Permuted graph bases for structured subspaces and pencils

Volker Mehrmann[1]   Federico Poloni[2]

[1]Technische U Berlin, Inst of Mathematics
[2]U Pisa, Italy, Dept of Computer Science

4th IMA Conference on Numerical Linear Algebra and Optimisation
September 2014

Subspaces, bases and graph bases

Definition

U, V tall thin matrices with full column rank.
$U \sim V$ if $U = VB$ for a square invertible $B \iff$ same column space.

Each V with $U \sim V$ can be used to work with the subspace $\operatorname{im} U$.

- If $U = QR$ (tall skinny QR), $U \sim Q$.
- If $U = \begin{bmatrix} B \\ N \end{bmatrix}$, with B square invertible, $U \sim \begin{bmatrix} I \\ NB^{-1} \end{bmatrix}$ graph basis.

  $B^{-1} \to$ danger: can be ill-conditioned.

Permuted graph bases

- If B is any square invertible submatrix of U, we can post-multiply by B^{-1} to enforce an identity in a subset of rows.

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

Ill-conditioning — how bad can it be?

Theorem [Knuth, '84 or earlier]

Each full-column-rank U has a permuted graph basis $P \left[\begin{smallmatrix} I \\ X \end{smallmatrix} \right]$ with $|x_{ij}| \leq 1$

How to compute them?

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

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

Related to rank-revealing factorizations, algebraic geometry but NP-hard!

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

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

Relax to $|x_{ij}| \leq \tau$ with $\tau > 1$ for better convergence.

Gains and losses

Condition number $\kappa(V) = \frac{\sigma_{\max}(V)}{\sigma_{\min}(V)}$ determines column space sensitivity.

Theorem

If $|x_{ij}| \leq \tau$, then $\kappa(P \left[\begin{smallmatrix} I \\ X \end{smallmatrix}\right]) \leq \sqrt{mn\tau^2 + 1}$

With respect to an orthogonal basis, we lose conditioning (but not too much!), but we gain an identity submatrix. What use is it?

Several applications in optimization:

- Approximate $\max(f)$ on a large grid, cross-tensor approximation.
  [Oseledets, Savostyanov, Tyrtishnikov et al, '10]
- Minimize function of a subspace (Grassmann manifold) $f(U)$.
  [Markovsky, Usevich '14]
- Precondition large-scale least-squares via "basis variables".
  [Arioli, Duff '14]

A structured version

Image of $U \in \mathbb{C}^{2n\times n}$ Lagrangian if $U^H J_{2n} U = 0$, with $J_{2n} = \begin{bmatrix} 0 & I_n \\ -I_n & 0 \end{bmatrix}$.

Graph matrix $U = \begin{bmatrix} I \\ X \end{bmatrix}$ Lagrangian $\iff$ X Hermitian.

Not true for $P \begin{bmatrix} I \\ X \end{bmatrix}$ though: we must change the concept of permutation.

Symplectic swaps

Vector transformations generated by J_2 on (x_k, x_{n+k}) for each k:

$$\left[\begin{array}{ccccc|ccccc} x_1 & \cdots & -x_{n+k} & \cdots & x_n & x_{n+1} & \cdots & x_k & \cdots & x_{2n} \end{array}\right].$$

Lagrangian permuted graph bases

Theorem [Mehrmann, P. '12]

If im U Lagrangian, then there exists Lagrangian permuted graph basis $U \sim S \begin{bmatrix} I \\ X \end{bmatrix}$ with S symplectic swap, $X = X^H$ and $|x_{ij}| \leq \sqrt{2}$.

Similar but not trivial, structure and allowed transformations must match.

Example

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

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

Results for pencils

Definitions: matrix pencil: degree-1 matrix polynomial $L(x) = L_1 x + L_0$. Assume here regular, i.e., $\det L(x) \not\equiv 0$.

Eigenvalue, eigenvector of a pencil: $L(\lambda)v = 0$. Unchanged if I premultiply:

Definition

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

Note that

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

So one can use results on subspaces to normalize pencils

Example

$$L(x) \sim \begin{bmatrix} 1 & * & 0 \\ 0 & * & 1 \\ 0 & * & 0 \end{bmatrix} x + \begin{bmatrix} * & * & 0 \\ * & * & 0 \\ * & * & 1 \end{bmatrix}, \quad |*| \leq 1.$$

Results for structured pencils

Symplectic pencils: $L(x) \in \mathbb{C}[x]^{2n\times 2n}$ such that $L_1 J_{2n} L_1^H = L_0 J_{2n} L_0^H$.

