MOONSHOT PROJECT / NUMERICAL LINEAR ALGEBRA

GPBiCG(m, ℓ, k)

An Extension of GPBiCG(m, ℓ) and Preliminary Adaptive Parameter Strategies

Ronan Dupont Tomohiro Sogabe (曽我部知広) Shao-Liang Zhang (張紹良)
Graduate School of Engineering, Nagoya University, Nagoya, Japan

MOONSHOT RESEARCH & DEVELOPMENT PROGRAM

名古屋大学 Nagoya University

Key message

GPBiCG(m, ℓ, k) adds a third product-type phase. Static evidence is strong; live control can rescue poor triplets. Full no-tuning superiority remains a research direction.

I Krylov reminder | II GPBiCG(m, ℓ, k) | III Numerical study | IV Adaptive strategy | V Conclusions

I. Krylov motivation and contribution

Motivation. Product-type Krylov methods avoid long Arnoldi storage, but convergence remains matrix- and parameter-dependent.

Extension. GPBiCG(m, ℓ, k) inserts a LOPBiCG-like phase into each GPBiCG(m, ℓ) cycle:

$$\mathrm{BiCGSTAB}^m \to \mathrm{GPBiCG}^\ell \to \mathrm{LOPBiCG}^k.$$

Cycle. Execute (m), ℓ, then (k) updates in that order. **Tension.** The added phase can accelerate convergence but makes the triplet $\theta = (m, \ell, k)$ harder to choose.

Research questions.

- test whether good MLK triplets exist on the article matrices;
- verify the conclusion with conventional ILU(0);
- use live residual information to rescue bad triplets;
- stress-test a parameter-free interface.

Scope. This is a sensitivity-reduction path, not a universal replacement for classical no-tuning solvers.

C1. New family
three phases: m, ℓ, k

C2. Fair tests
same matrices and baselines

C3. Adaptive angle
rescue then stress test

II. GPBiCG(m, ℓ, k) and experimental protocol

Linear systems	$b = A(1, \dots, 1)^T$, $x_0 = 0$, $r_0^* = r_0$.
Tolerance	$\|b - Ax_j\|_2 / \|b\|_2 \le 10^{-12}$, except rescue at 10^{-10}.
Grid	343 MLK triplets and 49 ML pairs.
Compared methods	MLK, ML, GPBiCG, BiCGSTAB, BiCGSTAB2.

Matrix	N	nnz	$\kappa_2(A)$	Matrix	N	nnz	$\kappa_2(A)$
utm300	300	3,155	1.50×10^6	memplus	17,758	99,147	1.29×10^5
add20	2,395	13,151	1.20×10^4	epb2	25,228	175,027	2.62×10^3
Goodwin_017	3,317	97,773	1.30×10^5	wang4	26,068	177,196	4.03×10^5
poisson3Da	13,514	352,762	1.12×10^3	crashbasis	160,000	1,750,416	–

Static question
Does k improve the grid optimum?

Conventional check
Repeat with ILU(0).

Adaptive question
Can live data reduce sensitivity?

