

GP-CMRH: An inner product free iterative method for block two-by-two nonsymmetric linear systems

Kui Du

kuidu@xmu.edu.cn

School of Mathematical Sciences, Xiamen University

<https://kuidu.github.io>

joint work with **Jia-Jun Fan**

Hunan Normal University, Changsha, January 20, 2026

Outline

- ① Iterative Krylov methods for large linear systems
- ② GMRES and CMRH
- ③ Block two-by-two linear systems
- ④ GPMR
- ⑤ GP-CMRH
- ⑥ Concluding remarks

Linear systems and iterative Krylov methods

- Linear systems of equations

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

- Iterative Krylov methods

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

- Some research hotspots

- (1) new preconditioning techniques (e.g., operator learning, NN, sketching)
- (2) randomization techniques (e.g., rGMRES, sGMRES)
- (3) inexact or mixed-precision computations
- (4) inner-product free (orthogonalization-free) algorithms (e.g., CMRH)

The Arnoldi process and GMRES

- Krylov subspaces: Let $\mathbf{r}_0 := \mathbf{b} - \mathbf{A}\mathbf{x}_0$,

$$\mathcal{K}_k(\mathbf{A}, \mathbf{r}_0) := \text{span}\{\mathbf{r}_0, \mathbf{A}\mathbf{r}_0, \dots, \mathbf{A}^{k-1}\mathbf{r}_0\}.$$

- The Arnoldi process generates an orthonormal basis

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

via the modified Gram–Schmidt (MGS) orthogonalization process. We have the QR factorization

$$[\mathbf{r}_0 \quad \mathbf{A}\mathbf{V}_k] = \mathbf{V}_{k+1} [\|\mathbf{r}_0\| \mathbf{e}_1 \quad \mathbf{H}_{k+1,k}] .$$

- The generalized minimal residual (GMRES) method:

$$\mathbf{x}_k := \underset{\mathbf{x} \in \mathbf{x}_0 + \mathcal{K}_k(\mathbf{A}, \mathbf{r}_0)}{\operatorname{argmin}} \|\mathbf{b} - \mathbf{A}\mathbf{x}\|.$$

The Hessenberg process and CMRH

- The Hessenberg process generates a linearly independent basis for $\mathcal{K}_k(\mathbf{A}, \mathbf{r}_0)$

$$\mathbf{L}_k = [\ell_1 \ \ell_2 \ \cdots \ \ell_k].$$

We have the “LU factorization”

$$[\mathbf{r}_0 \ \ \mathbf{A}\mathbf{L}_k] = \mathbf{L}_{k+1} \begin{bmatrix} \mathbf{e}_1^\top \mathbf{r}_0 \mathbf{e}_1 & \tilde{\mathbf{H}}_{k+1,k} \end{bmatrix}.$$

Sometimes, pivoting is necessary. Usually, it is “better” than the Krylov basis.

- The changing minimal residual Hessenberg (CMRH) method:

$$\mathbf{x}_k := \underset{\mathbf{x} \in \mathbf{x}_0 + \mathcal{K}_k(\mathbf{A}, \mathbf{r}_0)}{\operatorname{argmin}} \|\mathbf{L}_{k+1}^\dagger(\mathbf{b} - \mathbf{Ax})\|.$$

- CMRH is inner-product-free, less expensive and requires slightly less storage than GMRES.

Block two-by-two linear systems

- Block two-by-two linear systems of the form

$$\begin{bmatrix} \mathbf{M} & \mathbf{A} \\ \mathbf{B} & \mathbf{N} \end{bmatrix} \begin{bmatrix} \mathbf{x} \\ \mathbf{y} \end{bmatrix} = \begin{bmatrix} \mathbf{b} \\ \mathbf{c} \end{bmatrix}, \quad \mathbf{M} \in \mathbb{R}^{m \times m}, \quad \mathbf{N} \in \mathbb{R}^{n \times n}.$$

- **Monolithic** methods: solving the system as a whole.

For example: CG, MINRES; GMRES, Bi-CG, QMR, Bi-CGSTAB, CMRH ...

Segregated methods: exploiting the block structure, but not in preconditioning.

For example: LSQR, LSMR, TriCG, TriMR; **GPMR**, GPBiLQ, GPQMR ...

