Nearest singular pencil via Riemannian optimization

Froilán M. Dopico

with **Vanni Noferini and Lauri Nyman** (Aalto University, Finland)

Depto de Matemáticas, Universidad Carlos III de Madrid, Spain

Part of "Proyecto de I+D+i PID2019-106362GB-I00 financiado por MCIN/AEI/10.13039/501100011033"

Computational and Applied Mathematics 2023.
A workshop in honor of **Marc Van Barel's retirement.**
Selva di Fasano (Br, Italy). August 29, 2023

GOBIERNO DE ESPAÑA MINISTERIO DE CIENCIA E INNOVACIÓN

AGENCIA ESTATAL DE INVESTIGACIÓN

uc3m Universidad Carlos III de Madrid

Outline

Outline

Matrix Pencils

- This talk deals with square complex matrix pencils $A + \lambda B \in \mathbb{C}[\lambda]_1^{n\times n}$ or polynomial matrices of degree 1, where $A, B \in \mathbb{C}^{n\times n}$.
- Matrix pencils arise naturally in differential-algebraic equations and in linear time invariant control systems

$$-B\dot{x} = Ax + Fu, \qquad y = Cx \tag{1}$$

  by taking Laplace transforms.
- The pencil $A + \lambda B$ is regular if its characteristic polynomial $p(\lambda) = \det(A + \lambda B)$ is NOT identically zero. Otherwise, the pencil is singular, i.e., if $p(\lambda) = \det(A + \lambda B) \equiv 0$.
- The regularity of $A + \lambda B$ implies that a solution of (1) exists for all smooth enough controls and for consistent initial conditions.
- This existence is no longer guaranteed if the pencil $A + \lambda B$ is singular. Therefore, the distance of a regular pencil $A + \lambda B$ to a nearest singular pencil is a measure of the robustness of the problem (1).

- This talk deals with square complex matrix pencils $A + \lambda B \in \mathbb{C}[\lambda]_1^{n\times n}$ or polynomial matrices of degree 1, where $A, B \in \mathbb{C}^{n\times n}$.
- Matrix pencils arise naturally in differential-algebraic equations and in linear time invariant control systems

$$-B\dot{x} = Ax + Fu, \qquad y = Cx \tag{1}$$

by taking Laplace transforms.

- The pencil $A + \lambda B$ is regular if its characteristic polynomial $p(\lambda) = \det(A + \lambda B)$ is NOT identically zero. Otherwise, the pencil is singular, i.e., if $p(\lambda) = \det(A + \lambda B) \equiv 0$.
- The regularity of $A + \lambda B$ implies that a solution of (1) exists for all smooth enough controls and for consistent initial conditions.
- This existence is no longer guaranteed if the pencil $A + \lambda B$ is singular. Therefore, the distance of a regular pencil $A + \lambda B$ to a nearest singular pencil is a measure of the robustness of the problem (1).

Matrix Pencils

- This talk deals with square complex matrix pencils $A + \lambda B \in \mathbb{C}[\lambda]_1^{n\times n}$ or polynomial matrices of degree 1, where $A, B \in \mathbb{C}^{n\times n}$.
- Matrix pencils arise naturally in differential-algebraic equations and in linear time invariant control systems

$$-B\dot{x} = Ax + Fu, \qquad y = Cx \tag{1}$$

  by taking Laplace transforms.
- The pencil $A + \lambda B$ is regular if its characteristic polynomial $p(\lambda) = \det(A + \lambda B)$ is NOT identically zero. Otherwise, the pencil is singular, i.e., if $p(\lambda) = \det(A + \lambda B) \equiv 0$.
- The regularity of $A + \lambda B$ implies that a solution of (1) exists for all smooth enough controls and for consistent initial conditions.
- This existence is no longer guaranteed if the pencil $A + \lambda B$ is singular. Therefore, the distance of a regular pencil $A + \lambda B$ to a nearest singular pencil is a measure of the robustness of the problem (1).

- This talk deals with square complex matrix pencils $A + \lambda B \in \mathbb{C}[\lambda]_1^{n\times n}$ or polynomial matrices of degree 1, where $A, B \in \mathbb{C}^{n\times n}$.
- Matrix pencils arise naturally in differential-algebraic equations and in linear time invariant control systems

$$-B\dot{x} = Ax + Fu, \qquad y = Cx \tag{1}$$

  by taking Laplace transforms.
- The pencil $A + \lambda B$ is regular if its characteristic polynomial $p(\lambda) = \det(A + \lambda B)$ is NOT identically zero. Otherwise, the pencil is singular, i.e., if $p(\lambda) = \det(A + \lambda B) \equiv 0$.
- The regularity of $A + \lambda B$ implies that a solution of (1) exists for all smooth enough controls and for consistent initial conditions.
- This existence is no longer guaranteed if the pencil $A + \lambda B$ is singular. Therefore, the distance of a regular pencil $A + \lambda B$ to a nearest singular pencil is a measure of the robustness of the problem (1).

- This talk deals with square complex matrix pencils $A + \lambda B \in \mathbb{C}[\lambda]_1^{n\times n}$ or polynomial matrices of degree 1, where $A, B \in \mathbb{C}^{n\times n}$.
- Matrix pencils arise naturally in differential-algebraic equations and in linear time invariant control systems

$$-B\dot{x} = Ax + Fu, \qquad y = Cx \tag{1}$$

  by taking Laplace transforms.
- The pencil $A + \lambda B$ is regular if its characteristic polynomial $p(\lambda) = \det(A + \lambda B)$ is NOT identically zero. Otherwise, the pencil is singular, i.e., if $p(\lambda) = \det(A + \lambda B) \equiv 0$.
- The regularity of $A + \lambda B$ implies that a solution of (1) exists for all smooth enough controls and for consistent initial conditions.
- This existence is no longer guaranteed if the pencil $A + \lambda B$ is singular. Therefore, the distance of a regular pencil $A + \lambda B$ to a nearest singular pencil is a measure of the robustness of the problem (1).

Problem

Given a square regular pencil $A + \lambda B \in \mathbb{C}[\lambda]_1^{n\times n}$ find a singular pencil nearest to it.

We measure the distances in Frobenius norm:

$$\|A + \lambda B\|_F := \left\| \begin{bmatrix} A & B \end{bmatrix} \right\|_F .$$

It is also possible and interesting to look for a nearest *real* singular pencil when A and B are real. The approach we present can be extended to the real case, though it is "technically" considerably more involved and it is under development.

Problem (Refined)

Given $A + \lambda B \in \mathbb{C}[\lambda]_1^{n\times n}$, find a minimizer for the distance $\|(A + \lambda B) - (S + \lambda T)\|_F$ amongst all pencils $S + \lambda T \in \mathbb{C}[\lambda]_1^{n\times n}$ that satisfy $\det(S + \lambda T) \equiv 0$.

Problem

Given a square regular pencil $A + \lambda B \in \mathbb{C}[\lambda]_1^{n\times n}$ find a singular pencil nearest to it.

We measure the distances in Frobenius norm:

$$\|A + \lambda B\|_F := \left\| \begin{bmatrix} A & B \end{bmatrix} \right\|_F .$$

It is also possible and interesting to look for a nearest *real* singular pencil when A and B are real. The approach we present can be extended to the real case, though it is "technically" considerably more involved and it is under development.

Problem (Refined)

Given $A + \lambda B \in \mathbb{C}[\lambda]_1^{n\times n}$, find a minimizer for the distance $\|(A + \lambda B) - (S + \lambda T)\|_F$ amongst all pencils $S + \lambda T \in \mathbb{C}[\lambda]_1^{n\times n}$ that satisfy $\det(S + \lambda T) \equiv 0$.

The problem

Problem

Given a square regular pencil $A + \lambda B \in \mathbb{C}[\lambda]_1^{n\times n}$ find a singular pencil nearest to it.

We measure the distances in Frobenius norm:

$$\|A + \lambda B\|_F := \left\| \begin{bmatrix} A & B \end{bmatrix} \right\|_F .$$

It is also possible and interesting to look for a nearest *real* singular pencil when A and B are real. The approach we present can be extended to the real case, though it is "technically" considerably more involved and it is under development.

