Comparison of Conjugate Gradient Method for Nonsymmetric matrices

Hidehiko Hasegawa (Univ. of Tsukuba)
Tomohiro Sogabe(Nagoya Univ.)
Takeshi Ogita(Waseda Univ., JST)
Tamito Kajiyama(Tokyo Univ., JST)

Outline

- CG method for Nonsymmetric matrices:
  1) CGNR: $A^T A x = A^T b$
  2) Expanded System:

$$\begin{pmatrix} & A^T \\ A & \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix} = \begin{pmatrix} c \\ b \end{pmatrix}$$

- Necessity of Accurate Arithmetic
- **SILC**: Simple Interface for Library Collections for an analysis tool

CGNR: $A^T A x = A^T b$

- Converge in one iteration for Unitary matrix
- Optimal where the spectrum has certain symmetries
- However, $Cond(A^T A) = Cond(A)$

Hestenes & Stiefel. [1952]
Freund, Golub, Nachtigal. Acta Numerica, pp. 1-44 [1992]

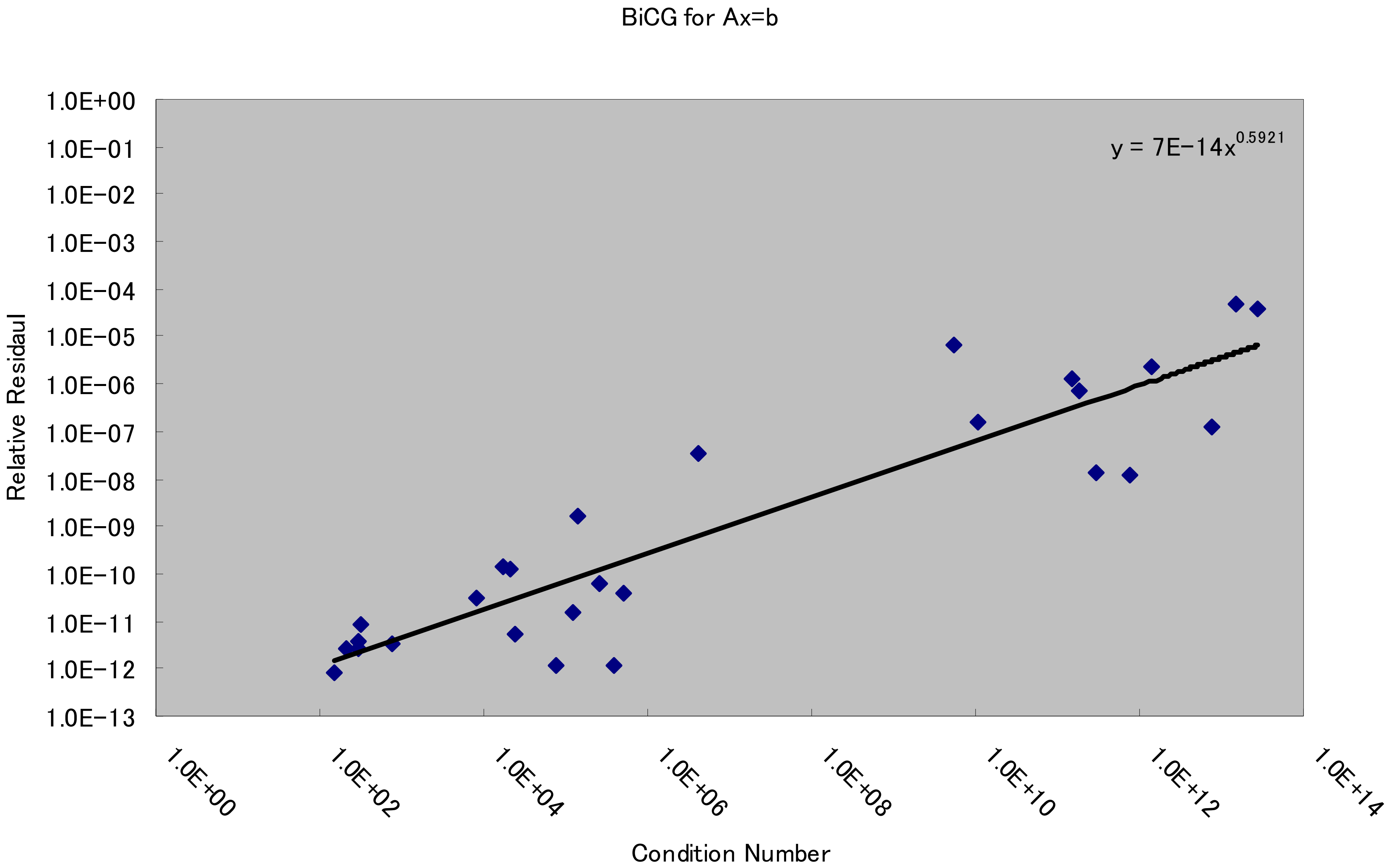
		BiCG (Ax=b)					CG (A'Ax=A'b)				
Matrix	N	nnz	Cond(A)	#	Sec.	Rel. Res.	nnz	Cond(A'A)	#	Sec.	Rel. Res
add20.mtx	2,395	17,319	1.76E+04	542	0.137761	9.99E−13	213,657	2.11E+08	8,478	7.982620	1.44E−13
add32.mtx	4,960	23,884	2.14E+02	124	0.047238	5.46E−13	102,422	2.98E+04	855	0.502985	1.31E−14
arc130.mtx	130	1,282	1.08E+10	18	0.001061	8.28E−13	15,656	6.39E+21	27	0.001384	8.15E−08
bp__1000.mtx	822	4,661	6.80E+07	－	－	－	107,374	2.86E+15	－	－	－
bp__1200.mtx	822	4,726	3.46E+08	－	－	－	109,180	1.08E+17	－	－	－
bp__1400.mtx	822	4,790	5.12E+07	－	－	－	109,810	5.15E+14	－	－	－
bp__1600.mtx	822	4,841	8.65E+06	－	－	－	106,338	4.87E+13	－	－	－
bp__200.mtx	822	3,802	8.92E+06	－	－	－	88,234	9.02E+13	－	－	－
bp__400.mtx	822	4,028	1.25E+07	－	－	－	95,908	5.77E+13	－	－	－
bp__600.mtx	822	4,172	5.08E+06	－	－	－	100,466	8.71E+12	－	－	－
bp__800.mtx	822	4,534	1.23E+07	－	－	－	103,950	1.98E+13	－	－	－
bp___0.mtx	822	3,276	2.03E+07	－	－	－	77,322	1.07E+14	－	－	－
fs_183_1.mtx	183	1,069	1.51E+13	1,300	0.064237	3.47E−13	11,803	8.24E+26	30	0.003158	8.10E−04
fs_183_3.mtx	183	1,069	2.69E+13	4,447	0.230575	3.63E−13	11,803	2.09E+27	32	0.001569	6.00E−04
fs_183_4.mtx	183	1,069	1.86E+11	476	0.052486	6.94E−13	11,803	9.53E+22	30	0.003130	7.59E−04
fs_183_6.mtx	183	1,069	1.50E+11	565	0.028580	1.10E−13	11,803	5.63E+22	32	0.003429	5.81E−04
fs_541_1.mtx	541	4,285	7.65E+04	15	0.001398	4.65E−13	14,005	4.13E+07	39	0.002300	2.19E−11
fs_541_2.mtx	541	4,285	7.70E+11	1,008	0.071680	7.34E−13	14,005	3.72E+21	－	－	－
fs_541_3.mtx	541	4,285	7.74E+12	2,616	0.185438	7.42E−13	14,005	1.96E+23	－	－	－
fs_541_4.mtx	541	4,285	2.92E+11	1,316	0.090755	9.16E−13	14,005	5.45E+20	－	－	－
fs_680_1.mtx	680	2,646	2.08E+04	462	0.032294	8.73E−13	6,178	5.34E+08	3,482	0.199662	2.55E+01
fs_680_2.mtx	680	2,646	1.36E+05	1,113	0.076019	7.34E−13	6,178	3.62E+10	－	－	－
fs_680_3.mtx	680	2,646	4.20E+06	4,317	0.298821	9.87E−13	6,178	2.19E+13	－	－	－
fs_760_1.mtx	760	5,976	8.36E+03	141	0.012546	9.18E−13	27,152	3.41E+07	714	0.070935	1.75E−04
fs_760_2.mtx	760	5,976	1.12E+16	－	－	－	27,152	2.61E+32	4,200	0.334110	2.44E−04
fs_760_3.mtx	760	5,976	9.93E+19	－	－	－	27,152	5.00E+40	5,265	0.448831	9.07E−05
gemat11.mtx	4,929	33,185	3.74E+08	－	－	－	83,627	1.13E+16	－	－	－
gemat12.mtx	4,929	33,111	3.74E+08	－	－	－	85,271	3.09E+16	－	－	－
gre_1107.mtx	1,107	5,664	9.75E+07	－	－	－	24,617	3.66E+15	1,508	0.124929	8.96E−13
gre_216a.mtx	216	876	3.05E+02	223	0.011079	7.28E−13	3,016	2.53E+04	123	0.004587	5.16E−13
gre_216b.mtx	216	876	8.10E+14	－	－	－	3,016	1.17E+19	12	0.000423	3.94E−13
gre__115.mtx	115	421	1.52E+02	85	0.004054	5.60E−13	1,267	6.63E+03	107	0.003585	4.96E−13
gre__185.mtx	185	1,005	5.09E+05	424	0.021626	8.46E−13	4,013	1.09E+11	291	0.011004	8.93E−13
gre__343.mtx	343	1,435	3.09E+02	509	0.027142	9.53E−13	5,095	4.37E+04	172	0.007116	5.78E−13
gre__512.mtx	512	2,192	3.77E+02	－	－	－	7,960	6.71E+04	231	0.010542	7.12E−13
impcol_a.mtx	207	572	4.35E+07	－	－	－	1,219	5.33E+16	－	－	－
impcol_b.mtx	59	312	2.67E+05	155	0.006890	5.22E−13	717	6.10E+10	140	0.004505	6.22E−12
impcol_c.mtx	137	411	2.38E+04	668	0.031102	6.47E−13	1,077	3.83E+08	141	0.004843	7.85E−11
impcol_d.mtx	425	1,339	1.87E+03	－	－	－	4,011	9.49E+06	811	0.032203	4.94E−12
impcol_e.mtx	225	1,308	9.27E+06	－	－	－	3,983	1.30E+14	1,919	0.075907	3.94E−09
jpwh_991.mtx	991	6,027	7.27E+02	－	－	－	25,141	5.72E+04	455	0.050271	9.55E−13
lns_3937.mtx	3,937	25,407	1.04E+17	－	－	－	101,673	8.22E+31	－	－	－

