

TriCG with deflated restarting for symmetric quasi-definite linear systems

Kui Du

kuidu@xmu.edu.cn

School of Mathematical Sciences, Xiamen University

<https://kuidu.github.io>

joint work with **Jia-Jun Fan**

Outline

- ① Symmetric quasi-definite (SQD) linear systems
- ② The generalized Saunders–Simon–Yip tridiagonalization
- ③ TriCG for SQD linear systems
- ④ Linear systems, Krylov subspace methods, and deflation
- ⑤ TriCG with deflated restarting
- ⑥ Summary

Symmetric quasi-definite (SQD) linear systems

- $\mathbf{M} \in \mathbb{R}^{m \times m}$ and $\mathbf{N} \in \mathbb{R}^{n \times n}$ are SPD, $\mathbf{A} \in \mathbb{R}^{m \times n}$ is nonzero, $\mathbf{b} \in \mathbb{R}^m$, and $\mathbf{c} \in \mathbb{R}^n$:

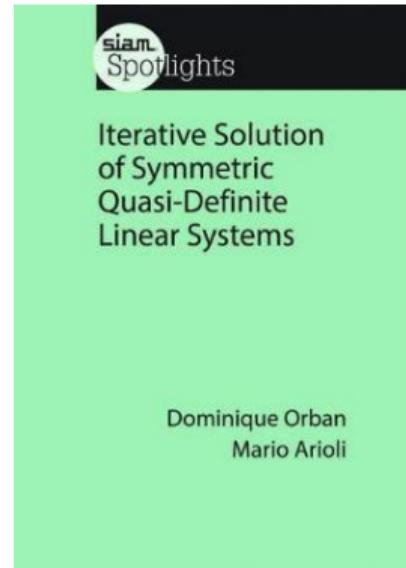
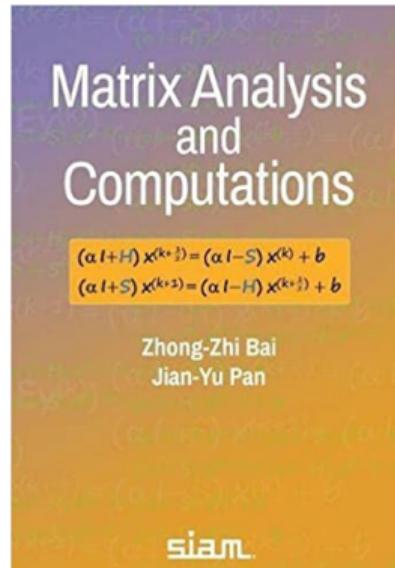
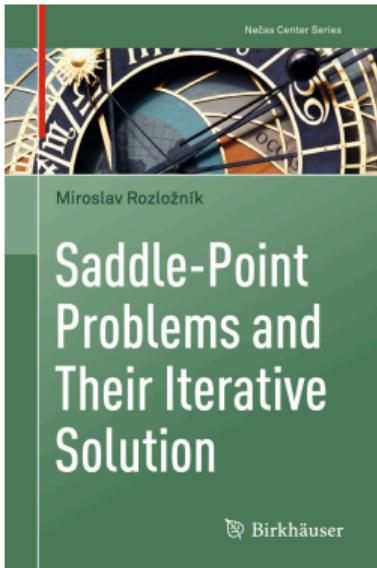
$$\begin{bmatrix} \mathbf{M} & \mathbf{A} \\ \mathbf{A}^\top & -\mathbf{N} \end{bmatrix} \begin{bmatrix} \mathbf{x} \\ \mathbf{y} \end{bmatrix} = \begin{bmatrix} \mathbf{b} \\ \mathbf{c} \end{bmatrix}, \quad \mathbf{H} = \begin{bmatrix} \mathbf{M} & \\ & \mathbf{N} \end{bmatrix}, \quad \mathbf{K} = \begin{bmatrix} \mathbf{M} & \mathbf{A} \\ \mathbf{A}^\top & -\mathbf{N} \end{bmatrix}.$$

- Computational optimization and computational partial differential equations, etc.
- Symmetric, indefinite, nonsingular
- **Monolithic** methods: solving the system as a whole, for example, SYMMLQ, MINRES

