# The benefits of changing identity
## in Lagrangian subspaces and doubling algorithms

V. Mehrmann[1] Federico Poloni[1,2]

[1]Technische Universität Berlin, Institut für Mathematik
[2]Supported by the A. von Humboldt Foundation

Workshop on Matrix Equations and Tensor Techniques
Aachen, 21–22 November 2011

# Lagrangian subspaces

**Definition**

A subspace $\mathcal{U} = \operatorname{Im} U$ of $\mathbb{C}^{2n}$ is Lagrangian if it has dimension $n$ and

$$U^* \mathcal{J}_{2n} U = 0 \qquad \mathcal{J}_{2n} = \begin{bmatrix} 0 & I_n \\ -I_n & 0 \end{bmatrix}$$

- Property of the subspace, not of the basis: can post-multiply $U \to UM$
- They arise naturally as stable invariant subspaces of Hamiltonian and symplectic problems
- Central role in control

# Dealing with Lagrangian subspaces

Problem: find a (Lagrangian) invariant subspace of ...
but first: represent suitably a Lagrangian subspace, operate on it and return it to the user preserving structure

Subspace $\mathcal{U}$ often represented as range of (full column rank) $U$...
...but this is not unique: $\operatorname{Im} U = \operatorname{Im} UM$ for any $M$ nonsingular

# Graph bases and Riccati equations

One of the first choices [Willems, '71] : graph basis

$$\mathcal{U} = \operatorname{Im} \begin{bmatrix} U_1 \\ U_2 \end{bmatrix} = \operatorname{Im} \begin{bmatrix} I \\ X \end{bmatrix} \qquad X = U_2 U_1^{-1}$$

Transforms invariant subspace problems into algebraic Riccati equations

$$\begin{bmatrix} A & B \\ C & D \end{bmatrix} \begin{bmatrix} I_n \\ X \end{bmatrix} = \begin{bmatrix} I_n \\ X \end{bmatrix} F \quad \Longleftrightarrow \quad XBX + XA - DX - C = 0$$

- ☺ Easy to ensure Lagrangianity: $\mathcal{U}$ Lagrangian $\Leftrightarrow$ $X$ Hermitian
- ☹ Not all subspaces well represented: what about these?

$$\begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 1 & 0 \\ 0 & 1 \end{bmatrix} \qquad \begin{bmatrix} 1 & 1 \\ 1 & 1+\varepsilon \\ 1 & 0 \\ 0 & 1 \end{bmatrix}$$

# Orthogonal bases

Natural solution: orthogonal basis

$$\mathcal{U} = \operatorname{Im} \begin{bmatrix} Q_1 \\ Q_2 \end{bmatrix} \qquad \text{with orthogonal columns}$$

- ☺ All subspaces representable
- ☺ Very stable, no element growth
- ☹ Computationally more expensive
- ☹ Too many parameters: Lagrangianity $\Leftrightarrow Q_1^* Q_2 = Q_2^* Q_1$
  Easily lost through numerical computation, difficult to enforce explicitly

Loss of Lagrangianity is a serious problem, e.g. in Laub Trick [Laub, '79]

# Trying to save the Riccati approach

A "folklore result": in some cases useful to switch to $\begin{bmatrix} Y \\ I \end{bmatrix}$ (so $Y = X^{-1}$)

**Dual ARE**

$$XBX + XA - DX - C = 0 \quad \Longrightarrow \quad B + AY - YD - YCY = 0$$

But still both approaches can fail: e.g.,

$$\begin{bmatrix} 1 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 1 \end{bmatrix}$$

Neither $\begin{bmatrix} I \\ X \end{bmatrix}$ nor $\begin{bmatrix} Y \\ I \end{bmatrix}$ work

Idea: The identity that we are looking for is already there, in rows 1 and 4!

