

On some Krylov subspace methods tailored for large-scale block two-by-two linear systems

Kui Du

kuidu@xmu.edu.cn

School of Mathematical Sciences, Xiamen University

<https://kuidu.github.io>

joint work with Jia-Jun Fan and Fang Wang

The 22nd Annual Meeting of CSIAM, Nanjing, 2024

Outline

- ① Block two-by-two linear systems
- ② SPMR-SC, SPQMR-SC, nsLSQR
- ③ GPMR
- ④ Randomized Gram–Schmidt process, randomized GMRES
- ⑤ Sketched GMRES + k -truncated Arnoldi
- ⑥ Summary

Block two-by-two linear systems

- Nonsymmetric saddle-point linear systems of the form:

$$\begin{bmatrix} \mathbf{M} & \mathbf{A}^\top \\ \mathbf{B} & \mathbf{0} \end{bmatrix} \begin{bmatrix} \mathbf{x} \\ \mathbf{y} \end{bmatrix} = \begin{bmatrix} \mathbf{0} \\ \mathbf{b} \end{bmatrix},$$

where $\mathbf{M} \in \mathbb{R}^{m \times m}$ is invertible, $\mathbf{A}, \mathbf{B} \in \mathbb{R}^{n \times m}$ are nonzero, and $\mathbf{b} \in \mathbb{R}^n$ is nonzero.

- Nonsymmetric partitioned linear systems of the form:

$$\begin{bmatrix} \lambda \mathbf{I} & \mathbf{A} \\ \mathbf{B} & \mu \mathbf{I} \end{bmatrix} \begin{bmatrix} \mathbf{x} \\ \mathbf{y} \end{bmatrix} = \begin{bmatrix} \mathbf{b} \\ \mathbf{c} \end{bmatrix},$$

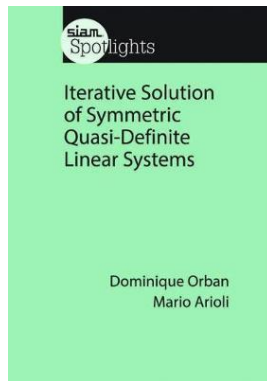
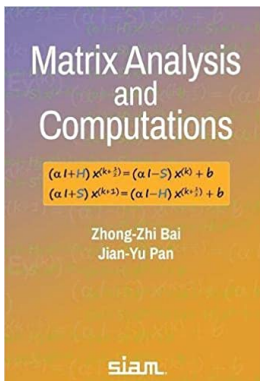
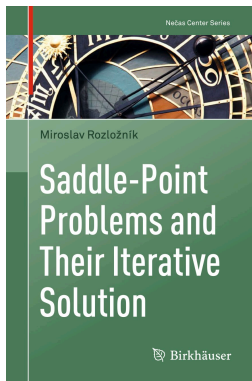
where $\mathbf{A} \in \mathbb{R}^{m \times n}$, $\mathbf{B} \in \mathbb{R}^{n \times m}$, and $\mathbf{b} \in \mathbb{R}^m$, and $\mathbf{c} \in \mathbb{R}^n$.
Note that λ and/or μ may be zero.

Review papers and books

- Michele Benzi, Gene H. Golub, and Jörg Liesen

Numerical solution of saddle point problems.

Acta Numerica (2005), pp. 1137.



Nonsymmetric saddle-point linear systems

- Nonsymmetric saddle-point linear systems of the form:

$$\begin{bmatrix} \mathbf{M} & \mathbf{A}^\top \\ \mathbf{B} & \mathbf{0} \end{bmatrix} \begin{bmatrix} \mathbf{x} \\ \mathbf{y} \end{bmatrix} = \begin{bmatrix} \mathbf{0} \\ \mathbf{b} \end{bmatrix},$$

where $\mathbf{M} \in \mathbb{R}^{m \times m}$ is invertible.

- **Monolithic** methods: solving the system as a whole, for example, GMRES
Segregated methods: exploiting the block structure, excluding the preconditioning stage, for example, SPMR, SPQMR, nsLSQR

R. Estrin and C. Greif. *SPMR: A family of saddle-point minimum residual solvers*. SISC, Vol. 40, No. 3 (2018)

K. Du, J.-J. Fan, and F. Wang. *nsLSQR: A quasi-minimum residual method for nonsymmetric saddle-point linear systems*. (2024)

Simultaneous bidiagonalization via M-conjugacy

Algorithm Simultaneous bidiagonalization via M-conjugacy

Require: $\mathbf{M} \in \mathbb{R}^{m \times m}$, $\mathbf{A}, \mathbf{B} \in \mathbb{R}^{n \times m}$, and $\mathbf{b}, \mathbf{c} \in \mathbb{R}^n$