Segregated methods: exploiting the block structure, excluding the preconditioning stage, for example, TriCG, TriMR

Review paper and books

- Michele Benzi, Gene H. Golub, and Jörg Liesen
Numerical solution of saddle point problems. Acta Numerica (2005).
- Books



The generalized Saunders–Simon–Yip tridiagonalization

- The generalized Saunders–Simon–Yip tridiagonalization:

$$\mathbf{A}\mathbf{V}_k = \mathbf{M}\mathbf{U}_{k+1}\mathbf{T}_{k+1,k} = \mathbf{M}\mathbf{U}_k\mathbf{T}_k + \beta_{k+1}\mathbf{M}\mathbf{u}_{k+1}\mathbf{e}_k^\top,$$

$$\mathbf{A}^\top\mathbf{U}_k = \mathbf{N}\mathbf{V}_{k+1}\mathbf{T}_{k,k+1}^\top = \mathbf{N}\mathbf{V}_k\mathbf{T}_k^\top + \gamma_{k+1}\mathbf{N}\mathbf{v}_{k+1}\mathbf{e}_k^\top,$$

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

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

$$\mathbf{T}_k = \text{tridiag}(\beta_i, \alpha_i, \gamma_{i+1}), \quad \mathbf{T}_{k+1,k} = \begin{bmatrix} \mathbf{T}_k \\ \beta_{k+1}\mathbf{e}_k^\top \end{bmatrix}, \quad \mathbf{T}_{k,k+1} = [\mathbf{T}_k \quad \gamma_{k+1}\mathbf{e}_k].$$

The generalized Saunders–Simon–Yip tridiagonalization

Algorithm 1 Generalized Saunders–Simon–Yip tridiagonalization

Require: $\mathbf{A} \in \mathbb{R}^{m \times n}$, $\mathbf{b} \in \mathbb{R}^m$ and $\mathbf{c} \in \mathbb{R}^n$ are nonzero, $\mathbf{M} \in \mathbb{R}^{m \times m}$ and $\mathbf{N} \in \mathbb{R}^{n \times n}$ are SPD, subroutines for performing $\mathbf{M}^{-1}\mathbf{u}$ and $\mathbf{N}^{-1}\mathbf{v}$

- 1: $\mathbf{u}_0 = \mathbf{0}$, $\mathbf{v}_0 = \mathbf{0}$, $\beta_1 \mathbf{M}\mathbf{u}_1 = \mathbf{b}$, $\gamma_1 \mathbf{N}\mathbf{v}_1 = \mathbf{c}$,
 - 2: **for** $k = 1, 2, \dots$ **do**
 - 3: $\mathbf{p} = \mathbf{A}\mathbf{v}_k - \gamma_k \mathbf{M}\mathbf{u}_{k-1}$, $\alpha_k = \mathbf{u}_k^\top \mathbf{p}$,
 - 4: $\beta_{k+1} \mathbf{M}\mathbf{u}_{k+1} = \mathbf{p} - \alpha_k \mathbf{M}\mathbf{u}_k$, $\gamma_{k+1} \mathbf{N}\mathbf{v}_{k+1} = \mathbf{A}^\top \mathbf{u}_k - \beta_k \mathbf{N}\mathbf{v}_{k-1} - \alpha_k \mathbf{N}\mathbf{v}_k$,
 - 5: **if** $\beta_{k+1} = 0$ or $\gamma_{k+1} = 0$, **terminate**
 - 6: **end for**
-

- Assume that \mathbf{U}_k , \mathbf{V}_k , and \mathbf{T}_k are well defined. The k th TriCG iterate is

$$\begin{bmatrix} \mathbf{x}_k \\ \mathbf{y}_k \end{bmatrix} = \begin{bmatrix} \mathbf{U}_k & \mathbf{0} \\ \mathbf{0} & \mathbf{V}_k \end{bmatrix} \begin{bmatrix} \mathbf{I}_k & \mathbf{T}_k \\ \mathbf{T}_k^\top & -\mathbf{I}_k \end{bmatrix}^{-1} \begin{bmatrix} \beta_1 \mathbf{e}_1 \\ \gamma_1 \mathbf{e}_1 \end{bmatrix},$$

