Block Krylov Space Solvers: a Survey

Martin H. Gutknecht

Seminar for Applied Mathematics
ETH Zurich

ICMSEC, Chinese Academy of Sciences, Sep. 2006

Partly joint work with Thomas Schmelzer, Oxford University
Given is a nonsingular linear system with $s$ RHSs,

$$Ax = b$$  \hspace{1cm} (1)

where

$$A \in \mathbb{C}^{N \times N}, \quad b \in \mathbb{C}^{N \times s}, \quad x \in \mathbb{C}^{N \times s}. \hspace{1cm} (2)$$

Using Gauss elimination we can solve it much more efficiently than $s$ single linear systems with different matrices, since the LU decomposition of $A$ is computed only once.

It does not matter if all the RHSs are known at the beginning or are produced one after another while the systems are solved.
Given is a nonsingular linear system with $s$ RHSs, 

$$Ax = b$$  \hspace{1cm} (1) 

where 

$$A \in \mathbb{C}^{N \times N}, \quad b \in \mathbb{C}^{N \times s}, \quad x \in \mathbb{C}^{N \times s}.$$  \hspace{1cm} (2) 

Using Gauss elimination we can solve it much more efficiently than $s$ single linear systems with different matrices, since the LU decomposition of $A$ is computed only once.

It does not matter if all the RHSs are known at the beginning or are produced one after another while the systems are solved.
If iterative methods are applied, it is hard to solve (1)–(2) much faster than $s$ systems with single RHS.

Two approaches:

- using the (iterative) solution of a seed system for solving subsequently the other systems faster,

- using block iterations: treat several RHSs at once.

In the second case, all RHSs are needed at once.

Most iterative methods are generalized easily to block methods, but the stability of block methods requires extra effort. Block methods may be, but need not be much faster than solving the $s$ systems separately.

Related iterative methods for eigenvalues allow us to find multiple eigenvalues and corresponding eigenspaces.
If iterative methods are applied, it is hard to solve (1)–(2) much faster than $s$ systems with single RHS.

Two approaches:

- using the (iterative) solution of a **seed system** for solving subsequently the other systems faster,
- using **block iterations**: treat several RHSs at once.

In the second case, all RHSs are needed at once.

Most iterative methods are generalized easily to block methods, but the stability of block methods requires extra effort. Block methods may be, but need not be much faster than solving the $s$ systems separately.

Related iterative methods for eigenvalues allow us to find multiple eigenvalues and corresponding eigenspaces.
We seek approximate solutions of the form

$$x_n \in x_0 + B_n(A, r_0),$$

(3)

where the **block Krylov space** $B_n := B_n(A, r_0)$ is defined by

$$B_n(A, r_0) := \text{block span} (r_0, Ar_0, \ldots, A^{n-1}r_0) \subset \mathbb{C}^{N \times s}$$

(4)

$$\equiv \left\{ \sum_{k=0}^{n-1} A^k r_0 \gamma_k ; \, \gamma_k \in \mathbb{C}^{s \times s} (k = 0, \ldots, n - 1) \right\}.$$  

(5)

**Definition.** A (complex) **block vector** is a matrix $y \in \mathbb{C}^{N \times s}$.

Hence, the elements of $B_n$ are block vectors.
We seek approximate solutions of the form

$$x_n \in x_0 + \mathcal{B}_n^\Box(A, r_0),$$

(3)

where the **block Krylov space** $\mathcal{B}_n^\Box := \mathcal{B}_n^\Box(A, r_0)$ is defined by

$$\mathcal{B}_n^\Box(A, r_0) := \text{block span}\ (r_0, Ar_0, \ldots, A^{n-1}r_0) \subset \mathbb{C}^{N \times s}$$

(4)

$$:= \left\{ \sum_{k=0}^{n-1} A^k r_0 \gamma_k ; \gamma_k \in \mathbb{C}^{s \times s} (k = 0, \ldots, n - 1) \right\}. $$

(5)

**Definition.** A (complex) **block vector** is a matrix $y \in \mathbb{C}^{N \times s}$.

Hence, the elements of $\mathcal{B}_n^\Box$ are block vectors.
This means that for an individual approximation $x^{(j)}$ holds

$$x^{(j)}_n \in x^{(j)}_0 + B_n(A, r_0),$$

(6)

where

$$B_n :\equiv B_n(A, r_0) :\equiv K^{(1)}_n + \cdots + K^{(s)}_n,$$

(7)

with the $s$ “usual” Krylov spaces for the $s$ systems,

$$K^{(j)}_n :\equiv K_n(A, r^{(j)}_0) :\equiv \left\{ \sum_{k=0}^{n-1} A^k r^{(j)}_0 \beta_{k,j}; \ \beta_{k,j} \in \mathbb{C} \ (\forall k) \right\}. \quad (8)$$

In other words, each approximation $x^{(j)}$ is from a space that is as large as all $s$ “usual” Krylov spaces together: $\dim B_n \leq ns$.

$B^\square_n$ is a Cartesian product of $s$ copies of $B_n$:

$$B^\square_n = B_n \times \cdots \times B_n \text{.}$$
This means that for an individual approximation \( x^{(j)} \) holds

\[
x_n^{(j)} \in x_0^{(j)} + B_n(A, r_0),
\]

where

\[
B_n \equiv B_n(A, r_0) \equiv K_n^{(1)} + \cdots + K_n^{(s)},
\]

with the \( s \) “usual” Krylov spaces for the \( s \) systems,

\[
K_n^{(j)} \equiv K_n(A, r_0^{(j)}) \equiv \left\{ \sum_{k=0}^{n-1} A^k r_0^{(j)} \beta_{k,j}; \ \beta_{k,j} \in \mathbb{C} \ (\forall k) \right\}.
\]

In other words, each approximation \( x^{(j)} \) is from a space that is as large as all \( s \) “usual” Krylov spaces together: \( \dim B_n \leq ns \).

