Short Recycling of Krylov Subspaces

Talk for NA group, TU Delft

Martin Neuenhofen

24th November 2015

Table of Contents

Focus

We consider

Let $\mathbf{A} \in \mathbb{C}^{N \times N}$ regular, $\mathbf{b} \in \mathbb{C}^N$. We aim to find $\mathbf{x} \in \mathbb{C}^N$ such that

$$\mathbf{A} \cdot \mathbf{x} + \mathbf{r} = \mathbf{b}$$

and its residual $\mathbf{r}$ is small.

We do not look at

1. Preconditioners; e.g. left, right, spd, flexible
2. linear subsolvers; e.g. projectors, deflators
3. roundoff errors

Notation

We use

1. $\#\text{MV}$ = number of matrix-vector-products
2. $\mathcal{U}$ as ansatz space with elements $\mathbf{u}$
3. $\mathcal{C} = \mathbf{A} \cdot \mathcal{U}$ as image space with elements $\mathbf{c} = \mathbf{A} \cdot \mathbf{u}$
4. $\mathcal{P}$ as test space, $\dim(\mathcal{P}) = \#\text{RDs}$ (number of reduced dimensions)

We use these operators:

$$\Phi(\mathcal{U}, \mathcal{P}) = \mathcal{U} \cdot (\mathcal{P}^H \cdot \mathcal{C})^\dagger \cdot \mathcal{P}^H$$
$$\Psi(\mathcal{U}, \mathcal{P}) = \mathbf{I} - \mathbf{A} \cdot \Phi(\mathcal{U}, \mathcal{P})$$

Basic: One System

$$\mathbf{A} \cdot \mathbf{x} = \mathbf{b}$$

Desire: full GMRES ← Krylov subspace = hold all information

Compute $\mathcal{U} = \mathcal{K}_n(\mathbf{A}; \mathbf{b})$ and find Residual-optimal $\mathbf{x} = \Phi(\mathcal{U}, \mathcal{C}) \cdot \mathbf{b}$.
$\Rightarrow$ eliminate one residual direction per MV (#RDs = #MVs)

Model problem

Solve Poisson problem:

$$\left\{ \begin{array}{rl} -\Delta u = f & \text{in } \Omega \\ u = 0 & \text{on } \partial\Omega \end{array} \right\}$$

Numerical treatment

Finite differences:

$$\sum_{\tilde{p} \in \mathcal{B}(p)} \frac{u_p - u_{\tilde{p}}}{\Delta x^2} = f_p$$

Advanced: Sequence of rhs with fixed matrix

$$\mathbf{A} \cdot \mathbf{x}^{(\iota)} = \mathbf{b}^{(\iota)}, \quad \iota = 1, \dots, n_{\text{Eqns}}$$

Desire: full GCR ← generalization of Krylov subspace = hold all information

Compute $\mathcal{U} := \mathcal{U} + \mathcal{K}_n\big(\mathbf{A}; \Psi(\mathcal{U}, \mathcal{C}) \cdot \mathbf{b}^{(\iota)}\big)$ and then find Residual-optimal $\mathbf{x} = \Phi(\mathcal{U}, \mathcal{C}) \cdot \mathbf{b}^{(\iota)}$ in it.
$\Rightarrow$ eliminate one residual direction per MV (#RDs = #MVs)

Model problem

Solve Fourier problem:

$$\left\{ \begin{array}{rl} \partial_t u - \Delta u = f & \text{in } \Omega \\ u(x,0) = 0 & \text{in } \Omega \\ u(x,t) = 0 & \text{on } \partial\Omega \end{array} \right\}$$

Numerical treatment

1. spatial finite differences
2. temporal implicit Euler

Voodoo: Sequence with 'slowly' changing matrix

$$\mathbf{A}^{(\iota)} \cdot \mathbf{x}^{(\iota)} = \mathbf{b}^{(\iota)}, \quad \iota = 1, \ldots, n_{\text{Eqns}}$$

Desire: no idea

Best hope: eliminate one residual direction per MV (#RDs = #MVs)

Model problem

Solve generalized Poisson problem:

$$\left\{ \begin{array}{rl} -\nabla \cdot \big(a(u) \cdot \nabla u\big) = f & \text{in } \Omega \\ u = 0 & \text{on } \partial\Omega \end{array} \right\}$$

Numerical treatment

Finite differences:

$$\sum_{\tilde{p} \in \mathcal{B}(p)} \frac{u_p - u_{\tilde{p}}}{\Delta x^2} \cdot \frac{a_p + a_{\tilde{p}}}{2} = f_p$$

GCR (k, m)

Algorithm 1: RGCRO

Data: $\mathbf{A}, \mathbf{r}, \mathbf{x}, \mathrm{tol}, \ \mathbf{U}, \mathbf{C}$

