A block minimum residual norm subspace solver with partial convergence management for sequences of linear systems -IB-BGCRO-DR

Presentation · July 2022

CITATIONS
0

READS
49

1 author:

Yan-Fei Xiang
University of Strasbourg
8 PUBLICATIONS **10** CITATIONS

SEE PROFILE

Inria

A block minimum residual norm subspace solver with partial convergence management for sequences of linear systems — IB-BGCRO-DR

DD27, Minisymposium MS7_1
July 28, 2022, Prague, Czechia

Yanfei Xiang (✉: yanfei.xiang@inria.fr)
Cerfacs, Inria Bordeaux Sud-Ouest, France

Summary

01

Overview

Minimum residual norm block "Krylov" solver: **first two years** of my PhD

- Joint work with
  Luc Giraud (Inria Bordeaux - Sud-Ouest, my PhD supervisor)
  Yan-Fei Jing (UESTC)

- Outcome
  1. SIMAX (SIAM J. Matrix Anal. Appl.) paper, 2022, pp.30
     https://doi.org/10.1137/21M1401127
  2. Research report in HAL-Inria, 2022, pp.60
     https://hal.inria.fr/hal-03146213v3
  3. C++ implementation in *Fabulous*
     https://gitlab.inria.fr/solverstack/fabulous/

02

Scientific background

- Generate **Arnoldi relation** (Arnoldi algorithm):

$$AV_j = V_{j+1}\underline{H}_j, \quad \text{span}(V_{j+1}) = \text{span}(v_1, v_2, \cdots, v_{j+1}) \quad (1)$$

- Generate nested and **orthonormal basis**:

$$V_{j+1}^H V_{j+1} = I_{j+1}$$

$$\mathcal{K}_{j+1}(A, v_1) = \text{span}(r_0, Ar_0, \cdots, A^j r_0) = \text{span}(V_{j+1}) \text{ with } v_1 = \frac{r_0}{\|r_0\|_2},$$

and $\underline{H}_j$ is upper Hessenberg matrix

- Generate **Arnoldi relation** (Arnoldi algorithm):

$$AV_j = V_{j+1}\underline{H}_j, \quad \text{span}(V_{j+1}) = \text{span}(v_1, v_2, \cdots, v_{j+1}) \qquad (1)$$

- Generate nested and **orthonormal basis**:

$$V_{j+1}^H V_{j+1} = I_{j+1}$$

- Solve a small dimensional **least-squares problem** (Minimal Residual approach):

$$x_j = \underset{x \in x_0 + \mathcal{K}_j(A, v_1)}{\text{argmin}} \|b - Ax\|_2 \text{ with } x_j = x_0 + V_j y_j, \text{ thus}$$

$$\underbrace{\|b - Ax_j\|_2}_{x_j \in \mathbb{C}^n} = \left\| V_{j+1}(V_{j+1}^H r_0 - \underline{H}_j y_j) \right\|_2 = \underbrace{\left\| \beta e_1 - \underline{H}_j y_j \right\|_2}_{y_j \in \mathbb{C}^j},$$

where $j \ll n$ and

$$y_j = \underset{y \in \mathbb{C}^j}{\text{argmin}} \left\| \beta e_1 - \underline{H}_j y \right\|_2 \text{ with } \beta = \|r_0\|_2 \qquad (2)$$

Inria

- Generate **block-Arnoldi relation** (Arnoldi algorithm):

$$A\mathscr{V}_j = \mathscr{V}_{j+1}\underline{\mathscr{H}}_j, \quad \text{span}(\mathscr{V}_{j+1}) = \text{span}(V_1, V_2, \cdots, V_{j+1}) \quad (3)$$

- Generate nested and **orthonormal basis**:

$$\mathscr{V}_{j+1}^H \mathscr{V}_{j+1} = I_{n_{j+1}} \text{ with } n_{j+1} = (j+1) \times p,$$

$$\mathcal{K}_{j+1}(A, V_1) = \text{span}(R_0, AR_0, \cdots, A^j R_0) = \text{span}(\mathscr{V}_{j+1}),$$

where p is the number of right-hand sides, and $R_0 = V_1\Lambda_1$;
$\underline{\mathscr{H}}_j$ is upper Hessenberg matrix with $p \times p$ block elements

- Generate **block-Arnoldi relation** (Arnoldi algorithm):

$$A\mathscr{V}_j = \mathscr{V}_{j+1}\underline{\mathscr{H}}_j, \quad \text{span}(\mathscr{V}_{j+1}) = \text{span}(V_1, V_2, \cdots, V_{j+1}) \quad (3)$$

- Generate nested and **orthonormal basis**:

$$\mathscr{V}_{j+1}^H \mathscr{V}_{j+1} = I_{n_{j+1}} \text{ with } n_{j+1} = (j+1) \times p,$$

- Solve a small dimensional **least-squares problem** (Minimal Residual approach):

$$X_j = \underset{X \in X_0 + \mathcal{K}_j(A, V_1)}{\text{argmin}} \|B - AX\|_F \text{ with } X_j = X_0 + \mathscr{V}_j Y_j, \text{ thus}$$

$$\underbrace{\|B - AX_j\|_F}_{X_j \in \mathbb{C}^{n \times p}} = \left\| \mathscr{V}_{j+1}(\mathscr{V}_{j+1}^H R_0 - \underline{\mathscr{H}}_j Y_j) \right\|_F = \underbrace{\left\| \Lambda_j - \underline{\mathscr{H}}_j Y_j \right\|_F}_{Y_j \in \mathbb{C}^{n_j \times p}},$$

where $n_j \ll n$ and

$$Y_j = \underset{Y \in \mathbb{C}^{n_j \times p}}{\text{argmin}} \left\| \Lambda_j - \underline{\mathscr{H}}_j Y \right\|_F \text{ with } \Lambda_j = [\Lambda_1^T, 0_{p \times n_j}]^T \quad (4)$$

Inria

- **Inexact Breakdown (IB) caused by partial convergence or linear combination**

$$A\mathscr{V}_j = \left[\mathscr{V}_j, [P_{j-1}, \widetilde{W}_j]\right] \mathscr{F}_j, \tag{5}$$

then $\mathscr{V}_{j+1} = [\mathscr{V}_j, \mathbb{V}_{j+1}]$ where $\mathbb{V}_{j+1} \in \mathbb{C}^{n \times p_{j+1}} (p_{j+1} \leq p)$

> Eq. (5) comes from block-Arnoldi relation in BGMRES: $A\mathscr{V}_j = [\mathscr{V}_j, V_{j+1}]\,\underline{\mathscr{H}_j}$ with $V_{j+1} \in \mathbb{C}^{n \times p}$

> $\mathbb{V}_{j+1}$ corresponds to the components within $[P_{j-1}, \widetilde{W}_j] \in \mathbb{C}^{n \times p}$ that contribute the most to the residual norms

> P_j are the abandoned directions at iteration j that contribute the less to the residuals.