III.a Without preconditioning: residual histories

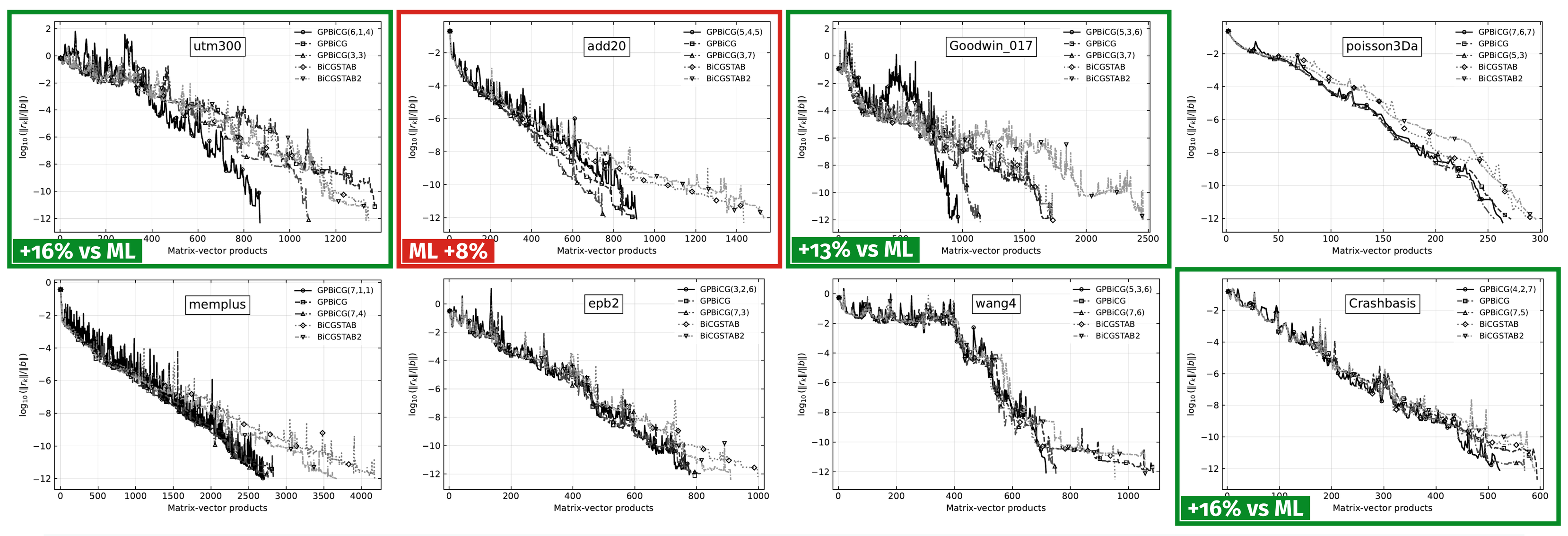

Figure. Eight true-residual histories without preconditioning. Green frames mark MLK gains; the red frame marks the ML exception. MLK is fastest on 6/8 rows and faster than GPBiCG(m, ℓ) on 7/8.

III.b CPU and accuracy without preconditioner

	MLK			ML			GPBiCG			Classical	
Matrix	Time	Resid.	Error	Time	Resid.	Error	Time	Resid.	Error	BiCGSTAB	BiCGSTAB2
utm300	**0.0032**	4.97e-13	8.13e-11	0.0038	8.00e-13	2.69e-10	0.0403†	3.03e-10†	1.51e-8†	0.0048	0.0055
add20	0.0240	8.53e-13	3.92e-10	**0.0222**	9.64e-13	5.11e-10	0.0273	9.73e-13	1.19e-9	0.0325	0.0436
Goodwin_017	**0.0846**	6.38e-13	6.81e-11	0.0974	6.75e-13	1.82e-9	0.1534	9.76e-13	3.17e-10	0.1366	0.2137
poisson3Da	**0.0819**	5.53e-13	8.75e-12	0.0843	9.30e-13	3.43e-11	0.0964	9.03e-13	1.55e-11	0.0890	0.1024
memplus	**0.5491**	9.72e-13	1.20e-9	0.5769	8.40e-13	7.33e-10	0.6410	9.88e-13	9.04e-10	0.6924	0.7847
epb2	**0.2009**	9.14e-13	3.95e-11	0.2124	8.80e-13	1.42e-10	0.2322	7.55e-13	1.16e-10	0.2053	0.2518
wang4	0.2009	8.49e-13	7.80e-11	0.2112	7.49e-13	6.88e-11	0.3285	8.29e-13	1.21e-9	**0.1997**	0.2982
crashbasis	**0.9497**	7.80e-13	1.08e-11	1.1351	8.36e-13	1.71e-11	1.4206	1.93e-13	3.31e-12	0.9954	1.3791

Table. CPU seconds, final true residual and offline error. Bold entries mark the fastest CPU row.