		BiCG (Ax=b)					CG (A'Ax=A'b)				
Matrix	N	nnz	Cond(A)	#	Sec.	Rel. Res.	nnz	Cond(A'A)	#	Sec.	Rel. Res
lns_131.mtx	131	536	1.49E+15	-	-	-	2,167	2.18E+30	33	0.001076	8.70E-06
lns_511.mtx	511	2,796	6.37E+15	-	-	-	11,277	3.68E+29	647	0.033220	2.01E-03
lnsp3937.mtx	3,937	25,407	1.04E+17	-	-	-	101,673	2.18E+37	-	-	-
lnsp_131.mtx	131	536	1.49E+15	-	-	-	2,167	2.18E+31	31	0.002603	1.71E-03
lnsp_511.mtx	511	2,796	6.37E+15	-	-	-	11,277	2.73E+34	648	0.035632	1.78E-03
mahindas.mtx	1,258	7,682	1.03E+13	-	-	-	40,338	5.81E+24	47	0.006343	1.46E-05
mbeacxc.mtx	496	49,920	1.03E+13	-	-	-	235,221	5.81E+24	-	-	-
mbeaflw.mtx	496	49,920	1.03E+13	-	-	-	235,221	5.81E+24	-	-	-
mbeause.mtx	496	41,063	1.03E+13	-	-	-	240,098	5.81E+24	-	-	-
mcca.mtx	180	2,659	3.59E+17	-	-	-	9,820	5.32E+34	-	-	-
mcfe.mtx	765	24,382	1.67E+14	-	-	-	145,741	3.73E+27	-	-	-
memplus.mtx	17,758	126,150	2.67E+05	1,675	4.094148	8.19E-13	5,121,784	3.60E+10	-	-	-
nnc1374.mtx	1,374	8,606	4.11E+15	-	-	-	36,080	2.52E+38	-	-	-
nnc261.mtx	261	1,500	1.17E+15	-	-	-	5,853	3.15E+25	1,727	0.069710	6.74E-10
nnc666.mtx	666	4,044	1.78E+11	-	-	-	16,518	9.60E+19	8,790	0.478705	7.76E-10
orani678.mtx	2,529	90,158	1.00E+07	-	-	-	1,861,423	1.25E+09	2,915	21.241944	8.14E-12
sherman2.mtx	1,080	23,094	1.42E+12	-	-	-	102,050	6.74E+24	-	-	-
sherman5.mtx	3,312	20,793	3.90E+05	2,022	0.438289	4.83E-13	89,766	1.25E+11	-	-	-
shl_200.mtx	663	1,726	1.95E+07	-	-	-	194,691	2.51E+18	405	0.338496	9.39E-09
shl_400.mtx	663	1,712	1.93E+07	-	-	-	182,599	2.29E+18	487	0.383940	8.02E-09
shl__0.mtx	663	1,687	2.21E+07	-	-	-	179,159	2.50E+18	312	0.247967	8.45E-09
str_200.mtx	363	3,068	1.26E+05	7,774	0.501839	1.02E-12	8,331	1.12E+09	677	0.032634	4.12E-12
str_400.mtx	363	3,157	3.70E+04	-	-	-	8,363	1.85E+08	759	0.039556	6.79E-12
str_600.mtx	363	3,279	1.80E+06	-	-	-	8,869	1.81E+11	1,572	0.075874	6.70E-12
str__0.mtx	363	2,454	7.62E+02	2,895	0.176748	9.05E-13	7,123	3.22E+05	238	0.011366	7.10E-12
watt__1.mtx	1,856	11,360	5.38E+09	327	0.036344	3.09E-13	37,964	2.42E+19	1	0.000422	3.85E-14
watt__2.mtx	1,856	11,550	1.37E+12	466	0.050161	4.46E-13	53,032	1.15E+21	1	0.000474	5.35E-14
west0067.mtx	67	294	3.00E+02	175	0.007963	9.51E-13	889	7.41E+04	120	0.003826	1.23E-12
west0132.mtx	132	414	5.94E+11	-	-	-	1,252	3.78E+23	835	0.028279	2.37E-07
west0156.mtx	156	371	1.64E+31	-	-	-	836	2.18E+32	7	0.000309	1.65E-05
west0167.mtx	167	507	1.72E+11	-	-	-	1,545	6.14E+21	1,113	0.038302	2.44E-07
west0381.mtx	381	2,157	2.02E+06	-	-	-	13,509	7.05E+12	-	-	-
west0479.mtx	479	1,910	1.42E+12	-	-	-	7,143	3.21E+23	8,523	0.386315	2.61E-07
west0497.mtx	497	1,727	1.38E+12	-	-	-	9,711	2.74E+23	4,083	0.210886	5.68E-07
west0655.mtx	655	2,854	1.58E+12	-	-	-	11,327	3.38E+23	-	-	-
west0989.mtx	989	3,537	5.68E+12	-	-	-	12,235	3.54E+24	-	-	-
west1505.mtx	1,505	5,445	9.01E+12	-	-	-	18,863	7.01E+24	-	-	-
west2021.mtx	2,021	7,353	7.50E+12	-	-	-	25,527	2.49E+25	-	-	-

