Inverse-free and permuted bases methods for algebraic Riccati equations

Federico Poloni[1]

Contains joint work with: T. Haqiri[2], V. Mehrmann[3], N. Strabić[4]

1: Department of Computer Science, University of Pisa, Italy.
2: Shahid Bahonar University, Kerman, Iran.
3: Institute for Mathematics, Technical University of Berlin, Germany.
4: University of Manchester, UK.

7th European Congress of Mathematics
Berlin, July 2016

Equations and subspaces

What we all do here ~~solving algebraic Riccati equations~~ computing invariant subspaces.

$$A^*X + XA + Q - XGX = 0 \iff \begin{bmatrix} A & -G \\ -Q & -A^* \end{bmatrix} \begin{bmatrix} I \\ X \end{bmatrix} = \begin{bmatrix} I \\ X \end{bmatrix} (A - GX)$$

CARE $\iff$ $H\mathcal{U} \subseteq \mathcal{U}$ with subspace in the Riccati basis $\begin{bmatrix} I \\ X \end{bmatrix}$.

Usually a good idea to use other bases: what if e.g. $\mathcal{U} \approx \operatorname{im} \begin{bmatrix} 1 & 1 \\ 1 & 1 \\ 2 & 1 \\ 2 & 2 \end{bmatrix}$?

Permuted Riccati bases

We can get an identity in correspondence of any invertible submatrix.

Example

$$U = \begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \\ 1 & 1 & 2 \\ 3 & 5 & 8 \end{bmatrix} \sim \begin{bmatrix} 1 & 0 & 0 \\ 0.5 & 0.5 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \\ 2 & 0 & 1 \end{bmatrix}.$$

We write this as $U \sim P \left[\begin{smallmatrix} I \\ Y \end{smallmatrix} \right]$, P permutation matrix.

$\sim$ notation for "spans the same subspace as".

Theorem [Knuth '84, Gu–Eisenstat '96]

Each full-col-rank U has a permuted Riccati basis $P \left[\begin{smallmatrix} I \\ Y \end{smallmatrix} \right]$ with $|Y_{ij}| \leq 1$

Why is it good? Identity + small entries = well-conditioned basis matrix.

How to compute them?

The theory Choose submatrix B with maximal $|\det B|$. Cramer's rule on

$$\begin{bmatrix}\text{row of } Y\end{bmatrix} = \begin{bmatrix}\text{row of } U\end{bmatrix} B^{-1} \quad \text{gives} \quad |Y_{ij}| = \frac{|\det(\text{other submatrix})|}{|\det B|} \leq 1.$$

The practice Find $|Y_{ij}| > 1$, update basis simplex-algorithm–style

$$\begin{bmatrix} \boxed{1 \quad 0 \quad 0} \\ 0.5 \quad 0.5 \quad 0 \\ 0 \quad 1 \quad 0 \\ 0 \quad 0 \quad 1 \\ \boxed{2 \quad 0 \quad 1} \end{bmatrix} \begin{array}{l} \leftarrow \text{this row out} \\ \\ \\ \\ \leftarrow \text{this row in} \end{array}$$

Principal pivot transform

The update process viewed in term of Y:

$$Y = \begin{bmatrix} \alpha & u \\ v^* & Y_{22} \end{bmatrix} \mapsto \begin{bmatrix} \alpha^{-1} & u\alpha^{-1} \\ -\alpha^{-1}v^* & Y_{22} - u\alpha^{-1}v^* \end{bmatrix}.$$

Known as PPT [Tsatsomeros,'00]; interesting "partial inversion" structure.

Quiz: apply this n times on a $n \times n$ matrix...

$$Y = \begin{bmatrix} * & * & * \\ * & * & * \\ * & * & * \end{bmatrix} \mapsto \begin{bmatrix} * & * & * \\ * & * & * \\ * & * & * \end{bmatrix} \mapsto \begin{bmatrix} * & * & * \\ * & * & * \\ * & * & * \end{bmatrix} \mapsto \begin{bmatrix} * & * & * \\ * & * & * \\ * & * & * \end{bmatrix} = ??$$

What's happening?

Principal pivot transform

The update process viewed in term of Y:

$$Y = \begin{bmatrix} \alpha & u \\ v^* & Y_{22} \end{bmatrix} \mapsto \begin{bmatrix} \alpha^{-1} & u\alpha^{-1} \\ -\alpha^{-1}v^* & Y_{22} - u\alpha^{-1}v^* \end{bmatrix}.$$

Known as PPT [Tsatsomeros,'00]; interesting "partial inversion" structure.

