

# Riemannian Optimization and a Riemannian Proximal Newton-CG Method

Speaker: Wen Huang

Xiamen University

October 31, 2024

东莞理工学院

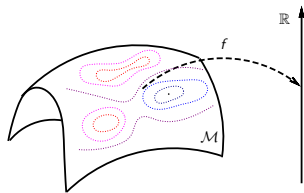
- Riemannian optimization;
  - Problem statement;
  - Applications;
  - Smooth optimization framework;
  - Research foci of Riemannian optimization;
- A Riemannian proximal Newton-CG method;
  - Optimization with a structure;
  - Proximal gradient-type methods;
  - A Riemannian proximal Newton method;
  - Globalization by truncated CG;
  - Numerical experiments;
- Summary;

# Riemannian Optimization

**Problem:** Given  $f(x) : \mathcal{M} \rightarrow \mathbb{R}$ ,  
solve

$$\min_{x \in \mathcal{M}} f(x)$$

where  $\mathcal{M}$  is a Riemannian manifold.

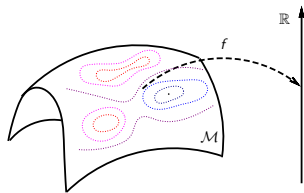


# Riemannian Optimization

**Problem:** Given  $f(x) : \mathcal{M} \rightarrow \mathbb{R}$ ,  
solve

$$\min_{x \in \mathcal{M}} f(x)$$

where  $\mathcal{M}$  is a Riemannian manifold.



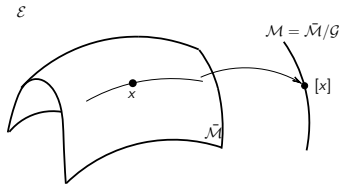
---

## Two kinds of commonly-encountered manifolds

Embedded submanifold of a Euclidean space



Quotient manifold from an embedded submanifold

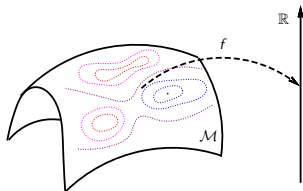


# Riemannian Optimization

**Problem:** Given  $f(x) : \mathcal{M} \rightarrow \mathbb{R}$ ,  
solve

$$\min_{x \in \mathcal{M}} f(x)$$

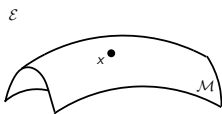
where  $\mathcal{M}$  is a Riemannian manifold.



## Examples:

- Sphere:  $\{x \in \mathbb{R}^n \mid \|x\| = 1\}$ ;
- Stiefel manifold:  
 $\text{St}(p, n) = \{X \in \mathbb{R}^{n \times p} \mid X^T X = I_p\}$ ;
- Fixed rank:  
 $\mathbb{R}_r^{m \times n} = \{X \in \mathbb{R}^{m \times n} : \text{rank}(X) = r\}$ ;
- etc;

Embedded submanifold of a Euclidean space

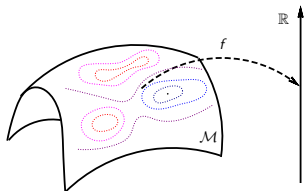


# Riemannian Optimization

**Problem:** Given  $f(x) : \mathcal{M} \rightarrow \mathbb{R}$ ,  
solve

$$\min_{x \in \mathcal{M}} f(x)$$

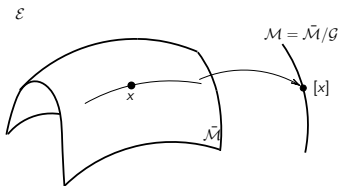
where  $\mathcal{M}$  is a Riemannian manifold.



## Examples:

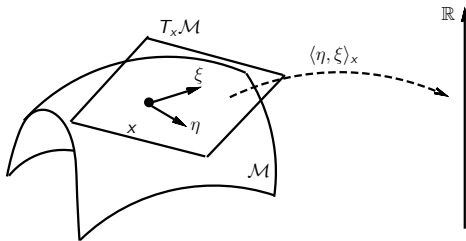
- Grassmann manifold:  
the set of  $p$  dimensional linear  
spaces in  $\mathbb{R}^n$   
 $\text{Gr}(p, n) = \text{St}(p, n) / \mathcal{O}_p$
- Shape space;
- etc;

Quotient manifold from an embedded submanifold



# Riemannian Optimization

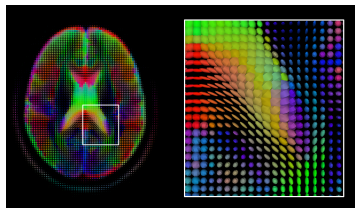
Roughly, a Riemannian manifold  $\mathcal{M}$  is a smooth set with a smoothly-varying inner product on the tangent spaces.



Riemannian manifold = Manifold + Riemannian metric (inner products)

## Embedded submanifold: Computation on SPD manifold

- SPD manifold:  
 $\mathcal{S}_{++}^n = \{X \in \mathbb{R}^{n \times n} : X = X^T, X \succ 0\};$
- Applications of SPD matrices
  - Diffusion tensors in medical imaging [CSV12, FJ07, RTM07]
  - Describing images and video [LWM13, SFD02, ASF<sup>+</sup>05, TPM06, HWSC15]
- Motivation of averaging SPD matrices
  - denoising / interpolation
  - clustering / classification





## Embedded submanifold: Computation on SPD manifold

One averaging SPD matrices method:

$$G(A_1, \dots, A_k) = \arg \min_{X \in \mathcal{S}_{++}^n} \frac{1}{2k} \sum_{i=1}^k \text{dist}^2(X, A_i),$$

where  $\text{dist}(X, Y) = \|\log(X^{-1/2} Y X^{-1/2})\|_F$  is the distance under the Riemannian metric  $\langle \eta_X, \xi_X \rangle_X = \text{trace}(\eta_X X^{-1} \xi_X X^{-1})$ .

---

## Embedded submanifold: Computation on SPD manifold

One averaging SPD matrices method:

$$G(A_1, \dots, A_k) = \arg \min_{X \in \mathcal{S}_{++}^n} \frac{1}{2k} \sum_{i=1}^k \text{dist}^2(X, A_i),$$

where  $\text{dist}(X, Y) = \|\log(X^{-1/2} Y X^{-1/2})\|_F$  is the distance under the Riemannian metric  $\langle \eta_X, \xi_X \rangle_X = \text{trace}(\eta_X X^{-1} \xi_X X^{-1})$ .

---

### Why shall we use Riemannian optimization approach?

Metric:  $\langle \eta_X, \xi_X \rangle_X = \text{trace}(\eta_X X^{-1} \xi_X X^{-1})$

Metric:  $\langle \eta, \xi \rangle_X = \text{trace}(\eta^T \xi)$

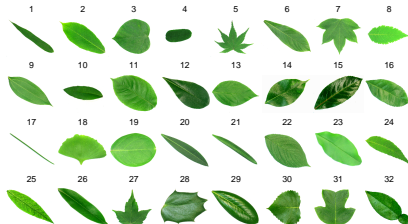
### Condition number of the Riemannian Hessian [YHAG2020]

- $\kappa(H^R) \leq 1 + \frac{\ln(\max \kappa_i)}{2}$ , where  $\kappa_i = \kappa(\mu^{-1/2} A_i \mu^{-1/2})$
- $\kappa(H^R) \leq 20$  if  $\max(\kappa_i) = 10^{16}$
- $\frac{\kappa^2(\mu)}{\kappa(H^R)} \leq \kappa(H^E) \leq \kappa(H^R) \kappa^2(\mu)$
- $\kappa(H^E) \geq \kappa^2(\mu)/20$

---

[YHAG2020]: X. Yuan, W. Huang\*, P.-A. Absil, K. A. Gallivan. "Computing the matrix geometric mean: Riemannian vs Euclidean conditioning, implementation techniques, and a Riemannian BFGS method", *Numerical Linear Algebra with Applications*, 27:5, 1-23, 2020.

## Quotient manifold: Computation on shape space



- Classification  
[LKS<sup>+</sup>12, HGSA15]
- Face recognition  
[DBS<sup>+</sup>13]



## Quotient manifold: Computation on shape space

- Elastic shape analysis invariants:
  - Rescaling
  - Translation
  - Rotation
  - Reparametrization
- The shape space is a quotient space

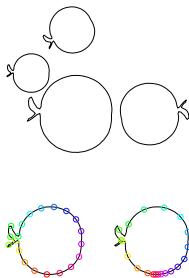
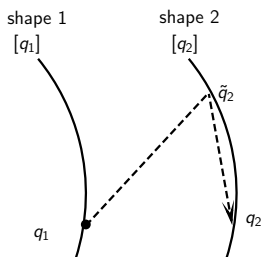


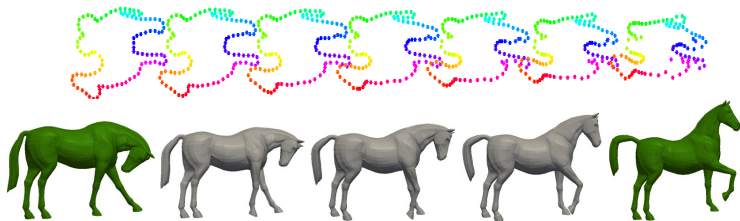
Figure: All are the same shape.

## Quotient manifold: Computation on shape space Registration



- Optimization problem  $\min_{q_2 \in [q_2]} \text{dist}(q_1, q_2)$  is defined on a Riemannian manifold

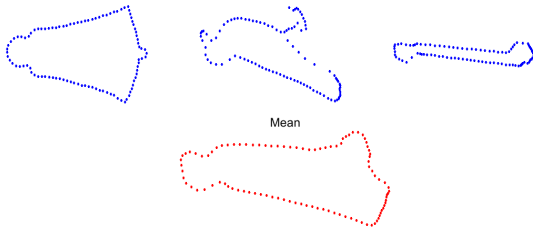
## Quotient manifold: Computation on shape space Geodesic / Interpolation



$$\min_{\alpha \in \mathcal{H}_{x,y}} \frac{1}{2} \int_0^1 \langle \dot{\alpha}(\tau), \dot{\alpha}(\tau) \rangle_{\alpha(\tau)} d\tau$$

- Computation of a geodesic between two shapes
- Interpolation in shape space

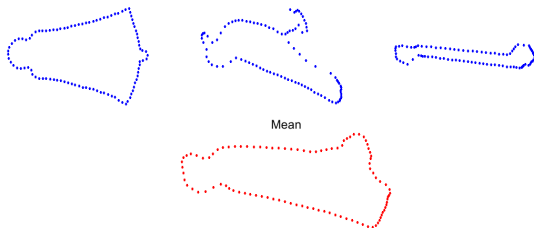
## Quotient manifold: Computation on shape space Karcher mean



$$\min_{x \text{ is a shape}} \frac{1}{2k} \sum_{i=1}^k \text{dist}^2(X, S_i),$$

- Computation of Karcher mean of a population of shapes

## Quotient manifold: Computation on shape space Karcher mean



$$\min_{x \text{ is a shape}} \frac{1}{2k} \sum_{i=1}^k \text{dist}^2(X, S_i),$$

- Computation of Karcher mean of a population of shapes

---

**Riemannian optimization is used since these problems naturally involve a Riemannian manifold**



# Smooth Optimization Framework

## Iterations on the Manifold

Consider the following generic update for an iterative Euclidean optimization algorithm:

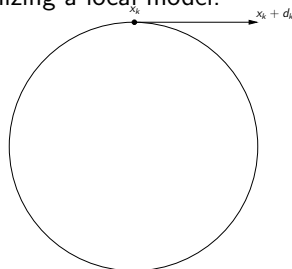
$$x_{k+1} = x_k + \Delta x_k = x_k + \alpha_k s_k .$$

This iteration is implemented in numerous ways, e.g.:

- Steepest descent:  $x_{k+1} = x_k - \alpha_k \nabla f(x_k)$
- Newton's method:  $x_{k+1} = x_k - [\nabla^2 f(x_k)]^{-1} \nabla f(x_k)$
- Trust region method:  $\Delta x_k$  is set by optimizing a local model.

### Riemannian Manifolds Provide

- Riemannian concepts describing **directions** and **movement** on the manifold
- Riemannian analogues for **gradient** and **Hessian**



# Smooth Optimization Framework

Riemannian gradient and Riemannian Hessian

## Definition

The **Riemannian gradient** of  $f$  at  $x$  is the unique tangent vector in  $T_x \mathcal{M}$  satisfying  $\forall \eta \in T_x \mathcal{M}$ , the directional derivative

$$Df(x)[\eta] = \langle \text{grad } f(x), \eta \rangle$$

and  $\text{grad } f(x)$  is the direction of steepest ascent.

## Definition

The **Riemannian Hessian** of  $f$  at  $x$  is a symmetric linear operator from  $T_x \mathcal{M}$  to  $T_x \mathcal{M}$  defined as

$$\text{Hess } f(x) : T_x \mathcal{M} \rightarrow T_x \mathcal{M} : \eta \rightarrow \nabla_\eta \text{grad } f,$$

where  $\nabla$  is the affine connection.