Result: $\mathbf{x}, \mathbf{U}, \mathbf{C}$

$\mathbf{x} := \mathbf{x} + \Phi(\mathcal{U}, \mathcal{C}) \cdot \mathbf{r}, \ \mathbf{r} := \Psi(\mathcal{U}, \mathcal{C}) \cdot \mathbf{r}$

while $\|\mathbf{r}\| > \mathrm{tol}$ **do**

- $\mathbf{u} := \mathbf{r}, \ \ \mathbf{c} := \mathbf{A} \cdot \mathbf{u}$
- $\mathbf{c} := \mathbf{c} - \mathbf{C} \cdot \gamma \perp \mathcal{C}, \ \ \mathbf{u} := \mathbf{u} - \mathbf{U} \cdot \gamma$
- $\mathbf{C} := [\mathbf{C}, \mathbf{c}], \ \ \mathbf{U} := [\mathbf{U}, \mathbf{u}]$
- $\mathbf{r} := \mathbf{r} - \omega \cdot \mathbf{c} \perp \mathcal{C}, \ \ \mathbf{x} := \mathbf{x} + \omega \cdot \mathbf{u}$
- **if** *size*($\mathbf{U}$,2)$>$ m **then**
  - Reduce $\mathbf{U}, \mathbf{C}$ to $\mathbb{C}^{N \times k}$

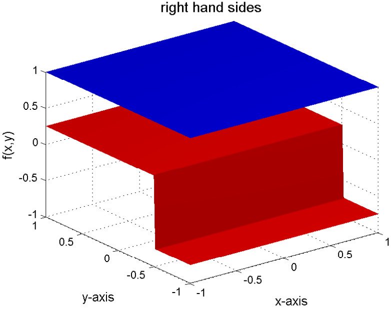
Recycling can be usefull...

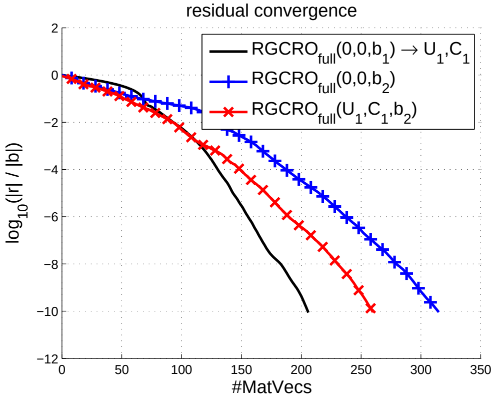
$$\mathbf{A} = \texttt{gallery('poisson',100)}$$

$$\mathbf{b}^{(1)} = \mathbf{1}$$

$$\mathbf{b}^{(2)} = 0.5 \cdot (\text{sign}(y + 0.5) - 0.5 \cdot \mathbf{1})$$

$$\mathbf{b}^{(1)} \perp \mathbf{b}^{(2)}$$





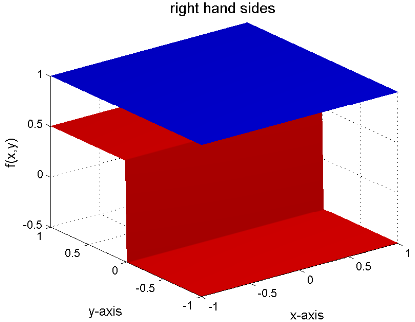
...but the problem must allow it!

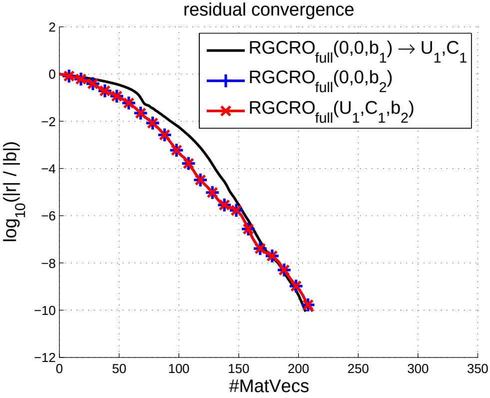
$$\mathbf{A} = \texttt{gallery('poisson',100)}$$

$$\mathbf{b}^{(1)} = \mathbf{1}$$

$$\mathbf{b}^{(2)} = 0.5 \cdot \mathrm{sign}(y)$$

$$\mathcal{K}(\mathbf{A}; \mathbf{b}^{(1)}) \perp \mathcal{K}(\mathbf{A}; \mathbf{b}^{(2)}) \rightarrow \text{negative test case}$$





A practical example:

Solve with impl. Euler, $\Delta x = 1/101$, $\Delta t = 0.1$:

