Preconditioners for the Least Squares Problems Based on the Incomplete Givens Orthogonalization

Jun-Feng Yin

National Institute of Informatics

Nov. 16, 2006

yinjf@nii.ac.jp

Joint work with

◇ **Zhong-Zhi Bai**

Institute of Computational Mathematics

Chinese Academy of China, China

◇ **Ken Hayami**

National Institute of Informatics, Japan

Research subsidized by

First NII International Internship Program

(2006, 1 – 2006, 3)

Outline

- Introduction
- Incomplete Givens Orthogonalization (IGO) Method
- Preconditioned Iterative Method
  - CGLS
  - LSQR
  - BA-GMRES
- Numerical Results
- Conclusion

Problems

Consider the solution of linear least squares problems

$$\min_{x\in\mathbb{R}} \|b - Ax\|, \quad A \in \mathbb{R}^{m\times n}.$$

where A is **large and sparse**, which arise from many scientific applications,

- **Statistical model**
- **Linear and nonlinear programming**
- **Signal and image processing**

Normal Equation

The iterative method solves the normal equation

$$A^T Ax = A^T b.$$

The coefficient matrix of the equation becomes:

- **Symmetric positive definite**
  suitable for conjugate gradient method.
- **More larger condition number**
  require the efficient preconditioners.

Recently Contributions

◇ **Iterative Method**:

- CGLS M.R. Hestenes and E. Stiefel, 1952.
- LSQR C.C. Paige and M.A. Saunders, 1982.
- BA(AB)-GMRES K. Hayami and T. ITO, 2005.

◇ **Preconditioning techniques**:

- CIMGS X. Wang and et al., 1997.
- IGO Z.Z. Bai, I.S. Duff and A.J. Wathen, 2001.
- RIF M. Benzi and M. Tuma, 2003.

General View Å. Björck's Book, 1996.

Review of Givens Rotation

A Givens rotation (or plane rotation) $G(i, j, \theta) \in \mathbb{R}^{n \times n}$ is equal to the identity matrix except that

$$G([i,j],[i,j]) = \begin{pmatrix} c & s \\ -s & c \end{pmatrix},$$

Algebraically, $y_k = \begin{cases} x_k, & \text{for } k \neq i, j, \\ cx_i + sx_j, & \text{for } k = i, \\ -sx_i + cx_j, & \text{for } k = j, \end{cases} \quad 1 \leq k \leq n,$

and so, $y_j = 0$ if

$$s = \frac{x_j}{\sqrt{x_i^2 + x_j^2}}, \qquad c = \frac{x_i}{\sqrt{x_i^2 + x_j^2}}.$$

The Process of the IGO Method

$$\begin{pmatrix} * & * & * & * \\ \color{red}{\longrightarrow} & * & * & * \\ \color{blue}{\longrightarrow} & \color{blue}{\longrightarrow} & * & * \\ \color{magenta}{\longrightarrow} & \color{magenta}{\longrightarrow} & \color{magenta}{\longrightarrow} & * \end{pmatrix} \text{ or } \begin{pmatrix} * & * & * & * \\ \color{red}{\downarrow} & * & * & * \\ \color{red}{\downarrow} & \color{blue}{\downarrow} & * & * \\ \color{red}{\downarrow} & \color{blue}{\downarrow} & \color{magenta}{\downarrow} & * \end{pmatrix} \text{ or } \begin{pmatrix} * & * & * & * \\ \color{red}{\uparrow} & * & * & * \\ \color{red}{\uparrow} & \color{blue}{\uparrow} & * & * \\ \color{red}{\uparrow} & \color{blue}{\uparrow} & \color{magenta}{\uparrow} & * \end{pmatrix}$$

Choice of the order depends on

- **Storage format**
- **Computational implementation**

Row IGO Method

Method : The General row IGO Method

1. For $i = 2, \ldots, m$ Do:
2. For $j = 1, \ldots, \min\{i-1, n\}$ and $a_{ij} \neq 0$ Do:
3. If $|a_{ij}| < \tau$, Then
4. Set $a_{ij} = 0$
5. Cycle to Next j
6. EndIf
7. Compute $\rho := \sqrt{a_{jj}^2 + a_{ij}^2}$
8. Compute $c := a_{jj}/\rho$
9. Compute $s := a_{ij}/\rho$
10. Set $a_{jj} = \rho$, $a_{ij} = 0$
11. Store c, s