# Smooth Optimization Framework

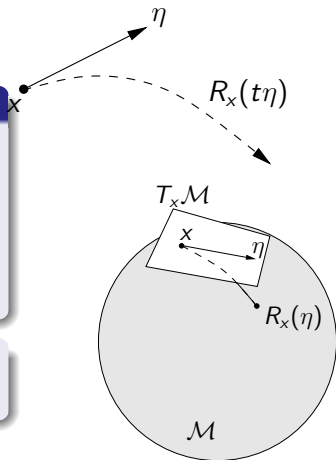
## Retractions

Euclidean	Riemannian
$x_{k+1} = x_k + \alpha_k d_k$	$x_{k+1} = R_{x_k}(\alpha_k \eta_k)$

### Definition

A **retraction** is a mapping  $R$  from  $T\mathcal{M}$  to  $\mathcal{M}$  satisfying the following:

- $R$  is continuously differentiable
  - $R_x(0) = x$
  - $D R_x(0)[\eta] = \eta$
- 
- maps tangent vectors back to the manifold
  - defines curves in a direction



# Smooth Optimization Framework

Categories of Riemannian smooth optimization methods

Retraction-based: local information only

Line search-based: use local tangent vector and  $R_x(t\eta)$  to define line

- Steepest decent
- Newton

Local model-based: series of flat space problems

- Riemannian trust region Newton (RTR)
- Riemannian adaptive cubic overestimation (RACO)

# Smooth Optimization Framework

Categories of Riemannian smooth optimization methods

## Retraction and transport-based: information from multiple tangent spaces

- Nonlinear conjugate gradient: multiple tangent vectors
- Quasi-Newton e.g. Riemannian BFGS: transport operators between tangent spaces

Additional element required for optimizing a cost function;

- formulas for combining information from multiple tangent spaces.

# Smooth Optimization Framework

Categories of Riemannian smooth optimization methods

## Retraction and transport-based: information from multiple tangent spaces

- Nonlinear conjugate gradient: multiple tangent vectors
- Quasi-Newton e.g. Riemannian BFGS: transport operators between tangent spaces

Additional element required for optimizing a cost function;

- formulas for combining information from multiple tangent spaces.

## Vector Transport:

- Vector transport: Transport a tangent vector from one tangent space to another;
- $\mathcal{T}_{\eta_x} \xi_x$ , denotes transport of  $\xi_x$  to tangent space of  $R_x(\eta_x)$ .  $R$  is a retraction associated with  $\mathcal{T}$ ;

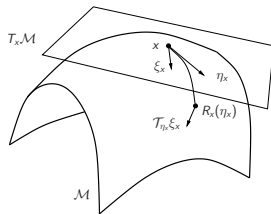


Figure: Vector transport.

# Smooth Optimization Framework

Retraction/Transport-based Riemannian optimization

Given a retraction and a vector transport, we can generalize classical unconstrained smooth optimization methods from Euclidean space to the Riemannian manifold.

# Smooth Optimization Framework

Retraction/Transport-based Riemannian optimization

Given a retraction and a vector transport, we can generalize classical unconstrained smooth optimization methods from Euclidean space to the Riemannian manifold.

---

Do the Riemannian versions of those methods work well?



# Smooth Optimization Framework

Retraction/Transport-based Riemannian optimization

Given a retraction and a vector transport, we can generalize classical unconstrained smooth optimization methods from Euclidean space to the Riemannian manifold.

---

Do the Riemannian versions of those methods work well?

No, generally

- Lose many theoretical results and important properties;
- Impose restrictions on retraction/vector transport;

# Research Foci of Riemannian Optimization

- ① Manifold recognition, geometry structure analyses and computations;
  - ② Generalization Euclidean algorithms to the Riemannian setting;
  - ③ Algorithms specialization for applications;
  - ④ Library developments;
-

# Research Foci of Riemannian Optimization

- 1 Manifold recognition, geometry structure analyses and computations;
  - 2 Generalization Euclidean algorithms to the Riemannian setting;
  - 3 Algorithms specialization for applications;
  - 4 Library developments;
- 

- Manifold recognition
- Riemannian metric
- Retraction / Geodesic
- Vector transport / Parallel translation

---

[EAS1998] A. Edelman, T. A. Arias, and S. T. Smith. The geometry of algorithms with orthogonality constraints. *SIAM Journal on Matrix Analysis and Applications*, 20(2):303–353, 1998

[CMV2017] T. Carson, D. G. Mixon, and S. Villar. Manifold optimization for k-means clustering. In *2017 International Conference on Sampling Theory and Applications (SampTA)*, 73–77. IEEE, 2017

[SDN2021] G. Song, W. Ding, and M. K. Ng. Low rank pure quaternion approximation for pure quaternion matrices, *SIAM Journal on Matrix Analysis and Applications*, 42, pp. 58–82, 2021

[VAV2013] B. Vandereycken, P.-A. Absil, and S. Vandewalle. A Riemannian geometry with complete geodesics for the set of positive semidefinite matrices of fixed rank, *IMA Journal of Numerical Analysis*, 33.2, 481–514, 2013.

[Zim2017] R. Zimmermann. A matrix-algebraic algorithm for the Riemannian logarithm on the Stiefel manifold under the canonical metric. *SIAM Journal on Matrix Analysis and Applications*, 38.2, 322–342, 2017.

# Research Foci of Riemannian Optimization

- 1 Manifold recognition, geometry structure analyses and computations;
  - 2 Generalization Euclidean algorithms to the Riemannian setting;
  - 3 Algorithms specialization for applications;
  - 4 Library developments;
- 

- Smooth unconstrained optimization algorithms
- Nonsmooth unconstrained optimization algorithms
- Constrained optimization algorithms

# Research Foci of Riemannian Optimization

- 1 Manifold recognition, geometry structure analyses and computations;
  - 2 Generalization Euclidean algorithms to the Riemannian setting;
  - 3 Algorithms specialization for applications;
  - 4 Library developments;
- 

- Smooth unconstrained optimization algorithms
- Nonsmooth unconstrained optimization algorithms
- Constrained optimization algorithms

Riemannian optimization mainly focuses on this topic.  
Discuss later.

# Research Foci of Riemannian Optimization

- 1 Manifold recognition, geometry structure analyses and computations;
  - 2 Generalization Euclidean algorithms to the Riemannian setting;
  - 3 Algorithms specialization for applications;
  - 4 Library developments;
- 

- Computations on the SPD manifold;
- Computations on the shape space;
- Clustering and graph partitions;
- Beamforming in wireless communication;
- Blind source separation;
- etc

# Research Foci of Riemannian Optimization

- 1 Manifold recognition, geometry structure analyses and computations;
  - 2 Generalization Euclidean algorithms to the Riemannian setting;
  - 3 Algorithms specialization for applications;
  - 4 Library developments;
- 

- Representation of a manifold and tangent spaces;
- Choose a Riemannian metric;
- Choose a retraction;
- Choose a vector transport;

# Research Foci of Riemannian Optimization

- 1 Manifold recognition, geometry structure analyses and computations;
  - 2 Generalization Euclidean algorithms to the Riemannian setting;
  - 3 Algorithms specialization for applications;
  - 4 Library developments;
- 

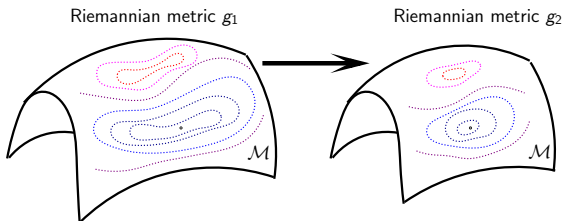
- Representation of a manifold and tangent spaces;
- Choose a Riemannian metric;
- Choose a retraction;
- Choose a vector transport;

Above factors may influence algorithms significantly.



# Research Foci of Riemannian Optimization

- 1 Manifold recognition, geometry structure analyses and computations;
  - 2 Generalization Euclidean algorithms to the Riemannian setting;
  - 3 Algorithms specialization for applications;
  - 4 Library developments;
- 



**Figure:** Changing Riemannian metric may influence the difficulty of a problem.

# Research Foci of Riemannian Optimization

- 1 Manifold recognition, geometry structure analyses and computations;
  - 2 Generalization Euclidean algorithms to the Riemannian setting;
  - 3 Algorithms specialization for applications;
  - 4 Library developments;
- 

- Manopt (Matlab library) [Boumal, Mishra, Absil, Sepulchre(2014)]
- Pymanopt (Python version of Manopt) [Townsend, Koep, Weichwald (2016)]
- Manoptjl (Julia, nonsmooth methods) [Bergmann (2019)]
- ROPTLIB (C++ library, interfaces to Matlab and Julia)  
[Huang, Absil, Gallivan, Hand (2018)]
- ManifoldOptim (R wrapper of ROPTLIB) [Martin, Raim, Huang, Adraghi (2018)]
- McTorch (Python, GPU acceleration)  
[Meghawanshi, Jawanpuria, Kunchukuttan, Kasai, Mishra (2018)]
- CDOpt (Python, embedded submanifold in the form of  $c(x) = 0$ )  
[Xiao, Hu, Liu, Toh (2022)]

# Research Foci of Riemannian Optimization

- 1 Manifold recognition, geometry structure analyses and computations;
  - 2 Generalization Euclidean algorithms to the Riemannian setting;
  - 3 Algorithms specialization for applications;
  - 4 Library developments;
- 

## Provide theories to explain behaviors of existing algorithms for particular applications

- [MBDG2023]: IRKA is a Riemannian gradient descent method;
- [YHAG2020]: Richardson-like iteration for matrix geometric mean is a Riemannian gradient descent method;
- [BM2006]: The improved BFGS method is a Riemannian BFGS method using vector transport by parallelization;

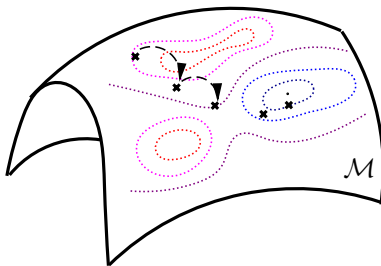
---

[MBDG2023] P. Mlinaric, C. Beattie, Z. Drmac, and S. Gugercin. IRKA is a Riemannian Gradient Descent Method. arxiv:2311.02031, 2023  
[YHAG2020] X. Yuan, W. Huang, P.-A. Absil, K. A. Gallivan. Computing the matrix geometric mean: Riemannian vs Euclidean conditioning, implementation techniques, and a Riemannian BFGS method, *Numerical Linear Algebra with Applications*, 27:5, 1-23, 2020  
[BM2006] I. Brace and J. H. Manton. An improved BFGS-on-manifold algorithm for computing weighted low rank approximations. *Proceedings of 17th international Symposium on Mathematical Theory of Networks and Systems*, P.1735–1738, 2006

# Comparison with Constrained Optimization

**Not all Riemannian optimization problem can be formulated as constrained optimization problems, and vice versa.**

- All iterates on the manifold
- Convergence properties of unconstrained optimization algorithms
- No need to consider Lagrange multipliers or penalty functions
- Exploit the structure of the constrained set



# A Non-exhaustive Review

- Smooth unconstrained problems
  - Steepest descent: Smith 1994; Helmke-Moore 1994; Iannazzo-Porcelli 2019;
  - Conjugate gradient: Smith 1994; Gallivan-Absil 2010; Ring-Wirth 2012; Sato-Iwai 2015;
  - Quasi-Newton: Ring-Wirth 2012; Huang-Absil-Gallivan 2018; Huang-Gallivan 2022
  - Newton-CG: Absil-Baker-Gallivan 2007; Huang-Huang 2023
- Nonsmooth unconstrained problems
  - Proximal point method: Ferreira-Oliveira 2002;
  - Optimality conditions: Yang-Zhang-Song 2014;
  - Gradient sampling: Huang 2013; Hosseini and Uschmajew 2017;
  - $\epsilon$ -subgradient-based methods: Grohs-Hosseini 2015;
  - Proximal gradient methods: Huang-Wei 2022;
  - Proximal Newton method: Si-Absil-Huang-Jiang-Vary 2023;
- Constrained problems:
  - Augmented Lagrangian methods: Boumal-Liu 2019;
  - Sequential quadratic programming: Obara-Okuno-Takeda 2022;
  - Frank-Wolfe Methods: Weber-Sra 2023;

# A Non-exhaustive Review

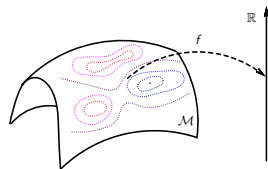
- Smooth unconstrained problems:
  - Stiefel manifold: Wen-Yin 2012; Jiang-Dai 2014; Xiao-Liu-Yuan 2020; Dai-Wang-Zhou 2020
  - Symplectic Stiefel manifold: Gao-Son-Absil-Stykel 2021
  - Symmetric positive definite manifold: Bini-Iannazzo 2013; Zhang 2017; Yuan-Huang-Absil-Gallivan 2020;
  - Fixed rank manifold: Wen-Yin-Zhang 2012; Mishra 2014; Sutti-Vandereycken 2021; Levin-Kileel-Boumal 2022
- Nonsmooth unconstrained problems:
  - Stiefel Manifold: Huang-Wei 2019; Chen-Ma-So-Zhang 2020; Xiao-Liu-Yuan 2020;
  - Fixed rank manifold: Cambier-Absil 2016;
  - Matrix manifolds: Zhou-Bao-Ding-Zhu 2022
  - Smooth equation constraints: Xiao-Liu-Toh 2023
- Constrained problems:
  - Stiefel + non-negativity: Jiang-Meng-Wen-Chen 2019;
  - Symmetric positive definite + zeros: Phan-Menickelly 2020;

# A Riemannian Proximal Newton-CG Method

## Problem statement

### Optimization on Manifolds with Structure:

$$\min_{x \in \mathcal{M}} F(x) = f(x) + h(x),$$



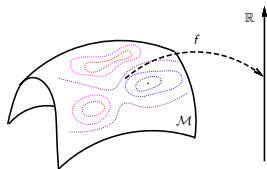
- $\mathcal{M}$  is a finite-dimensional Riemannian manifold;
- $f$  is smooth and may be nonconvex; and
- $h(x)$  is continuous and convex but may be nonsmooth;

# A Riemannian Proximal Newton-CG Method

## Problem statement

### Optimization on Manifolds with Structure:

$$\min_{x \in \mathcal{M}} F(x) = f(x) + h(x),$$



- $\mathcal{M}$  is a finite-dimensional Riemannian manifold;
- $f$  is smooth and may be nonconvex; and
- $h(x)$  is continuous and convex but may be nonsmooth;

**Applications:** sparse PCA [ZHT06], compressed modes [OLCO13], sparse partial least squares regression [CSG<sup>+</sup>18], sparse inverse covariance estimation [BESS19], sparse blind deconvolution [ZLK<sup>+</sup>17], and clustering [HWGVD22].



