Block and Band Lanczos Algorithms: a Review of Options

Martin H. Gutknecht and Damian Loher

ETH Zurich, Seminar for Applied Mathematics

COMSON Autumn School in Model Order Reduction, Sep. 21–25, 2009

Prerequisites: Krylov (sub)spaces

Given: $\mathbf{A} \in \mathbb{C}^{N \times N}$, $\mathbf{y}_0 \in \mathbb{C}^N$.

nth Krylov subspace generated by A from $\mathbf{y}_0$:

$$\mathcal{K}_n :\equiv \mathcal{K}_n(\mathbf{A}, \mathbf{y}_0) :\equiv \operatorname{span}\{\mathbf{y}_0, \mathbf{A}\mathbf{y}_0, \ldots, \mathbf{A}^{n-1}\mathbf{y}_0\}.$$

Applications:

- Solving linear systems of equations $\mathbf{A}\mathbf{x} = \mathbf{b}$.
  Here, $\mathbf{y}_0 \rightsquigarrow \mathbf{r}_0 :\equiv \mathbf{b} - \mathbf{A}\mathbf{x}_0$.
- Solving linear (matrix) eigenvalue problems $\mathbf{A}\mathbf{y} = \mathbf{y}\lambda$.
- Model order reduction of linear time-invariant SISO systems. Here, $N \rightsquigarrow n$, $\mathbf{y}_0 \rightsquigarrow \mathbf{b}$.

The nonsymmetric/two-sided Lanczos process

Given $\mathbf{A}$, $\mathbf{y}_0$, $\widetilde{\mathbf{y}}_0$, the Lanczos process generates

$$\begin{aligned} \mathbf{y}_n &\in \mathcal{K}_{n+1} :\equiv \text{span}\,(\mathbf{y}_0, \mathbf{A}\mathbf{y}_0, \ldots, \mathbf{A}^n\mathbf{y}_0) \subset \mathbb{C}^N\,, \\ \widetilde{\mathbf{y}}_m &\in \widetilde{\mathcal{K}}_{m+1} :\equiv \text{span}\,(\widetilde{\mathbf{y}}_0, \mathbf{A}^\star\widetilde{\mathbf{y}}_0, \ldots, (\mathbf{A}^\star)^m\widetilde{\mathbf{y}}_0) \subset \mathbb{C}^N\,, \end{aligned}$$

with

$$\boxed{\widetilde{\mathbf{y}}_m^\star\,\mathbf{y}_n = \begin{cases} 0\,, & m \neq n, \\ \delta_n\,, & m = n, \end{cases}}$$

Preferably, we choose $\|\mathbf{y}_n\| = \|\widetilde{\mathbf{y}}_n\| = 1$.

We need $\delta_n \neq 0$ for all n; otherwise, breakdown.

The look-ahead Lanczos algorithm can cure most breakdowns.

It means that at (near-)breakdowns we “cluster” several successive $\mathbf{y}_n$ and $\widetilde{\mathbf{y}}_n$ and require block orthogonality only.

SISO model reduction with the Lanczos algorithm

Consider linear time-invariant (LTI) SISO system

$$\begin{aligned} \dot{\mathbf{x}}(t) &= \mathbf{A}\mathbf{x}(t) + \mathbf{b}u(t), \qquad \mathbf{x}(0) = \mathbf{x}_0\,, \\ y(t) &= \mathbf{c}^{\mathsf{T}}\mathbf{x}(t)\,, \end{aligned}$$

and its transfer function

$$G(s) :\equiv \mathbf{c}^{\mathsf{T}}(s\mathbf{I} - \mathbf{A})^{-1}\mathbf{b}\,.$$

The nonsymmetric Lanczos algorithm yields **partial realizations** = **Padé approximations** (**moment matching**) **at** $s = \infty$ of this transfer function if we let

$$\mathbf{y}_0 \rightsquigarrow \mathbf{b}\,, \quad \widetilde{\mathbf{y}}_0 \rightsquigarrow \mathbf{c}\,.$$

"Padé via Lanczos (PVL)"; known to Rutishauser (1955).