\( B_n^{\square} \) is a Cartesian product of \( s \) copies of \( B_n \):

\[
B_n^{\square} = B_n \times \cdots \times B_n.
\]
This means that for an individual approximation $x^{(i)}$ holds

$$x^{(i)}_n \in x^{(i)}_0 + B_n(A, r_0),$$

where

$$B_n \equiv B_n(A, r_0) \equiv \mathcal{K}_n^{(1)} + \cdots + \mathcal{K}_n^{(s)},$$

with the $s$ “usual” Krylov spaces for the $s$ systems,

$$\mathcal{K}_n^{(j)} \equiv \mathcal{K}_n(A, r_0^{(j)}) \equiv \left\{ \sum_{k=0}^{n-1} A^k r_0^{(j)} \beta_{k,j}; \beta_{k,j} \in \mathbb{C} \text{ (}\forall k\text{)} \right\}. \quad (8)$$

In other words, each approximation $x^{(i)}$ is from a space that is as large as all $s$ “usual” Krylov spaces together: $\dim B_n \leq ns$.

$B_n^{\Box}$ is a Cartesian product of $s$ copies of $B_n$:

$$B_n^{\Box} = B_n \times \cdots \times B_n.$$
Main reasons for using block Krylov spaces:

- The search space for each $x^{(j)}$ is much bigger, namely as big as all $s$ Krylov spaces together.
  
  *But do these extra dimensions really help much?*

- In some implementations, $s$ matrix-vector products with $A$ can be computed at once, and this is much faster than $s$ separate matrix-vector products, even on sequential computers (due to better usage of cached data).

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

Nearly all work for nonsymmetric matrices since 1990.

There are on the order of 100 publications on block methods.
Main reasons for using block Krylov spaces:

- The search space for each $x^{(j)}$ is much bigger, namely as big as all $s$ Krylov spaces together.  
  \textit{But do these extra dimensions really help much?}

- In some implementations, $s$ matrix-vector products with $A$ can be computed at once, and this is much faster than $s$ separate matrix-vector products, even on sequential computers (due to better usage of cached data).

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.  
There are on the order of 100 publications on block methods.
Deflation

The extra challenge comes from the possible linear dependence of the residuals (of the \( s \) systems).

In most block methods such a dependence requires an explicit reduction of the number of RHSs. We call this deflation.

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

In the literature on block methods deflation is only treated in a few papers, and there are hardly any investigations about its necessity and its effects.

Deflation may be possible at startup or in a later step.

In particular: when “one of the systems converges”.
Actually: when “a linear combination of the \( s \) systems converges”.
The extra challenge comes from the possible linear dependence of the residuals (of the $s$ systems).

In most block methods such a dependence requires an explicit reduction of the number of RHSs. We call this deflation. (The term “deflation” is also used with different meanings.)

In the literature on block methods deflation is only treated in a few papers, and there are hardly any investigations about its necessity and its effects.

Deflation may be possible at startup or in a later step.

In particular: when “one of the systems converges”.
Actually: when “a linear combination of the $s$ systems converges”.
EXAMPLES (of extreme cases)

1. $r_0$ is made up of $s$ identical vectors $r$,

$$r_0 := \begin{pmatrix} r & r & r & \ldots & r \end{pmatrix}.$$

These might come from different $b^{(i)}$ and suitably chosen $x^{(i)}$:

$$r = b^{(i)} - Ax^{(i)} \quad (i = 1, \ldots, s)$$

Here, it suffices to solve one system.

2. $r_0 := \begin{pmatrix} r & Ar & A^2r & \ldots & A^{s-1}r \end{pmatrix}$.

Here, even if $\text{rank } r_0 = s$, still $\text{rank } (r_0 \quad Ar_0) \leq s + 1$.

3. $r_0$ has $s$ columns that are linear combinations of $s$ eigenvectors of $A$. Then $\text{rank } (r_0 \quad Ar_0) \leq s$.

Hence, one block iteration is enough to solve all systems. A non-block solver may require $s^2$ iterations.
**EXAMPLES (of extreme cases)**

1. \( r_0 \) is made up of \( s \) identical vectors \( r \),

\[
    r_0 := \begin{pmatrix} r & r & r & \ldots & r \end{pmatrix}.
\]

These might come from different \( b^{(i)} \) and suitably chosen \( x_0^{(i)} \):

\[
    r = b^{(i)} - Ax_0^{(i)} \quad (i = 1, \ldots, s)
\]

Here, it suffices to solve one system.

2. \( r_0 := \begin{pmatrix} r & Ar & A^2r & \ldots & A^{s-1}r \end{pmatrix} \).

Here, even if \( \text{rank } r_0 = s \), still

\[
    \text{rank } \begin{pmatrix} r_0 & Ar_0 \end{pmatrix} \leq s + 1.
\]

3. \( r_0 \) has \( s \) columns that are linear combinations of \( s \) eigenvectors of \( A \). Then

\[
    \text{rank } \begin{pmatrix} r_0 & Ar_0 \end{pmatrix} \leq s.
\]

Hence, one block iteration is enough to solve all systems. A non-block solver may require \( s^2 \) iterations.
EXAMPLES (of extreme cases)

1. $r_0$ is made up of $s$ identical vectors $r$,

$$r_0 := \begin{pmatrix} r & r & r & \ldots & r \end{pmatrix}.$$

These might come from different $b^{(i)}$ and suitably chosen $x_0^{(i)}$:

$$r = b^{(i)} - Ax_0^{(i)} \quad (i = 1, \ldots, s)$$

Here, it suffices to solve one system.