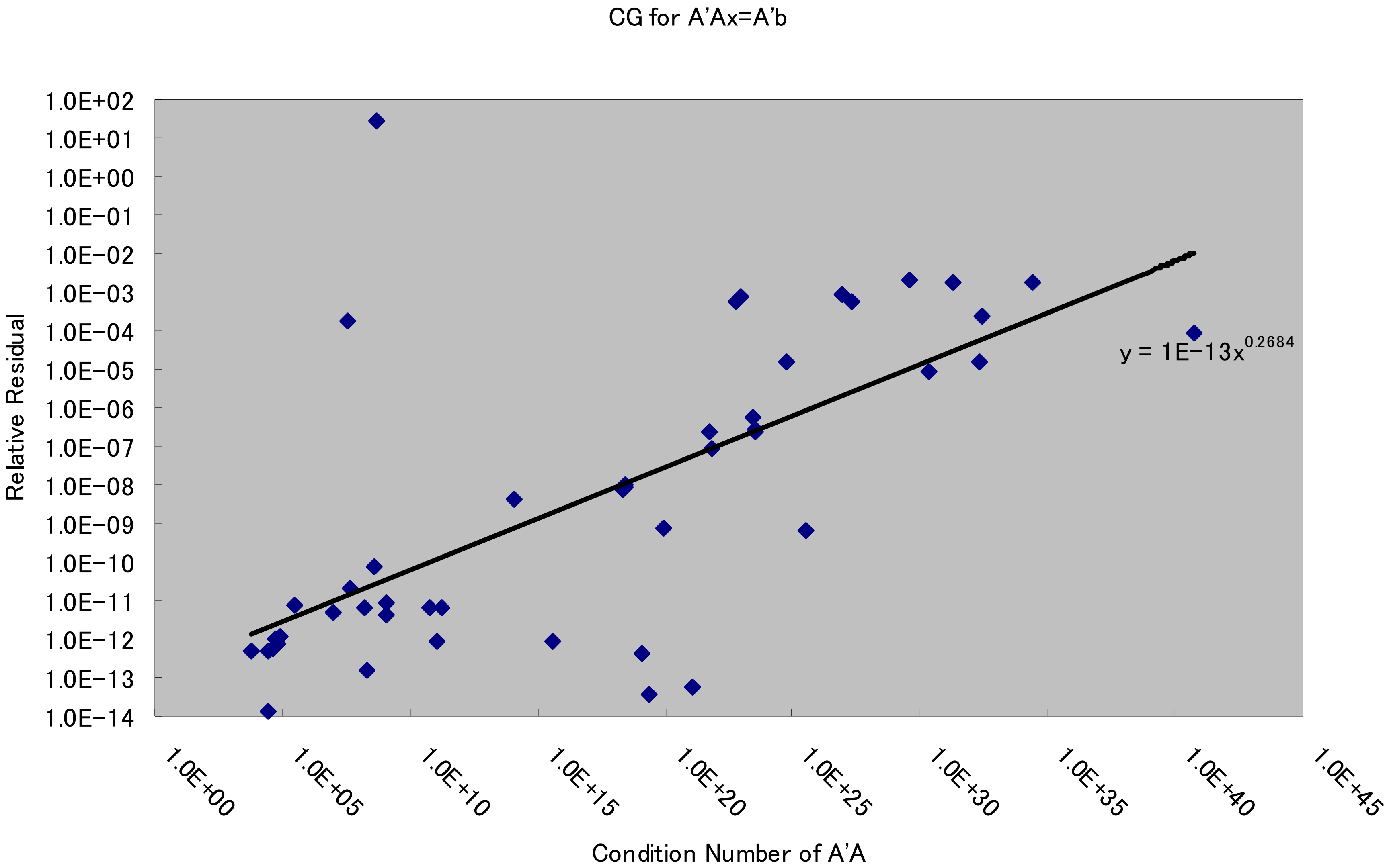
80 Real Nonsymmetric matrices from Matrix Market