# Permuted graph bases

$$\begin{bmatrix} 1 & 0 \\ * & * \\ * & * \\ 0 & 1 \end{bmatrix}$$

We look for bases with an identity submatrix spread along different rows

$$= \Pi \begin{bmatrix} 1 & 0 \\ 0 & 1 \\ * & * \\ * & * \end{bmatrix}$$

Equivalently: keep identity on top, but premultiply with a permutation matrix

But permutations are not the right tool here: want to preserve Lagrangianity

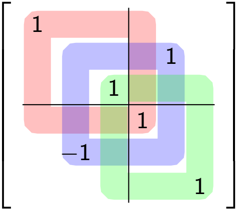
## The right thing: symplectic swap matrices

Symplectic row swap matrices: those that act (separately) as either

$$I = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \text{ or } \mathcal{J}_2 = \begin{bmatrix} 0 & 1 \\ -1 & 0 \end{bmatrix}$$

on every pair of indices $(i, n+i)$

Examples:




$$I_{2n} = \begin{bmatrix} I_n & 0 \\ 0 & I_n \end{bmatrix} \qquad \mathcal{J}_{2n} = \begin{bmatrix} 0 & I_n \\ -I_n & 0 \end{bmatrix}$$

- Can only swap row $i$ with $n+i$
- There's $2^n$ of them
- All preserve Lagrangianity, so $\Pi \begin{bmatrix} I \\ X \end{bmatrix}$ Lagrangian $\Leftrightarrow$ $X$ Hermitian

# $2 \times 2$ case

$$U = \begin{bmatrix} \boxed{1} \\ \boxed{2} \\ \boxed{3} \\ \boxed{4} \end{bmatrix}$$

If $\begin{array}{c}\boxed{1}\\ \boxed{2}\end{array}$ well-conditioned $\Rightarrow U \cdot \operatorname{inv}\left(\begin{array}{c}\boxed{1}\\ \boxed{2}\end{array}\right) = \begin{bmatrix} 1 & 0 \\ 0 & 1 \\ * & * \\ * & * \end{bmatrix}$

If $\begin{array}{c}\boxed{3}\\ \boxed{4}\end{array}$ well-conditioned $\Rightarrow U \cdot \operatorname{inv}\left(\begin{array}{c}\boxed{3}\\ \boxed{4}\end{array}\right) = \begin{bmatrix} * & * \\ * & * \\ 1 & 0 \\ 0 & 1 \end{bmatrix}$

# $2 \times 2$ case

If $\begin{array}{|c|}\hline 1 \\ \hline 4 \\ \hline\end{array}$ well-conditioned $\Rightarrow U \cdot \operatorname{inv}\left(\begin{array}{|c|}\hline 1 \\ \hline 4 \\ \hline\end{array}\right) = \begin{bmatrix} 1 & 0 \\ * & * \\ * & * \\ 0 & 1 \end{bmatrix} = \Pi \begin{bmatrix} 1 & 0 \\ 0 & 1 \\ * & * \\ * & * \end{bmatrix}$

If $\begin{array}{|c|}\hline 3 \\ \hline 2 \\ \hline\end{array}$ well-conditioned $\Rightarrow U \cdot \operatorname{inv}\left(\begin{array}{|c|}\hline 3 \\ \hline 2 \\ \hline\end{array}\right) = \begin{bmatrix} * & * \\ 0 & 1 \\ 1 & 0 \\ * & * \end{bmatrix} = \Pi \begin{bmatrix} 1 & 0 \\ 0 & 1 \\ * & * \\ * & * \end{bmatrix}$

What if none of them works? E.g., $\begin{bmatrix} 1 & 0 \\ 0 & 0 \\ 0 & 1 \\ 0 & 0 \end{bmatrix}$? Not Lagrangian!

# How good are row swaps?

$\mathcal{U} = \operatorname{Im} U$ has a permuted graph basis with a prescribed $\Pi$ iff some subset of $n$ rows of $U$ is linearly independent

$$U = \Pi \begin{bmatrix} Y \\ Z \end{bmatrix} \sim \Pi \begin{bmatrix} I \\ ZY^{-1} \end{bmatrix} \stackrel{def}{=} \Pi \begin{bmatrix} I \\ X \end{bmatrix}$$

**Theorem**

For each Lagrangian $\mathcal{U}$ there's a $\Pi$ such that $Y$ is invertible...