> $\mathscr{F}_j$ is no longer upper Hessenberg as soon as one IB occurs

IB-BGMRES [Robbé and Sadkane, 2006]

Inria

- **Inexact Breakdown (IB) caused by partial convergence or linear combination**

$$A\mathscr{V}_j = \left[\mathscr{V}_j, [P_{j-1}, \widetilde{W}_j]\right] \mathscr{F}_j, \tag{5}$$

then $\mathscr{V}_{j+1} = [\mathscr{V}_j, \mathbb{V}_{j+1}]$ where $\mathbb{V}_{j+1} \in \mathbb{C}^{n \times p_{j+1}} (p_{j+1} \leq p)$
IB-BGMRES [Robbé and Sadkane, 2006]

- **The growth of memory and computational requirements**
deflated restarting/subspace recycling strategy for accelerating convergence – BGMRES-DR [Morgan, 2005], IB-BGMRES-DR [Agullo et al., 2014], BGCRO-DR [Parks et al., 2016]

Bibliography (not exhaustive)

- BGMRES-DR: Restarted block-GMRES with deflation of eigenvalues. [R. B. Morgan. *Applied Numerical Mathematics*, 2005, 54:(2), pp. 222-236]
- IB-BGMRES: Exact and inexact breakdowns in the block GMRES method. [M. Robbé and M. Sadkane. *Linear Algebra and its Applications*, 2006, 419:(1), pp. 265-285]
- IB-BGMRES-DR: Block GMRES Method with Inexact Breakdowns and Deflated Restarting. [E. Agullo, L. Giraud and Y.-F. Jing. *SIAM Journal on Matrix Analysis and Applications*, 2014, 35:(4), pp. 1625-1651]
- BGCRO-DR: A block Recycled GMRES method with investigations into aspects of solver performance. [M. L. Parks, K. M. Soodhalter and D. B. Szyld. *arXiv*, 2016]

Inria

03

IB-BGCRO-DR

Motivation: devise **block solver** for **sequences of linear systems** with multiple left and right hand sides: $A^{(\ell)}X^{(\ell)} = B^{(\ell)}, \ell = 1, 2, \ldots$ families

1. **For block method**: Block GMRES method with inexact breakdown and deflated restarting – IB-BGMRES-DR

   > Combine the inexact breakdown mechanism aiming for handling rank deficiency in block iterations and the deflated restarting augmenting with some approximated eigenvectors for accelerating convergence of the next cycles.

Inria

Motivation: devise **block solver** for **sequences of linear systems** with multiple left and right hand sides: $A^{(\ell)}X^{(\ell)} = B^{(\ell)}, \ell = 1, 2, \ldots$ families

1. **For block method**: Block GMRES method with inexact breakdown and deflated restarting – IB-BGMRES-DR

2. **For sequences of linear systems**: A block recycled GMRES method– BGCRO-DR

   > Subspace recycling: accelerate convergence rate of the next cycles as well as the next families to be solved.

Inria

Motivation: devise **block solver** for **sequences of linear systems** with multiple left and right hand sides: $A^{(\ell)}X^{(\ell)} = B^{(\ell)}, \ell = 1, 2, \ldots$ families

1. **For block method**: Block GMRES method with inexact breakdown and deflated restarting – IB-BGMRES-DR

How to combine inexact breakdown mechanism and subspace recycling strategy for solving sequences of linear systems with multiple left and right-hand sides ?

2. **For sequences of linear systems**: A block recycled GMRES method– BGCRO-DR

Inria

- Generate **block-Arnoldi like relation**

$$A[U_k, \mathscr{V}_j] = \left[C_k, \mathscr{V}_j, [P_{j-1}, \widetilde{W}_j]\right] \underline{\mathscr{F}}_j = \widehat{\mathscr{V}}_{j+1} \underline{\mathscr{F}}_j, \qquad (6)$$

  - **IB**: $A\mathscr{V}_j = * + \left[\mathscr{V}_j, [P_{j-1}, \widetilde{W}_j]\right] \mathscr{F}_j$ from IB-BGMRES
  - **BGCRO-DR**: $AU_k = C_k$, $C_k^H C_k = I_k$ (computed by Harmonic-Ritz, Rayleigh-Ritz or other techniques); and $\underline{\mathscr{F}}_j = \begin{bmatrix} I_k & \mathcal{B}_j \\ 0_{(n_j+p)\times k} & \mathscr{F}_j \end{bmatrix} \in \mathbb{C}^{(k+n_j+p)\times(k+n_j)}$, $\mathcal{B}_j = C_k^H A\mathscr{V}_j \in \mathbb{C}^{k\times n_j}$

Inria

- Generate **block-Arnoldi like relation**

$$A[U_k, \mathscr{V}_j] = \left[C_k, \mathscr{V}_j, [P_{j-1}, \widetilde{W}_j]\right] \underline{\mathscr{F}}_j = \widehat{\mathscr{V}}_{j+1} \underline{\mathscr{F}}_j, \qquad (6)$$

- Generate **orthonormal basis**:

$$\widehat{\mathscr{V}}_{j+1}^H \widehat{\mathscr{V}}_{j+1} = I_{(k+n_j+p)} \text{ with } n_j = \sum_{i=1}^{j} p_i$$

Inria

- Generate **block-Arnoldi like relation**

$$A[U_k, \mathscr{V}_j] = \left[C_k, \mathscr{V}_j, [P_{j-1}, \widetilde{W}_j]\right] \underline{\mathscr{F}}_j = \widehat{\mathscr{V}}_{j+1} \underline{\mathscr{F}}_j, \qquad (6)$$

- Generate **orthonormal basis**:

$$\widehat{\mathscr{V}}_{j+1}^H \widehat{\mathscr{V}}_{j+1} = I_{(k+n_j+p)} \text{ with } n_j = \sum_{i=1}^{j} p_i$$

- Solve a small dimensional **least-squares problem**:

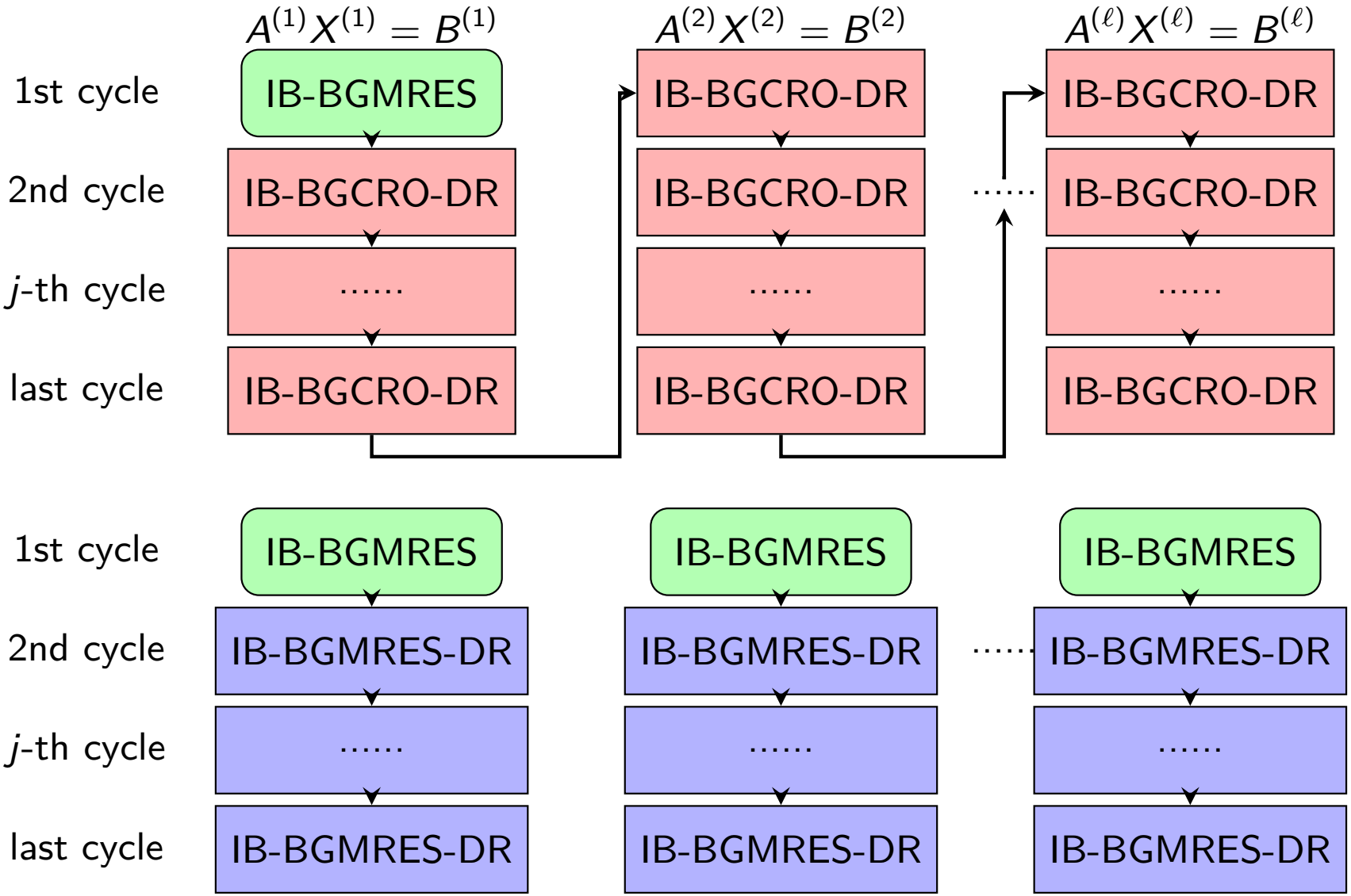
$$X_j = \underset{X \in X_0 + \text{Range}([U_k, \mathscr{V}_j])}{\text{argmin}} \|B - AX\|_F \text{ with } X_j = X_0 + [U_k, \mathscr{V}_j] Y_j,$$