- We consider a special case: $\mathbf{A} \neq \mathbf{B}^\top$, $\lambda \in \mathbb{R}$, $\mu \in \mathbb{R}$,

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

Simultaneous orthogonal Hessenberg reduction for (A, B)

- Simultaneous orthogonal Hessenberg reduction

$$AU_k = V_{k+1}H_{k+1,k}, \quad BV_k = U_{k+1}F_{k+1,k},$$

$$V_{k+1}^\top V_{k+1} = U_{k+1}^\top U_{k+1} = I_{k+1},$$

where

$$H_{k+1,k} = \begin{bmatrix} h_{11} & \cdots & h_{1k} \\ h_{21} & \ddots & \vdots \\ \ddots & & h_{kk} \\ & & h_{k+1,k} \end{bmatrix}, \quad F_{k+1,k} = \begin{bmatrix} f_{11} & \cdots & f_{1k} \\ f_{21} & \ddots & \vdots \\ \ddots & & f_{kk} \\ & & f_{k+1,k} \end{bmatrix}.$$

Simultaneous orthogonal Hessenberg reduction for (\mathbf{A}, \mathbf{B})

Algorithm 1: Simultaneous orthogonal Hessenberg reduction

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

- 1: $\beta \mathbf{v}_1 := \mathbf{b}, \gamma \mathbf{u}_1 := \mathbf{c}$ $\beta > 0, \gamma > 0$ so that $\|\mathbf{v}_1\| = \|\mathbf{u}_1\| = 1$
 - 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: $h_{k+1,k} \mathbf{v}_{k+1} = \mathbf{A} \mathbf{u}_k - \sum_{i=1}^k h_{ik} \mathbf{v}_i$ $h_{k+1,k} > 0$ so that $\|\mathbf{v}_{k+1}\| = 1$
 - 8: $f_{k+1,k} \mathbf{u}_{k+1} = \mathbf{B} \mathbf{v}_k - \sum_{i=1}^k f_{ik} \mathbf{u}_i$ $f_{k+1,k} > 0$ so that $\|\mathbf{u}_{k+1}\| = 1$
 - 9: **end for**
-

- The k th GPMR iterate is

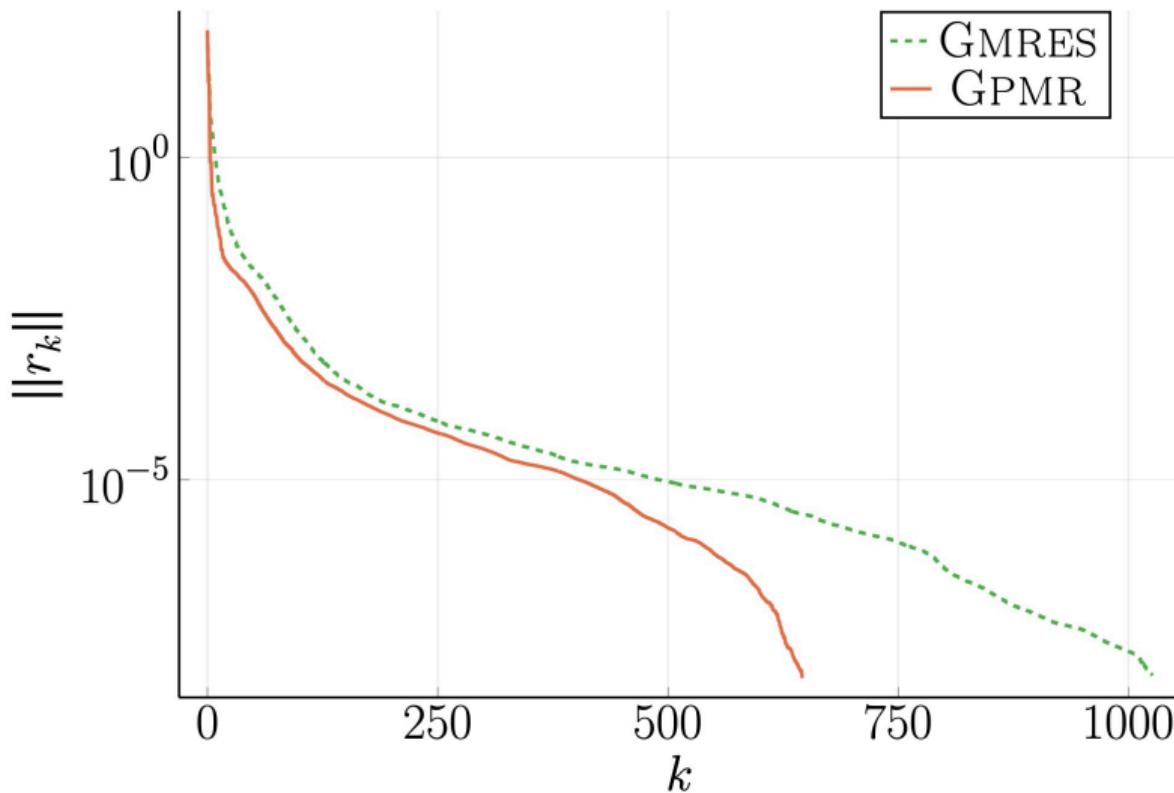
$$\begin{bmatrix} \mathbf{x}_k \\ \mathbf{y}_k \end{bmatrix} = \underset{\mathbf{x} \in \text{range}(\mathbf{V}_k), \mathbf{y} \in \text{range}(\mathbf{U}_k)}{\operatorname{argmin}} \left\| \begin{bmatrix} \mathbf{b} \\ \mathbf{c} \end{bmatrix} - \begin{bmatrix} \lambda \mathbf{I} & \mathbf{A} \\ \mathbf{B}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