Follows easily from a result on symplectic matrices [Dopico, Johnson '06]

**Theorem** [Mehrmann, P., preprint]

...moreover, there's one with $X$ entrywise small:

$$|(X)_{ij}| \leq \begin{cases} 1 & \text{if } i = j \\ \sqrt{2} & \text{otherwise} \end{cases}$$

# Geometrical interpretation

In geometrical terms: the $2^n$ maps

$$f_{\Pi} : X \text{ Hermitian and bounded} \mapsto \operatorname{Im} \Pi \begin{bmatrix} I \\ X \end{bmatrix}$$

are an atlas for the manifold of Lagrangian subspaces

For each subspace, we can find $\Pi$ giving "tame" structure-preserving basis

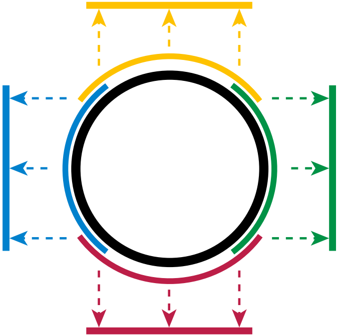


Image: ©Wikimedia

# Unstructured case

Similar, unstructured version known for a generic subspace:

**Theorem** [Knuth, ≈'84??]

For every $n$-dimensional subspace $\mathcal{U} \subseteq \mathbb{C}^{n+m}$, there are a permutation matrix $\Pi$ and an $X \in \mathbb{C}^{m \times n}$ with $|x_{i,j}| \leq 1$ for all $i, j$ such that

$$\mathcal{U} = \operatorname{Im} \Pi \begin{bmatrix} I_n \\ X \end{bmatrix}$$

Connected to rank revealing QR: existing work by Knuth, C.-T. Pan, Gu–Eisenstat, Goreinov *et al.*...

Key word: Plücker coordinates

# Sketch of the proof

$$U = \Pi \begin{bmatrix} Y_\Pi \\ Z_\Pi \end{bmatrix} \sim \Pi \begin{bmatrix} I \\ Z_\Pi Y_\Pi^{-1} \end{bmatrix} \stackrel{def}{=} \Pi \begin{bmatrix} I \\ X_\Pi \end{bmatrix}$$

Different $\Pi$ give different $Y_\Pi$; take $R$ so that $|\det Y_R|$ maximal

Cramer's rule on $X_R = Z_R Y_R^{-1}$ gives

$$|x_{ii}| = \left| \frac{\det Y_Q}{\det Y_R} \right| \leq 1$$

Can only swap $i$ with $n+i \Rightarrow$ this works only for diagonal elements $x_{ii}$

But similarly

$$\left| \det \begin{bmatrix} x_{i,i} & x_{i,j} \\ \overline{x_{i,j}} & x_{j,j} \end{bmatrix} \right| = \left| \frac{\det Y_Q}{\det Y_R} \right| \leq 1 \quad \Rightarrow \quad |x_{i,j}| \leq \sqrt{2}$$

# Computing a good $\Pi$

The proof can be turned into a "greedy" algorithm: given $U$ s.t. $\operatorname{Im} U = \mathcal{U}$,

1. choose an admissible $\Pi$
2. compute basis $\Pi \begin{bmatrix} I \\ X \end{bmatrix}$
3. if $|x_{i,i}| > 1$, update $\Pi$ with a row swap to enlarge $\det Y_\Pi$, goto 2
4. if $|x_{i,j}| > \sqrt{2}$, two row swaps, goto 2

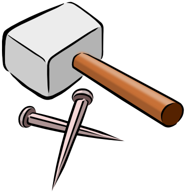
- Best to work with thresholds $S > 1$, $T > \sqrt{2}$
- Ends with a matrix $X$ with $|(X)_{ij}| \leq \begin{cases} S & \text{if } i = j \\ T & \text{otherwise} \end{cases}$
- Every update of $X$ is essentially a rank-1 update, $O(n^2)$
- Can be made very robust

Similar (less robust) algorithm for the unstructured case: [Goreinov et al., '08]

Applications?

# Applications

Several different problems can be "reshaped" in order to use this theorem

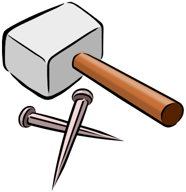


**Maslow's law**

"If all you have is a hammer, everything looks like a nail"

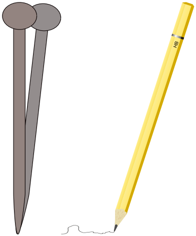
# Applications

Several different problems can be "reshaped" in order to use this theorem



**Maslow's law**

"If all you have is a hammer, everything looks like a nail"



First, a pencil and a nail are very similar objects. . .

