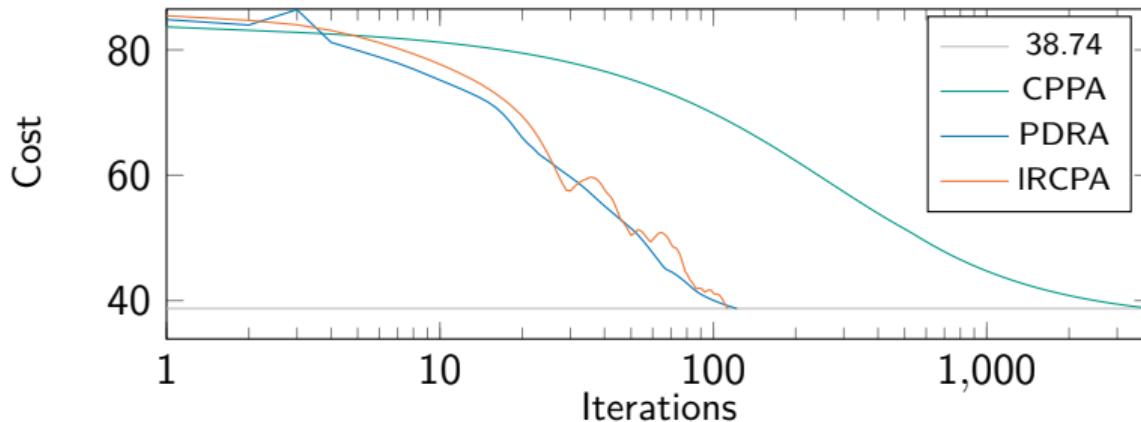
anisotropic TV,  $\alpha = 6$ .

**Approach.** CPPA as benchmark

[Bačák 2014; RB, Herzog, Silva Louzeiro, Tenbrinck, and Vidal-Núñez 2021]

	CPPA	PDRA	IRCPA
<b>parameters</b>	$\lambda_k = \frac{4}{k}$	$\lambda = 0.58$ $\beta = 0.93$	$\sigma = \tau = 0.4$ $\gamma = 0.2, m = l$
<b>iterations</b>	4000	122	<b>113</b>
<b>runtime</b>	1235 s.	380 s.	<b>96.1 s.</b>

# Numerical Example for a $\mathcal{P}(3)$ -valued Image



**Approach.** CPPA as benchmark

[Bacák 2014; RB, Herzog, Silva Louzeiro, Tenbrinck, and Vidal-Núñez 2021]

	CPPA	PDRA	IRCPA
parameters	$\lambda_k = \frac{4}{k}$	$\lambda = 0.58$ $\beta = 0.93$	$\sigma = \tau = 0.4$ $\gamma = 0.2, m = I$
iterations	4000	122	<b>113</b>
runtime	1235 s.	380 s.	<b>96.1 s.</b>

# Selected References

-  Axen, S. D., M. Baran, RB, and K. Rzecki (2021). *Manifolds.jl: An Extensible Julia Framework for Data Analysis on Manifolds*. arXiv: 2106.08777.
-  Baćák, M. (2014). "Computing medians and means in Hadamard spaces". In: *SIAM Journal on Optimization* 24.3, pp. 1542–1566. DOI: 10.1137/140953393.
-  RB (2022). "Manopt.jl: Optimization on Manifolds in Julia". In: *Journal of Open Source Software* 7.70, p. 3866. DOI: 10.21105/joss.03866.
-  RB, R. Herzog, and M. Silva Louzeiro (2021). *Fenchel duality and a separation theorem on Hadamard manifolds*. arXiv: 2102.11155.
-  RB, R. Herzog, M. Silva Louzeiro, D. Tenbrinck, and J. Vidal-Núñez (Jan. 2021). "Fenchel duality theory and a primal-dual algorithm on Riemannian manifolds". In: *Foundations of Computational Mathematics*. DOI: 10.1007/s10208-020-09486-5. arXiv: 1908.02022.
-  RB, J. Persch, and G. Steidl (2016). "A parallel Douglas Rachford algorithm for minimizing ROF-like functionals on images with values in symmetric Hadamard manifolds". In: *SIAM Journal on Imaging Sciences* 9.4, pp. 901–937. DOI: 10.1137/15M1052858.
-  Chambolle, A. and T. Pock (2011). "A first-order primal-dual algorithm for convex problems with applications to imaging". In: *Journal of Mathematical Imaging and Vision* 40.1, pp. 120–145. DOI: 10.1007/s10851-010-0251-1.