- 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: $\mathbf{A} = \text{well1850}$, $\mathbf{B} = \text{illc1850}$, $\lambda = 1$, $\mu = 0$



Simultaneous Hessenberg reduction for (\mathbf{A}, \mathbf{B})

- Simultaneous Hessenberg reduction

$$\mathbf{A}\mathbf{L}_k = \mathbf{D}_{k+1}\tilde{\mathbf{H}}_{k+1,k}, \quad \mathbf{B}\mathbf{D}_k = \mathbf{L}_{k+1}\tilde{\mathbf{F}}_{k+1,k},$$

where

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

We have

$$\text{range}(\mathbf{D}_k) = \text{range}(\mathbf{V}_k), \quad \text{range}(\mathbf{L}_k) = \text{range}(\mathbf{U}_k).$$

Simultaneous Hessenberg reduction for (\mathbf{A}, \mathbf{B})

Algorithm 2: Simultaneous Hessenberg reduction with pivoting

Require: \mathbf{A} , \mathbf{B} , \mathbf{b} , \mathbf{c} , all nonzero, $\mathbf{p} = [1 \ 2 \ \cdots \ m]$, $\mathbf{q} = [1 \ 2 \ \cdots \ n]$

1: Determine i_0 and j_0 such that $|b_{i_0}| = \max_{1 \leq i \leq m} |\mathbf{e}_i^\top \mathbf{b}|$ and $|c_{j_0}| = \max_{1 \leq j \leq n} |\mathbf{e}_j^\top \mathbf{c}|$

2: $\beta = b_{i_0}$, $\mathbf{d}_1 = \mathbf{b}/\beta$, $\gamma = c_{j_0}$, $\ell_1 = \mathbf{c}/\gamma$, $\mathbf{p}(1) \leftarrow \mathbf{p}(i_0)$, $\mathbf{q}(1) \leftarrow \mathbf{q}(j_0)$

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

4: $\mathbf{d} = \mathbf{A}\ell_k$, $\ell = \mathbf{B}\mathbf{d}_k$

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

6: $h_{i,k} = \mathbf{d}(\mathbf{p}(i))$, $f_{i,k} = \ell(\mathbf{q}(i))$, $\mathbf{d} = \mathbf{d} - h_{i,k}\mathbf{d}_i$, $\ell = \ell - f_{i,k}\ell_i$

7: **end for**

8: Determine i_0, j_0 such that $|d_{i_0}| = \max_{k+1 \leq i \leq m} |\mathbf{d}(\mathbf{p}(i))|$ and $|\ell_{j_0}| = \max_{k+1 \leq j \leq n} |\ell(\mathbf{q}(j))|$

9: $h_{k+1,k} = d_{i_0}$, $\mathbf{d}_{k+1} = \mathbf{d}/h_{k+1,k}$, $f_{k+1,k} = \ell_{j_0}$, $\ell_{k+1} = \ell/f_{k+1,k}$
 $\mathbf{p}(k+1) \leftarrow \mathbf{p}(i_0)$, $\mathbf{q}(k+1) \leftarrow \mathbf{q}(j_0)$