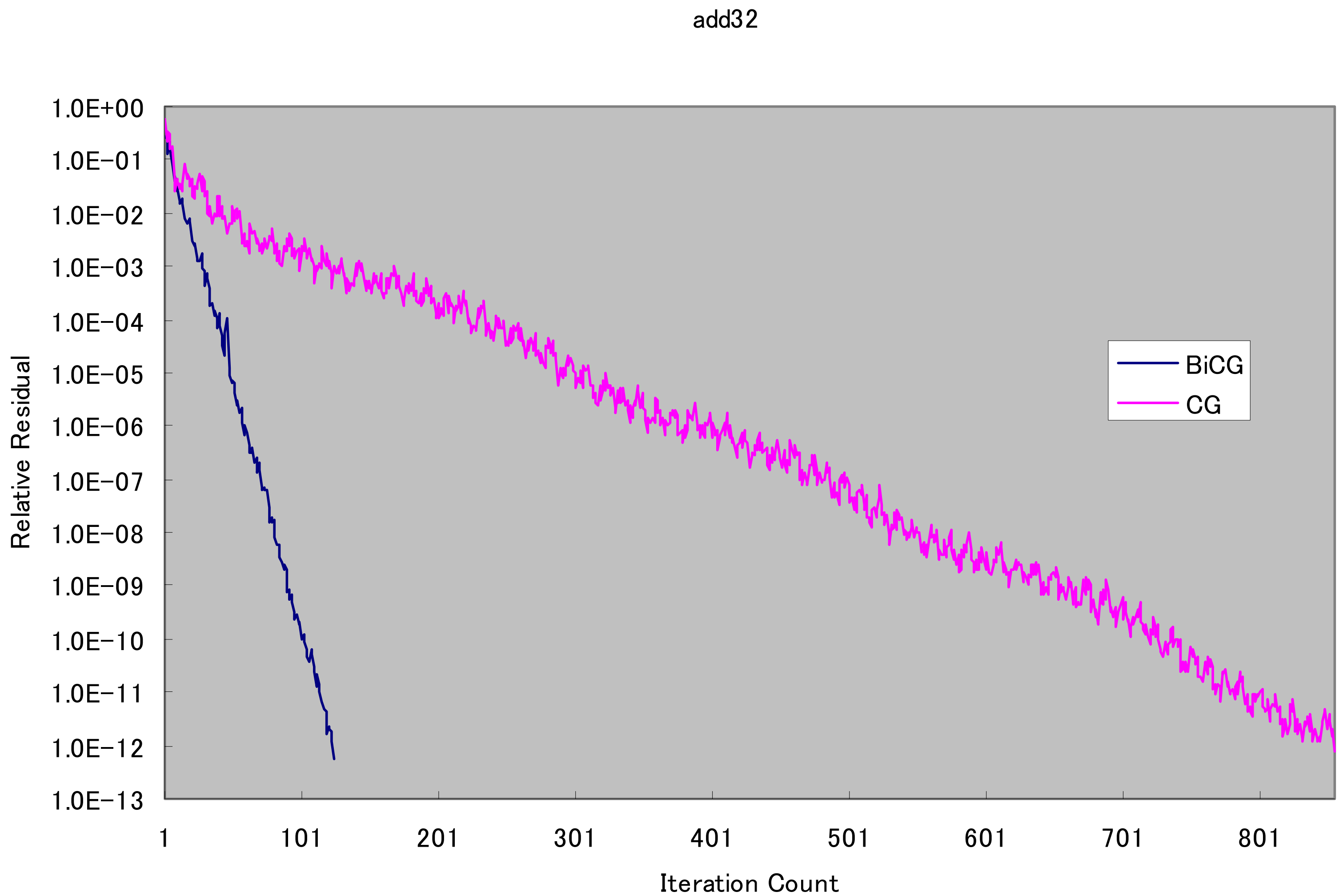
$x = (1, \ldots, 1)^T$, $x_0 = o$

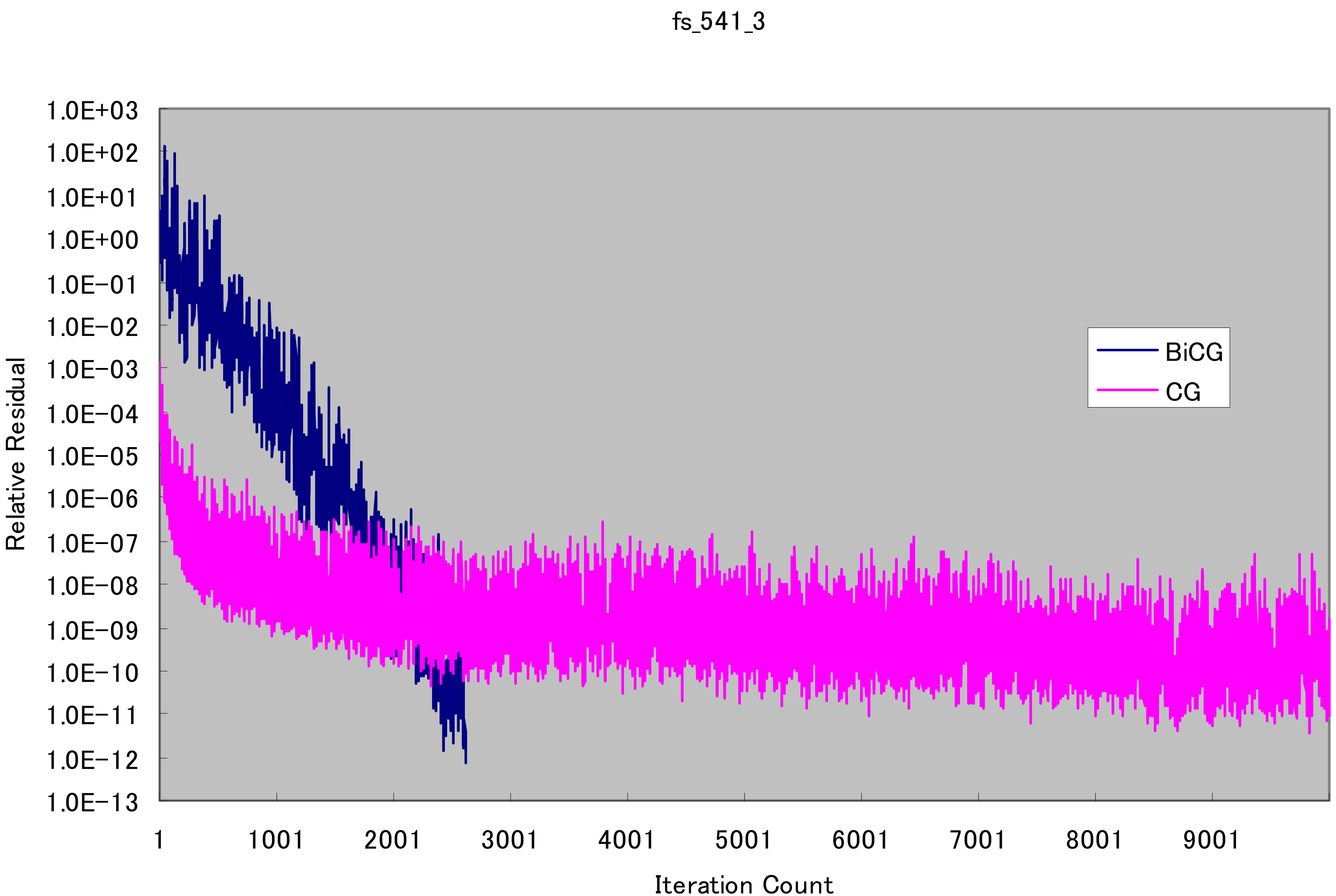
Convergent Both	21	CG is faster 8
CGNR only	26	Good 5
BiCG only	7	
Not Convergent	26	

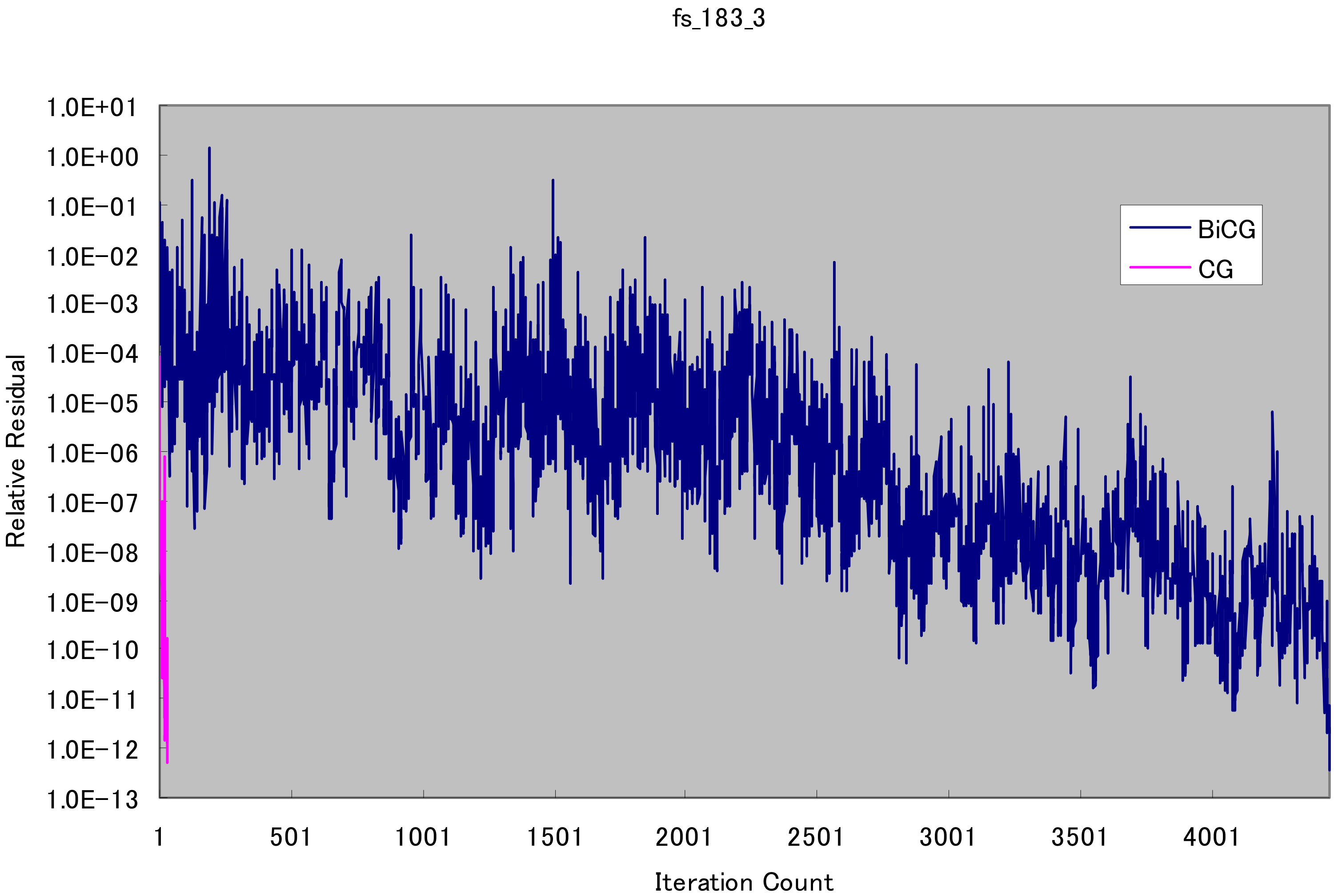


BiCG for Ax=b
Relative Residaul
Condition Number
y = 7E-14x0.5921
1.0E+00
1.0E-01
1.0E-02
1.0E-03
1.0E-04
1.0E-05
1.0E-06
1.0E-07
1.0E-08
1.0E-09
1.0E-10
1.0E-11
1.0E-12
1.0E-13
1.0E+00
1.0E+02
1.0E+04
1.0E+06
1.0E+08
1.0E+10
1.0E+12
1.0E+14


CG for A'Ax=A'b
1.0E+02
1.0E+01
1.0E+00
1.0E-01
1.0E-02
1.0E-03
1.0E-04
1.0E-05
1.0E-06
1.0E-07
1.0E-08
1.0E-09
1.0E-10
1.0E-11
1.0E-12
1.0E-13
1.0E-14
Relative Residual
y = 1E-13x^0.2684
1.0E+00
1.0E+05
1.0E+10
1.0E+15
1.0E+20
1.0E+25
1.0E+30
1.0E+35
1.0E+40
1.0E+45
Condition Number of A'A


add32
1.0E+00
1.0E-01
1.0E-02
1.0E-03
1.0E-04
1.0E-05
1.0E-06
1.0E-07
1.0E-08
1.0E-09
1.0E-10
1.0E-11
1.0E-12
1.0E-13
Relative Residual
1
101
201
301
401
501
601
701
801
Iteration Count
BiCG
CG


fs_541_3
Relative Residual
1.0E+03
1.0E+02
1.0E+01
1.0E+00
1.0E-01
1.0E-02
1.0E-03
1.0E-04
1.0E-05
1.0E-06
1.0E-07
1.0E-08
1.0E-09
1.0E-10
1.0E-11
1.0E-12
1.0E-13
BiCG
CG
1
1001
2001
3001
4001
5001
6001
7001
8001
9001
Iteration Count