# A Riemannian Proximal Newton-CG Method

Euclidean proximal gradient/Newton method

**Optimization with Structure:**  $\mathcal{M} = \mathbb{R}^n$

$$\min_{x \in \mathbb{R}^n} F(x) = f(x) + h(x),$$

---

Given  $x_0$ ,

$$\begin{cases} d_k = \arg \min_{p \in \mathbb{R}^n} f(x_k) + \langle \nabla f(x_k), p \rangle + \frac{1}{2} \langle p, H_k p \rangle + h(x_k + p) \\ x_{k+1} = x_k + d_k. \end{cases}$$

# A Riemannian Proximal Newton-CG Method

Euclidean proximal gradient/Newton method

**Optimization with Structure:**  $\mathcal{M} = \mathbb{R}^n$

$$\min_{x \in \mathbb{R}^n} F(x) = f(x) + h(x),$$

---

Given  $x_0$ ,

$$\begin{cases} d_k = \arg \min_{p \in \mathbb{R}^n} f(x_k) + \langle \nabla f(x_k), p \rangle + \frac{1}{2} \langle p, H_k p \rangle + h(x_k + p) \\ x_{k+1} = x_k + d_k. \end{cases}$$

**proximal gradient:**  $H_k = L I_n$

- $h \equiv 0 \Rightarrow$  Steepest descent;
- Linear convergence;

**proximal Newton:**  $H_k = \nabla^2 f(x_k)$

- $h \equiv 0 \Rightarrow$  Newton;
- Superlinear convergence;

# A Riemannian Proximal Newton-CG Method

Euclidean proximal gradient/Newton method

**Optimization with Structure:**  $\mathcal{M} = \mathbb{R}^n$

$$\min_{x \in \mathbb{R}^n} F(x) = f(x) + h(x),$$

---

Given  $x_0$ ,

$$\begin{cases} d_k = \arg \min_{p \in \mathbb{R}^n} f(x_k) + \langle \nabla f(x_k), p \rangle + \frac{1}{2} \langle p, H_k p \rangle + h(x_k + p) \\ x_{k+1} = x_k + d_k. \end{cases}$$

**proximal gradient:**  $H_k = L I_n$

- $h \equiv 0 \Rightarrow$  Steepest descent;
- Linear convergence;

**proximal Newton:**  $H_k = \nabla^2 f(x_k)$

- $h \equiv 0 \Rightarrow$  Newton;
- Superlinear convergence;

How to generalize to the Riemannian setting?

# A Riemannian Proximal Newton-CG Method

Generalizations of proximal gradient method

## Euclidean Proximal gradient:

Given  $x_0$ ,

$$\begin{cases} d_k = \arg \min_{p \in \mathbb{R}^n} f(x_k) + \langle \nabla f(x_k), p \rangle + \frac{L}{2} \langle p, p \rangle + h(x_k + p) \\ x_{k+1} = x_k + d_k. \end{cases}$$

---

## Riemannian generalization 1: (for embedded submanifold)

$$\left. \begin{array}{l} \nabla f(x_k) \implies \text{grad } f(x_k) \\ x_{k+1} = x_k + d_k \implies x_{k+1} = R_{x_k}(d_k) \\ p \in \mathbb{R}^n \implies p \in T_{x_k} \mathcal{M} \end{array} \right\} \implies \text{Converge globally}$$

$$\begin{cases} d_k = \arg \min_{p \in T_{x_k} \mathcal{M}} f(x_k) + \langle \text{grad } f(x_k), p \rangle + \frac{L}{2} \langle p, p \rangle + h(x_k + p) \\ x_{k+1} = R_{x_k}(d_k). \end{cases}$$

# A Riemannian Proximal Newton-CG Method

Generalizations of proximal gradient method

## Euclidean Proximal gradient:

Given  $x_0$ ,

$$\begin{cases} d_k = \arg \min_{p \in \mathbb{R}^n} f(x_k) + \langle \nabla f(x_k), p \rangle + \frac{L}{2} \langle p, p \rangle + h(x_k + p) \\ x_{k+1} = x_k + d_k. \end{cases}$$

---

## Riemannian generalization 2: (for general manifold)

$$\left. \begin{aligned} \nabla f(x_k) &\implies \text{grad } f(x_k) \\ x_{k+1} = x_k + d_k &\implies x_{k+1} = R_{x_k}(d_k) \\ p \in \mathbb{R}^n &\implies p \in T_{x_k} \mathcal{M} \\ h(x_k + p) &\implies h(R_{x_k}(p)) \end{aligned} \right\} \implies \begin{aligned} &\text{Converge globally} \\ &\text{Convergence rate analyses} \end{aligned}$$

$$\begin{cases} d_k = \arg \min_{p \in T_{x_k} \mathcal{M}} f(x_k) + \langle \text{grad } f(x_k), p \rangle + \frac{L}{2} \langle p, p \rangle + h(R_{x_k}(p)) \\ x_{k+1} = R_{x_k}(d_k). \end{cases}$$

# A Riemannian Proximal Newton-CG Method

A native generalization

## Euclidean proximal Newton:

$$\begin{cases} d_k = \operatorname{argmin}_{p \in \mathbb{R}^n} f(x_k) + \langle \nabla f(x_k), p \rangle + \frac{1}{2} \langle p, \nabla^2 f(x_k) p \rangle + h(x_k + p) \\ x_{k+1} = x_k + d_k \end{cases}$$

---

A native generalization by replacing the Euclidean gradient and Hessian by the Riemannian gradient and Hessian:

$$\begin{cases} \eta_k = \arg \min_{\eta \in T_{x_k}} \mathcal{M} f(x_k) + \langle \operatorname{grad} f(x_k), \eta \rangle + \frac{1}{2} \langle \eta, \operatorname{Hess} f(x_k) \eta \rangle + h(x_k + \eta) \\ x_{k+1} = R_{x_k}(\eta_k) \end{cases}$$

# A Riemannian Proximal Newton-CG Method

A native generalization

## Euclidean proximal Newton:

$$\begin{cases} d_k = \operatorname{argmin}_{p \in \mathbb{R}^n} f(x_k) + \langle \nabla f(x_k), p \rangle + \frac{1}{2} \langle p, \nabla^2 f(x_k) p \rangle + h(x_k + p) \\ x_{k+1} = x_k + d_k \end{cases}$$

---

A native generalization by replacing the Euclidean gradient and Hessian by the Riemannian gradient and Hessian:

$$\begin{cases} \eta_k = \arg \min_{\eta \in T_{x_k} \mathcal{M}} f(x_k) + \langle \operatorname{grad} f(x_k), \eta \rangle + \frac{1}{2} \langle \eta, \operatorname{Hess} f(x_k) \eta \rangle + h(x_k + \eta) \\ x_{k+1} = R_{x_k}(\eta_k) \end{cases}$$

Does it converge superlinearly locally?

# A Riemannian Proximal Newton-CG Method

A native generalization

## Euclidean proximal Newton:

$$\begin{cases} d_k = \operatorname{argmin}_{p \in \mathbb{R}^n} f(x_k) + \langle \nabla f(x_k), p \rangle + \frac{1}{2} \langle p, \nabla^2 f(x_k) p \rangle + h(x_k + p) \\ x_{k+1} = x_k + d_k \end{cases}$$

---

A native generalization by replacing the Euclidean gradient and Hessian by the Riemannian gradient and Hessian:

$$\begin{cases} \eta_k = \arg \min_{\eta \in T_{x_k} \mathcal{M}} f(x_k) + \langle \operatorname{grad} f(x_k), \eta \rangle + \frac{1}{2} \langle \eta, \operatorname{Hess} f(x_k) \eta \rangle + h(x_k + \eta) \\ x_{k+1} = R_{x_k}(\eta_k) \end{cases}$$

Does it converge superlinearly locally?

Not necessarily!



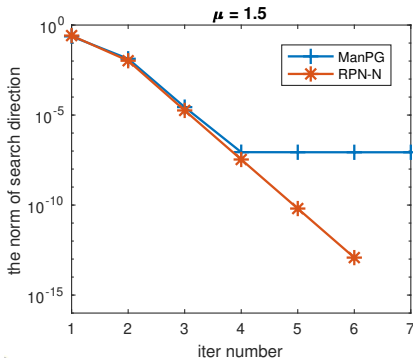
# A Riemannian Proximal Newton-CG Method

A native generalization

Consider the Sparse PCA over sphere:

$$\min_{x \in \mathbb{S}^{n-1}} -x^T A^T A x + \mu \|x\|_1,$$

where  $f(x) = -x^T A^T A x$ ,  $h(x) = \mu \|x\|_1$ .



**Figure:** Comparisons of native generalization (RPN-N) and the proximal gradient method (ManPG) in [CMSZ20].

# A Riemannian Proximal Newton-CG Method

A native generalization

Euclidean version:

$$\begin{cases} d_k = \operatorname{argmin}_p \langle \nabla f(x_k), p \rangle + \frac{1}{2} \langle p, \nabla^2 f(x_k) p \rangle + h(x_k + p) \\ x_{k+1} = x_k + d_k \end{cases}$$

---

A native generalization by replacing the Euclidean gradient and Hessian by the Riemannian gradient and Hessian:

$$\begin{cases} \eta_k = \operatorname{arg min}_{\eta \in T_{x_k}} \mathcal{M} f(x_k) + \langle \operatorname{grad} f(x_k), \eta \rangle + \frac{1}{2} \langle \eta, \operatorname{Hess} f(x_k) \eta \rangle + h(x_k + \eta) \\ x_{k+1} = R_{x_k}(\eta_k) \end{cases}$$

- $x_k + \eta$  in  $h$  is only a first order approximation;

# A Riemannian Proximal Newton-CG Method

A native generalization

Euclidean version:

$$\begin{cases} d_k = \operatorname{argmin}_p \langle \nabla f(x_k), p \rangle + \frac{1}{2} \langle p, \nabla^2 f(x_k) p \rangle + h(x_k + p) \\ x_{k+1} = x_k + d_k \end{cases}$$

---

A native generalization by replacing the Euclidean gradient and Hessian by the Riemannian gradient and Hessian:

$$\begin{cases} \eta_k = \operatorname{arg min}_{\eta \in T_{x_k}} \mathcal{M} f(x_k) + \langle \operatorname{grad} f(x_k), \eta \rangle + \frac{1}{2} \langle \eta, \operatorname{Hess} f(x_k) \eta \rangle + h(x_k + \eta) \\ x_{k+1} = R_{x_k}(\eta_k) \end{cases}$$
$$\begin{cases} \eta_k = \operatorname{arg min}_{\eta \in T_{x_k}} \mathcal{M} f(x_k) + \langle \operatorname{grad} f(x_k), \eta \rangle + \frac{1}{2} \langle \eta, \operatorname{Hess} f(x_k) \eta \rangle + h(x_k + \eta + \frac{1}{2} \Pi(\eta, \eta)) \\ x_{k+1} = R_{x_k}(\eta_k) \end{cases}$$

- $x_k + \eta$  in  $h$  is only a first order approximation;
- If an second order approximation is used, then the subproblem is difficult to solve;

# A Riemannian proximal Newton-CG method

A Riemannian proximal Newton method: description

$$\min_{x \in \mathcal{M}} F(x) = f(x) + h(x), h(x) = \mu \|x\|_1$$

## A Riemannian proximal Newton method (RPN)

- 1 Compute the ManPG direction

$$v(x_k) = \operatorname{argmin}_{v \in T_{x_k} \mathcal{M}} f(x_k) + \langle \nabla f(x_k), v \rangle + \frac{1}{2t} \|v\|_F^2 + h(x_k + v);$$

- 2 Find  $u(x_k) \in T_{x_k} \mathcal{M}$  by solving

$$J(x_k)[u(x_k)] = -v(x_k),$$

where  $J(x_k) = -[I_n - \Lambda_{x_k} + t\Lambda_{x_k}(\nabla^2 f(x_k) - \mathcal{L}_{x_k})]$ ,  $\Lambda_{x_k}$  and  $\mathcal{L}_{x_k}$  are defined later;

- 3  $x_{k+1} = R_{x_k}(u(x_k));$

# A Riemannian proximal Newton-CG method

A Riemannian proximal Newton method: description

$$\min_{x \in \mathcal{M}} F(x) = f(x) + h(x), h(x) = \mu \|x\|_1$$

## A Riemannian proximal Newton method (RPN)

- 1 Compute the ManPG direction

$$v(x_k) = \operatorname{argmin}_{v \in T_{x_k} \mathcal{M}} f(x_k) + \langle \nabla f(x_k), v \rangle + \frac{1}{2t} \|v\|_F^2 + h(x_k + v);$$

- 2 Find  $u(x_k) \in T_{x_k} \mathcal{M}$  by solving

$$J(x_k)[u(x_k)] = -v(x_k),$$

where  $J(x_k) = -[I_n - \Lambda_{x_k} + t\Lambda_{x_k}(\nabla^2 f(x_k) - \mathcal{L}_{x_k})]$ ,  $\Lambda_{x_k}$  and  $\mathcal{L}_{x_k}$  are defined later;