Quiz: apply this n times on a $n \times n$ matrix...

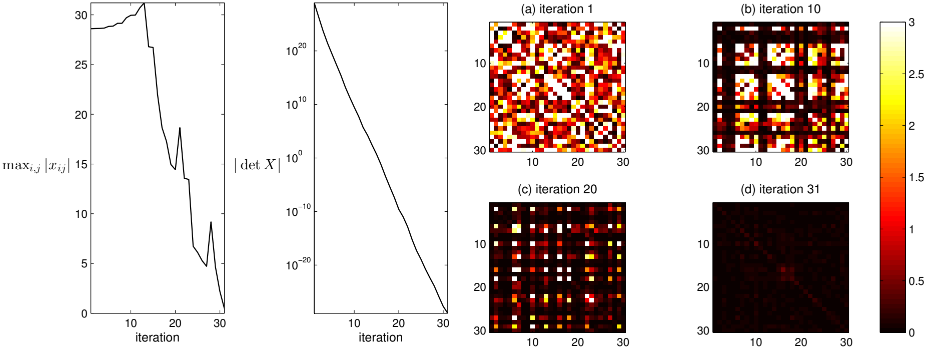
$$Y = \begin{bmatrix} * & * & * \\ * & * & * \\ * & * & * \end{bmatrix} \mapsto \begin{bmatrix} * & * & * \\ * & * & * \\ * & * & * \end{bmatrix} \mapsto \begin{bmatrix} * & * & * \\ * & * & * \\ * & * & * \end{bmatrix} \mapsto \begin{bmatrix} * & * & * \\ * & * & * \\ * & * & * \end{bmatrix} = ??$$

What's happening?

Answer: $?? = Y^{-1}$, because we switch from $\begin{bmatrix} I \\ Y \end{bmatrix}$ to $\begin{bmatrix} Y^{-1} \\ I \end{bmatrix}$.

Gaussian elimination in disguise. "Alien linear algebra".

Reduction process



Pictures from [P. Strabić '15]

A structured version

Why is $P\left[\begin{smallmatrix} I \\ Y \end{smallmatrix}\right]$ preferable to orthonormal bases?
Because it has a structure-preserving version!

To solve Riccati equations, we need Lagrangian subspaces.
Image of $U \in \mathbb{C}^{2n\times n}$ Lagrangian if $U^H J_{2n} U = 0$, with $J_{2n} = \left[\begin{smallmatrix} 0 & I_n \\ -I_n & 0 \end{smallmatrix}\right]$.
In Riccati basis: im $\left[\begin{smallmatrix} I \\ X \end{smallmatrix}\right]$ Lagrangian $\iff$ X Hermitian.

Theorem [Mehrmann, P. '12]

Every Lagrangian subspace im U has a Lagrangian permuted graph basis $U \sim P\left[\begin{smallmatrix} I \\ Y \end{smallmatrix}\right]$ with:

- P permutation+sign changes,
- $Y = Y^H$,
- $|Y_{ij}| \leq \sqrt{2}$.

Lagrangian permuted Riccati bases

Example

$$\begin{bmatrix}1 & 0 & 0\\ 0 & 1 & 0\\ 0 & 0 & 1\\ 1 & 2 & 3\\ 2 & 4 & 5\\ 3 & 5 & 6\end{bmatrix} \sim \begin{bmatrix}1 & 0 & 0\\ 0 & 1 & 0\\ -1/2 & -5/6 & 1/6\\ -1/2 & -1/2 & 1/2\\ -1/2 & -1/6 & 5/6\\ 0 & 0 & 1\end{bmatrix}.$$

Permutations here can only swap $i \leftrightarrow n+i$.

Allows us to store and operate on exactly Lagrangian subspace stably.

An application: Riccati verification [Haqiri P. '16]

Problem

Compute a guaranteed enclosure for the stabilizing solution of a CARE $A^*X + XA + Q - XGX = 0$ in $\mathcal{O}(n^3)$.

