

Block Krylov Space Methods for Linear Systems With Multiple Right-hand Sides

Martin H. Gutknecht

Seminar for Applied Mathematics
ETH Zurich

Joint Workshop on Computational Chemistry and Numerical Analysis
Tokyo, 5/6 Dec. 2005

Joint work with Thomas Schmelzer, Oxford University

Thanks for inviting me to come here ...

... and for answering all my emails



Systems with multiple RHSs

Given is a nonsingular linear system with s RHSs,

$$\mathbf{Ax} = \mathbf{b} \quad (1)$$

where

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

Using Gauss elimination we can solve it much more efficiently than s single linear systems with different matrices, since the LU decomposition of \mathbf{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.

Systems with multiple RHSs

Given is a nonsingular linear system with s RHSs,

$$\mathbf{Ax} = \mathbf{b} \quad (1)$$

where

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

Using Gauss elimination we can solve it much more efficiently than s single linear systems with different matrices, since the LU decomposition of \mathbf{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.

Block Krylov space methods

We seek approximate solutions of the form

$$\mathbf{x}_n \in \mathbf{x}_0 + \mathcal{B}_n^\square(\mathbf{A}, \mathbf{r}_0), \quad (3)$$

where the **block Krylov space** $\mathcal{B}_n^\square := \mathcal{B}_n^\square(\mathbf{A}, \mathbf{r}_0)$ is defined by

$$\mathcal{B}_n^\square(\mathbf{A}, \mathbf{r}_0) := \text{block span}(\mathbf{r}_0, \mathbf{A}\mathbf{r}_0, \dots, \mathbf{A}^{n-1}\mathbf{r}_0) \subset \mathbb{C}^{N \times s} \quad (4)$$

$$:= \left\{ \sum_{k=0}^{n-1} \mathbf{A}^k \mathbf{r}_0 \gamma_k; \gamma_k \in \mathbb{C}^{s \times s} (k = 0, \dots, n-1) \right\}. \quad (5)$$

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

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

Block Krylov space methods

We seek approximate solutions of the form

$$\mathbf{x}_n \in \mathbf{x}_0 + \mathcal{B}_n^\square(\mathbf{A}, \mathbf{r}_0), \quad (3)$$

where the **block Krylov space** $\mathcal{B}_n^\square := \mathcal{B}_n^\square(\mathbf{A}, \mathbf{r}_0)$ is defined by

$$\mathcal{B}_n^\square(\mathbf{A}, \mathbf{r}_0) := \text{block span}(\mathbf{r}_0, \mathbf{A}\mathbf{r}_0, \dots, \mathbf{A}^{n-1}\mathbf{r}_0) \subset \mathbb{C}^{N \times s} \quad (4)$$

$$:= \left\{ \sum_{k=0}^{n-1} \mathbf{A}^k \mathbf{r}_0 \gamma_k; \gamma_k \in \mathbb{C}^{s \times s} (k = 0, \dots, n-1) \right\}. \quad (5)$$

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

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

This means that for an individual approximation $x^{(j)}$ holds

$$x_n^{(j)} \in x_0^{(j)} + \mathcal{B}_n(\mathbf{A}, \mathbf{r}_0), \quad (6)$$

where

$$\mathcal{B}_n \equiv \mathcal{B}_n(\mathbf{A}, \mathbf{r}_0) \equiv \mathcal{K}_n^{(1)} + \dots + \mathcal{K}_n^{(s)}, \quad (7)$$

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

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

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

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

$$\mathcal{B}_n^\square = \underbrace{\mathcal{B}_n \times \dots \times \mathcal{B}_n}_{s \text{ times}}.$$

This means that for an individual approximation $x^{(j)}$ holds

$$x_n^{(j)} \in x_0^{(j)} + \mathcal{B}_n(\mathbf{A}, \mathbf{r}_0), \quad (6)$$

where

$$\mathcal{B}_n \equiv \mathcal{B}_n(\mathbf{A}, \mathbf{r}_0) \equiv \mathcal{K}_n^{(1)} + \dots + \mathcal{K}_n^{(s)}, \quad (7)$$

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

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

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

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

$$\mathcal{B}_n^\square = \underbrace{\mathcal{B}_n \times \dots \times \mathcal{B}_n}_{s \text{ times}}.$$

This means that for an individual approximation $x^{(j)}$ holds

$$x_n^{(j)} \in x_0^{(j)} + \mathcal{B}_n(\mathbf{A}, \mathbf{r}_0), \quad (6)$$

where

$$\mathcal{B}_n \equiv \mathcal{B}_n(\mathbf{A}, \mathbf{r}_0) \equiv \mathcal{K}_n^{(1)} + \dots + \mathcal{K}_n^{(s)}, \quad (7)$$