where $X_j \in \mathbb{C}^{n \times p}$, $Y_j \in \mathbb{C}^{(k+n_j) \times p}$ with $k + n_j \ll n$ and

$$Y_j = \underset{Y \in \mathbb{C}^{(k+n_j) \times p}}{\text{argmin}} \left\|\Lambda_j - \underline{\mathscr{F}}_j Y\right\|_F, \quad \Lambda_j \in \mathbb{C}^{(k+n_j+p) \times p} \qquad (7)$$

Inria

Flowchart of reusing spectral information



Different search space expansion policies can be defined

- Residual based (IB-BGMRES, Robbé-Sadkane, 2006): all the residual norms will be smaller than prescribed threshold
- Backward error on b based (IB-BGMRES-DR, Agullo et al., 2014): all the backward error η_b will be smaller than prescribed threshold

$$\begin{aligned}\eta_b(x_j) &= \min_{\Delta b} \left\{\tau > 0 : \|\Delta b\|_2 \leq \tau \|b\|_2 \text{ and } Ax_j = b + \Delta b\right\} \\ &= \frac{\|Ax_j - b\|_2}{\|b\|_2}\end{aligned}$$

- Backward error on A and b based (IB-BGCRO-DR): all the backward error $\eta_{A,b}$ will be smaller than prescribed threshold

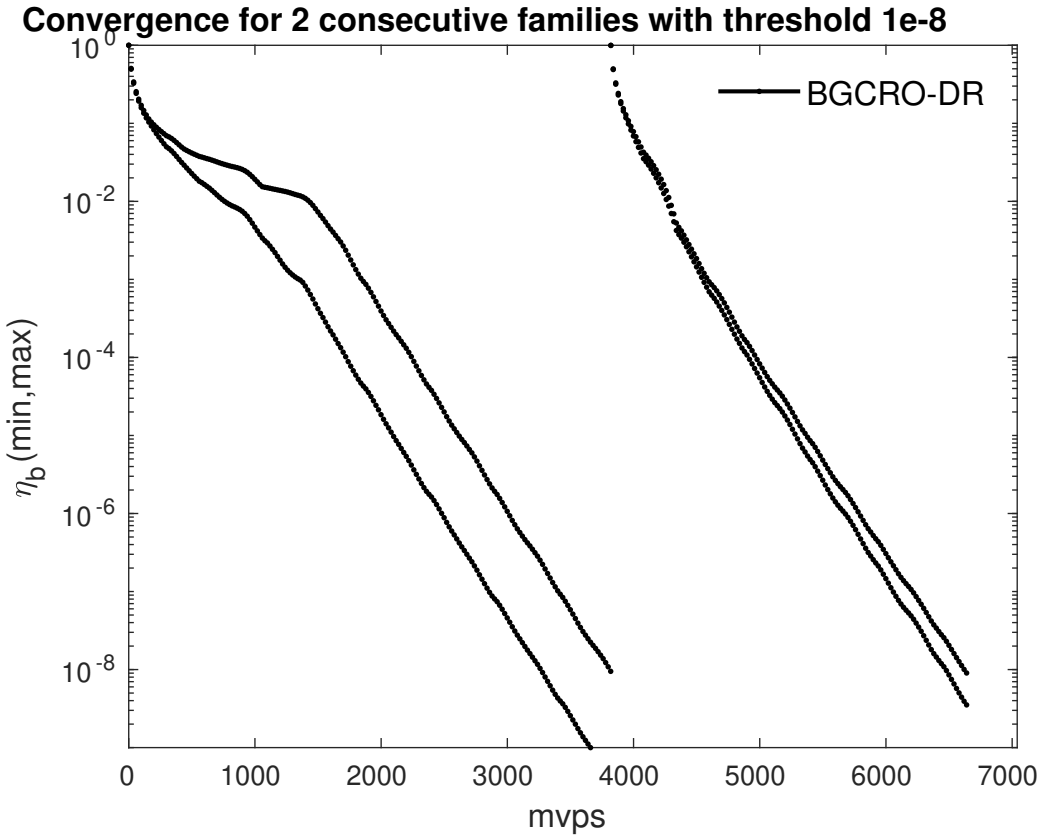
$$\begin{aligned}\eta_{A,b}(x_j) = \min_{\Delta A, \Delta b} &\left\{\tau > 0 : \|\Delta A\|_2 \leq \tau \|A\|_2 \,,\, \|\Delta b\|_2 \leq \tau \|b\|_2 \right. \\ &\left. \text{and } (A + \Delta A)x_j = b + \Delta b\right\} \\ = &\frac{\|Ax_j - b\|_2}{\|A\|_2 \|x_j\|_2 + \|b\|_2}\end{aligned}$$