Last week in ILAS conference (ask me for the slides!); one of the ideas:

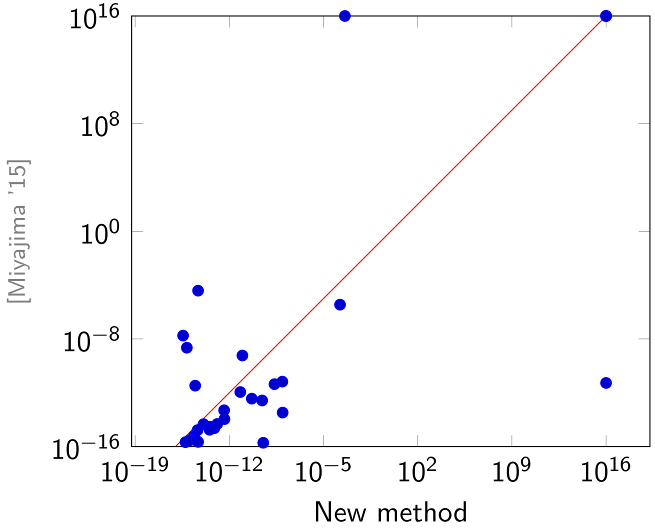
$$\begin{bmatrix} A & -G \\ -Q & -A^* \end{bmatrix} \begin{bmatrix} I \\ X \end{bmatrix} = \begin{bmatrix} I \\ X \end{bmatrix} (A - GX)$$

replaced by a CARE for Y:

$$\begin{bmatrix} \hat{A} & -\hat{G} \\ -\hat{Q} & -\hat{A}^* \end{bmatrix} \begin{bmatrix} I \\ Y \end{bmatrix} = \begin{bmatrix} I \\ Y \end{bmatrix} (\hat{A} - \hat{G}Y), \quad \begin{bmatrix} \hat{A} & -\hat{G} \\ -\hat{Q} & -\hat{A}^* \end{bmatrix} = P^{-1} \begin{bmatrix} A & -G \\ -Q & -A^* \end{bmatrix} P.$$

Then, $X = U_2 U_1^{-1}$, with $\begin{bmatrix} U_1 \\ U_2 \end{bmatrix} = P \begin{bmatrix} I \\ Y \end{bmatrix}$.

Experiments: relative width of **X**



Results for pencils

Matrix pencils in a nutshell: a pair (L_0, L_1) "represents" the matrix $L_0^{-1}L_1$.

Some operations can be performed implicitly on the pair, no matter if L_0 is ill-conditioned or even singular (inverse-free algorithms).

In this talk: we only assume $\begin{bmatrix} L_0 & L_1 \end{bmatrix}$ has full row rank.

Definition

$$(L_0, L_1) \sim (M_0, M_1) \quad \text{if} \quad L_0 = BM_0, L_1 = BM_1 \text{ for } B \text{ square invertible.}$$

Note that

$$(L_0, L_1) \sim (M_0, M_1) \iff \begin{bmatrix} L_0 & L_1 \end{bmatrix}^H \sim \begin{bmatrix} M_0 & M_1 \end{bmatrix}^H.$$

So one can use results on subspaces to normalize pencils

Example

$$(L_0, L_1) \sim \left(\begin{bmatrix} 1 & * & 0 \\ 0 & * & 1 \\ 0 & * & 0 \end{bmatrix}, \begin{bmatrix} * & * & 0 \\ * & * & 0 \\ * & * & 1 \end{bmatrix} \right), \quad |*| \leq 1.$$

Results for structured pencils [Mehrmann P. '12]

Symplectic pencils: $L_i \in \mathbb{C}^{2n \times 2n}$ such that $L_1 J_{2n} L_1^H = L_0 J_{2n} L_0^H$.

$\begin{bmatrix} L_0 & L_1 \end{bmatrix}^H$ is essentially Lagrangian (after some row/sign changes).

$$\left(\begin{bmatrix} 1 & 0 & * & * \\ 0 & 1 & * & * \\ 0 & 0 & * & * \\ 0 & 0 & * & * \end{bmatrix}, \begin{bmatrix} * & * & 0 & 0 \\ * & * & 0 & 0 \\ * & * & 1 & 0 \\ * & * & 0 & 1 \end{bmatrix} \right).$$

- Among each two same-color columns, one is a column of I_{2n}
- The other entries satisfy $|*| \leq \sqrt{2}$, and can be pieced together (modulo signs) into a Hermitian matrix

Hamiltonian pencils: $L_i \in \mathbb{C}^{2n \times 2n}$ such that $L_1 J_{2n} L_0^H = -L_0 J_{2n} L_1^H$.

$$\left(\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \begin{bmatrix} * & * & * & * \\ * & * & * & * \\ * & * & * & * \\ * & * & * & * \end{bmatrix} \right).$$

Deflating the "3×3 control pencil" [Mehrmann P. '13]