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

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

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

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

$$\mathcal{B}_n^\square = \underbrace{\mathcal{B}_n \times \dots \times \mathcal{B}_n}_{s \text{ times}}.$$

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 \mathbf{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.
(Cullum/Donath [*IEEE Dec.&Control*'74], Golub/Underwood [*'75* Diss., *Math.SW*'77], Kahan/Parlett [*Sparse Matrix Comp.*'76])

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.
But do these extra dimensions really help much?
- In some implementations, s matrix-vector products with \mathbf{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.

(Cullum/Donath [*IEEE Dec.&Control*'74], Golub/Underwood [*'75*_{Diss}, *Math.SW*'77], Kahan/Parlett [*Sparse Matrix Comp.*'76])

Nearly all work for nonsymmetric matrices since 1990.

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

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 \mathbf{r}_0 is made up of s identical vectors r ,

$$\mathbf{r}_0 := (r \ r \ r \ \dots \ r) .$$

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

$$r = b^{(i)} - \mathbf{A}x_0^{(i)} \quad (i = 1, \dots, s)$$

Here, it suffices to solve one system.

- 2 $\mathbf{r}_0 := (r \ \mathbf{A}r \ \mathbf{A}^2r \ \dots \ \mathbf{A}^{s-1}r) .$

Here, even if $\text{rank } \mathbf{r}_0 = s$, still
 $\text{rank} (\mathbf{r}_0 \ \mathbf{A}\mathbf{r}_0) \leq s + 1 .$

- 3 \mathbf{r}_0 has s columns that are linear combinations of s eigenvectors of \mathbf{A} . Then $\text{rank} (\mathbf{r}_0 \ \mathbf{A}\mathbf{r}_0) \leq s$.
 Hence, one block iteration is enough to solve all systems.
 A non-block solver may require s^2 iterations.

EXAMPLES (of extreme cases)

- 1 \mathbf{r}_0 is made up of s identical vectors r ,

$$\mathbf{r}_0 := (r \ r \ r \ \dots \ r) .$$

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

$$r = b^{(i)} - \mathbf{A}x_0^{(i)} \quad (i = 1, \dots, s)$$

Here, it suffices to solve one system.

- 2 $\mathbf{r}_0 := (r \ \mathbf{A}r \ \mathbf{A}^2r \ \dots \ \mathbf{A}^{s-1}r) .$

Here, even if $\text{rank } \mathbf{r}_0 = s$, still

$$\text{rank} (\mathbf{r}_0 \ \mathbf{A}\mathbf{r}_0) \leq s + 1 .$$

- 3 \mathbf{r}_0 has s columns that are linear combinations of s eigenvectors of \mathbf{A} . Then $\text{rank} (\mathbf{r}_0 \ \mathbf{A}\mathbf{r}_0) \leq s$.
Hence, one block iteration is enough to solve all systems.

A non-block solver may require s^2 iterations.

EXAMPLES (of extreme cases)

- 1 \mathbf{r}_0 is made up of s identical vectors r ,

$$\mathbf{r}_0 := (r \ r \ r \ \dots \ r) .$$

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

$$r = b^{(i)} - \mathbf{A}x_0^{(i)} \quad (i = 1, \dots, s)$$

Here, it suffices to solve one system.

- 2 $\mathbf{r}_0 := (r \ \mathbf{A}r \ \mathbf{A}^2r \ \dots \ \mathbf{A}^{s-1}r) .$

Here, even if $\text{rank } \mathbf{r}_0 = s$, still
 $\text{rank} (\mathbf{r}_0 \ \mathbf{A}\mathbf{r}_0) \leq s + 1 .$

- 3 \mathbf{r}_0 has s columns that are linear combinations of s eigenvectors of \mathbf{A} . Then $\text{rank} (\mathbf{r}_0 \ \mathbf{A}\mathbf{r}_0) \leq s$.
 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** $\bar{\nu}(\mathbf{y}, \mathbf{A})$ of \mathbf{y} with resp. to \mathbf{A} :

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

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

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

where $\partial \widehat{\chi}_{\mathbf{A}} \equiv$ degree of minimal polynomial of \mathbf{A} ;

- $$\bar{\nu} = \min \{ n \mid \mathbf{x}_* \in \mathbf{x}_0 + \mathcal{K}_n(\mathbf{A}, \mathbf{r}_0) \},$$

where $\mathbf{A} \mathbf{x}_* = \mathbf{b}$, $\mathbf{r}_0 \equiv \mathbf{b} - \mathbf{A} \mathbf{x}_0$.