Problem (Refined)

Given $A + \lambda B \in \mathbb{C}[\lambda]_1^{n\times n}$, find a minimizer for the distance $\|(A + \lambda B) - (S + \lambda T)\|_F$ amongst all pencils $S + \lambda T \in \mathbb{C}[\lambda]_1^{n\times n}$ that satisfy $\det(S + \lambda T) \equiv 0$.

Problem

Given a square regular pencil $A + \lambda B \in \mathbb{C}[\lambda]_1^{n\times n}$ find a singular pencil nearest to it.

We measure the distances in Frobenius norm:

$$\|A + \lambda B\|_F := \left\| \begin{bmatrix} A & B \end{bmatrix} \right\|_F .$$

It is also possible and interesting to look for a nearest *real* singular pencil when A and B are real. The approach we present can be extended to the real case, though it is "technically" considerably more involved and it is under development.

Problem (Refined)

Given $A + \lambda B \in \mathbb{C}[\lambda]_1^{n\times n}$, find a minimizer for the distance $\|(A + \lambda B) - (S + \lambda T)\|_F$ amongst all pencils $S + \lambda T \in \mathbb{C}[\lambda]_1^{n\times n}$ that satisfy $\det(S + \lambda T) \equiv 0$.

Previous works (I)

- **The problem was posed for the first time by Byers, He and Mehrmann in 1998**, who collected lower and upper bounds and proposed several characterizations of the distance to singularity but could not provide an exact solution (except in very special cases) nor an efficient algorithm.

R. Byers, C. He and V. Mehrmann, Where is the nearest non-regular pencil?, Linear Algebra Appl., 285 (1998) 81–105.

- D. Kressner and M. Voigt wrote in 2015 the survey

D. Kressner and M. Voigt, Distance problems for linear dynamical systems, in: P. Benner, M. Bollhöfer, D. Kressner, C. Mehl, T. Stykel (Editors), Numerical Algebra, Matrix Theory, Differential-Algebraic Equations and Control Theory (pp. 559-583), Springer, Berlin, 2015.

where they comment

"...explicit formulas are not known for the general case and, even worse, devising a numerical method for computing the distance to singularity or bounds thereof turned out to be an extremely difficult problem. Since the publication of [15], almost no progress has been made in this direction."

Previous works (I)

- **The problem was posed for the first time by Byers, He and Mehrmann in 1998**, who collected lower and upper bounds and proposed several characterizations of the distance to singularity but could not provide an exact solution (except in very special cases) nor an efficient algorithm.

R. Byers, C. He and V. Mehrmann, Where is the nearest non-regular pencil?, Linear Algebra Appl., 285 (1998) 81–105.

- D. Kressner and M. Voigt wrote in 2015 the survey

D. Kressner and M. Voigt, Distance problems for linear dynamical systems, in: P. Benner, M. Bollhöfer, D. Kressner, C. Mehl, T. Stykel (Editors), Numerical Algebra, Matrix Theory, Differential-Algebraic Equations and Control Theory (pp. 559-583), Springer, Berlin, 2015.

where they comment

"...explicit formulas are not known for the general case and, even worse, devising a numerical method for computing the distance to singularity or bounds thereof turned out to be an extremely difficult problem. Since the publication of [15], almost no progress has been made in this direction."

Previous works (II)

Since then, several works have been published on this problem. We mention the following ones:

- M. Giesbrecht, J. Haraldson and G. Labahn presented in 2017 a method based on structured perturbations of mosaic Toeplitz matrices with an asymptotic complexity of $O(n^{12})$ flops per iteration.

M. Giesbrecht, J. Haraldson and G. Labahn, Computing the nearest rank-deficient matrix polynomial, in: Proceedings of the 2017 ACM on International Symposium on Symbolic and Algebraic Computation (pp. 181-188), 2017.

- N. Guglielmi, C. Lubich and V. Mehrmann presented in 2017 an ODE-approach based on expressing the set of $n \times n$ singular pencils as those pencils whose characteristic polynomial is zero when it is evaluated in $n+1$ given distinct points.

N. Guglielmi, C. Lubich and V. Mehrmann, On the nearest singular matrix pencil, SIAM J. Matrix Anal. Appl., 38(3) (2017) 776–806.

This method needs to compute the smallest singular value of n matrices in each iteration and therefore is slow, though it is flexible for incorporating structures and for being extended to polynomial matrices.

Previous works (II)

Since then, several works have been published on this problem. We mention the following ones:

- M. Giesbrecht, J. Haraldson and G. Labahn presented in 2017 a method based on structured perturbations of mosaic Toeplitz matrices with an asymptotic complexity of $O(n^{12})$ flops per iteration.

M. Giesbrecht, J. Haraldson and G. Labahn, Computing the nearest rank-deficient matrix polynomial, in: Proceedings of the 2017 ACM on International Symposium on Symbolic and Algebraic Computation (pp. 181-188), 2017.

- N. Guglielmi, C. Lubich and V. Mehrmann presented in 2017 an ODE-approach based on expressing the set of $n \times n$ singular pencils as those pencils whose characteristic polynomial is zero when it is evaluated in $n+1$ given distinct points.

N. Guglielmi, C. Lubich and V. Mehrmann, On the nearest singular matrix pencil, SIAM J. Matrix Anal. Appl., 38(3) (2017) 776–806.

This method needs to compute the smallest singular value of n matrices in each iteration and therefore is slow, though it is flexible for incorporating structures and for being extended to polynomial matrices.

Previous works (III)

- B. Das and S. Bora have presented in 2023 a method based on structured perturbations of the Gantmacher's block Toeplitz matrices associated with a pencil and on a careful analysis of the properties of singular polynomial matrices.

B. Das and S. Bora, Nearest rank deficient matrix polynomials, Linear Algebra Appl., 674 (2023) 304–350.

This method is still very slow, but much more efficient than previous methods.

In summary:

1. **The problem is very difficult:**
   - no general solution formula exists,
   - the running time of all the numerical methods proposed so far is very high even for pencils of moderate size,
   - the number of local minima seems to increase fast with the size of the pencil, making it hard to find global minima (which in general are not unique).
2. The existing methods rely generally on either
   - ODE-based techniques or
   - structured perturbations of (potentially very large for moderate sizes) block (or mosaic) Toeplitz matrices.
3. The method in this talk uses a novel approach based on Riemannian optimization inspired in the recent work by V. Noferini and F. Poloni (2021)

V. Noferini and F. Poloni, Nearest Ω-stable matrix via Riemannian optimization, Numer. Math., 148(4) (2021) 817–851.

for solving some pure matrix nearness problems (not pencil).

In summary:

1. **The problem is very difficult:**
   - no general solution formula exists,
   - the running time of all the numerical methods proposed so far is very high even for pencils of moderate size,
   - the number of local minima seems to increase fast with the size of the pencil, making it hard to find global minima (which in general are not unique).
2. The existing methods rely generally on either
   - ODE-based techniques or
   - structured perturbations of (potentially very large for moderate sizes) block (or mosaic) Toeplitz matrices.
3. The method in this talk uses a novel approach based on Riemannian optimization inspired in the recent work by V. Noferini and F. Poloni (2021)

V. Noferini and F. Poloni, Nearest Ω-stable matrix via Riemannian optimization, Numer. Math., 148(4) (2021) 817–851.

for solving some pure matrix nearness problems (not pencil).

1. **The problem is very difficult:**
   - no general solution formula exists,
   - the running time of all the numerical methods proposed so far is very high even for pencils of moderate size,
   - the number of local minima seems to increase fast with the size of the pencil, making it hard to find global minima (which in general are not unique).
2. The existing methods rely generally on either
   - ODE-based techniques or
   - structured perturbations of (potentially very large for moderate sizes) block (or mosaic) Toeplitz matrices.
3. The method in this talk uses a novel approach based on Riemannian optimization inspired in the recent work by V. Noferini and F. Poloni (2021)

V. Noferini and F. Poloni, Nearest Ω-stable matrix via Riemannian optimization, Numer. Math., 148(4) (2021) 817–851.

for solving some pure matrix nearness problems (not pencil).

