Riemannian Optimization with its Application to Solving Generalized Lyapunov Equations

Speaker: Wen Huang

Xiamen University

October 16, 2025

Hebei University of Technology

Outline

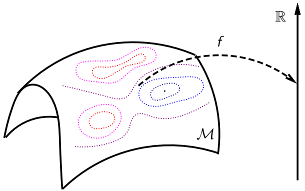
- Riemannian optimization;
- Applications;
- Smooth optimization framework;
- Research foci of Riemannian optimization;
- Application: generalized Lyapunov equations;
- Summary;

Riemannian Optimization

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

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

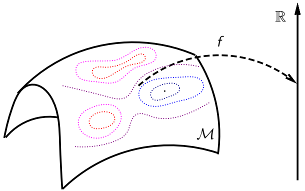
where $\mathcal{M}$ is a Riemannian manifold.



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

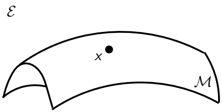
$$\min_{x \in \mathcal{M}} f(x)$$

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

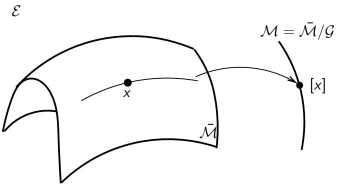


Two kinds of commonly-encountered manifolds

Embedded submanifold of a Euclidean space



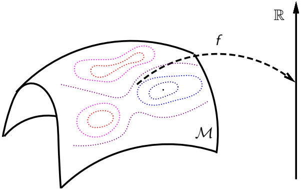
Quotient manifold from an embedded submanifold



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

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

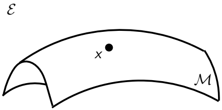
where $\mathcal{M}$ is a Riemannian manifold.



Examples:

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

Embedded submanifold of a Euclidean space

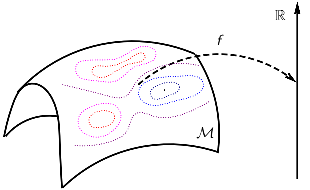


Riemannian Optimization

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

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

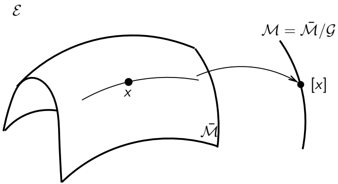
where $\mathcal{M}$ is a Riemannian manifold.



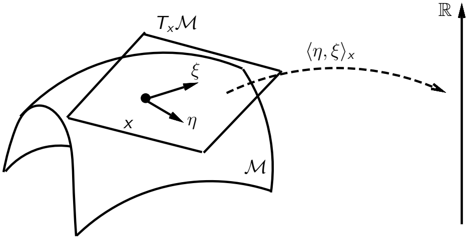
Examples:

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

Quotient manifold from an embedded submanifold



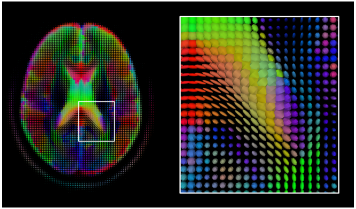
Roughly, a Riemannian manifold $\mathcal{M}$ is a smooth set with a smoothly-varying inner product on the tangent spaces.



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$^+$05, TPM06, HWSC15]
- Motivation of averaging SPD matrices
  - denoising / interpolation
  - clustering / classification



Embedded submanifold: Computation on SPD manifold

One averaging SPD matrices method:

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

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

Embedded submanifold: Computation on SPD manifold

One averaging SPD matrices method:

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

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

Why shall we use Riemannian optimization approach?

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

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

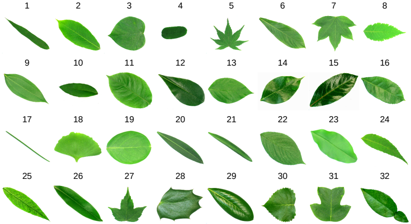
Condition number of the Riemannian Hessian [YHAG2020]

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

- $\frac{\kappa^2(\mu)}{\kappa(H^{\mathrm{R}})} \leq \kappa(H^{\mathrm{E}}) \leq \kappa(H^{\mathrm{R}}) \kappa^2(\mu)$
- $\kappa(H^E) \geq \kappa^2(\mu)/20$

[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$^+$12, HGSA15]
- Face recognition [DBS$^+$13]

Quotient manifold: Computation on shape space

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

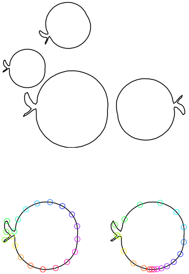
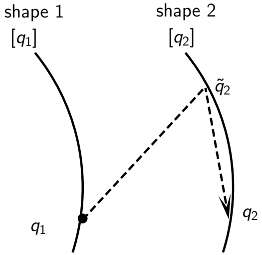
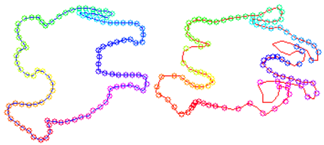


Figure: All are the same shape.

Quotient manifold: Computation on shape space

Registration



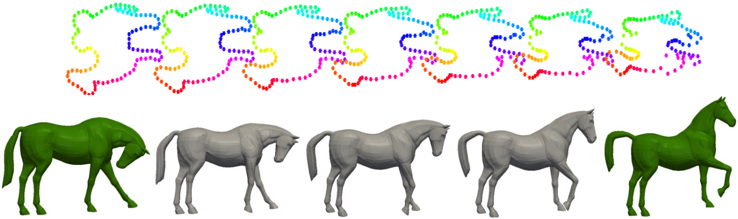


- Optimization problem $\min_{q_2 \in [q_2]} \mathrm{dist}(q_1, q_2)$ is defined on a Riemannian manifold

Applications

Quotient manifold: Computation on shape space

Geodesic / Interpolation



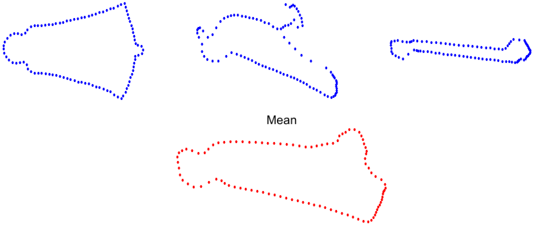
$$\min_{\alpha \in \mathcal{H}_{x,y}} \frac{1}{2} \int_0^1 \langle \dot{\alpha}(\tau), \dot{\alpha}(\tau) \rangle_{\alpha(\tau)} \mathrm{d}\tau$$