- 3  $x_{k+1} = R_{x_k}(u(x_k));$

- 1 Step 1: compute a Riemannian proximal gradient direction (ManPG)

# A Riemannian proximal Newton-CG method

A Riemannian proximal Newton method: description

$$\min_{x \in \mathcal{M}} F(x) = f(x) + h(x), h(x) = \mu \|x\|_1$$

## A Riemannian proximal Newton method (RPN)

- 1 Compute the ManPG direction

$$v(x_k) = \operatorname{argmin}_{v \in T_{x_k} \mathcal{M}} f(x_k) + \langle \nabla f(x_k), v \rangle + \frac{1}{2t} \|v\|_F^2 + h(x_k + v);$$

- 2 Find  $u(x_k) \in T_{x_k} \mathcal{M}$  by solving

$$J(x_k)[u(x_k)] = -v(x_k),$$

where  $J(x_k) = -[I_n - \Lambda_{x_k} + t\Lambda_{x_k}(\nabla^2 f(x_k) - \mathcal{L}_{x_k})]$ ,  $\Lambda_{x_k}$  and  $\mathcal{L}_{x_k}$  are defined later;

- 3  $x_{k+1} = R_{x_k}(u(x_k));$

- 1 Step 1: compute a Riemannian proximal gradient direction (ManPG)
- 2 Step 2: compute the Riemannian proximal Newton direction, where  $J(x_k)$  is from a generalized Jacobi of  $v(x_k)$ ;

# A Riemannian proximal Newton-CG method

A Riemannian proximal Newton method: description

$$\min_{x \in \mathcal{M}} F(x) = f(x) + h(x), h(x) = \mu \|x\|_1$$

## A Riemannian proximal Newton method (RPN)

- 1 Compute the ManPG direction

$$v(x_k) = \operatorname{argmin}_{v \in T_{x_k} \mathcal{M}} f(x_k) + \langle \nabla f(x_k), v \rangle + \frac{1}{2t} \|v\|_F^2 + h(x_k + v);$$

- 2 Find  $u(x_k) \in T_{x_k} \mathcal{M}$  by solving

$$J(x_k)[u(x_k)] = -v(x_k),$$

where  $J(x_k) = -[I_n - \Lambda_{x_k} + t\Lambda_{x_k}(\nabla^2 f(x_k) - \mathcal{L}_{x_k})]$ ,  $\Lambda_{x_k}$  and  $\mathcal{L}_{x_k}$  are defined later;

- 3  $x_{k+1} = R_{x_k}(u(x_k));$

- 1 Step 1: compute a Riemannian proximal gradient direction (ManPG)
- 2 Step 2: compute the Riemannian proximal Newton direction, where  $J(x_k)$  is from a generalized Jacobi of  $v(x_k)$ ;
- 3 Step 3: Update iterate by a retraction;

# A Riemannian proximal Newton-CG method

A Riemannian proximal Newton method: local superlinear convergence rate

Without loss of generality, we assume that the nonzero entries of  $x_*$  are in the first part, i.e.,  $x_* = [\bar{x}_*^T, 0^T]^T$ .  $B_x$  denotes an orthonormal basis of  $T_x^\perp \mathcal{M}$  at  $x$ .

---

Assumption:

- 1 Let  $B_{x_*}^T = [\bar{B}_{x_*}^T, \hat{B}_{x_*}^T]$ , where  $\bar{B}_{x_*} \in \mathbb{R}^{j \times d}$  and  $\hat{B}_{x_*} \in \mathbb{R}^{(n-j) \times d}$ . It is assumed that  $j \geq d$  and  $\bar{B}_{x_*}$  is full column rank;



# A Riemannian proximal Newton-CG method

A Riemannian proximal Newton method: local superlinear convergence rate

Without loss of generality, we assume that the nonzero entries of  $x_*$  are in the first part, i.e.,  $x_* = [\bar{x}_*^T, 0^T]^T$ .  $B_x$  denotes an orthonormal basis of  $T_x^\perp \mathcal{M}$  at  $x$ .

---

Assumption:

- 1 Let  $B_{x_*}^T = [\bar{B}_{x_*}^T, \hat{B}_{x_*}^T]$ , where  $\bar{B}_{x_*} \in \mathbb{R}^{j \times d}$  and  $\hat{B}_{x_*} \in \mathbb{R}^{(n-j) \times d}$ . It is assumed that  $j \geq d$  and  $\bar{B}_{x_*}$  is full column rank;
- 2 There exists a neighborhood  $\mathcal{U}$  of  $x_* = [\bar{x}_*^T, 0^T]^T$  on  $\mathcal{M}$  such that for any  $x = [\bar{x}^T, \hat{x}^T]^T \in \mathcal{U}$ , it holds that  $\bar{x} + \bar{v} \neq 0$  and  $\hat{x} + \hat{v} = 0$ .

# A Riemannian proximal Newton-CG method

A Riemannian proximal Newton method: local superlinear convergence rate

## Theorem

*Suppose that  $x_*$  be a local optimal minimizer. Under the above Assumptions, assume that  $J(x_*)$  is nonsingular. Then there exists a neighborhood  $\mathcal{U}$  of  $x_*$  on  $\mathcal{M}$  such that for any  $x_0 \in \mathcal{U}$ , RPN Algorithm generates the sequence  $\{x_k\}$  converging superlinearly to  $x_*$ .*

The convergence rate is improved to quadratically convergence in [SAH<sup>+</sup>24a]

# A Riemannian proximal Newton-CG method

A Riemannian proximal Newton method: a hybrid version

- Similar to the Riemannian Newton method, this Riemannian proximal Newton method does not guarantee global convergence;

# A Riemannian proximal Newton-CG method

A Riemannian proximal Newton method: a hybrid version

- Similar to the Riemannian Newton method, this Riemannian proximal Newton method does not guarantee global convergence;
- A hybrid method that merges ManPG with RPN is proposed in [SAH<sup>+</sup>24b];

---

**Require:**  $x_0 \in \mathcal{M}$ ,  $t > 0$ ,  $\epsilon > 0$ ;

```
1: for  $k = 0, 1, \dots$  do  
2:   Compute a ManPG direction  $v_k$ ;  
3:   If  $\|v_k\| \leq \epsilon$ , then  $K = k$  and break;  
4:    $x_{k+1} = R_{x_k}(\alpha v_k)$  with an appropriate step size;  
5: end for  
6: for  $k = K+1, K+2, \dots$  do  
7:   Compute  $u_k$  by solving  $J(x_k)u_k = -v_k$  with  $v_k$  being the ManPG  
   direction;  
8:    $x_{k+1} = R_{x_k}(u_k)$ ;  
9: end for
```

---

# A Riemannian proximal Newton-CG method

A Riemannian proximal Newton method: a hybrid version

- Similar to the Riemannian Newton method, this Riemannian proximal Newton method does not guarantee global convergence;
- A hybrid method that merges ManPG with RPN is proposed in [SAH<sup>+</sup>24b];

---

**Require:**  $x_0 \in \mathcal{M}$ ,  $t > 0$ ,  $\epsilon > 0$ ;

```
1: for  $k = 0, 1, \dots$  do  
2:   Compute a ManPG direction  $v_k$ ;  
3:   If  $\|v_k\| \leq \epsilon$ , then  $K = k$  and break;  
4:    $x_{k+1} = R_{x_k}(\alpha v_k)$  with an appropriate step size;  
5: end for  
6: for  $k = K+1, K+2, \dots$  do  
7:   Compute  $u_k$  by solving  $J(x_k)u_k = -v_k$  with  $v_k$  being the ManPG  
   direction;  
8:    $x_{k+1} = R_{x_k}(u_k)$ ;  
9: end for
```

---

The switching parameter  $\epsilon$  is crucial for the performance.

# A Riemannian proximal Newton-CG method

## Truncated conjugate gradient

### A Riemannian proximal Newton method (RPN)

- 1 Compute the ManPG direction

$$v(x_k) = \operatorname{argmin}_{v \in T_{x_k} \mathcal{M}} f(x_k) + \langle \nabla f(x_k), v \rangle + \frac{1}{2t} \|v\|_F^2 + h(x_k + v);$$

- 2 Find  $u(x_k) \in T_{x_k} \mathcal{M}$  by solving

$$J(x_k)[u(x_k)] = -v(x_k);$$

- 3  $x_{k+1} = R_{x_k}(u(x_k));$

Smooth case:

- $v(x_k) = -t \operatorname{grad} f(x_k);$
- $J(x_k) = -t \operatorname{Hess} f(x_k);$
- $J(x_k)[u(x_k)] = -v(x_k) \implies$   
 $\underbrace{\operatorname{Hess} f(x_k)[u(x_k)] = -\operatorname{grad} f(x_k)}_{\text{truncated conjugate gradient (tCG)}} .$

# A Riemannian proximal Newton-CG method

## Truncated conjugate gradient

### A Riemannian proximal Newton method (RPN)

- 1 Compute the ManPG direction

$$v(x_k) = \operatorname{argmin}_{v \in T_{x_k} \mathcal{M}} f(x_k) + \langle \nabla f(x_k), v \rangle + \frac{1}{2t} \|v\|_F^2 + h(x_k + v);$$

- 2 Find  $u(x_k) \in T_{x_k} \mathcal{M}$  by solving

$$J(x_k)[u(x_k)] = -v(x_k);$$

- 3  $x_{k+1} = R_{x_k}(u(x_k));$

Smooth case:

- $v(x_k) = -t \operatorname{grad} f(x_k);$
- $J(x_k) = -t \operatorname{Hess} f(x_k);$
- $J(x_k)[u(x_k)] = -v(x_k) \implies$   
 $\underbrace{\operatorname{Hess} f(x_k)[u(x_k)] = -\operatorname{grad} f(x_k)}_{\text{truncated conjugate gradient (tCG)}}$

Nonsmooth case:

- $v(x_k)$ : ManPG direction;
- $J(x_k)$ : Generalized Jacobi of  $v$ ;
- $u(x_k)$ : solving a linear system by  
 $\underbrace{J(x_k)[u(x_k)] = -v(x_k)}_{\text{tCG?}}$

# A Riemannian proximal Newton-CG method

Truncated conjugate gradient

## A Riemannian proximal Newton method (RPN)

- 1 Compute the ManPG direction

$$v(x_k) = \operatorname{argmin}_{v \in T_{x_k} \mathcal{M}} f(x_k) + \langle \nabla f(x_k), v \rangle + \frac{1}{2t} \|v\|_F^2 + h(x_k + v);$$

- 2 Find  $u(x_k) \in T_{x_k} \mathcal{M}$  by solving

$$J(x_k)[u(x_k)] = -v(x_k);$$

- 3  $x_{k+1} = R_{x_k}(u(x_k));$

Smooth case:

- $v(x_k) = -t \operatorname{grad} f(x_k);$
- $J(x_k) = -t \operatorname{Hess} f(x_k);$
- $J(x_k)[u(x_k)] = -v(x_k) \implies$   
 $\underbrace{\operatorname{Hess} f(x_k)[u(x_k)] = -\operatorname{grad} f(x_k)}_{\text{truncated conjugate gradient (tCG)}}$

Nonsmooth case:

- $v(x_k)$ : ManPG direction;
- $J(x_k)$ : Generalized Jacobi of  $v$ ;
- $u(x_k)$ : solving a linear system by  
 $\underbrace{J(x_k)[u(x_k)] = -v(x_k)}_{\text{tCG?}}$

Problem:  $J(x_k)$  is not symmetric!



# A Riemannian proximal Newton-CG method

## Truncated conjugate gradient

Notation:

$$\mathfrak{B}_{x_k} = \nabla^2 f(x_k) - \mathcal{L}_{x_k} = \begin{pmatrix} \mathfrak{B}_{x_k}^{(11)} & \mathfrak{B}_{x_k}^{(12)} \\ \mathfrak{B}_{x_k}^{(21)} & \mathfrak{B}_{x_k}^{(22)} \end{pmatrix}, \mathcal{B}_{x_k} = \mathfrak{B}_{x_k}^{(11)}.$$

$$J(x_k) = - \begin{pmatrix} \bar{B}_{x_k} \bar{B}_{x_k}^\dagger + t(l_{j_k} - \bar{B}_{x_k} \bar{B}_{x_k}^\dagger) \mathcal{B}_{x_k} & t(l_{j_k} - \bar{B}_{x_k} \bar{B}_{x_k}^\dagger) \mathfrak{B}_{x_k}^{(12)} \\ 0_{(n-j_k) \times j_k} & l_{n-j_k} \end{pmatrix}$$

$$\begin{cases} [\bar{B}_{x_k} \bar{B}_{x_k}^\dagger + t(l_{j_k} - \bar{B}_{x_k} \bar{B}_{x_k}^\dagger) \mathcal{B}_{x_k}] \bar{u}(x_k) = \bar{v}(x_k) - t(l_{j_k} - \bar{B}_{x_k} \bar{B}_{x_k}^\dagger) \mathfrak{B}_{x_k}^{(12)} \hat{u}(x_k) \\ \hat{u}(x_k) = \hat{v}(x_k) \end{cases}.$$

$$\implies \bar{u}(x_k) = \bar{v}(x_k) - \{l_{j_k} + (l_{j_k} - \bar{B}_{x_k} \bar{B}_{x_k}^\dagger) N_{x_k}\}^{-1} (l_{j_k} - \bar{B}_{x_k} \bar{B}_{x_k}^\dagger) \ell_{x_k}$$

where  $\ell_{x_k} = \frac{1}{t_k}(-l_{j_k} + t_k \mathcal{B}_{x_k}) \bar{v}(x_k) + \mathfrak{B}_{x_k}^{(12)} \hat{v}(x_k)$  and  $N_{x_k} = -l_{j_k} + t \mathcal{B}_{x_k}$  is symmetric.