- **Pros:**
  - Relatively fast → works in reasonable times for larger pencils than previous approaches (e.g. 100×100).
  - Yields competitive candidate solutions.
  - Publicly available and easy to use.
- **Cons:**
  - It cannot be (at least easily) extended to find the nearest singular polynomial matrix to a given regular polynomial matrix of arbitrary degree.
  - It is less flexible than the ODE-approach for dealing with pencils with particular structures.

Pros and cons of Riemannian optimization for nearest singular pencil

- **Pros:**
  - Relatively fast → works in reasonable times for larger pencils than previous approaches (e.g. 100×100).
  - Yields competitive candidate solutions.
  - Publicly available and easy to use.
- **Cons:**
  - It cannot be (at least easily) extended to find the nearest singular polynomial matrix to a given regular polynomial matrix of arbitrary degree.
  - It is less flexible than the ODE-approach for dealing with pencils with particular structures.

Reformulating the problem (I)

The main tool for the reformulation is the Generalized Schur form [Stewart, 1972] of matrix pencils, which let us split the problem in

- finding a nearest singular *upper triangular* pencil and
- solving a minimization problem over unitary matrices (justified later).

We denote by $U(n)$ the set of $n \times n$ unitary matrices.

Theorem (Generalized Schur form)

For any pair $A, B \in \mathbb{C}^{n \times n}$ there exist $Q, Z \in U(n)$ such that QAZ and QBZ are both upper triangular.

Lemma (Singular upper triangular pencil)

An upper triangular square pencil $A + \lambda B$ is singular if and only if it has at least one zero diagonal element.

The main tool for the reformulation is the Generalized Schur form [Stewart, 1972] of matrix pencils, which let us split the problem in

- finding a nearest singular *upper triangular* pencil and
- solving a minimization problem over unitary matrices (justified later).

We denote by $U(n)$ the set of $n \times n$ unitary matrices.

Theorem (Generalized Schur form)

For any pair $A, B \in \mathbb{C}^{n\times n}$ there exist $Q, Z \in U(n)$ such that QAZ and QBZ are both upper triangular.

Lemma (Singular upper triangular pencil)

An upper triangular square pencil $A + \lambda B$ is singular if and only if it has at least one zero diagonal element.

The main tool for the reformulation is the Generalized Schur form [Stewart, 1972] of matrix pencils, which let us split the problem in

- finding a nearest singular *upper triangular* pencil and
- solving a minimization problem over unitary matrices (justified later).

We denote by $U(n)$ the set of $n \times n$ unitary matrices.

Theorem (Generalized Schur form)

For any pair $A, B \in \mathbb{C}^{n\times n}$ there exist $Q, Z \in U(n)$ such that QAZ and QBZ are both upper triangular.

Lemma (Singular upper triangular pencil)

An upper triangular square pencil $A + \lambda B$ is singular if and only if it has at least one zero diagonal element.

The main tool for the reformulation is the Generalized Schur form [Stewart, 1972] of matrix pencils, which let us split the problem in

- finding a nearest singular *upper triangular* pencil and
- solving a minimization problem over unitary matrices (justified later).

We denote by $U(n)$ the set of $n \times n$ unitary matrices.

Theorem (Generalized Schur form)

For any pair $A, B \in \mathbb{C}^{n\times n}$ there exist $Q, Z \in U(n)$ such that QAZ and QBZ are both upper triangular.

Lemma (Singular upper triangular pencil)

An upper triangular square pencil $A + \lambda B$ is singular if and only if it has at least one zero diagonal element.

Proposition (Nearest singular upper triangular pencil)

Let $A + \lambda B \in \mathbb{C}[\lambda]_1^{n\times n}$. *Let* k *be any index such that*

$$|A_{kk}|^2 + |B_{kk}|^2 = \min_{1\le i\le n} \{|A_{ii}|^2 + |B_{ii}|^2\}.$$

An upper triangular singular pencil nearest to $A + \lambda B$ *is* $\mathcal{P}(A) + \lambda \mathcal{P}(B)$ *where*

$$\mathcal{P}(A)_{ij} = \begin{cases} A_{ij} & \text{if } i<j \text{ or } i=j\neq k; \\ 0 & \text{otherwise;} \end{cases} \qquad \mathcal{P}(B)_{ij} = \begin{cases} B_{ij} & \text{if } i<j \text{ or } i=j\neq k; \\ 0 & \text{otherwise;} \end{cases}$$

i.e., $\mathcal{P}(A)$ *and* $\mathcal{P}(B)$ *are obtained by setting to zero the lower triangular parts of* A *and* B*, respectively, and* A_{kk} *and* B_{kk}.

In particular, the squared distance of $A + \lambda B$ *from* $\mathcal{P}(A) + \lambda \mathcal{P}(B)$ *is*

$$\mathcal{F}(A + \lambda B) = \sum_{i>j} (|A_{ij}|^2 + |B_{ij}|^2) + \min_{1\le i\le n} \{|A_{ii}|^2 + |B_{ii}|^2\}.$$

Reformulating the problem (II)

Proposition (Nearest singular upper triangular pencil)

Let $A + \lambda B \in \mathbb{C}[\lambda]_1^{n\times n}$. Let k be any index such that

$$|A_{kk}|^2 + |B_{kk}|^2 = \min_{1\le i\le n} \{|A_{ii}|^2 + |B_{ii}|^2\}.$$

An upper triangular singular pencil nearest to $A + \lambda B$ is $\mathcal{P}(A) + \lambda\mathcal{P}(B)$ where

$$\mathcal{P}(A)_{ij} = \begin{cases} A_{ij} & \text{if } i < j \text{ or } i = j \neq k; \\ 0 & \text{otherwise}; \end{cases} \qquad \mathcal{P}(B)_{ij} = \begin{cases} B_{ij} & \text{if } i < j \text{ or } i = j \neq k; \\ 0 & \text{otherwise}; \end{cases}$$

i.e., $\mathcal{P}(A)$ and $\mathcal{P}(B)$ are obtained by setting to zero the lower triangular parts of A and B, respectively, and A_{kk} and B_{kk}.

In particular, the squared distance of $A + \lambda B$ from $\mathcal{P}(A) + \lambda\mathcal{P}(B)$ is

$$\mathcal{F}(A + \lambda B) = \sum_{i>j}(|A_{ij}|^2 + |B_{ij}|^2) + \min_{1\le i\le n} \{|A_{ii}|^2 + |B_{ii}|^2\}.$$

Reformulating the problem (III)

Theorem (Nearest singular pencil via minimization over $U(n) \times U(n)$)

If $A + \lambda B \in \mathbb{C}[\lambda]_1^{n\times n}$, then the squared distance of $A + \lambda B$ to a nearest singular pencil is

$$\min_{(Q,Z)\in U(n)\times U(n)} f(Q,Z),$$

where

$$f(Q,Z) := \mathcal{F}(QAZ + \lambda QBZ) = \|(QAZ + \lambda QBZ) - (\mathcal{P}(QAZ) + \lambda \mathcal{P}(QBZ))\|_F^2.$$

Moreover, if (Q_0, Z_0) is a global minimizer of $f(Q,Z)$ over $U(n) \times U(n)$, then the pencil

$$Q_0^* \mathcal{P}(Q_0 A Z_0) Z_0^* + \lambda Q_0^* \mathcal{P}(Q_0 B Z_0) Z_0^*$$

is a singular pencil nearest to $A + \lambda B$.

Theorem (Nearest singular pencil via minimization over $U(n) \times U(n)$)

If $A + \lambda B \in \mathbb{C}[\lambda]_1^{n\times n}$, then the squared distance of $A + \lambda B$ to a nearest singular pencil is

$$\min_{(Q,Z)\in U(n)\times U(n)} f(Q,Z),$$

where

$$f(Q,Z) := \mathcal{F}(QAZ + \lambda QBZ) = \|(QAZ + \lambda QBZ) - (\mathcal{P}(QAZ) + \lambda \mathcal{P}(QBZ))\|_F^2.$$

Moreover, if (Q_0, Z_0) is a global minimizer of $f(Q,Z)$ over $U(n) \times U(n)$, then the pencil

$$Q_0^* \mathcal{P}(Q_0 A Z_0) Z_0^* + \lambda Q_0^* \mathcal{P}(Q_0 B Z_0) Z_0^*$$

is a singular pencil nearest to $A + \lambda B$.