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

Applications

Quotient manifold: Computation on shape space

Karcher mean



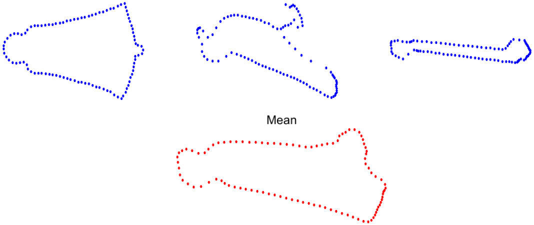
$$\min_{X \text{ is a shape}} \frac{1}{2k} \sum_{i=1}^{k} \mathrm{dist}^2(X, S_i),$$

- Computation of Karcher mean of a population of shapes

Applications

Quotient manifold: Computation on shape space

Karcher mean



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

- 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:

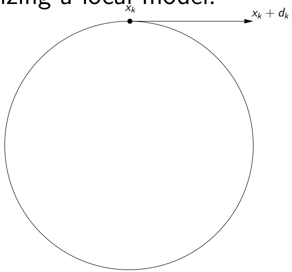
$$x_{k+1} = x_k + \Delta x_k = x_k + \alpha_k s_k \ .$$

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: $x_{k+1} = x_k - \left[\nabla^2 f(x_k)\right]^{-1} \nabla f(x_k)$
- Trust region method: Δx_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



Definition

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

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

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

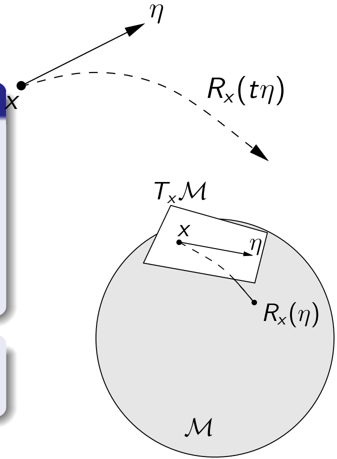
Definition

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

$$\mathrm{Hess} f(x) : \mathrm{T}_x \mathcal{M} \to \mathrm{T}_x \mathcal{M} : \eta \to \nabla_\eta \mathrm{grad} f,$$

where ∇ is the affine connection.

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 $\mathrm{T}\mathcal{M}$ to $\mathcal{M}$ satisfying the following:

- R is continuously differentiable
- $R_x(0) = x$
- $\mathrm{D}\, R_x(0)[\eta] = \eta$

- maps tangent vectors back to the manifold
- defines curves in a direction

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)

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.

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 ξ_x to tangent space of $R_x(\eta_x)$. R is a retraction associated with $\mathcal{T}$;

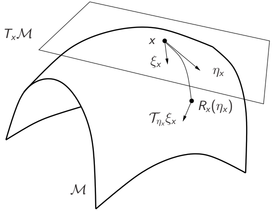


Figure: Vector transport.

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

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;

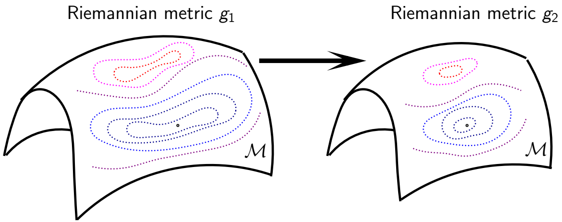


Figure: Changing Riemannian metric may influence the difficulty of a problem.

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, Adragni (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

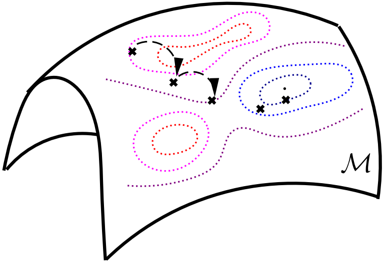
[YHAG2020] X. Yuan, W. Huang, P.-A. Absil, K. A. Gallivan. Computing the matrix geometric mean: Riemannian vs Euclidean conditioning, implementation techniques, and a Riemannian BFGS method, *Numerical Linear Algebra with Applications*, 27:5, 1-23, 2020

[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;
  - ϵ-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;

Generalized Lyapunov equation: Given matrix A, M and C, find X such that

$$AXM^T + MXA^T = C \tag{1}$$

Applications: signal processing, model reduction, and system and control theory [Moo03, Ben06]

Generalized Lyapunov equation: Given matrix A, M and C, find X such that

$$AXM^T + MXA^T = C \tag{1}$$

Applications: signal processing, model reduction, and system and control theory [Moo03, Ben06]

Problem: We focus on the problem:

- $A, M, C \in \mathbb{R}^{n\times n}$ are symmetric;
- $A \succ 0, M \succ 0$ (positive definite), $C \succeq 0$ (positive semidefinite);
- A, M are sparse;
- medium- to large-scale problems;

$A \succ 0, M \succ 0$ and $C \succeq 0$, A, M, and C are symmetric:

$$AXM + MXA - C = 0$$

- X is not sparse, even A and M are sparse;
- How to solve it for large-scale problems?

$A \succ 0, M \succ 0$ and $C \succeq 0$, A, M, and C are symmetric:

$$AXM + MXA - C = 0$$

- X is not sparse, even A and M are sparse;
- How to solve it for large-scale problems? Low rank solution

$A \succ 0, M \succ 0$ and $C \succeq 0$, A, M, and C are symmetric:

$$AXM + MXA - C = 0$$

- X is not sparse, even A and M are sparse;
- How to solve it for large-scale problems? Low rank solution
- Reasonable: For low rank C, the solution X has low numerical rank [Pen00b]

$A \succ 0, M \succ 0$ and $C \succeq 0$, A, M, and C are symmetric:

$$AXM + MXA - C = 0$$

Unique solution X and $X = X^T, X \succeq 0$ [Pen98] $\Longrightarrow X = YY^T$

$A \succ 0, M \succ 0$ and $C \succeq 0$, A, M, and C are symmetric:

$$AXM + MXA - C = 0$$

Unique solution X and $X = X^T, X \succeq 0$ [Pen98] $\Longrightarrow X = YY^T$

- Alternating Direction Implicit Iteration (ADI) or Smith method;
- Krylov subspace technique;
- Optimization method;

$A \succ 0, M \succ 0$ and $C \succeq 0$, A, M, and C are symmetric:

$$AXM + MXA - C = 0$$

Unique solution X and $X = X^T, X \succeq 0$ [Pen98] $\Longrightarrow X = YY^T$

- Alternating Direction Implicit Iteration (ADI) or Smith method;
- Krylov subspace technique;

Reformulate well-known iterative method to a low-rank setting. Work on the factor Y of $X = YY^T$.

$A \succ 0, M \succ 0$ and $C \succeq 0$, A, M, and C are symmetric:

$$AXM + MXA - C = 0$$

Unique solution X and $X = X^T, X \succeq 0$ [Pen98] $\Longrightarrow X = YY^T$

- Alternating Direction Implicit Iteration (ADI) or Smith method;
- Krylov subspace technique;
- Optimization method;

- Consider a cost function on the set of symmetric matrices:
  - Cost function: $F : \mathbb{S}^{n \times n} \to \mathbb{R} : X \mapsto \mathrm{trace}(XAXM) - \mathrm{trace}(XC)$;
  - Gradient: $AXM + MXA - C$;
  - The critical point is unique [Pen98].
  - The minimizer is the solution.