# Pencils and subspaces

Eigenvalues and right invariant subspaces of a pencil: well defined up to

$$sE - A \sim s(ME) - MA \qquad \text{“right-handed equivalence”}$$

Or: they depend on the subspace $\operatorname{Im} \begin{bmatrix} E^* \\ A^* \end{bmatrix}$, not on the matrix $\begin{bmatrix} E^* \\ A^* \end{bmatrix}$

Results

# Structured pencils

## Symplectic pencils

$$E\mathcal{J}_{2n}E^* = A\mathcal{J}_{2n}A^* \quad \iff \quad \begin{bmatrix} E & A \end{bmatrix} \begin{bmatrix} \mathcal{J}_{2n} & 0 \\ 0 & -\mathcal{J}_{2n} \end{bmatrix} \begin{bmatrix} E^* \\ A^* \end{bmatrix} = 0$$

## Hamiltonian pencils

$$E\mathcal{J}_{2n}A^* = -A\mathcal{J}_{2n}E^* \quad \iff \quad \begin{bmatrix} E & A \end{bmatrix} \begin{bmatrix} 0 & \mathcal{J}_{2n} \\ \mathcal{J}_{2n} & 0 \end{bmatrix} \begin{bmatrix} E^* \\ A^* \end{bmatrix} = 0$$

By exchanging blocks, we can transform the two matrices in red into $\mathcal{J}_{4n}$

Up to some reordering, $\operatorname{Im} \left[\begin{smallmatrix} E^* \\ A^* \end{smallmatrix}\right]$ is Lagrangian!

Our theory can be used to give tame, structure-preserving representations. . .

# Bounded representations of symplectic pencils

**Theorem**

Every symplectic pencil is (right-handed-)equivalent to one in the form

$$s \begin{bmatrix} I_n & X_{21} \\ 0 & X_{22} \end{bmatrix} \Pi_1 - \begin{bmatrix} X_{11} & 0 \\ X_{21} & I_n \end{bmatrix} \Pi_2$$

with $\Pi_i$ symplectic swap matrices, $X = \begin{bmatrix} X_{11} & X_{12} \\ X_{21} & X_{22} \end{bmatrix}$ bounded Hermitian

(Without the parts in red, well known)

# Bounded representations of Hamiltonian pencils

**Theorem**

Every Hamiltonian pencil is (right-handed-)equivalent to one in the form

$$s \begin{bmatrix} E_1 & E_2 \end{bmatrix} - \begin{bmatrix} A_1 & A_2 \end{bmatrix}$$

with

$$\begin{bmatrix} E_1 & A_2 \end{bmatrix} = \begin{bmatrix} I & X_{11} \\ 0 & X_{21} \end{bmatrix} \Pi_1, \qquad \begin{bmatrix} -A_1 & E_2 \end{bmatrix} = \begin{bmatrix} X_{12} & 0 \\ X_{22} & I \end{bmatrix} \Pi_2,$$

and $X = \left[\begin{smallmatrix} X_{11} & X_{12} \\ X_{21} & X_{22} \end{smallmatrix}\right]$ bounded Hermitian

Idea: start from $sI - H$, you can swap vectors between the "outer" blocks and between the "inner" ones

# Deflating $R$ in general-form control problems

In control problems, originally an "extended" $(2n+m) \times (2n+m)$ pencil
Need deflation to get a Hamiltonian matrix/pencil

$$s \begin{bmatrix} 0 & I & 0 \\ -I & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} - \begin{bmatrix} 0 & A & B \\ A^* & Q & S \\ B^* & S^* & R \end{bmatrix} \implies s \begin{bmatrix} I & 0 & 0 \\ 0 & I & 0 \\ 0 & 0 & 0 \end{bmatrix} - \begin{bmatrix} \multicolumn{2}{c}{\boxed{\mathcal{H}}} & 0 \\ & & 0 \\ * & * & I \end{bmatrix}$$

When $R$ ill-conditioned or singular, trouble
Solution: allow identities to move!

$$s \begin{bmatrix} 0 & I & 0 \\ -I & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} - \begin{bmatrix} 0 & A & B \\ A^* & Q & S \\ B^* & S^* & R \end{bmatrix} \implies s \begin{bmatrix} \multicolumn{2}{c}{\boxed{\mathcal{H}_1}} & 0 \\ & & 0 \\ * & * & 0 \end{bmatrix} - \begin{bmatrix} \multicolumn{2}{c}{\boxed{\mathcal{H}_2}} & 0 \\ & & 0 \\ * & * & I \end{bmatrix}$$