10: **end for**

- The k th GP-CMRH iterate is

$$\begin{bmatrix} \mathbf{x}_k \\ \mathbf{y}_k \end{bmatrix} = \underset{\mathbf{x} \in \text{range}(\mathbf{D}_k), \mathbf{y} \in \text{range}(\mathbf{L}_k)}{\operatorname{argmin}} \left\| \begin{bmatrix} \mathbf{D}_{k+1} & \\ & \mathbf{L}_{k+1} \end{bmatrix}^\dagger \left(\begin{bmatrix} \mathbf{b} \\ \mathbf{c} \end{bmatrix} - \begin{bmatrix} \lambda \mathbf{I} & \mathbf{A} \\ \mathbf{B} & \mu \mathbf{I} \end{bmatrix} \begin{bmatrix} \mathbf{x} \\ \mathbf{y} \end{bmatrix} \right) \right\|.$$

Theorem

Let $\mathbf{r}_k^{\text{GP-CMRH}}$ and $\mathbf{r}_k^{\text{GPMR}}$ be the k th residuals of GP-CMRH and GPMR, respectively. Let $\mathbf{W}_{k+1} = \begin{bmatrix} \mathbf{D}_{k+1} & \\ & \mathbf{L}_{k+1} \end{bmatrix}$. Then,

$$\|\mathbf{r}_k^{\text{GPMR}}\| \leq \|\mathbf{r}_k^{\text{GP-CMRH}}\| \leq \kappa(\mathbf{W}_{k+1}) \|\mathbf{r}_k^{\text{GPMR}}\|,$$

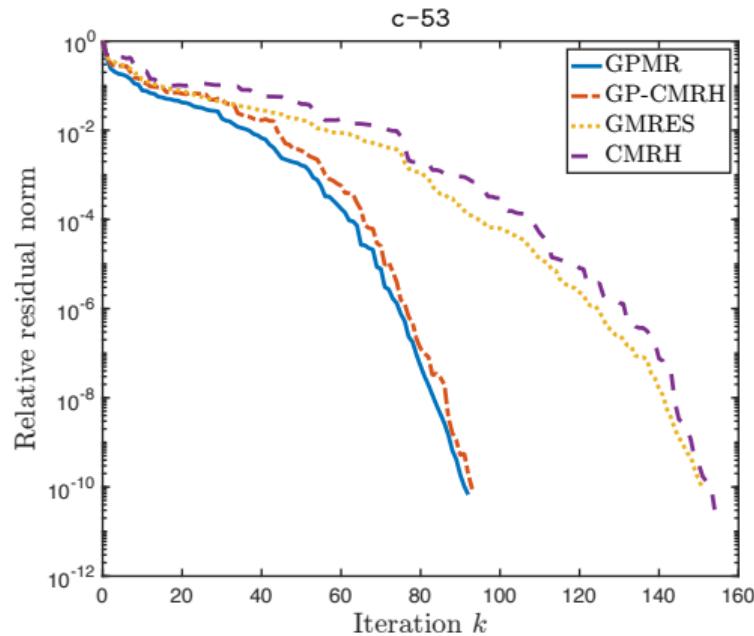
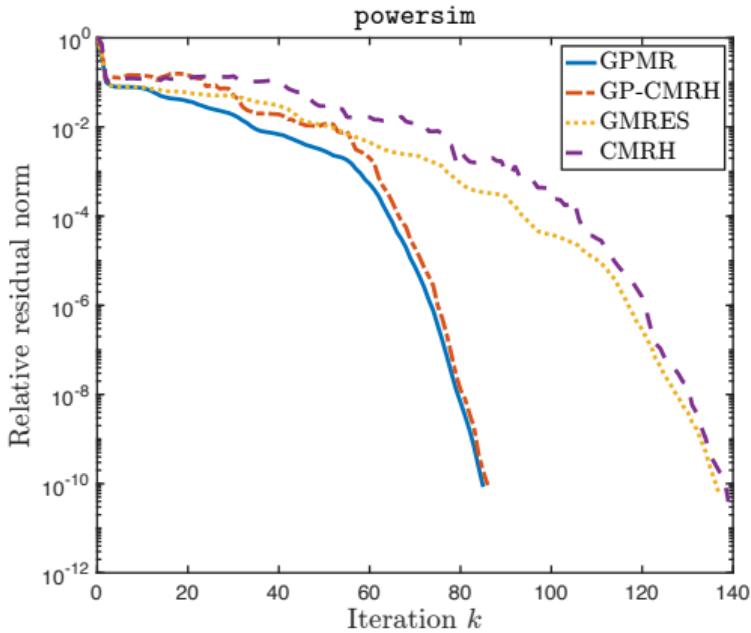
where $\kappa(\mathbf{W}_{k+1}) = \|\mathbf{W}_{k+1}\| \|\mathbf{W}_{k+1}^\dagger\|$ is the condition number of \mathbf{W}_{k+1} .