6/8
fastest rows

7/8
faster than ML

13–16%
selected reductions vs ML

III.c Conventional ILU(0): residual histories

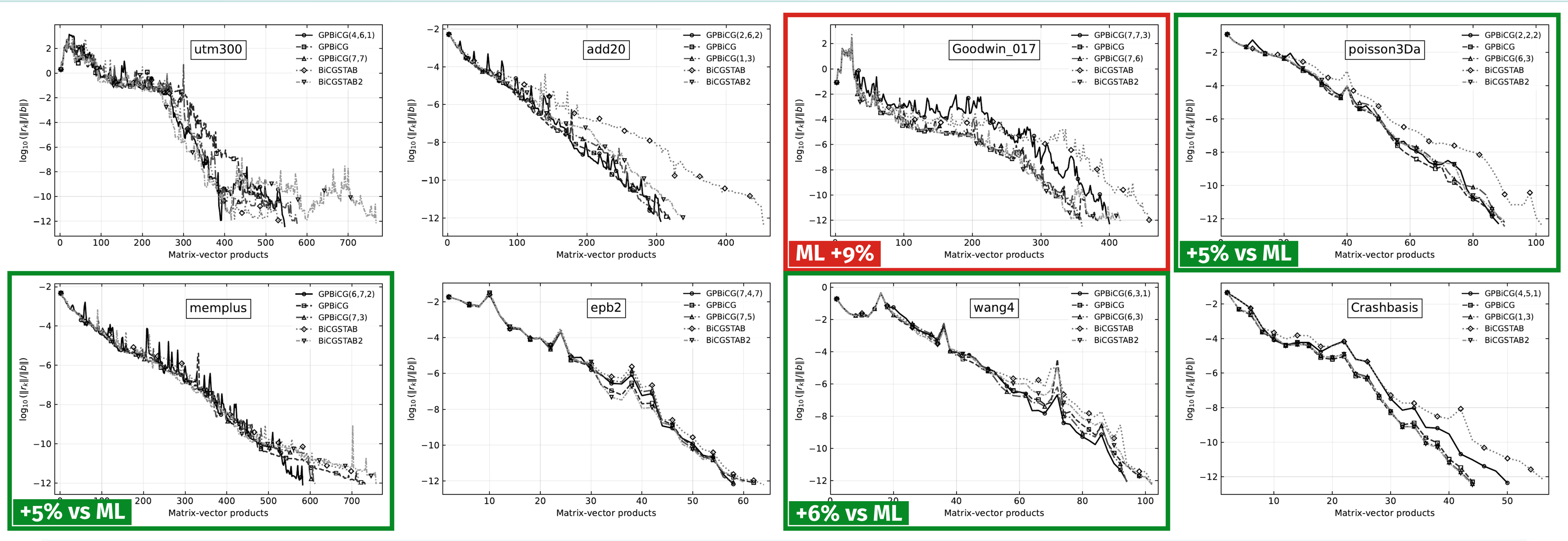

Figure. The same eight matrices with ILU(0). Green frames mark MLK gains; the red frame marks the ML advantage. Gaps narrow, but MLK remains fastest on 4/7 comparable rows.

III.d Conventional ILU(0): CPU table

Matrix	Best MLK	Best ML	CPU MLK	CPU ML	vs GPBiCG	vs ML
utm300	GPBiCG(7,5,2)†	GPBiCG(3,4)†	–	–	–	–
add20	GPBiCG(2,3,1)	GPBiCG(2,4)	0.0136	**0.0133**	1.13	0.982
Goodwin_017	GPBiCG(7,7,3)	GPBiCG(2,5)	0.0647	**0.0587**	–	0.907
poisson3Da	GPBiCG(6,5,5)	GPBiCG(4,3)	**0.0673**	0.0710	1.21	1.05
memplus	GPBiCG(3,5,4)	GPBiCG(7,3)	**0.1884**	0.1973	1.37	1.05
epb2	GPBiCG(3,3,3)	GPBiCG(7,4)	**0.0304**	0.0305	1.20	1.00
wang4	GPBiCG(4,5,6)	GPBiCG(6,3)	**0.0469**	0.0497	1.23	1.06
crashbasis	GPBiCG(7,7,2)	GPBiCG(7,7)	0.1949	**0.1815**	1.21	0.931

Table. Speedups larger than one favor MLK. Dagger rows are diagnostic/non-converged. MLK is fastest on 4/7 comparable rows; gains are narrower after ILU(0).