# A Riemannian proximal Newton-CG method

## Truncated conjugate gradient

$$\bar{u}(x_k) = \bar{v}(x_k) - \underbrace{\{I_{j_k} + (I_{j_k} - \bar{B}_{x_k} \bar{B}_{x_k}^\dagger) N_{x_k}\}}_{\text{symmetric}}^{-1} (I_{j_k} - \bar{B}_{x_k} \bar{B}_{x_k}^\dagger) \ell_{x_k}$$

### Lemma

Let  $N \in \mathbb{R}^{j \times j}$  and  $B \in \mathbb{R}^{j \times m}$  with  $m \leq j$ . Suppose that  $I_j + N$  is symmetric positive definite on  $\{w \mid B^T w = 0\}$  and that  $B$  is full column rank. Then it holds that the unique solution of the problem

$$\min_{B^T w = 0} \ell^T w + \frac{1}{2} w^T (I_j + N) w$$

is given by

$$w_* = - [I_j + (I_j - BB^\dagger)N]^{-1} [I_j - BB^\dagger] \ell.$$

# A Riemannian proximal Newton-CG method

Truncated conjugate gradient

$$\bar{u}(x_k) = \bar{v}(x_k) - \{I_{j_k} + (I_{j_k} - \bar{B}_{x_k} \bar{B}_{x_k}^\dagger) \underbrace{N_{x_k}}_{\text{symmetric}}\}^{-1} (I_{j_k} - \bar{B}_{x_k} \bar{B}_{x_k}^\dagger) \ell_{x_k}$$

## Corollary

*Suppose  $\bar{B}_{x_k}$  has full column rank,  $B_{x_k}$  is symmetric positive definite on  $\{w \mid B^T w = 0\}$ . Then the proximal Newton equation  $J(x_k)[u(x_k)] = -v(x_k)$  can be computed by*

$$u(x_k) = \begin{pmatrix} \bar{v}(x_k) + w(x_k) \\ \hat{v}(x_k) \end{pmatrix},$$

*where  $w(x_k) = \operatorname{argmin}_{\bar{B}_{x_k}^T w = 0} \ell_{x_k}^T w + \frac{1}{2} w^T B_{x_k} w$ .*

# A Riemannian proximal Newton-CG method

Truncated conjugate gradient

$$\bar{u}(x_k) = \bar{v}(x_k) - \{l_{j_k} + (l_{j_k} - \bar{B}_{x_k} \bar{B}_{x_k}^\dagger) \underbrace{N_{x_k}}_{\text{symmetric}}\}^{-1} (l_{j_k} - \bar{B}_{x_k} \bar{B}_{x_k}^\dagger) \ell_{x_k}$$

## Corollary

Suppose  $\bar{B}_{x_k}$  has full column rank,  $B_{x_k}$  is symmetric positive definite on  $\{w \mid B^T w = 0\}$ . Then the proximal Newton equation  $J(x_k)[u(x_k)] = -v(x_k)$  can be computed by

$$u(x_k) = \begin{pmatrix} \bar{v}(x_k) + w(x_k) \\ \hat{v}(x_k) \end{pmatrix},$$

where  $w(x_k) = \operatorname{argmin}_{\bar{B}_{x_k}^T w = 0} \ell_{x_k}^T w + \frac{1}{2} w^T B_{x_k} w$ .

tCG can be used for the computation of  $w(x_k)$ .

# A Riemannian proximal Newton-CG method

Truncated conjugate gradient

## A Riemannian proximal Newton method (RPN)

- 1  $v(x_k) = \operatorname{argmin}_{v \in T_{x_k} \mathcal{M}} f(x_k) + \langle \nabla f(x_k), v \rangle + \frac{1}{2t} \|v\|_F^2 + h(x_k + v);$
- 2  $d(x_k) = \begin{pmatrix} \bar{d}(x_k) \\ \hat{d}(x_k) \end{pmatrix} = \begin{pmatrix} \bar{v}(x_k) + w(x_k) \\ \hat{v}(x_k) \end{pmatrix},$  where  $w(x_k)$  is an output of tCG for solving  $\min_{\bar{B}_{x_k}^T w = 0} \langle \ell_{x_k}, w \rangle + \frac{1}{2} \langle w, \mathcal{B}_{x_k} w \rangle.$
- 3  $x_{k+1} = R_{x_k}(\alpha_k d(x_k))$  with an appropriate step size  $\alpha_k$ ;

Question:

- Is  $\mathcal{B}_{x_k}$  symmetric positive definite near a local minimizer  $x_*$ ?
- What is the early termination conditions for tCG?
  - Guarantee global convergence;
  - Guarantee local superlinear convergence;

# A Riemannian proximal Newton-CG method

Truncated conjugate gradient

Is  $\mathcal{B}_{x_k}$  symmetric positive definite near  $x_*$ ?

# A Riemannian proximal Newton-CG method

Truncated conjugate gradient

Is  $\mathcal{B}_{x_k}$  symmetric positive definite near  $x_*$ ?

---

Assumption:

- 1 The function  $f$  is twice continuously differentiable with a Lipschitz continuous Euclidean Hessian;
- 2 Let  $B_{x_*}^T = [\bar{B}_{x_*}^T, \hat{B}_{x_*}^T]$ , where  $\bar{B}_{x_*} \in \mathbb{R}^{j \times d}$  and  $\hat{B}_{x_*} \in \mathbb{R}^{(n-j) \times d}$ . It is assumed that  $j \geq d$  and  $\bar{B}_{x_*}$  is full column rank;
- 3 There exists a neighborhood  $\mathcal{U}$  of  $x_* = [\bar{x}_*^T, 0^T]^T$  on  $\mathcal{M}$  such that for any  $x = [\bar{x}^T, \tilde{x}^T]^T \in \mathcal{U}$ , it holds that  $\bar{x} + \bar{v} \neq 0$  and  $\hat{x} + \hat{v} = 0$ ;
- 4 The linear operator  $\mathcal{B}_{x_*}$  is positive definite on the subspace  $\mathfrak{L}_{x_*} = \{w \mid \bar{B}_{x_*}^T w = 0\}$ .

# A Riemannian proximal Newton-CG method

Truncated conjugate gradient

Is  $\mathcal{B}_{x_k}$  symmetric positive definite near  $x_*$ ?

---

Assumption:

- 1 The function  $f$  is twice continuously differentiable with a Lipschitz continuous Euclidean Hessian;
  - 2 Let  $B_{x_*}^T = [\bar{B}_{x_*}^T, \hat{B}_{x_*}^T]$ , where  $\bar{B}_{x_*} \in \mathbb{R}^{j \times d}$  and  $\hat{B}_{x_*} \in \mathbb{R}^{(n-j) \times d}$ . It is assumed that  $j \geq d$  and  $\bar{B}_{x_*}$  is full column rank;
  - 3 There exists a neighborhood  $\mathcal{U}$  of  $x_* = [\bar{x}_*^T, 0^T]^T$  on  $\mathcal{M}$  such that for any  $x = [\bar{x}^T, \tilde{x}^T]^T \in \mathcal{U}$ , it holds that  $\bar{x} + \bar{v} \neq 0$  and  $\hat{x} + \hat{v} = 0$ ;
  - 4 The linear operator  $\mathcal{B}_{x_*}$  is positive definite on the subspace  $\mathfrak{L}_{x_*} = \{w \mid \bar{B}_{x_*}^T w = 0\}$ .
- 

- Under the second assumption, the intersection of the manifold and the sparsity constraints forms an embedded submanifold around  $x_*$ ;
- $\mathcal{B}_{x_*}$  is the Riemannian Hessian of  $F$  at  $x_*$  for the submanifold;
- $\mathcal{B}_{x_*}$  is symmetric positive semidefinite on  $\mathfrak{L}_{x_*}$ ;



# A Riemannian proximal Newton-CG method

Truncated conjugate gradient

Is  $\mathcal{B}_{x_k}$  symmetric positive definite near  $x_*$ ?

Assumption:

- 1 The function  $f$  is twice continuously differentiable with a Lipschitz continuous Euclidean Hessian;
- 2 Let  $B_{x_*}^T = [\bar{B}_{x_*}^T, \hat{B}_{x_*}^T]$ , where  $\bar{B}_{x_*} \in \mathbb{R}^{j \times d}$  and  $\hat{B}_{x_*} \in \mathbb{R}^{(n-j) \times d}$ . It is assumed that  $j \geq d$  and  $\bar{B}_{x_*}$  is full column rank;
- 3 There exists a neighborhood  $\mathcal{U}$  of  $x_* = [\bar{x}_*^T, 0^T]^T$  on  $\mathcal{M}$  such that for any  $x = [\bar{x}^T, \tilde{x}^T]^T \in \mathcal{U}$ , it holds that  $\bar{x} + \bar{v} \neq 0$  and  $\hat{x} + \hat{v} = 0$ ;
- 4 The linear operator  $\mathcal{B}_{x_*}$  is positive definite on the subspace  $\mathfrak{L}_{x_*} = \{w \mid \bar{B}_{x_*}^T w = 0\}$ .

## Lemma

*Suppose the above Assumption holds. Then there exists a neighborhood of  $x_*$ , denoted by  $\mathcal{V}_2$ , and a positive constant  $\chi_\epsilon$  such that the smallest eigenvalue of  $\mathcal{B}_x$  on  $\mathfrak{L}_x$  is greater than  $\chi_\epsilon$  for all  $x \in \mathcal{V}_2$ . This implies  $\mathcal{B}_x$  is positive definite on  $\mathfrak{L}_x$  for all  $x \in \mathcal{V}_2$ .*

# A Riemannian proximal Newton-CG method

Truncated conjugate gradient

## Early termination conditions in tCG

tCG step

2  $d(x_k) = \begin{pmatrix} \bar{d}(x_k) \\ \hat{d}(x_k) \end{pmatrix} = \begin{pmatrix} \bar{v}(x_k) + w(x_k) \\ \hat{v}(x_k) \end{pmatrix}$ , where  $w(x_k)$  is an output of tCG for solving  $\min_{\bar{B}_{x_k}^T w=0} \langle \ell_{x_k}, w \rangle + \frac{1}{2} \langle w, \mathcal{B}_{x_k} w \rangle$ .

# A Riemannian proximal Newton-CG method

Truncated conjugate gradient

## Early termination conditions in tCG

tCG step

- 2  $d(x_k) = \begin{pmatrix} \bar{d}(x_k) \\ \hat{d}(x_k) \end{pmatrix} = \begin{pmatrix} \bar{v}(x_k) + w(x_k) \\ \hat{v}(x_k) \end{pmatrix}$ , where  $w(x_k)$  is an output of tCG for solving  $\min_{\bar{B}_{x_k}^T w=0} \langle \ell_{x_k}, w \rangle + \frac{1}{2} \langle w, \mathcal{B}_{x_k} w \rangle$ .

## Difficulty

- Smooth:

$$\text{approximately } \min_{d \in T_{x_k} \mathcal{M}} \langle \text{grad } f(x_k), d \rangle + \frac{1}{2} \langle \text{Hess } f(x_k)[d], d \rangle,$$

find  $d(x_k)$  such that  $\langle d(x_k), \text{grad } f(x_k) \rangle < 0$ ;

- Nonsmooth:

$$\text{approximately } \min_{\bar{B}_{x_k}^T w=0} \langle \ell_{x_k}, w \rangle + \frac{1}{2} \langle w, \mathcal{B}_{x_k} w \rangle,$$

find  $w(x_k)$  such that  $d(x_k)$  is a descent direction;

# A Riemannian proximal Newton-CG method

Truncated conjugate gradient

## Early termination conditions in tCG

tCG step

- 2  $d(x_k) = \begin{pmatrix} \bar{d}(x_k) \\ \hat{d}(x_k) \end{pmatrix} = \begin{pmatrix} \bar{v}(x_k) + w(x_k) \\ \hat{v}(x_k) \end{pmatrix}$ , where  $w(x_k)$  is an output of tCG for solving  $\min_{\bar{B}_{x_k}^T w=0} \langle \ell_{x_k}, w \rangle + \frac{1}{2} \langle w, \mathcal{B}_{x_k} w \rangle$ .

## Difficulty

- Smooth:

$$\text{approximately } \min_{d \in T_{x_k} \mathcal{M}} \langle \text{grad } f(x_k), d \rangle + \frac{1}{2} \langle \text{Hess } f(x_k)[d], d \rangle,$$

find  $d(x_k)$  such that  $\langle d(x_k), \text{grad } f(x_k) \rangle < 0$ ;

- Nonsmooth:

$$\text{approximately } \min_{\bar{B}_{x_k}^T w=0} \langle \ell_{x_k}, w \rangle + \frac{1}{2} \langle w, \mathcal{B}_{x_k} w \rangle,$$

find  $w(x_k)$  such that  $d(x_k)$  is a descent direction;

The early termination conditions for the smooth case are not sufficient.