2. $r_0 := \begin{pmatrix} r & Ar & A^2r & \ldots & A^{s-1}r \end{pmatrix}$.

Here, even if $\text{rank } r_0 = s$, still $\text{rank } (r_0 \ A r_0) \leq s + 1$.

3. $r_0$ has $s$ columns that are linear combinations of $s$ eigenvectors of $A$. Then $\text{rank } (r_0 \ A r_0) \leq s$.

Hence, one block iteration is enough to solve all systems. A non-block solver may require $s^2$ iterations.
Recall from *single RHS case* \((s = 1)\):

Characteristic properties of grade \(\bar{\nu}(y, A)\) of \(y\) with resp. to \(A\):

\[
\dim \mathcal{K}_n(A, y) = \begin{cases} 
  n & \text{if } n \leq \bar{\nu}, \\
  \bar{\nu} & \text{if } n \geq \bar{\nu}; 
\end{cases}
\]

\[
\bar{\nu} = \min \{ n \mid \dim \mathcal{K}_n(A, y) = \dim \mathcal{K}_{n+1}(A, y) \} ;
\]

\[
\bar{\nu} = \min \{ n \mid A^{-1}y \in \mathcal{K}_n(A, y) \} \leq \partial \hat{\chi}_A,
\]

where \(\partial \hat{\chi}_A \equiv \) degree of minimal polynomial of \(A\);

\[
\bar{\nu} = \min \{ n \mid x_\star \in x_0 + \mathcal{K}_n(A, r_0) \} ,
\]

where \(Ax_\star = b,\ r_0 \equiv b - Ax_0\).
Recall from single RHS case ($s = 1$):

Characteristic properties of grade $\bar{\nu}(y, A)$ of $y$ with respect to $A$:

- $\dim \mathcal{K}_n(A, y) = \begin{cases} n & \text{if } n \leq \bar{\nu}, \\ \bar{\nu} & \text{if } n \geq \bar{\nu} \end{cases}$

- $\bar{\nu} = \min \{ n \mid \dim \mathcal{K}_n(A, y) = \dim \mathcal{K}_{n+1}(A, y) \}$

- $\bar{\nu} = \min \{ n \mid A^{-1}y \in \mathcal{K}_n(A, y) \} \leq \partial \hat{\chi}_A,$

where $\partial \hat{\chi}_A \equiv$ degree of minimal polynomial of $A$;

- $\bar{\nu} = \min \{ n \mid x_* \in x_0 + \mathcal{K}_n(A, r_0) \}$,

where $Ax_* = b$, $r_0 :\equiv b - Ax_0$. 

Martin H. Gutknecht  
Block Krylov Space Solvers: a Survey
Recall from single RHS case \((s = 1)\):

Characteristic properties of grade \(\bar{\nu}(y, A)\) of \(y\) with resp. to \(A\):

\[
\dim \mathcal{K}_n(A, y) = \begin{cases} 
  n & \text{if } n \leq \bar{\nu}, \\
  \bar{\nu} & \text{if } n \geq \bar{\nu}; 
\end{cases}
\]

\[
\bar{\nu} = \min \left\{ n \mid \dim \mathcal{K}_n(A, y) = \dim \mathcal{K}_{n+1}(A, y) \right\};
\]

\[
\bar{\nu} = \min \left\{ n \mid A^{-1}y \in \mathcal{K}_n(A, y) \right\} \leq \partial \hat{\chi}_A,
\]

where \(\partial \hat{\chi}_A \equiv \text{degree of minimal polynomial of } A;\)

\[
\bar{\nu} = \min \left\{ n \mid x_\star \in x_0 + \mathcal{K}_n(A, r_0) \right\},
\]

where \(Ax_\star = b, \ r_0 \equiv b - Ax_0.\)
Recall from *single RHS case* \((s = 1)\):

Characteristic properties of grade \(\bar{\nu}(y, A)\) of \(y\) with resp. to \(A\):

\[
\dim \mathcal{K}_n(A, y) = \begin{cases} 
n & \text{if } n \leq \bar{\nu}, \\
\bar{\nu} & \text{if } n \geq \bar{\nu}; 
\end{cases}
\]

\[
\bar{\nu} = \min \left\{ n \mid \dim \mathcal{K}_n(A, y) = \dim \mathcal{K}_{n+1}(A, y) \right\};
\]

\[
\bar{\nu} = \min \left\{ n \mid A^{-1}y \in \mathcal{K}_n(A, y) \right\} \leq \partial \hat{\chi}_A,
\]

where \(\partial \hat{\chi}_A\) :\(\equiv\) degree of minimal polynomial of \(A\);

\[
\bar{\nu} = \min \left\{ n \mid x_* \in x_0 + \mathcal{K}_n(A, r_0) \right\},
\]

where \(Ax_* = b\), \(r_0 :\equiv b - Ax_0\).
The block grade

In *multiple RHS case* \((s > 1)\):

Introduce **block grade** \(\bar{\nu}(y, A)\) of \(y\) with respect to \(A\) with characteristic properties:

\[
\bar{\nu} = \min \left\{ n \mid \dim B_n(A, y) = \dim B_{n+1}(A, y) \right\} ;
\]

\[
\bar{\nu} = \min \left\{ n \mid A^{-1}y \in B_n(A, y) \right\} \leq \partial \hat{\chi}_A,
\]

where \(\partial \hat{\chi}_A\) :\(\equiv\) degree of minimal polynomial of \(A\);

\[
\bar{\nu} = \min \left\{ n \mid x_\star \in x_0 + B_n(A, r_0) \right\} ,
\]

where \(Ax_\star = b,\ r_0 :\equiv b - Ax_0\).
In multiple RHS case \((s > 1)\):