Common control-theory structure:

$$\left(\begin{bmatrix} 0 & I_n & 0 \\ -I_n & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}, \begin{bmatrix} 0 & A & B \\ A^T & Q & S \\ B^T & S^T & R \end{bmatrix} \right).$$

Traditional way to handle it (recast in our language): first put an identity in

$$\left(\begin{bmatrix} \boxed{\begin{matrix} I_n & 0 \\ 0 & I_n \end{matrix}} & \begin{matrix} 0 \\ 0 \end{matrix} \\ \begin{matrix} 0 & 0 \end{matrix} & 0 \end{bmatrix}, \begin{bmatrix} \boxed{\begin{matrix} * & * \\ * & * \end{matrix}} & \begin{matrix} 0 \\ 0 \end{matrix} \\ \begin{matrix} * & * \end{matrix} & I_m \end{bmatrix} \right).$$

Block triangular; deflate and work on Hamiltonian pencil in orange.

Key point: invertibing R or determining its kernel.

A different deflation

Why must the identity go there?

$$\left(\begin{bmatrix} \boxed{\begin{matrix} * & * \\ * & * \end{matrix}} & \begin{matrix} 0 \\ 0 \end{matrix} \\ \begin{matrix} * & * \end{matrix} & 0 \end{bmatrix}, \begin{bmatrix} \boxed{\begin{matrix} * & * \\ * & * \end{matrix}} & \begin{matrix} 0 \\ 0 \end{matrix} \\ \begin{matrix} * & * \end{matrix} & I_m \end{bmatrix}\right)$$

Put columns of I in half of the green and blue columns.

Example Perturbation of 'the death pencil'

$$\left(\begin{bmatrix} 0 & 1 & 0 \\ -1 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}, \begin{bmatrix} 0 & 0 & 1 \\ 0 & 0 & 0 \\ 1 & 0 & \varepsilon \end{bmatrix}\right) \sim \left(\begin{bmatrix} \boxed{\begin{matrix} 1 & 0 \\ 0 & \varepsilon \end{matrix}} & \begin{matrix} 0 \\ 0 \end{matrix} \\ \begin{matrix} 0 & -1 \end{matrix} & 0 \end{bmatrix}, \begin{bmatrix} \boxed{\begin{matrix} 0 & 0 \\ 1 & 0 \end{matrix}} & \begin{matrix} 0 \\ 0 \end{matrix} \\ \begin{matrix} 0 & 0 \end{matrix} & 1 \end{bmatrix}\right)$$

The deflation process is well-conditioned no matter how small ε is.
(unlike many other algorithms.)

Preserving definiteness [P. Strabić '16]

Definition (new) If $U = \begin{bmatrix} I \\ X \end{bmatrix}$ with $X \succ 0$, we call im U definite Lagrangian. Semidefinite Lagrangian defined by continuity.

"Hidden structure" that plays a role in CARE theory, e.g., solution existence / semidefiniteness.

Theorem [P. Strabić '16]

If U Lagrangian (semi)definite, then all matrices Y appearing in $U \sim P\begin{bmatrix} I \\ Y \end{bmatrix}$ are quasidefinite (with blocks depending on P).

Most subspaces appearing in practice are Lagrangian semidefinite: e.g., Hamiltonian pencils from "ABCD" problems associated to

$$Y = \begin{bmatrix} -C^*C & A^* \\ A & BB^* \end{bmatrix}. \tag{1}$$

Algorithms to do PPTs updating generators A, B, C directly [P. Strabić '16], `https://bitbucket.org/fph/pgdoubling-quad`.

Solving dense Riccati equations

Doubling algorithm a "pencil version" of the matrix sign iteration $H \mapsto \frac{1}{2}(H + H^{-1})$:

$$(L_0, L_1) \mapsto (2M_0L_1, M_1L_1 + M_0L_0) \qquad \text{with } \begin{bmatrix} -M_0 & M_1 \end{bmatrix} \begin{bmatrix} L_1 \\ L_0 \end{bmatrix} = 0.$$

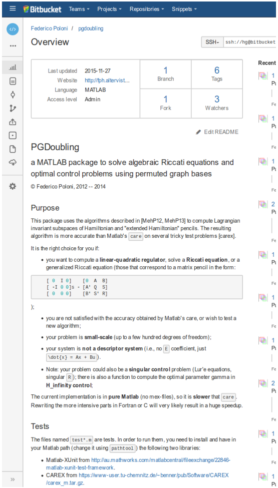
Two things needed at each step:

- Left kernel of $\begin{bmatrix} L_1 \\ L_0 \end{bmatrix}$ with permuted Riccati bases:

$$\begin{bmatrix} -Y & I \end{bmatrix} P^{-1} P \begin{bmatrix} I \\ Y \end{bmatrix} = 0.$$

- Hamiltonian pencil representation: permuted Riccati basis of $\begin{bmatrix} L_1^H \\ L_0^H \end{bmatrix}$.