The grade

Recall from *single RHS case* ($s = 1$):

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

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

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

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

where $\partial \widehat{\chi}_{\mathbf{A}} \equiv$ degree of minimal polynomial of \mathbf{A} ;

- $$\bar{\nu} = \min \{ n \mid \mathbf{x}_* \in \mathbf{x}_0 + \mathcal{K}_n(\mathbf{A}, \mathbf{r}_0) \},$$

where $\mathbf{A} \mathbf{x}_* = \mathbf{b}$, $\mathbf{r}_0 \equiv \mathbf{b} - \mathbf{A} \mathbf{x}_0$.

The grade

Recall from *single RHS case* ($s = 1$):

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

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

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

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

where $\partial \widehat{\chi}_{\mathbf{A}} \equiv$ degree of minimal polynomial of \mathbf{A} ;

- $$\bar{\nu} = \min \{ n \mid \mathbf{x}_* \in \mathbf{x}_0 + \mathcal{K}_n(\mathbf{A}, \mathbf{r}_0) \},$$

where $\mathbf{A} \mathbf{x}_* = \mathbf{b}$, $\mathbf{r}_0 \equiv \mathbf{b} - \mathbf{A} \mathbf{x}_0$.

Recall from *single RHS case* ($s = 1$):

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

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

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

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

where $\partial \widehat{\chi}_{\mathbf{A}} \equiv$ degree of minimal polynomial of \mathbf{A} ;

- $$\bar{\nu} = \min \{ n \mid \mathbf{x}_* \in \mathbf{x}_0 + \mathcal{K}_n(\mathbf{A}, \mathbf{r}_0) \},$$

where $\mathbf{A} \mathbf{x}_* = \mathbf{b}$, $\mathbf{r}_0 \equiv \mathbf{b} - \mathbf{A} \mathbf{x}_0$.

In practice, in most problems the grade $\bar{\nu}$ is irrelevant, because $\bar{\nu}$ is large and we need convergence for $n \ll \bar{\nu}$.

There are exceptions, where $\bar{\nu}$ is small.

For such problems projection methods (CG, BICG, GMRES, ...) are very effective.

In any case, considerations about the grade can help us understand the effectiveness of Krylov methods and block Krylov methods.

To justify this, we must replace the grade by a more subtle measure that takes into account approximate solutions (Ilić/Turner [’03_{ANZIAM J.}], [’05_{NLAA}]).

In practice, in most problems the grade $\bar{\nu}$ is irrelevant, because $\bar{\nu}$ is large and we need convergence for $n \ll \bar{\nu}$.

There are exceptions, where $\bar{\nu}$ is small.

For such problems projection methods (CG, BICG, GMRES, ...) are very effective.

In any case, considerations about the grade can help us understand the effectiveness of Krylov methods and block Krylov methods.

To justify this, we must replace the grade by a more subtle measure that takes into account approximate solutions (Ilić/Turner [['03_{ANZIAM J.}](#)], [['05_{NLAA}](#)]).

The block grade

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

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

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

- $\bar{\nu} = \min \{ n \mid \mathbf{A}^{-1} \mathbf{y} \in \mathcal{B}_n^{\square}(\mathbf{A}, \mathbf{y}) \} \leq \partial \widehat{\chi}_{\mathbf{A}},$

where $\partial \widehat{\chi}_{\mathbf{A}} \equiv$ degree of minimal polynomial of \mathbf{A} ;

- $\bar{\nu} = \min \{ n \mid \mathbf{x}_{*} \in \mathbf{x}_0 + \mathcal{B}_n^{\square}(\mathbf{A}, \mathbf{r}_0) \} ,$

where $\mathbf{A} \mathbf{x}_{*} = \mathbf{b}$, $\mathbf{r}_0 \equiv \mathbf{b} - \mathbf{A} \mathbf{x}_0$.

The block grade

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

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

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

- $\bar{\nu} = \min \{ n \mid \mathbf{A}^{-1} \mathbf{y} \in \mathcal{B}_n^{\square}(\mathbf{A}, \mathbf{y}) \} \leq \partial \widehat{\chi}_{\mathbf{A}},$

where $\partial \widehat{\chi}_{\mathbf{A}} \equiv$ degree of minimal polynomial of \mathbf{A} ;

- $\bar{\nu} = \min \{ n \mid \mathbf{x}_{*} \in \mathbf{x}_0 + \mathcal{B}_n^{\square}(\mathbf{A}, \mathbf{r}_0) \} ,$

where $\mathbf{A} \mathbf{x}_{*} = \mathbf{b}$, $\mathbf{r}_0 \equiv \mathbf{b} - \mathbf{A} \mathbf{x}_0$.