Row IGO Method(continue)

For $k = \min\{j+1, n\}, \ldots, n$ Do:
    Compute $temp_j := ca_{jk} + sa_{ik}$
    Compute $temp_i := -sa_{jk} + ca_{ik}$
    If $|a_{jk}| > \tau$, Set $a_{jk} := temp_j$
    Else, Set $a_{jk} := 0$
    If $|a_{ik}| > \tau$, Set $a_{ik} := temp_i$
    Else, Set $a_{ik} := 0$
EndDo
EndDo
EndDo
For $i = 1, \ldots, n$
  Keep only the p largest entries in the i-th row
  Set $r_{ij} := a_{ij}$
EndDo

Theorem of the IGO-method

Theorem (Bai/Duff/Wathen BIT(2001))

Let $A \in \mathbb{R}^{n\times n}$ be nonsingular, and $Q, R \in \mathbb{R}^{n\times n}$ be the incomplete orthogonal and upper triangular matrices, respectively, produced by the IGO-method. Then

(i) R is sparse and nonsingular, and its diagonal entries are positive except possibly for the last one;

(ii) $Q = G^T$ is orthogonal, provided $P_Q = P_n$, where

$$G = G_{n-1}G_{n-2}\cdots G_1 \equiv \prod_{j=n-1}^{1} G_j.$$

Theorem of the Row IGO Method

Theorem Let $A \in \mathbb{R}^{m\times n}$ be full column rank, and $Q, R \in \mathbb{R}^{n\times n}$ be the incomplete orthogonal and upper triangular matrices, respectively, produced by the IGO-method. Then

(i) $R \in \mathbb{R}^{n\times n}$ is sparse and nonsingular, and its diagonal entries are positive except possibly for the last one;

(ii) $Q = G^T \in \mathbb{R}^{m\times m}$ is orthogonal, where $G = G_{n-1}G_{n-2}\cdots G_1 \equiv \prod_{j=n-1}^{1} G_j$.

More Consideration for the IGO Method

For the squares matrices,

$$G_n \dots G_1 A \approx R$$

For the rectangular matrices

$$G_n \dots G_1 A \approx \begin{pmatrix} R \\ 0 \end{pmatrix}, \quad A \approx Q \begin{pmatrix} R \\ 0 \end{pmatrix} \triangleq M.$$

More details on

♢ Sparse structure and storage scheme

♢ The magnitude of the fill-ins

♢ How to solve the equation $Mz = v$

Sparse Structure

By define

$$P_{A,L} = \{(i,j) \mid a_{ij} \neq 0, \quad i > j, \quad 1 \leq i,j \leq n\},$$
$$P_{A,R} = \{(i,j) \mid a_{ij} \neq 0, \quad i \leq j, \quad 1 \leq i,j \leq n\},$$

we have the dropping rule

$$P_R = P_{A,R} \quad \text{and} \quad P_G = P_{A,L}$$

By introducing a parameter $l, 0 \leq l \leq n$, we have

$$P_{A,L,l} = P_{A,L} \cup \{(i,j) \mid i \leq j \leq i + l, \quad 1 \leq i,j \leq n\},$$
$$P_{A,R,l} = P_{A,R} \cup \{(i,j) \mid i - \lceil \frac{m}{n} \rceil l \leq j \leq i, \quad 1 \leq i,j \leq n\},$$

General Remarks

- The IGO method will never breakdown and produce an orthogonal matrix Q and an upper triangular matrix R,
- The upper triangular matrix R is nonsingular in case of the original matrix is full column rank ,
- All the factorization processes are numerically stable,
- Need more operations than ILU method.