Block Krylov (sub)spaces

Given: $\mathbf{A} \in \mathbb{C}^{N\times N}$, $\mathbf{Y}_0 \in \mathbb{C}^{N\times \ell}$.

nth block Krylov subspace $\mathcal{B}_n^{\square} := \mathcal{B}_n^{\square}(\mathbf{A}, \mathbf{Y}_0)$ generated by $\mathbf{A}$ from $\mathbf{Y}_0$:

$$\mathcal{B}_n^{\square}(\mathbf{A}, \mathbf{Y}_0) := \textbf{block span}\,(\mathbf{Y}_0, \mathbf{A}\mathbf{Y}_0, \ldots, \mathbf{A}^{n-1}\mathbf{Y}_0) \subset \mathbb{C}^{N\times\ell}$$
$$:= \left\{ \sum_{k=0}^{n-1} \mathbf{A}^k \mathbf{Y}_0 \gamma_k \,;\, \gamma_k \in \mathbb{C}^{\ell\times\ell}\ (k = 0, \ldots, n-1) \right\}.$$

DEFINITION. A **block vector** is a matrix $\mathbf{Y} \in \mathbb{C}^{N\times\ell}$.

Hence, the elements of $\mathcal{B}_n^{\square}$ are block vectors.

Block Krylov spaces (cont'd)

Each of the ℓ columns $y^{(j)}$ $(j = 1, \ldots, \ell)$ of some $\mathbf{Y} \in \mathcal{B}_n^{\square}$ lies in the space

$$\boxed{\mathcal{B}_n := \mathcal{B}_n(\mathbf{A}, \mathbf{Y}_0) := \mathcal{K}_n^{(1)} + \cdots + \mathcal{K}_n^{(\ell)} \subseteq \mathbb{C}^N ,}$$

with the ℓ "usual" Krylov (sub)spaces,

$$\mathcal{K}_n^{(j)} := \mathcal{K}_n(\mathbf{A}, y_0^{(j)}) := \operatorname{span}(y_0^{(j)}, \mathbf{A}y_0^{(j)}, \ldots, \mathbf{A}^{n-1}y_0^{(j)}) \subset \mathbb{C}^N .$$

In other words, each column $y^{(j)}$ is from a space that is as large as all ℓ "usual" Krylov spaces together: $\dim \mathcal{B}_n \leq n\ell$.

$\mathcal{B}_n^{\square}$ is a Cartesian product of ℓ copies of $\mathcal{B}_n$:

$$\boxed{\mathcal{B}_n^{\square} = \underbrace{\mathcal{B}_n \times \cdots \times \mathcal{B}_n}_{\ell \text{ times}} .}$$

Block Krylov (sub)spaces (cont'd)

Applications:

- Solving linear systems of equations with multiple right-hand sides $\mathbf{AX} = \mathbf{B}$.
  Here, $\mathbf{Y}_0 \rightsquigarrow \mathbf{R}_0 :\equiv \mathbf{B} - \mathbf{AX}_0$.
- Solving linear (matrix) eigenvalue problems $\mathbf{Ay} = \mathbf{y}\lambda$ with multiple eigenvalues.
- Model order reduction of linear time-invariant MIMO systems. Here, $N \rightsquigarrow n$, $\ell \rightsquigarrow m$, $\widetilde{\ell} \rightsquigarrow p$:
  Nonsymmetric block Lanczos algorithms with
  $$\mathbf{Y}_0 \rightsquigarrow \mathbf{B} \in \mathbb{C}^{n\times m}, \quad \widetilde{\mathbf{Y}}_0 \rightsquigarrow \mathbf{C}^\star \in \mathbb{C}^{n\times p}$$
  yield partial realizations (matrix Padé approximations at $s = \infty$) of the transfer function $\mathbf{G}(s) :\equiv \mathbf{C}(s\mathbf{I} - \mathbf{A})^{-1}\mathbf{B}$.

Block Krylov spaces (cont'd)

Main reasons for using block Krylov spaces for solving linear systems or eigenvalue problems:

- The search space for each $x^{(j)}$ is much bigger, namely as big as all ℓ Krylov spaces together.
  But do these extra dimensions really help much?
- In good implementations, ℓ matrix-vector products with **A** are computed at once, and this is much faster than ℓ separate matrix-vector products, even on sequential computers (due to better usage of cached data).
- The geometric multiplicity of eigenvalues can be found.

Work on block methods started in the 1970ies with block Lanczos for symmetric EVal problems and block CG.

Nearly all work for nonsymmetric matrices since 1990.

Linear dependence, deflation

$$\mathcal{B}_n := \mathcal{B}_n(\mathbf{A}, \mathbf{Y}_0) := \mathcal{K}_n^{(1)} + \cdots + \mathcal{K}_n^{(\ell)}$$

is, in general, not a direct sum.