Inria

04

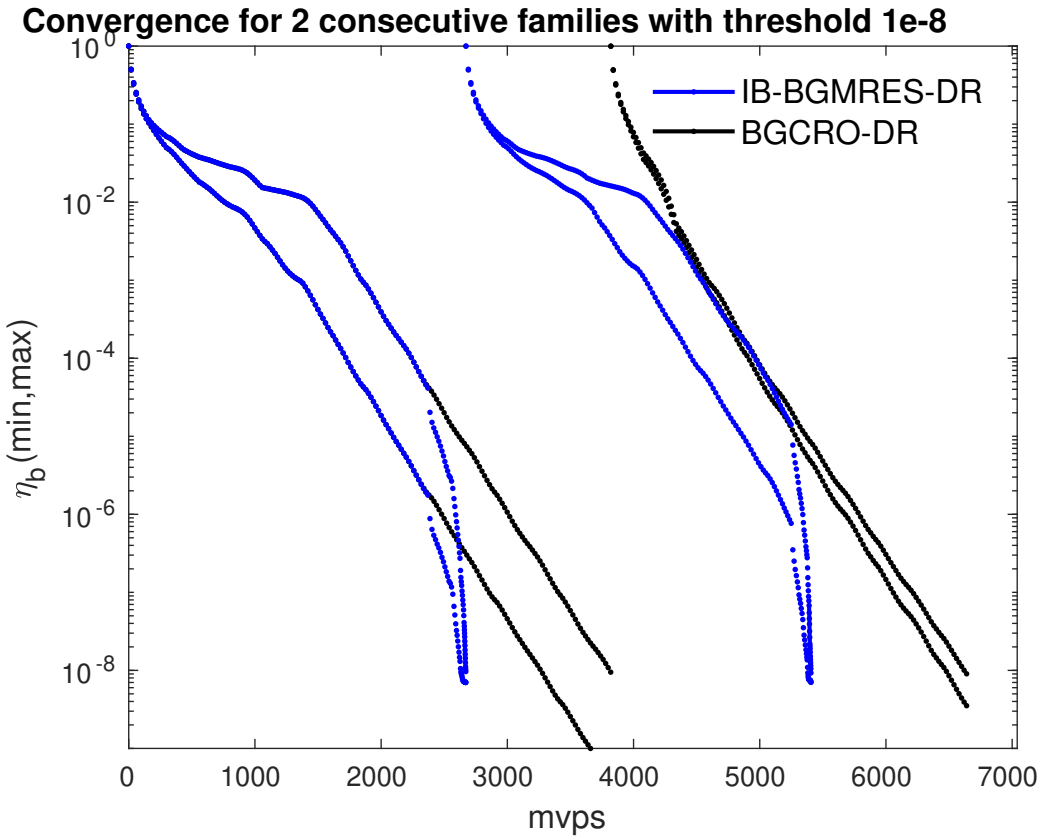
Experimental results

Benefits of IB and DR between the families



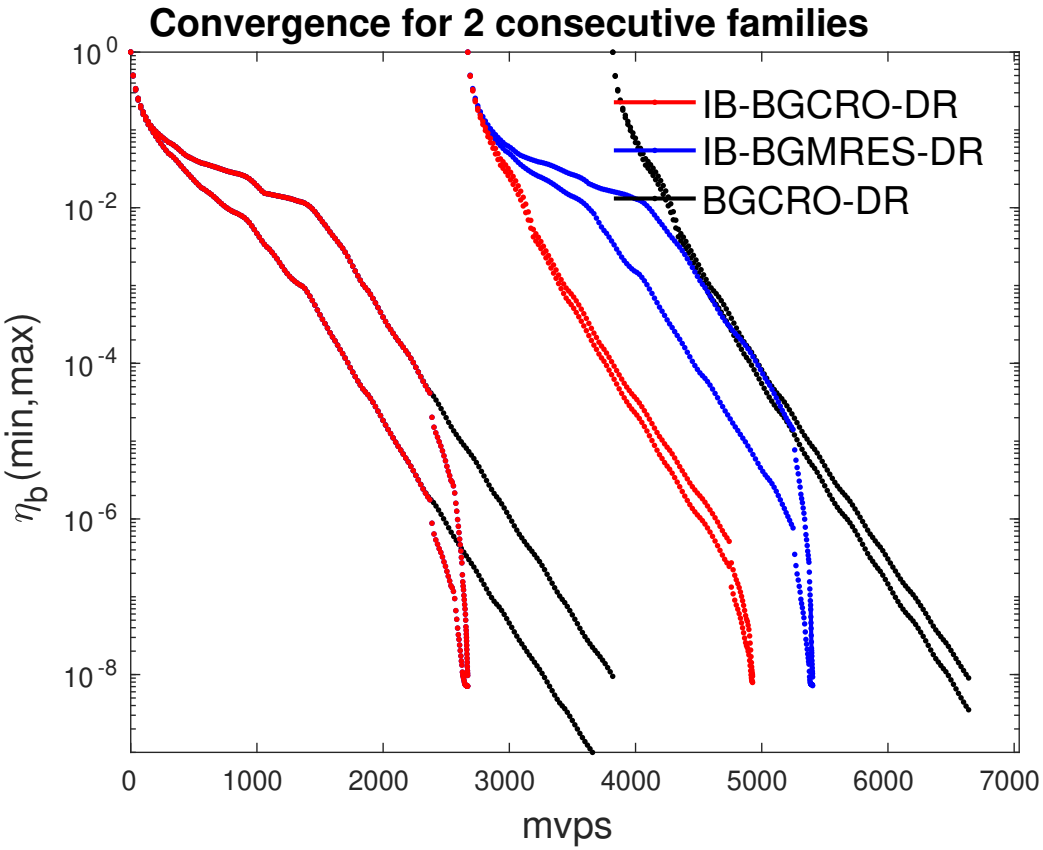
- The solution of the second family benefits from recycled information from the previous family solution

Benefits of IB and DR between the families



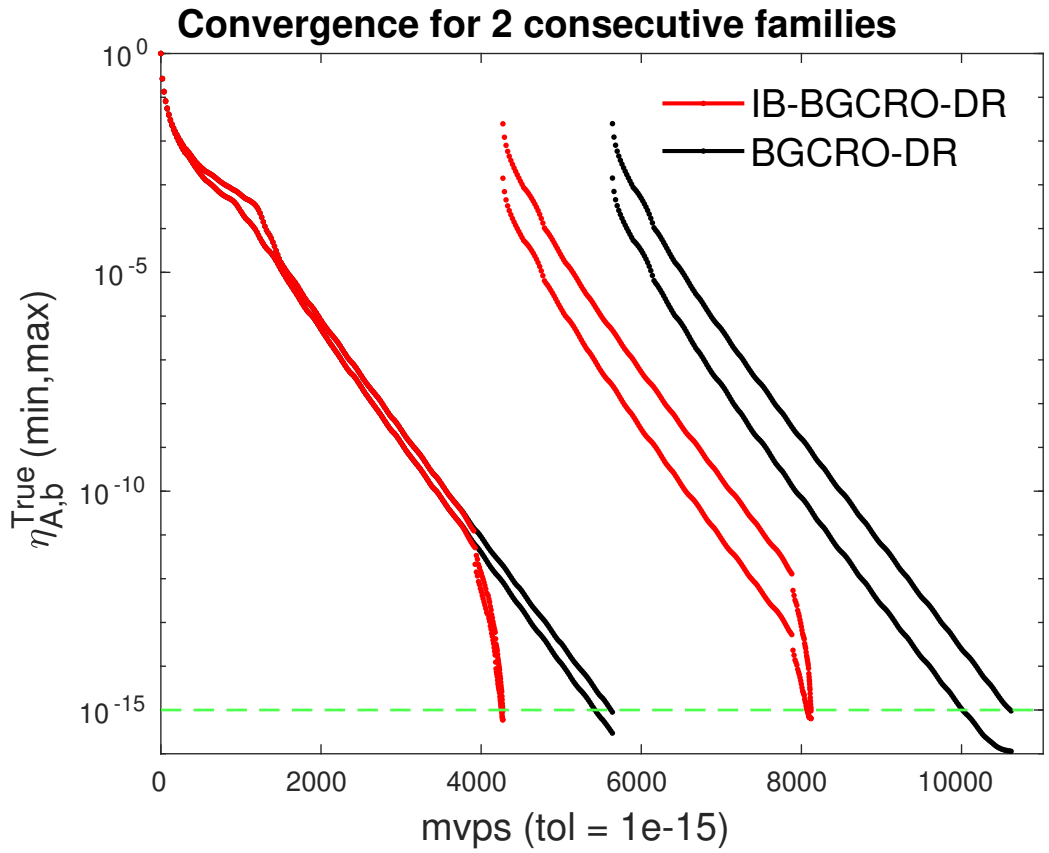
- IB-BGMRES-DR history overlaps BGCRO-DR until the first IB
- IB-BGMRES-DR history is the same for the two families

Benefits of IB and DR between the families



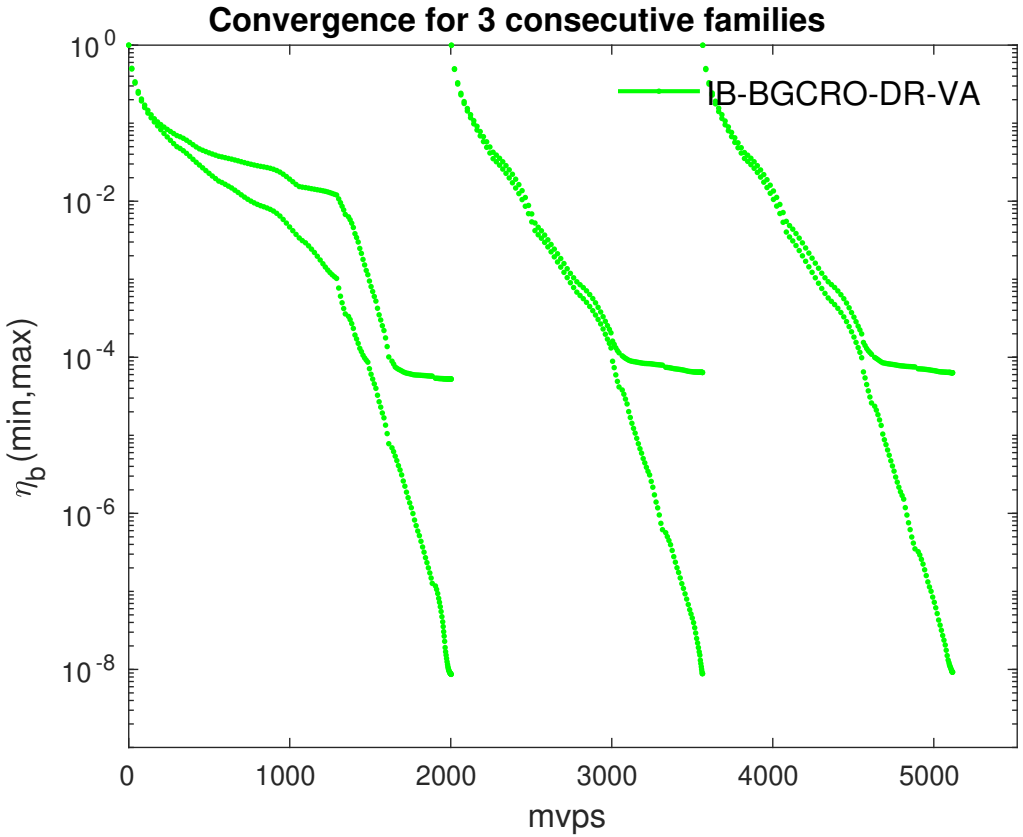
- IB-BGCRO-DR history overlaps IB-BGMRES-DR for the first family
- IB-BGCRO-DR inherits all the good genes of IB, DR and GCRO