[VV10]: B. Vandereycken and S. Vandewalle, A Riemannian optimization approach for computing low-rank solutions of Lyapunov equations, SIAM Journal on Matrix Analysis and Applications, 31(5):2553-2579, 2010.

- Consider a cost function on the set of symmetric matrices:
  - Cost function: $F : \mathbb{S}^{n \times n} \to \mathbb{R} : X \mapsto \mathrm{trace}(XAXM) - \mathrm{trace}(XC)$;
  - Gradient: $AXM + MXA - C$;
  - The critical point is unique [Pen98].
  - The minimizer is the solution.
- Add low-rank constraints by fixing the rank to be r:
  - Cost function: $f : \mathbb{S}_r^{n \times n} \to \mathbb{R} : X \mapsto \mathrm{trace}(XAXM) - \mathrm{trace}(XC)$;
  - Gradient: $P_{\mathrm{T}_X \mathbb{S}_r^{n \times n}}(AXM + MXA - C)$;
  - Minimizer can be viewed as a low-rank approximation of the solution;

[VV10]: B. Vandereycken and S. Vandewalle, A Riemannian optimization approach for computing low-rank solutions of Lyapunov equations, SIAM Journal on Matrix Analysis and Applications, 31(5):2553-2579, 2010.

Optimization problem on the symmetric positive semidefinite with rank r

$$\min_{X \in \mathbb{S}_r^{n \times n}} f(X) = \mathrm{trace}(XAXM) - \mathrm{trace}(XC)$$

- Ingredients for Riemannian optimization;
- Trust-region Newton method
- Preconditioner

- Tangent space at $X = YY^T$ is

$$\mathrm{T}_X\,\mathbb{S}_r^{n\times n} = \left\{ \begin{bmatrix} Y & Y_\perp \end{bmatrix} \begin{bmatrix} 2S & N^T \\ N & 0 \end{bmatrix} \begin{bmatrix} Y^T \\ Y_\perp^T \end{bmatrix} \mid S \in \mathbb{S}^{r\times r}, N \in \mathbb{R}^{(n-r)\times r} \right\}$$
$$= \left\{ YZ^T + ZY^T \mid Z \in \mathbb{R}^{n\times r} \right\};$$

- Tangent space at $X = YY^T$ is $\{YZ^T + ZY^T \mid Z \in \mathbb{R}^{n\times r}\}$;
- Riemannian metric:

$$g_X(\eta_X, \xi_X) = \mathrm{trace}(\eta_X^T \xi_X).$$

for any $\eta_X, \xi_X \in \mathrm{T}_X\, \mathbb{S}_r^{n\times n}$;

- Tangent space at $X = YY^T$ is $\left\{YZ^T + ZY^T \mid Z \in \mathbb{R}^{n\times r}\right\}$;
- Riemannian metric: $g_X(\eta_X, \xi_X) = \mathrm{trace}(\eta_X^T \xi_X)$;
- Retraction:

$$R_X(\eta_X) = P_{\mathbb{S}_r^{n\times n}}(X + \eta_X),$$

where $P_{\mathbb{S}_r^{n\times n}}(Z) = \sum_{i=1}^r \sigma_i v_i v_i^T$, $Z = V\Sigma V$, $V = [v_1, \ldots, v_n]$, $\Sigma = \mathrm{diag}(\sigma_1, \ldots, \sigma_n)$ and $\sigma_1 \geq \sigma_2 \geq \ldots \geq \sigma_n \geq 0$.

- Tangent space at $X = YY^T$ is $\left\{YZ^T + ZY^T \mid Z \in \mathbb{R}^{n\times r}\right\}$;
- Riemannian metric: $g_X(\eta_X, \xi_X) = \mathrm{trace}(\eta_X^T \xi_X)$;
- Retraction: $R_X(\eta_X) = P_{\mathbb{S}_r^{n\times n}}(X + \eta_X)$;
- Riemannian gradient:

$$\mathrm{grad}\, f(X) = P_{\mathrm{T}_X\, \mathbb{S}_r^{n\times n}}(AXM + MXA - C),$$

where $P_{\mathrm{T}_X\, \mathbb{S}_r^{n\times n}}(Z) = P_Y Z P_Y + P_Y^{\perp} Z P_Y + P_Y Z P_Y^{\perp}$, $P_Y^{\perp} = I - P_Y$ and $P_Y = Y(Y^T Y)^{-1} Y^T$;

- Tangent space at $X = YY^T$ is $\left\{YZ^T + ZY^T \mid Z \in \mathbb{R}^{n\times r}\right\}$;
- Riemannian metric: $g_X(\eta_X, \xi_X) = \mathrm{trace}(\eta_X^T \xi_X)$;
- Retraction: $R_X(\eta_X) = P_{\mathbb{S}_r^{n\times n}}(X + \eta_X)$;
- Riemannian gradient: $\mathrm{grad}\, f(X) = P_{\mathrm{T}_X\, \mathbb{S}_r^{n\times n}}(AXM + MXA - C)$;
- Action of the Riemannian Hessian:

$$
\begin{aligned}
\mathrm{Hess}\, f(X)[\eta_X] =& P_{\mathrm{T}_X\, \mathbb{S}_r^{n\times n}}(A\eta_X M + M\eta_X A) \\
&+ P_{\mathrm{T}_X\, \mathbb{S}_r^{n\times n}}\left(\mathrm{D}\, P_{\mathrm{T}_X\, \mathbb{S}_r^{n\times n}}[\eta_X](AXM + MXA - C)\right)
\end{aligned}
$$

1: **for** $k = 0, 1, 2, \ldots$ **do**
2: Let $m_k(\eta) = f(X_k) + g_{X_k}(\mathrm{grad}\, f(X_k), \eta) + \frac{1}{2} g_{X_k}(\mathrm{Hess}\, f(X_k)[\eta], \eta)$;
3: Obtain η_k by approximately solving $\min_{\eta \in \mathrm{T}_{X_k} \mathbb{S}_r^{n \times n}, \|\eta\| \leq \Delta_k} m_k(\eta)$;
4: Compute $\rho_k = \frac{f(X_k) - f(R_{X_k}(\eta_k))}{m_k(0) - m_k(\eta_k)}$;
5: Set $X_{k+1} = R_{X_k}(\eta_k)$ if ρ_k is sufficient large, Otherwise $X_{k+1} = X_k$;
6: Set $\Delta_{k+1} = 2\Delta_k$ if ρ_k is sufficient large;
7: Set $\Delta_{k+1} = \Delta_k / 4$ if ρ_k is small;
8: **end for**