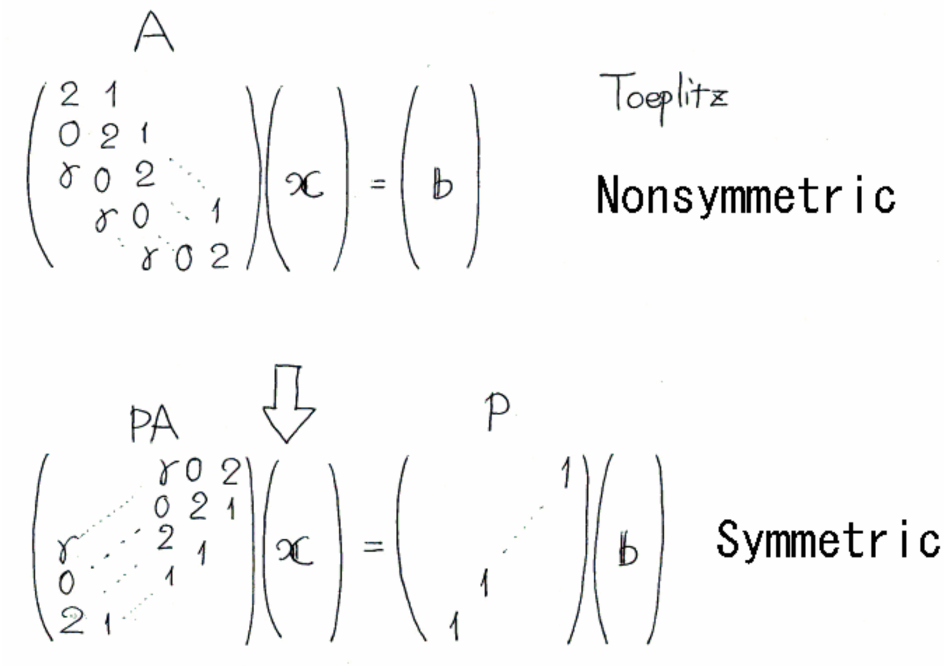

fs_183_3
1.0E+01
1.0E+00
1.0E-01
1.0E-02
1.0E-03
1.0E-04
1.0E-05
1.0E-06
1.0E-07
1.0E-08
1.0E-09
1.0E-10
1.0E-11
1.0E-12
1.0E-13
Relative Residual
1
501
1001
1501
2001
2501
3001
3501
4001
Iteration Count
BiCG
CG

Constructing A symmetric Matrix

- $$\begin{pmatrix} & A^T \\ A & \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix} = \begin{pmatrix} c \\ b \end{pmatrix}$$ Choice of c and x_0
- $$\frac{1}{2} \begin{pmatrix} A + A^T & (A - A^T) \\ A^T - A & -(A + A^T) \end{pmatrix} \begin{pmatrix} x \\ x \end{pmatrix} = \begin{pmatrix} b \\ -b \end{pmatrix}$$
- Symmetric
- Same Condition Number
- Dimension of a target matrix is 2N

Persymmetric matrices

- The best method (CG, BiCG, BiCGSTAB, CGS, QMR) for a matrix A and PA depends on the problems
- Good conv. for CG
- Same Cond. #



Other Usage (Controling λ and δ)

Ayachour

$$\begin{pmatrix} \lambda I & A \\ A^T & 0 \end{pmatrix}$$ as Preconditioner

$\lambda x + A^T x' = b'$

$\lambda \to 0$

Saunders

$$\begin{pmatrix} \delta I & A \\ A^T & -\delta I \end{pmatrix}$$ as Iterative Method

CG method for Nonsymmetric matrix

初期ベクトル $\boldsymbol{x}_0$, $\boldsymbol{y}_0$ を用意

$$\boldsymbol{r}_0 = \boldsymbol{b} - A\boldsymbol{x}_0; \quad \boldsymbol{r'}_0 = \boldsymbol{b'} - A^T\boldsymbol{y}_0$$

$$\boldsymbol{p}_0 = \boldsymbol{r}_0, \quad \boldsymbol{p'}_0 = \boldsymbol{r'}_0$$

for $k = 0, 1, \cdots$

$$\alpha_k = \frac{[\,(\boldsymbol{r}_k, \boldsymbol{r}_k) + (\boldsymbol{r'}_k, \boldsymbol{r'}_k)\,]}{[\,(\boldsymbol{p}_k, A\boldsymbol{p'}_k) + (\boldsymbol{p'}_k, A^T\boldsymbol{p}_k)\,]} \quad \left\{ = \frac{\|\boldsymbol{r}_k\|^2 + \|\boldsymbol{r'}_k\|^2}{2\,(\boldsymbol{p}_k, A\boldsymbol{p'}_k)} \right\}$$

$$\boldsymbol{x}_{k+1} = \boldsymbol{x}_k + \alpha_k \boldsymbol{p'}_k \quad \{; \quad \boldsymbol{y}_{k+1} = \boldsymbol{y}_k + \alpha_k \boldsymbol{p}_k\}$$

$$\boldsymbol{r}_{k+1} = \boldsymbol{r}_k - \alpha_k A\boldsymbol{p'}_k; \quad \boldsymbol{r'}_{k+1} = \boldsymbol{r'}_k - \alpha_k A^T\boldsymbol{p}_k$$

収束判定

$$\beta_k = \frac{[\,(\boldsymbol{r}_{k+1}, \boldsymbol{r}_{k+1}) + (\boldsymbol{r'}_{k+1}, \boldsymbol{r'}_{k+1})\,]}{[\,(\boldsymbol{r}_k, \boldsymbol{r}_k) + (\boldsymbol{r'}_k, \boldsymbol{r'}_k)\,]}$$

$$\boldsymbol{p}_{k+1} = \boldsymbol{r}_{k+1} + \beta_k \boldsymbol{p}_k; \quad \boldsymbol{p'}_{k+1} = \boldsymbol{r'}_{k+1} + \beta_k \boldsymbol{p'}_k$$

BiCG method

初期ベクトル $\boldsymbol{x}_0$, $\tilde{\boldsymbol{x}}_0$ を用意

$\boldsymbol{r}_0 = \boldsymbol{b} - A\boldsymbol{x}_0$; $\tilde{\boldsymbol{r}}_0 = \tilde{\boldsymbol{b}} - A^T\tilde{\boldsymbol{x}}_0$

$\boldsymbol{p}_0 = \boldsymbol{r}_0$; $\tilde{\boldsymbol{p}}_0 = \tilde{\boldsymbol{r}}_0$

for $k = 0, 1, \cdots$

$\alpha_k = \frac{(\boldsymbol{r}_k, \tilde{\boldsymbol{r}}_k)}{(A\boldsymbol{p}_k, \tilde{\boldsymbol{p}}_k)}$

$\boldsymbol{x}_{k+1} = \boldsymbol{x}_k + \alpha_k \boldsymbol{p}_k$

$\boldsymbol{r}_{k+1} = \boldsymbol{r}_k - \alpha_k A\boldsymbol{p}_k$; $\tilde{\boldsymbol{r}}_{k+1} = \tilde{\boldsymbol{r}}_k - \alpha_k A^T\tilde{\boldsymbol{p}}_k$

収束判定

$\beta_k = \frac{(\boldsymbol{r}_{k+1}, \tilde{\boldsymbol{r}}_{k+1})}{(\boldsymbol{r}_k, \tilde{\boldsymbol{r}}_k)}$

$\boldsymbol{p}_{k+1} = \boldsymbol{r}_{k+1} + \beta_k \boldsymbol{p}_k$; $\tilde{\boldsymbol{p}}_{k+1} = \tilde{\boldsymbol{r}}_{k+1} + \beta_k \tilde{\boldsymbol{p}}_k$