The backward error stopping criterion: $\eta_{A,b}$



- IB-BGCRO-DR is able to decrease $\eta_{A,b}$ to a very low value close to the machine epsilon: $\mathcal{O}(10^{-16})$

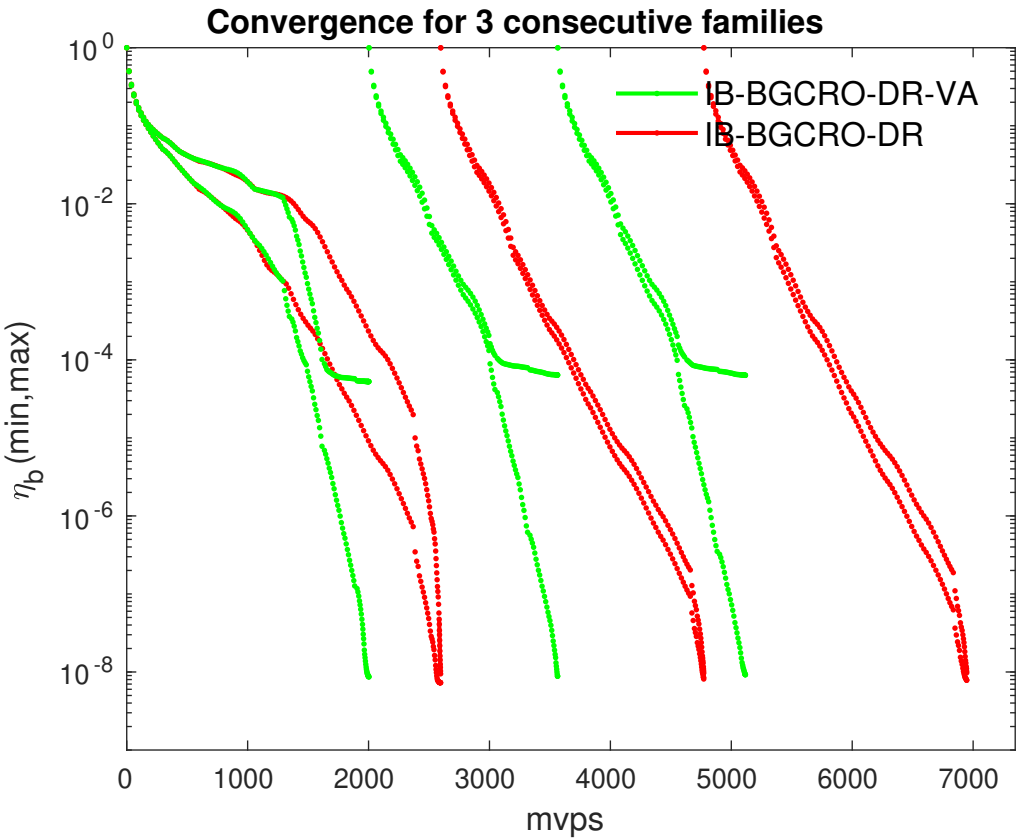
*-VA : individual convergence threshold



Once converged:

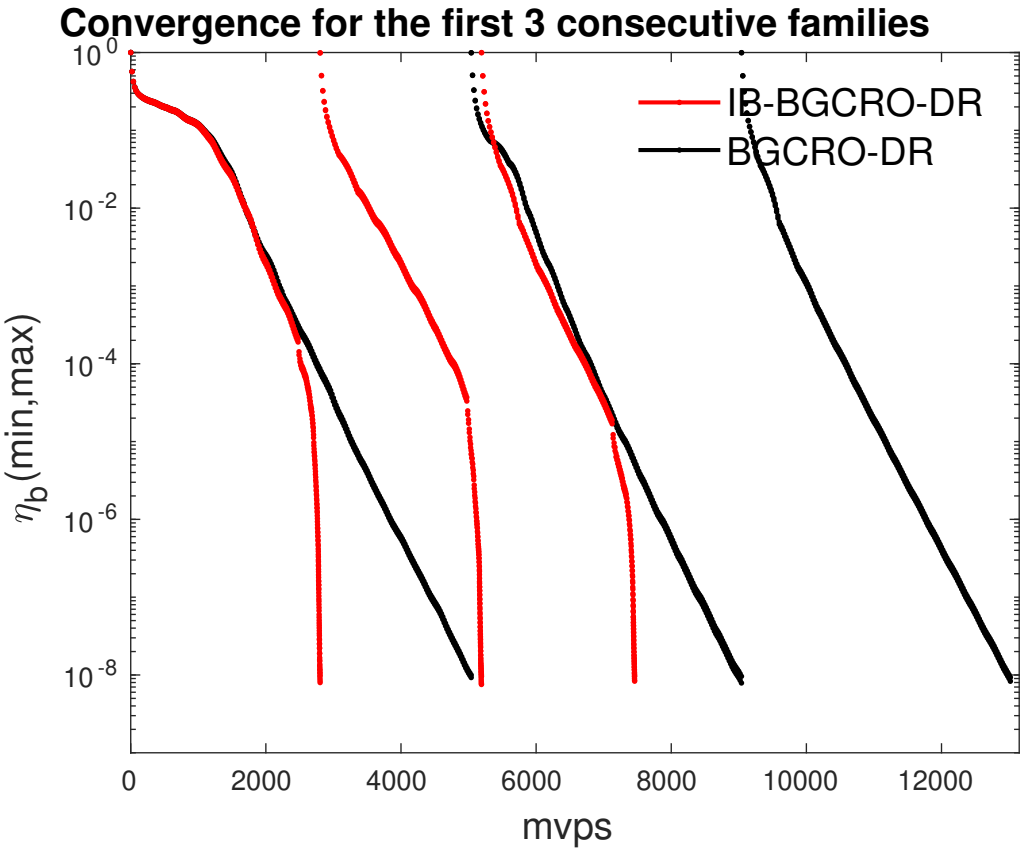
- the ones with largest thresholds do not progress anymore
- computation effort focuses on the ones with smallest thresholds

*-VA : individual convergence threshold



- Significant gains compared to converging all solutions with the most stringent threshold

Test system with slowly varying left-hand sides



- $A^{(\ell)}X^{(\ell)} = B^{(\ell)}$ with $\ell = 1, 2, 3$
- Obvious gains compared to another related block iterative solver

05

Conclusion

IB-BGCRO-DR: **subspace recycling** block GCRO-DR method with **partial convergence** management

- **Subspace recycling** strategies : Harmonic-Ritz, Rayleigh-Ritz or other techniques can be used to define the space to be recycled

IB-BGCRO-DR: **subspace recycling** block GCRO-DR method with **partial convergence** management

- **Subspace recycling** strategies
- **Search space expansion policies** governed by
  - > the partial convergence management (IB-, *-VA and *-CB)
  - > the backward error stopping criterion (η_b and $\eta_{A,b}$)