- 1: $\beta_1 \mathbf{v}_1 := \mathbf{c}, \delta_1 \mathbf{z}_1 := \mathbf{b}$
- 2: $\mathbf{u} = \mathbf{A}^\top \mathbf{v}_1, \mathbf{w} = \mathbf{M}^{-\top} \mathbf{B}^\top \mathbf{z}_1$
- 3: $\alpha_1 = |\mathbf{w}^\top \mathbf{u}|^{1/2}, \gamma_1 = \mathbf{w}^\top \mathbf{u} / \alpha_1$
- 4: $\mathbf{u}_1 = \mathbf{M}^{-1} \mathbf{u} / \alpha_1, \mathbf{w}_1 = \mathbf{w} / \gamma_1$
- 5: **for** $k = 1, 2, \dots$ **do**
- 6: $\beta_{k+1} \mathbf{v}_{k+1} := \mathbf{A} \mathbf{w}_k - \alpha_k \mathbf{v}_k, \delta_{k+1} \mathbf{z}_{k+1} := \mathbf{B} \mathbf{u}_k - \gamma_k \mathbf{z}_k$
- 7: $\mathbf{u} = \mathbf{A}^\top \mathbf{v}_{k+1} - \beta_{k+1} \mathbf{M} \mathbf{u}_k, \mathbf{w} = \mathbf{M}^{-\top} \mathbf{B}^\top \mathbf{z}_{k+1} - \delta_{k+1} \mathbf{w}_k$
- 8: $\alpha_{k+1} = |\mathbf{w}^\top \mathbf{u}|^{1/2}, \gamma_{k+1} = \mathbf{w}^\top \mathbf{u} / \alpha_{k+1}$
- 9: $\mathbf{u}_{k+1} = \mathbf{M}^{-1} \mathbf{u} / \alpha_{k+1}, \mathbf{w}_{k+1} = \mathbf{w} / \gamma_{k+1}$
- 10: **end for**

Simultaneous bidiagonalization via M-conjugacy

- Simultaneous bidiagonalization via M-conjugacy:

$$\begin{aligned} \mathbf{A}\mathbf{W}_k &= \mathbf{V}_{k+1}\mathbf{C}_{k+1,k}, & \mathbf{A}^\top\mathbf{V}_{k+1} &= \mathbf{M}\mathbf{U}_{k+1}\mathbf{C}_{k+1}^\top, \\ \mathbf{B}\mathbf{U}_k &= \mathbf{Z}_{k+1}\mathbf{F}_{k+1,k}, & \mathbf{B}^\top\mathbf{Z}_{k+1} &= \mathbf{M}^\top\mathbf{W}_{k+1}\mathbf{F}_{k+1}^\top, \\ \mathbf{W}_k^\top\mathbf{M}\mathbf{U}_k &= \mathbf{V}_k^\top\mathbf{V}_k = \mathbf{Z}_k^\top\mathbf{Z}_k = \mathbf{I}_k, \end{aligned}$$

where

$$\begin{aligned} \mathbf{U}_k &= [\mathbf{u}_1 \quad \cdots \quad \mathbf{u}_k], & \mathbf{V}_k &= [\mathbf{v}_1 \quad \cdots \quad \mathbf{v}_k], \\ \mathbf{W}_k &= [\mathbf{w}_1 \quad \cdots \quad \mathbf{w}_k], & \mathbf{Z}_k &= [\mathbf{z}_1 \quad \cdots \quad \mathbf{z}_k], \\ \mathbf{C}_k &= \text{bidiag}(\beta_i, \alpha_i), & \mathbf{C}_{k+1,k} &= \begin{bmatrix} \mathbf{C}_k \\ \beta_{k+1}\mathbf{e}_k^\top \end{bmatrix}, \\ \mathbf{F}_k &= \text{bidiag}(\delta_i, \gamma_i), & \mathbf{F}_{k+1,k} &= \begin{bmatrix} \mathbf{F}_k \\ \delta_{k+1}\mathbf{e}_k^\top \end{bmatrix}. \end{aligned}$$

SPMR-SC

- The k th SPMR-SC iterate is $\mathbf{x}_k = \mathbf{U}_k \tilde{\mathbf{x}}_k$, $\mathbf{y}_k = \mathbf{V}_k \tilde{\mathbf{y}}_k$, where

$$\begin{bmatrix} \tilde{\mathbf{x}}_k \\ \tilde{\mathbf{y}}_k \end{bmatrix} = \underset{\tilde{\mathbf{x}} \in \mathbb{R}^k, \tilde{\mathbf{y}} \in \mathbb{R}^k}{\operatorname{argmin}} \left\| \begin{bmatrix} \mathbf{0} \\ \delta_1 \mathbf{e}_1 \end{bmatrix} - \begin{bmatrix} \mathbf{I}_k & \mathbf{C}_k^\top \\ \mathbf{F}_{k+1,k} & \mathbf{0} \end{bmatrix} \begin{bmatrix} \tilde{\mathbf{x}} \\ \tilde{\mathbf{y}} \end{bmatrix} \right\|_2.$$

- Equivalent to USYMQR applied to the Schur complement system:

$$-\mathbf{S}\mathbf{y} = \mathbf{b}, \quad \mathbf{S} = \mathbf{B}\mathbf{M}^{-1}\mathbf{A}^\top.$$

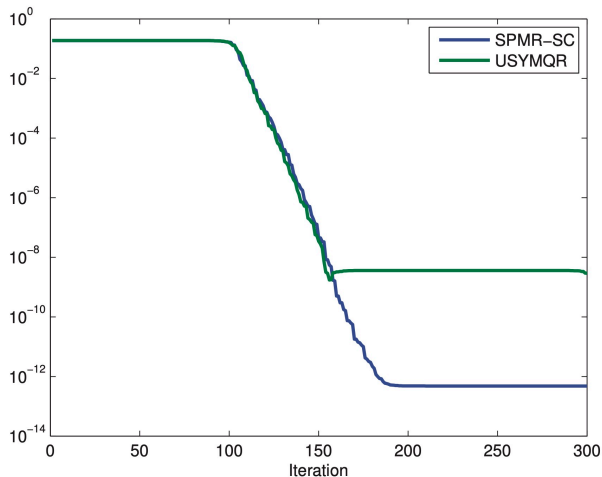
- SPMR-SC can be more numerically stable than USYMQR when the Schur complement is ill-conditioned.