The block grade

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

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

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

- $$\bar{\nu} = \min \{ n \mid \mathbf{A}^{-1} \mathbf{y} \in \mathcal{B}_n^{\square}(\mathbf{A}, \mathbf{y}) \} \leq \partial \widehat{\chi}_{\mathbf{A}},$$

where $\partial \widehat{\chi}_{\mathbf{A}} \equiv$ degree of minimal polynomial of \mathbf{A} ;

- $$\bar{\nu} = \min \{ n \mid \mathbf{x}_{\star} \in \mathbf{x}_0 + \mathcal{B}_n^{\square}(\mathbf{A}, \mathbf{r}_0) \} ,$$

where $\mathbf{A} \mathbf{x}_{\star} = \mathbf{b}$, $\mathbf{r}_0 \equiv \mathbf{b} - \mathbf{A} \mathbf{x}_0$.

In the single RHS case, in exact arithmetic, computing \mathbf{x}_* requires

$$\dim \mathcal{K}_{\bar{\nu}} = \bar{\nu} \quad \text{MVs.}$$

In the multiple RHS case, in exact arithmetic, computing \mathbf{x}_* requires

$$\dim \mathcal{B}_{\bar{\nu}} \in [\bar{\nu}, s \cdot \bar{\nu}] \quad \text{MVs.}$$

This is a big interval!

Block methods are most effective (compared to single RHS methods) if

$$\dim \mathcal{B}_{\bar{\nu}} \ll s \cdot \bar{\nu}.$$

More exactly: block methods are most effective if

$$\dim \mathcal{B}_{\bar{\nu}(\mathbf{r}_0, \mathbf{A})} \ll \sum_{k=1}^s \dim \mathcal{K}_{\bar{\nu}(r_0^{(k)}, \mathbf{A})}.$$

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.

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 $\mathcal{B}_n(\mathbf{A}, \mathbf{r}_0)$.
- Then, we determine simultaneously the coordinates of the s systems, *i.e.*, solve them at once in coordinate space.
- What about deflation? When is it required for stability? If not, when is it still worth it?
- In which ways can we profit from restarts in case of MRHSs?
- In which situations is BLGMRES more economical than GMRES?

For block MINRES (BLMINRES) we start instead from the symmetric block Lanczos process.

- In which respect does BLMINRES differ from BLGMRES?
- Are there different aspects of deflation?

Block Arnoldi/GMRES with deflation: introduction

Algorithm (m STEPS OF BLOCK ARNOLDI ALGORITHM)

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

$$\mathbf{y}_0 \boldsymbol{\rho}_0 := \tilde{\mathbf{y}}_0 \quad (\text{QR factorization: } \boldsymbol{\rho}_0 \in \mathbb{C}^{s \times s}, \quad \mathbf{y}_0 \in \mathbb{C}^{N \times s})$$

Loop:

for $n = 1$ **to** m **do**

$$\tilde{\mathbf{y}} := \mathbf{A} \mathbf{y}_{n-1} \quad (\text{s MVs in parallel})$$

for $k = 0$ **to** $n - 1$ **do** (blockwise MGS)

$$\boldsymbol{\eta}_{k,n-1} := \mathbf{y}_k^* \tilde{\mathbf{y}} \quad (\text{s}^2 \text{ SDOTs in parallel})$$

$$\tilde{\mathbf{y}} := \tilde{\mathbf{y}} - \mathbf{y}_k \boldsymbol{\eta}_{k,n-1} \quad (\text{s}^2 \text{ SAXPYs in parallel})$$

end

$$\mathbf{y}_n \boldsymbol{\eta}_{n,n-1} := \tilde{\mathbf{y}} \quad (\text{QR factorization: } \boldsymbol{\eta}_{n,n-1} \in \mathbb{C}^{s \times s})$$

end

We apply in both

$$\underbrace{\mathbf{y}_0}_{\mathbf{Q}} \underbrace{\rho_0}_{\mathbf{R}} := \tilde{\mathbf{y}}_0 \quad \text{and} \quad \underbrace{\mathbf{y}_n}_{\mathbf{Q}} \underbrace{\eta_{n,n-1}}_{\mathbf{R}} := \tilde{\mathbf{y}}$$

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

Columns in \mathbf{y}_0 or \mathbf{y}_n that are multiplied only with small elements of ρ_0 or $\eta_{n,n-1}$, respectively, can be deleted \rightsquigarrow **deflation**.