Inria

IB-BGCRO-DR: **subspace recycling** block GCRO-DR method with **partial convergence** management

- **Subspace recycling** strategies
- **Search space expansion policies** governed by
  - the partial convergence management (IB-, *-VA and *-CB)
  - the backward error stopping criterion (η_b and $\eta_{A,b}$)
- **Flexible preconditioning** (mixed-precision): IB-BFGCRO-DR

Inria

IB-BGCRO-DR: **subspace recycling** block GCRO-DR method with **partial convergence** management

- **Subspace recycling** strategies
- **Search space expansion policies** governed by
  - the partial convergence management (IB-, *-VA and *-CB)
  - the backward error stopping criterion (η_b and $\eta_{A,b}$)
- **Flexible preconditioning** (mixed-precision): IB-BFGCRO-DR
- Remarks on some **computational** and **algorithmic aspects**
  - Partial convergence detecting in the initial residuals $\rightarrow$ address possible initial rank deficiency & reduce computation
  - The MGS re-orthogonalization among $k + p$ columns (between the columns of recycling space and initial block Arnoldi basis) at restart $\rightarrow$ generate a good enough orthonormal basis

Inria

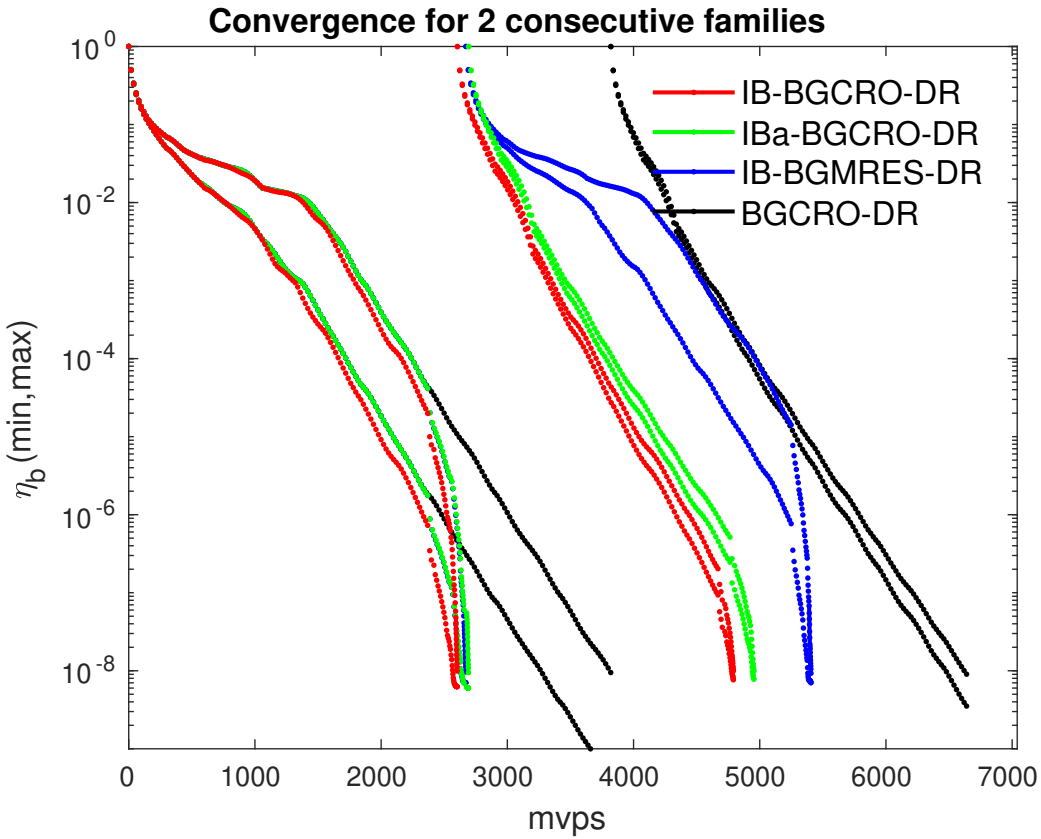
IB-BGCRO-DR: **subspace recycling** block GCRO-DR method with **partial convergence** management

- **Subspace recycling** strategies
- **Search space expansion policies** governed by
  - > the partial convergence management (IB-, *-VA and *-CB)
  - > the backward error stopping criterion (η_b and $\eta_{A,b}$)
- **Flexible preconditioning** (mixed-precision): IB-BFGCRO-DR
- Remarks on some **computational** and **algorithmic aspects**
  - > Partial convergence detecting in the initial residuals $\rightarrow$ address possible initial rank deficiency & reduce computation
  - > The MGS re-orthogonalization among $k + p$ columns (between the columns of recycling space and initial block Arnoldi basis) at restart $\rightarrow$ generate a good enough orthonormal basis

Thanks for your attention. Questions ?

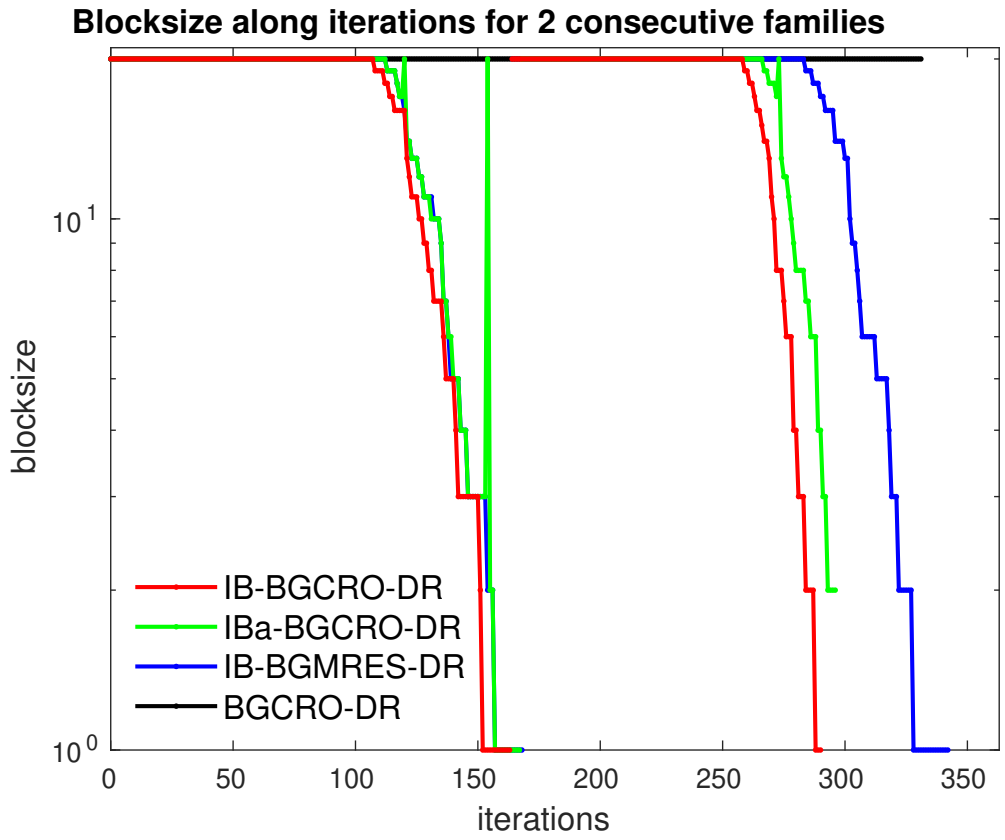
Inria

IB detecting in the initial basis or after ?



- Execute IB detecting in the initial basis helps reduce (even though slightly) the consuming mvps

IB detecting in the initial basis or after ?