R. Estrin and C. Greif. *SPMR: A family of saddle-point minimum residual solvers*. SISC, Vol. 40, No. 3 (2018)

M. A. Saunders, H. D. Simon, and E. L. Yip. *Two conjugate-gradient-type methods for unsymmetric linear equations*. SINUM, Vol. 25, Iss. 4 (1988)

Example: an ill-conditioned Schur complement

$$\text{cond}(\mathbf{A}) \approx 10^5, \text{cond}(\mathbf{B}) \approx 10^5, \text{cond}(\mathbf{S}) \approx 10^8$$



Simultaneous bidiagonalization via biorthogonality

Algorithm Simultaneous bidiagonalization via biorthogonality

Require: $\mathbf{M} \in \mathbb{R}^{m \times m}$, \mathbf{A} , $\mathbf{B} \in \mathbb{R}^{n \times m}$, and \mathbf{b} , $\mathbf{c} \in \mathbb{R}^n$

- 1: $\delta_1 = |\mathbf{c}^\top \mathbf{b}|^{1/2}$, $\beta_1 = \mathbf{c}^\top \mathbf{b} / \delta_1$
- 2: $\mathbf{v}_1 = \mathbf{b} / \delta_1$, $\mathbf{z}_1 = \mathbf{c} / \beta_1$
- 3: $\mathbf{u} = \mathbf{A}^\top \mathbf{v}_1$, $\mathbf{w} = \mathbf{M}^{-\top} \mathbf{B}^\top \mathbf{z}_1$
- 4: $\alpha_1 = |\mathbf{w}^\top \mathbf{u}|^{1/2}$, $\gamma_1 = \mathbf{w}^\top \mathbf{u} / \alpha_1$
- 5: $\mathbf{u}_1 = \mathbf{M}^{-1} \mathbf{u} / \alpha_1$, $\mathbf{w}_1 = \mathbf{w} / \gamma_1$
- 6: **for** $k = 1, 2, \dots$ **do**
- 7: $\mathbf{v} = \mathbf{B} \mathbf{u}_k - \gamma_k \mathbf{v}_k$, $\mathbf{z} = \mathbf{A} \mathbf{w}_k - \alpha_k \mathbf{z}_k$
- 8: $\delta_{k+1} = |\mathbf{z}^\top \mathbf{v}|^{1/2}$, $\beta_{k+1} = |\mathbf{z}^\top \mathbf{v}| / \delta_{k+1}$
- 9: $\mathbf{v}_{k+1} = \mathbf{v} / \delta_{k+1}$, $\mathbf{z}_{k+1} = \mathbf{z} / \beta_{k+1}$
- 10: $\mathbf{u} = \mathbf{A}^\top \mathbf{v}_{k+1} - \beta_{k+1} \mathbf{M} \mathbf{u}_k$, $\mathbf{w} = \mathbf{M}^{-\top} \mathbf{B}^\top \mathbf{z}_{k+1} - \delta_{k+1} \mathbf{w}_k$
- 11: $\alpha_{k+1} = |\mathbf{w}^\top \mathbf{u}|^{1/2}$, $\gamma_{k+1} = \mathbf{w}^\top \mathbf{u} / \alpha_{k+1}$
- 12: $\mathbf{u}_{k+1} = \mathbf{M}^{-1} \mathbf{u} / \alpha_{k+1}$, $\mathbf{w}_{k+1} = \mathbf{w} / \gamma_{k+1}$
- 13: **end for**

Simultaneous bidiagonalization via biorthogonality

- Simultaneous bidiagonalization via biorthogonality:

$$\begin{aligned} \mathbf{A}\mathbf{W}_k &= \mathbf{Z}_{k+1}\mathbf{C}_{k+1,k}, & \mathbf{A}^\top\mathbf{V}_{k+1} &= \mathbf{M}\mathbf{U}_{k+1}\mathbf{C}_{k+1}^\top, \\ \mathbf{B}\mathbf{U}_k &= \mathbf{V}_{k+1}\mathbf{F}_{k+1,k}, & \mathbf{B}^\top\mathbf{Z}_{k+1} &= \mathbf{M}^\top\mathbf{U}_{k+1}\mathbf{F}_{k+1}^\top, \\ \mathbf{W}_k^\top\mathbf{M}\mathbf{U}_k &= \mathbf{V}_k^\top\mathbf{Z}_k = \mathbf{I}_k, \end{aligned}$$

where

$$\begin{aligned} \mathbf{U}_k &= [\mathbf{u}_1 \quad \cdots \quad \mathbf{u}_k], & \mathbf{V}_k &= [\mathbf{v}_1 \quad \cdots \quad \mathbf{v}_k], \\ \mathbf{W}_k &= [\mathbf{w}_1 \quad \cdots \quad \mathbf{w}_k], & \mathbf{Z}_k &= [\mathbf{z}_1 \quad \cdots \quad \mathbf{z}_k], \\ \mathbf{C}_k &= \text{bidiag}(\beta_i, \alpha_i), & \mathbf{C}_{k+1,k} &= \begin{bmatrix} \mathbf{C}_k \\ \beta_{k+1}\mathbf{e}_k^\top \end{bmatrix}, \\ \mathbf{F}_k &= \text{bidiag}(\delta_i, \gamma_i), & \mathbf{F}_{k+1,k} &= \begin{bmatrix} \mathbf{F}_k \\ \delta_{k+1}\mathbf{e}_k^\top \end{bmatrix}. \end{aligned}$$