$s\mathcal{H}_1 - \mathcal{H}_2$ Hamiltonian pencil (bounded as in the theorem)

# Doubling algorithms – I

Doubling algorithms compute invariant subspaces of Hamiltonian / symplectic pencils

## Key operation

$$\text{Given } A - sE\text{, find full-rank } \begin{bmatrix} C & S \end{bmatrix} \text{ such that } \begin{bmatrix} C & S \end{bmatrix} \begin{bmatrix} A \\ E \end{bmatrix} = 0$$

Two main strategies:

- QR-factorize $\left[\begin{smallmatrix} A \\ E \end{smallmatrix}\right]$ and construct $C, S$ from $Q$
- permute and invert a block to reduce to $\left[\begin{smallmatrix} A \\ E \end{smallmatrix}\right] = \left[\begin{smallmatrix} I \\ X \end{smallmatrix}\right]$; then use

$$\begin{bmatrix} -X & I \end{bmatrix} \begin{bmatrix} I \\ X \end{bmatrix} = 0$$

# Doubling algorithms – II

- QR-factorize $\begin{bmatrix} A \\ E \end{bmatrix}$: inverse-free matrix sign/disc method [Benner, '96] , [Benner, Byers '06] , [Bai et al., '97]
- enforce identity: structure-preserving doubling algorithm (SDA) [Anderson, '78] , [Chu et al., '04]

| | QR-based | SDA |
|---|---|---|
| $O(n^3)$? | Yes | Yes |
| Structure-preserving? | No way! | Yes |
| Stable? | Yes | No way! |

This looks familiar. . . again, it's hammer time!

# An attempt at a new doubling algorithm

## Doubling + permuted graph bases

1. Compute bounded permuted graph basis $\begin{bmatrix} E \\ A \end{bmatrix} = \widetilde{\Pi} \begin{bmatrix} I \\ \widetilde{X} \end{bmatrix}$
2. $\begin{bmatrix} C & S \end{bmatrix} = \begin{bmatrix} -\widetilde{X} & I \end{bmatrix} \widetilde{\Pi}^{-1}$
3. Use $C, S$ to perform a doubling step
4. Compute bounded permuted graph basis of $\begin{bmatrix} E^* \\ A^* \end{bmatrix} = \Pi \begin{bmatrix} I \\ X \end{bmatrix}$

6. Repeat until convergence

# An attempt at a new doubling algorithm

## Doubling + permuted graph bases

1. Compute bounded permuted graph basis $\left[\begin{smallmatrix} E \\ A \end{smallmatrix}\right] = \widetilde{\Pi} \left[\begin{smallmatrix} I \\ \widetilde{X} \end{smallmatrix}\right]$
2. $[\, c \; s \,] = [\, -\widetilde{X} \; I \,]\, \widetilde{\Pi}^{-1}$ (unstructured version)
3. Use $C, S$ to perform a doubling step
4. Compute bounded permuted graph basis of $\left[\begin{smallmatrix} E^* \\ A^* \end{smallmatrix}\right] = \Pi \left[\begin{smallmatrix} I \\ X \end{smallmatrix}\right]$
5. Enforce Lagrangianity: $X \leftarrow \frac{1}{2}(X + X^*)$
6. Repeat until convergence

Still not 100% satisfactory: $\left[\begin{smallmatrix} E \\ A \end{smallmatrix}\right]$ is not Lagrangian: switch to unstructured arithmetic and then project back

Looking for a compact version of 1–4 using only Hermitian arithmetic
Possible in the known special cases (SDA, Cyclic Reduction)

But still, great numerical results. . .