Preconditioned Iterative Method

Preconditioned CGLS Method

$$M = R^{\mathrm{T}} R,$$

$$R^{-\mathrm{T}} A^{\mathrm{T}} A R^{-1} u = R^{-\mathrm{T}} A^{\mathrm{T}} b, \quad u = Rx.$$

- Q should not be stored.
- The preconditioner M is symmetric positive definite.
- The additional computation is only solving the two triangle equations:

$$R^{\mathrm{T}} w = v, Rz = w.$$

Preconditioned LSQR Method

The preconditioned LSQR method solves the problem

$$\min \|b - AR^{-1}x\|,$$

instead of

$$\min \|b - Ax\|,$$

which imply

$$\theta_1 v_1 = A^{\mathrm{T}} b, \quad \rho_1 p_1 = AR^{-1} v_1.$$

$$\left.\begin{aligned} \theta_{i+1} v_{i+1} &= A^{\mathrm{T}} R^{-\mathrm{T}} p_i - \rho_i v_i \\ \rho_{i+1} p_{i+1} &= AR^{-1} v_i - \theta_{i+1} p_i \end{aligned}\right\}, \quad i = 1, 2, \ldots$$

Preconditioned BA-GMRES Method

The method solve

$$BAx = Bb$$

instead of the normal equation, where B should satisfied:

$$\mathcal{R}(A) = \mathcal{R}(B^{\mathrm{T}}), \quad \mathcal{R}(A^{\mathrm{T}}) = \mathcal{R}(B)$$

Theorem (Hayami/ITO (2005)) If $\mathcal{R}(A) = \mathcal{R}(B^{\mathrm{T}})$ and $\mathcal{R}(A^{\mathrm{T}}) = \mathcal{R}(B)$, the BA-GMRES method determines a least squares solution of $\min_{\boldsymbol{x} \in \mathbf{R}^n} \|\boldsymbol{b} - A\boldsymbol{x}\|_2$ for all $\boldsymbol{b} \in \mathbf{R}^m$ and for all $\boldsymbol{x}_0 \in \mathbf{R}^n$ without breakdown.

Choice of the mapping matrix B

When A is full column rank, R is nonsingular, there are at least three choices:

$$B = A^{\mathrm{T}} \tag{1}$$

$$B = [R^{-1}\ 0]G \tag{2}$$

$$B = (R^{\mathrm{T}}R)^{-1}A^{\mathrm{T}} \tag{3}$$

$$\mathcal{R}(A) = \mathcal{R}(B^{\mathrm{T}}), \quad \mathcal{R}(A^{\mathrm{T}}) = \mathcal{R}(B)$$

Theoretical Analysis

	$\mathcal{R}(A) = \mathcal{R}(B^{\mathrm{T}})$	$\mathcal{R}(A^{\mathrm{T}}) = \mathcal{R}(B)$
$B = A^{\mathrm{T}}$	$\surd$	$\surd$
$B = [R^{-1}\ 0]G$	?	$\surd$
$B = (R^{\mathrm{T}}R)^{-1}A^{\mathrm{T}}$	$\surd$	$\surd$

Numerical Results

Comparison between B(1) and B(3)