Already the columns of $\mathbf{Y}_0$ could be linearly dependent.

But also some of the later generated Krylov subspace dimensions may not enlarge the block Krylov subspace.

The treatment of these cases requires **deflation**: the explicit determination of linear dependencies during the generation of the block Krylov subspaces ($\rightsquigarrow$ application of **RRQR** or **SVD**).

(The term "deflation" is also used with different meanings.)

Linear dependence, deflation (cont'd)

EXAMPLES (of extreme cases)

1. $\mathbf{Y}_0$ is made up of ℓ identical vectors y,

$$\mathbf{Y}_0 := \left[\begin{array}{ccccc} y & y & y & \dots & y \end{array} \right] .$$

2. $$\mathbf{Y}_0 := \left[\begin{array}{ccccc} y & \mathbf{A}y & \mathbf{A}^2y & \dots & \mathbf{A}^{\ell-1}y \end{array} \right] .$$
   Here, even if $\operatorname{rank} \mathbf{R}_0 = \ell$, still $\operatorname{rank} \left[\begin{array}{cc} \mathbf{Y}_0 & \mathbf{A}\mathbf{Y}_0 \end{array} \right] \leq \ell + 1$.

3. $\mathbf{Y}_0$ has ℓ columns that are linear combinations of ℓ eigenvectors of $\mathbf{A}$. Then $\operatorname{rank} \left[\begin{array}{cc} \mathbf{Y}_0 & \mathbf{A}\mathbf{Y}_0 \end{array} \right] \leq \ell$.

When and how to deflate?

We can use the **block Arnoldi process** to construct an orthonormal basis of

$$\mathcal{B}_n := \mathcal{B}_n(\mathbf{A}, \mathbf{R}_0) := \mathcal{K}_n^{(1)} + \cdots + \mathcal{K}_n^{(\ell)} .$$

Need deflation when

$$\dim \mathcal{B}_n < \dim \mathcal{K}_n^{(1)} + \cdots + \dim \mathcal{K}_n^{(\ell)} .$$

More exactly: let $\ell_n := \dim \mathcal{B}_{n+1} - \dim \mathcal{B}_n$. (Here, $\mathcal{B}_0 := \emptyset$.)

Then we need deflation in nth step if $\ell_n < \ell_{n-1}$.

In single RHS case ($\ell = 1$):

$\dim \mathcal{K}_{n+1} = \dim \mathcal{K}_n \Longrightarrow$ termination of Arnoldi / GMRES

In block case: termin. when $\dim \mathcal{B}_{n+1} = \dim \mathcal{B}_n$, i.e., $\ell_n = 0$.

When and how to deflate? (cont'd)

The answer to the question of when to deflate also depends on the application and the algorithm used.

- When solving linear systems it is good if $\mathcal{B}_n$ becomes invariant for a small ℓ ($\rightsquigarrow$ all systems are solved).
  If we apply block GMRES, deflation reduces cost, but is not really needed for stability.
- If we apply block Lanczos / block BICG, deflation is crucial for stability.

The block Lanczos or BLBIO algorithm

Aim: For given $\mathbf{A} \in \mathbb{C}^{N\times N}$ and $\mathbf{Y}_0,\ \widetilde{\mathbf{Y}}_0 \in \mathbb{C}^{N\times \ell}$, generate a pair of *block-biorthogonal* (or: *block-dual*) *bases* of *Lanczos block vectors*

$$\mathbf{Y}_n \in \mathcal{B}^{\square}_{n+1} :\equiv \text{block span}\,(\mathbf{Y}_0, \mathbf{A}\mathbf{Y}_0, \ldots, \mathbf{A}^n\mathbf{Y}_0) \subset \mathbb{C}^{N\times \ell},$$
$$\widetilde{\mathbf{Y}}_m \in \widetilde{\mathcal{B}}^{\square}_{m+1} :\equiv \text{block span}\,(\widetilde{\mathbf{Y}}_0, \mathbf{A}^\star\widetilde{\mathbf{Y}}_0, \ldots, (\mathbf{A}^\star)^m\widetilde{\mathbf{Y}}_0) \subset \mathbb{C}^{N\times \ell},$$

satisfying

$$\boxed{\widetilde{\mathbf{Y}}_m^\star\,\mathbf{Y}_n = \begin{cases} \boldsymbol{o}\,, & m \neq n, \\ \boldsymbol{\delta}_n\,, & m = n, \end{cases}} \tag{1}$$

with nonsingular $\boldsymbol{\delta}_n \in \mathbb{C}^{\ell\times\ell}$.