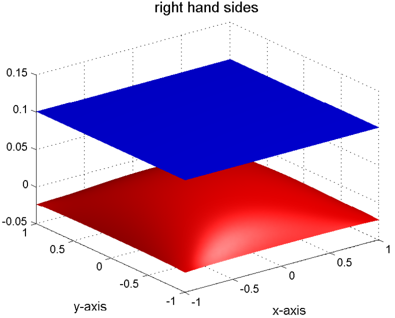
$$\left\{ \begin{array}{rl} \partial_t u - \Delta u = f & \text{in } \Omega \\ u(x,0) = 0 & \text{in } \Omega \\ u(x,t) = 0 & \text{auf } \partial\Omega \end{array} \right\}$$

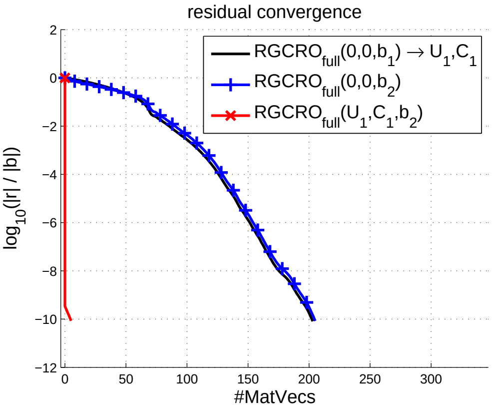
$$\mathbf{B} = \texttt{gallery('poisson',100)}$$

$$\mathbf{A} = \mathbf{I} + 0.1 \cdot (101)^2 \cdot \mathbf{B}$$

$$\mathbf{b}^{(1)} = \mathbf{1}$$

$$\mathbf{b}^{(2)} = \mathbf{A}^{-1} \cdot \mathbf{b}^{(1)} - \xi \cdot \mathbf{b}^{(1)} \perp \mathbf{b}^{(1)} \quad // \textit{ update}$$





With Recycling: Five MVs for second solve!

Summary on Recycling

Idea of reusing all information is natural.

1. start with $\mathcal{U} = \emptyset$
2. update $\mathcal{U} := \mathcal{U} + \{\mathbf{r}_{\text{current}}\}$

$\rightarrow$ can be interpreted as generalization of $\mathcal{K}$ for sequence of multiple rhs

Advantage

1. no loss of already computed information
   $\rightarrow$ optimality in #MVs

Drawback

1. additional orthogonalizations
2. additional storage

Not using a Recycling method for a sequence is comparable to not using a Krylov method for a single system.

Scope

We want

a full recycling method, but with ...

1. short recurrences, small storage
2. nearly optimal residual
3. $\#\text{MVs}_1 \approx \#\text{RDs}_1 \approx \#\text{RDs}_2 \gg \#\text{MV}_2$
4. no transpose

I will present

short-term recurrence methods recycling $k \cdot J$-dimensional $\mathcal{U}$ by

1. storage of only k columns of size N
2. additional computational cost of
    2.1. $2 \cdot J$ MVs with $\mathbf{A}$
    2.2. $2 \cdot J$ MVs with a dense $N \times k$-Matrix

Structure

In the following I present these methods

1. SRIDR: first prototype
2. SRMR: fundamental theory
3. (SRBiCG: non-hermitian generalization)
4. Outlook: SRse-ML(k)BiCG-IDR(s)

For each method I show

1. Theory
2. Building blocks
3. Performance

SRIDR - Theory : *The* Short Recycling idea

The SRIDR method...

- has only little practial use
- but elegant theory

Theoretical use:

- incorporates extension theory
- offers modification strategies

conventional IDR(2)

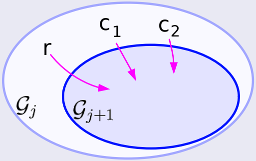


Figure 1: Each dimension reduction costs 1 MV.
$\rightarrow$ #RDs = #MVs $\cdot$ 2/3

modified IDR(2)

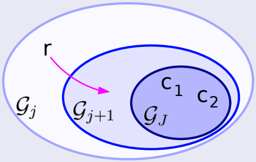


Figure 2: Skip auxiliary steps if $\mathbf{c}_i$ are already of higher level.
$\rightarrow$ #RDs = #MVs $\cdot$ 2

SRIDR - Method

Algorithm 2: SRIDR

Data: $\mathbf{A}, \mathbf{r}, \mathbf{x}, J, \ \mathbf{U}, \mathbf{C}, \mathbf{P}, \omega, J^\star$

Result: $\mathbf{x}, \mathbf{U}, \mathbf{C}, \mathbf{P}, \omega, J$

for $j = 1, ..., J-1$ **do**

  $\mathbf{x} := \mathbf{x} + \Phi(\mathcal{U},\mathcal{P}) \cdot \mathbf{r}, \ \ \mathbf{r} := \Psi(\mathcal{U},\mathcal{P}) \cdot \mathbf{r} \quad //\mathbf{r} \in \mathcal{G}_{j-1} \cap \mathcal{S}$

  if $j > J^\star$ **then**

    Choose ω_j

  $\mathbf{x} := \mathbf{x} + \omega_j \cdot \mathbf{r}, \ \ \mathbf{r} := (\mathbf{I} - \omega_j \cdot \mathbf{A}) \cdot \mathbf{r} \quad //\mathbf{r} \in \mathcal{G}_j$