III.e Static speedup synthesis

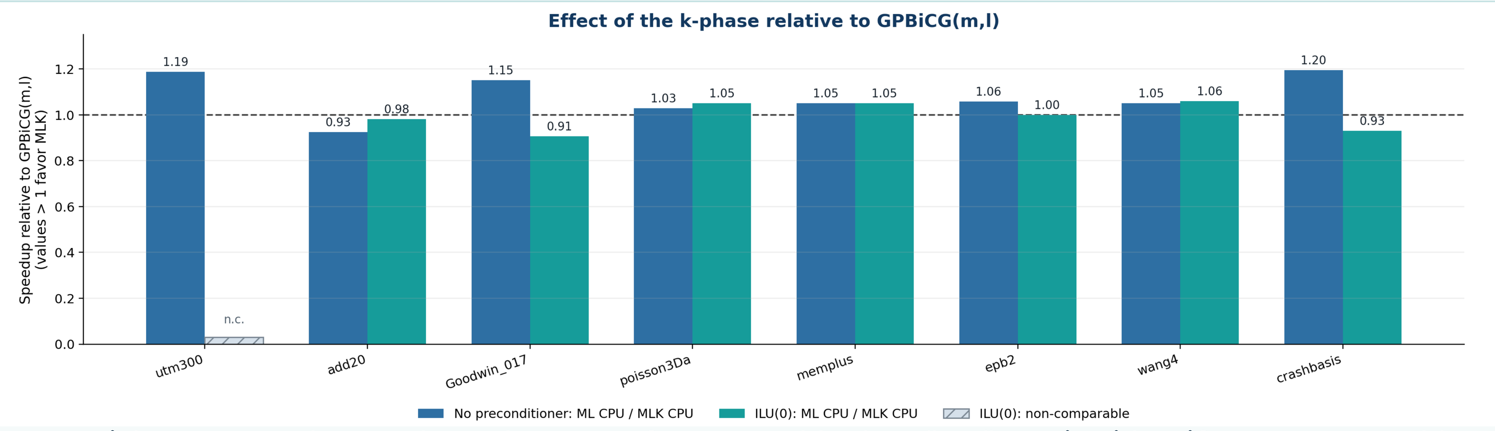

Figure. Article CPU speedups. Values above one favor MLK; blank or "n.c." entries identify non-comparable diagnostic rows where status differs.

IV.a Adaptive controller: one cycle, one decision

1. Observe. Run one complete cycle q with fixed $\theta_q = (m_q, \ell_q, k_q) \in \Theta = \{1, \dots, 7\}^3$. For each phase $p \in \{B, G, L\}$, compute its mean contraction per iteration,

$$\rho_{q,p} = \left(\frac{\|r_{q,p}^{\mathrm{out}}\|_2}{\|r_{q,p}^{\mathrm{in}}\|_2} \right)^{1/n_{q,p}}, \qquad g_q = \frac{\|r_q^{\mathrm{out}}\|_2}{\|r_q^{\mathrm{in}}\|_2}.$$

2. Propose. Let $p_+ = \arg\min_p \rho_{q,p}$ and $p_- = \arg\max_p \rho_{q,p}$. Transfer one iteration: $\theta' = \theta_q + e_{p_+} - e_{p_-}$, with $e_B = (1,0,0)$, $e_G = (0,1,0)$, $e_L = (0,0,1)$.

3. Accept or keep. The next cycle uses

$$\theta_{q+1} = \begin{cases} \theta', & g_q < 1,\ \max_p \rho_{q,p} < 1,\ \theta' \in \Theta, \\ \theta_q, & \text{otherwise.} \end{cases}$$

Accepted example

$\theta_q = (2, 4, 3)$, $\quad g_q = 0.84$,
$(\rho_B, \rho_G, \rho_L) = (0.91, 0.97, 0.88)$.
Best phase: $p_+ = L$. Worst phase: $p_- = G$.
$\theta' = (2, 4-1, 3+1) = (2, 3, 4) \in \Theta$.
All three conditions hold.
Accept $\theta_{q+1} = (2, 3, 4)$

Cycle q, fixed θ_q → **Measure** $g_q, \rho_{q,p}$ → **Propose** local θ' → **Decide** 3 tests → **Cycle** $q+1$ accept / keep

IV.b Experiment I: prescribed-start rescue

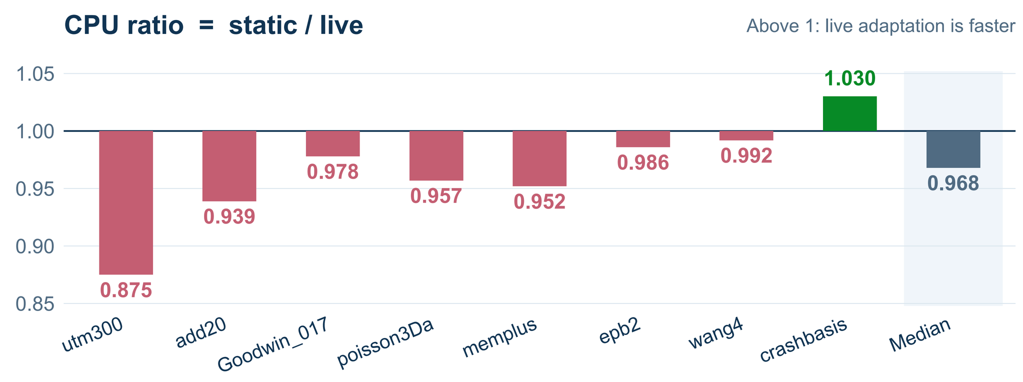

Figure. Static/live CPU ratios for all eight matrices and all 343 prescribed starts at 10^{-10}. Ratios above one favor live adaptation.

Question. Can live information rescue an unfavorable prescribed triplet?

Protocol. Pair each static run with a live run from the same initial triplet.