Petrov–Galerkin conditions implied by (1):

$$\boxed{\widetilde{\mathcal{B}}_n \perp y_n^{(i)} \in \mathcal{B}_{n+1}\,, \qquad \mathcal{B}_n \perp \widetilde{y}_n^{(i)} \in \widetilde{\mathcal{B}}_{n+1} \qquad (i = 1, \ldots, \ell)\,.}$$

The band Lanczos algorithm

Ruhe **'79**$_{\text{MathComp}}$ (sym.), Freund/Malhotra *'95*/**'97**$_{\text{LAA}}$,
Aliaga/Boley/Freund/Hernández *'96*/**'99**$_{\text{MathComp}}$,
Freund **'00**$_{\text{EV Templates}}$, Bai/Freund **'00**$_{01}$SISC (sym.), ...

Mathematically just a special case of a block Lanczos algorithm: δ_n **is diagonal** $(\forall n)$. No pivoting.

Analogous to Gauss elimination without pivoting.

Algorithmically different from block Lanczos:
only a pair of vectors is added to the bases $\mathcal{B}_n$ and $\widetilde{\mathcal{B}}_n$ at every step, not a pair of blocks.

How to start nonsymmetric block Lanczos?

Depending on the application, there is more or less freedom:

- When solving linear systems with block Lanczos / block BICG we may choose $\widetilde{\mathbf{Y}}_0$, while $\mathbf{Y}_0 := \mathbf{R}_0$.
- When solving eigenvalue problems we may choose both $\widetilde{\mathbf{Y}}_0$ and $\mathbf{Y}_0$.
- For model order reduction
  $$\mathbf{Y}_0 := \mathbf{B} \in \mathbb{C}^{n\times m}, \quad \widetilde{\mathbf{Y}}_0 := \mathbf{C}^\star \in \mathbb{C}^{n\times p}.$$
  So, $\ell := m$, $\widetilde{\ell} := p$; i.e., in general, $\widetilde{\ell} \neq \ell$.

Implementing nonsymmetric block Lanczos / BiCG

- May need deflation in (right) block Krylov space.
- May need deflation in left/shadow block Krylov space.
- May need to avoid “serious” Lanczos breakdowns.
- May need to avoid pivot breakdowns in block BiCG.

PhD thesis of Damian Loher (Diss. no. 16337, ETH Zurich, 2006).

Here in this talk, we look at the case $\widetilde{\ell} = \ell$ first.

BLBIO algorithm:

Choose $\mathbf{Y}_0$, $\widetilde{\mathbf{Y}}_0 \in \mathbb{C}^{N\times\ell}$ such that $\boldsymbol{\delta}_0 := \widetilde{\mathbf{Y}}_0^\star \mathbf{Y}_0$ is nonsingular, and set $\boldsymbol{\beta}_{-1} := \mathbf{o} \in \mathbb{C}^{\ell\times\ell}$. For $n = 0, 1, \ldots$ compute

$$\begin{aligned}
\boldsymbol{\delta}_n^{\mathbf{A}} &:= \widetilde{\mathbf{Y}}_n^\star \mathbf{A}\mathbf{Y}_n\,, && \text{(2a)}\\
\boldsymbol{\alpha}_n &:= \boldsymbol{\delta}_n^{-1}\boldsymbol{\delta}_n^{\mathbf{A}}\,, && \text{(2b)}\\
\widetilde{\boldsymbol{\alpha}}_n &:= \boldsymbol{\delta}_n^{-\star}(\boldsymbol{\delta}_n^{\mathbf{A}})^\star = (\boldsymbol{\delta}_n^{\mathbf{A}}\boldsymbol{\delta}_n^{-1})^\star = \boldsymbol{\delta}_n^{-\star}\boldsymbol{\alpha}_n^\star\boldsymbol{\delta}_n^\star\,, && \text{(2c)}\\
\boldsymbol{\beta}_{n-1} &:= \boldsymbol{\delta}_{n-1}^{-1}\widetilde{\boldsymbol{\gamma}}_{n-1}^\star\boldsymbol{\delta}_n \qquad (\text{if} \quad n > 0), && \text{(2d)}\\
\widetilde{\boldsymbol{\beta}}_{n-1} &:= \boldsymbol{\delta}_{n-1}^{-\star}\boldsymbol{\gamma}_{n-1}^\star\boldsymbol{\delta}_n^\star \qquad (\text{if} \quad n > 0), && \text{(2e)}\\
\mathbf{Y}_{\mathbf{tmp}} &:= \mathbf{A}\mathbf{Y}_n - \mathbf{Y}_n\boldsymbol{\alpha}_n - \mathbf{Y}_{n-1}\boldsymbol{\beta}_{n-1}\,, && \text{(2f)}\\
\widetilde{\mathbf{Y}}_{\mathbf{tmp}} &:= \mathbf{A}^\star\widetilde{\mathbf{Y}}_n - \widetilde{\mathbf{Y}}_n\widetilde{\boldsymbol{\alpha}}_n - \widetilde{\mathbf{Y}}_{n-1}\widetilde{\boldsymbol{\beta}}_{n-1}\,, && \text{(2g)}\\
\boldsymbol{\delta}_{\mathbf{tmp}} &:= \widetilde{\mathbf{Y}}_{\mathrm{tmp}}^\star\mathbf{Y}_{\mathrm{tmp}}\,. && \text{(2h)}
\end{aligned}$$