Two types: **initial deflation** and **Arnoldi deflation**.

ρ_0 and $\eta_{n,n-1}$ are upper triangular up to a column permutation.

In case of deflation ρ_0 and $\eta_{n,n-1}$ are (nearly) singular.

We apply in both

$$\underbrace{\mathbf{y}_0}_{\mathbf{Q}} \underbrace{\boldsymbol{\rho}_0}_{\mathbf{R}} := \tilde{\mathbf{y}}_0 \quad \text{and} \quad \underbrace{\mathbf{y}_n}_{\mathbf{Q}} \underbrace{\boldsymbol{\eta}_{n,n-1}}_{\mathbf{R}} := \tilde{\mathbf{y}}$$

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

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

Two types: **initial deflation** and **Arnoldi deflation**.

$\boldsymbol{\rho}_0$ and $\boldsymbol{\eta}_{n,n-1}$ are upper triangular up to a column permutation.

In case of deflation $\boldsymbol{\rho}_0$ and $\boldsymbol{\eta}_{n,n-1}$ are (nearly) singular.

We apply in both

$$\underbrace{\mathbf{y}_0}_{\mathbf{Q}} \underbrace{\boldsymbol{\rho}_0}_{\mathbf{R}} := \tilde{\mathbf{y}}_0 \quad \text{and} \quad \underbrace{\mathbf{y}_n}_{\mathbf{Q}} \underbrace{\boldsymbol{\eta}_{n,n-1}}_{\mathbf{R}} := \tilde{\mathbf{y}}$$

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

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

Two types: **initial deflation** and **Arnoldi deflation**.

$\boldsymbol{\rho}_0$ and $\boldsymbol{\eta}_{n,n-1}$ are upper triangular up to a column permutation.

In case of deflation $\boldsymbol{\rho}_0$ and $\boldsymbol{\eta}_{n,n-1}$ are (nearly) singular.

$$\mathbf{r}_n = \mathbf{Y}_{n+1} \underbrace{(\mathbf{e}_1 \rho_0 - \mathbf{H}_n \mathbf{k}_n)}_{\equiv: \mathbf{q}_n}$$

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.

\rightsquigarrow *initial deflation reduces # MVs, but introduces errors if not exact.*

$$\mathbf{r}_n = \mathbf{Y}_{n+1} \underbrace{(\mathbf{e}_1 \rho_0 - \mathbf{H}_n \mathbf{k}_n)}_{\equiv: \mathbf{q}_n}$$

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.

\rightsquigarrow *initial deflation reduces # MVs, but introduces errors if not exact.*

(2) **Arnoldi deflation:** $\tilde{\mathbf{y}}$ in block Arnoldi rank-deficient

Rather unlikely, because we start from $\mathbf{A}\mathbf{y}_{n-1}$.

Unless we deflate, search space contains extra basis vectors:

$$\mathcal{R}(\mathbf{Y}_n) \supsetneq \mathcal{B}_n$$

But they are unlikely to help much, since the block solution lies in $\mathbf{x}_0 + \mathcal{B}_n$ for some n .

↪ 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 \mathbf{r}_0 is rank-deficient.
- We *may* deflate in the Arnoldi process if $\tilde{\mathbf{y}}$ is rank-deficient.

(2) **Arnoldi deflation:** $\tilde{\mathbf{y}}$ in block Arnoldi rank-deficient

Rather unlikely, because we start from $\mathbf{A}\mathbf{y}_{n-1}$.

Unless we deflate, search space contains extra basis vectors:

$$\mathcal{R}(\mathbf{Y}_n) \supsetneq \mathcal{B}_n$$

But they are unlikely to help much, since the block solution lies in $\mathbf{x}_0 + \mathcal{B}_n$ for some n .

↪ 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 \mathbf{r}_0 is rank-deficient.
- We *may* deflate in the Arnoldi process if $\tilde{\mathbf{y}}$ is rank-deficient.

(2) **Arnoldi deflation:** $\tilde{\mathbf{y}}$ in block Arnoldi rank-deficient

Rather unlikely, because we start from $\mathbf{A}\mathbf{y}_{n-1}$.

Unless we deflate, search space contains extra basis vectors:

$$\mathcal{R}(\mathbf{Y}_n) \supsetneq \mathcal{B}_n$$

But they are unlikely to help much, since the block solution lies in $\mathbf{x}_0 + \mathcal{B}_n$ for some n .