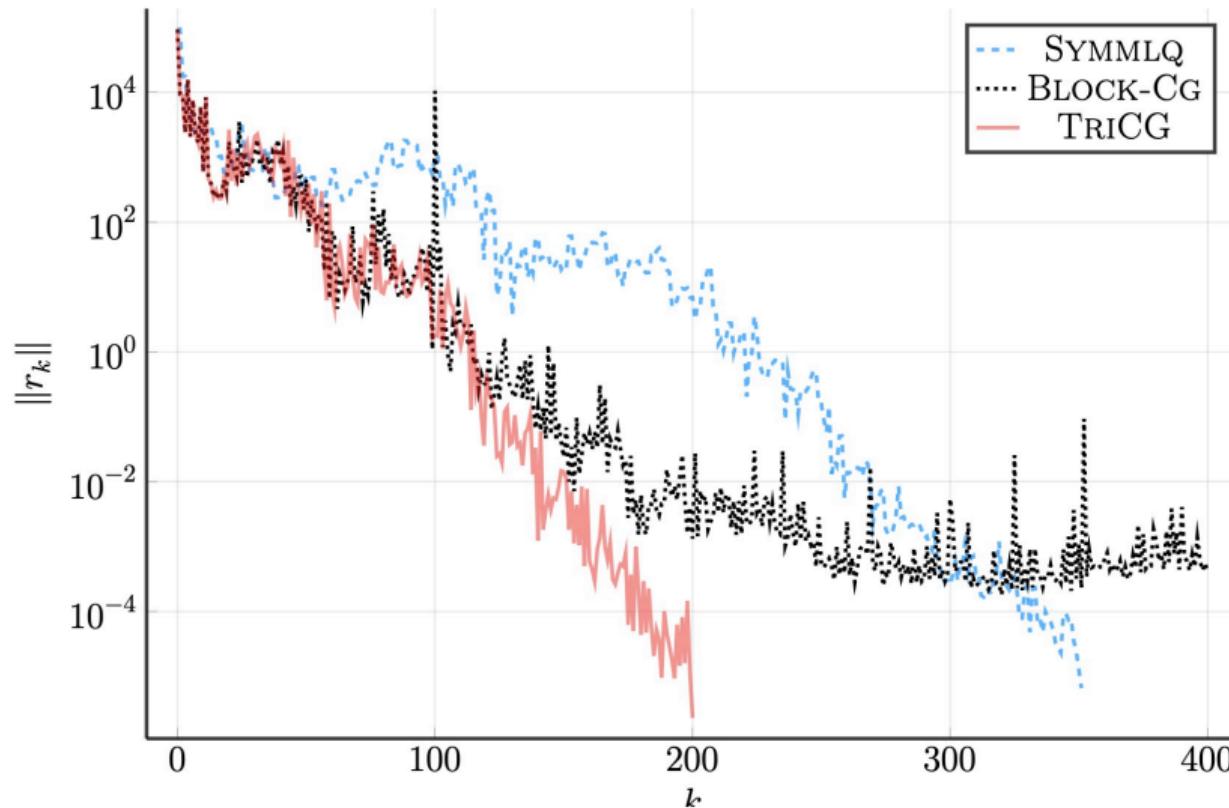
which satisfies the Galerkin condition

$$\begin{bmatrix} \mathbf{U}_k & \mathbf{0} \\ \mathbf{0} & \mathbf{V}_k \end{bmatrix}^\top \left(\begin{bmatrix} \mathbf{b} \\ \mathbf{c} \end{bmatrix} - \begin{bmatrix} \mathbf{M} & \mathbf{A} \\ \mathbf{A}^\top & -\mathbf{N} \end{bmatrix} \begin{bmatrix} \mathbf{x}_k \\ \mathbf{y}_k \end{bmatrix} \right) = \mathbf{0}.$$