			(1) BiCG		(2) Expanded systems with CG											
					c = b		c = Pb		c = $(1, 0, \ldots, 0)^T$		c = $(1, 1, \ldots, 1)^T$		c = 乱数 * $\lVert b \rVert$		c = 乱数	
Matrix	N	NNZ	#iters	Residual	#iters	Residual	#iters	Residual	#iters	Residual	#iters	Residual	#iters	Residual	#iters	Residual
add20	2,395	17,319	542	9.99E-13	over		over		over		over		over		over	
add32	4,960	23,884	124	5.46E-13	1,763	7.99E-13	1,984	6.75E-13	1,908	1.17E-12	2,256	3.69E-11	2,224	1.24E-11	2,324	3.79E-11
arc130	130	1,282	18	8.28E-13	3,361	3.00E-05	2,899	7.21E-05	over		3,472	7.15E-10	over		2,754	2.74E-11
bp__1000	822	4,661	over		over		over		over		over		over		over	
bp__1200	822	4,726	over		over		over		over		over		over		over	
bp__1400	822	4,790	over		over		over		over		over		over		over	
bp__1600	822	4,841	over		over		over		over		over		over		over	
bp__200	822	3,802	over		over		over		over		over		over		over	
bp__400	822	4,028	over		over		over		over		over		over		over	
bp__600	822	4,172	over		over		over		over		over		over		over	
bp__800	822	4,534	over		over		over		over		over		over		over	
bp___0	822	3,276	over		over		over		over		over		over		over	
fs_183_1	183	1,069	1,300	3.47E-13	over		over		over		over		over		over	
fs_183_3	183	1,069	4,447	3.63E-13	over		over		over		over		over		over	
fs_183_4	183	1,069	476	6.94E-13	over		over		over		over		over		over	
fs_183_6	183	1,069	565	1.10E-13	over		over		over		over		over		over	
fs_541_1	541	4,285	15	4.65E-13	105	3.73E-12	85	5.62E-13	96	5.85E-13	104	5.78E-12	111	3.12E-11	87	7.43E-13
fs_541_2	541	4,285	1,008	7.34E-13	over		over		over		over		over		over	
fs_541_3	541	4,285	2,616	7.42E-13	over		over		over		over		over		over	
fs_541_4	541	4,285	1,316	9.16E-13	over		over		over		over		over		over	
fs_680_1	680	2,646	462	8.73E-13	over		over		over		over		over		over	
fs_680_2	680	2,646	1,113	7.34E-13	over		over		over		over		over		over	
fs_680_3	680	2,646	4,317	9.87E-13	over		over		over		over		over		over	
fs_760_1	760	5,976	141	9.18E-13	3,132	8.51E-13	2,602	2.90E-12	8,352	1.10E-12	over		3,423	5.86E-12	over	
fs_760_2	760	5,976	over		over		over		over		over		over		over	
fs_760_3	760	5,976	over		over		over		over		over		over		over	
gemat11	4,929	33,185	over		over		over		over		over		over		over	
gemat12	4,929	33,111	over		over		over		over		over		over		over	
gre_1107	1,107	5,664	over		over		over		over		over		over		over	
gre_216a	216	876	223	7.28E-13	252	9.50E-13	256	4.39E-13	434	5.20E-13	253	6.56E-13	454	1.04E-12	440	8.41E-13
gre_216b	216	876	over		over		over		over		over		over		over	
gre__115	115	421	85	5.60E-13	216	8.47E-13	218	3.47E-13	218	3.93E-13	217	8.12E-13	224	3.27E-13	218	4.64E-13
gre__185	185	1,005	424	8.46E-13	2,819	5.02E-10	2,519	3.00E-11	2,522	2.15E-09	1,880	9.42E-13	2,765	6.83E-08	2,781	4.62E-09
gre__343	343	1,435	**509**	9.53E-13	368	9.94E-13	364	9.52E-13	592	8.73E-13	353	8.41E-13	802	4.22E-12	620	8.25E-13
gre__512	512	2,192	over		487	8.58E-13	492	8.52E-13	814	8.43E-13	488	5.75E-13	1,166	1.35E-11	834	9.63E-13
impcol_a	207	572	over		over		over		over		over		over		over	
impcol_b	59	312	155	5.22E-13	444	5.95E-10	501	8.27E-10	345	2.86E-11	3,310	2.92E-05	433	1.23E-07	471	3.76E-08
impcol_c	137	411	**668**	6.47E-13	757	7.04E-12	686	8.42E-12	422	4.53E-13	526	2.30E-12	713	6.89E-11	431	7.50E-13
impcol_d	425	1,339	over		2,842	5.75E-12	2,748	2.01E-12	1,804	7.54E-13	3,066	8.73E-11	3,075	4.83E-10	2,573	9.25E-13
impcol_e	225	1,308	over		over		over		over		over		over		over	
jpwh_991	991	6,027	over		1,038	9.04E-13	1,023	9.89E-13	937	9.96E-13	over		950	4.00E-12	922	1.52E-12

			(1) BiCG		(2) Expanded systems with CG											
					c = b		c = Pb		c = (1, 0, ..., 0)^T		c = (1, 1, ..., 1)^T		c = 乱数 * ||b||		c = 乱数	
Matrix	N	NNZ	#iters	Residual	#iters	Residual	#iters	Residual	#iters	Residual	#iters	Residual	#iters	Residual	#iters	Residual
lns_3937	3,937	25,407	over		over		over		over		over		over		over	
lns__131	131	536	over		over		over		over		over		over		over	
lns__511	511	2,796	over		over		over		over		over		over		over	
lnsp3937	3,937	25,407	over		over		over		over		over		over		over	
lnsp_131	131	536	over		over		over		over		over		over		over	
lnsp_511	511	2,796	over		over		over		over		over		over		over	
mahindas	1,258	7,682	over		over		over		over		over		over		over	
mbeacxc	496	49,920	over		over		over		over		over		over		over	
mbeaflw	496	49,920	over		over		over		over		over		over		over	
mbeause	496	41,063	over		over		over		over		over		over		over	
mcca	180	2,659	over		over		over		over		over		over		over	
mcfe	765	24,382	over		over		over		over		over		over		over	
memplus	17,758	######	1,675	8.19E-13	over		over		over		over		over		over	
nnc1374	1,374	8,606	over		over		over		over		over		over		over	
nnc261	261	1,500	over		over		over		over		over		over		over	
nnc666	666	4,044	over		over		over		over		over		over		over	
orani678	2,529	90,158	over		7,978	5.82E-12	8,126	1.85E-12	7,876	8.26E-13	8,361	4.98E-10	8,655	2.37E-09	8,134	2.59E-11
sherman2	1,080	23,094	over		over		over		over		over		over		over	
sherman5	3,312	20,793	2,022	4.83E-13	over		over		over		over		over		over	
shl_200	663	1,726	over		4,125	2.20E-07	3,273	2.34E-10	over		4,758	7.13E-11	8,113	8.03E-07	4,147	1.05E-10
shl_400	663	1,712	over		6,035	1.88E-06	3,537	1.22E-10	9,117	3.30E-02	over		7,388	1.78E-06	3,066	1.23E-11
shl__0	663	1,687	over		3,575	1.98E-06	2,568	1.68E-11	5,580	1.74E-09	2,562	2.91E-10	9,902	3.15E-05	3,503	1.10E-11
str_200	363	3,068	7,774	1.02E-12	2,433	6.61E-12	2,441	1.05E-10	1,821	8.72E-13	2,354	7.21E-11	3,207	8.28E-09	2,430	2.19E-10
str_400	363	3,157	over		2,499	2.08E-11	2,005	9.10E-13	2,208	8.49E-13	2,453	3.40E-12	2,594	1.30E-10	2,259	1.23E-12
str_600	363	3,279	over		7,146	8.31E-10	8,687	1.36E-08	5,021	7.98E-13	over		7,617	9.71E-08	over	
str__0	363	2,454	2,895	9.05E-13	535	9.77E-13	678	1.07E-12	538	9.33E-13	620	1.83E-12	564	4.38E-13	524	7.30E-13
watt_1	1,856	11,360	327	3.09E-13	over		over		over		over		over		over	
watt_2	1,856	11,550	466	4.46E-13	over		over		over		over		over		over	
west0067	67	294	175	9.51E-13	304	9.19E-13	306	8.87E-13	264	5.03E-13	401	3.30E-12	333	4.34E-13	256	5.86E-13
west0132	132	414	over		over		over		over		over		over		over	
west0156	156	371	over		over		over		over		over		over		over	
west0167	167	507	over		over		over		over		over		over		over	
west0381	381	2,157	over		over		over		over		over		over		over	
west0479	479	1,910	over		over		over		over		over		over		over	
west0497	497	1,727	over		over		over		over		over		over		over	
west0655	655	2,854	over		over		over		over		over		over		over	
west0989	989	3,537	over		over		over		over		over		over		over	
west1505	1,505	5,445	over		over		over		over		over		over		over	
west2021	2,021	7,353	over		over		over		over		over		over		over	

80 Real Nonsymmetric matrices from Matrix Market

$x = (1, \ldots, 1)^T, x_0 = o$

Convergent Both	13	CG is faster 4
CG only	9	
BiCG only	15	
Not	43	

Choice of c is difficult!