Reformulating the problem (IV)

Proof.

Let $\mathcal{S}_n, \mathcal{T}_n \subset \mathbb{C}[\lambda]_1^{n\times n}$ denote the set of singular pencils and the set of singular upper triangular pencils, respectively. Then,

$$\begin{aligned}
\min_{S+\lambda T\in\mathcal{S}_n} \|(A-S)+\lambda(B-T)\|_F^2 &= \min_{Q,Z\in U(n)} \min_{X+\lambda Y\in\mathcal{T}_n} \|(A-Q^*XZ^*)+\lambda(B-Q^*YZ^*)\|_F^2 \\
&= \min_{Q,Z\in U(n)} \min_{X+\lambda Y\in\mathcal{T}_n} \|(QAZ-X)+\lambda(QBZ-Y)\|_F^2 \\
&= \min_{Q,Z\in U(n)} \mathcal{F}(QAZ+\lambda QBZ).
\end{aligned}$$

□

Summary of the reformulation

- Squared distance of $A + \lambda B$ to a nearest singular upper triangular pencil is

$$\mathcal{F}(A + \lambda B) = \sum_{i>j} (|A_{ij}|^2 + |B_{ij}|^2) + \min_{1 \le i \le n} (|A_{ii}|^2 + |B_{ii}|^2).$$

- The objective function in $U(n) \times U(n)$ is

$$f(Q, Z) := \mathcal{F}(QAZ + \lambda QBZ).$$

We are interested in finding

$$(Q_0, Z_0) \in \underset{(Q,Z) \in U(n) \times U(n)}{\operatorname{argmin}} f(Q, Z).$$

- A singular pencil nearest to $A + \lambda B$ is given by

$$Q_0^* \mathcal{P}(Q_0 A Z_0) Z_0^* + \lambda Q_0^* \mathcal{P}(Q_0 B Z_0) Z_0^*.$$

Outline

Minimizing the objective function

- How to find

$$(Q_0, Z_0) \in \underset{(Q,Z)\in U(n)\times U(n)}{\operatorname{argmin}} f(Q,Z)?$$

- We use **MATLAB toolbox Manopt 7.1 for optimization on matrix manifolds**, in particular its `trustregions` method.

N. Boumal, B. Mishra, P. A. Absil and R. Sepulchre, Manopt, a Matlab toolbox for optimization on manifolds, The Journal of Machine Learning Research, 15(1) (2014) 1455-1459.

P. A. Absil, R. Mahony and R. Sepulchre, Optimization Algorithms on Matrix Manifolds, Princeton University Press, 2008.

N. Boumal, An Introduction to Optimization on Smooth Manifolds, Cambridge University Press, 2023.

- Problem is **non-convex**: computed minimum is not necessarily global.
- Manopt requires for high-efficiency that the user provides MATLAB functions for the **Riemannian gradient** and the **Riemannian Hessian** on the manifold $U(n) \times U(n)$ of the objective function.
- For brevity, we only explain how to obtain the **Riemannian gradient**.

Minimizing the objective function

- How to find

$$(Q_0, Z_0) \in \underset{(Q,Z)\in U(n)\times U(n)}{\operatorname{argmin}} f(Q,Z)?$$

- We use **MATLAB toolbox Manopt 7.1 for optimization on matrix manifolds**, in particular its `trustregions` method.

N. Boumal, B. Mishra, P. A. Absil and R. Sepulchre, Manopt, a Matlab toolbox for optimization on manifolds, The Journal of Machine Learning Research, 15(1) (2014) 1455-1459.

P. A. Absil, R. Mahony and R. Sepulchre, Optimization Algorithms on Matrix Manifolds, Princeton University Press, 2008.

N. Boumal, An Introduction to Optimization on Smooth Manifolds, Cambridge University Press, 2023.

- Problem is **non-convex**: computed minimum is not necessarily global.
- Manopt requires for high-efficiency that the user provides MATLAB functions for the **Riemannian gradient** and the **Riemannian Hessian** on the manifold $U(n) \times U(n)$ of the objective function.
- For brevity, we only explain how to obtain the **Riemannian gradient**.

Minimizing the objective function

- How to find

$$(Q_0, Z_0) \in \underset{(Q,Z)\in U(n)\times U(n)}{\operatorname{argmin}} f(Q,Z)?$$

- We use **MATLAB toolbox Manopt 7.1 for optimization on matrix manifolds**, in particular its `trustregions` method.

N. Boumal, B. Mishra, P. A. Absil and R. Sepulchre, Manopt, a Matlab toolbox for optimization on manifolds, The Journal of Machine Learning Research, 15(1) (2014) 1455-1459.

P. A. Absil, R. Mahony and R. Sepulchre, Optimization Algorithms on Matrix Manifolds, Princeton University Press, 2008.

N. Boumal, An Introduction to Optimization on Smooth Manifolds, Cambridge University Press, 2023.

- Problem is **non-convex**: computed minimum is not necessarily global.
- Manopt requires for high-efficiency that the user provides MATLAB functions for the **Riemannian gradient** and the **Riemannian Hessian** on the manifold $U(n) \times U(n)$ of the objective function.
- For brevity, we only explain how to obtain the **Riemannian gradient**.

Minimizing the objective function

- How to find

$$(Q_0, Z_0) \in \underset{(Q,Z)\in U(n)\times U(n)}{\operatorname{argmin}} f(Q,Z)?$$

- We use **MATLAB toolbox Manopt 7.1 for optimization on matrix manifolds**, in particular its `trustregions` method.

N. Boumal, B. Mishra, P. A. Absil and R. Sepulchre, Manopt, a Matlab toolbox for optimization on manifolds, The Journal of Machine Learning Research, 15(1) (2014) 1455-1459.

P. A. Absil, R. Mahony and R. Sepulchre, Optimization Algorithms on Matrix Manifolds, Princeton University Press, 2008.

N. Boumal, An Introduction to Optimization on Smooth Manifolds, Cambridge University Press, 2023.

- Problem is **non-convex**: computed minimum is not necessarily global.
- Manopt requires for high-efficiency that the user provides MATLAB functions for the **Riemannian gradient** and the **Riemannian Hessian** on the manifold $U(n) \times U(n)$ of the objective function.
- For brevity, we only explain how to obtain the **Riemannian gradient**.

Minimizing the objective function

- How to find

$$(Q_0, Z_0) \in \underset{(Q,Z)\in U(n)\times U(n)}{\operatorname{argmin}} f(Q,Z)?$$

- We use **MATLAB toolbox Manopt 7.1 for optimization on matrix manifolds**, in particular its `trustregions` method.

N. Boumal, B. Mishra, P. A. Absil and R. Sepulchre, Manopt, a Matlab toolbox for optimization on manifolds, The Journal of Machine Learning Research, 15(1) (2014) 1455-1459.

P. A. Absil, R. Mahony and R. Sepulchre, Optimization Algorithms on Matrix Manifolds, Princeton University Press, 2008.

N. Boumal, An Introduction to Optimization on Smooth Manifolds, Cambridge University Press, 2023.

- Problem is **non-convex**: computed minimum is not necessarily global.
- Manopt requires for high-efficiency that the user provides MATLAB functions for the **Riemannian gradient** and the **Riemannian Hessian** on the manifold $U(n) \times U(n)$ of the objective function.
- For brevity, we only explain how to obtain the **Riemannian gradient**.

Minimizing the objective function

- How to find

$$(Q_0, Z_0) \in \underset{(Q,Z)\in U(n)\times U(n)}{\operatorname{argmin}} f(Q,Z)?$$

- We use **MATLAB toolbox Manopt 7.1 for optimization on matrix manifolds**, in particular its `trustregions` method.

N. Boumal, B. Mishra, P. A. Absil and R. Sepulchre, Manopt, a Matlab toolbox for optimization on manifolds, The Journal of Machine Learning Research, 15(1) (2014) 1455-1459.