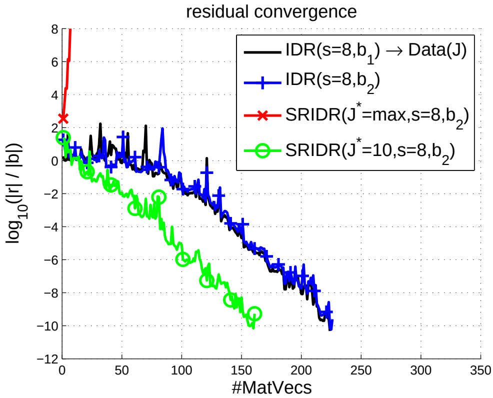
  if $j > J^\star$ **then**

    for $i := 1, ..., s$ **do**

      $\mathbf{u}_i := \big(\Phi(\mathcal{U},\mathcal{P}) + \omega_j \cdot \Psi(\mathcal{U},\mathcal{P})\big) \cdot \mathbf{r}$

      $\mathbf{c}_i := \mathbf{A} \cdot \mathbf{u}_k \quad //\mathbf{c}_i \in \mathcal{G}_j$

SRIDR - Performance



Explanation

red: $(\mathbf{U}, \mathbf{C}, \mathbf{P}, \omega, J^\star)$ obtained from last IDR-cycle of first system (black curve)

green: $(\mathbf{U}, \mathbf{C}, \mathbf{P}, \omega, J^\star)$ obtained earlier after 10^{th} IDR-cycle of first system (black curve)

→ still improving, but far from optimal

SRMR - Theory

After SRIDR I developed simple building blocks: Short Representations.

Krylov Recurrence

Hessenberg form: $\mathbf{A} \cdot \mathbf{V} = \overline{\mathbf{V}} \cdot \overline{\mathbf{H}}$
Store only: $\tilde{\mathbf{V}} = \mathbf{V}(:, 1 : J : m) \in \mathbb{C}^{N \times k}$ and $\overline{\mathbf{H}} \in \mathbb{C}^{(m+1) \times m}$, $k \cdot J = m$.

Theorem 1 (Short Representation)

There exist permutation $\mathbf{\Pi} \in \mathbb{C}^{m \times m}$ *depending on* k, J, *and triangular* $\mathbf{K} \in \mathbb{C}^{m \times m}$ *depending on* $k, J, \mathbf{H}$, *such that*

$$\overline{\mathbf{V}} \cdot \overline{\mathbf{H}} \cdot \mathbf{K} = [\tilde{\mathbf{V}}, \mathbf{A} \cdot \tilde{\mathbf{V}}, ..., \mathbf{A}^{J-1} \cdot \tilde{\mathbf{V}}] \cdot \mathbf{\Pi} \,.$$

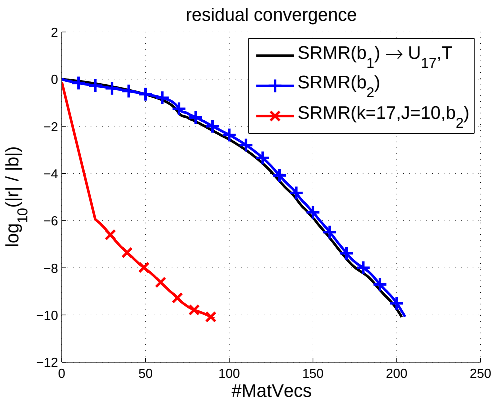
SRMR - Method

With this we get:

SRMR Prototype

1. Solve first system $\mathbf{A}\cdot\mathbf{x}^{(\iota)}=\mathbf{b}^{(\iota)}$. On the fly
   1.1. store each J^{th} vector $\mathbf{v}_i$, beginning with first.
   1.2. store tridiagonal $\mathbf{T}$ from Lanczos procedure.
2. Recycle for solve of $\mathbf{A}\cdot\mathbf{x}^{(\iota+\mu)}=\mathbf{b}^{(\iota+\mu)}$ by $\mathbf{x}^{(\iota+\mu)}=\mathbf{V}\cdot(\overline{\mathbf{V}}\cdot\overline{\mathbf{T}})^{\dagger}\cdot\mathbf{b}^{(\iota+\mu)}$.
   2.1. For this compute $\mathbf{\Pi}$ and $\mathbf{K}$, latter in $\mathcal{O}(m\cdot J)$.
   2.2. $[\tilde{\mathbf{V}},\mathbf{A}\cdot\tilde{\mathbf{V}},\ldots,\mathbf{A}^{J-1}\cdot\tilde{\mathbf{V}}]$ and its transpose can be multiplied to vector in J MVs and m scalar products.
3. Naive: If $\mathbf{x}^{(\iota+\mu)}$ is not good enough, use it as initial guess.