BLBIO algorithm: (cont'd)

If δ_{tmp} is singular, stop (termination, deflation or breakdown); otherwise, **choose γ_n, $\widetilde{\gamma}_n$, and δ_{n+1}** such that

$$\boxed{\widetilde{\gamma}_n^{\star}\delta_{n+1}\gamma_n = \delta_{\text{tmp}}\,,} \tag{3}$$

set

$$\mathbf{Y}_{n+1} := \mathbf{Y}_{\text{tmp}}\gamma_n^{-1}\,, \qquad \widetilde{\mathbf{Y}}_{n+1} := \widetilde{\mathbf{Y}}_{\text{tmp}}\widetilde{\gamma}_n^{-1}\,, \tag{4}$$

and proceed with the next step.

The choices in (3) distinguish the various versions of 3-term block Lanczos algorithms mathematically.
See slide 'Normalization' below.

Termination / deflation / breakdowns

The block Lanczos algorithm in the form described above stops whenever the cross product δ_{tmp} is singular. Reasons:

- $\mathbf{Y}_{\text{tmp}}$ may not have full rank ($\rightsquigarrow$ *deflation* or *termination*)
- $\widetilde{\mathbf{Y}}_{\text{tmp}}$ may not have full rank ($\rightsquigarrow$ *deflation* or *termination*)
- δ_{tmp} may be singular although $\mathbf{Y}_{\text{tmp}}$ and $\widetilde{\mathbf{Y}}_{\text{tmp}}$ have full rank ($\rightsquigarrow$ *serious breakdown* or *block Lanczos breakdown*)
  More generally: the rank of δ_{tmp} may be smaller than $\min\{\operatorname{rank}\mathbf{Y}_{\text{tmp}}, \operatorname{rank}\widetilde{\mathbf{Y}}_{\text{tmp}}\}$ ($\rightsquigarrow$ *serious breakdown*)

In the case where $\operatorname{rank}\mathbf{Y}_{\text{tmp}} \neq \operatorname{rank}\widetilde{\mathbf{Y}}_{\text{tmp}}$ it is at this point unclear how to proceed, since we assumed that $\mathbf{Y}_n$ and $\widetilde{\mathbf{Y}}_n$ have the same number of columns.

Normalization: some options

The choice of $\widetilde{\gamma}_n,\ \delta_{n+1},\ \gamma_n$ in

$$\boxed{\widetilde{\gamma}_n^{\star}\delta_{n+1}\gamma_n = \delta_{\text{tmp}}\ ,} \tag{3}$$

distinguish the various versions of 3-term block Lanczos. This choice has a much greater effect than in the 'normal' Lanczos process. Here are some options:

1. *Choose diagonal* γ_n *and* $\widetilde{\gamma}_n$ *such that all vectors are* normalized
2. *Choose* $\gamma_n := \widetilde{\gamma}_n := \mathbf{I}_s$
3. *Choose* (3) *as* LDU factorization *of* δ_{tmp} *(no pivoting)*
4. *Combine 3)*[LDU] *with 1)*[normal.] [Aliaga/B/F/H *'96*/**'99**$_{\text{MC}}$]

Normalization: some options (cont'd)