Software: PGDoubling



Matlab library to work with dense Riccati equations and permuted Riccati bases.

⊕ More reliable than Matlab's care and other competing algorithms on benchmark examples.

⊖ Matlab code (no mex), not optimized for speed.

https://bitbucket.org/fph/pgdoubling

Figure: Relative subspace residual for the 33 CAREX problems in [Chu et al., '07]

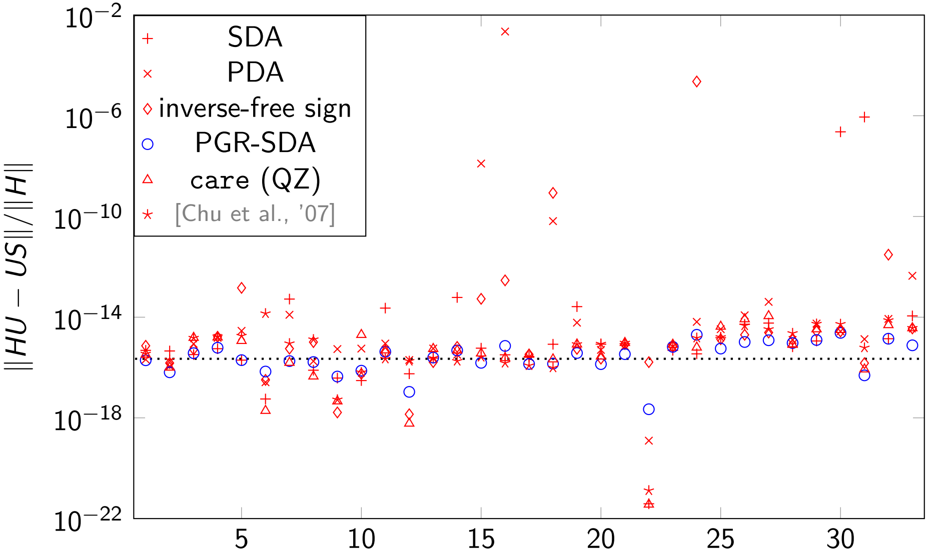
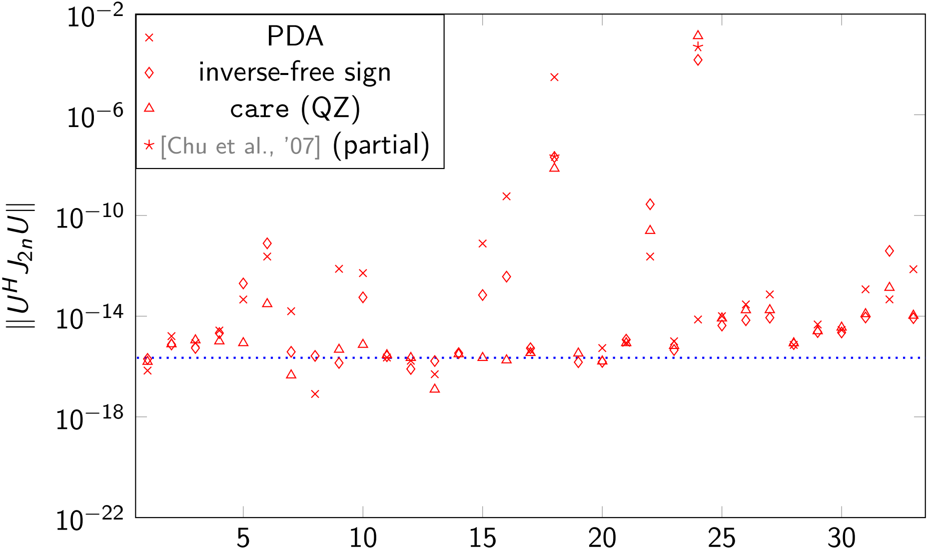


Figure: Lagrangianity residual for the 33 CAREX problems in [Chu et al., '07]



Large scale AREs

What we do ~~computing invariant subspaces~~ solving algebraic Riccati equations (in the large and sparse case).