P. A. Absil, R. Mahony and R. Sepulchre, Optimization Algorithms on Matrix Manifolds, Princeton University Press, 2008.

N. Boumal, An Introduction to Optimization on Smooth Manifolds, Cambridge University Press, 2023.

- Problem is **non-convex**: computed minimum is not necessarily global.
- Manopt requires for high-efficiency that the user provides MATLAB functions for the **Riemannian gradient** and the **Riemannian Hessian** on the manifold $U(n) \times U(n)$ of the objective function.
- For brevity, we only explain how to obtain the **Riemannian gradient**.

Formula for the Riemannian gradient (I)

- Consider the ambient real vector space $\mathbb{C}^{n\times n} \times \mathbb{C}^{n\times n} (\cong \mathbb{R}^{4n^2})$ equipped with the inner product

$$\langle (A_1, A_2), (B_1, B_2) \rangle = \mathcal{R}e \left(\text{trace}(A_1^* B_1) + \text{trace}(A_2^* B_2) \right).$$

- Obtain the expression of the standard Euclidean gradient at $(Q, Z) \in U(n) \times U(n)$ in the real ambient space $\mathbb{C}^{n\times n} \times \mathbb{C}^{n\times n}$

$$\nabla_{(Q,Z)} f(Q,Z) = (\nabla_Q f(Q,Z) \, , \, \nabla_Z f(Q,Z))$$

where

$$\nabla_Q f(Q,Z) = 2\, L(QAZ)\, (AZ)^* + 2\, L(QBZ)\, (BZ)^*,$$
$$\nabla_Z f(Q,Z) = 2\, (QA)^*\, L(QAZ) + 2\, (QB)^*\, L(QBZ),$$

and $L(A) + \lambda L(B) := (A - \mathcal{P}(A)) + \lambda(B - \mathcal{P}(B))$ and $\mathcal{P}(A) + \lambda \mathcal{P}(B)$ is an upper triangular singular pencil nearest to $A + \lambda B$.

- Consider the ambient real vector space $\mathbb{C}^{n\times n} \times \mathbb{C}^{n\times n} (\cong \mathbb{R}^{4n^2})$ equipped with the inner product

$$\langle (A_1, A_2), (B_1, B_2) \rangle = \mathcal{R}e \left(\text{trace}(A_1^* B_1) + \text{trace}(A_2^* B_2) \right).$$

- Obtain the expression of the standard Euclidean gradient at $(Q, Z) \in U(n) \times U(n)$ in the real ambient space $\mathbb{C}^{n\times n} \times \mathbb{C}^{n\times n}$

$$\nabla_{(Q,Z)} f(Q,Z) = (\nabla_Q f(Q,Z) \,,\, \nabla_Z f(Q,Z))$$

where

$$\nabla_Q f(Q,Z) = 2\, L(QAZ)\,(AZ)^* + 2\, L(QBZ)\,(BZ)^*,$$
$$\nabla_Z f(Q,Z) = 2\,(QA)^*\, L(QAZ) + 2\,(QB)^*\, L(QBZ),$$

and $L(A) + \lambda L(B) := (A - \mathcal{P}(A)) + \lambda(B - \mathcal{P}(B))$ and $\mathcal{P}(A) + \lambda \mathcal{P}(B)$ is an upper triangular singular pencil nearest to $A + \lambda B$.

Formula for the Riemannian gradient (I)

- Consider the ambient real vector space $\mathbb{C}^{n\times n} \times \mathbb{C}^{n\times n} (\cong \mathbb{R}^{4n^2})$ equipped with the inner product

$$\langle (A_1, A_2), (B_1, B_2) \rangle = \mathcal{R}e\left(\text{trace}(A_1^* B_1) + \text{trace}(A_2^* B_2)\right).$$

- Obtain the expression of the standard Euclidean gradient at $(Q, Z) \in U(n) \times U(n)$ in the real ambient space $\mathbb{C}^{n\times n} \times \mathbb{C}^{n\times n}$

$$\nabla_{(Q,Z)} f(Q,Z) = (\nabla_Q f(Q,Z), \nabla_Z f(Q,Z))$$

where

$$\nabla_Q f(Q,Z) = 2\, L(QAZ)\,(AZ)^* + 2\, L(QBZ)\,(BZ)^*,$$
$$\nabla_Z f(Q,Z) = 2\,(QA)^*\, L(QAZ) + 2\,(QB)^*\, L(QBZ),$$

and $L(A) + \lambda L(B) := (A - \mathcal{P}(A)) + \lambda(B - \mathcal{P}(B))$ and $\mathcal{P}(A) + \lambda\mathcal{P}(B)$ is an upper triangular singular pencil nearest to $A + \lambda B$.

Formula for the Riemannian gradient (II)

- Obtain the orthogonal projection of $\nabla_{(Q,Z)} f(Q,Z)$ onto the tangent space at (Q,Z) of $U(n) \times U(n)$. This tangent space is

$$T_{(Q,Z)}(U(n) \times U(n)) = (T_Q U(n)) \times (T_Z U(n))$$
$$= \{(Q S_Q, Z S_Z) : S_Q = -S_Q^*, S_Z = -S_Z^*\}.$$

- The orthogonal projection of $(M_1, M_2) \in \mathbb{C}^{n\times n} \times \mathbb{C}^{n\times n}$ onto the tangent space $T_{(Q,Z)}(U(n) \times U(n))$ is

$$\mathrm{Proj}_{T_{(Q,Z)}}(M_1, M_2) = (Q\,\mathsf{skew}(Q^* M_1), Z\,\mathsf{skew}(Z^* M_2))$$

where

$$\mathsf{skew}(W) = \frac{1}{2}(W - W^*),$$

- which allows us to finally obtain the Riemannian gradient as

$$\mathrm{Proj}_{T_{(Q,Z)}} \nabla_{(Q,Z)} f(Q,Z) = \mathrm{Proj}_{T_{(Q,Z)}}(\nabla_Q f(Q,Z), \nabla_Z f(Q,Z)).$$

Formula for the Riemannian gradient (II)

- Obtain the orthogonal projection of $\nabla_{(Q,Z)} f(Q,Z)$ onto the tangent space at (Q,Z) of $U(n) \times U(n)$. This tangent space is

$$T_{(Q,Z)}(U(n) \times U(n)) = (T_Q U(n)) \times (T_Z U(n))$$
$$= \{(Q S_Q, Z S_Z) : S_Q = -S_Q^*, S_Z = -S_Z^*\}.$$

- The orthogonal projection of $(M_1, M_2) \in \mathbb{C}^{n\times n} \times \mathbb{C}^{n\times n}$ onto the tangent space $T_{(Q,Z)}(U(n) \times U(n))$ is

$$\mathrm{Proj}_{T_{(Q,Z)}}(M_1, M_2) = (Q\,\mathrm{skew}(Q^* M_1), Z\,\mathrm{skew}(Z^* M_2))$$

where

$$\mathrm{skew}(W) = \frac{1}{2}(W - W^*),$$

- which allows us to finally obtain the Riemannian gradient as

$$\mathrm{Proj}_{T_{(Q,Z)}} \nabla_{(Q,Z)} f(Q,Z) = \mathrm{Proj}_{T_{(Q,Z)}}(\nabla_Q f(Q,Z), \nabla_Z f(Q,Z)).$$

Formula for the Riemannian gradient (II)

- Obtain the orthogonal projection of $\nabla_{(Q,Z)} f(Q,Z)$ onto the tangent space at (Q,Z) of $U(n) \times U(n)$. This tangent space is

$$T_{(Q,Z)}\,(U(n) \times U(n)) = (T_Q\, U(n)) \times (T_Z\, U(n)) \\ = \{(Q\, S_Q\,, Z\, S_Z\,) \,:\, S_Q = -S_Q^*, S_Z = -S_Z^*\}.$$

- The orthogonal projection of $(M_1, M_2) \in \mathbb{C}^{n\times n} \times \mathbb{C}^{n\times n}$ onto the tangent space $T_{(Q,Z)}\,(U(n) \times U(n))$ is

$$\mathrm{Proj}_{T_{(Q,Z)}}(M_1, M_2) = (Q\, \mathsf{skew}(Q^* M_1), Z\, \mathsf{skew}(Z^* M_2))$$

where

$$\mathsf{skew}(W) = \frac{1}{2}(W - W^*),$$

- which allows us to finally obtain the Riemannian gradient as

$$\mathrm{Proj}_{T_{(Q,Z)}}\, \nabla_{(Q,Z)} f(Q,Z) = \mathrm{Proj}_{T_{(Q,Z)}}(\nabla_Q f(Q,Z)\,,\, \nabla_Z f(Q,Z))\,.$$

Outline

- We benchmark against the ODE-approach [Guglielmi et al., 2017].
- We use 10^3 complex random 6×6 pencils.
- Statistical comparisons with much larger pencils are not feasible because the current implementation of the ODE-approach is too slow.
- Real and imaginary parts of the matrix coefficients are drawn from $\mathcal{N}(0, 1)$.