- TriCG is faster than SYMMLQ and is equivalent to preconditioned Block-CG:

$$\begin{bmatrix} \mathbf{M} & \mathbf{A} \\ \mathbf{A}^\top & -\mathbf{N} \end{bmatrix} \begin{bmatrix} \mathbf{x}^1 & \mathbf{x}^2 \\ \mathbf{y}^1 & \mathbf{y}^2 \end{bmatrix} = \begin{bmatrix} \mathbf{b} & \mathbf{0} \\ \mathbf{0} & \mathbf{c} \end{bmatrix}.$$

Example: $M = I$, $N = I$, $A = 1p_osa_07$



Elliptic singular value decomposition (ESVD)

- Given SPD \mathbf{M} and \mathbf{N} , ESVD of \mathbf{A} is

$$\mathbf{A} = \mathbf{M}\mathbf{P}\Sigma\mathbf{Q}^\top\mathbf{N},$$

where $\Sigma = \text{diag}(\sigma_1, \dots, \sigma_p)$, $\sigma_1 \geq \sigma_2 \geq \dots \geq \sigma_p \geq 0$, $p = \min(m, n)$, and \mathbf{P} and \mathbf{Q} satisfy

$$\mathbf{P}^\top\mathbf{M}\mathbf{P} = \mathbf{I}_m, \quad \mathbf{Q}^\top\mathbf{N}\mathbf{Q} = \mathbf{I}_n.$$

- Eigenvalues of a two-sided preconditioned matrix (let $r = \text{rank}(\mathbf{A})$):

$$\lambda \left(\mathbf{H}^{-\frac{1}{2}} \mathbf{K} \mathbf{H}^{-\frac{1}{2}} \right) = \begin{cases} \pm \sqrt{\sigma_i^2 + 1}, & i = 1, \dots, r, \\ 1, & (m - r) \text{ times}, \\ -1, & (n - r) \text{ times}. \end{cases}$$

Linear systems, Krylov subspace methods, and deflation

- Linear systems of equations

$$\mathbf{Ax} = \mathbf{b}, \quad \mathbf{A} \in \mathbb{R}^{m \times m}, \quad \mathbf{b} \in \mathbb{R}^m$$

- Krylov subspace methods

CG, MINRES; GMRES, CMRH, Bi-CG, QMR, Bi-CGSTAB ...

- Acceleration techniques

preconditioning, randomization, inexact or mixed-precision, inner-product free (orthogonalization-free), **deflation** ...

- When solving linear systems, deflation refers to reducing the influence of some eigenvalues that tend to slow convergence. Deflation can be implemented
 - (1) by adding approximate eigenvectors to a subspace, or
 - (2) by building a preconditioner from eigenvectors.

Some solvers incorporating augmentation-based deflation

- Nonsymmetric linear systems: Arnoldi + augmentation + restart
FOM-IR, GMRES-IR ([Morgan, SIMAX, 2000](#)), FOM-DR, GMRES-DR
([Morgan, SISC, 2002](#))

$$\mathbf{AV}_m^{(i)} = \mathbf{V}_{m+1}^{(i)} \mathbf{H}_{m+1,m}^{(i)}, \quad i = 1, 2, \dots$$

- Symmetric linear systems: Lanczos + augmentation + restart
Lanczos-DR, MINRES-DR ([Abdel-Rehim et al., SISC, 2010](#))

$$\mathbf{AV}_m^{(i)} = \mathbf{V}_{m+1}^{(i)} \mathbf{T}_{m+1,m}^{(i)}, \quad i = 1, 2, \dots$$

- Symmetric saddle point linear systems: Golub–Kahan + augmentation + restart
Augmented LSQR ([Baglama, Reichel, and Richmond, NA, 2013](#))
Augmented CRAIG ([Dumitras, Kruse, and Rüde, SIMAX, 2024](#))

$$\mathbf{AV}_m^{(i)} = \mathbf{U}_{m+1}^{(i)} \mathbf{B}_{m+1,m}^{(i)}, \quad \mathbf{A}^\top \mathbf{U}_{m+1}^{(i)} = \mathbf{V}_{m+1}^{(i)} (\mathbf{B}_{m+1}^{(i)})^\top, \quad i = 1, 2, \dots$$