Introduce **block grade** \(\bar{\nu}(y, A)\) of \(y\) with respect to \(A\) with characteristic properties:

\[
\bar{\nu} = \min \left\{ n \mid \dim B_n(A, y) = \dim B_{n+1}(A, y) \right\} ;
\]

\[
\bar{\nu} = \min \left\{ n \mid A^{-1} y \in B_n(A, y) \right\} \leq \partial \hat{\chi}_A,
\]

where \(\partial \hat{\chi}_A :\equiv\) degree of minimal polynomial of \(A\);

\[
\bar{\nu} = \min \left\{ n \mid x_\star \in x_0 + B_n(A, r_0) \right\} ,
\]

where \(Ax_\star = b,\ r_0 :\equiv b - Ax_0\).
The block grade

In *multiple RHS case* $(s > 1)$:

Introduce **block grade** $\bar{\nu}(y, A)$ of $y$ with respect to $A$ with characteristic properties:

- $\bar{\nu} = \min \{ n \mid \dim B_n(A, y) = \dim B_{n+1}(A, y) \}$;

- $\bar{\nu} = \min \{ n \mid A^{-1}y \in B_n(A, y) \} \leq \partial \lambda_A$,

where $\partial \lambda_A \equiv \text{degree of minimal polynomial of } A$;

- $\bar{\nu} = \min \{ n \mid x_\star \in x_0 + B_n(A, r_0) \}$,

where $Ax_\star = b$, $r_0 \equiv b - Ax_0$. 
In the single RHS case, in exact arithmetic, computing $x_\star$ requires
\[
\dim K_{\bar{\nu}} = \bar{\nu} \quad \text{MVs.}
\]

In the multiple RHS case, in exact arithmetic, computing $x_\star$ requires
\[
\dim B_{\bar{\nu}} \in [\bar{\nu}, s \cdot \bar{\nu}] \quad \text{MVs.}
\]

This is a big interval!

**Block methods are most effective (compared to single RHS methods) if**
\[
\dim B_{\bar{\nu}} \ll s \cdot \bar{\nu}.
\]

**More exactly: block methods are most effective if**
\[
\dim B_{\bar{\nu}}(r_0, A) \ll \sum_{k=1}^{s} \dim K_{\bar{\nu}}(r_0^{(k)}, A).
\]
In other words: **block methods are most effective (compared to single RHS methods) if deflation is possible and used!**

However, exact deflation is rare, and we need approximate deflation depending on a **deflation tolerance** in RRQR.

Approximate deflation introduces a **deflation error**.

The deflation error may deteriorate the convergence speed and/or the accuracy of the computed solution.

Restarting the iteration can be useful from this point of view.
In the 1970ies a number of people started around the same time with block Lanczos for symmetric EVal problems. It is hard to tell now who had the idea first.

**Cullum/Donath** [[IEEE Decision Control’’74], [’74] (symmetric, EV)

**Kahan/Parlett** [Sparse Matrix Comp.’’76] (symmetric, EV)

**Underwood** [’75Diss] (symmetric, EV + CG)

**Golub/Underwood** [Math. Software’’77] (symmetric, EV)

**Lewis** [’77Diss] (symmetric)

**Cullum** [’78BIT] (symmetric, EV)
Algorithm (Symmetric Block Lanczos Algorithm)

Start: Given \( \tilde{y}_0 \in \mathbb{C}^{N \times s} \) let

\[
\begin{align*}
    y_0 \rho_0 &:= \tilde{y}_0 \quad \text{(QR factorization: } \rho_0 \in \mathbb{C}^{s \times s}, \ y_0 \in \mathbb{C}^{N \times s}) \\
\end{align*}
\]

Loop:

\[
\begin{align*}
    \text{for } n = 1, 2, \ldots \text{ do} \\
    \tilde{y} &:= Ay_{n-1} \quad \text{(s MVs in parallel)} \\
    \tilde{y} &:= \tilde{y} - y_{n-2} \beta_{n-2}^* \quad \text{if } n > 1 \quad \text{(s² SAXPYs in parallel)} \\
    \alpha_{n-1} &:= y_{n-1}^* \tilde{y} \quad \text{(s² SDOTs in parallel)} \\
    \tilde{y} &:= \tilde{y} - y_{n-1} \alpha_{n-1} \quad \text{(s² SAXPYs in parallel)} \\
    y_n \beta_{n-1} &:= \tilde{y} \quad \text{(QR factorization: } \beta_{n-1} \in \mathbb{C}^{s \times s}) \\
    \text{end}
\end{align*}
\]

Need to add stopping criterion and deflation.
Deflation (not \[?\] treated in old papers): We apply in both

\[
\begin{align*}
\mathbf{y}_0 & \quad \mathbf{\rho}_0 := \tilde{\mathbf{y}}_0 & \quad \text{and} & \quad \mathbf{y}_n & \quad \mathbf{\beta}_{n-1} := \tilde{\mathbf{y}} \\
\mathbf{Q} \quad \mathbf{R} & & \mathbf{Q} \quad \mathbf{R}
\end{align*}
\]

a (high) rank-revealing QR factorization (RRQR).

Columns in \(\mathbf{y}_0\) or \(\mathbf{y}_n\) that are multiplied only with small elements of \(\mathbf{\rho}_0\) or \(\mathbf{\eta}_{n,n-1}\), respectively, can be deleted \(\rightsquigarrow\) deflation.

\(s\) is replaced by \(s_n\), where \(s \geq s_0 \geq s_1 \geq \ldots\).

Two types: initial deflation and Lanczos deflation.

\(\mathbf{\rho}_0\) and \(\mathbf{\beta}_{n-1}\) are upper triangular up to a column permutation.