2744
paired starts

0.968
median ratio

69.1%
crashbasis wins

$> 2\times$
selected rescues

Reading. Selected poor starts are rescued; the median (0.968) exposes monitoring overhead when rescue is unnecessary.

IV.c Experiment II: fully parameter-free starts

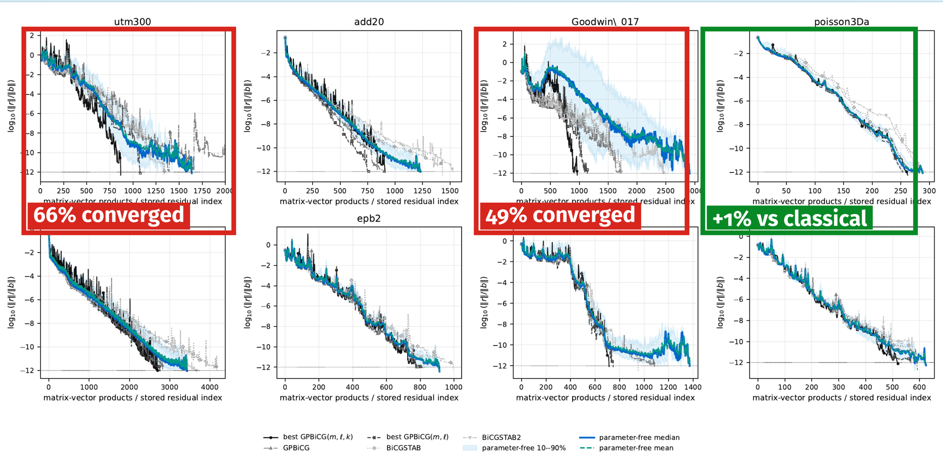

Figure. 100 seeded starts per matrix at 10^{-12}: median and 10–90% envelope. Red: low convergence. Green: sole gain vs best classical baseline.

Mean CPU and convergence over 800 runs

Matrix	Conv.	PF CPU	Baseline	Best/PF
utm300	66%	0.0051	0.0048	0.95
add20	100%	0.0315	0.0273	0.87
Goodwin_017	49%	0.1605	0.1366	0.85
poisson3Da	100%	0.0885	0.0890	**1.01**
memplus	97%	0.6646	0.6410	0.96
epb2	99%	0.2277	0.2053	0.90
wang4	93%	0.3000	0.1997	0.67
crashbasis	100%	1.1870	0.9954	0.84

Best/PF > 1 favors the parameter-free method; the baseline is the best of GPBiCG, BiCGSTAB, and BiCGSTAB2.

Current limitation. Blind-uniform starts remove manual triplet selection, but do not yet yield a winning no-tuning solver. Monitoring and transitions are included.

88%
convergence / 800 runs

5/8
faster than GPBiCG

1/8
beats best no-tuning

V. Conclusions, limitations, and outlook

Limits to state clearly

- The grid $\{1, \dots, 7\}^3$ is systematic but finite.
- CPU rankings depend on compiler flags, cache effects, hardware, and residual logging.
- ILU(0) reduces the room for the added k-phase to create large gains.
- The current controller is a first-generation live policy.

Next adaptive step

- Replace blind random starts by a short deterministic portfolio warm-up.
- Freeze (m, ℓ, k) after stable phase evidence and reactivate only during stagnation.
- Add cheap matrix-family, non-normality, or spectral proxies to guide the first triplet.
- Normalize phase scores by computational cost.

References

[1] S. Fujino, "GPBiCG(m, ℓ): A hybrid of BiCGSTAB and GPBiCG methods with efficiency and robustness", *Appl. Numer. Math.* 41(1), 107–117, 2002.
[2] H. A. van der Vorst, "Bi-CGSTAB: A fast and smoothly converging variant of Bi-CG for nonsymmetric linear systems", *SIAM J. Sci. Statist. Comput.* 13(2), 631–644, 1992.
[3] S.-L. Zhang, "GPBi-CG: Generalized product-type methods based on Bi-CG for solving nonsymmetric linear systems", *SIAM J. Sci. Comput.* 18(2), 537–551, 1997.

Keywords. Product-type Krylov methods; GPBiCG; BiCGSTAB; ILU(0); adaptive parameter selection.

Contact. `r-dupont@na.nuap.nagoya-u.ac.jp`

Quantitative evidence at a glance

6/8
fastest static
7/8 faster than ML

4/7
fastest with ILU(0)
narrower gains

2744
paired starts
selected rescue $> 2\times$

800
random starts
1/8 beats best no-tuning