Numerical experiments

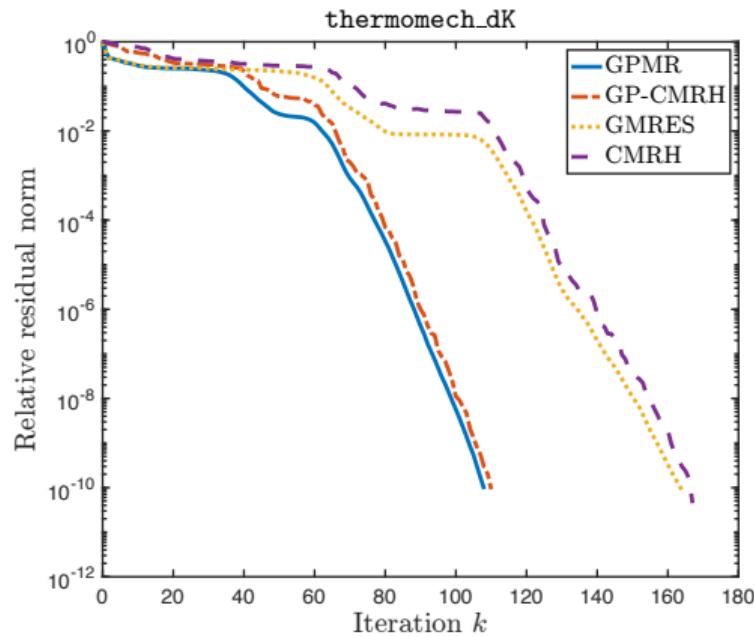
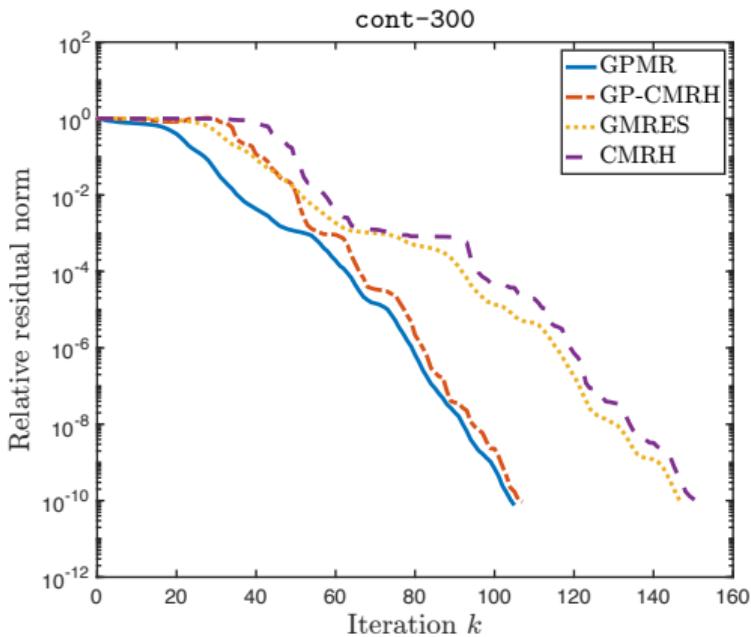
Table 1: Numbers of iterations (Iter), runtimes (Time), and relative residual norms (Rel) of GPMR, GP-CMRH, GMRES, and CMRH on twenty-two matrices from the SuiteSparse Matrix Collection. “Nnz” denotes the number of nonzero elements in each sparse matrix. Bold-faced values in the runtime column highlight the shortest time taken among the four methods.