- Solution subspaces can be represented cheaply as $U \sim \begin{bmatrix} I \\ ZZ^T \end{bmatrix}$, with Z tall skinny.
- Other bases, e.g. orthogonal, are not as practical.

A first attempt to use these ideas:

1. Run a standard solution algorithm (ADI) keeping not Z but (L_0, L_1) such that $Z = L_1 L_0^{-1}$.
2. Convert this to $\begin{bmatrix} I \\ ZZ^T \end{bmatrix} \sim \begin{bmatrix} U_1 \\ U_2 \end{bmatrix}$, with U_1, U_2 'storage-sparse'.

How does it work? No improvement in "typical" cases; subspace residual improves in some ill-conditioned examples, though.

Large scale AREs

Theorem [Mehrmann, P. '15]

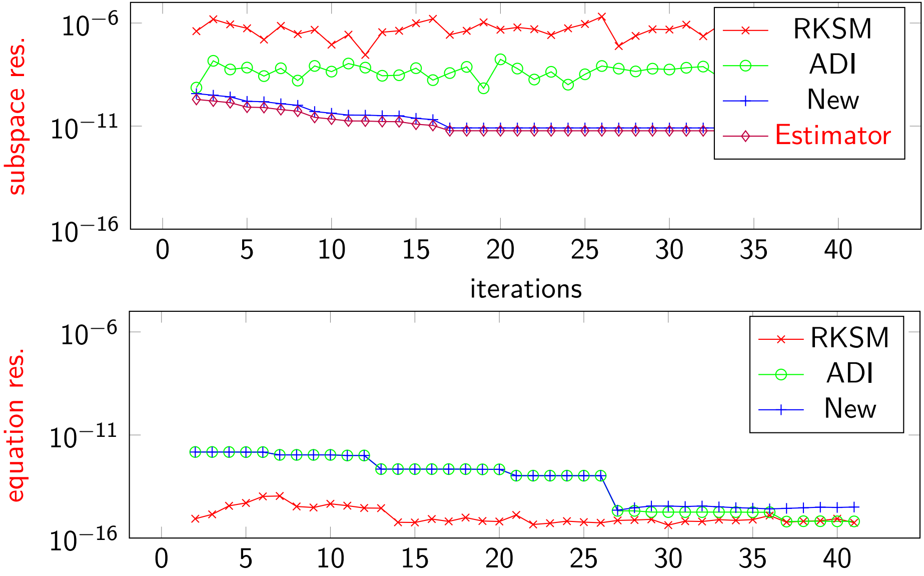
Given orthogonal $\begin{bmatrix} L_0 \\ L_1 \end{bmatrix}$ such that $Z = L_1 L_0^{-1} \in \mathbb{C}^{n\times m}$, we can build (quickly and stably) tall skinny V_i such that and

$$\begin{bmatrix} I \\ ZZ^T \end{bmatrix} \sim V = \begin{bmatrix} I - V_1 V_2^T \\ V_3 V_4^T \end{bmatrix}, \quad \kappa(V) \leq \frac{\sqrt{3}}{\sqrt{2}}(mn\tau^2 + n\tau).$$

Main idea: convert $X = \boxed{L_1}\,\boxed{L_0}^{-1}$ into $X = \boxed{M_0}^{-1}\,\boxed{M_1}$ using kernel trick

$$\begin{bmatrix} \boxed{Y} & \boxed{-I} \end{bmatrix} P^T P \begin{bmatrix} \boxed{I} \\ \boxed{Y} \end{bmatrix} = 0.$$

Experiments (random A, $B \approx$ smallest eigenvectors)



Conclusions

Useful primitives

- Converting between equivalent CAREs (e.g., in interval verification)
- Representing structured pencils (e.g., to deflate infinite eigenvalues)

Solving small dense CAREs

- Very robust solution algorithm.

Solving large-scale CAREs

- We can use these techniques also in sparse problems.
- Visible improvements only in edge cases (for now).

Take-home message:

- Bases with identities are a very promising tool for structured matrix computations. Try them!

Conclusions

Useful primitives

- Converting between equivalent CAREs (e.g., in interval verification)
- Representing structured pencils (e.g., to deflate infinite eigenvalues)

Solving small dense CAREs

- Very robust solution algorithm.

Solving large-scale CAREs

- We can use these techniques also in sparse problems.
- Visible improvements only in edge cases (for now).

Take-home message:

- Bases with identities are a very promising tool for structured matrix computations. Try them!

Thanks for your attention!