An Increasing Rank Riemannian Method for Generalized Lyapunov Equations

Wen Huang

Xiamen University

November 25, 2023

This is joint work with Zhenwei Huang at Xiamen University.

CORO2023 Shenzhen University

Problem Statement

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 Statement

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;

Problem Statement

$A \succ 0, M \succ 0$ 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?

Problem Statement

$A \succ 0, M \succ 0$ 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

Problem Statement

$A \succ 0, M \succ 0$ 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]

Existing Methods

$A \succ 0, M \succ 0$ 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] $\Longrightarrow X = YY^T$

Existing Methods

$A \succ 0, M \succ 0$ 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;

Existing Methods

$A \succ 0, M \succ 0$ 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 $X = YY^T$.

Existing Methods

$A \succ 0, M \succ 0$ 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;

Problem Reformulation [VV10]

- 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.

Problem Reformulation [VV10]

- 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.

Existing Riemannian Optimization technique

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

Ingredients for Riemannian optimization

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

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

Ingredients for Riemannian optimization

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

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

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

Ingredients for Riemannian optimization

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

Ingredients for Riemannian optimization

- 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:

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

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 $P_Y = Y(Y^T Y)^{-1} Y^T$;

Ingredients for Riemannian optimization

- 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:

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

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;

Riemannian Trust-region Newton method

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;

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

Preconditioner

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

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

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

Preconditioner

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

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

- 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

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

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 $M = I$;

Our Work

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

Our Work: Optimization on Quotient Manifold

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

- 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;

Our Work: Optimization on Quotient Manifold

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

- 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:

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

Our Work: Optimization on Quotient Manifold

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

- 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:

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

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

Our Work: Optimization on Quotient Manifold

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

- 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;

Our Work: Optimization on Quotient Manifold

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;

Our Work: Optimization on Quotient Manifold

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

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

Preferences to quotient manifold:

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

Our Work: A Riemannian Newton-tCG method

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.

Our Work: A Riemannian Newton-tCG method

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

Our Work: A Riemannian Newton-tCG method

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.

Our Work: A Riemannian Newton-tCG method

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.

Our Work: A Riemannian Newton-tCG method

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

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

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

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

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].

Our Work: A Riemannian Newton-tCG method

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 $x_k \to x^*$.

Our Work: A Riemannian Newton-tCG method

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 $1 + \min(1, t)$.*

Our Work: New preconditioners

Consider g_Y^1 as an example

Newton equation:

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

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

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

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

Our Work: New preconditioners

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.

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 $\{\tau_p : p \in \{p_{\min}, p_{\min}+p_{\text{inc}}, p_{\min}+2p_{\text{inc}}, \ldots, p_{\max}\}\}$; residual tolerance τ;

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

1: **for** $p = p_{\min}, p_{\min} + p_{\text{inc}}, p_{\min} + 2p_{\text{inc}}, \ldots, p_{\max}$ **do**

2: Invoke an optimization algorithm, such as Algorithm 1, to approximately solve Problem (3.2) with the initial iterate $\pi(Y_p^{\text{initial}})$ until the last iterate $\pi(Y_p)$ satisfies $\|\operatorname{grad} f(\pi(Y_p))\| \leq \tau_p \|\operatorname{grad} f(\pi(Y_p^{\text{initial}}))\|$;

3: Compute relative residual of Y_p: $r_p \leftarrow \|AY_pY_p^TM + MY_pY_p^TA - C\|_F / \|C\|_F$;

4: **if** $r_p \leq \tau$ **then**

5: Return $\widetilde{Y} \leftarrow Y_p$;

6: **else**

7: Calculate the next initial iterate $Y_{p+p_{\text{inc}}}^{\text{initial}}$ by performing one step of steepest descent on $\begin{bmatrix} Y_p & \mathbf{0}_{n \times p_{\text{inc}}} \end{bmatrix}$;

8: **end if**

9: **end for**

10: Return $\widetilde{Y} \leftarrow Y_{p_{\max}}$;

Numerical Experiments

- 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

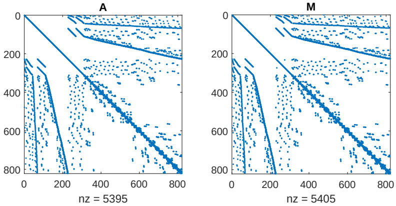
- $n = 50^2$; $r = 10$; Stop if $\|\operatorname{grad} f(x_i)\| / \|\operatorname{grad} f(x_0)\| < 10^{-10}$;
- A: the negative stiffness matrix of PDE $\nabla u(x, y) = f$ on unit square Ω and $u = 0$ on $\partial\Omega$ (Lyapack [Pen00a]);
- M: diagonal matrix;
- C: rank one matrix bb^T with entries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

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

		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

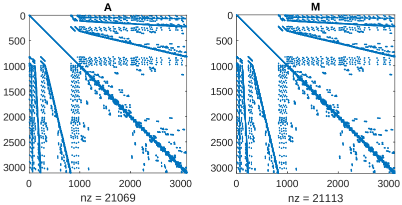
- A, M and C; from semidiscretization of a steel rail cooling problem [Pen06];
- Coarse discretization: $n = 821$; $r = 20$; Stop if $\|\operatorname{grad} f(x_i)\| / \|\operatorname{grad} f(x_0)\| < 10^{-10}$;



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

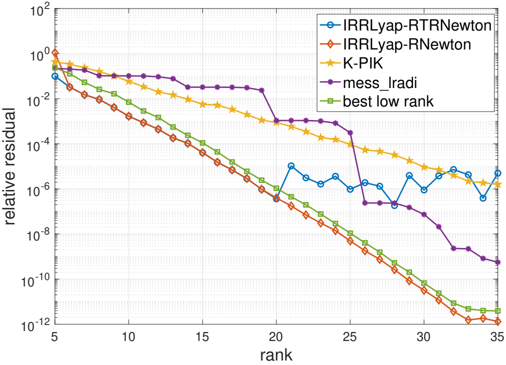
Riemannian metric g_Y^1

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



		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

- Briefly introduced the generalized Lyapunov equation;
- Review the existing Riemannian method;
- Our approach
  - Optimization over a quotient manifold with three metrics
  - A Riemannian Newton-tCG method with convergence analysis
  - New preconditioner with $M \neq I$;
  - Increasing rank method;

For more details, see
Zhenwei Huang, Wen Huang. An increasing rank Riemannian method for generalized Lyapunov equations. arXiv.2308.00213, 2023.

References I

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.

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.

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.

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

References II

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

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.

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.