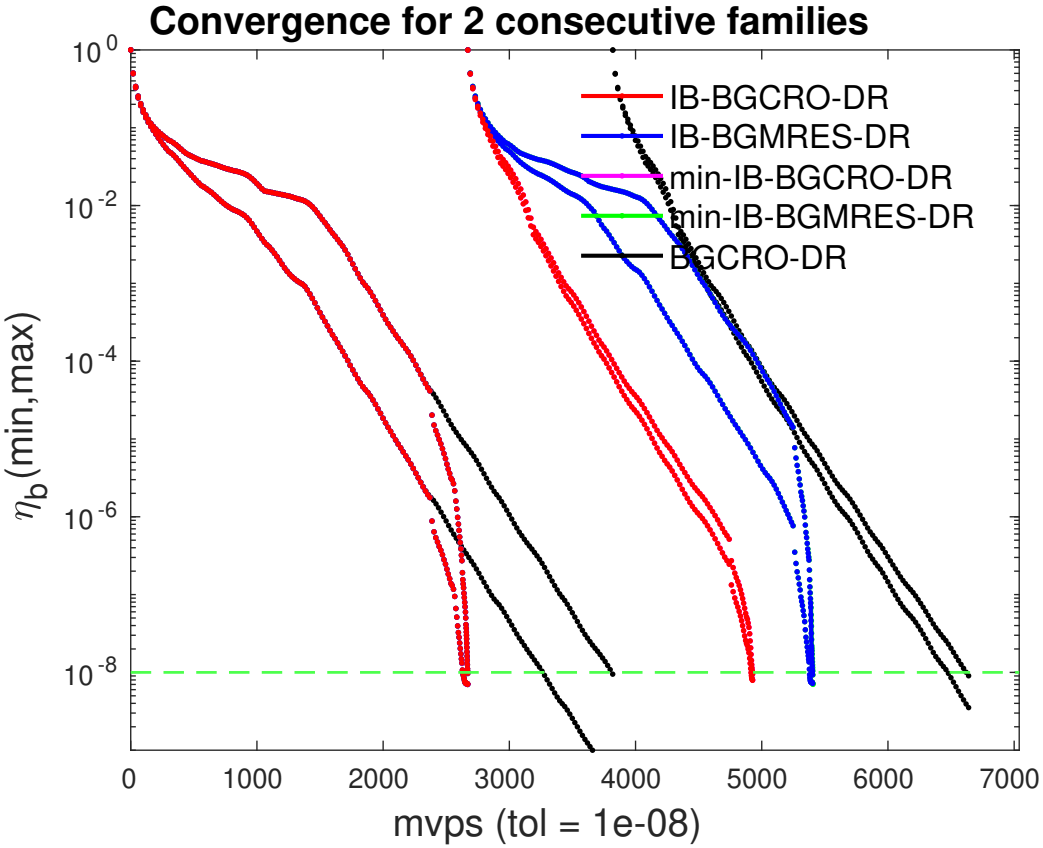


- Such slight gains comes from the benefits of addressing the initial rank deficiency accumulated at restart

Diagonal scaled IB vs min-IB

$\|(B - AX_j)D_\varepsilon\|_2 \leq 1$ **vs** $\|b^{(i)} - Ax_j^{(i)}\|_2 < \|B - AX_j\|_2 \leq \varepsilon \min_{i=1,\ldots,p} \|b^{(i)}\|_2$

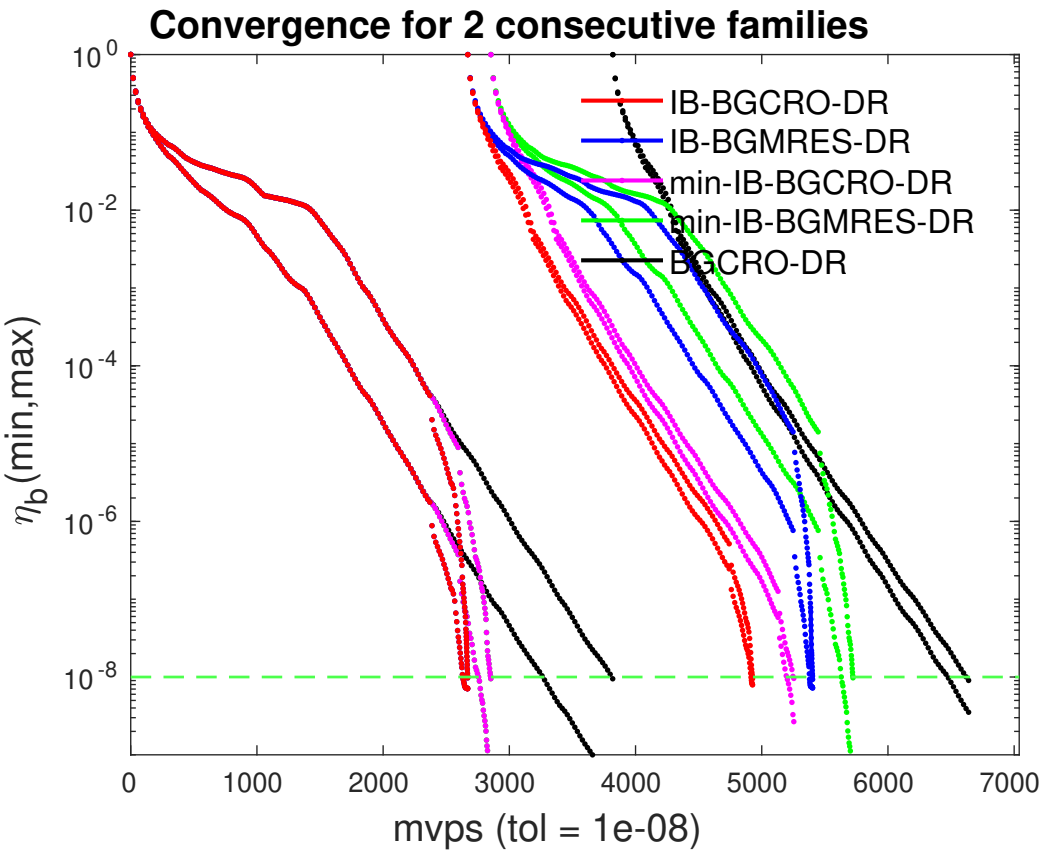
[E. Agullo, L. Giraud and Y.-F. Jing. *SIAM J. Matrix Anal. Appl.*, 2014]



- $D_\varepsilon = \varepsilon^{-1}\mathrm{diag}(\|(b^{(1)})^{-1}\|_2, \ldots, \|(b^{(p)})^{-1}\|_2)$; $B = rand(n, p)$

Diagonal scaled IB vs min-IB

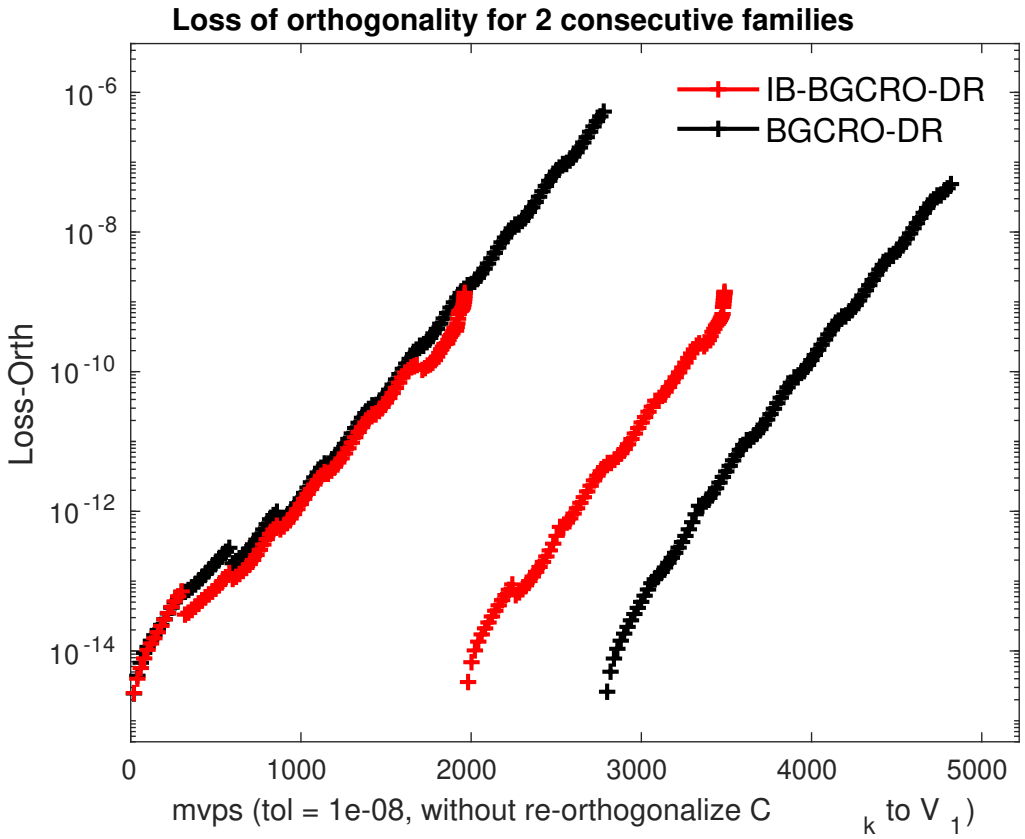
$\|(B - AX_j)D_\varepsilon\|_2 \leq 1$ **vs** $\|b^{(i)} - Ax_j^{(i)}\|_2 < \|B - AX_j\|_2 \leq \varepsilon \min_{i=1,\ldots,p} \|b^{(i)}\|_2$