A gSSY process with deflated restarting

- gSSY-DR(p, k):

$$\mathbf{A}\mathbf{V}_p^{(i)} = \mathbf{M}\mathbf{U}_p^{(i)}\mathbf{T}_p^{(i)} + \beta_{p+1}^{(i)}\mathbf{M}\mathbf{u}_{p+1}^{(i)}\mathbf{e}_p^\top,$$

$$\mathbf{A}^\top\mathbf{U}_p^{(i)} = \mathbf{N}\mathbf{V}_p^{(i)}(\mathbf{T}_p^{(i)})^\top + \gamma_{p+1}^{(i)}\mathbf{N}\mathbf{v}_{p+1}^{(i)}\mathbf{e}_p^\top.$$

For $i = 2, 3, \dots$,

$$\mathbf{T}_p^{(i)} = \begin{bmatrix} \alpha_1^{(i)} & & & & \gamma_2^{(i)} & & & \\ & \ddots & & & \vdots & & & \\ & & \ddots & & \gamma_{k+1}^{(i)} & & & \\ \beta_2^{(i)} & \dots & \beta_{k+1}^{(i)} & \alpha_{k+1}^{(i)} & \gamma_{k+2}^{(i)} & & & \\ & & & \beta_{k+2}^{(i)} & \alpha_{k+2}^{(i)} & \ddots & & \\ & & & & \ddots & \ddots & \gamma_p^{(i)} & \\ & & & & & & \beta_p^{(i)} & \alpha_p^{(i)} \end{bmatrix}.$$

TriCG with deflated restarting

- The recurrences in the first cycle are the same as that of TriCG. Now consider cycle $i \geq 2$. The j th ($k+1 \leq j \leq p$) TriCG-DR(p, k) iterate is

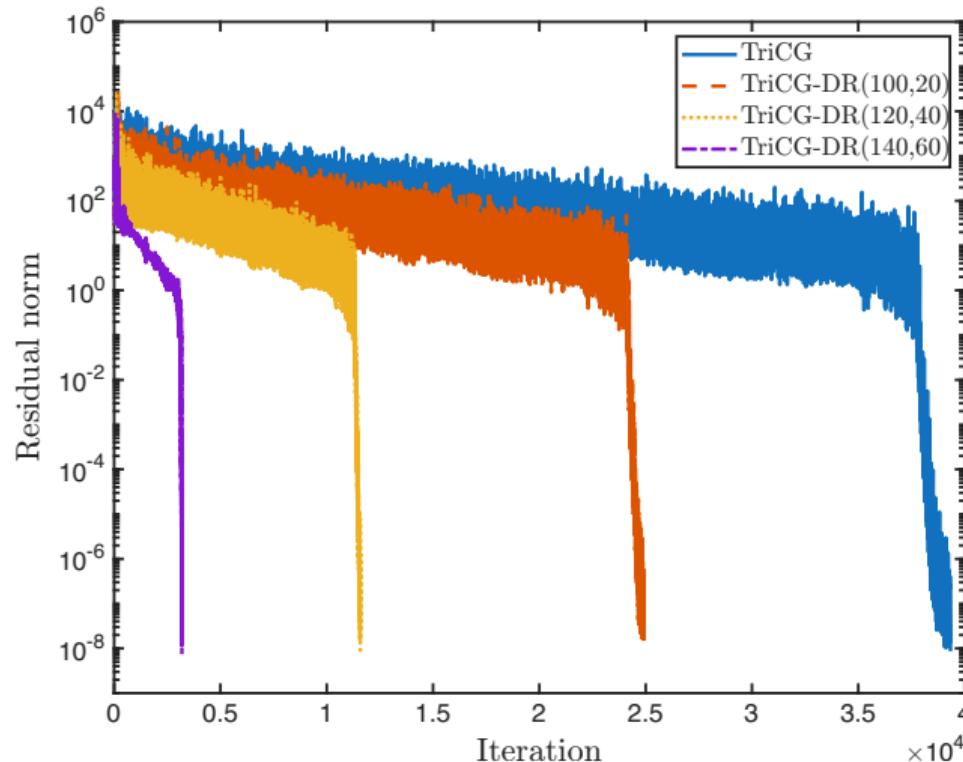
$$\begin{bmatrix} \mathbf{x}_j^{(i)} \\ \mathbf{y}_j^{(i)} \end{bmatrix} = \begin{bmatrix} \mathbf{U}_j^{(i)} & \mathbf{0} \\ \mathbf{0} & \mathbf{V}_j^{(i)} \end{bmatrix} \begin{bmatrix} \mathbf{I}_j & \mathbf{T}_j^{(i)} \\ (\mathbf{T}_j^{(i)})^\top & -\mathbf{I}_j \end{bmatrix}^{-1} \begin{bmatrix} \beta_1^{(i)} \mathbf{e}_{k+1} \\ \gamma_1^{(i)} \mathbf{e}_{k+1} \end{bmatrix},$$