	N	(1) BiCG	(2) Expanded systems with CG						
			b	Pb	$(1, 0, ..., 0)^T$	$(1, 1, ..., 1)^T$	rand. * ||b||	random	best/worst
gre_343	343	509	368	364	592	**353**	**802**	620	44.01%
gre_512	512	over	**487**	492	814	488	**1,166**	834	41.77%
impcol_c	137	668	**757**	686	**422**	526	713	431	55.75%
impcol_d	425	over	2,842	2,748	**1,804**	3,066	**3,075**	2,573	58.67%
jpwh_991	991	over	1,038	1,023	937	**over**	950	**922**	N/A
orani678	2529	over	7,978	8,126	**7,876**	8,361	**8,655**	8,134	91.00%
shl_200	663	over	4,125	**3,273**	**over**	4,758	8,113	4,147	N/A
shl_400	663	over	6,035	3,537	**9,117**	over	7,388	**3,066**	33.63%
shl__0	663	over	3,575	2,568	5,580	**2,562**	**9,902**	3,503	25.87%
str_200	363	7,774	2,433	2,441	**1,821**	2,354	**3,207**	2,430	56.78%
str_400	363	over	2,499	**2,005**	2,208	2,453	**2,594**	2,259	77.29%
str_600	363	over	7,146	8,687	**5,021**	**over**	7,617	**over**	N/A
str__0	363	2,895	535	**678**	538	620	564	**524**	77.29%

Red: bad choice

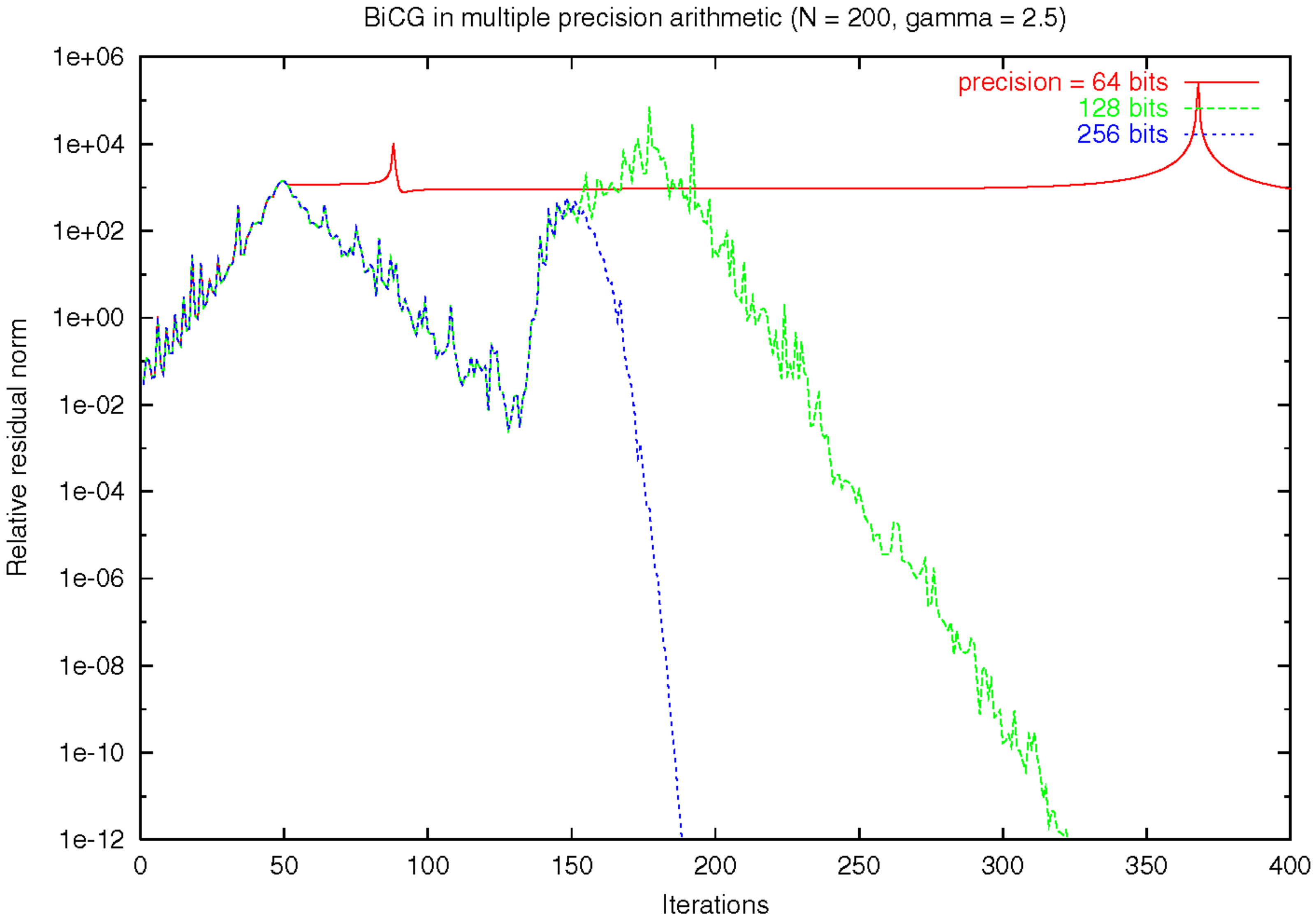
Difficulty to Numerical Computation

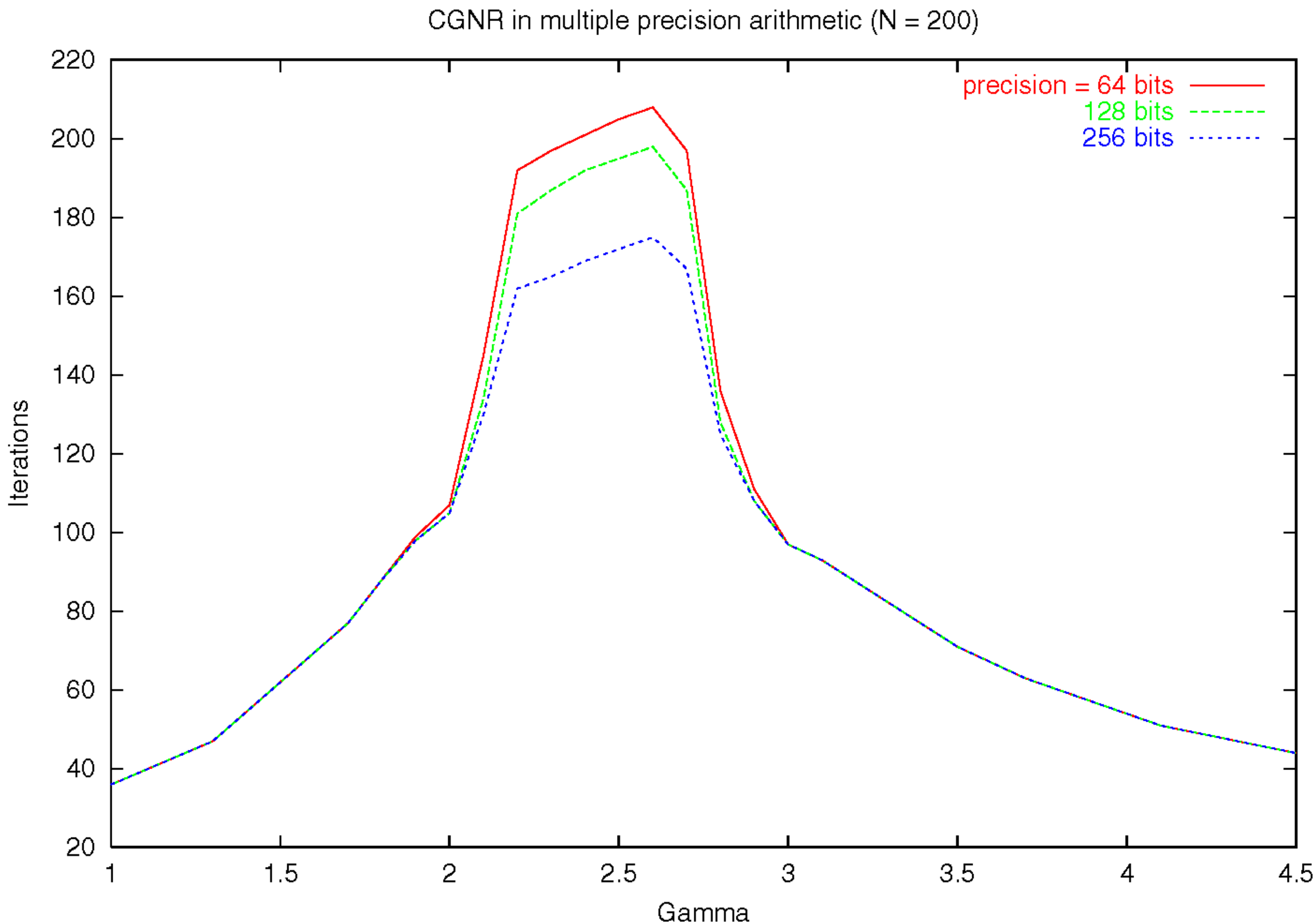
- Convergence of Krylov subspace method is affected by round-off error
- Use multiple precission arithmetic for more accurate computation