- Build a local quadratic model;
- Solve the local model approximately by truncated CG;
- Accept the candidate if the local model is good enough;
- Update the radius of the trust region;

Solving Generalized Lyapunov Equations

Riemannian Trust-region Newton method

---

for $k = 0, 1, 2, \ldots$ **do**
  Let $m_k(\eta) = f(X_k) + g_{X_k}(\mathrm{grad}\, f(X_k), \eta) + \frac{1}{2} g_{X_k}(\mathrm{Hess}\, f(X_k)[\eta], \eta)$;
  Obtain η_k by approximately solving $\min_{\eta \in \mathrm{T}_{X_k} \mathbb{S}_r^{n \times n}, \|\eta\| \leq \Delta_k} m_k(\eta)$;
  Compute $\rho_k = \frac{f(X_k) - f(R_{X_k}(\eta_k))}{m_k(0) - m_k(\eta_k)}$;
  Set $X_{k+1} = R_{X_k}(\eta_k)$ if ρ_k is sufficient large, Otherwise $X_{k+1} = X_k$;
  Set $\Delta_{k+1} = 2\Delta_k$ if ρ_k is sufficient large;
  Set $\Delta_{k+1} = \Delta_k / 4$ if ρ_k is small;
end for

- Build a local quadratic model;
- Solve the local model approximately by truncated CG;
- Accept the candidate if the local model is good enough;
- Update the radius of the trust region;

(1) RTR-Newton converges quadratically locally; (2) Solving the local model is expensive.

The action of the Riemannian Hessian is

$$
\begin{aligned}
\operatorname{Hess} f(X)[\eta_X] = & P_{\mathrm{T}_X \mathbb{S}_r^{n\times n}}(A\eta_X M + M\eta_X A) \\
& + P_{\mathrm{T}_X \mathbb{S}_r^{n\times n}}\left(\mathrm{D}\, P_{\mathrm{T}_X \mathbb{S}_r^{n\times n}}[\eta_X](AXM + MXA - C)\right)
\end{aligned}
$$

The action of the Riemannian Hessian is

$$\begin{aligned}\operatorname{Hess} f(X)[\eta_X] = & P_{\mathrm{T}_X \mathbb{S}_r^{n\times n}}(A\eta_X M + M\eta_X A) \\ & + P_{\mathrm{T}_X \mathbb{S}_r^{n\times n}}\left(\mathrm{D}\, P_{\mathrm{T}_X \mathbb{S}_r^{n\times n}}[\eta_X](AXM + MXA - C)\right)\end{aligned}$$

- Preconditioner for the first term in the Riemannian Hessian: for any $\xi_X \in \mathrm{T}_X \mathbb{S}_r^{n\times n}$, find η_X such that

$$P_{\mathrm{T}_X \mathbb{S}_r^{n\times n}}(A\eta_X M + M\eta_X A) = \xi_X \tag{2}$$

The action of the Riemannian Hessian is

$$\begin{aligned}\mathrm{Hess}\, f(X)[\eta_X] = & P_{\mathrm{T}_X \mathbb{S}_r^{n\times n}}(A\eta_X M + M\eta_X A) \\ & + P_{\mathrm{T}_X \mathbb{S}_r^{n\times n}} \left(\mathrm{D}\, P_{\mathrm{T}_X \mathbb{S}_r^{n\times n}}[\eta_X](AXM + MXA - C)\right)\end{aligned}$$

- Preconditioner for the first term in the Riemannian Hessian: for any $\xi_X \in \mathrm{T}_X \mathbb{S}_r^{n\times n}$, find η_X such that

$$P_{\mathrm{T}_X \mathbb{S}_r^{n\times n}}(A\eta_X M + M\eta_X A) = \xi_X \tag{2}$$

- Is equation (2) solvable? Yes, it can be written as

$$P_{\mathrm{T}_X \mathbb{S}_r^{n\times n}}(A \otimes M + M \otimes A) P_{\mathrm{T}_X \mathbb{S}_r^{n\times n}} \mathrm{vec}(\eta_X) = \mathrm{vec}(\xi_X),$$

Preconditioner:

$$P_{\mathrm{T}_X \mathbb{S}_r^{n\times n}}(A \otimes M + M \otimes A) P_{\mathrm{T}_X \mathbb{S}_r^{n\times n}} \mathrm{vec}(\eta_X) = \mathrm{vec}(\xi_X)$$

Existing Preconditioner in [VV10]

- The preconditioner need be solved in $O(nr^c)$ with a reasonable constant c;

Preconditioner:

$$P_{\mathrm{T}_X \mathbb{S}_r^{n\times n}}(A \otimes M + M \otimes A) P_{\mathrm{T}_X \mathbb{S}_r^{n\times n}} \mathrm{vec}(\eta_X) = \mathrm{vec}(\xi_X)$$

Existing Preconditioner in [VV10]

- The preconditioner need be solved in $O(nr^c)$ with a reasonable constant c;
- The existing one
  - Assumption: solve $(A + \lambda I)x = b$ in $O(n)$
  - Only for $M = I$;

- Optimization formulation on quotient manifold;
- A Riemannian Newton-tCG method based on line search;
- New preconditioners considering $M \neq I$;
- Increasing rank method;

$$\min_{X \in \mathbb{S}_r^{n \times n}} f(X) = \mathrm{trace}(XAXM) - \mathrm{trace}(XC) \tag{3}$$

- Any $X \in \mathbb{S}_r^{n \times n}$, there exists $Y \in \mathbb{R}_*^{n \times r}$ such that $X = YY^T$;
- For any $O \in \mathcal{O}_r$, $\tilde{Y} = YO$ also satisfies $X = \tilde{Y}\tilde{Y}^T$;
- Define equivalence class: $[Y] = \{YO \mid O \in \mathcal{O}_r\}$;
- Quotient manifold $\mathbb{R}_*^{n \times r}/\mathcal{O}_r = \{[Y] \mid Y \in \mathbb{R}_*^{n \times r}\}$;
- Map $\beta : \mathbb{R}_*^{n \times r}/\mathcal{O}_r \to \mathbb{S}_r^{n \times n} : [Y] \mapsto YY^*$ is a diffeomorphism;

$$\min_{X \in \mathbb{S}_r^{n\times n}} f(X) = \mathrm{trace}(XAXM) - \mathrm{trace}(XC) \tag{3}$$