SRMR - Performance



Can we do better?

Explanation

$k = 17,\ J = 10$

Stored: $\tilde{\mathbf{U}} \in \mathbb{C}^{N \times k}$

Recycled: $\mathcal{K}^{\star}_{J}(\mathbf{A}; \tilde{\mathbf{U}}) = \mathcal{K}_{170}(\mathbf{A}; \mathbf{b}^{(1)})$

Add. Cost: 170 orthogonalizations.

Desire: Recycle $\mathcal{K}_{206}(\mathbf{A}; \mathbf{b}^{(1)})$

Problem: Instability for high k, J

→ speed-up of 2, but far from optimal

SRBiCG - Theory

As BiCG is neither competitive nor residual minimizing, this method is only for theory.

Idea

1. Adapt SRMR to unsymmetric systems by use of Bi-Lanczos procedure.

$$\mathbf{A} \cdot \mathbf{V} = \overline{\mathbf{V}} \cdot \overline{\mathbf{T}}$$
$$\mathbf{A}^H \cdot \mathbf{W} = \overline{\mathbf{W}} \cdot \underline{\mathbf{T}}^H$$
$$\overline{\mathbf{W}}^H \cdot \overline{\mathbf{V}} = \underline{\overline{\mathbf{I}}}$$

2. For this use short representations for both $\mathbf{V}$ and $\mathbf{W}$.
3. Notice: For MV with $\mathbf{W}^H$ no MV with $\mathbf{A}^H$ is needed!

Practical improvements

Stabilization

Stability of short representations depends on:

1. Size m: $cond(\mathbf{V})$ or $cond(\mathbf{W}^H \cdot \mathbf{V})$ grows.
2. Compression J: $cond([\tilde{\mathbf{V}}, \mathbf{A} \cdot \tilde{\mathbf{V}}, ..., \mathbf{A}^{J-1} \cdot \tilde{\mathbf{V}}])$ grows.
3. MGS becomes GS: no iterative orthogonalization of $\mathbf{r}$.

All these aspects can be handled.

A-posteriori-orthogonalization

For the a-posteriori iterates, we would like to

1. conserve orthogonality of $\mathbf{r}$ to recycled $\mathcal{P}$.
2. use short recurrences, not depending on size of recycling space.

We already know how this can be done. ☺

Stabilization - Idea

Idea: Split the recurrence

Under slight modification of $\mathbf{A}$, one can split

$$\mathbf{A}\cdot\mathbf{U}=\overline{\mathbf{U}}\cdot\overline{\mathbf{T}},\ \mathbf{A}\cdot\overline{\mathbf{U}}=\overline{\mathbf{V}},\ \mathbf{A}\cdot\mathbf{V}=\overline{\mathbf{V}}\cdot\overline{\mathbf{T}},\ \mathbf{A}^H\cdot\mathbf{W}=\overline{\mathbf{W}}\cdot\underline{\mathbf{T}}^H$$

to $\mathbf{U}=[\mathbf{U}_1,\mathbf{U}_2,\ldots],\ \mathbf{V}=[\mathbf{V}_1,\mathbf{V}_2,\ldots],\ \mathbf{W}=[\mathbf{W}_1,\mathbf{W}_2,\ldots]$ with kind of

$$\mathbf{A}\cdot\overline{\mathbf{U}}_i=\overline{\mathbf{V}}_i,\ \mathbf{A}\cdot\mathbf{V}_i=\overline{\mathbf{V}}_i\cdot\overline{\mathbf{T}}_i,\ \mathbf{A}^H\cdot\mathbf{W}_i=\overline{\mathbf{W}}_i\cdot\underline{\mathbf{T}}_i^H\,,$$

where $\mathbf{T}_i$ are diagonal blocks of $\mathbf{T}$ and columns $\xi_{m+1}^{(i)}=\xi_1^{(i+1)}$.

Now for each $\mathbf{U}_i$ and $\mathbf{W}_i$, you need compressed $\tilde{\mathbf{U}}_i$, $\tilde{\mathbf{W}}_i$.
$\rightarrow$ memory tradeoff

A-posteriori-orthogonalization - Idea

Given from recycling procedure

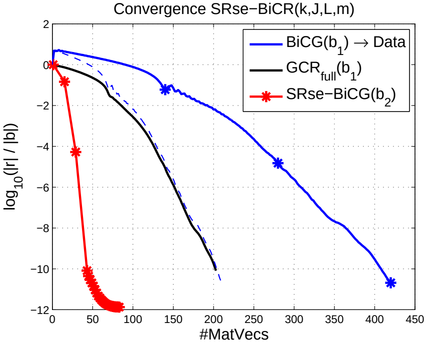
$\mathbf{x}, \mathbf{r} \in \mathbb{C}^N$ and $\mathbf{U}, \mathbf{V} \in \mathbb{C}^{N \times k}$, such that
$\mathbf{r}, \mathbf{v}_1, \ldots, \mathbf{v}_k \perp \mathcal{K}_{k \cdot J}(\mathbf{A}^H; [\mathbf{p}_1, \ldots, \mathbf{p}_k])$.

a-posteriori recurrence

After slight modification, $\mathbf{r}, \mathbf{v}_i \in \mathcal{G}_J$. If $J > k$, then one does not need further MVs for this modification.
$\rightarrow$ use IDR-type method, $s \geq k$

Remark: For efficient extension s should be > 1.