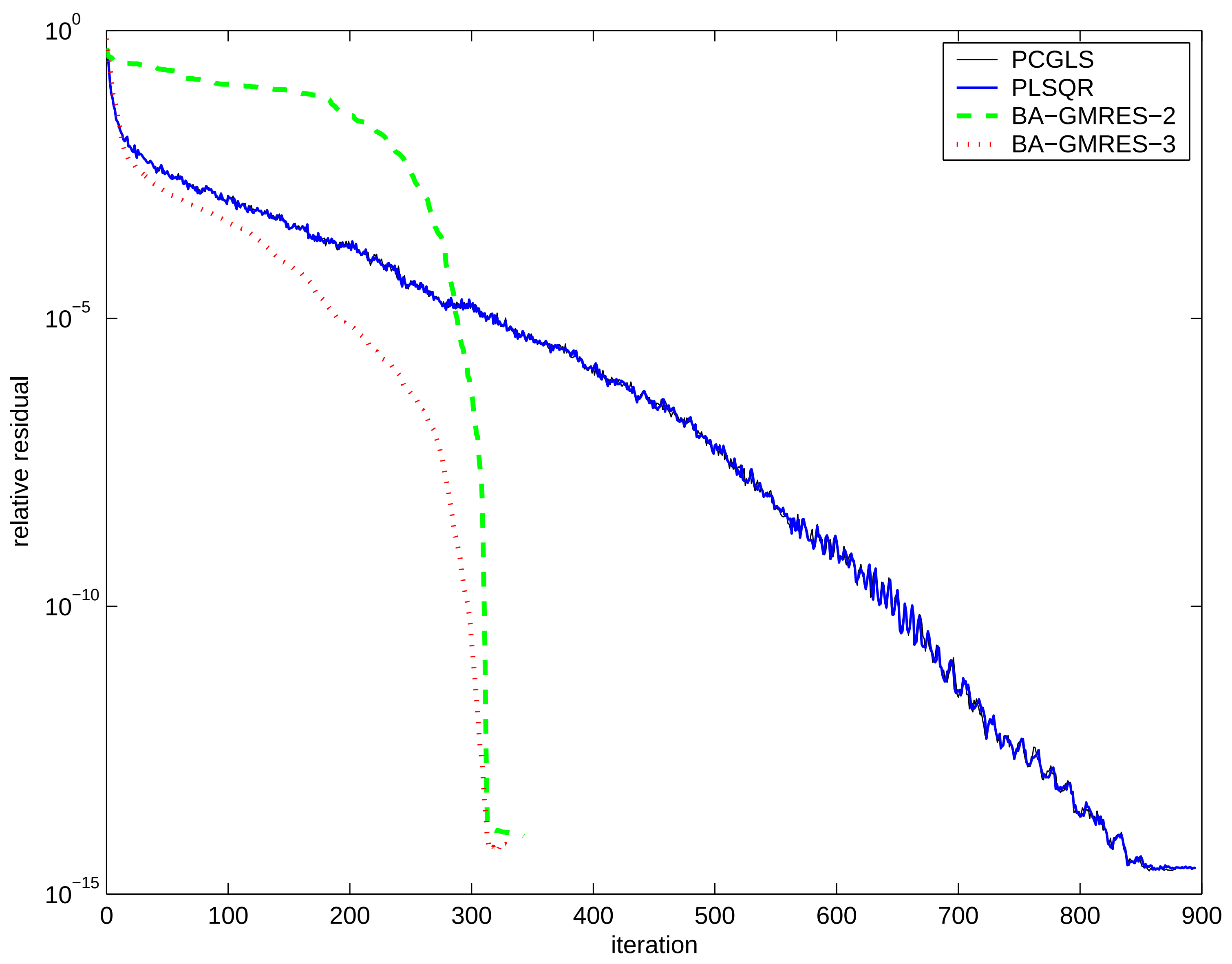
	CGLS	LSQR	B(1)	PCGLS	PLSQR	B(3)
RANDS2	408	411	162	35	35	36
	1.47	1.69	1.11	0.25	0.26	0.28
RANDS4	1891	1926	239	86	87	73
	6.81	7.84	2.00	0.53	0.56	0.56
RANDS6	9871	10148	258	325	337	130
	35.61	42.10	2.23	1.80	2.03	1.09
RANDS8	46520	44634	266	1035	1051	153
	168.54	185.11	2.38	5.49	6.11	1.34