SPQMR-SC

- The k th SPQMR-SC iterate is $\mathbf{x}_k = \mathbf{U}_k \tilde{\mathbf{x}}_k$, $\mathbf{y}_k = \mathbf{V}_k \tilde{\mathbf{y}}_k$, where

$$\begin{bmatrix} \tilde{\mathbf{x}}_k \\ \tilde{\mathbf{y}}_k \end{bmatrix} = \underset{\tilde{\mathbf{x}} \in \mathbb{R}^k, \tilde{\mathbf{y}} \in \mathbb{R}^k}{\operatorname{argmin}} \left\| \begin{bmatrix} \mathbf{0} \\ \delta_1 \mathbf{e}_1 \end{bmatrix} - \begin{bmatrix} \mathbf{I}_k & \mathbf{C}_k^\top \\ \mathbf{F}_{k+1,k} & \mathbf{0} \end{bmatrix} \begin{bmatrix} \tilde{\mathbf{x}} \\ \tilde{\mathbf{y}} \end{bmatrix} \right\|_2.$$

- Equivalent to QMR applied to the Schur complement system:

$$-\mathbf{S}\mathbf{y} = \mathbf{b}, \quad \mathbf{S} = \mathbf{B}\mathbf{M}^{-1}\mathbf{A}^\top.$$

- The convergence of SPMR-SC is monotonic, while the convergence of SPQMR-SC is erratic.

R. Estrin and C. Greif. *SPMR: A family of saddle-point minimum residual solvers*. SISC, Vol. 40, No. 3 (2018)

R. W. Freund and N. M. Nachtigal. *QMR: A Quasi-Minimal Residual Method for Non-Hermitian Linear-Systems*. NM, Vol. 60, Iss. 3 (1991)

Bidiagonal-Hessenberg reduction

Algorithm Bidiagonal-Hessenberg reduction

Require: $\mathbf{M} \in \mathbb{R}^{m \times m}$, $\mathbf{A}, \mathbf{B} \in \mathbb{R}^{n \times m}$, $\mathbf{b} \in \mathbb{R}^n$

- 1: $\mathbf{u}_1 = \mathbf{b} / \beta_1$ with $\beta_1 = \|\mathbf{b}\|_2$
- 2: $\mathbf{v} = \mathbf{A}^\top \mathbf{u}_1$, $\mathbf{v}_1 = \mathbf{M}^{-1} \mathbf{v}$, $\alpha_1 = \begin{cases} |\mathbf{v}_1^\top \mathbf{v}|^{1/2} & \text{if } \mathbf{v}_1^\top \mathbf{v} \neq 0 \\ \|\mathbf{v}_1\|_2 & \text{if } \mathbf{v}_1^\top \mathbf{v} = 0 \end{cases}$
- 3: $\mathbf{v}_1 = \mathbf{v}_1 / \alpha_1$
- 4: **for** $k = 1, 2, \dots$ **do**
- 5: $\mathbf{u} = \mathbf{B} \mathbf{v}_k$
- 6: **for** $i = 1, 2, \dots, k$ **do**
- 7: $h_{ik} = \mathbf{u}_i^\top \mathbf{u}$
- 8: $\mathbf{u} = \mathbf{u} - h_{ik} \mathbf{u}_i$
- 9: **end for**
- 10: $\mathbf{u}_{k+1} = \mathbf{u} / \beta_{k+1}$ with $\beta_{k+1} = \|\mathbf{u}\|_2$
- 11: $\mathbf{v} = \mathbf{A}^\top \mathbf{u}_{k+1} - \beta_{k+1} \mathbf{M} \mathbf{v}_k$, $\mathbf{v}_{k+1} = \mathbf{M}^{-1} \mathbf{v}$, $\alpha_{k+1} = \begin{cases} |\mathbf{v}_{k+1}^\top \mathbf{v}|^{1/2} & \text{if } \mathbf{v}_{k+1}^\top \mathbf{v} \neq 0 \\ \|\mathbf{v}_{k+1}\|_2 & \text{if } \mathbf{v}_{k+1}^\top \mathbf{v} = 0 \end{cases}$
- 12: $\mathbf{v}_{k+1} = \mathbf{v}_{k+1} / \alpha_{k+1}$
- 13: **end for**

Bidiagonal-Hessenberg reduction

- Bidiagonal-Hessenberg reduction:

$$\begin{aligned}\mathbf{A}^\top \mathbf{U}_k &= \mathbf{M} \mathbf{V}_k \mathbf{C}_k^\top, & \mathbf{U}_{k+1}^\top \mathbf{U}_{k+1} &= \mathbf{I}_{k+1}, \\ \mathbf{B} \mathbf{V}_k &= \mathbf{U}_{k+1} \mathbf{H}_{k+1,k} = \mathbf{U}_k \mathbf{H}_k + \beta_{k+1} \mathbf{u}_{k+1} \mathbf{e}_k^\top,\end{aligned}$$

where

$$\mathbf{U}_k = [\mathbf{u}_1 \quad \cdots \quad \mathbf{u}_k], \quad \mathbf{V}_k = [\mathbf{v}_1 \quad \cdots \quad \mathbf{v}_k],$$

$$\mathbf{C}_k = \begin{bmatrix} \alpha_1 & & & \\ \beta_2 & \alpha_2 & & \\ & \ddots & \ddots & \\ & & \beta_k & \alpha_k \end{bmatrix}, \quad \mathbf{H}_{k+1,k} = \begin{bmatrix} h_{11} & \cdots & h_{1k} \\ h_{21} & \ddots & \vdots \\ & \ddots & h_{kk} \\ & & h_{k+1,k} \end{bmatrix}.$$