5. *Modify 3) and 4) by using an* LDU factor. with pivoting
6. Apply *QR factorization with column pivoting to* $\mathbf{Y}_{tmp}$ *and* $\widetilde{\mathbf{Y}}_{tmp}$ *to get* $\mathbf{Y}_{n+1}$ *and* $\widetilde{\mathbf{Y}}_{n+1}$ *(alternative: another rank-revealing QR factorization)*
7. *Choose* (3) *as* SVD *of* δ_{tmp}
8. *Combine 1)* [normal.] *with 7)* [SVD]
9. *Combine 6)* [QR] *with 7)* [SVD] [Bai/Day/Ye **'99**$_{\text{SIMAX}}$]
10. *Standard* BLBICG *requires that* $\gamma_n := -\alpha_n - \beta_{n-1}$

Some details on a step with option 9

First: compute RRQR factorizations of $\mathbf{Y}_{\text{tmp}}$ and $\widetilde{\mathbf{Y}}_{\text{tmp}}$ to get orthonormal columns in these block vectors:

$$\boxed{\mathbf{Y}_{\text{tmp}}\boldsymbol{\pi} \equiv: \mathbf{Q}\boldsymbol{\rho}\,, \quad \widetilde{\mathbf{Y}}_{\text{tmp}}\widetilde{\boldsymbol{\pi}} \equiv: \widetilde{\mathbf{Q}}\widetilde{\boldsymbol{\rho}}} \tag{5}$$

with upper triangular $\boldsymbol{\rho}$, $\widetilde{\boldsymbol{\rho}} \in \mathbb{C}^{\ell\times\ell}$, permutation matrices $\boldsymbol{\pi}$, $\widetilde{\boldsymbol{\pi}}$.

Define $\boxed{\boldsymbol{\delta} :\equiv \widetilde{\mathbf{Q}}^{\star}\mathbf{Q}}$ and compute its SVD:

$$\boxed{\boldsymbol{\delta} \equiv: \widetilde{\boldsymbol{\gamma}}^{\star}\boldsymbol{\delta}_{n+1}\boldsymbol{\gamma}\,.} \tag{6}$$

Then let

$$\boxed{\mathbf{Y}_{n+1} :\equiv \mathbf{Q}\boldsymbol{\gamma}^{\star}\,, \quad \widetilde{\mathbf{Y}}_{n+1} :\equiv \widetilde{\mathbf{Q}}\widetilde{\boldsymbol{\gamma}}^{\star}\,.} \tag{7}$$

Some details on a step with option 9 (cont'd)

Eq. (3), $\widetilde{\gamma}_n^\star \delta_{n+1} \gamma_n = \delta_{\text{tmp}}$, holds then with

$$\boxed{\gamma_n :\equiv \gamma\rho\pi^\star\,, \quad \widetilde{\gamma}_n :\equiv \widetilde{\gamma}\widetilde{\rho}\widetilde{\pi}^\star\,.} \tag{8}$$

Indeed,

$$\boxed{\begin{aligned}\delta_{\text{tmp}} :\equiv \widetilde{\mathbf{Y}}_{\text{tmp}}^\star \mathbf{Y}_{\text{tmp}} &= \widetilde{\pi}\widetilde{\rho}^\star\delta\rho\pi^\star = \widetilde{\pi}\widetilde{\rho}^\star\widetilde{\gamma}^\star\delta_{n+1}\gamma\rho\pi^\star \\ &= \widetilde{\gamma}_n^\star\delta_{n+1}\gamma_n\,,\end{aligned}} \tag{9}$$

but these quantities are not needed in the algorithm.

Now the SVD provides all information about the *canonical angles* between the subspaces spanned by $\mathbf{Y}_{\text{tmp}}$ and $\widetilde{\mathbf{Y}}_{\text{tmp}}$.

We obtain optimal information about breakdowns.

Loss of biorthogonality

Locally, biorthogonality (duality) is enforced. Globally, it is inherited in theory but lost in practice.

Loss can be reduced by "extended local biorthogonality", *i.e.*, repeated computation of δ_n, $\delta_n^{\mathbf{A}}$, α_n, $\widetilde{\alpha}_n$, β_n, $\widetilde{\beta}_n$, $\mathbf{Y}_{\text{tmp}}$, $\widetilde{\mathbf{Y}}_{\text{tmp}}$, ...; see Simon ['*82*/**'84**$_{\text{LAA}}$] for the symmetric non-block case, Day [**'93**$_{\text{Diss}}$,'*95*/**'97**$_{\text{SIMAX}}$] for the nonsymmetric non-block case.