- Any $X \in \mathbb{S}_r^{n\times n}$, there exists $Y \in \mathbb{R}_*^{n\times r}$ such that $X = YY^T$;
- For any $O \in \mathcal{O}_r$, $\tilde{Y} = YO$ also satisfies $X = \tilde{Y}\tilde{Y}^T$;
- Define equivalence class: $[Y] = \{YO \mid O \in \mathcal{O}_r\}$;
- Quotient manifold $\mathbb{R}_*^{n\times r}/\mathcal{O}_r = \{[Y] \mid Y \in \mathbb{R}_*^{n\times r}\}$;
- Map $\beta : \mathbb{R}_*^{n\times r}/\mathcal{O}_r \to \mathbb{S}_r^{n\times n} : [Y] \mapsto YY^*$ is a diffeomorphism;

Optimization on quotient manifold:

$$\Longrightarrow \min_{[Y] \in \mathbb{R}_*^{n\times r}/\mathcal{O}_r} \tilde{f}([Y]) = \mathrm{trace}(Y^*AYY^*MY) - \mathrm{trace}(Y^*CY) \tag{4}$$

$$\min_{X \in \mathbb{S}_r^{n\times n}} f(X) = \mathrm{trace}(XAXM) - \mathrm{trace}(XC) \qquad (3)$$

- Any $X \in \mathbb{S}_r^{n\times n}$, there exists $Y \in \mathbb{R}_*^{n\times r}$ such that $X = YY^T$;
- For any $O \in \mathcal{O}_r$, $\tilde{Y} = YO$ also satisfies $X = \tilde{Y}\tilde{Y}^T$;
- Define equivalence class: $[Y] = \{YO \mid O \in \mathcal{O}_r\}$;
- Quotient manifold $\mathbb{R}_*^{n\times r}/\mathcal{O}_r = \{[Y] \mid Y \in \mathbb{R}_*^{n\times r}\}$;
- Map $\beta : \mathbb{R}_*^{n\times r}/\mathcal{O}_r \to \mathbb{S}_r^{n\times n} : [Y] \mapsto YY^*$ is a diffeomorphism;

Optimization on quotient manifold:

$$\Longrightarrow \min_{[Y] \in \mathbb{R}_*^{n\times r}/\mathcal{O}_r} \tilde{f}([Y]) = \mathrm{trace}(Y^*AYY^*MY) - \mathrm{trace}(Y^*CY) \qquad (4)$$

Problem (3) and Problem (4) are equivalent.

Three metrics on the total space $\mathbb{R}_*^{n \times r}$ [ZHVZ23]:

$$\begin{cases} g_Y^1(\eta_Y, \xi_Y) = 2\mathrm{trace}(Y^T \eta_Y Y^T \xi_Y + Y^T Y \eta_Y^T \xi_Y) + \mathrm{trace}\left(Y^T Y (\eta_Y^{\mathrm{V}})^T (\xi_Y^{\mathrm{V}})\right), \\ g_Y^2(\eta_Y, \xi_Y) = \mathrm{trace}(Y^T Y \eta_Y^T \xi_Y), \\ g_Y^3(\eta_Y, \xi_Y) = \mathrm{trace}(\eta_Y^T \xi_Y), \end{cases}$$

where $\eta_Y^{\mathrm{V}} = Y\left((Y^T Y)^{-1} Y^T \eta_Y - \eta_Y^T Y (Y^T Y)^{-1}\right)/2$ and $\xi_Y^{\mathrm{V}} = Y((Y^T Y)^{-1} Y^T \xi_Y - \xi_Y^T Y (Y^T Y)^{-1})/2$.

- Metrics above yield three Riemannian metrics on $\mathbb{R}_*^{n \times r} / \mathcal{O}_r$;
- g_Y^1 is equivalent to the Euclidean metric on $\mathbb{S}_r^{n \times n}$;
- g_Y^3 is the Euclidean metric on the total space;

Riemannian gradient and Riemannian Hessian depend on Riemannian metric

Choose g_Y^1 for example:

- Riemannian gradient:

$$(\mathrm{grad} \tilde{f}([Y]))_{\uparrow_Y} = (I - \frac{1}{2} Y(Y^T Y)^{-1} Y^T) \nabla h(YY^T) Y (Y^T Y)^{-1}$$

  where $\nabla h(X) = AXM + MXA - C$;

- The action of the Riemannian Hessian:

$$\begin{aligned}(\mathrm{Hess} \tilde{f}([Y])[\eta_{[Y]}])_{\uparrow_Y} =& (1 - \frac{1}{2} P_Y) \nabla^2 h(YY^T)[Y \eta_{\uparrow_Y}^T + \eta_{\uparrow_Y} Y^T] Y (Y^T Y)^{-1} \\ &+ (I - P_Y) \nabla h(YY^T)(I - P_Y) \eta_{\uparrow_Y} (Y^T Y)^{-1}\end{aligned}$$

  where $\nabla^2 h(X)[V] = AVM + MVA$ and $P_Y = Y(Y^T Y)^{-1} Y^T$.

- Riemannian gradient and Hessian can be derived for g_Y^1 and g_Y^2;

$$\min_{X \in \mathbb{S}_r^{n \times n}} f(X) = \mathrm{trace}(XAXM) - \mathrm{trace}(XC)$$

$$\min_{[Y] \in \mathbb{R}_*^{n \times r}/\mathcal{O}_r} \tilde{f}([Y]) = \mathrm{trace}(Y^* AYY^* MY) - \mathrm{trace}(Y^* CY)$$

Preferences to quotient manifold:

- More Riemannian metric;
- Lower complexity retraction: $R_{[Y]}(\eta_{[Y]}) = [Y + \eta_{\uparrow_Y}]$;
- Easier derivation for new preconditioners;

Consider the Riemannian optimization problem in the form of

$$\min_{x} f(x) \text{ s.t. } x \in \mathcal{M}$$

where $\mathcal{M}$ is a finite dimension Riemannian manifold, and $f : \mathcal{M} \to \mathbb{R}$ is a real-valued function.

Riemannian Newton-tCG

- Approximate $\mathrm{Hess} f(x_k)[\eta_k] = -\mathrm{grad} f(x_k)$ for η_k.
- Find a step-size α_k such that

$$h_k(\alpha_k) - h_k(0) \leq -\chi_1 \frac{h_k'(0)^2}{\|\eta_k\|^2}, \text{ or}$$
$$h_k(\alpha_k) - h_k(0) \leq \chi_2 h_k'(0),$$

where $h_k(t) = f(R_k(t\eta_k))$.

Euclidean Newton-tCG

- Approximate $\nabla^2 f(x_k)[p_k] = -\nabla f(x_k)$ for p_k.
- Find a step-size α_k such that

$$h_k(\alpha_k) \leq h_k(0) + c_1 \alpha_k h_k'(0), \text{ and}$$
$$h_k'(\alpha_k) \geq c_2 h_k'(0),$$

where $h_k(t) = f(x_k + tp_k)$.