- $B = rand(n, p)$ and then multiply 20 to the first $\frac{p}{2}$ columns of B

Loss of orthogonality of the Krylov basis

$$\text{Loss-Orth} = \left\| \widehat{\mathcal{V}}_{j+1}^H \widehat{\mathcal{V}}_{j+1} - I_{j+1} \right\|_2$$

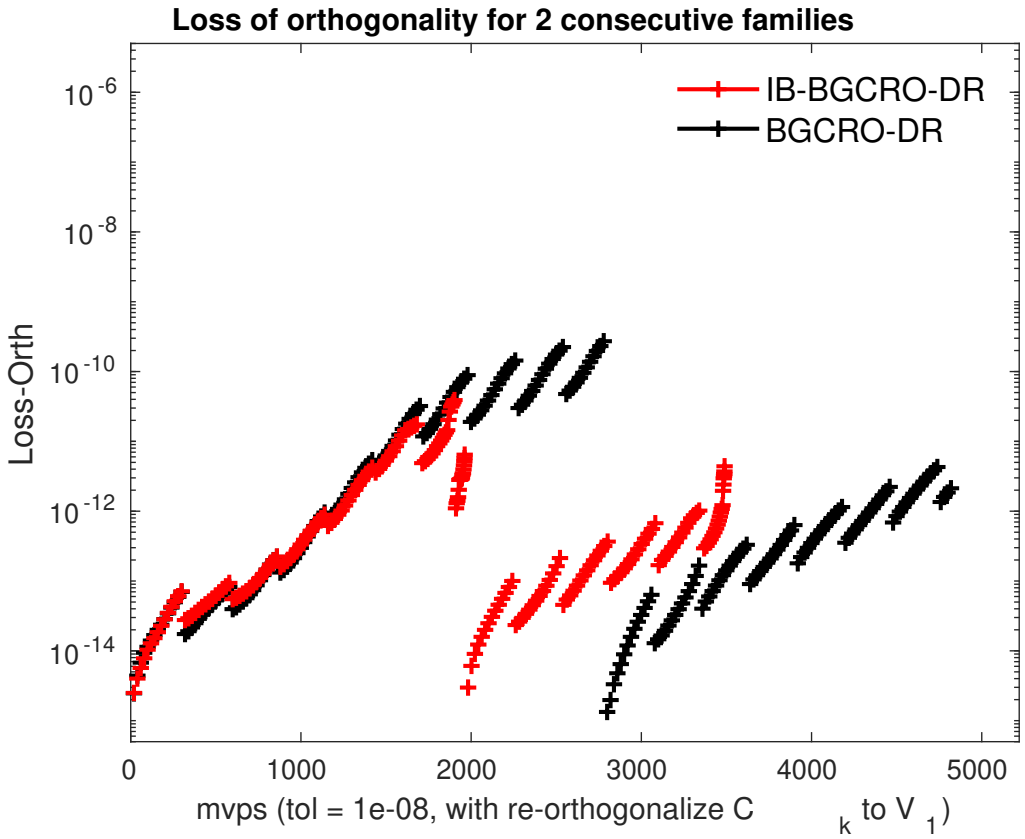


- Without re-orthogonalization of C_k to $\mathbb{V}_1$ (or V_1) at restart

Inria

Loss of orthogonality of the Krylov basis

$$\text{Loss-Orth} = \left\| \widehat{\mathcal{V}}_{j+1}^H \widehat{\mathcal{V}}_{j+1} - I_{j+1} \right\|_2$$

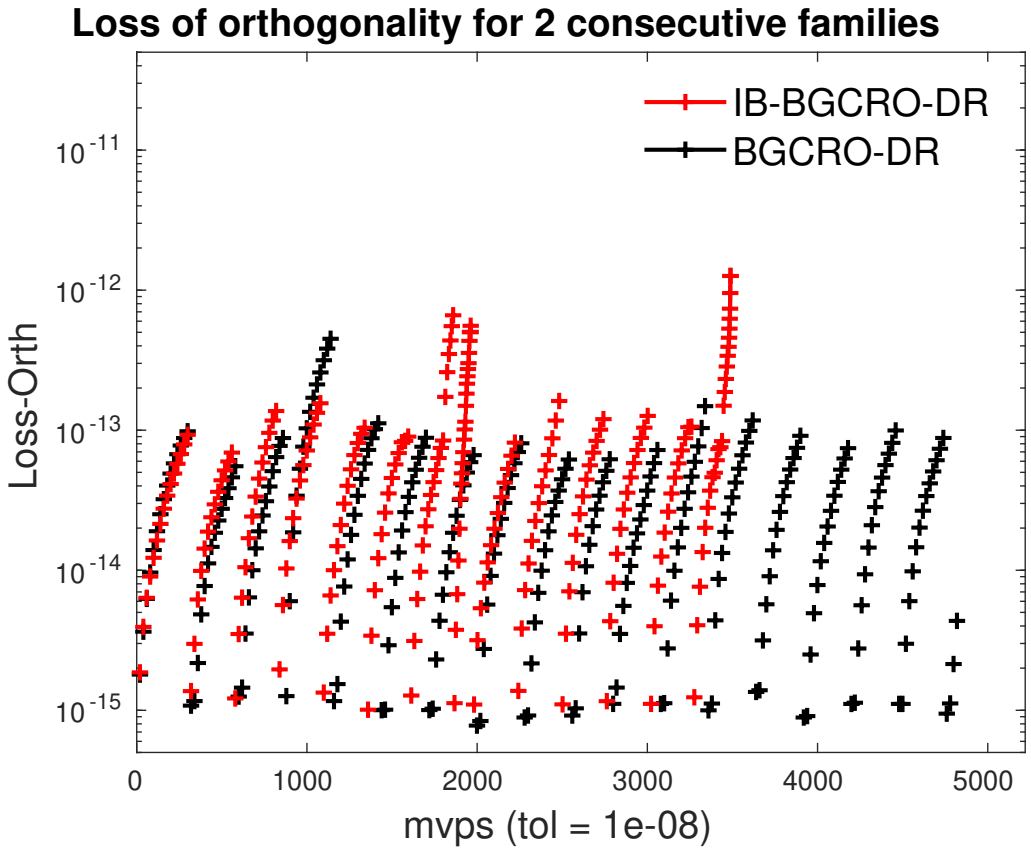


- With re-orthogonalization of C_k to $\mathbb{V}_1$ (or V_1) at restart (by $\mathbb{V}_1 = (I - C_k * C_k^H)\mathbb{V}_1$ implemented in column-column way)

Inria

Loss of orthogonality of the Krylov basis

$$\text{Loss-Orth} = \left\| \widehat{\mathcal{V}}_{j+1}^H \widehat{\mathcal{V}}_{j+1} - I_{j+1} \right\|_2$$



- Apply re-orthogonalization to all the columns of $[C_k, [\mathbb{V}_1, P_0]]$ by the modified Gram-Schmidt (MGS) process at restart