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

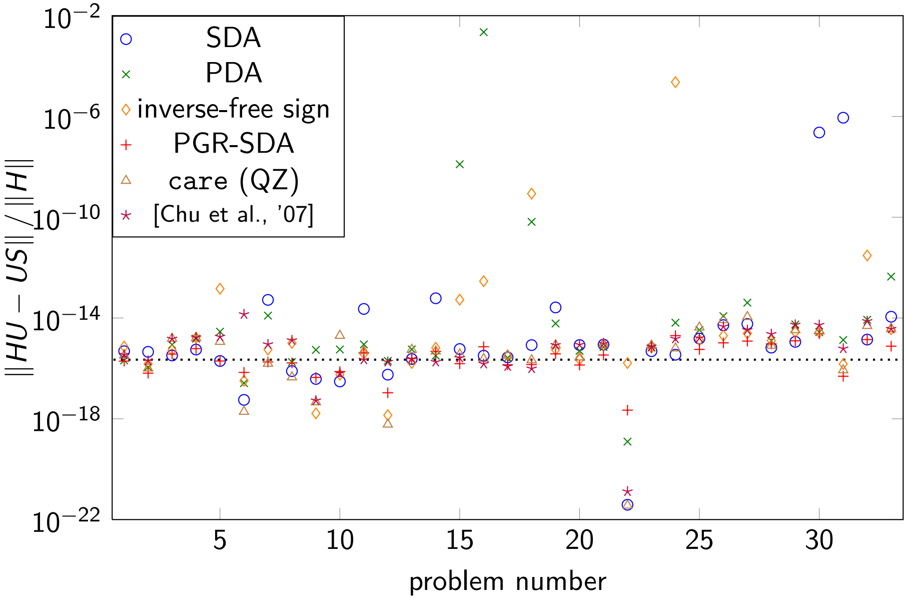
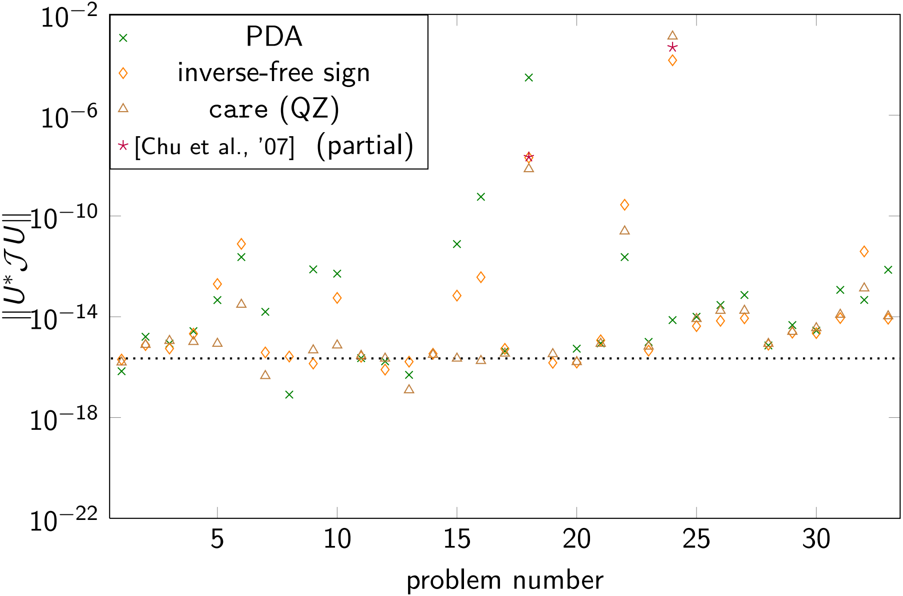


Figure: Lagrangianity residual for the 33 CAREX problems in [Chu et al., '07]



# Now, stability analysis...

## Stability results

- $\kappa(\Pi \left[\begin{smallmatrix} I \\ X \end{smallmatrix}\right]) \leq Cn$, with $\kappa(Z) = \sigma_{\max}(Z)/\sigma_{\min}(Z)$
- Given an initial basis $U$, can construct $\Pi \left[\begin{smallmatrix} I \\ ZY^{-1} \end{smallmatrix}\right]$ with $\kappa(Y) \leq Cn\kappa(U)$

Unfortunately, textbook backward stability analysis not well suited to doubling (or, more generally, matrix squaring):

$$\begin{array}{ccc}
\begin{bmatrix} 0 & 0 \\ 0 & 0 \end{bmatrix} & \xrightarrow{\text{squaring}} & \begin{bmatrix} 0 & 0 \\ 0 & 0 \end{bmatrix} \\
 & & \big\downarrow \text{perturb} \\
??? & \xrightarrow{\text{squaring}} & \begin{bmatrix} 0 & \varepsilon \\ 0 & 0 \end{bmatrix}
\end{array}$$

Note: 0 **not** a critical eigenvalue here

# How good is this in theory?

| | QR-based | SDA | Permuted SDA |
|---|---|---|---|
| $O(n^3)$? | Yes | Yes | |
| Structure-preserving? | No way! | Yes | |
| Stable? | Yes | No way! | |