↪ 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 \mathbf{r}_0 is rank-deficient.
- We *may* deflate in the Arnoldi process if $\tilde{\mathbf{y}}$ is rank-deficient.

Block GMRES with deflation: further remarks

Remark: Since $\tilde{\mathbf{y}}$ is a projection of the image under \mathbf{A} of the orthonormal columns of \mathbf{y}_{n-1} , it is much more unlikely that $\tilde{\mathbf{y}}$ is numerically rank-deficient than if $\tilde{\mathbf{y}}$ were the projection of $\mathbf{A}\mathbf{r}_{n-1}$, because \mathbf{r}_{n-1} may be close to rank-deficient or even exactly so.

- \rightsquigarrow *It is unlikely that $\eta_{n,n-1}$ is singular.*
- \rightsquigarrow *Arnoldi deflation is quite rare.*

Ass.: *Some $\eta_{n,n-1}$ is singular, but deflation is suppressed in Arnoldi.*

Even then it is rather unlikely \mathbf{H}_n is rank-deficient.

Conclusion: *Even when rank-deficient projections $\tilde{\mathbf{y}}$ come up in Arnoldi, the QR decomposition of \mathbf{H}_n (without deflation) is well defined, except, possibly, in very rare situations.*

- \rightsquigarrow *Even without deflation block GMRES does not break down (or, hardly ever).*

Ass.: Deflation is suppressed in Arnoldi, and $\underline{\mathbf{H}}_n$ happens to be rank-deficient (very exceptional situation).

Then the least squares problem is ill-conditioned.

The QR decomposition of $\underline{\mathbf{H}}_n$ would yield a rank-deficient \mathbf{R}_n .

But the SVD of $\underline{\mathbf{H}}_n$ would still allow us to compute the solution \mathbf{k}_n with minimum F-norm of the least-squares problems

$$\|\mathbf{q}_k\|_F = \min! \quad \text{subject to} \quad \mathbf{k}_n \in \mathbb{C}^{ns \times s}.$$

\rightsquigarrow *With some extra work, this exceptional case can be handled.*

EXPERIMENT (1)

A very simple example to test the programs:

$\mathbf{A} :=$ 5-point stencil difference method discretization of
Laplace operator on a 10×10 square
($\rightsquigarrow N = 100$).

$s := 5$

$m := 4$

Either

$\mathbf{b} := (\mathbf{e}_1 \quad \dots \quad \mathbf{e}_s)$

or

$\mathbf{b} :=$ random rank-one matrix $+ 10^{-3} \times$ random $N \times s$

matrix.

We compare:

- $\text{BlMRes}(m)$: a straightforward BLGMRES implementation using Matlab's `qr` without pivoting and with no deflation implemented.
- $\text{BlMResDefl}(m)$: BLGMRES using rank-revealing QR (the Chan/Foster high-rank-revealing QR implemented as `hrrqr` in Hansen's UTV package with tolerance set to 0.005) and deflation (both "initial" and "Arnoldi"); at each restart we check all s residuals for size and linear dependence.

We can plot:

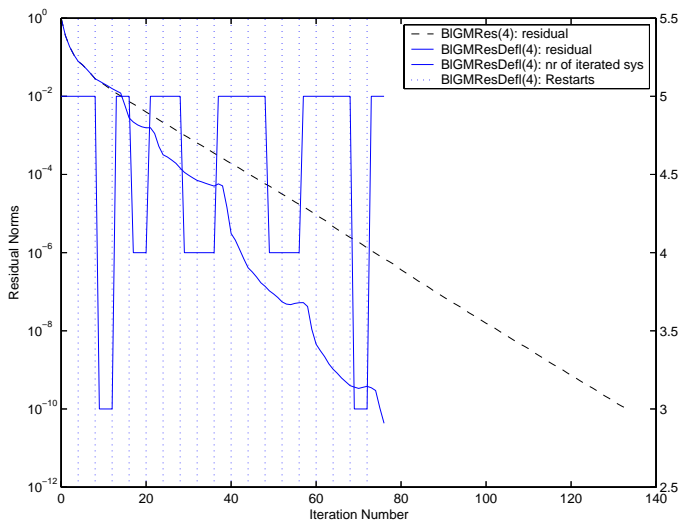
- the maximum of the 2-norms of all s residuals
- the actual number of RHSs treated (s_n)
- either iteration number n or number of MVs on x-axis

We compare:

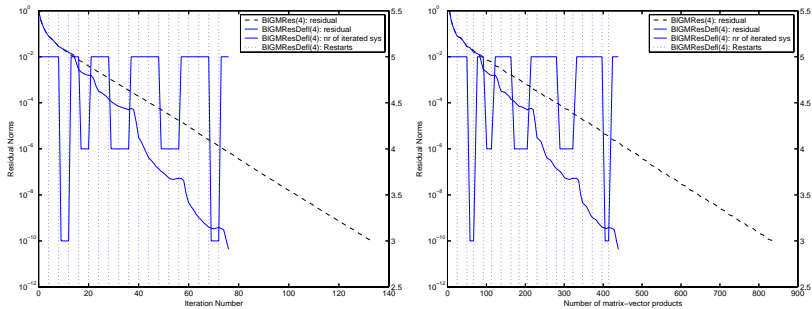
- $\text{BlMRes}(m)$: a straightforward BLGMRES implementation using Matlab's `qr` without pivoting and with no deflation implemented.
- $\text{BlMResDefl}(m)$: BLGMRES using rank-revealing QR (the Chan/Foster high-rank-revealing QR implemented as `hrrqr` in Hansen's UTV package with tolerance set to 0.005) and deflation (both "initial" and "Arnoldi"); at each restart we check all s residuals for size and linear dependence.

We can plot:

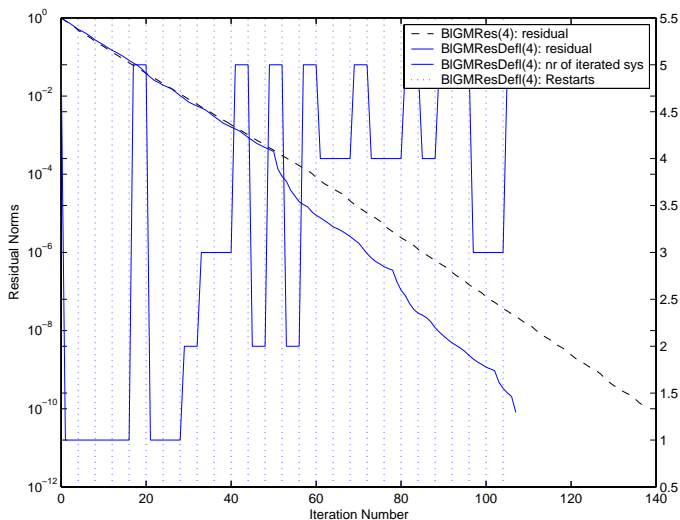
- the maximum of the 2-norms of all s residuals
- the actual number of RHSs treated (s_n)
- either iteration number n or number of MVs on x-axis



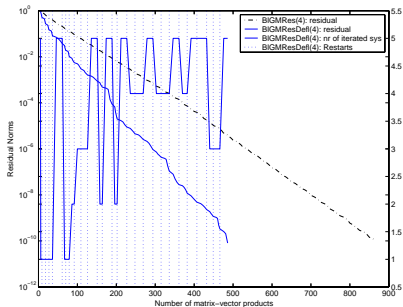
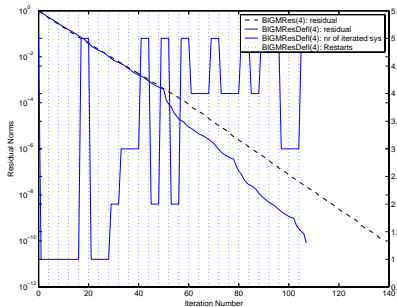
Block GMRES: Laplace on 10×10 grid ($n = 100$),
5 RHSs chosen as first 5 unit vectors



Block GMRES: Laplace on 10×10 grid ($n = 100$),
 5 RHSs chosen as first 5 unit vectors:
 iteration count vs. matrix-vector product count



Block GMRES: Laplace on 10×10 grid ($n = 100$),
5 RHSs chosen nearly linearly dependent



Block GMRES: Laplace on 10×10 grid ($n = 100$),
 5 RHSs chosen nearly linearly dependent:
 iteration count vs. matrix-vector product count

EXPERIMENT (2)

\mathbf{A} is a sparse 100×100 random matrix.

In the block vector $\tilde{\mathbf{y}}_0$ each of the first two columns is a random linear combinations of 20 distinct eigenvectors of \mathbf{A} . The third column is a linear combination of 5 other eigenvectors.

Hence these 45 eigenvectors are an orthonormal basis for the \mathbf{A} -invariant subspace

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

Constructing $\mathbf{y}_0, \dots, \mathbf{y}_4$ we expect no problems. However, the Krylov subspace $\mathcal{K}_5(\mathbf{A}, \tilde{\mathbf{y}}_0^{(3)})$ is exhausted.

