Permuted Graph Bases for Solving Large and Sparse Matrix Equations

Volker Mehrmann (TU Berlin, mehrmann@math.tu-berlin.de) and Federico Poloni (U Pisa, fpoloni@di.unipi.it)

Inverse-free representation

A matrix $\mathbf{M} = \mathbf{A}\mathbf{E}^{-1}$ is uniquely determined by the subspace $\mathbf{span} \begin{bmatrix}\mathbf{E}\\ \mathbf{A}\end{bmatrix}$ (*not* $\begin{bmatrix}\mathbf{E}\\ \mathbf{A}\end{bmatrix}$ itself! May transform $\mathbf{E} \to \mathbf{EK}$, $\mathbf{A} \to \mathbf{AK}$).

Plan: store and work on pair $(\mathbf{E}, \mathbf{A})$ rather than $\mathbf{M}$.
Most linear algebra operations ("primitives") can be done without inversions working on $(\mathbf{E}, \mathbf{A})$ [Benner, Byers 2006]: e.g., addition:

$$\mathbf{A_1E_1^{-1}} + \mathbf{A_2E_2^{-1}} = (\mathbf{A_1P} + \mathbf{A_2Q})(\mathbf{E_1Q})^{-1},$$

where $\mathbf{P}, \mathbf{Q}$ chosen so that $\begin{bmatrix}\mathbf{P}\\ \mathbf{Q}\end{bmatrix} = \ker \begin{bmatrix}-\mathbf{E_2} & \mathbf{E_1}\end{bmatrix}$

Advantage: more accurate when $\mathbf{E_1}$ and/or $\mathbf{E_2}$ almost singular

Permuted graph bases

Pair $(\mathbf{E}, \mathbf{A}) \leftrightarrow$ basis of the subspace $\leftrightarrow \mathbf{K}$ above. How to choose it?

- Orthogonal bases: everyone likes them!
- Permuted graph bases: another possible choice. What are they?

Theorem [Knuth, 1986]

Every $\mathbf{m}$-dim subspace of $\mathbb{R}^{\mathbf{n}}$ is spanned by a matrix $\tilde{\mathbf{U}} \in \mathbb{R}^{\mathbf{n}\times\mathbf{m}}$ that has $\mathbf{I_m}$ as a submatrix and all other entries ≤ 1.

We can bound the condition number $\kappa_2(\tilde{\mathbf{U}})$ for this basis, numerically "almost as good" as orthogonal bases.

Example

$$\mathbf{U} = \begin{bmatrix}1 & 2\\ 3 & 4\\ 5 & 6\\ 7 & 8\end{bmatrix}, \text{ PG basis } \tilde{\mathbf{U}} = \begin{bmatrix}0 & 1\\ 0.333 & 0.667\\ 0.667 & 0.333\\ 1 & 0\end{bmatrix} \qquad \begin{aligned}\mathbf{span}\,\mathbf{U} &= \mathbf{span}\,\tilde{\mathbf{U}}\\ \kappa_2(\mathbf{U}) &= 22.76\\ \kappa_2(\tilde{\mathbf{U}}) &= 1.34\end{aligned}$$

What's the advantage over orthogonal?

- Sparser, stable representation of the subspace
- Some primitives much more efficient, e.g. $\mathbf{M} \to \mathbf{M}^\mathbf{T}$
- Easier to preserve some structures (Lagrangian/Hamiltonian/symplectic).

Lyapunov equations and ADI

We have already used permuted graph bases machinery for dense matrix equations [M., P. 2012]. Time to move to large sparse!

Lyapunov equation

$$\mathbf{F^TX} + \mathbf{XF} + \mathbf{GG^T} = \mathbf{0} \qquad \text{(LE)}$$

$\mathbf{F} \in \mathbb{R}^{\mathbf{n}\times\mathbf{n}}$ large sparse, $\mathbf{G} \in \mathbb{R}^{\mathbf{n}\times\mathbf{m}}$ tall skinny. Looking for solution $\mathbf{X} = \mathbf{X^T} \geqslant \mathbf{0}$; often in applications it's (approximately) low-rank

Main workhorse: Low Rank – ADI algorithm [Benner, Li, Penzl 2008]

LR-ADI algorithm

- Store at each step $\mathbf{k}$ a low-rank factor $\mathbf{Z_k}$ of current solution guess $\mathbf{X_k}$; start from $\mathbf{Z_0} = [\,]$
- At each step, generate new component $\mathbf{W_{k+1}}$ using a rational-Krylov-like iteration and set $\tilde{\mathbf{Z}}_{\mathbf{k+1}} = \begin{bmatrix}\mathbf{Z_k} & \mathbf{W_{k+1}}\end{bmatrix}$
- Compress $\tilde{\mathbf{Z}}_{\mathbf{k+1}}$ to a thinner $\mathbf{Z_{k+1}}$ such that $\tilde{\mathbf{Z}}_{\mathbf{k+1}}\tilde{\mathbf{Z}}_{\mathbf{k+1}}^\mathbf{T} \approx \mathbf{Z_{k+1}Z_{k+1}^T}$

Typical setting: $\mathbf{n} \gg \mathbf{m}$; generating $\mathbf{W_k}$ requires solving a sparse system, expensive; dealing with tall skinny $\mathbf{Z_k}$ matrices is cheap.