Method	Frequency of best output	Median distance	Average distance
ODE	37.3 %	1.8925	2.0601
Riemann	62.7 %	1.8042	1.8231

- We benchmark against the ODE-approach [Guglielmi et al., 2017].
- We use 10^3 complex random 6×6 pencils.
- Statistical comparisons with much larger pencils are not feasible because the current implementation of the ODE-approach is too slow.
- Real and imaginary parts of the matrix coefficients are drawn from $\mathcal{N}(0, 1)$.

Method	Frequency of best output	Median distance	Average distance
ODE	37.3 %	1.8925	2.0601
Riemann	62.7 %	1.8042	1.8231

- We benchmark against the ODE-approach [Guglielmi et al., 2017].
- We use 10^3 complex random 6×6 pencils.
- Statistical comparisons with much larger pencils are not feasible because the current implementation of the ODE-approach is too slow.
- Real and imaginary parts of the matrix coefficients are drawn from $\mathcal{N}(0, 1)$.

Method	Frequency of best output	Median distance	Average distance
ODE	37.3 %	1.8925	2.0601
Riemann	62.7 %	1.8042	1.8231

Numerical experiment I: Comparison with ODE-approach

- We benchmark against the ODE-approach [Guglielmi et al., 2017].
- We use 10^3 complex random 6×6 pencils.
- Statistical comparisons with much larger pencils are not feasible because the current implementation of the ODE-approach is too slow.
- Real and imaginary parts of the matrix coefficients are drawn from $\mathcal{N}(0,1)$.

Method	Frequency of best output	Median distance	Average distance
ODE	37.3 %	1.8925	2.0601
Riemann	62.7 %	1.8042	1.8231

- We benchmark against the ODE-approach [Guglielmi et al., 2017].
- We use 10^3 complex random 6×6 pencils.
- Statistical comparisons with much larger pencils are not feasible because the current implementation of the ODE-approach is too slow.
- Real and imaginary parts of the matrix coefficients are drawn from $\mathcal{N}(0, 1)$.

Method	Frequency of best output	Median distance	Average distance
ODE	37.3 %	1.8925	2.0601
Riemann	62.7 %	1.8042	1.8231

Numerical experiment II: Comparison with Das-Bora algorithm (1)

- We benchmark against the Das-Bora algorithm [Das and Bora, 2023].
- For each $n = 6, 15, 30, 50$ we use 10^3 complex random $n \times n$ pencils as in Experiment I.
- Statistical comparisons with larger pencils are not feasible because the Das-Bora algorithm is too slow. (We sincerely thank Das and Bora for providing the MATLAB codes of their algorithm).
- The quality of the output of the Riemannian algorithm was typically worse than that of Das-Bora algorithm for very small inputs $n = 6, 15$, but slightly better for $n = 30$ and already much better for $n = 50$.
- In terms of running time, the Riemannian algorithm outperformed Das-Bora algorithm already for $n = 15$; for $n = 50$ the difference was already striking, with a ratio of average running times ≈ 29 in favour of our method.

Numerical experiment II: Comparison with Das-Bora algorithm (1)

- We benchmark against the Das-Bora algorithm [Das and Bora, 2023].
- For each $n = 6, 15, 30, 50$ we use 10^3 complex random $n \times n$ pencils as in Experiment I.
- Statistical comparisons with larger pencils are not feasible because the Das-Bora algorithm is too slow. (We sincerely thank Das and Bora for providing the MATLAB codes of their algorithm).
- The quality of the output of the Riemannian algorithm was typically worse than that of Das-Bora algorithm for very small inputs $n = 6, 15$, but slightly better for $n = 30$ and already much better for $n = 50$.
- In terms of running time, the Riemannian algorithm outperformed Das-Bora algorithm already for $n = 15$; for $n = 50$ the difference was already striking, with a ratio of average running times ≈ 29 in favour of our method.

- We benchmark against the Das-Bora algorithm [Das and Bora, 2023].
- For each $n = 6, 15, 30, 50$ we use 10^3 complex random $n \times n$ pencils as in Experiment I.
- Statistical comparisons with larger pencils are not feasible because the Das-Bora algorithm is too slow. (We sincerely thank Das and Bora for providing the MATLAB codes of their algorithm).
- The quality of the output of the Riemannian algorithm was typically worse than that of Das-Bora algorithm for very small inputs $n = 6, 15$, but slightly better for $n = 30$ and already much better for $n = 50$.
- In terms of running time, the Riemannian algorithm outperformed Das-Bora algorithm already for $n = 15$; for $n = 50$ the difference was already striking, with a ratio of average running times ≈ 29 in favour of our method.

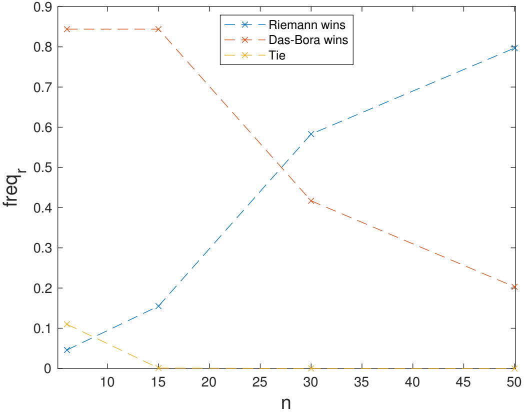
Numerical experiment II: Comparison with Das-Bora algorithm (1)

- We benchmark against the Das-Bora algorithm [Das and Bora, 2023].
- For each $n = 6, 15, 30, 50$ we use 10^3 complex random $n \times n$ pencils as in Experiment I.
- Statistical comparisons with larger pencils are not feasible because the Das-Bora algorithm is too slow. (We sincerely thank Das and Bora for providing the MATLAB codes of their algorithm).
- The quality of the output of the Riemannian algorithm was typically worse than that of Das-Bora algorithm for very small inputs $n = 6, 15$, but slightly better for $n = 30$ and already much better for $n = 50$.
- In terms of running time, the Riemannian algorithm outperformed Das-Bora algorithm already for $n = 15$; for $n = 50$ the difference was already striking, with a ratio of average running times ≈ 29 in favour of our method.

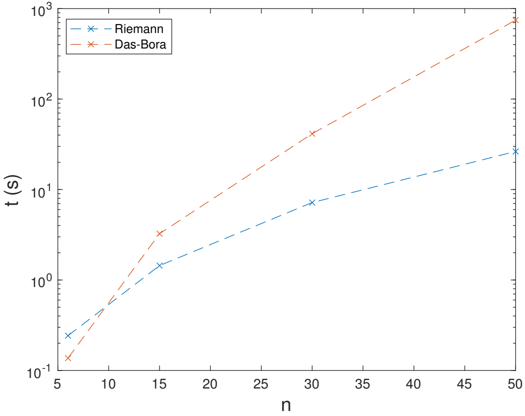
Numerical experiment II: Comparison with Das-Bora algorithm (1)

- We benchmark against the Das-Bora algorithm [Das and Bora, 2023].
- For each $n = 6, 15, 30, 50$ we use 10^3 complex random $n \times n$ pencils as in Experiment I.
- Statistical comparisons with larger pencils are not feasible because the Das-Bora algorithm is too slow. (We sincerely thank Das and Bora for providing the MATLAB codes of their algorithm).
- The quality of the output of the Riemannian algorithm was typically worse than that of Das-Bora algorithm for very small inputs $n = 6, 15$, but slightly better for $n = 30$ and already much better for $n = 50$.
- In terms of running time, the Riemannian algorithm outperformed Das-Bora algorithm already for $n = 15$; for $n = 50$ the difference was already striking, with a ratio of average running times ≈ 29 in favour of our method.