In case of deflation \(\mathbf{\rho}_0\) and \(\mathbf{\beta}_{n-1}\) are (nearly) singular.
Deflation (not treated in old papers): We apply in both

\[ y_0 \begin{bmatrix} Q \\ R \end{bmatrix} \rho_0 := \tilde{y}_0 \quad \text{and} \quad y_n \begin{bmatrix} Q \\ R \end{bmatrix} \beta_{n-1} := \tilde{y} \]

a (high) rank-revealing QR factorization (RRQR).

Columns in \( y_0 \) or \( y_n \) that are multiplied only with small elements of \( \rho_0 \) or \( \eta_{n,n-1} \), respectively, can be deleted \( \sim \) deflation.

\( s \) is replaced by \( s_n \), where \( s \geq s_0 \geq s_1 \geq \ldots \).

Two types: initial deflation and Lanczos deflation.

\( \rho_0 \) and \( \beta_{n-1} \) are upper triangular up to a column permutation.

In case of deflation \( \rho_0 \) and \( \beta_{n-1} \) are (nearly) singular.
Deflation (not [?] treated in old papers): We apply in both

\[
\begin{align*}
\begin{bmatrix} y_0 \\ Q \end{bmatrix} \begin{bmatrix} \rho_0 \\ R \end{bmatrix} & := \begin{bmatrix} \tilde{y}_0 \\ \end{bmatrix} \quad \text{and} \quad \begin{bmatrix} y_n \\ Q \end{bmatrix} \begin{bmatrix} \beta_{n-1} \\ R \end{bmatrix} & := \begin{bmatrix} \tilde{y} \\ \end{bmatrix}
\end{align*}
\]

a (high) rank-revealing QR factorization (RRQR).

Columns in \( y_0 \) or \( y_n \) that are multiplied only with small elements of \( \rho_0 \) or \( \eta_{n,n-1} \), respectively, can be deleted \( \sim \) deflation.

\( s \) is replaced by \( s_n \), where \( s \geq s_0 \geq s_1 \geq \ldots \).

Two types: initial deflation and Lanczos deflation.

\( \rho_0 \) and \( \beta_{n-1} \) are upper triangular up to a column permutation.

In case of deflation \( \rho_0 \) and \( \beta_{n-1} \) are (nearly) singular.
HHQR for Lanczos deflation in detail:

\[
\tilde{y} =: \begin{pmatrix} y_n & y_n^\Delta \end{pmatrix} \begin{pmatrix} \rho_n & \rho_n^\boxtimes \\ \mathbf{0} & \rho_n^\Delta \end{pmatrix} \pi_n^T =: \begin{pmatrix} y_n & y_n^\Delta \end{pmatrix} \begin{pmatrix} \beta_{n-1} \\ \beta_{n-1}^\Delta \end{pmatrix},
\]

where:
- \(\pi_n\) is an \(s_{n-1} \times s_{n-1}\) permutation matrix,
- \(y_n\) is an \(N \times s_n\) block vector with full numerical column rank, which goes into the basis,
- \(y_n^\Delta\) is an \(N \times (s_{n-1} - s_n)\) matrix that will be deflated (deleted),
- \(\rho_n\) is an \(s_n \times s_n\) upper triangular, nonsingular matrix,
- \(\rho_n^\boxtimes\) is an \(s_n \times (s_{n-1} - s_n)\) matrix,
- \(\rho_n^\Delta\) is an upper triangular \((s_{n-1} - s_n) \times (s_{n-1} - s_n)\) matrix with \(\|\rho_n^\Delta\|_F = O(\sigma_{s_{n+1}})\), where \(\sigma_{s_{n+1}}\) is the largest singular value of \(\tilde{y}\) smaller or equal to tol.
The fundamental block Lanczos relation $AY_m = Y_{m+1}T_m$ (with a block tridiagonal matrix $T_m$ extended at the bottom with $s_m$ rows) is in case of inexact deflation replaced by

$$AY_m = Y_{m+1}T_m + Y_{m+1}^\Delta T_{m}^\Delta,$$

where

$$T_{m}^\Delta \equiv \begin{pmatrix} o & o & \cdots & o \\ \beta^\Delta_0 & o & \cdots & o \\ \beta^\Delta_1 & \ddots & \ddots \\ \vdots & \ddots & o \\ \beta^\Delta_{m-1} & \end{pmatrix}$$

is $(s - s_m) \times t_{m-1}$, where $t_m \equiv \sum_{k=0}^{m} s_k$.

*Is deflation important? YES!*
Symmetric block Lanczos with deflation (cont’d)

The fundamental block Lanczos relation $A Y_m = Y_{m+1} T_m$ (with a block tridiagonal matrix $T_m$ extended at the bottom with $s_m$ rows) is in case of inexact deflation replaced by

$$A Y_m = Y_{m+1} T_m + Y^\Delta_{m+1} T^\Delta_m,$$

where

$$T^\Delta_m = \begin{pmatrix}
o & o & \cdots & o \\
o & o & \cdots & o \\
o & o & \cdots & o \\
o & o & \cdots & o \\
o & o & \cdots & o \\
o & o & \cdots & o \\
o & o & \cdots & o \\
\end{pmatrix}$$

is $(s - s_m) \times t_{m-1}$, where $t_m :\equiv \sum_{k=0}^{m} s_k$.

Is deflation important? YES!
The fundamental block Lanczos relation $AY_m = Y_{m+1}T_m$ (with a block tridiagonal matrix $T_m$ extended at the bottom with $s_m$ rows) is in case of inexact deflation replaced by

$$AY_m = Y_{m+1}T_m + Y_{m+1}^\Delta T_m^\Delta,$$

where

$$T_m^\Delta := \begin{pmatrix}
0 & 0 & \cdots & 0 \\
\beta^\Delta_0 & 0 & \cdots & 0 \\
\beta^\Delta_1 & \ddots & \ddots & 0 \\
\vdots & \ddots & \ddots & 0 \\
\beta^\Delta_{m-1} & & & 
\end{pmatrix}$$

is $(s - s_m) \times t_{m-1}$, where $t_m := \sum_{k=0}^{m} s_k$.