Name	Size	Nnz	GPMR			GP-CMRH			GMRES			CMRH		
			Iter	Time	Rel	Iter	Time	Rel	Iter	Time	Rel	Iter	Time	Rel
bcsstk17	10974	428650	121	0.39	3.66e-11	121	0.28	8.51e-11	213	1.57	8.17e-11	216	0.72	8.64e-11
bcsstk25	15439	252241	62	0.18	5.54e-11	63	0.15	8.63e-11	100	0.47	8.13e-11	116	0.37	8.41e-11
powersim	15838	64424	85	0.26	8.40e-11	86	0.20	9.28e-11	137	1.17	5.64e-11	139	0.42	3.91e-11
raefsky3	21200	1488768	37	0.68	9.56e-11	40	0.69	8.86e-11	63	1.24	8.59e-11	67	1.15	9.28e-11
sme3Db	29067	2081063	65	2.42	6.60e-11	66	2.23	7.30e-11	97	4.65	6.30e-11	98	3.27	9.40e-11
c-53	30235	355139	92	1.38	6.73e-11	93	1.21	8.85e-11	151	3.29	9.34e-11	154	2.17	3.08e-11
sme3Dc	42930	3148656	97	5.72	6.12e-11	98	5.36	7.67e-11	161	11.05	7.39e-11	163	8.75	6.85e-11
bcsstk39	46772	2060662	205	5.72	7.54e-11	209	3.25	7.33e-11	381	23.32	9.73e-11	392	5.39	9.50e-11
rma10	46835	2329092	41	1.43	6.39e-11	42	1.33	6.34e-11	49	1.69	7.02e-11	51	1.53	5.08e-11
copter2	55476	759952	211	19.86	7.38e-11	214	16.11	9.18e-11	367	50.06	7.04e-11	371	27.06	6.81e-11
Goodwin_071	56021	1797934	70	2.77	8.93e-11	72	2.39	7.56e-11	88	4.24	8.20e-11	91	2.94	9.34e-11
water_tank	60740	2035281	324	46.00	8.07e-11	338	35.09	7.20e-11	430	75.59	9.73e-11	464	51.15	8.34e-11
venkat50	62424	1717777	34	1.06	5.23e-11	35	0.97	6.24e-11	46	1.66	7.99e-11	48	1.29	4.17e-11
poisson3Db	85623	2374949	50	7.57	6.94e-11	51	7.64	8.84e-11	57	8.76	6.66e-11	59	9.00	5.97e-11
ifiss_mat	96307	3599932	33	2.27	8.76e-11	35	2.32	3.38e-11	42	3.03	7.08e-11	43	2.73	9.70e-11
hcircuit	105676	513072	46	0.80	9.69e-11	46	0.38	7.84e-11	58	1.44	4.99e-11	58	0.49	8.66e-11
PR02R	161070	8185136	61	25.13	8.31e-11	64	26.39	5.15e-11	100	42.66	8.91e-11	105	46.23	4.92e-11
cont-300	180895	988195	105	24.34	7.55e-11	107	21.57	9.34e-11	147	39.66	8.68e-11	151	32.55	9.49e-11
thermomech_dK	204316	2846228	108	14.06	9.26e-11	110	8.59	9.20e-11	164	27.30	8.81e-11	167	13.53	4.48e-11
pwtk	217918	11524432	190	28.61	8.36e-11	197	13.83	8.64e-11	283	56.32	9.94e-11	292	21.83	7.55e-11

Numerical experiments



Numerical experiments



Concluding remarks

- We propose an inner product free iterative method called GP-CMRH for solving block two-by-two nonsymmetric linear systems.
- GP-CMRH relies on a new simultaneous Hessenberg process that reduces two rectangular matrices to upper Hessenberg form simultaneously, without employing inner products.
- GP-CMRH requires less computational cost per iteration and may be more suitable for high performance computing and low or mixed precision arithmetic due to its inner product free property.

Concluding remarks

- Our numerical experiments demonstrate that GP-CMRH and GPMR exhibit comparable convergence behavior (with GP-CMRH requiring slightly more iterations), yet GP-CMRH consumes less computational time in most cases.
- GP-CMRH significantly outperforms GMRES and CMRH in terms of number of iterations and runtime efficiency.

The manuscript

- Kui Du and Jia-Jun Fan

GP-CMRH: An inner product free iterative method for block two-by-two nonsymmetric linear systems.

arXiv:2509.11272, 2025.

Thanks!