Numerical experiment II: Comparison with Das-Bora algorithm (2)



Comparison of the quality of the output between the Riemannian method and the Das-Bora algorithm for $n \in \{6, 15, 30, 50\}$. The performance profile reports the relative frequency of which method yielded a better solution.



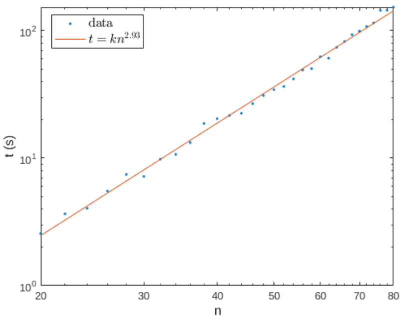
Comparison of the running time between the Riemannian method and the Das-Bora algorithm for $n \in \{6, 15, 30, 50\}$. Running times were measured using MATLAB R2023a on an Intel Core i5-12600K.

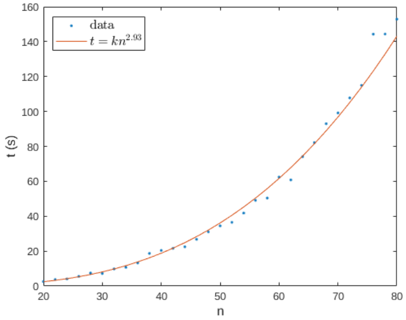
- **How large pencils can we handle?**
- How does the running time change with the matrix size?
- For each n, we generate random $n \times n$ pencils as before, and measure the running time.

- **How large pencils can we handle?**
- How does the running time change with the matrix size?
- For each n, we generate random $n \times n$ pencils as before, and measure the running time.

- **How large pencils can we handle?**
- How does the running time change with the matrix size?
- For each n, we generate random $n \times n$ pencils as before, and measure the running time.

Numerical experiment III (2)



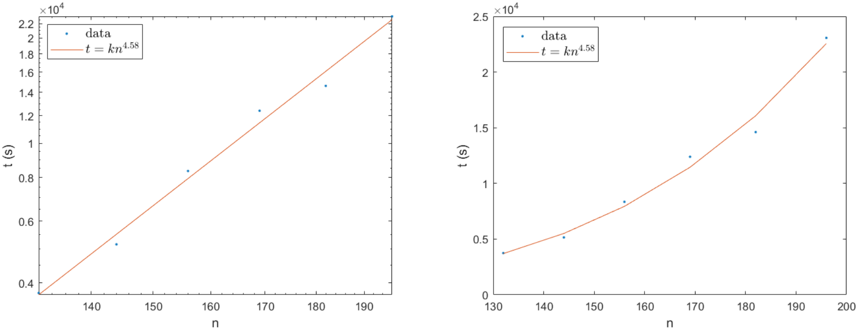


Average running time (of 50 runs) in logarithmic scale (left) and linear scale (right) for $20 \le n \le 80$. The least squares fit yields approximately

$$t = k\,n^{2.93},$$

where $k \approx 3.8310 \times 10^{-4}$. We used MATLAB R2023a and an Intel Core i5-12600K Processor.

Numerical experiment III (3)



Average running time (of 24 runs) in logarithmic scale (left) and linear scale (right) for $130 \leq n \leq 200$. The least squares fit yields approximately

$$t = k\,n^{4.58},$$

where $k \approx 7.3423 \cdot 10^{-7}$. We used MATLAB R2023a and its internal parallelization with 24 processes on a 2x12 core Xeon E5 2690 v3 2.60GHz. The computational resources were provided by the Aalto Science-IT project.

Outline

Nearest singular pencil with fixed minimal index (I)

- Almost all $n \times n$ singular pencils have normal rank $n-1$, i.e., the set of singular pencils with normal rank $n-1$ is open and dense in the set of singular pencils.
- Moreover almost all $n \times n$ singular pencils have only one left minimal index, only one right minimal index and no eigenvalues.

P. Van Dooren, The generalized eigenstructure problem: Applications in linear system theory, PhD thesis, Kath. Univ. Leuven, Leuven, Belgium, 1979.

F. De Terán and F.M. Dopico, A note on generic Kronecker orbits of matrix pencils with fixed rank, SIAM J. Matrix Anal. Appl., 30 (2008) 491–496

- We have exploited this type of ideas to prove that the squared distance of $A + \lambda B \in \mathbb{C}[\lambda]_1^{n \times n}$ to a nearest singular upper triangular pencil with right minimal index m, $0 \le m \le n-1$, is

$$\mathcal{F}_m(A + \lambda B) = \sum_{i>j} (|A_{ij}|^2 + |B_{ij}|^2) + (|A_{m+1,m+1}|^2 + |B_{m+1,m+1}|^2).$$

Nearest singular pencil with fixed minimal index (I)

- Almost all $n \times n$ singular pencils have normal rank $n-1$, i.e., the set of singular pencils with normal rank $n-1$ is open and dense in the set of singular pencils.
- Moreover almost all $n \times n$ singular pencils have only one left minimal index, only one right minimal index and no eigenvalues.

P. Van Dooren. The generalized eigenstructure problem: Applications in linear system theory. PhD thesis, Kath. Univ. Leuven, Leuven, Belgium, 1979.

F. De Terán and F.M. Dopico, A note on generic Kronecker orbits of matrix pencils with fixed rank, SIAM J. Matrix Anal. Appl., 30 (2008) 491–496

- We have exploited this type of ideas to prove that the squared distance of $A + \lambda B \in \mathbb{C}[\lambda]_1^{n\times n}$ to a nearest singular upper triangular pencil with right minimal index m, $0 \leq m \leq n-1$, is

$$\mathcal{F}_m(A + \lambda B) = \sum_{i>j}(|A_{ij}|^2 + |B_{ij}|^2) + (|A_{m+1,m+1}|^2 + |B_{m+1,m+1}|^2).$$

Nearest singular pencil with fixed minimal index (I)

- Almost all $n \times n$ singular pencils have normal rank $n-1$, i.e., the set of singular pencils with normal rank $n-1$ is open and dense in the set of singular pencils.
- Moreover almost all $n \times n$ singular pencils have only one left minimal index, only one right minimal index and no eigenvalues.

P. Van Dooren. The generalized eigenstructure problem: Applications in linear system theory. PhD thesis, Kath. Univ. Leuven, Leuven, Belgium, 1979.

F. De Terán and F.M. Dopico, A note on generic Kronecker orbits of matrix pencils with fixed rank, SIAM J. Matrix Anal. Appl., 30 (2008) 491–496

- We have exploited this type of ideas to prove that the squared distance of $A + \lambda B \in \mathbb{C}[\lambda]_1^{n\times n}$ to a nearest singular upper triangular pencil with right minimal index m, $0 \leq m \leq n-1$, is

$$\mathcal{F}_m(A+\lambda B) = \sum_{i>j} (|A_{ij}|^2 + |B_{ij}|^2) + (|A_{m+1,m+1}|^2 + |B_{m+1,m+1}|^2).$$

Nearest singular pencil with fixed minimal index (I)

- Almost all $n \times n$ singular pencils have normal rank $n-1$, i.e., the set of singular pencils with normal rank $n-1$ is open and dense in the set of singular pencils.
- Moreover almost all $n \times n$ singular pencils have only one left minimal index, only one right minimal index and no eigenvalues.

P. Van Dooren. The generalized eigenstructure problem: Applications in linear system theory. PhD thesis, Kath. Univ. Leuven, Leuven, Belgium, 1979.