Riemannian Newton-tCG

- Approximate $\mathrm{Hess} f(x_k)[\eta_k] = -\mathrm{grad} f(x_k)$ for η_k.
- Find a step-size α_k such that

$$h_k(\alpha_k) - h_k(0) \leq -\chi_1 \frac{h_k'(0)^2}{\|\eta_k\|^2}, \text{ or}$$
$$h_k(\alpha_k) - h_k(0) \leq \chi_2 h_k'(0),$$

where $h_k(t) = f(R_k(t\eta_k))$.

Euclidean Newton-tCG

- Approximate $\nabla^2 f(x_k)[p_k] = -\nabla f(x_k)$ for p_k.
- Find a step-size α_k such that

$$h_k(\alpha_k) \leq h_k(0) + c_1 \alpha_k h_k'(0), \text{ and}$$
$$h_k'(\alpha_k) \geq c_2 h_k'(0),$$

where $h_k(t) = f(x_k + tp_k)$.

- Truncated conjugate gradient for Approximating Newton equation in both methods.

Riemannian Newton-tCG

- Approximate $\mathrm{Hess} f(x_k)[\eta_k] = -\mathrm{grad} f(x_k)$ for η_k.
- Find a step-size α_k such that

$$h_k(\alpha_k) - h_k(0) \leq -\chi_1 \frac{h_k'(0)^2}{\|\eta_k\|^2}, \text{ or}$$

$$h_k(\alpha_k) - h_k(0) \leq \chi_2 h_k'(0),$$

where $h_k(t) = f(R_k(t\eta_k))$.

Euclidean Newton-tCG

- Approximate $\nabla^2 f(x_k)[p_k] = -\nabla f(x_k)$ for p_k.
- Find a step-size α_k such that

$$h_k(\alpha_k) \leq h_k(0) + c_1 \alpha_k h_k'(0), \text{ and}$$

$$h_k'(\alpha_k) \geq c_2 h_k'(0),$$

where $h_k(t) = f(x_k + t p_k)$.

- Truncated conjugate gradient for Approximating Newton equation in both methods.
- BN conditions [BN89] for RNewton-tCG and the Wolfe conditions for Newton-tCG.

Riemannian Newton-tCG

- Approximate $\mathrm{Hess} f(x_k)[\eta_k] = -\mathrm{grad} f(x_k)$ for η_k.
- Find a step-size α_k such that

$$h_k(\alpha_k) - h_k(0) \leq -\chi_1 \frac{h_k'(0)^2}{\|\eta_k\|^2}, \text{ or}$$
$$h_k(\alpha_k) - h_k(0) \leq \chi_2 h_k'(0),$$

where $h_k(t) = f(R_k(t\eta_k))$.

Euclidean Newton-tCG

- Approximate $\nabla^2 f(x_k)[p_k] = -\nabla f(x_k)$ for p_k.
- Find a step-size α_k such that

$$h_k(\alpha_k) \leq h_k(0) + c_1 \alpha_k h_k'(0), \text{ and}$$
$$h_k'(\alpha_k) \geq c_2 h_k'(0),$$

where $h_k(t) = f(x_k + tp_k)$.

- Truncated conjugate gradient for Approximating Newton equation in both methods.
- BN conditions [BN89] for RNewton-tCG and the Wolfe conditions for Newton-tCG.
- If f is radially L-C^1, then the Wolfe conditions, Armijo-Goldstein conditions imply the BN conditions [HAG18].

Assumption 1. f is twice continuously differentiable.
Assumption 2. For all starting $x_0 \in \mathcal{M}$, the level set $L(x_0) := \{x \in \mathcal{M} : f(x) \leq f(x_0)\}$ is bounded.

Theorem

Let $\{x_n\}$ denote the sequence generated by Riemannian Newton-tCG method. Then it holds that

$$\lim_{n \to \infty} \|\mathrm{grad} f(x_n)\| = 0.$$

If x^* is an accumulation point of the sequence $\{x_k\}$ and $\mathrm{Hess} f(x^*)$ is positive definite, then $x_k \to x^*$.

Assumption 1. f is twice continuously differentiable.
Assumption 2. For all starting $x_0 \in \mathcal{M}$ the level set $L(x_0) := \{x \in \mathcal{M} : f(x) \leq f(x_0)\}$ is bounded.
Assumption 3. f is radially L-Lipschitz continuous.

Theorem

Let $\{x_k\}$ be the sequence generated by Riemannian Newton-tCG method. Suppose that $\{x_k\}$ converges to x^* at which $\mathrm{Hess} f(x^*)$ is positive definite and $\mathrm{Hess} f(x)$ is continuous in a neighborhood of x^*. Then

1. *the stepsize $\alpha_k = 1$ is acceptable for sufficiently large k; and*
2. *the convergence rate is superlinear.*

Moreover, suppose that $\mathrm{Hess}\hat{f}$ satisfies that $\|\mathrm{Hess} f(x_k) - \mathrm{Hess}\hat{f}_{x_k}(0_{x_k})\| \leq \beta_1 \|\mathrm{grad} f(x_k)\|$, $\hat{f} = f \circ R : \mathrm{T}\mathcal{M} \to \mathbb{R}$, with a positive constant β_1. and that there exist $\beta_2 > 0$, $\mu_1 > 0$ and $\mu_2 > 0$ such that for all $x \in B_{\mu_1}(x^*)$ and all $\eta_x \in B_{\mu_2}(0_x)$, it holds that $\|\mathrm{Hess}\hat{f}_x(\eta_x) - \mathrm{Hess}\hat{f}_x(0_x)\| \leq \beta_2 \|\eta_x\|$. Then,

3. *the convergence rate is $1 + \min(1, t)$.*

Consider g_Y^1 as an example

Newton equation:

$$\mathrm{Hess}\, \tilde{f}([Y])[\eta_{[Y]}] = \xi_{[Y]} \qquad \Longrightarrow$$

$$(1 - \frac{1}{2} P_Y) \nabla^2 h(YY^T)[Y\eta_{\uparrow_Y}^T + \eta_{\uparrow_Y} Y^T] Y (Y^T Y)^{-1} + (1 - P_Y)\nabla h(YY^T)(I - P_Y)\eta_{\uparrow_Y}(Y^T Y)^{-1} = \xi_{\uparrow_Y}.$$

where $\eta_{\uparrow_Y}, \xi_{\uparrow_Y}$ are in the horizontal space at Y, $\mathcal{H}_Y$.