which satisfies the Galerkin condition.

- Using an LDL^\top decomposition and the same strategy in TriCG, short recurrences can be obtained to compute $\mathbf{x}_j^{(i)}$ and $\mathbf{y}_j^{(i)}$ for $k+1 \leq j \leq p$.
- If the target k approximate elliptic singular triplets are sufficiently accurate, we stop restarting. In other words, the last cycle is implemented completely until a sufficiently accurate approximate solution is found or the maximum number of iterations is reached. **Partial reorthogonalization is required.**

Numerical experiment I

```
 $M = I$ ,  $N = I$ ,  $A = \text{diag}([\text{linspace}(0, 800, 2000), \text{linspace}(1e3, 1e5, 60)])$ ;
```



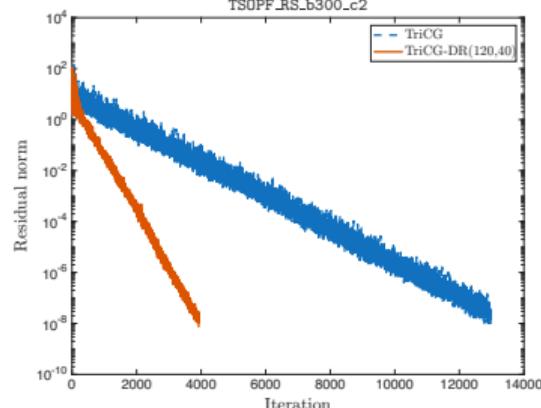
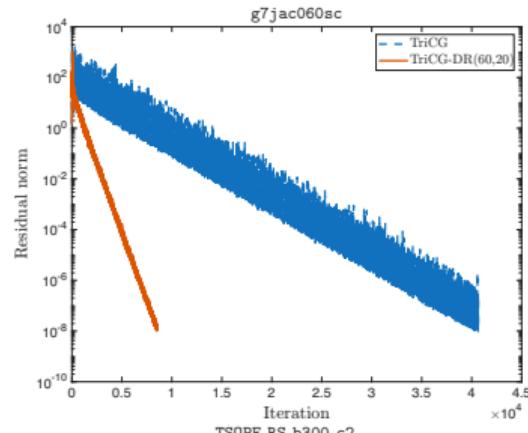
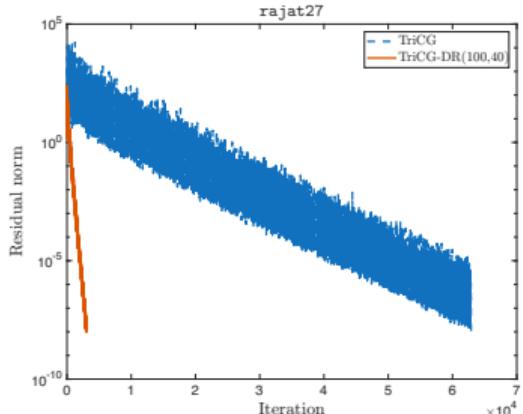
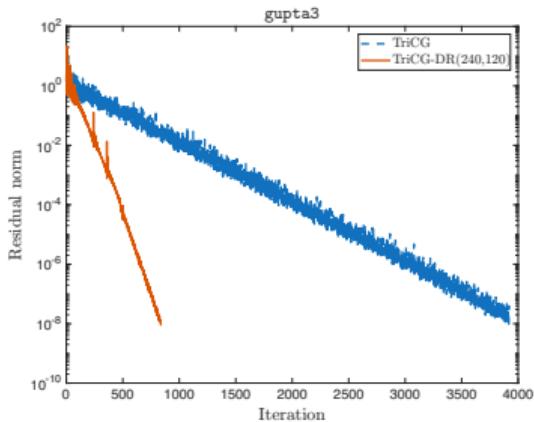
Numerical experiment II