# A Riemannian proximal Newton-CG method

Truncated conjugate gradient

## Early termination conditions in tCG

---

**Algorithm:** Truncated conjugate gradient (tCG)

---

**Require:**  $\vartheta > 0$ ,  $\gamma > 0$ ,  $\tau > 0$ ,  $\theta > 0$ , and  $\kappa \in (0, 1)$ ;

**Ensure:**  $(w(x), \text{status})$ ;

1: **if**  $G_x(v(x)) > G_x(0)$  **then**

2:   return  $w(x) = 0$  and  $\text{status} = \text{'early1'}$ ;

3: **end if**

4:  $z = \mathfrak{B}v(x)$ ;

5: **if**  $\langle v(x), z \rangle + \tau \|\hat{v}(x)\|_F^2 < \gamma \|v(x)\|_F^2$  **then**

6:   return  $w(x) = 0$  and  $\text{status} = \text{'early2'}$ ;

7: **end if**

8:  $w_0 = 0$ ,  $r_0 = P_x(\ell_x)$ ,  $o_0 = -r_0$ ,  $\delta_0 = \langle r_0, r_0 \rangle$ ,  $t_0 = z$ ;

9: ..... (CG iterations)

---

Omit subscript  $k$  for simplicity

# A Riemannian proximal Newton-CG method

Truncated conjugate gradient

## Early termination conditions in tCG

---

**Algorithm:** Truncated conjugate gradient (tCG)

---

**Require:**  $\vartheta > 0$ ,  $\gamma > 0$ ,  $\tau > 0$ ,  $\theta > 0$ , and  $\kappa \in (0, 1)$ ;

**Ensure:**  $(w(x), \text{status})$ ;

- 1: **if**  $G_x(v(x)) > G_x(0)$  **then**
  - 2:   return  $w(x) = 0$  and status = 'early1';
  - 3: **end if**
  - 4:  $z = \mathfrak{B}v(x)$ ;
  - 5: **if**  $\langle v(x), z \rangle + \tau \|\hat{v}(x)\|_F^2 < \gamma \|v(x)\|_F^2$  **then**
  - 6:   return  $w(x) = 0$  and status = 'early2';
  - 7: **end if**
  - 8:  $w_0 = 0$ ,  $r_0 = P_x(\ell_x)$ ,  $o_0 = -r_0$ ,  $\delta_0 = \langle r_0, r_0 \rangle$ ,  $t_0 = z$ ;
  - 9: ..... (CG iterations)
- 

- $G_x(u) = f(x) + \langle \nabla f(x), u \rangle + \frac{1}{2} \langle u, \mathfrak{B}_x u \rangle + \frac{\tau}{2} \|\hat{u}(x)\|_F^2 + h(x + u)$ ;
- Use to guarantee global convergence;
- $\frac{\tau}{2} \|\hat{u}(x)\|_F^2$  is added for the condition in Step 5;

# A Riemannian proximal Newton-CG method

Truncated conjugate gradient

## Early termination conditions in tCG

---

**Algorithm:** Truncated conjugate gradient (tCG)

---

**Require:**  $\vartheta > 0$ ,  $\gamma > 0$ ,  $\tau > 0$ ,  $\theta > 0$ , and  $\kappa \in (0, 1)$ ;

**Ensure:**  $(w(x), \text{status})$ ;

1: **if**  $G_x(v(x)) > G_x(0)$  **then**

2:   return  $w(x) = 0$  and  $\text{status} = \text{'early1'}$ ;

3: **end if**

4:  $z = \mathfrak{B}v(x)$ ;

5: **if**  $\langle v(x), z \rangle + \tau \|\hat{v}(x)\|_F^2 < \gamma \|v(x)\|_F^2$  **then**

6:   return  $w(x) = 0$  and  $\text{status} = \text{'early2'}$ ;

7: **end if**

8:  $w_0 = 0$ ,  $r_0 = P_x(\ell_x)$ ,  $o_0 = -r_0$ ,  $\delta_0 = \langle r_0, r_0 \rangle$ ,  $t_0 = z$ ;

9: ..... (CG iterations)

---

- Use to guarantee global convergence;
- $\tau \|\hat{v}(x)\|_F^2$  is used since  $\mathfrak{B}_x \succ 0$  may not hold;

# A Riemannian proximal Newton-CG method

Truncated conjugate gradient

## Early termination conditions in tCG

---

**Algorithm:** Truncated conjugate gradient (tCG)

---

**Require:**  $\vartheta > 0$ ,  $\gamma > 0$ ,  $\tau > 0$ ,  $\theta > 0$ , and  $\kappa \in (0, 1)$ ;

**Ensure:**  $(w(x), \text{status})$ ;

- 1: ..... (See the previous slide)
  - 2:  $w_0 = 0$ ,  $r_0 = P_x(\ell_x)$ ,  $o_0 = -r_0$ ,  $\delta_0 = \langle r_0, r_0 \rangle$ ,  $t_0 = z$ ;
  - 3: **for**  $i = 0, 1, \dots$  **do**
  - 4:    $p_i = \mathcal{B}o_i$  and  $q_i = P_x(p_i)$ ;
  - 5:   **if**  $\langle o_i, q_i \rangle \leq \vartheta \delta_i$  **then**
  - 6:     return  $w(x) = w_i$  and status = 'neg';
  - 7:   **end if**
  - 8:   ..... (Remaining CG iterations)
  - 9: **end for**
- 

An existing early termination condition



# A Riemannian proximal Newton-CG method

Truncated conjugate gradient

## Early termination conditions in tCG

---

**Algorithm:** Truncated conjugate gradient (tCG)

---

**Require:**  $\vartheta > 0$ ,  $\gamma > 0$ ,  $\tau > 0$ ,  $\theta > 0$ , and  $\kappa \in (0, 1)$ ;

**Ensure:**  $(w(x), \text{status})$ ;

```
1: ..... (See previous slides)
2: for  $i = 0, 1, \dots$  do
3:   ..... (See previous slides)
4:    $\alpha_i = \frac{\langle r_i, r_i \rangle}{\langle o_i, q_i \rangle}$ ;  $w_{i+1} = w_i + \alpha_i o_i$ ;  $r_{i+1} = r_i + \alpha_i q_i$ ;
5:    $d_{i+1} = \begin{pmatrix} \bar{v}(x) + w_{i+1} \\ \hat{v}(x) \end{pmatrix}$ ,  $t_{i+1} = t_i + \alpha_i \begin{pmatrix} p_i \\ \mathfrak{B}_{21} o_i \end{pmatrix}$ ;
6:   if  $\langle d_{i+1}, t_{i+1} \rangle + \tau \|\hat{v}(x)\|_F^2 < \gamma \|d_{i+1}\|_F^2$  or  $G_x(d_{i+1}) > G_x(0)$  then
7:     return  $w(x) = w_i$  and  $\text{status} = \text{'early3'}$ ;
8:   end if
9:   ..... (Remaining CG iterations)
10: end for
```

---

Use to guarantee global convergence

# A Riemannian proximal Newton-CG method

## Truncated conjugate gradient

### Early termination conditions in tCG

---

**Algorithm:** Truncated conjugate gradient (tCG)

---

**Require:**  $\vartheta > 0$ ,  $\gamma > 0$ ,  $\tau > 0$ ,  $\theta > 0$ , and  $\kappa \in (0, 1)$ ;

**Ensure:**  $(w(x), \text{status})$ ;

```
1: ..... (See previous slides)
2: for  $i = 0, 1, \dots$  do
3:   ..... (See previous slides)
4:    $\beta_{i+1} = \frac{\langle r_{i+1}, r_{i+1} \rangle}{\langle r_i, r_i \rangle}$ ;  $o_{i+1} = -r_{i+1} + \beta_{i+1} o_i$ ;
5:    $\delta_{i+1} = \langle r_{i+1}, r_{i+1} \rangle + \beta_{i+1}^2 \delta_i$ ; (Note that  $\delta_{i+1} = \langle o_{i+1}, o_{i+1} \rangle$ )
6:    $i = i + 1$ ;
7:   if  $\|r_i\|_F \leq \|r_0\|_F \min(\|r_0\|_F^\theta, \kappa)$  then
8:     return  $w(x) = w_i$ , and  $\text{status} = 'lin'$  if  $\|r_0\|_F^\theta > \kappa$  and
        $\text{status} = 'sup'$  otherwise;
9:   end if
10: end for
```

---

An existing early termination condition

# A Riemannian proximal Newton-CG method

RPN-CG: global convergence

Assumption:

- 1 The function  $f$  is twice continuously differentiable with a Lipschitz continuous gradient;

## Theorem

*Suppose the above Assumption holds and the parameters are appropriately chosen. Then it holds that*

$$\lim_{k \rightarrow \infty} \|v(x_k)\|_F = 0.$$

# A Riemannian proximal Newton-CG method

RPN-CG: local superlinear convergence

Assumption:

- 1 The function  $f$  is twice continuously differentiable with a Lipschitz continuous Euclidean Hessian;
- 2 Let  $B_{x_*}^T = [\bar{B}_{x_*}^T, \hat{B}_{x_*}^T]$ , where  $\bar{B}_{x_*} \in \mathbb{R}^{j \times d}$  and  $\hat{B}_{x_*} \in \mathbb{R}^{(n-j) \times d}$ . It is assumed that  $j \geq d$  and  $\bar{B}_{x_*}$  is full column rank;
- 3 There exists a neighborhood  $\mathcal{U}$  of  $x_* = [\bar{x}_*^T, 0^T]^T$  on  $\mathcal{M}$  such that for any  $x = [\bar{x}^T, \tilde{x}^T]^T \in \mathcal{U}$ , it holds that  $\bar{x} + \bar{v} \neq 0$  and  $\hat{x} + \hat{v} = 0$ ;
- 4 The function  $F$  is  $\varsigma$ -geodesically strongly convex at  $x_*$ , i.e., there exists a neighborhood  $\tilde{\mathcal{U}}_{x_*}$  of  $x_*$  in  $\mathcal{M}$  such that

$$F(y) \geq F(x_*) + \frac{\varsigma}{2} \|\text{Exp}_{x_*}^{-1}(y)\|_F^2$$

holds for any  $y \in \tilde{\mathcal{U}}_{x_*}$ .

# A Riemannian proximal Newton-CG method

RPN-CG: local superlinear convergence

Assumption:

- ① The function  $f$  is twice continuously differentiable with a Lipschitz continuous Euclidean Hessian;
- ② Let  $B_{x_*}^T = [\bar{B}_{x_*}^T, \hat{B}_{x_*}^T]$ , where  $\bar{B}_{x_*} \in \mathbb{R}^{j \times d}$  and  $\hat{B}_{x_*} \in \mathbb{R}^{(n-j) \times d}$ . It is assumed that  $j \geq d$  and  $\bar{B}_{x_*}$  is full column rank;
- ③ There exists a neighborhood  $\mathcal{U}$  of  $x_* = [\bar{x}_*^T, 0^T]^T$  on  $\mathcal{M}$  such that for any  $x = [\bar{x}^T, \tilde{x}^T]^T \in \mathcal{U}$ , it holds that  $\bar{x} + \bar{v} \neq 0$  and  $\hat{x} + \hat{v} = 0$ ;
- ④ The function  $F$  is  $\varsigma$ -geodesically strongly convex at  $x_*$ , i.e., there exists a neighborhood  $\tilde{\mathcal{U}}_{x_*}$  of  $x_*$  in  $\mathcal{M}$  such that

$$F(y) \geq F(x_*) + \frac{\varsigma}{2} \|\text{Exp}_{x_*}^{-1}(y)\|_F^2$$

holds for any  $y \in \tilde{\mathcal{U}}_{x_*}$ .

## Lemma

*Suppose the last Assumption holds, that is, the function  $F = f + h$  is  $\varsigma$ -geodesically strongly convex at  $x_*$ . Then the linear operator  $\mathcal{B}_{x_*}$  is positive definite on  $\mathfrak{L}_{x_*}$ .*

# A Riemannian proximal Newton-CG method

RPN-CG: local superlinear convergence

Assumption:

- 1 The function  $f$  is twice continuously differentiable with a Lipschitz continuous Euclidean Hessian;
- 2 Let  $B_{x_*}^T = [\bar{B}_{x_*}^T, \hat{B}_{x_*}^T]$ , where  $\bar{B}_{x_*} \in \mathbb{R}^{j \times d}$  and  $\hat{B}_{x_*} \in \mathbb{R}^{(n-j) \times d}$ . It is assumed that  $j \geq d$  and  $\bar{B}_{x_*}$  is full column rank;
- 3 There exists a neighborhood  $\mathcal{U}$  of  $x_* = [\bar{x}_*^T, 0^T]^T$  on  $\mathcal{M}$  such that for any  $x = [\bar{x}^T, \tilde{x}^T]^T \in \mathcal{U}$ , it holds that  $\bar{x} + \bar{v} \neq 0$  and  $\hat{x} + \hat{v} = 0$ ;
- 4 The function  $F$  is  $\varsigma$ -geodesically strongly convex at  $x_*$ , i.e., there exists a neighborhood  $\tilde{\mathcal{U}}_{x_*}$  of  $x_*$  in  $\mathcal{M}$  such that

$$F(y) \geq F(x_*) + \frac{\varsigma}{2} \|\text{Exp}_{x_*}^{-1}(y)\|_F^2$$

holds for any  $y \in \tilde{\mathcal{U}}_{x_*}$ .