Numerical example: stabilized & a.p.-orthogonalized



... and we can do better!

1. We use **BiCG** to generate a 210-dimensional recycling space
2. For stabilization we devide into $\ell = 3$ blocks of each $J = 7$ and $k = 10$.
3. For a-posteriori-iterations we only used **IDR(1)**.

extra cost

Store: $2 \cdot 30$ columns
Compute: 210 orthogonalizations
#RDs: 210+20
#MVs: 42+40

Overview

We have

So far we have three methods, for each #RDs= $k\cdot$#MVs for recycling.

method	general	$\mathcal{U}=\mathcal{K}$	1^{st}: RD≈MV	good $\|\mathbf{r}\|$	TF
SRIDR	✓	✗	✓	✗	✓
SRMR	✗	✓	✓	✓	✓
SRBiCG	✓	✓	✗	✗	✗

To get the best from all, we start from SRBiCG and try to replace its Bi-Lanczos decomposition.

fiddling around transpose products

Maybe with IDR?

Gen. Hessenberg decomposition: $\mathbf{A} \cdot \mathbf{U} = \overline{\mathbf{U}} \cdot \overline{\mathbf{H}} \cdot \mathbf{R}^{-1}$, $\mathbf{R}$ upper triangular.
Maybe $\overline{\mathbf{T}} \approx \overline{\mathbf{H}} \cdot \mathbf{R}^{-1}$? No!

For $\mathbf{V} = \mathbf{A} \cdot \mathbf{U}$, $\mathbf{p}_i := (\mathbf{A}^H)^{g_s(i)} \cdot \mathbf{p}_{r_s(i)}$, $\mathbf{v}_i \perp \mathbf{p}_j$ for $i \neq j$ does not hold!

Maybe with ML(k)BiCGstab?

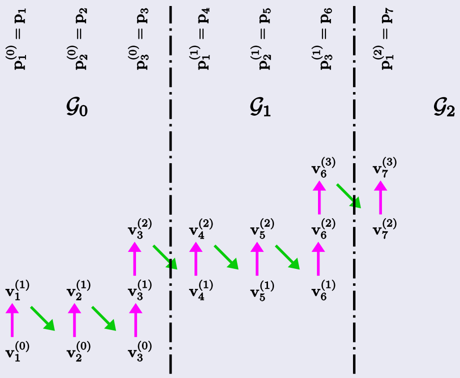
Hessenberg decomposition: $\mathbf{A} \cdot \mathbf{V} = \overline{\mathbf{V}} \cdot \overline{\mathbf{T}}$.
Canonical choose $\mathbf{W}$ with $range(\mathbf{W}(:, 1:i)) = \mathcal{K}_i(\mathbf{A}^H; [\mathbf{p}_1, ..., \mathbf{p}_s])$.

This only leads to biorthogonality, thus $\mathbf{W}^H \cdot \mathbf{V} = \mathbf{\Lambda} \neq \mathbf{I}$.

$$\mathbf{A} \cdot \mathbf{x}^{(\iota+\mu)} = \mathbf{b}^{(\iota+\mu)} \ \Rightarrow\ \mathbf{W}^H \cdot \mathbf{A} \cdot \mathbf{U} \cdot \mathbf{y}^{(\iota+\mu)} = \mathbf{\Lambda} \cdot \mathbf{y}^{(\iota+\mu)} = \mathbf{W}^H \cdot \mathbf{b}^{(\iota+\mu)}$$

Build method by hand

Idea



$\mathbf{p}_i^{(j)} = \mathbf{p}_{i \cdot k+j} = (\mathbf{A}^{\mathrm{H}})^j \cdot \mathbf{p}_i$ $\quad$ $\mathbf{v}_i^{(j)} = \mathbf{A}^j \cdot \mathbf{v}_i$ $\quad$ $\Omega_j(t) = t^j$

Bi-Lanczos result

We have biortho<u>normal</u> **V** and **W** with

$$\mathbf{v}_i = \mathbf{v}_i^{(0)}$$

$$\mathbf{w}_i = (\mathbf{A}^H)^{g_s(i)} \cdot \mathbf{p}_{r_s(i)} - \sum_{\iota < i} \gamma_{i,\iota} \cdot \mathbf{p}_\iota \,.$$