- The k th nsLSQR iterate is $\mathbf{x}_k = \mathbf{V}_k \tilde{\mathbf{x}}_k$, $\mathbf{y}_k = \mathbf{U}_k \tilde{\mathbf{y}}_k$, where

$$\begin{bmatrix} \tilde{\mathbf{x}}_k \\ \tilde{\mathbf{y}}_k \end{bmatrix} = \underset{\tilde{\mathbf{x}} \in \mathbb{R}^k, \tilde{\mathbf{y}} \in \mathbb{R}^k}{\operatorname{argmin}} \left\| \begin{bmatrix} \mathbf{0} \\ \beta_1 \mathbf{e}_1 \end{bmatrix} - \begin{bmatrix} \mathbf{I}_k & \mathbf{C}_k^\top \\ \mathbf{H}_{k+1,k} & \mathbf{0} \end{bmatrix} \begin{bmatrix} \tilde{\mathbf{x}} \\ \tilde{\mathbf{y}} \end{bmatrix} \right\|_2.$$

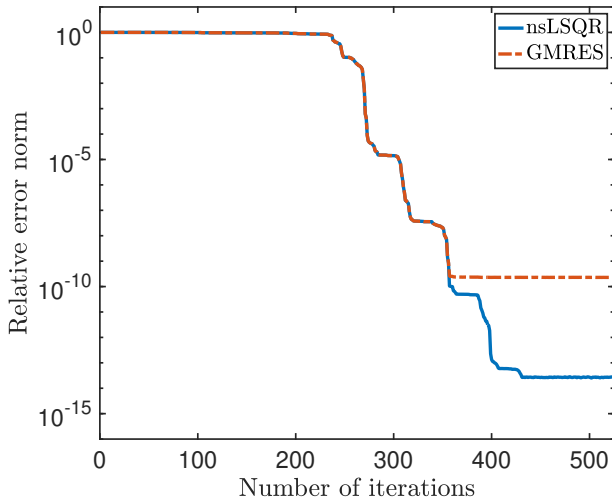
- Equivalent to GMRES applied to the Schur complement system:

$$-\mathbf{S}\mathbf{y} = \mathbf{b}, \quad \mathbf{S} = \mathbf{B}\mathbf{M}^{-1}\mathbf{A}^\top.$$

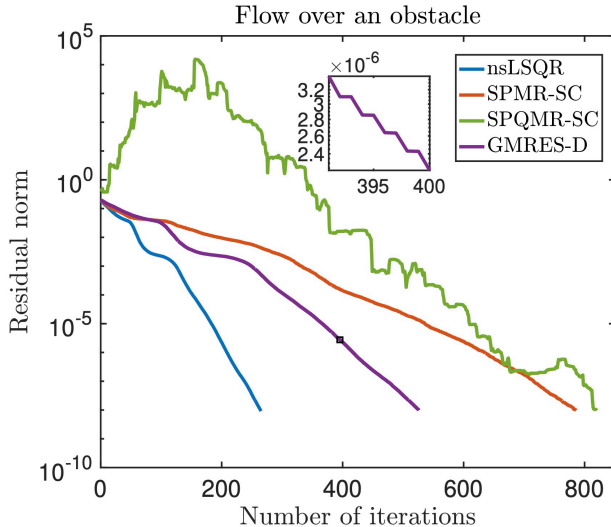
- nsLSQR can be more numerically stable than GMRES when the Schur complement is ill-conditioned.

Example: an ill-conditioned Schur complement

$$\text{cond}(\mathbf{A}) \approx 7 \times 10^3, \quad \text{cond}(\mathbf{B}) \approx 7 \times 10^3, \quad \text{cond}(\mathbf{S}) \approx 5 \times 10^7$$



Example: Flow over an obstacle (IFISS)



Nonsymmetric partitioned linear systems

- Nonsymmetric partitioned linear systems of the form

$$\begin{bmatrix} \lambda \mathbf{I} & \mathbf{A} \\ \mathbf{B} & \mu \mathbf{I} \end{bmatrix} \begin{bmatrix} \mathbf{x} \\ \mathbf{y} \end{bmatrix} = \begin{bmatrix} \mathbf{b} \\ \mathbf{c} \end{bmatrix}.$$

Note that λ and/or μ may be zero.