Combining the ideas

Idea: store all the iterates $\mathbf{Z_k}$, $\mathbf{W_k}$ with permuted graph bases.
Need new inverse-free primitives: horizontal stacking (not obvious in this setting!) and column compression
At the end, using another new primitive we return $\mathbf{A}, \mathbf{E} \in \mathbb{R}^{\mathbf{n}\times\mathbf{n}}$ (stored efficiently) such that $\mathbf{X} = \mathbf{AE^{-1}}$.

Stacking and column compression

Input: $\mathbf{A_1}, \mathbf{E_1}, \mathbf{A_2}, \mathbf{E_2}$ such that $\mathbf{Z_k} = \mathbf{A_1E_1^{-1}}$, $\mathbf{W_{k+1}} = \mathbf{A_2E_2^{-1}}$

1. Set $\mathbf{E_3} = \begin{bmatrix}\mathbf{E_1} & \mathbf{0}\\ \mathbf{0} & \mathbf{E_2}\end{bmatrix}$, $\mathbf{A_3} = \begin{bmatrix}\mathbf{A_1} & \mathbf{A_2}\end{bmatrix}$: now $\tilde{\mathbf{Z}}_{\mathbf{k+1}} = \begin{bmatrix}\mathbf{Z_k} & \mathbf{W_{k+1}}\end{bmatrix} = \mathbf{A_3E_3^{-1}}$
2. CS decomposition $\mathbf{E_3} = \mathbf{USK}$, $\mathbf{A_3} = \mathbf{VCK}$, $\mathbf{S}, \mathbf{C}$ diagonal $\mathbf{s_{ii}^2} + \mathbf{c_{ii}^2} = \mathbf{1}$ $\mathbf{U}, \mathbf{V}$ orthogonal. It's like an inverse-free SVD
3. $\mathbf{K}$ and $\mathbf{U}$ can be dropped, they simplify
4. columns $\mathbf{i}$ with small ratio $\mathbf{c_{ii}}/\mathbf{s_{ii}}$ negligible in $\tilde{\mathbf{Z}}_{\mathbf{k+1}}\tilde{\mathbf{Z}}_{\mathbf{k+1}}^\mathbf{T}$, drop them

Error measures

According to applications, either an accurate $\mathbf{X}$ or an accurate subspace $\mathcal{U} = \mathbf{span}\begin{bmatrix}\mathbf{I_n}\\ \mathbf{X}\end{bmatrix}$ (or other error measures) are important.

ADI delivers a good $\mathbf{X}$, but when $\|\mathbf{X}\|$ is large, $\begin{bmatrix}\mathbf{I_n}\\ \mathbf{X}\end{bmatrix}$ is an ill-conditioned basis for its range.

Inverse-free arithmetic delivers $(\mathbf{A}, \mathbf{E})$ such that $\mathbf{X} = \mathbf{AE^{-1}}$, and $\begin{bmatrix}\mathbf{E}\\ \mathbf{A}\end{bmatrix}$ is a well-conditioned basis for $\mathcal{U}$.