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

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

- 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: $L(x) \in \mathbb{C}[x]^{2n\times 2n}$ such that $L_1 J_{2n} L_0^H = -L_0 J_{2n} L_1^H$.

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

Linear-quadratic optimal control

Common control-theory problem: compute stable (eigenvalues with $\operatorname{Re}\lambda < 0$) invariant subspace of

$$\begin{bmatrix} 0 & I_n & 0 \\ -I_n & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} x - \begin{bmatrix} 0 & A & B \\ A^T & Q & S \\ B^T & S^T & R \end{bmatrix}$$

Traditional solution (recast in our language): first enforce identity

$$\begin{bmatrix} I_n & 0 & 0 \\ 0 & I_n & 0 \\ 0 & 0 & 0 \end{bmatrix} x - \begin{bmatrix} * & * & 0 \\ * & * & 0 \\ * & * & I_m \end{bmatrix};$$

Now it's block triangular; deflate and work on block-2×2 pencil in orange.

The orange pencil is Hamiltonian, better to preserve structure.

A different deflation

Why must the identity go there?

$$\begin{bmatrix} \boxed{\begin{matrix} * & * \\ * & * \end{matrix}} & \begin{matrix} 0 \\ 0 \end{matrix} \\ \begin{matrix} * & * \end{matrix} & 0 \end{bmatrix} x - \begin{bmatrix} \boxed{\begin{matrix} * & * \\ * & * \end{matrix}} & \begin{matrix} 0 \\ 0 \end{matrix} \\ \begin{matrix} * & * \end{matrix} & I_m \end{bmatrix}$$

Put columns of I in half of the green and blue columns. The deflated top block-2×2 pencil is Hamiltonian (in the format of our previous slide).

Example

$$\begin{bmatrix} 0 & 1 & 0 \\ -1 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} x - \begin{bmatrix} 0 & 0 & 1 \\ 0 & 0 & 0 \\ 1 & 0 & \varepsilon \end{bmatrix} \sim \begin{bmatrix} \boxed{\begin{matrix} 1 & 0 \\ 0 & \varepsilon \end{matrix}} & \begin{matrix} 0 \\ 0 \end{matrix} \\ \begin{matrix} 0 & -1 \end{matrix} & 0 \end{bmatrix} x + \begin{bmatrix} \boxed{\begin{matrix} 0 & 0 \\ 1 & 0 \end{matrix}} & \begin{matrix} 0 \\ 0 \end{matrix} \\ \begin{matrix} 0 & 0 \end{matrix} & 1 \end{bmatrix}$$

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

Invariant subspaces of Hamiltonians

Problem: compute stable ($\operatorname{Re}\lambda < 0$) inv. subspace of a Hamiltonian pencil ($\iff$ solve a Riccati equation, if subspace U in graph basis)

Algorithm: a "pencil variant" of the matrix sign function iteration $A \mapsto \frac{1}{2}(A + A^{-1})$

Two things needed at each step:

- Compute left kernel of $\begin{bmatrix} L_1 \\ L_0 \end{bmatrix}$: use permuted graph bases:

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

- Normalize Hamiltonian pencil $L_1 x + L_0$ keeping structure: use Lagrangian permuted graph bases (i.e., of $\begin{bmatrix} L_1 & L_0 \end{bmatrix}^H$).

Inverse-free sign method (with permuted graph bases)

Algorithm [Mehrmann, P. '12 and '13]

Input: $L_1 x + L_0$ Hamiltonian;

1. compute $\begin{bmatrix} -M_0 & M_1 \end{bmatrix}$ left kernel of $\begin{bmatrix} L_1 \\ L_0 \end{bmatrix}$;
2. replace $L(x)$ with $M_0 L_1 x + \frac{1}{2}(M_1 L_1 + M_0 L_0)$;
3. compute Lagrangian permuted representation of $L(x)$;
4. repeat 1–3 until convergence;
5. find kernel of $L_1 + L_0$;

How well does it go in practice? On a known set of benchmark problems (CAREX, [Benner et al, '95, Chu et al '07]), first algorithm to get perfect results on both:

- subspace residual down to machine precision;
- Lagrangian Structure preserved (exactly or up to machine precision).