- Preconditioner: solve for $\eta_{\uparrow_Y} \in \mathcal{H}_Y$ in

$$(I - \frac{1}{2} Y(Y^T Y)^{-1} Y^T) \nabla^2 h(YY^T)[Y\eta_{\uparrow_Y}^T + \eta_{\uparrow_Y} Y^T] Y (Y^T Y)^{-1} = \xi_{\uparrow_Y}$$

Preconditioner:

$$(I - \frac{1}{2} Y(Y^T Y)^{-1} Y^T) \nabla^2 h(YY^T)[Y \eta_{\uparrow_Y}^T + \eta_{\uparrow_Y} Y^T] Y (Y^T Y)^{-1} = \xi_{\uparrow_Y} \quad (5)$$

- Key idea: for any $\eta_Y \in \mathrm{T}_Y \mathbb{R}_*^{n \times r}$, η_Y can be decomposed into

$$\eta_Y = YS + Y_{\perp_M} K$$

  where $S \in \mathcal{S}_r^{\mathrm{sym}}$ and $K \in \mathbb{R}^{(n-r) \times r}$, $Y^T M Y_{\perp_M} = 0$ and $Y_{\perp_M}^T Y_{\perp_M} = I_{n-r}$.
- Assumption: solve $(A + \lambda M)x = b$ in $O(n)$.
- Using such decomposition for η_Y, one can solve for $\eta_{\uparrow_Y}$ in $O(nr^c)$ with a constant c.

Solving Generalized Lyapunov Equations

Our Work: Increasing Rank Method

- Use Riemannian Newton-tCG method, if the rank is known;
- Use increasing rank technique if rank is unknown;
- Used in [VV10];

Algorithm 3 An Increasing Rank Riemannian Method for Lyapunov Equations (IRRLyap)

Input: minimum rank $p_{\min}$; maximum rank $p_{\max}$; rank increment p_{inc}; initial iterate $Y_{p_{\min}}^{\text{initial}} \in \mathbb{R}_*^{n \times p_{\min}}$; tolerance sequence of inner iteration $\{\tau_p : p \in \{p_{\min}, p_{\min} + p_{\text{inc}}, p_{\min} + 2p_{\text{inc}}, \dots, p_{\max}\}\}$; residual tolerance τ;

Output: low-rank approximation $\widetilde{Y}$;

```
for p = p_min, p_min + p_inc, p_min + 2p_inc, ..., p_max do
    Invoke an optimization algorithm, such as Algorithm 1, to approximately solve Problem
    (3.2) with the initial iterate π(Y_p^initial) until the last iterate π(Y_p) satisfies ‖grad f(π(Y_p))‖ ≤
    τ_p‖grad f(π(Y_p^initial))‖;
    Compute relative residual of Y_p: r_p ← ‖A Y_p Y_p^T M + M Y_p Y_p^T A − C‖_F / ‖C‖_F;
    if r_p ≤ τ then
        Return Ỹ ← Y_p;
    else
        Calculate the next initial iterate Y_{p+p_inc}^initial by performing one step of steepest descent on
        [Y_p  0_{n×p_inc}];
    end if
end for
Return Ỹ ← Y_{p_max};
```

- Influence of Riemannian metrics
- Riemannian Newton-tCG versus Riemannian trust region Newton-tCG
- Comparisons with existing methods

Random data: Stopping criterion $\|\mathrm{grad} f(x_k)\|/\|\mathrm{grad} f(x_0)\| \leq 10^{-8}$

RNewton	$n = 500, p = 2$						$n = 1000, p = 2$					
	non-preconditioner			preconditioner			non-preconditioner			preconditioner		
	metric 1	metric 2	metric 3	metric 1	metric 2	metric 3	metric 1	metric 2	metric 3	metric 1	metric 2	metric 3
success	20	20	20	20	20	20	20	20	20	20	20	20
iter	43	71	45	21	26	21	43	65	40	18	29	19
nf	53	84	54	24	29	25	52	79	48	21	34	22
ng	44	72	46	22	27	22	44	66	41	19	30	20
nH	2361	2140	3576	57	339	206	2611	2307	3515	46	413	219
time	3.56	3.28	5.27	1.21	7.15	4.37	1.35_1	1.20_1	1.79_1	5.09	4.16_1	2.18_1
gfgf0	3.00_{-9}	3.97_{-9}	4.12_{-9}	2.11_{-9}	4.50_{-9}	3.39_{-9}	3.40_{-9}	2.48_{-9}	4.39_{-9}	1.41_{-9}	4.64_{-9}	4.05_{-9}

- iter: number of iterations
- nf: number of evaluations of cost function
- ng: number of evaluations of norm of gradient
- nH: number of evaluations of action of Hessian
- time: running time
- gfgf0: $\|\mathrm{grad} f(x_k)\|/\|\mathrm{grad} f(x_0)\|$.

The performance under the first metric is the best among three metrics.

The finite difference discretized $2D$ poisson problem on the square

RNewton	$n = 4000, p = 3$						$n = 40000, p = 3$					
	non-preconditioner			preconditioner			non-preconditioner			preconditioner		
	metric 1	metric 2	metric 3	metric 1	metric 2	metric 3	metric 1	metric 2	metric 3	metric 1	metric 2	metric 3
success	20	20	20	20	20	20	20	20	18	20	20	20
iter	13	41	53	2	8	6	10	35	63	1	7	5
nf	15	49	67	3	9	7	12	43	80	2	8	6
ng	14	42	54	3	9	7	11	36	64	2	8	6
nH	723	602	287	2	22	9	550	536	314	1	16	6
time	1.07	9.50_{-1}	4.24_{-1}	1.75_{-2}	1.70_{-1}	6.43_{-2}	7.96	7.67	3.94	6.98_{-2}	9.89_{-1}	3.33_{-1}
gfgf0	8.03_{-9}	8.00_{-9}	7.78_{-9}	7.04_{-9}	6.97_{-9}	4.43_{-9}	6.30_{-9}	7.32_{-9}	4.53_{-9}	1.02_{-10}	3.65_{-9}	2.56_{-9}

- iter: number of iterations
- nf: number of evaluations of cost function
- ng: number of evaluations of norm of gradient
- nH: number of evaluations of action of Hessian
- time: running time
- gfgf0: $\|\mathrm{grad} f(x_k)\| / \|\mathrm{grad} f(x_0)\|$.

The performance under the first metric is the best among three metrics.