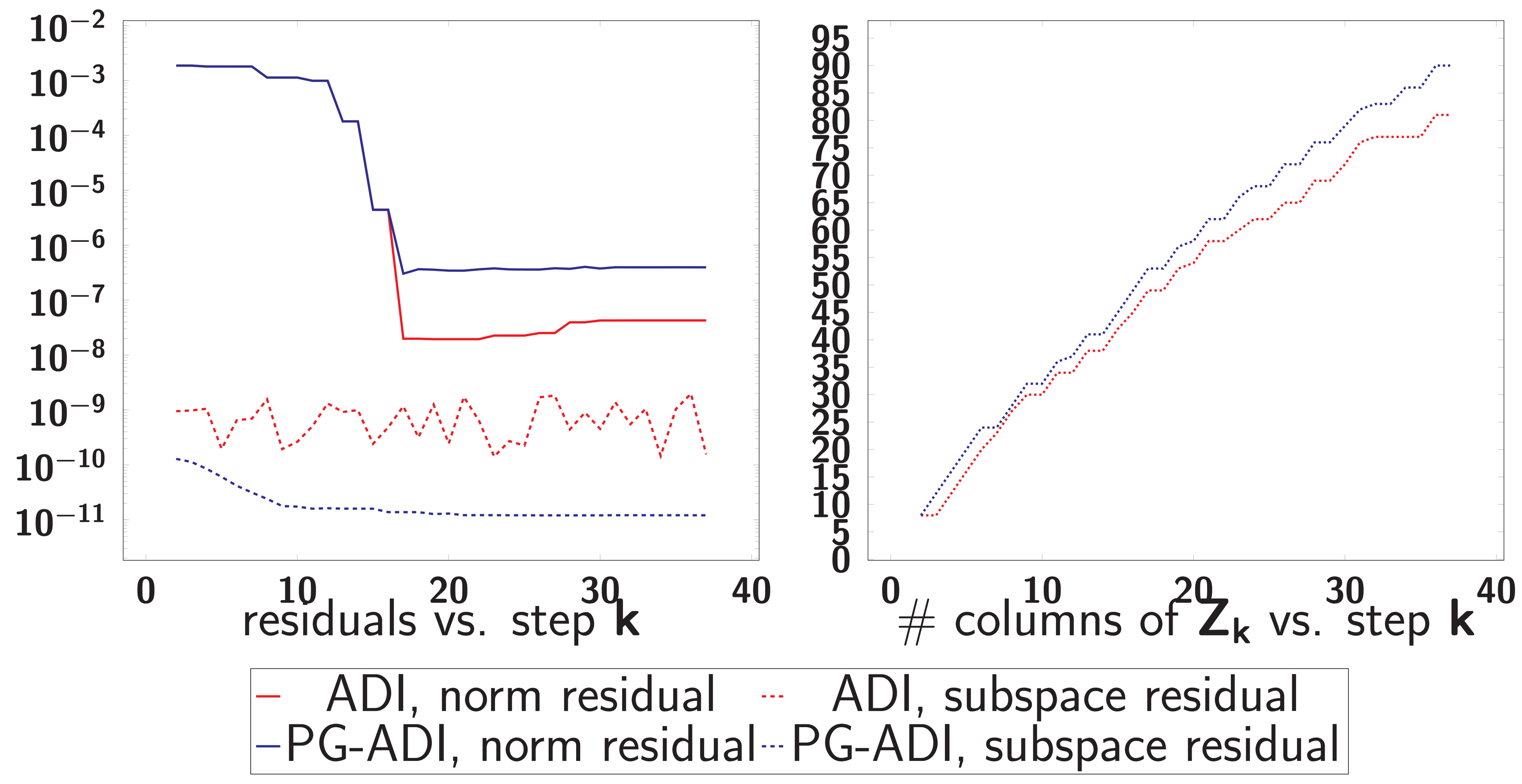
Example

$$\mathbf{U} = \begin{bmatrix}\mathbf{E}\\ \mathbf{A}\end{bmatrix} = \begin{bmatrix}\delta_1 & 0\\ 0 & 1\\ 1 & 0\\ 0 & \delta_2\end{bmatrix}, \quad \mathbf{X} = \begin{bmatrix}\delta_1^{-1} & 0\\ 0 & \delta_2\end{bmatrix}, \quad \delta_1, \delta_2 \ll 1$$

$\mathbf{X}$ well-approximated by low-rank $\tilde{\mathbf{X}} = \begin{bmatrix}\delta_1^{-1} & 0\\ 0 & 0\end{bmatrix}$, very sensitive to perturbations of δ_1 but info on $\mathbf{u_2}$ and δ_2 completely lost.
$\mathbf{U}$ better representation for the subspace (but worse for $\mathbf{X}$!), equally sensitive to perturbations to δ_1 and δ_2.

Example

Toy example random, symmetric, ill-conditioned sparse $\mathbf{F} \in \mathbb{R}^{400\times400}$, "hard" right-hand side ($\mathbf{G}$ = smallest eigs of $\mathbf{F}$, perturbed)



Conclusions

- Better subspaces out of ADI when using permuted graph bases and inverse-free arithmetic.
- Trade-off between rank of computed $\mathbf{X}$ and subspace accuracy.
- Permuted graph/inverse-free machinery can find some use in many algorithms. Trouble with instabilities? Try it!
- Permuted graph basis Matlab code available on http://bitbucket.org/fph/pgdoubling.
- Preprint coming soon! Keep an eye on http://www.di.unipi.it/~fpoloni/publications/publications.php.