$\mathbf{M} = \mathbf{I}$, $\mathbf{N} = \mathbf{I}$, \mathbf{A} is from the SuiteSparse Matrix Collection.

Table: The information of square matrices from the SuiteSparse Matrix Collection, runtime of TriCG and TriCG-DR, and values of parameters p and k of TriCG-DR.

Matrix	Size	Nnz	TriCG	TriCG-DR		
			Time(s)	Time(s)	p	k
gupta3	16783	9323427	17.55	7.61	240	120
g7jac060sc	17730	183325	16.82	10.10	60	20
rajab27	20640	97353	24.70	6.42	100	40
TSOPF_RS_b300_c2	28338	2943887	30.48	17.64	120	40

Numerical experiment II



Numerical experiment III

- The matrices are obtained via IFISS on the test problem `channel_domain` for the unsteady incompressible Stokes equation.

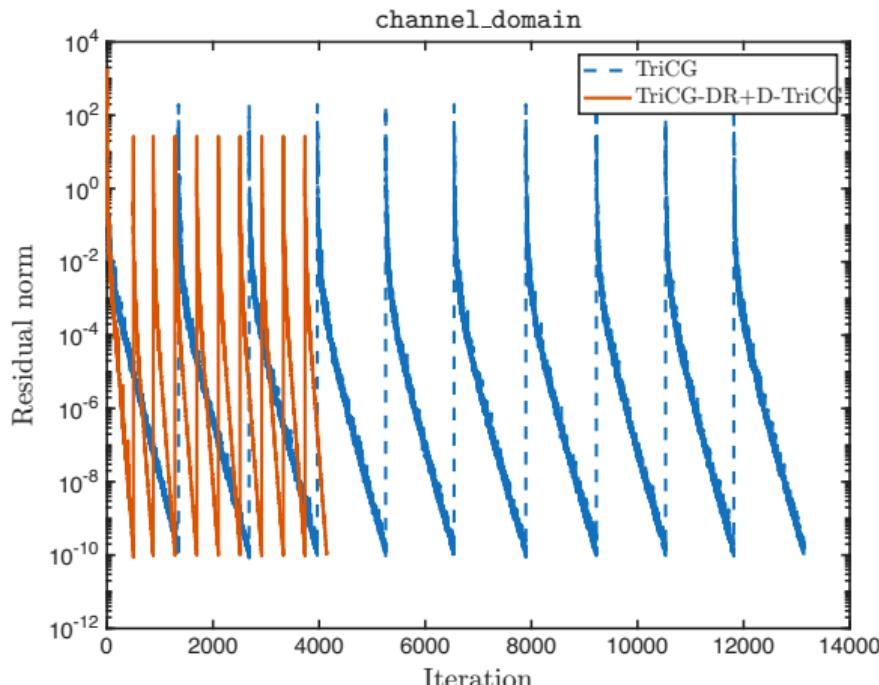
CPU time

TriCG

163.42

TriCG-DR+D-TriCG

108.85



Summary

- We have proposed TriCG-DR for solving symmetric quasi-definite linear systems. The new method has faster convergence and demonstrates a significant advantage in computational time.
- The gSSY-DR process can be used to compute partial spectral information. And the computed spectral information can be used to help solve linear systems with sequential multiple right-hand sides.

The related work

- Kui Du and Jia-Jun Fan

TriCG with deflated restarting for symmetric quasi-definite linear systems

In preparation, 2025.

- Kui Du, Jia-Jun Fan, and Ya-Lan Zhang

Improved TriCG and TriMR methods for symmetric quasi-definite linear systems

Numerical Linear Algebra with Applications, 2025, 32:e70026.

Thanks!