- **Monolithic** methods: solving the system as a whole, for example, GMRES
- **Segregated** methods: exploiting the block structure, excluding the preconditioning stage, for example, GPMR, GPBiLQ, GPQMR

A. Montoison and D. Orban. *GPMR: An iterative method for unsymmetric partitioned linear systems*. SIMAX, Vol. 44, No. 1 (2023)

K. Du, J.-J. Fan, and F. Wang. *GPBiLQ and GPQMR: Two iterative methods for unsymmetric partitioned linear systems*. arXiv:2401.02608 (2024)

Simultaneous orthogonal Hessenberg reduction

Algorithm Simultaneous orthogonal Hessenberg reduction

Require: \mathbf{A} , \mathbf{B} , \mathbf{b} , \mathbf{c} , all nonzero

1: $\beta \mathbf{v}_1 := \mathbf{b}$, $\gamma \mathbf{u}_1 := \mathbf{c}$

2: **for** $k = 1, 2, \dots$ **do**

3: **for** $i = 1, 2, \dots, k$ **do**

4: $h_{ik} = \mathbf{v}_i^\top \mathbf{A} \mathbf{u}_k$

5: $f_{ik} = \mathbf{u}_i^\top \mathbf{B} \mathbf{v}_k$

6: **end for**

7: $h_{k+1,k} \mathbf{v}_{k+1} = \mathbf{A} \mathbf{u}_k - \sum_{i=1}^k h_{ik} \mathbf{v}_i$

8: $f_{k+1,k} \mathbf{u}_{k+1} = \mathbf{B} \mathbf{v}_k - \sum_{i=1}^k f_{ik} \mathbf{u}_i$

9: **end for**

Simultaneous orthogonal Hessenberg reduction

- Simultaneous orthogonal Hessenberg reduction

$$\mathbf{A}\mathbf{U}_k = \mathbf{V}_k\mathbf{H}_k + h_{k+1,k}\mathbf{v}_{k+1}\mathbf{e}_k^\top = \mathbf{V}_{k+1}\mathbf{H}_{k+1,k},$$

$$\mathbf{B}\mathbf{V}_k = \mathbf{U}_k\mathbf{F}_k + f_{k+1,k}\mathbf{u}_{k+1}\mathbf{e}_k^\top = \mathbf{U}_{k+1}\mathbf{F}_{k+1,k},$$

$$\mathbf{V}_{k+1}^\top \mathbf{V}_{k+1} = \mathbf{U}_{k+1}^\top \mathbf{U}_{k+1} = \mathbf{I}_{k+1},$$

where

$$\mathbf{U}_k = [\mathbf{u}_1 \quad \mathbf{u}_2 \quad \cdots \quad \mathbf{u}_k], \quad \mathbf{V}_k = [\mathbf{v}_1 \quad \mathbf{v}_2 \quad \cdots \quad \mathbf{v}_k],$$

and

$$\mathbf{H}_{k+1,k} = \begin{bmatrix} h_{11} & \cdots & h_{1k} \\ h_{21} & \ddots & \vdots \\ & \ddots & h_{kk} \\ & & h_{k+1,k} \end{bmatrix}, \quad \mathbf{F}_{k+1,k} = \begin{bmatrix} f_{11} & \cdots & f_{1k} \\ f_{21} & \ddots & \vdots \\ & \ddots & f_{kk} \\ & & f_{k+1,k} \end{bmatrix}.$$

- The k th GPMR iterate is

$$\begin{bmatrix} \mathbf{x}_k \\ \mathbf{y}_k \end{bmatrix} = \underset{\mathbf{x} \in \text{range}(\mathbf{V}_k), \mathbf{y} \in \text{range}(\mathbf{U}_k)}{\text{argmin}} \left\| \begin{bmatrix} \mathbf{b} \\ \mathbf{c} \end{bmatrix} - \begin{bmatrix} \lambda \mathbf{I} & \mathbf{A} \\ \mathbf{B} & \mu \mathbf{I} \end{bmatrix} \begin{bmatrix} \mathbf{x} \\ \mathbf{y} \end{bmatrix} \right\|_2.$$

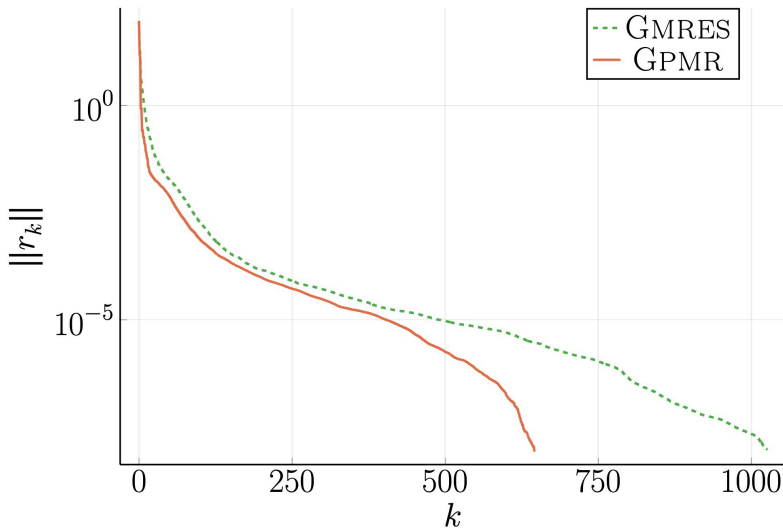
- Equivalent to Block-GMRES:

$$\begin{bmatrix} \lambda \mathbf{I} & \mathbf{A} \\ \mathbf{B} & \mu \mathbf{I} \end{bmatrix} \begin{bmatrix} \mathbf{x}^1 & \mathbf{x}^2 \\ \mathbf{y}^1 & \mathbf{y}^2 \end{bmatrix} = \begin{bmatrix} \mathbf{b} & \mathbf{0} \\ \mathbf{0} & \mathbf{c} \end{bmatrix}.$$

- GPMR terminates significantly earlier than GMRES on a residual-based stopping criterion with an improvement up to 50% in terms of number of iterations.