From our construction $\mathbf{A} \cdot \mathbf{V} = \overline{\mathbf{V}} \cdot \overline{\mathbf{T}}$ it follows from Bi-Lanczos-correlation

$$\mathbf{A}^H \cdot \mathbf{W} = \overline{\mathbf{W}} \cdot \underline{\mathbf{T}}^H \,.$$

→ obtain short reps for $\overline{\mathbf{V}}$ and $\overline{\mathbf{W}}$

Conclusion

1. motivation: reuse already computed orthogonality information
2. building block: compress basis matrices
3. sophistications:
   3.1. stability, a-posteriori-orthogonality
   3.2. (increasing efficiency of first solve: #RDs≈#MVs)
   3.3. (changing matrices)

[GCR-full] P. Benner and L. Feng, *Recycling Krylov Subspaces for Solving Linear Systems with successively changing Right-Hand-Sides arising in Model Reduction*, Lecture Notes in Electrical Engineering, Vol. 74, pp. 125-140, Springer 2011.

[RGMRES] R. B. Morgan, *GMRES with Deflated Restarting*, SIAM J. Sci. Comput., 24(1), pp. 20-37, 2002.

[GCROT] E. de Sturler, *Truncation Strategies for optimal Krylov subspace methods*, SIAM J. Numer. Anal., Vol. 36(3), pp. 864-889, 1999.

[GCRO-DR] M. Parks and E. de Sturler and G. Mackey and D.D. Johnson and S. Maiti, *Recycling Krylov subspaces for sequences of linear systems*, SIAM J. Sci. Comput. Vol. 28(5), pp. 1651-1674, 2006.

[R-MINRES] S. Wang and E. de Sturler and G. H. Paulino, *Large-scale topology optimization using preconditioned Krylov subspace methods with recycling*, Int. J. for Num. Meth. in Engineering, Vol. 69(12), pp. 2441-2468, 2006.

[R-BiCG] K. Ahuja and E. de Sturler and P. Benner, *Recycling BiCGSTAB with an Application to Parametric Model Order Reduction*, MPI Magdeburg preprints, pp. 13-21, 2013.

[brought me to SRIDR] M. Miltenberger, *Die IDR(s)-Methode zur Lösung von parametrisierten Gleichungssystemen*, Diplombarbeit, TU Berlin, 2009.

Thanks for your attention!

Notation

We write

1. n = iteration count = #MVs (number of matrix-vector-products)
2. typically: m =restart parameter, k =number of stored N-dimensional columns

$$\mathcal{K}_n(\mathbf{A};\mathbf{b}) = \operatorname*{span}_{i=1,\dots,n}\left\{\mathbf{A}^{i-1}\cdot\mathbf{b}\right\}$$

$$\mathcal{K}_n(\mathbf{A};[\mathbf{p}_1,\mathbf{p}_2,\dots,\mathbf{p}_k]) = \operatorname*{span}_{i=1,\dots,n}\left\{\mathbf{A}^{g_k(i)}\cdot\mathbf{p}_{r_k(i)}\right\}$$

$$\mathcal{K}_J^\star(\mathbf{A};\tilde{\mathbf{U}}) = \left\{\mathbf{x}\in\mathbb{C}^N\,|\,\mathbf{x}=\sum_{j=0}^{J-1}\mathbf{A}^j\cdot\tilde{\mathbf{U}}\cdot\gamma_j\right\}$$

$$g_k(i) = \lfloor (i-1)/k \rfloor, \qquad r_k(i) = \operatorname{mod}(i-1,k)+1$$

SRBiCG - Assessment

SRBiCG is uncompetitive

1. For first solve: $\#\text{RDs} = 2 \cdot \#\text{MVs}$, too bad ratio
2. need to compute shadow basis, leads to
   2.1. need for $\mathbf{A}^H$ products
   2.2. lack of residual minimizing property

Outlook: SRse-ML(k)BiCG

I found a method with these properties

1. For first solve: $\#\text{RDs} = k/(k+1) \cdot \#\text{MVs}$
2. no need to compute shadow basis, leads to
   2.1. no need for $\mathbf{A}^H$ products
   2.2. optional use of residual minimizing property

Idea for Solution of Voodoo type

Recycling is linear operator

The recycling procedure can be interpreted as matrix:

$$\mathcal{L}^{(\iota)}(\mathcal{U}) := \mathcal{U} \cdot (\mathbf{A}^{(\iota)} \cdot \mathcal{U})^{\dagger}$$