Shorthand notation

For the *shorthand notation* or *matrix representation*

$$\mathbf{A}\mathbb{Y}_n = \mathbb{Y}_{n+1}\underline{\mathbf{T}}_n\,, \qquad \mathbf{A}^\star\widetilde{\mathbb{Y}}_n = \widetilde{\mathbb{Y}}_{n+1}\underline{\widetilde{\mathbf{T}}}_n \qquad (n \le \bar{\nu})\,,$$

$$\widetilde{\mathbb{Y}}_n\mathbb{Y}_n = \mathbf{D}_{\delta;n} \qquad (n \le \bar{\nu})$$

we need to define block matrices of right and left Lanczos vectors

$$\mathbb{Y}_n :\equiv \left[\begin{array}{cccc} \mathbf{Y}_0 & \mathbf{Y}_1 & \dots & \mathbf{Y}_{n-1} \end{array}\right], \qquad \widetilde{\mathbb{Y}}_n :\equiv \left[\begin{array}{cccc} \widetilde{\mathbf{Y}}_0 & \widetilde{\mathbf{Y}}_1 & \dots & \widetilde{\mathbf{Y}}_{n-1} \end{array}\right],$$

the block diagonal Gramian matrix

$$\mathbf{D}_{\delta;n} :\equiv \mathrm{blockdiag}(\boldsymbol{\delta}_0, \dots, \boldsymbol{\delta}_{n-1}) = \widetilde{\mathbb{Y}}_n^\star\mathbb{Y}_n\,,$$

Shorthand notation (cont'd)

the (extended) block tridiagonal $(n+1)\ell \times n\ell$ matrix

$$\underline{\mathbf{T}}_n := \left[\begin{array}{c} \mathbf{T}_n \\ \hline \gamma_{n-1}\mathbf{L}_n^{\mathsf{T}} \end{array}\right] = \left[\begin{array}{cccc} \alpha_0 & \beta_0 & & \\ \gamma_0 & \alpha_1 & \beta_1 & \\ & \gamma_1 & \alpha_2 & \ddots \\ & & \ddots & \ddots & \beta_{n-2} \\ & & & \gamma_{n-2} & \alpha_{n-1} \\ \hline & & & & \gamma_{n-1} \end{array}\right],$$

and the analogous matrix $\underline{\widetilde{\mathbf{T}}}_n$ whose coefficients have tildes. It is related to $\underline{\mathbf{T}}_n$ by $\widetilde{\mathbf{T}}_n = \mathbf{D}_n^{-\star}\widetilde{\mathbf{T}}_n^{\star}\mathbf{D}_n^{\star} = (\mathbf{D}_n\widetilde{\mathbf{T}}_n\mathbf{D}_n^{-1})^{\star}$.

Dealing with left and right blocks of unequal size

1st fundamental observation: We can choose the sizes of the diagonal blocks of

$$\mathbf{D}_{\delta;n} = \widetilde{\mathbb{Y}}_n^\star \mathbb{Y}_n \,,$$

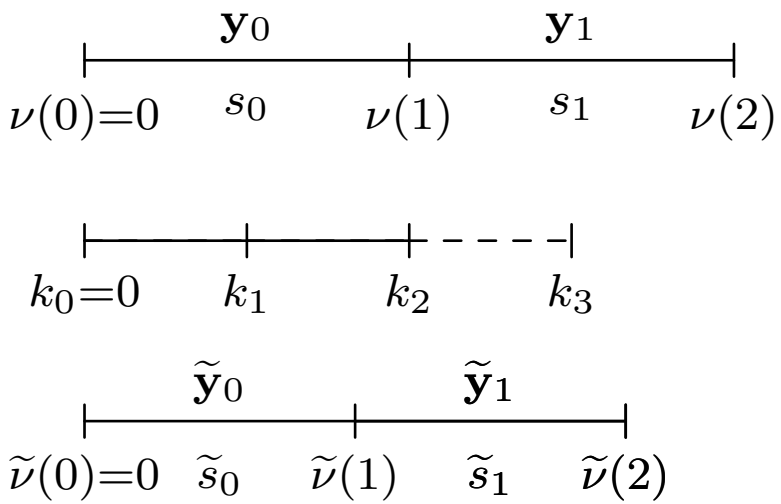
independently from the block sizes of the left and right block Krylov bases, $\mathbb{Y}_n$ and $\mathbb{Y}_n$.