Example: $A = \text{well1850}$, $B = \text{illc1850}$

$$\lambda = 1, \mu = 0$$



Randomized Numerical Linear Algebra +

- Arguably, the most exciting recent development in NLA is the advent of new randomized algorithms that are fast, scalable, robust, and reliable.
- Intelligent solvers.

M. Dereziński, Michael W. Mahoney. *Recent and Upcoming Developments in Randomized Numerical Linear Algebra for Machine Learning*. arXiv:2406.11151 (2024)

A. Kireeva and J. A. Tropp. *Randomized matrix computations: Themes and variations*. arXiv:2402.17873 (2024)

J. A. Tropp and R. J. Webber. *Randomized algorithms for low-rank matrix approximation: Design, analysis, and applications*. arXiv:2306.12418 (2023)

R. Murray et al. *Randomized Numerical Linear Algebra : A perspective on the field with an eye to software*, arXiv:2302.11474 (2023)

Haifeng Zou, Xiaowen Xu, Chen-Song Zhang. *A Survey on Intelligent Iterative Methods for Solving Sparse Linear Algebraic Equations*. arXiv:2310.06630 (2023)

Randomized Gram–Schmidt process

Algorithm 2.1. RGS algorithm

Given: $n \times m$ matrix \mathbf{W} and $k \times n$ matrix Θ , $m \leq k \ll n$.

Output: $n \times m$ factor \mathbf{Q} and $m \times m$ upper triangular factor \mathbf{R} .

for $i = 1 : m$ **do**

1. Sketch \mathbf{w}_i : $\mathbf{p}_i = \Theta \mathbf{w}_i$. # macheps: u_{fine}

2. Solve $k \times (i - 1)$ least-squares problem:

$[\mathbf{R}]_{(1:i-1,i)} = \arg \min_{\mathbf{y}} \|\mathbf{S}_{i-1} \mathbf{y} - \mathbf{p}_i\|.$ # macheps: u_{fine}

3. Compute projection of \mathbf{w}_i : $\mathbf{q}'_i = \mathbf{w}_i - \mathbf{Q}_{i-1} [\mathbf{R}]_{(1:i-1,i)}.$ # macheps: u_{crs}

4. Sketch \mathbf{q}'_i : $\mathbf{s}'_i = \Theta \mathbf{q}'_i$. # macheps: u_{fine}

5. Compute the sketched norm $r_{i,i} = \|\mathbf{s}'_i\|.$ # macheps: u_{fine}

6. Scale vector $\mathbf{s}_i = \mathbf{s}'_i / r_{i,i}.$ # macheps: u_{fine}

7. Scale vector $\mathbf{q}_i = \mathbf{q}'_i / r_{i,i}.$ # macheps: u_{fine}

end for

8. (Optional) compute $\Delta_m = \|\mathbf{I}_{m \times m} - \mathbf{S}_m^T \mathbf{S}_m\|_F$ and $\tilde{\Delta}_m = \frac{\|\mathbf{P}_m - \mathbf{S}_m \mathbf{R}_m\|_F}{\|\mathbf{P}_m\|_F}.$

Use Theorem 3.2 to certify the output. # macheps: u_{fine}

$\mathbf{W} = \mathbf{Q}\mathbf{R}$, $\Theta \mathbf{W} = \Theta \mathbf{Q}\mathbf{R}$, $(\Theta \mathbf{Q})^\top (\Theta \mathbf{Q}) = \mathbf{I}_m$, $\text{cond}(\mathbf{Q})$ is small

O. Balabanov and L. Grigori. *Randomized Gram–Schmidt process with application to GM-RES*. SISC, Vol. 44, No. 3 (2022)

Randomized GMRES (rGMRES) for $\mathbf{Ax} = \mathbf{b}$

Algorithm 4.1. RGS-Arnoldi algorithm

Given: $n \times n$ matrix \mathbf{A} , $n \times 1$ vector \mathbf{b} , $k \times n$ matrix Θ with $k \ll n$, parameter m .

Output: $n \times m$ factor \mathbf{Q}_m and $m \times m$ upper triangular factor \mathbf{R}_m .

1. Set $\mathbf{w}_1 = \mathbf{b}$.

2. Perform 1st iteration of Algorithm 2.1.

for $i = 2 : m$ **do**

3. Compute $\mathbf{w}_i = \mathbf{A}\mathbf{q}_{i-1}$.

macheps: u_{fine}

4. Perform i th iteration of Algorithm 2.1.

end for

5. (Optional) Compute Δ_m and $\tilde{\Delta}_m$.