Riemannian metric g_Y^1

- $n = 50^2$; $r = 10$; Stop if $\|\operatorname{grad} f(x_i)\| / \|\operatorname{grad} f(x_0)\| < 10^{-10}$;
- A: the negative stiffness matrix of PDE $\nabla u(x, y) = f$ on unit square Ω and $u = 0$ on $\partial\Omega$ (Lyapack [Pen00a]);
- M: diagonal matrix;
- C: rank one matrix bb^T with entries of b from standard normal distribution;

Table: $M = I$

		No precon.	precon. [VV10]	New precon.
RTRNewton	iter	89	48	47
	nH	439	57	54
RNewton	iter	21	14	14
	nH	328	22	25

Riemannian metric g_Y^1

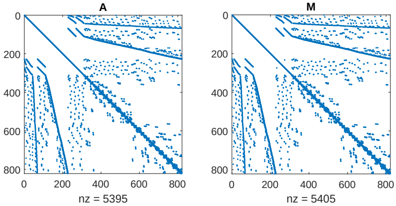
- $n = 50^2$; $r = 10$; Stop if $\|\operatorname{grad} f(x_i)\| / \|\operatorname{grad} f(x_0)\| < 10^{-10}$;
- A: the negative stiffness matrix of PDE $\nabla u(x, y) = f$ on unit square Ω and $u = 0$ on $\partial\Omega$ (Lyapack [Pen00a]);
- M: diagonal matrix;
- C: rank one matrix bb^T with entries of b from standard normal distribution;

Table: $M = \mathrm{diag}([\mathrm{rand}(n-1, 1); 0] + 0.1)$

		No precon.	precon. [VV10]	New precon.
RTRNewton	iter	48	57	49
	nH	398	114	84
RNewton	iter	23	33	19
	nH	324	95	46

Riemannian metric g_Y^1

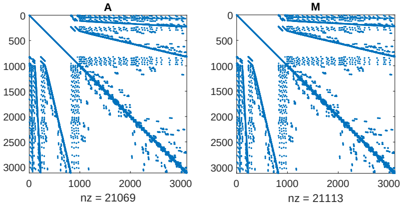
- A, M and C; from semidiscretization of a steel rail cooling problem [Pen06];
- Coarse discretization: $n = 821$; $r = 20$; Stop if $\|\operatorname{grad} f(x_i)\| / \|\operatorname{grad} f(x_0)\| < 10^{-10}$;



		No precon.	precon. [VV10]	New precon.
RTRNewton	iter	1476	68	83
	nH	3838	155	114
RNewton	iter	260	47	21
	nH	1160	129	51

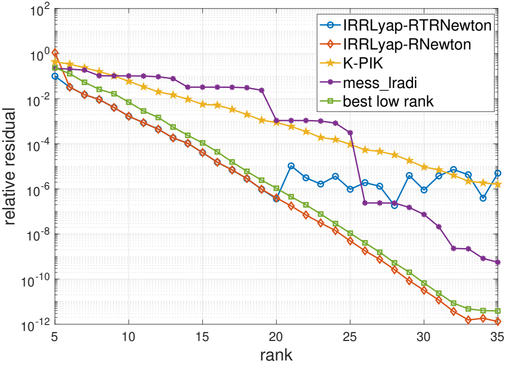
Riemannian metric g_Y^1

- A, M and C; from semidiscretization of a steel rail cooling problem [Pen06];
- Dense discretization: $n = 3113$; $r = 20$; Stop if $\|\operatorname{grad} f(x_i)\| / \|\operatorname{grad} f(x_0)\| < 10^{-10}$;



		No precon.	precon. [VV10]	New precon.
RTRNewton	iter	2000	79	79
	nH	5942	195	127
RNewton	iter	320	60	30
	nH	2015	267	91

A, M and C from semidiscretization of a steel rail cooling problem [BS05, SB04] with $n = 1357$.



- K-PIK is based on Krylov subspace technique.
- mess_lradi is based on ADI.
- best low rank is given by the truncation of the first p singular values for the exact solution.

A, M and C from semidiscretization of a steel rail cooling problem [BS05, SB04] with $n = 5177, 20209, 79841$.

Table 3: Comparison for the simplified RAIL benchmark with existing methods. "rank", "time", "rel_res" and "numSys" denote the rank of the approximation, running time, the relative residual of the approximation and the number of solving shift systems $(A + \lambda M)X = B$ for X with given A, λ, M and B. The subscript $-k$ indicates a scale of 10^{-k}.

	rank	times(s.)	rel_res	numSys	rank	times(s.)	rel_res	numSys	rank	times(s.)	rel_res	numSys
	$n = 5177$				$n = 20209$				$n = 79841$			
K-PIK	63	3.41	1.46_{-6}	64	91	4.44_{1}	2.65_{-6}	92	122	5.06_{2}	4.39_{-6}	123
mess_lradi	32	1.57_{-1}	1.47_{-7}	64	37	8.65_{-1}	5.90_{-7}	74	38	3.85	6.12_{-8}	76
RLyap	22	1.42_{2}	4.92_{-7}	15784	27	1.06_{3}	2.25_{-7}	23060	27	6.09_{3}	8.58_{-7}	29481
IRRLyap(RNewton)	22	5.70	6.94_{-7}	588	27	4.29_{1}	3.38_{-7}	841	27	2.56_{2}	5.10_{-7}	1100

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;
- Solving generalized Lyapunov equations;
  - Motivation;
  - Riemannian inexact Newton;
  - A Rank increasing algorithm;
  - Numerical experiments;
- Summary;

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.

P. Benner.
Control Theory.
Handbook of Linear Algebra, Chapman and Hall/CRC, 2006.

R. H. Byrd and J. Nocedal.
A tool for the analysis of quasi-Newton methods with application to unconstrained minimization.
SIAM Journal on Numerical Analysis, 26(3):727–739, 1989.

Peter Benner and Jens Saak.
A semi-discretized heat transfer model for optimal cooling of steel profiles.
In *Dimension Reduction of Large-Scale Systems*, pages 353–356. Springer Berlin Heidelberg, 2005.

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.

Wen Huang, P.-A. Absil, and K. A. Gallivan.
A Riemannian BFGS method without differentiated retraction for nonconvex optimization problems.
SIAM Journal on Optimization, 28(1):470–495, 2018.

References II

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.

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. Kurtek, 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.

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.

B. Moore.
Principal component analysis in linear systems: Controllability, observability, and model reduction.
IEEE Transactions on Automatic Control, 26(1):17–32, 2003.

Thilo Penzl.
Numerical solution of generalized lyapunov equations.
Advances in Computational Mathematics, 8(1-2):33–48, 1998.

Thilo Penzl.
A cyclic low-rank smith method for large sparse lyapunov equations.
Siam Journal on Scientific Computing, 21(4):1401–1418, 2000.

References III

Thilo Penzl.
Eigenvalue decay bounds for solutions of lyapunov equations: the symmetric case.
Systems and Control Letters, 40(2):139–144, 2000.

Thilo Penzl.
Algorithms for model reduction of large dynamical systems.
Linear Algebra and Its Applications, 415(2):322–343, 2006.

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.

Jens Saak and Peter Benner.
Efficient numerical solution of the LQR-problem for the heat equation.
Proceedings in Applied Mathematics and Mechanics, 4(1), 2004.

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.

Bart Vandereycken and Stefan Vandewalle.
A riemannian optimization approach for computing low-rank solutions of lyapunov equations.
Siam Journal on Matrix Analysis and Applications, 31(5):2553–2579, 2010.

Shixin Zheng, Wen Huang, Bart Vandereycken, and Xiangxiong Zhang.
Riemannian optimization using three different metrics for hermitian psd fixed-rank constraints: an extended version, 2023.

Thank you

Thank you!