2nd fundamental observation: With little extra cost we can change the order of vectors (“pivot”) within blocks.

$\Longrightarrow$ We just extract from $\widetilde{\mathbb{Y}}_n$ and $\mathbb{Y}_n$ **“clusters”** of equal size.

Dealing with ... blocks of unequal size (cont'd)

We need three index sequences, $\{\nu(i)\}$, $\{\widetilde{\nu}(i)\}$, $\{k_i\}$ to keep track of the block sizes of the right block Krylov basis, of the left block Krylov basis, and of the "clusters" common to both:

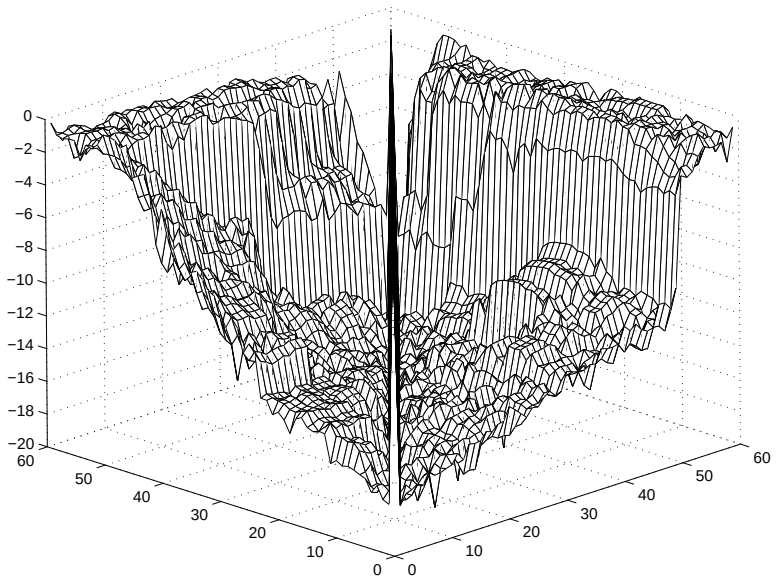


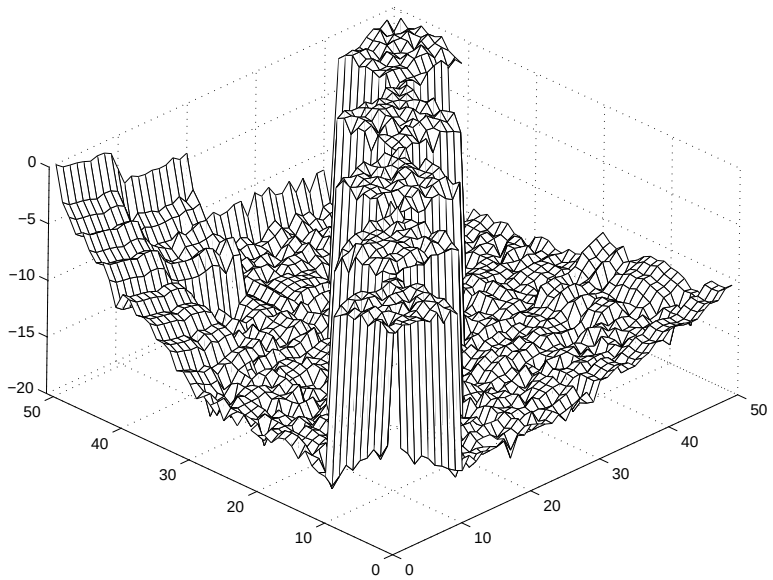
Here, k_3 is suitably chosen to obtain a well-conditioned δ_3.

Additionally, we need to keep track of permutations or of local linear transformations of basis vectors in a block.

Dealing with ... blocks of unequal size (cont'd)

$\log_{10}$ of absolute values of the inner products of the first 50 left and right basis vectors if "small" or "large" clusters are used, respectively:





Conclusions

- In theory, block Lanczos algorithms are a perfect tool for MIMO model order reduction.
- However, a robust implementation is very difficult.
- May need deflation in (right) block Krylov space.
- May need deflation in left/shadow block Krylov space.
- May need to avoid “serious” Lanczos breakdowns.
- Capitalizing upon the many options can improve the stability drastically.
- Further improvements of Loher’s code are being discussed.