$m = 1000, n = 320, den(A) = 4.9\%, \text{CPU}_P = 0.08$ sec

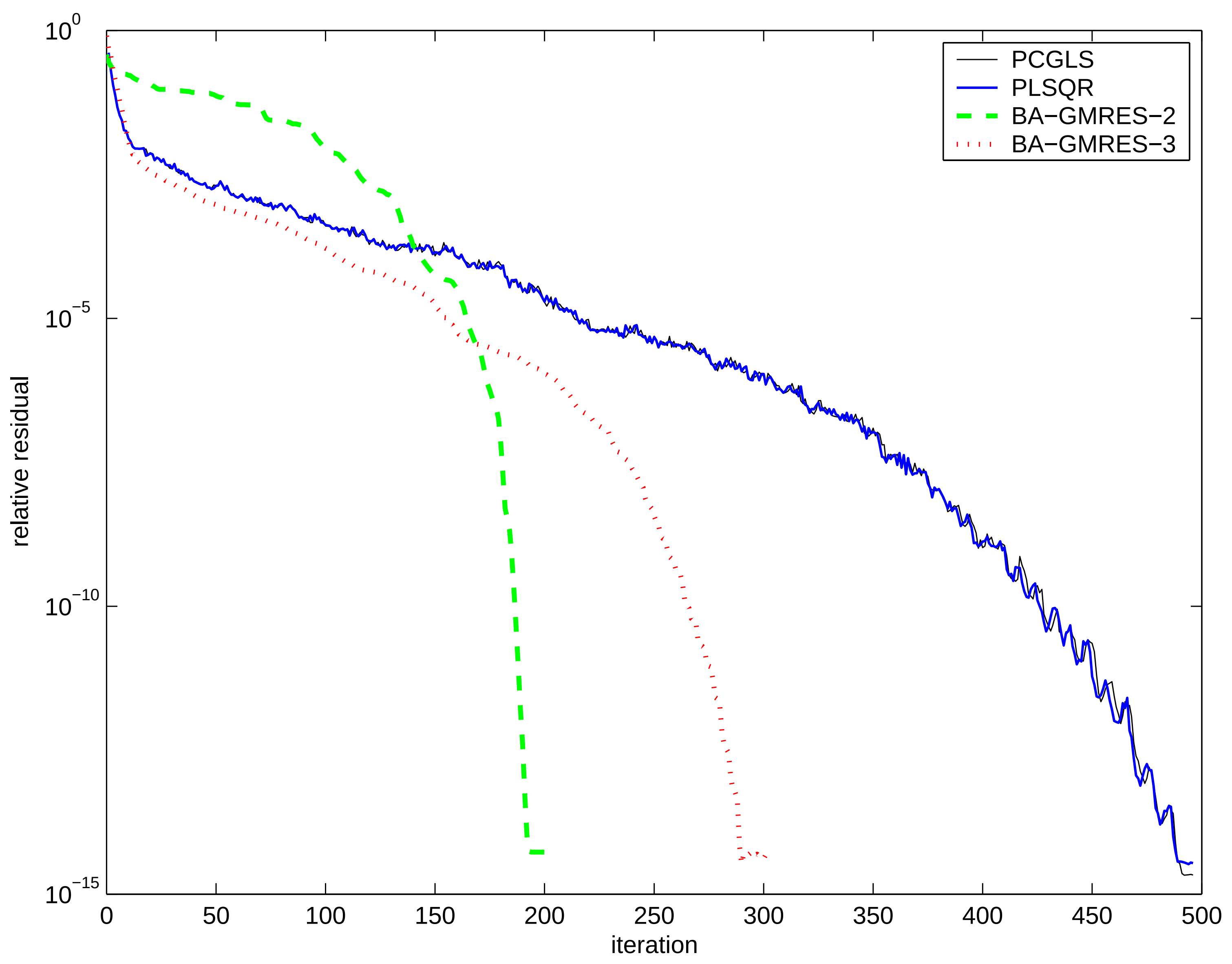
Numerical Comparison of B(2) and B(3)

l		PCGLS	PLSQR	B(2)	B(3)
100	5.04	438	431	967	256
		17.76	17.59	37.94	13.72
200	7.12	387	379	961	244
		15.58	15.95	41.91	13.48
400	12.62	333	324	837	220
		14.39	14.29	42.28	12.23
600	17.75	256	248	512	188
		11.20	11.13	28.14	10.21
800	22.72	188	179	160	142
		8.38	8.14	9.40	7.45
1000	24.91	1	1	1	1
		0.05	0.05	0.07	0.05