F. De Terán and F.M. Dopico, A note on generic Kronecker orbits of matrix pencils with fixed rank, SIAM J. Matrix Anal. Appl., 30 (2008) 491–496

- We have exploited this type of ideas to prove that the squared distance of $A + \lambda B \in \mathbb{C}[\lambda]_1^{n \times n}$ to a nearest singular upper triangular pencil with right minimal index m, $0 \leq m \leq n-1$, is

$$\mathcal{F}_m(A + \lambda B) = \sum_{i>j} (|A_{ij}|^2 + |B_{ij}|^2) + (|A_{m+1,m+1}|^2 + |B_{m+1,m+1}|^2).$$

Nearest singular pencil with fixed minimal index (II)

- This

$$\mathcal{F}_m(A+\lambda B) = \sum_{i>j}(|A_{ij}|^2 + |B_{ij}|^2) + (|A_{m+1,m+1}|^2 + |B_{m+1,m+1}|^2).$$

should be compared with the following result **seen earlier** "the squared distance of $A+\lambda B$ to a nearest singular upper triangular pencil" is

$$\mathcal{F}(A+\lambda B) = \sum_{i>j}(|A_{ij}|^2 + |B_{ij}|^2) + \min_{1\le i\le n}\{|A_{ii}|^2 + |B_{ii}|^2\},$$

This

$$\mathcal{F}_m(A+\lambda B) = \sum_{i>j}(|A_{ij}|^2 + |B_{ij}|^2) + (|A_{m+1,m+1}|^2 + |B_{m+1,m+1}|^2).$$

should be compared with the following result **seen earlier** "the squared distance of $A+\lambda B$ to a nearest singular upper triangular pencil" is

$$\mathcal{F}(A+\lambda B) = \sum_{i>j}(|A_{ij}|^2 + |B_{ij}|^2) + \min_{1\le i\le n}\{|A_{ii}|^2 + |B_{ii}|^2\},$$

where we observe that $\mathcal{F}_m(A+\lambda B)$ is always a smooth function.

- Using $\mathcal{F}_m(A+\lambda B)$ and arguments similar to those in the first part of the talk, we prove that one can find a nearest singular pencil with right

Nearest singular pencil with fixed minimal index (II)

- This

$$\mathcal{F}_m(A+\lambda B) = \sum_{i>j}(|A_{ij}|^2 + |B_{ij}|^2) + (|A_{m+1,m+1}|^2 + |B_{m+1,m+1}|^2).$$

should be compared with the following result **seen earlier** “the squared distance of $A+\lambda B$ to a nearest singular upper triangular pencil” is

$$\mathcal{F}(A+\lambda B) = \sum_{i>j}(|A_{ij}|^2 + |B_{ij}|^2) + \min_{1\le i\le n}\{|A_{ii}|^2 + |B_{ii}|^2\},$$

where we observe that $\mathcal{F}_m(A+\lambda B)$ is always a smooth function.

- Using $\mathcal{F}_m(A+\lambda B)$ and arguments similar to those in the first part of the talk, we prove that one can find a nearest singular pencil with right minimal index m to $A+\lambda B$ by minimizing the objective function

$$f_m(Q,Z) := \mathcal{F}_m(QAZ+\lambda QBZ)$$

over $U(n)\times U(n)$.

Nearest singular pencil with fixed minimal index (II)

- This

$$\mathcal{F}_m(A+\lambda B) = \sum_{i>j}(|A_{ij}|^2 + |B_{ij}|^2) + (|A_{m+1,m+1}|^2 + |B_{m+1,m+1}|^2).$$

should be compared with the following result **seen earlier** "the squared distance of $A+\lambda B$ to a nearest singular upper triangular pencil" is

$$\mathcal{F}(A+\lambda B) = \sum_{i>j}(|A_{ij}|^2 + |B_{ij}|^2) + \min_{1\leq i\leq n}\{|A_{ii}|^2 + |B_{ii}|^2\},$$

where we observe that $\mathcal{F}_m(A+\lambda B)$ is always a smooth function.

- Using $\mathcal{F}_m(A+\lambda B)$ and arguments similar to those in the first part of the talk, we prove that one can find a nearest singular pencil with right minimal index m to $A+\lambda B$ by minimizing the objective function

$$f_m(Q,Z) := \mathcal{F}_m(QAZ + \lambda QBZ)$$

over $U(n)\times U(n)$.

Outline

Conclusions

- We have described a novel algorithm to compute the nearest singular pencil to a given one, based on Riemannian optimization.
- The new method makes it practically feasible, for the first time, to solve the problem for pencils of moderate size, say, a few hundreds rows-columns.
- The Riemannian method does better than other methods in terms of the quality of the output when the size of the problem is not very small.
- Furthermore, the performance is also very favourable to the new algorithm in terms of computational time.
- For example, on average for randomly generated inputs of size $n = 50$ and using one of the authors' personal computer, the new method converged in about 25 seconds while Das-Bora algorithm required more than 12 minutes.
- We have also proved that the new Riemannian approach can be adapted with minor algorithmic modifications to compute the nearest singular pencil with prescribed (right) minimal index.

Conclusions

- We have described a novel algorithm to compute the nearest singular pencil to a given one, based on Riemannian optimization.
- The new method makes it practically feasible, for the first time, to solve the problem for pencils of moderate size, say, a few hundreds rows-columns.
- The Riemannian method does better than other methods in terms of the quality of the output when the size of the problem is not very small.
- Furthermore, the performance is also very favourable to the new algorithm in terms of computational time.
- For example, on average for randomly generated inputs of size $n = 50$ and using one of the authors' personal computer, the new method converged in about 25 seconds while Das-Bora algorithm required more than 12 minutes.
- We have also proved that the new Riemannian approach can be adapted with minor algorithmic modifications to compute the nearest singular pencil with prescribed (right) minimal index.

Conclusions

- We have described a novel algorithm to compute the nearest singular pencil to a given one, based on Riemannian optimization.
- The new method makes it practically feasible, for the first time, to solve the problem for pencils of moderate size, say, a few hundreds rows-columns.
- The Riemannian method does better than other methods in terms of the quality of the output when the size of the problem is not very small.
- Furthermore, the performance is also very favourable to the new algorithm in terms of computational time.
- For example, on average for randomly generated inputs of size $n = 50$ and using one of the authors' personal computer, the new method converged in about 25 seconds while Das-Bora algorithm required more than 12 minutes.
- We have also proved that the new Riemannian approach can be adapted with minor algorithmic modifications to compute the nearest singular pencil with prescribed (right) minimal index.

Conclusions

- We have described a novel algorithm to compute the nearest singular pencil to a given one, based on Riemannian optimization.
- The new method makes it practically feasible, for the first time, to solve the problem for pencils of moderate size, say, a few hundreds rows-columns.
- The Riemannian method does better than other methods in terms of the quality of the output when the size of the problem is not very small.
- Furthermore, the performance is also very favourable to the new algorithm in terms of computational time.
- For example, on average for randomly generated inputs of size $n = 50$ and using one of the authors' personal computer, the new method converged in about 25 seconds while Das-Bora algorithm required more than 12 minutes.
- We have also proved that the new Riemannian approach can be adapted with minor algorithmic modifications to compute the nearest singular pencil with prescribed (right) minimal index.

Conclusions

- We have described a novel algorithm to compute the nearest singular pencil to a given one, based on Riemannian optimization.
- The new method makes it practically feasible, for the first time, to solve the problem for pencils of moderate size, say, a few hundreds rows-columns.
- The Riemannian method does better than other methods in terms of the quality of the output when the size of the problem is not very small.
- Furthermore, the performance is also very favourable to the new algorithm in terms of computational time.
- For example, on average for randomly generated inputs of size $n = 50$ and using one of the authors' personal computer, the new method converged in about 25 seconds while Das-Bora algorithm required more than 12 minutes.
- We have also proved that the new Riemannian approach can be adapted with minor algorithmic modifications to compute the nearest singular pencil with prescribed (right) minimal index.

Conclusions

- We have described a novel algorithm to compute the nearest singular pencil to a given one, based on Riemannian optimization.
- The new method makes it practically feasible, for the first time, to solve the problem for pencils of moderate size, say, a few hundreds rows-columns.
- The Riemannian method does better than other methods in terms of the quality of the output when the size of the problem is not very small.
- Furthermore, the performance is also very favourable to the new algorithm in terms of computational time.
- For example, on average for randomly generated inputs of size $n = 50$ and using one of the authors' personal computer, the new method converged in about 25 seconds while Das-Bora algorithm required more than 12 minutes.
- We have also proved that the new Riemannian approach can be adapted with minor algorithmic modifications to compute the nearest singular pencil with prescribed (right) minimal index.