# How good is this in theory?

| | QR-based | SDA | Permuted SDA |
|---|---|---|---|
| $O(n^3)$? | Yes | Yes | Kind of |
| Structure-preserving? | No way! | Yes | Kind of |
| Stable? | Yes | No way! | Kind of |

- $O(n^3)$: Need to bound total number of row swaps
  Well-studied in the unstructured case
  In practice $2n$ row swaps (overall) suffice on all experiments
- Structure-preserving: Would like "fully Hermitian" update formula
  We haven't nailed it down yet...
- Stable: can mimic [Bai *et al.*, '97] , but large worst-case constants

Turning those "kind of" into yes looks possible for the first time

# Conclusions

- Doubling now competitive with state-of-the-art algorithms for dense control problems
  (Matlab code soon to be released — contact me for info)
- Recipe to add stability to existing structure-preserving algorithms

Other possible "nail" applications:

- Large scale Riccati and ADI:
  $X$ (approx) low rank $\Rightarrow$ many determinants (approx) 0
- $\mathcal{H}_\infty$ control: Riccati with unbounded solutions show up
- "Butterfly" SR/SZ algorithms
- Are you working with symplectic matrices?
  Maybe your problem looks like a nail, too

# Conclusions

- Doubling now competitive with state-of-the-art algorithms for dense control p
  (Matlab
- Recipe t                                                                                    hms

Thanks for your attention! Questions?

Other possibl

- Large sc
  $X$ (approx) low rank $\Rightarrow$ many determinants (approx) 0
- $\mathcal{H}_\infty$ control: Riccati with unbounded solutions show up
- "Butterfly" SR/SZ algorithms
- Are you working with symplectic matrices?
  Maybe your problem looks like a nail, too

# How does it compare to [Mehrmann, Schröder, Watkins '09] ?

| | QR-based | SDA | Perm-SDA | MehSW |
|---|---|---|---|---|
| $O(n^3)$? | Yes | Yes | Kind of | Yes |
| Structure-preserving? | No way! | Yes | Kind of | Yes?* |
| Stable? | Yes | No way! | Kind of | Yes?* |
| BLAS3/Parallel/ Communication optimal? | Yes | Yes | Kind of | No way! |

- *MehSW (essentially: block Schur + Laub trick on every block) uses orthogonal bases and can have the same problems as Laub trick
- Schur-type algorithms not suited for large communication optimal linear algebra — for instance, [Demmel et al., '06] use doubling instead of QR for eigenvalues

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

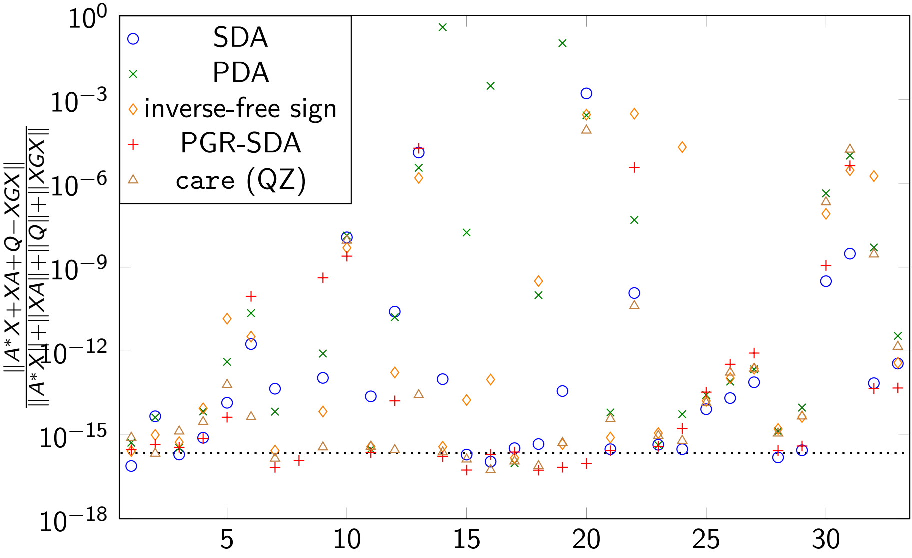


Figure: Unstructured pencil backward error for the problems in [Chu et al., '07]