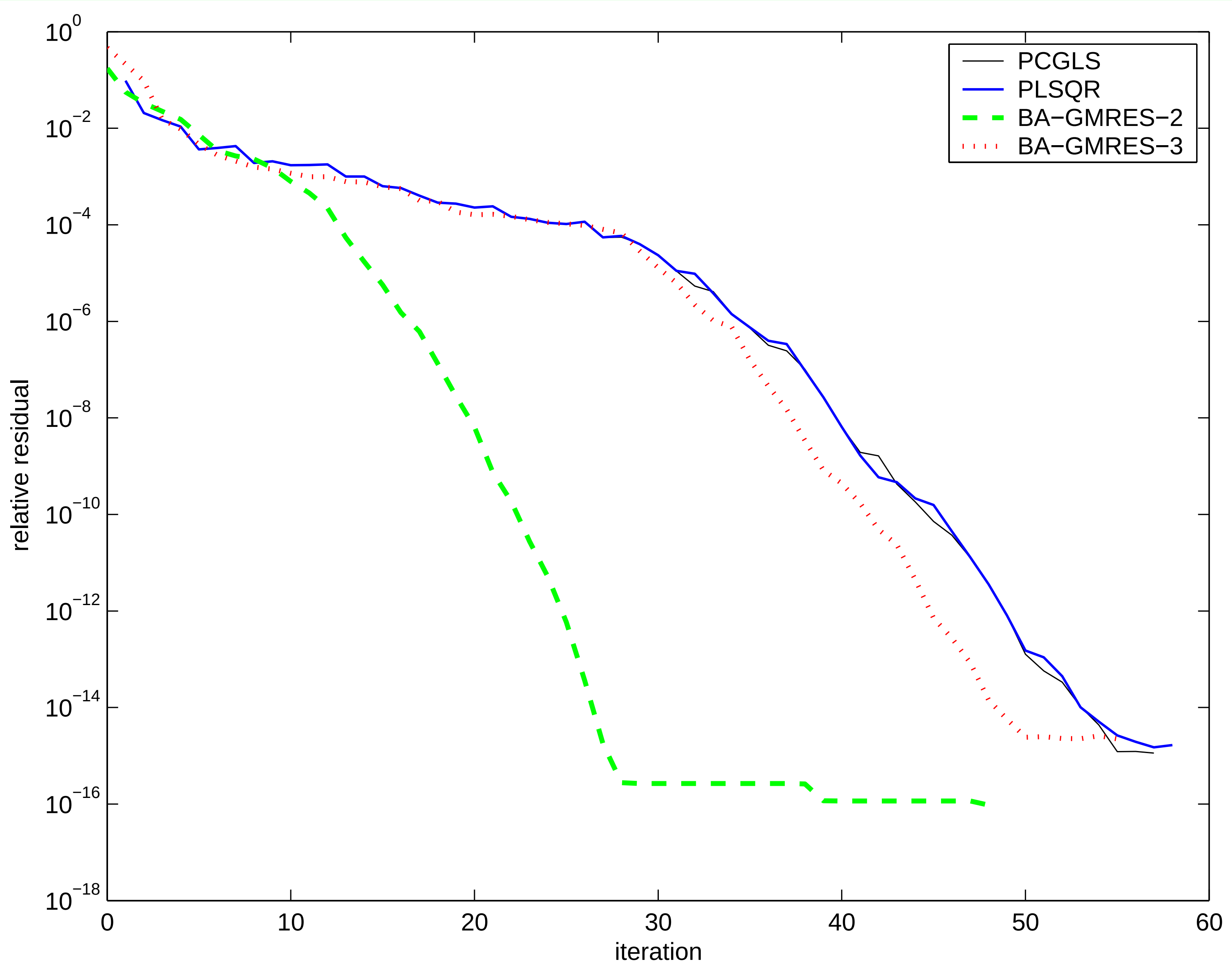
$m = 10000, n = 1000, \kappa(A) = 80, den(A) = 1.5\%, \frac{A^T r}{A^T b} < 10^{-6}$



$$m = 10000, n = 1000, l = 700, \kappa(A) = 10^4$$



$m = 10000, n = 1000, l = 800, \kappa(A) = 10^4$



$m = 10000, n = 1000, l = 900, \kappa(A) = 10^4$

Numerical Results when $l = 200$

	PCGLS		PLSQR		BAGMRES	
	IT	CPU	IT	CPU	IT	CPU
RAND1	387	15.58	379	15.95	244	13.48
RAND2	914	36.81	917	37.74	420	26.94
RAND3	1932	78.16	1942	80.02	598	44.53
RAND4	5708	230.38	5897	243.06	721	58.81
RAND5	15211	613.45	16561	682.27	770	65.17
RAND6	44023	1777.38	50540	2082.52	786	67.06
RAND7	90389	3671.70	–	–	802	69.61
RAND8	–	–	–	–	776	66.06

MAXIT=100000, $m = 10000$, $n = 1000$, $\mathrm{CPU}_P = 7.12$ sec

Conclusion

- Introduce a class of preconditioner base on incomplete Givens orthogonal factorization.
- Analysis their convergence performance of the preconditioned iterative methods for the least square problems.
- Examine their properties numerically, especially for the BA-GMRES method.

Thank You!