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

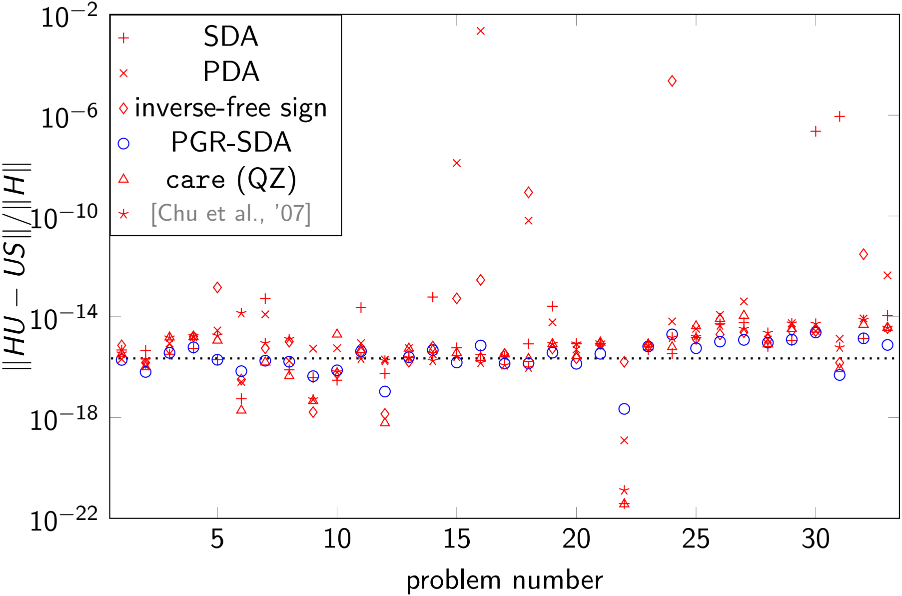
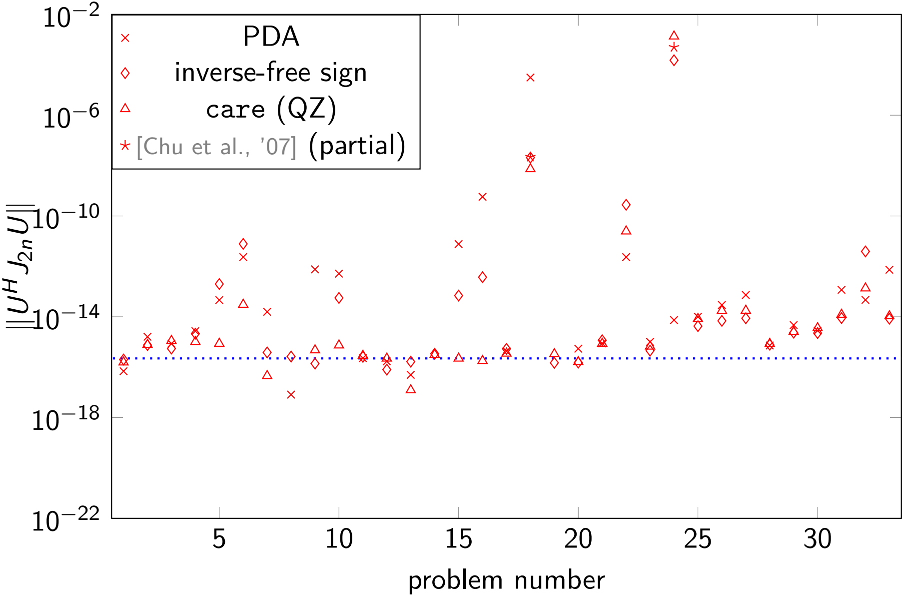


Figure : Lagrangianity residual for the 33 CAREX problems in [Chu et al., '07]



Large scale AREs

This was for small-case dense problems; what about large, sparse control?

- Often, the invariant subspace can be represented cheaply as $U \sim \begin{bmatrix} I \\ ZZ^T \end{bmatrix}$, with Z tall skinny.
- Orthogonal basis not pursued, difficult to use this low-rank property.