## Theorem

*Suppose the previous assumptions hold. If  $x$  is sufficiently close  $x_*$  and the parameters are appropriately chosen, then tCG terminates only due to the accurate condition, i.e.,  $\|r_i\|_F \leq \|r_0\|_F \min(\|r_0\|_F^\theta, \kappa)$ .*

# A Riemannian proximal Newton-CG method

RPN-CG: local superlinear convergence

## Theorem

*Suppose the previous Assumptions hold and the parameters are appropriately chosen. Then there exists a neighborhood of  $x_*$ , denoted by  $\mathcal{V}_8$ , such that if the step size one is used, then the convergence rate is  $\min(1 + \theta, 2)$ , i.e.,  $\|R_x(d(x)) - x_*\|_F \leq C_{\text{up}} \|x - x_*\|_F^{\min(1+\theta, 2)}$  holds for any  $x \in \mathcal{V}_8$  and a constant  $C_{\text{up}} > 0$ .*

# A Riemannian proximal Newton-CG method

RPN-CG: local superlinear convergence

## Theorem

*Suppose the previous Assumptions hold and the parameters are appropriately chosen. Then there exists a neighborhood of  $x_*$ , denoted by  $\mathcal{V}_\theta$ , such that if the step size one is used, then the convergence rate is  $\min(1 + \theta, 2)$ , i.e.,  $\|R_x(d(x)) - x_*\|_F \leq C_{\text{up}} \|x - x_*\|_F^{\min(1+\theta, 2)}$  holds for any  $x \in \mathcal{V}_\theta$  and a constant  $C_{\text{up}} > 0$ .*

Is step size one acceptable for  $x$  sufficiently close to  $x_*$ ?  
That is to make objective function sufficiently descent.



# A Riemannian proximal Newton-CG method

RPN-CG: local superlinear convergence

## Theorem

*Suppose the previous Assumptions hold and the parameters are appropriately chosen. Then there exists a neighborhood of  $x_*$ , denoted by  $\mathcal{V}_8$ , such that if the step size one is used, then the convergence rate is  $\min(1 + \theta, 2)$ , i.e.,  $\|R_x(d(x)) - x_*\|_F \leq C_{\text{up}} \|x - x_*\|_F^{\min(1+\theta, 2)}$  holds for any  $x \in \mathcal{V}_8$  and a constant  $C_{\text{up}} > 0$ .*

Is step size one acceptable for  $x$  sufficiently close to  $x_*$ ?

That is to make objective function sufficiently descent.

- For smooth Riemannian optimization problem, step size one is acceptable eventually for Riemannian Newton method;
- For Euclidean nonsmooth optimization problem  $F = f + g$ , step size one is also acceptable eventually for proximal Newton method [LSS14];

# A Riemannian proximal Newton-CG method

RPN-CG: local superlinear convergence

## Example

- Consider  $F : \mathbb{R}^2 \rightarrow \mathbb{R} : (x_1, x_2)^T \mapsto \underbrace{x_1^2 - 3x_1 + 1 + x_2^2}_{f(x)} + \underbrace{|x_1| + |x_2|}_{g(x)}$ ;
- The unique minimizer:  $x_* = (1, 0)^T$ ;
- $x = (1 + \epsilon, 0)^T$  with  $|\epsilon|$  being arbitrarily small;
- Proximal Newton direction:  $u(x) = -(\epsilon, 0)^T$ ;
- Retraction:  $R : \mathcal{T}\mathcal{M} \rightarrow \mathcal{M} : \eta_x \mapsto x + \eta_x + \begin{pmatrix} 0 \\ 2\eta_x^T \eta_x \end{pmatrix}$ ;
- $R(u(x)) = (1, 2\epsilon^2)^T$ ;
- $F(R_x(u(x))) - F(x) = 4\epsilon^4 + \epsilon^2 > 0$ ;
- Step size one is not acceptable for any  $\epsilon > 0$ ;

# A Riemannian proximal Newton-CG method

RPN-CG: local superlinear convergence

## Example

- Consider  $F : \mathbb{R}^2 \rightarrow \mathbb{R} : (x_1, x_2)^T \mapsto \underbrace{x_1^2 - 3x_1 + 1 + x_2^2}_{f(x)} + \underbrace{|x_1| + |x_2|}_{g(x)}$ ;
- The unique minimizer:  $x_* = (1, 0)^T$ ;
- $x = (1 + \epsilon, 0)^T$  with  $|\epsilon|$  being arbitrarily small;
- Proximal Newton direction:  $u(x) = -(\epsilon, 0)^T$ ;
- Retraction:  $R : \mathcal{T}\mathcal{M} \rightarrow \mathcal{M} : \eta_x \mapsto x + \eta_x + \begin{pmatrix} 0 \\ 2\eta_x^T \eta_x \end{pmatrix}$ ;
- $R(u(x)) = (1, 2\epsilon^2)^T$ ;
- $F(R_x(u(x))) - F(x) = 4\epsilon^4 + \epsilon^2 > 0$ ;
- Step size one is not acceptable for any  $\epsilon > 0$ ;

The answer is negative for nonsmooth Riemannian problems.

Difficulty comes from the nonsmoothness and the curvature.

# A Riemannian proximal Newton-CG method

RPN-CG: local superlinear convergence

Two consecutive iterations near  $x_*$  guarantee sufficient descent.

## Theorem

*Suppose that the previous Assumptions hold and that there exists a neighborhood of  $x_*$ , denoted by  $\mathcal{V}_9$ , such that for any  $x \in \mathcal{V}_9$ , it holds that  $\|R_x(d(x)) - x_*\|_F \leq C_{\text{up}}\|x - x_*\|_F^\varkappa$  for a  $\varkappa > \sqrt{2}$  and  $R_x(d(x)) \in \mathcal{V}_9$ . Then there exists a neighborhood of  $x_*$ , denoted by  $\mathcal{V}_{10}$ , and a constant  $\rho_1 > 0$  such that for any  $x \in \mathcal{V}_{10}$ , it holds that*

$$F(x_{++}) \leq F(x) - \rho_1 \|v(x)\|_F^2,$$

*where  $x_+ = R_x(d(x))$  and  $x_{++} = R_{x_+}(d(x_+))$ .*

# A Riemannian proximal Newton-CG method

RPN-CG: local superlinear convergence

Two consecutive iterations near  $x_*$  guarantee sufficient descent.

## Theorem

*Suppose that the previous Assumptions hold and that there exists a neighborhood of  $x_*$ , denoted by  $\mathcal{V}_9$ , such that for any  $x \in \mathcal{V}_9$ , it holds that  $\|R_x(d(x)) - x_*\|_F \leq C_{\text{up}}\|x - x_*\|_F^\varkappa$  for a  $\varkappa > \sqrt{2}$  and  $R_x(d(x)) \in \mathcal{V}_9$ . Then there exists a neighborhood of  $x_*$ , denoted by  $\mathcal{V}_{10}$ , and a constant  $\rho_1 > 0$  such that for any  $x \in \mathcal{V}_{10}$ , it holds that*

$$F(x_{++}) \leq F(x) - \rho_1 \|v(x)\|_F^2,$$

*where  $x_+ = R_x(d(x))$  and  $x_{++} = R_{x_+}(d(x_+))$ .*

The global convergence result becomes:  $\liminf_{k \rightarrow \infty} \|v(x_k)\|_F = 0$ .

## A new interpretation of RPN

### Lemma

*Suppose the previous Assumptions hold. Then there exists a neighborhood of  $x_*$ , denoted by  $\mathcal{V}_5$ , such that*

$$u(x) = \operatorname{argmin}_{u \in T_x \mathcal{M}, \hat{u} = \hat{v}(x)} G_x(u) = \frac{1}{2} \langle u, \mathfrak{B}_x u \rangle + \nabla f(x)^T u + \mu \|x + u\|_1 \quad (1)$$

*holds for any  $x \in \mathcal{V}_5$ .*

- First, find the ManPG search direction  $v(x)$ ;
- Fixed the entries that corresponds to the zero of  $x + v$ ;
- Solve (1) for  $u(x)$ ;

## A new interpretation of RPN

### Lemma

*Suppose the previous Assumptions hold. Then there exists a neighborhood of  $x_*$ , denoted by  $\mathcal{V}_5$ , such that*

$$u(x) = \operatorname{argmin}_{u \in T_x \mathcal{M}, \hat{u} = \hat{v}(x)} G_x(u) = \frac{1}{2} \langle u, \mathfrak{B}_x u \rangle + \nabla f(x)^T u + \mu \|x + u\|_1 \quad (1)$$

*holds for any  $x \in \mathcal{V}_5$ .*

- $\mathcal{M}_{sub}$ : submanifold of the intersection of  $\mathcal{M}$  and the sparse constraints;
- $\mathfrak{B}_x^{(11)}$  is the Riemannian Hessian at  $x$  with respect to  $\mathcal{M}_{sub}$ ;
- $u(x)$  is the Riemannian Newton direction on  $\mathcal{M}_{sub}$ ;

## A new interpretation of RPN

### Lemma

*Suppose the previous Assumptions hold. Then there exists a neighborhood of  $x_*$ , denoted by  $\mathcal{V}_5$ , such that*

$$u(x) = \operatorname{argmin}_{u \in T_x \mathcal{M}, \hat{u} = \hat{v}(x)} G_x(u) = \frac{1}{2} \langle u, \mathfrak{B}_x u \rangle + \nabla f(x)^T u + \mu \|x + u\|_1 \quad (1)$$

*holds for any  $x \in \mathcal{V}_5$ .*

- $\mathcal{M}_{sub}$ : submanifold of the intersection of  $\mathcal{M}$  and the sparse constraints;
- $\mathfrak{B}_x^{(11)}$  is the Riemannian Hessian at  $x$  with respect to  $\mathcal{M}_{sub}$ ;
- $u(x)$  is the Riemannian Newton direction on  $\mathcal{M}_{sub}$ ;

No counterpart in the Euclidean space.



# A Riemannian proximal Newton-CG method

Numerical experiments: sparse PCA

Sparse PCA problem

$$\min_{X \in \text{St}(p, n)} -\text{trace}(X^T A^T A X) + \mu \|X\|_1,$$

where  $A \in \mathbb{R}^{m \times n}$  is a data matrix and

$\text{St}(p, n) = \{X \in \mathbb{R}^{n \times p} \mid X^T X = I_p\}$  is the compact Stiefel manifold.

# A Riemannian proximal Newton-CG method

Numerical experiments: sparse PCA

**Table:** An average result of 20 random runs for random data. Multiple values of  $n$ ,  $p$ , and  $\mu$  are used. The subscript  $k$  indicates a scale of  $10^k$ .

$(n, p, \mu)$	Algo	iter	Fval	$\ v(x_k)\ _F$	time	sparsity
(400, 8, 0.8)	ManPG	3416.15	$-2.16_1$	$3.66_{-9}$	2.69	0.63
(400, 8, 0.8)	ManPG-Ada	1281.55	$-2.16_1$	$1.06_{-10}$	1.21	0.63
(400, 8, 0.8)	ManPQN	1260.40	$-2.16_1$	$9.83_{-11}$	0.72	0.63
(400, 8, 0.8)	RPN-CG	204.85	$-2.16_1$	$1.16_{-11}$	0.37	0.63
(800, 8, 0.8)	ManPG	4232.80	$-5.92_1$	$1.84_{-7}$	3.56	0.48
(800, 8, 0.8)	ManPG-Ada	1867.05	$-5.92_1$	$2.57_{-10}$	1.80	0.48
(800, 8, 0.8)	ManPQN	1883.80	$-5.92_1$	$1.22_{-10}$	1.43	0.48
(800, 8, 0.8)	RPN-CG	215.05	$-5.92_1$	$1.07_{-11}$	0.60	0.48

# A Riemannian proximal Newton-CG method

Numerical experiments: sparse PCA

**Table:** An average result of 20 random runs for random data. Multiple values of  $n$ ,  $p$ , and  $\mu$  are used. The subscript  $k$  indicates a scale of  $10^k$ .

$(n, p, \mu)$	Algo	iter	Fval	$\ v(x_k)\ _F$	time	sparsity
(400, 8, 0.8)	ManPG	3416.15	$-2.16_1$	$3.66_{-9}$	2.69	0.63
(400, 8, 0.8)	ManPG-Ada	1281.55	$-2.16_1$	$1.06_{-10}$	1.21	0.63
(400, 8, 0.8)	ManPQN	1260.40	$-2.16_1$	$9.83_{-11}$	0.72	0.63
(400, 8, 0.8)	RPN-CG	204.85	$-2.16_1$	$1.16_{-11}$	0.37	0.63
(800, 8, 0.8)	ManPG	4232.80	$-5.92_1$	$1.84_{-7}$	3.56	0.48
(800, 8, 0.8)	ManPG-Ada	1867.05	$-5.92_1$	$2.57_{-10}$	1.80	0.48
(800, 8, 0.8)	ManPQN	1883.80	$-5.92_1$	$1.22_{-10}$	1.43	0.48
(800, 8, 0.8)	RPN-CG	215.05	$-5.92_1$	$1.07_{-11}$	0.60	0.48

- Proximal gradient on Stiefel manifold: ManPG, ManPG-Ada [CMSZ20];
- Proximal quasi-Newton on Stiefel manifold: ManPQN [WY23];
- The proposed method: RPN-CG;

# A Riemannian proximal Newton-CG method

Numerical experiments: sparse PCA

**Table:** An average result of 20 random runs for random data. Multiple values of  $n$ ,  $p$ , and  $\mu$  are used. The subscript  $k$  indicates a scale of  $10^k$ .