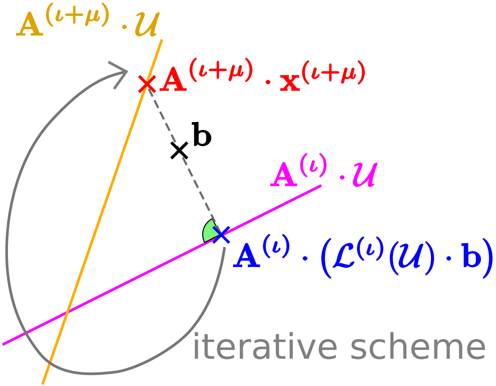
To approximate $\mathcal{L}^{(\iota+\mu)}(\mathcal{U})$, we use $\mathcal{L}^{(\iota)}(\mathcal{U})$ as preconditioner for $\mathbf{A}^{(\iota+\mu)} \cdot \mathbf{x}^{(\iota+\mu)} = \mathbf{b}^{(\iota+\mu)}$:

$$\mathcal{L}^{(\iota)}(\mathcal{U}) \cdot \mathbf{A}^{(\iota+\mu)} \cdot \mathbf{x}^{(\iota+\mu)} = \mathcal{L}^{(\iota)}(\mathcal{U}) \cdot \mathbf{b}^{(\iota+\mu)}\,.$$

This is solved iteratively.
$\rightarrow$ converges to

$$\mathbf{x}^{(\iota+\mu)} = \mathcal{U} \cdot \left((\mathbf{A}^{(\iota)} \cdot \mathbf{U})^{\dagger} \cdot (\mathbf{A}^{(\iota+\mu)} \cdot \mathcal{U}) \right)^{\dagger} \cdot (\mathbf{A}^{(\iota)} \cdot \mathcal{U})^{\dagger} \cdot \mathbf{b}^{(\iota+\mu)}.$$

Geometric interpretation (hermitian case)



The orthogonal residual becomes biorthogonal.

$$\left\{ \begin{array}{rl} -\nabla \cdot \big(a(u) \cdot \nabla u\big) = f & \text{in } \Omega = (0,1)^2 \\ u = 0 & \text{on } \partial\Omega \\ \sin(\pi \cdot x) \cdot \sin(\pi \cdot y)^2 = f & \end{array} \right\}$$

How meaningful?

We cannot check. **R** of **GCR**'s QR-decomposition is too ill.

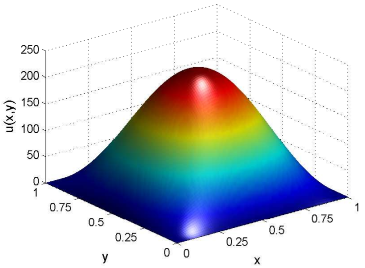


Figure 3: for $a(u) = 1$

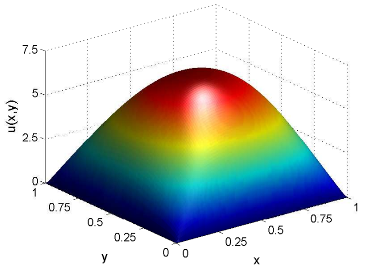


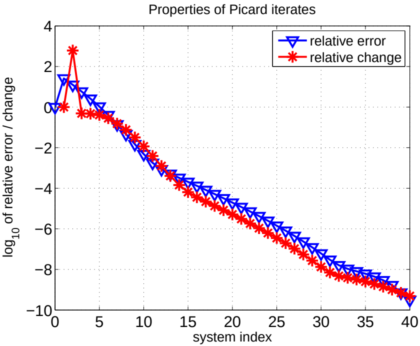
Figure 4: for $a(u) = 1 + 10 \cdot u$

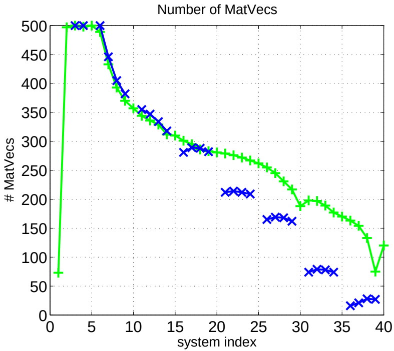
Finite differences: $$\sum_{\tilde{p}\in\mathcal{B}(p)} \frac{u_p - u_{\tilde{p}}}{\Delta x^2} \cdot \frac{a_p + a_{\tilde{p}}}{2} = f_p$$

Damped Picard iteration, $\alpha = 0.5$

$$\mathbf{x}^{(0)} = \mathbf{0}$$

$$\left\{ \begin{aligned} \mathbf{A}(\mathbf{x}^{(\iota)}) \cdot \tilde{\mathbf{x}}^{(\iota+1)} &= \mathbf{b} \in \mathbb{R}^{10000} \\ \mathbf{x}^{(\iota+1)} &= (1-\alpha)\,\mathbf{x}^{(\iota)} + \alpha\,\tilde{\mathbf{x}}^{(\iota+1)} \end{aligned} \right\}, \quad \iota = 0, \ldots, 40$$





Conjugate Gradients vs. SRMR($k = 40, w = 3$)