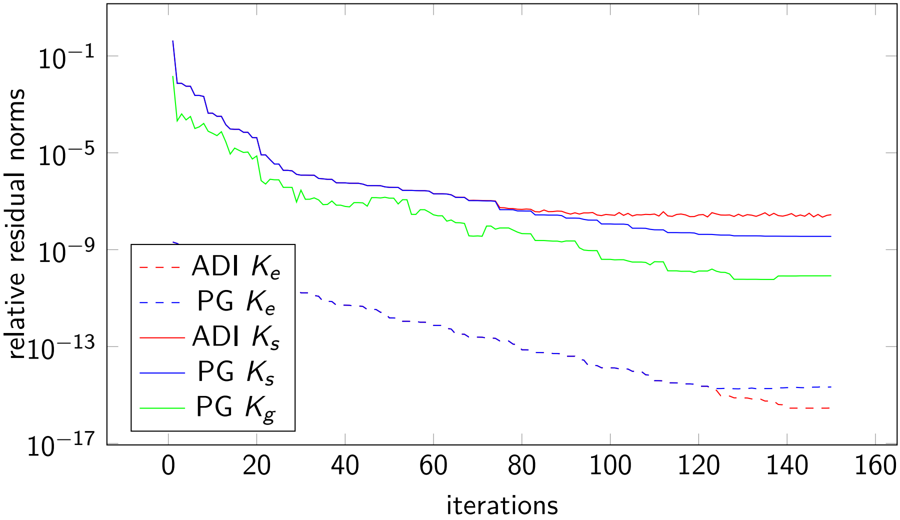
A first attempt to use these ideas:

1. Run a standard solution algorithm (ADI) keeping not Z but $\begin{bmatrix} B \\ N \end{bmatrix}$ (up to $\sim$) such that $Z = NB^{-1}$;
2. using the kernel trick $\begin{bmatrix} -X & I \end{bmatrix} P^{-1} P \begin{bmatrix} I \\ X \end{bmatrix} = 0$, build stable low-rank representation $U \sim \begin{bmatrix} I - V_1 V_2^T \\ V_3 V_4^T \end{bmatrix}$, all the V_i tall skinny.

How does it work? Beneficial in some ill-conditioned cases, large $\|Z\|$.

An experiment

Figure : Comparison of ADI and PG-ADI, random matrix and RHS



Large scale AREs

Theorem [Mehrmann, P. preprint]

Given orthogonal $\left[\begin{smallmatrix} B \\ N \end{smallmatrix}\right]$ such that $Z = NB^{-1} \in \mathbb{C}^{n \times m}$, we can build (quickly and stably) tall skinny V_i such that and

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

Conclusions

Small dense case

- Works great!

Large-scale case still preliminary work; interesting messages:

- We can use permuted graph bases also in sparse problems.
- The kernel trick $\begin{bmatrix} -X & I \end{bmatrix} P^{-1} P \begin{bmatrix} I \\ X \end{bmatrix} = 0$ seems even more useful in the tall skinny case.
- Another reflection: for each Hamiltonian H, there is S such that for $S^{-1}HS$ the invariant subspace problem "in Riccati form" $U = \begin{bmatrix} I \\ X \end{bmatrix}$ is well-conditioned.
  How to exploit this? Can we run permuted graph Newton?

And, finally:

- Bases with identities are underrated. They work well if you keep flexible on the position of the I submatrix. Try them!

Conclusions

Small dense case

- Works great!

Large-scale case still preliminary work; interesting messages:

- We can use permuted graph bases also in sparse problems.
- The kernel trick $\begin{bmatrix} -X & I \end{bmatrix} P^{-1} P \begin{bmatrix} I \\ X \end{bmatrix} = 0$ seems even more useful in the tall skinny case.
- Another reflection: for each Hamiltonian H, there is S such that for $S^{-1}HS$ the invariant subspace problem "in Riccati form" $U = \begin{bmatrix} I \\ X \end{bmatrix}$ is well-conditioned.
  How to exploit this? Can we run permuted graph Newton?

And, finally:

- Bases with identities are underrated. They work well if you keep flexible on the position of the I submatrix. Try them!

Thanks for your attention!