The smallest eigenvalue of β_4 is close to 10^{-10} .

Proceeding without deflation we construct a highly indetermined vector in order to complete the block vector \mathbf{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 $\mathcal{K}_n(\mathbf{A}, \tilde{\mathbf{y}}_0^{(1)})$ and $\mathcal{K}_n(\mathbf{A}, \tilde{\mathbf{y}}_0^{(2)})$.

In particular one might hope that the corresponding columns in the block vector \mathbf{y}_6 remain orthogonal to all previously constructed vectors.

However, this experiment shows that the orthogonality is lost.

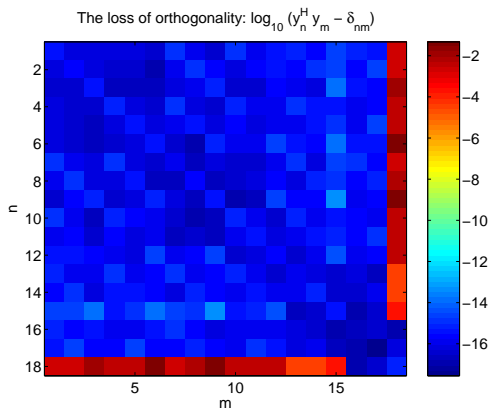


Figure: Experiment 2: The vector corresponding to a singular value of approximately 10^{-10} is highly indetermined. It is not orthogonal to the vectors of the previous blocks. However, it is orthogonal to the two other vectors of the block vector y_5 .

Symmetric Block Lanczos: experiments

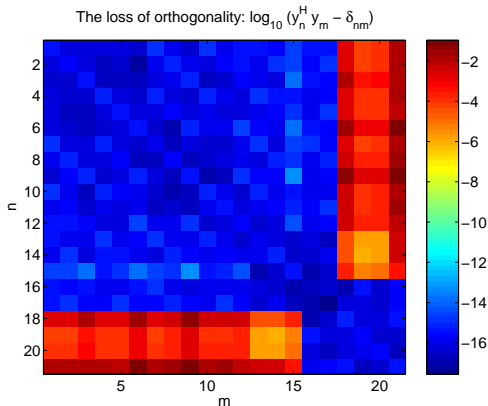


Figure: Experiment 2: The block vector \mathbf{y}_6 is far away from being orthogonal to all previous blocks.

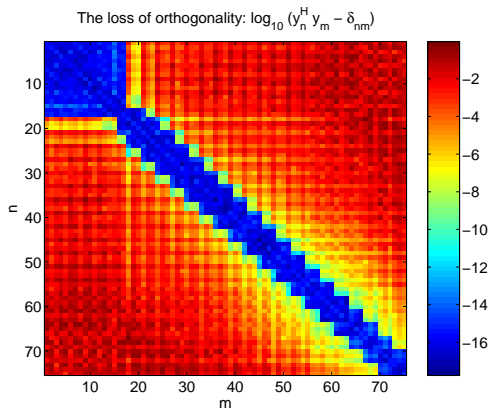


Figure: Experiment 2: Colormap of the matrix $\mathbf{V} = \log |\mathbf{Y}_{20}^* \mathbf{Y}_{20} - \mathbf{I}_{20}|$. Orthogonality is completely lost after ignoring the exhausted Krylov space.

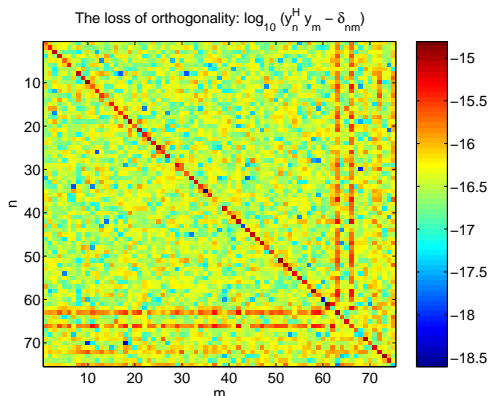


Figure: Experiment 2: Colormap of the matrix $\mathbf{V} = \log |\mathbf{Y}_{25}^* \mathbf{Y}_{25} - \mathbf{I}_{25}|$ using the block Arnoldi process without deflation.

Thanks for listening and come to ...

6th International Congress on Industrial and Applied Mathematics



www.iciam07.ch



Zurich, Switzerland
16 - 20 July 2007