Use Proposition 4.1 to certify the output.

macheps: u_{fine}

- $(\Theta\mathbf{Q}_m)^\top(\Theta\mathbf{Q}_m) = \mathbf{I}_m$ and \mathbf{Q}_m is not orthonormal.
- rGMRES: $\mathbf{x}_m = \mathbf{Q}_m\mathbf{y}_m$, where \mathbf{y}_m solves

$$\min_{\mathbf{y}} \|\mathbf{H}_{m+1,m}\mathbf{y} - r_{11}\mathbf{e}_1\|_2.$$

- $\|\mathbf{r}^{\text{GMRES}}\|_2 \leq \|\mathbf{r}^{\text{rGMRES}}\|_2 \leq \text{const} \cdot \|\mathbf{r}^{\text{GMRES}}\|_2 \quad \text{whp.}$

Sketched GMRES (sGMRES) for $\mathbf{Ax} = \mathbf{b}$

- Columns of \mathbf{B} form a basis of the Krylov subspace $\mathcal{K}_j(\mathbf{A}, \mathbf{b})$.
- GMRES: $\mathbf{x}_B = \mathbf{B}\mathbf{y}_*$, where Arnoldi process is used for \mathbf{B} and \mathbf{y}_* solves the overdetermined least-squares problem

$$\min_{\mathbf{y}} \|\mathbf{A}\mathbf{B}\mathbf{y} - \mathbf{b}\|_2.$$

- sGMRES uses “sketch-and-solve” and is faster: $\hat{\mathbf{x}} = \mathbf{B}\hat{\mathbf{y}}$, where \mathbf{B} can be a nonorthogonal basis (for example, using k -truncated Arnoldi) and $\hat{\mathbf{y}}$ solves the sketched problem

$$\min_{\mathbf{y}} \|\mathbf{S}(\mathbf{A}\mathbf{B}\mathbf{y} - \mathbf{b})\|_2.$$

- $\|\mathbf{r}^{\text{GMRES}}\|_2 \leq \|\mathbf{r}^{\text{sGMRES}}\|_2 \leq \text{const} \cdot \|\mathbf{r}^{\text{GMRES}}\|_2 \quad \text{whp.}$

sGMRES + k -truncated Arnoldi

Algorithm 1.1. sGMRES + k -truncated Arnoldi.

Input: Matrix $\mathbf{A} \in \mathbb{C}^{n \times n}$, right-hand side $\mathbf{f} \in \mathbb{C}^n$, initial guess $\mathbf{x} \in \mathbb{C}^n$, basis dimension d , number k of vectors for truncated orthogonalization, stability tolerance $\text{tol} = O(u^{-1})$.

Output: Approximate solution $\hat{\mathbf{x}} \in \mathbb{C}^n$ to linear system (1.5) and estimated residual norm \hat{r}_{est}

```
1 function sGMRES
2   Draw subspace embedding  $\mathbf{S} \in \mathbb{C}^{s \times n}$  with  $s = 2(d+1)$            ▷ See subsection 2.4
3   Form residual and sketch:  $\mathbf{r} = \mathbf{f} - \mathbf{A}\mathbf{x}$  and  $\mathbf{g} = \mathbf{S}\mathbf{r}$ 
4   Normalize basis vector  $\mathbf{b}_1 = \mathbf{r}/\|\mathbf{r}\|_2$  and apply matrix  $\mathbf{m}_1 = \mathbf{A}\mathbf{b}_1$ 
5   for  $j = 2, 3, 4, \dots, d$  do                                         ▷ See also subsection 5.2
6     Truncated Arnoldi:  $\mathbf{w}_j = (\mathbf{I} - \mathbf{b}_{j-1}\mathbf{b}_{j-1}^* - \dots - \mathbf{b}_{j-k}\mathbf{b}_{j-k}^*)\mathbf{m}_{j-1}$    ▷  $\mathbf{b}_{-i} = \mathbf{0}$  for  $i \geq 0$ 
7     Normalize basis vector  $\mathbf{b}_j = \mathbf{w}_j/\|\mathbf{w}_j\|_2$  and apply matrix  $\mathbf{m}_j = \mathbf{A}\mathbf{b}_j$ 
8     Sketch reduced matrix:  $\mathbf{C} = \mathbf{S}[\mathbf{m}_1, \dots, \mathbf{m}_d]$ 
9     Thin QR factorization:  $\mathbf{C} = \mathbf{U}\mathbf{T}$ 
10    if condition number  $\kappa_2(\mathbf{T}) > \text{tol}$  then warning...
11    Either whiten  $\mathbf{B} \leftarrow \mathbf{B}\mathbf{T}^{-1}$  or form new residual and restart   ▷ See subsection 5.3
12    Solve least-squares problem:  $\hat{\mathbf{y}} = \mathbf{T}^{-1}(\mathbf{U}^*\mathbf{g})$            ▷ See (3.7)
13    Residual estimate:  $\hat{r}_{\text{est}} = \|(\mathbf{I} - \mathbf{U}\mathbf{U}^*)\mathbf{g}\|_2$            ▷ See (3.8)
14    Construct solution:  $\hat{\mathbf{x}} = \mathbf{x} + [\mathbf{b}_1, \dots, \mathbf{b}_d]\hat{\mathbf{y}}$ 
```

Implementation: In line 6, use double Gram–Schmidt for stability. In line 9, the QR factorization may require pivoting. In lines 11–12, apply \mathbf{T}^{-1} via triangular substitution.

Summary

- We have presented nsLSQR for nonsymmetric saddle-point linear systems.
- nsLSQR is mathematically equivalent to GMRES applied to the corresponding Schur complement system, but may be numerically superior.
- nsLSQR usually is faster than SPMR-SC and SPQMR-SC in terms of the number of iterations, and if the iteration cost is dominated by the \mathbf{M} -solve rather than reorthogonalization, then nsLSQR should be the preferred method.
- The ideas of rGMRES and sGMRES can be used for GPMR and nsLSQR.
- Intelligent iterative methods for block two-by-two linear systems?

Thanks!