Is deflation important? YES!
**EXPERIMENT (1)**

$A$ is a sparse $100 \times 100$ random matrix. In the block vector $\tilde{y}_0$ each of the first two columns is a random linear combination of 20 distinct eigenvectors of $A$. The third column is a linear combination of 5 other eigenvectors. Hence these 45 eigenvectors are an orthonormal basis for the $A$-invariant subspace

$$B_{20} (A, \tilde{y}_0) = \mathcal{K}_{20} (A, \tilde{y}_0^{(1)}) \oplus \mathcal{K}_{20} (A, \tilde{y}_0^{(2)}) \oplus \mathcal{K}_5 (A, \tilde{y}_0^{(3)}).$$

Constructing $y_0, \ldots, y_4$ we expect no problems. However, the Krylov subspace $\mathcal{K}_5 (A, \tilde{y}_0^{(3)})$ is exhausted. The smallest eigenvalue of $\beta_4$ is close to $10^{-10}$. Proceeding without deflation we construct a highly indetermined vector in order to complete the block vector $y_5$. 
One might hope that this vector does not disturb the Lanczos process, and that it does not influence the construction of the Krylov subspaces $\mathcal{K}_n \left( A, \tilde{y}_0^{(1)} \right)$ and $\mathcal{K}_n \left( A, \tilde{y}_0^{(2)} \right)$.

In particular one might hope that the corresponding columns in the block vector $y_6$ remain orthogonal to all previously constructed vectors.

**However, this experiment shows that the orthogonality is lost.**
The loss of orthogonality: $\log_{10}(y_n^H y_m - \delta_{nm})$

Figure: Experiment 1: The vector corresponding to a singular value of approximately $10^{-10}$ is highly indetermined. It is not orthogonal to the vectors of the previous blocks. However, it is orthogonal to the two other vectors of the block vector $y_5$. 
Figure: Experiment 1: The block vector $y_6$ is far away from being orthogonal to all previous blocks.
The loss of orthogonality: \( \log_{10} (y_n^H y_m - \delta_{nm}) \)

**Figure:** Experiment 1: Colormap of the matrix \( V = \log |Y_{20}^* Y_{20} - I_{20}|. \) Orthogonality is completely lost after ignoring the exhausted Krylov space.
O’Leary ['78/'80LAA] (*nonsym./symmetric:* BiCG/CG)
First statement of block BICG, but there is only a very short
discussion of the added problems in the nonsymmetric case.

Ruhe ['79MathComp] (*symmetric, band, EV*)
Ruhe shows that the orthonormal basis can be built up vector
by vector. He also discusses reorthogonalization: it suffices to
reorthogonalize against \( y_{n-1} \).

*However, his alg. does not allow RRQR: no pivoting possible.
Therefore less stable than our current implementation.*

Parlett ['80Book] (*symmetric, block and band, EV*)

Saad ['80SINUM] (*symmetric, EV, convergence*)

O’Leary ['87Par. Comp.] (*symmetric*)
Boley/Golub [’91 Syst. Control Lett.] (nonsymmetric, control)

Kim/Craig [’90 Int. J. Num. Meth. Eng.] (nonsymmetric, EV)


Broyden [’94/’95 Optim. Methods Softw.] (sym., indef., nonsym., look-ahead)

Grimes/Lewis/Simon [’88/’94 SIMAX] (symmetric, EV)

Kim/Chronopoulos [’92 JCAM] (nonsymmetric)

Ruiz [’92 Diss] (BI-CG and symmetric block Lanczos)

Nikishin/Yeremin [’93/’95 SIMAX] (symmetric, defl.)

First detailed treatment of deflation for CG.
Aliaga/Hernández/Boley [Lanczos/'94]
    (nonsym., look-ahead (cluster), model red.)
Bai [5th SIAM ALA/'94] (nonsym., EV, spectral trafo)
Cullum [Lanczos/'94] (symmetric, EV)
Cullum [Lanczos/'94] (nonsym., EV)
Freund [AT VIII/'95] (nonsym., band, matrix Padé)

Boyse/Seidl [’94/’96 SISC] (compl. symmetric, QMR)
Simoncini [’94/’97 SIMAX]
    (nonsym., block-2-term, band, deflation, QMR)
Ye [’94/’96 Num. Alg.] (symmetric, EV, adapt. block size)
Aliaga/Boley/Freund/Hernández ['96/'99/00MathComp] (nonsym., band, defl., look-ahead, QMR)

Bai/Day/Ye ['97/'99SIMAX] (nonsym., EV, adapt. block size, look-ahead, \(\leadsto\) ABLE)

Freund/Malhotra ['97LAA] (nonsym., band, defl., \(\leadsto\) Bl-QMR)


Freund [Systems, Control 21st Cent./’97] (nonsym., band, model red.)


Freund ['99/’00JCAM] (nonsym., band, model red.)