$(n, p, \mu)$	Algo	iter	Fval	$\ v(x_k)\ _F$	time	sparsity
(400, 8, 0.8)	ManPG	3416.15	$-2.16_1$	$3.66_{-9}$	2.69	0.63
(400, 8, 0.8)	ManPG-Ada	1281.55	$-2.16_1$	$1.06_{-10}$	1.21	0.63
(400, 8, 0.8)	ManPQN	1260.40	$-2.16_1$	$9.83_{-11}$	0.72	0.63
(400, 8, 0.8)	RPN-CG	204.85	$-2.16_1$	$1.16_{-11}$	0.37	0.63
(800, 8, 0.8)	ManPG	4232.80	$-5.92_1$	$1.84_{-7}$	3.56	0.48
(800, 8, 0.8)	ManPG-Ada	1867.05	$-5.92_1$	$2.57_{-10}$	1.80	0.48
(800, 8, 0.8)	ManPQN	1883.80	$-5.92_1$	$1.22_{-10}$	1.43	0.48
(800, 8, 0.8)	RPN-CG	215.05	$-5.92_1$	$1.07_{-11}$	0.60	0.48

- Stop criterion:  $\text{iter} \geq 5000$  or  $\|v(x)\|_F \leq 10^{-10}$ ;
- The entries of  $A$  are drawn from the standard normal distribution;
- Runs that converges to the same minimizer are reported;
- Support estimation:  $(x + v(x))_i$  nonzero and  $|(x)_i| \geq \|v(x)\|_F$ ;

# A Riemannian proximal Newton-CG method

Numerical experiments: sparse PCA

**Table:** An average result of 20 random runs for random data. Multiple values of  $n$ ,  $p$ , and  $\mu$  are used. The subscript  $k$  indicates a scale of  $10^k$ .

$(n, p, \mu)$	Algo	iter	Fval	$\ v(x_k)\ _F$	time	sparsity
(400, 8, 0.8)	ManPG	3416.15	$-2.16_1$	$3.66_{-9}$	2.69	0.63
(400, 8, 0.8)	ManPG-Ada	1281.55	$-2.16_1$	$1.06_{-10}$	1.21	0.63
(400, 8, 0.8)	ManPQN	1260.40	$-2.16_1$	$9.83_{-11}$	0.72	0.63
(400, 8, 0.8)	RPN-CG	204.85	$-2.16_1$	$1.16_{-11}$	0.37	0.63
(800, 8, 0.8)	ManPG	4232.80	$-5.92_1$	$1.84_{-7}$	3.56	0.48
(800, 8, 0.8)	ManPG-Ada	1867.05	$-5.92_1$	$2.57_{-10}$	1.80	0.48
(800, 8, 0.8)	ManPQN	1883.80	$-5.92_1$	$1.22_{-10}$	1.43	0.48
(800, 8, 0.8)	RPN-CG	215.05	$-5.92_1$	$1.07_{-11}$	0.60	0.48

RPN-CG always stops due to  $\|v\|_F \leq 10^{-10}$   
and is the most efficient one.

# A Riemannian proximal Newton-CG method

Numerical experiments: sparse PCA

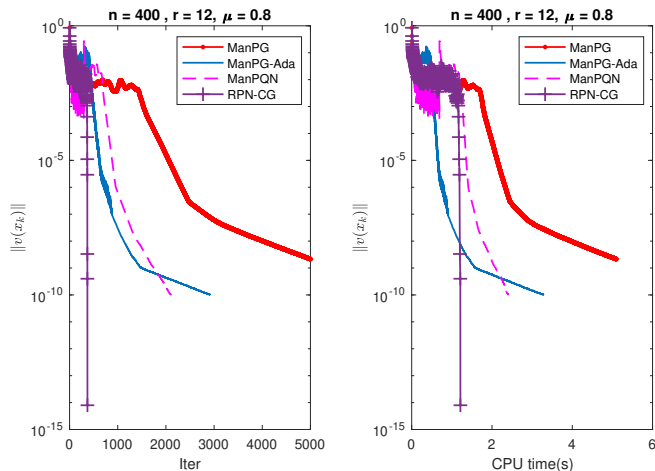


Figure: Sparse PCA: plots of  $\|v(x_k)\|$  versus iterations and CPU times respectively.

# Numerical Experiments

## Compressed modes

The compressed modes (CM) problem aims to seek sparse solution of the independent-particle Schrödinger equation. It can be formulated as

$$\min_{X \in \text{St}(p, n)} \text{trace}(X^T H X) + \mu \|X\|_1,$$

where  $H \in \mathbb{R}^{n \times n}$  denotes the discretized Schrödinger operator.

# A Riemannian proximal Newton-CG method

Numerical experiments: compressed modes

**Table:** An average result of 50 random runs for random data. Multiple values of  $n$ ,  $p$ , and  $\mu$  are used. The subscript  $k$  indicates a scale of  $10^k$ .

$(n, p, \mu)$	Algo	iter	Fval	$\ v(x_k)\ _F$	time	sparsity
(256, 4, 0.1)	ManPG	3000.00	2.49	4.03 <sub>-5</sub>	0.75	0.85
(256, 4, 0.1)	ManPG-Ada	3000.00	2.49	9.49 <sub>-5</sub>	0.88	0.85
(256, 4, 0.1)	ManPQN	3000.00	2.49	9.06 <sub>-6</sub>	1.22	0.84
(256, 4, 0.1)	RPN-CG	92.54	2.49	2.66 <sub>-9</sub>	0.20	0.86
(512, 4, 0.1)	ManPG	3000.00	3.29	3.83 <sub>-5</sub>	0.76	0.86
(512, 4, 0.1)	ManPG-Ada	3000.00	3.29	1.16 <sub>-4</sub>	0.88	0.86
(512, 4, 0.1)	ManPQN	3000.00	3.30	1.44 <sub>-6</sub>	2.98	0.86
(512, 4, 0.1)	RPN-CG	147.40	3.29	2.29 <sub>-9</sub>	0.48	0.88

- Stop criterion:  $\text{iter} \geq 3000$  or  $\|v(x)\|_F \leq 10^{-8}$ ;
- Different runs may converge to different points;



# A Riemannian proximal Newton-CG method

Numerical experiments: compressed modes

**Table:** An average result of 50 random runs for random data. Multiple values of  $n$ ,  $p$ , and  $\mu$  are used. The subscript  $k$  indicates a scale of  $10^k$ .

$(n, p, \mu)$	Algo	iter	Fval	$\ v(x_k)\ _F$	time	sparsity
(256, 4, 0.1)	ManPG	3000.00	2.49	$4.03_{-5}$	0.75	0.85
(256, 4, 0.1)	ManPG-Ada	3000.00	2.49	$9.49_{-5}$	0.88	0.85
(256, 4, 0.1)	ManPQN	3000.00	2.49	$9.06_{-6}$	1.22	0.84
(256, 4, 0.1)	RPN-CG	92.54	2.49	$2.66_{-9}$	0.20	0.86
(512, 4, 0.1)	ManPG	3000.00	3.29	$3.83_{-5}$	0.76	0.86
(512, 4, 0.1)	ManPG-Ada	3000.00	3.29	$1.16_{-4}$	0.88	0.86
(512, 4, 0.1)	ManPQN	3000.00	3.30	$1.44_{-6}$	2.98	0.86
(512, 4, 0.1)	RPN-CG	147.40	3.29	$2.29_{-9}$	0.48	0.88

RPN-CG always stops due to  $\|v\|_F \leq 10^{-8}$   
and is the most efficient one.

None of other methods find a solution with  $\|v\|_F \leq 10^{-8}$ .

# A Riemannian proximal Newton-CG method

Numerical experiments: compressed modes

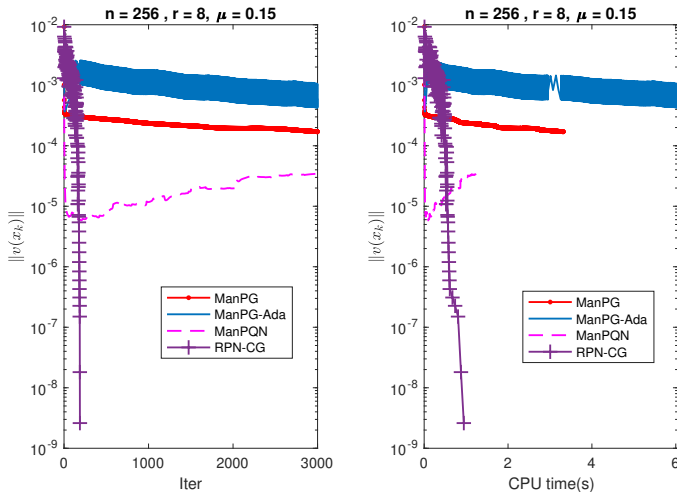


Figure: CM: plots of  $\|v(x_k)\|$  versus iterations and CPU times respectively.

# Summary

- Riemannian optimization;
- Applications;
  - An example on an embedded submanifold;
  - An example on a quotient manifold;
- Smooth optimization framework;
  - Search direction/Riemannian metric;
  - Riemannian gradient/Hessian;
  - Retraction/vector transport;
- Research foci of Riemannian optimization;
  - Manifold recognition/structures;
  - Algorithm generalizations;
  - Applications/Libraries;
- A Riemannian proximal Newton-CG method;
  - A Riemannian proximal Newton method;
  - Truncated conjugate gradient;
  - Superlinear convergence approach;
  - Numerical experiments;
- Summary;

# Thank you

Thank you!

# References I



Ognjen Arandjelovic, Gregory Shakhnarovich, John Fisher, Roberto Cipolla, and Trevor Darrell.

Face recognition with image sets using manifold density divergence.

In *Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on*, volume 1, pages 581–588. IEEE, 2005.



Matthias Bollh ofer, Aryan Eftekhari, Simon Scheidegger, and Olaf Schenk.

Large-scale sparse inverse covariance matrix estimation.

*SIAM Journal on Scientific Computing*, 41(1):A380–A401, 2019.



Shixiang Chen, Shiqian Ma, Anthony Man-Cho So, and Tong Zhang.

Proximal gradient method for nonsmooth optimization over the Stiefel manifold.

*SIAM Journal on Optimization*, 30(1):210–239, 2020.



Haoran Chen, Yanfeng Sun, Junbin Gao, Yongli Hu, and Baocai Yin.

Fast optimization algorithm on riemannian manifolds and its application in low-rank learning.

*Neurocomputing*, 291:59 – 70, 2018.



Guang Cheng, Hesamoddin Salehian, and Baba Vemuri.

Efficient recursive algorithms for computing the mean diffusion tensor and applications to DTI segmentation.

*Computer Vision–ECCV 2012*, pages 390–401, 2012.



H. Drira, B. Ben Amor, A. Srivastava, M. Daoudi, and R. Slama.

3D face recognition under expressions, occlusions, and pose variations.

*Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 35(9):2270–2283, 2013.



P. T. Fletcher and S. Joshi.

Riemannian geometry for the statistical analysis of diffusion tensor data.

*Signal Processing*, 87(2):250–262, 2007.



W. Huang, K. A. Gallivan, Anuj Srivastava, and P.-A. Absil.

Riemannian optimization for registration of curves in elastic shape analysis.

*Journal of Mathematical Imaging and Vision*, 54(3):320–343, 2015.

DOI:10.1007/s10851-015-0606-8.

# References II



Wen Huang, Meng Wei, Kyle A. Gallivan, and Paul Van Dooren.  
A Riemannian Optimization Approach to Clustering Problems, 2022.



Zhiwu Huang, Ruiping Wang, Shiguang Shan, and Xilin Chen.  
Face recognition on large-scale video in the wild with hybrid Euclidean-and-Riemannian metric learning.  
*Pattern Recognition*, 48(10):3113–3124, 2015.



H. Laga, S. Kurtsek, A. Srivastava, M. Golzarian, and S. J. Miklavcic.  
A Riemannian elastic metric for shape-based plant leaf classification.  
*2012 International Conference on Digital Image Computing Techniques and Applications (DICTA)*, pages 1–7, December 2012.  
doi:10.1109/DICTA.2012.6411702.



Jason D Lee, Yuekai Sun, and Michael A Saunders.  
Proximal newton-type methods for minimizing composite functions.  
*SIAM Journal on Optimization*, 24(3):1420–1443, 2014.



Jiwen Lu, Gang Wang, and Pierre Moulin.  
Image set classification using holistic multiple order statistics features and localized multi-kernel metric learning.  
*In Proceedings of the IEEE International Conference on Computer Vision*, pages 329–336, 2013.



Vidvuds Ozoliņš, Rongjie Lai, Russel Caflisch, and Stanley Osher.  
Compressed modes for variational problems in mathematics and physics.  
*Proceedings of the National Academy of Sciences*, 110(46):18368–18373, 2013.



Y. Rathi, A. Tannenbaum, and O. Michailovich.  
Segmenting images on the tensor manifold.  
*In IEEE Conference on Computer Vision and Pattern Recognition*, pages 1–8, June 2007.



Wutao Si, P. A. Absil, Wen Huang, Rujun Jiang, and Simon Vary.  
A Riemannian Proximal Newton Method, 2024.  
arXiv: 2304.04032v3.

# References III



Wutao Si, P.-A. Absil, Wen Huang, Rujun Jiang, and Simon Vary.

A Riemannian Proximal Newton Method.

*SIAM Journal on Optimization*, 34(1):654–681, 2024.



Gregory Shakhnarovich, John W Fisher, and Trevor Darrell.

Face recognition from long-term observations.

In *European Conference on Computer Vision*, pages 851–865. Springer, 2002.



Oncel Tuzel, Fatih Porikli, and Peter Meer.

Region covariance: A fast descriptor for detection and classification.

In *European conference on computer vision*, pages 589–600. Springer, 2006.



Qinsi Wang and Weihong Yang.

Proximal quasi-Newton method for composite optimization over the Stiefel manifold.

*Journal of Scientific Computing*, 95, 5 2023.



Hui Zou, Trevor Hastie, and Robert Tibshirani.

Sparse principal component analysis.

*Journal of Computational and Graphical Statistics*, 15(2):265–286, 2006.



Y. Zhang, Y. Lau, H.-W. Kuo, S. Cheung, A. Pasupathy, and J. Wright.

On the global geometry of sphere-constrained sparse blind deconvolution.

In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017.