Freund ['99/’01JCAM] (nonsym., band, block Hankel, FOPs)
Broyden ['97 Optim. Methods Softw.] (indef. sym., nonsym.)
Broyden [Alg. large scale lin. sys. ’98] (indef. sym., look-ahead)
Dai [’98] (symmetric)
Dai [’98] (nonsymmetric)
El Guennouni/Jbilou/Sadok [’99] (nonsym.)
El Guennouni/Jbilou/Sadok [’99] (BIBiCGStab)
El Guennouni [’00Diss] (nonsym.)
El Guennouni/Jbilou [’00] (nonsym., bl/gl-BiCGStab, deflation, seed BiCGStab)
Jbilou/Sadok [’97] (nonsym., global, Lanczos-based)
Yeung/Chan [’97/’99 SISC] (nonsym., 1 eq., ML(k)BiCGStab)
Bai/Freund ['00/'01SISC] (symmetric, band, EV, model red.)
Bai/Freund ['00/'01LAA] (nonsym.?, band, Padé, model red.)
Baglama/Calvetti/Reichel [preprint]
  (nonsym., implic. restarted)
Kilmer/Miller/Rappaport ['99/'01SISC]
  (Bl-QMR combined with seeds)
Meerbergen/Scott ['00]
  (sym., EV, partial reorth., impl. restarts, ⇝ EA16)
Hsu ['03] (symmetric, EV, block size choice)
It is seemingly straightforward to define and implement block GMRES (BLGMRES), but some questions come up quickly.

- First, we apply block Arnoldi process to create an orthonormal basis of $\mathcal{B}_n(A, r_0)$.
- Then, we determine simultaneously the coordinates of the $s$ systems, i.e., solve them at once in coordinate space.
- This requires to solve a least square problem with $s$ RHSs in every iteration.
- To solve it we update the QR decomposition of a rectangular block Hessenberg matrix to which $s$ columns and rows are added in every iteration.

For block MINRES (BLMINRES) we start instead from the symmetric block Lanczos process.
Algorithm \((m\ \text{STEPS OF BLOCK ARNOLDI ALGORITHM})\)

\textbf{Start: Given} \(\tilde{y}_0 \in \mathbb{C}^{N \times s}\) \textit{let}

\[
\begin{align*}
y_0 \ \rho_0 &:= \tilde{y}_0 \quad \text{(QR factorization:} \quad \rho_0 \in \mathbb{C}^{s \times s}, \quad y_0 \in \mathbb{C}^{N \times s})
\end{align*}
\]

\textbf{Loop:}

\[
\begin{align*}
\text{for} \ n = 1 \ \text{to} \ m \ \text{do} & \\
\tilde{y} &:= Ay_{n-1} \quad (s \ \text{MV}s \ \text{in parallel}) \\
\text{for} \ k = 0 \ \text{to} \ n - 1 \ \text{do} & \\
\eta_{k,n-1} &:= y_k^* \ \tilde{y} \\
\tilde{y} &:= \tilde{y} - y_k \ \eta_{k,n-1} \quad (s^2 \ \text{SDOT}s \ \text{in parallel}) \\
\text{end} & \\
y_n \ \eta_{n,n-1} &:= \tilde{y} \quad (QR \ \text{factorization:} \quad \eta_{n,n-1} \in \mathbb{C}^{s \times s}) \\
\text{end} &
\end{align*}
\]
We apply in both
\[
\begin{align*}
\underbrace{y_0}_{Q} & \underbrace{\rho_0}_{R} \coloneqq \tilde{y}_0 \\
\underbrace{y_n}_{Q} & \underbrace{\eta_{n,n-1}}_{R} \coloneqq \tilde{y}
\end{align*}
\]
a (high) rank-revealing QR factorization (RRQR).

Columns in \(y_0\) or \(y_n\) that are multiplied only with small elements of \(\rho_0\) or \(\eta_{n,n-1}\), respectively, can be deleted \(\leadsto\) deflation.

\(s\) is replaced by \(s_n\), where \(s \geq s_0 \geq s_1 \geq \ldots\).

Two types: initial deflation and Arnoldi deflation.

\(\rho_0\) and \(\eta_{n,n-1}\) are upper triangular up to a column permutation.

In case of deflation \(\rho_0\) and \(\eta_{n,n-1}\) are (nearly) singular.
We apply in both

\[
\begin{align*}
\begin{bmatrix} y_0 \\ \rho_0 \end{bmatrix} &:= \tilde{y}_0 \\
Q & \\ R &:= \tilde{y}_0
\end{align*}
\quad \text{and} \quad
\begin{align*}
\begin{bmatrix} y_n \\ \eta_{n,n-1} \end{bmatrix} &:= \tilde{y} \\
Q & \\ R &:= \tilde{y}
\end{align*}
\]

a (high) rank-revealing QR factorization (RRQR).

Columns in \( y_0 \) or \( y_n \) that are multiplied only with small elements of \( \rho_0 \) or \( \eta_{n,n-1} \), respectively, can be deleted \( \leadsto \) deflation.

\( s \) is replaced by \( s_n \), where \( s \geq s_0 \geq s_1 \geq \ldots \).

Two types: initial deflation and Arnoldi deflation.

\( \rho_0 \) and \( \eta_{n,n-1} \) are upper triangular up to a column permutation.

In case of deflation \( \rho_0 \) and \( \eta_{n,n-1} \) are (nearly) singular.
We apply in both

\[
\begin{align*}
\mathbf{y}_0 & \quad \mathbf{Q} \rho_0 \quad \mathbf{R} := \tilde{\mathbf{y}}_0 \\
\mathbf{y}_n & \quad \mathbf{Q} \eta_{n,n-1} \quad \mathbf{R} := \tilde{\mathbf{y}}
\end{align*}
\]

a (high) rank-revealing QR factorization (RRQR).

Columns in \(\mathbf{y}_0\) or \(\mathbf{y}_n\) that are multiplied only with small elements of \(\rho_0\) or \(\eta_{n,n-1}\), respectively, can be deleted \(\leadsto\) deflation.

\(s\) is replaced by \(s_n\), where \(s \geq s_0 \geq s_1 \geq \ldots\).

Two types: initial deflation and Arnoldi deflation.

\(\rho_0\) and \(\eta_{n,n-1}\) are upper triangular up to a column permutation.

In case of deflation \(\rho_0\) and \(\eta_{n,n-1}\) are (nearly) singular.
\[ r_n = Y_{n+1} (\mathbf{e}_1 \rho_0 - H_n k_n) \equiv q_n \]

Ass.: \( H_n \) has full rank.
(This is most likely even when some \( \eta_{n,n-1} \) is singular.)

(1) **Initial deflation:**

\( r_0 \) rank-deficient \( \implies \rho_0, k_n, q_n, r_n, x_n - x_0 \) rank-def.

\( \rightsquigarrow \) initial deflation reduces \# MVs, but introduces errors if not exact.
\[ \mathbf{r}_n = \mathbf{Y}_{n+1} (\mathbf{e}_1 \rho_0 - \mathbf{H}_n \mathbf{k}_n) \equiv \mathbf{q}_n \]

**Ass.:** \( \mathbf{H}_n \) has full rank.

(This is most likely even when some \( \eta_{n,n-1} \) is singular.)

(1) **Initial deflation:**

\( \mathbf{r}_0 \) rank-deficient \( \implies \rho_0, \mathbf{k}_n, \mathbf{q}_n, \mathbf{r}_n, \mathbf{x}_n - \mathbf{x}_0 \) rank-def.

\( \sim \) initial deflation reduces \# MVs, but introduces errors if not exact.
(2) **Arnoldi deflation:** $\tilde{y}$ in block Arnoldi rank-deficient

Rather unlikely, because we start from $Ay_{n-1}$.

Unless we deflate, search space contains extra basis vectors:

$$\mathcal{R}(Y_n) \supset \mathcal{B}_n$$

But they are unlikely to help much, since the block solution lies in $x_0 + \mathcal{B}_n$ for some $n$.

$\Rightarrow$ Arnoldi deflation reduces cost (MVVs) too, but is rare; in particular if the restart period $m$ is small. 
*The block Arnoldi matrix relation is valid only with an error term.*

Hence:

- We deflate at startup and each restart if $r_0$ is rank-deficient.
- We *may* deflate in the Arnoldi process if $\tilde{y}$ is rank-deficient.
(2) **Arnoldi deflation:** \(\tilde{y}\) in block Arnoldi rank-deficient

Rather unlikely, because we start from \(Ay_{n-1}\).

Unless we deflate, search space contains extra basis vectors:

\[
\mathcal{R}(Y_n) \supsetneq B_n
\]

But they are unlikely to help much, since the block solution lies in \(x_0 + B_n\) for some \(n\).

\[\Rightarrow\] **Arnoldi deflation reduces cost (MVs) too, but is rare; in particular if the restart period \(m\) is small. The block Arnoldi matrix relation is valid only with an error term.**

Hence:
- We deflate at startup and each restart if \(r_0\) is rank-deficient.
- We *may* deflate in the Arnoldi process if \(\tilde{y}\) is rank-deficient.
(2) **Arnoldi deflation:** \( \tilde{y} \) in block Arnoldi rank-deficient

Rather unlikely, because we start from \( Ay_{n-1} \).

Unless we deflate, search space contains extra basis vectors:

\[
\mathcal{R}(Y_n) \supsetneq B_n
\]

But they are unlikely to help much, since the block solution lies in \( x_0 + B_n \) for some \( n \).

\( \Rightarrow \) **Arnoldi deflation reduces cost (MVs) too, but is rare; in particular if the restart period \( m \) is small. The block Arnoldi matrix relation is valid only with an error term.**

Hence:

- We deflate at startup and each restart if \( r_0 \) is rank-deficient.
- We *may* deflate in the Arnoldi process if \( \tilde{y} \) is rank-deficient.
**Vital** ['90\text{Diss}] (\textit{BI-GMRES})

Sadkane ['93\text{NM}] (nonsym., block Arnoldi-Chebyshev)

Sadkane ['93\text{NM}] (nonsym., block Arnoldi / Davidson, EV)

Chapman/Saad ['95/'97\text{NLAA}] (\textit{BI-GMRES, FGMRES, ...})

Jia ['94\text{Diss}] (nonsym., EV, “general. Lanczos” \supset \textit{BI-Arnoldi})

Jia ['94/'98\text{NM}] (nonsym., EV, “general. Lanczos” \supset \textit{BI-Arnoldi})

Jia ['98\text{LAA}] (nonsym., EV, \textit{BI-Arnoldi})

Jbilou ['99\text{JCAM}] (nonsym., residual smoothing)

Li ['97\text{Par. Comp.}] (\textit{parallelization of BLGMRES})

Saad ['96\text{Book}] (\textit{overview of BLGMRES versions})

Simoncini/Gallopoulos ['94/'96\text{LAA}] (BLGMRES, convergence)
Cullum/Zhang [’98/’02SIMAX]  
(two-sided BLGMRES, deflation, control, rel. to Lanczos)

El Guennouni/Jbilou/Riquet [’00/’02NumAlg] (Sylvester eq.)
Fattebert [’98/’98ETNA] (Rayleigh quot. iter., gen. EV)
Jbilou [’99JCAM] (nonsym., block smoothing)
Jbilou/Messaoudi/Sadok [’99ApNuM]  
(nonsym., global FOM/GMRES)
Langou [’03Diss] (BLGMRES)
Saad [’03Book] (overview of BLGMRES versions)
Robbé/Sadkane [’02LAA] (error bounds for BLGMRES)
Robbé/Sadkane [’02Num. Alg.]  
(BLGRES, BLFOM for Sylvester eq.)
Robbé/Sadkane [’04] (BLGMRES and BLFOM with deflation)
Schmelzer [’04Dipl] (BLMINRES and BLSYMMLQ w/deflation)
da Cunha/Becker [’05] (BLGMRES w/deflation)
6th International Congress on Industrial and Applied Mathematics

Zurich, Switzerland
16 - 20 July 2007