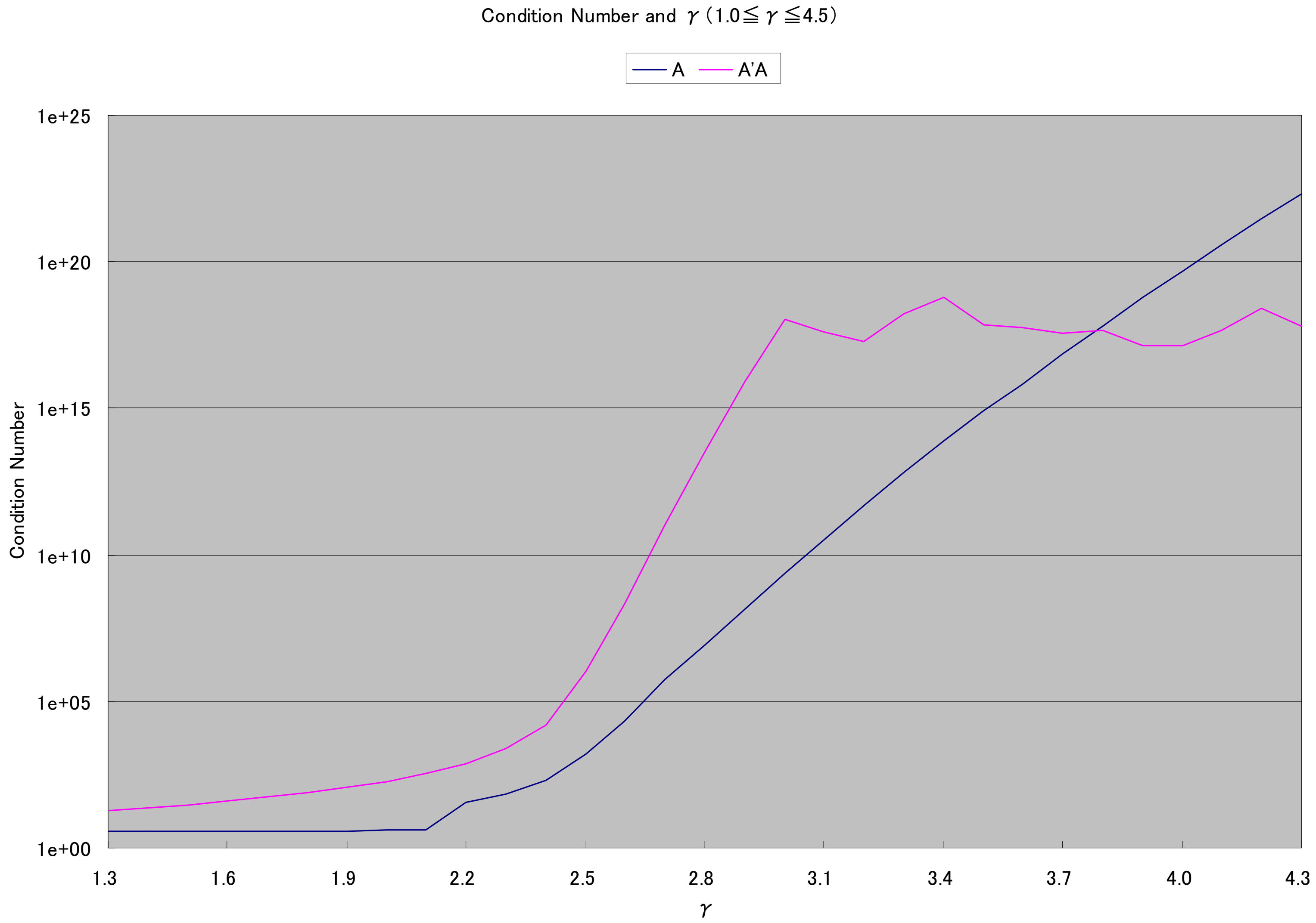
Toeplitz Matrix, N = 200, $\gamma = 1.7$

$$A := \begin{bmatrix} 2 & 1 & & & \\ 0 & 2 & 1 & & \\ \gamma & 0 & 2 & 1 & \\ & \gamma & 0 & 2 & \ddots \\ & & \ddots & \ddots & \ddots \end{bmatrix}$$

Right-hand side $b = (1, 1, \cdots, 1)^T$


BiCG in multiple precision arithmetic (N = 200, gamma = 2.5)
precision = 64 bits
128 bits
256 bits
Relative residual norm
Iterations
1e+06
1e+04
1e+02
1e+00
1e-02
1e-04
1e-06
1e-08
1e-10
1e-12
0
50
100
150
200
250
300
350
400


CGNR in multiple precision arithmetic (N = 200)
Iterations
Gamma
precision = 64 bits
128 bits
256 bits
220
200
180
160
140
120
100
80
60
40
20
1
1.5
2
2.5
3
3.5
4
4.5



Condition Number and γ (1.0≦γ≦4.5)
A
A'A
Condition Number
1e+25
1e+20
1e+15
1e+10
1e+05
1e+00
1.3
1.6
1.9
2.2
2.5
2.8
3.1
3.4
3.7
4.0
4.3
γ

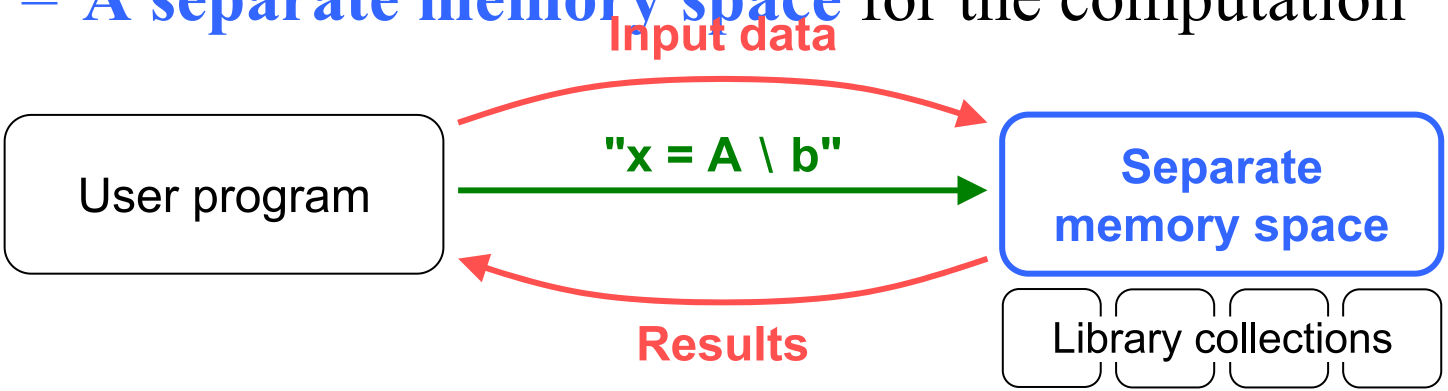
Conclusion (tentative)

- Symmetricity is a good property
- CG may be a good method
  + CGNR is robust
  + Expanded system is good for some problems
    (difficult to control artificial equation)
  + Accurate arithmetic as a choice
- Many tests are needed
- Different precision arithmetic should be tested

Are there any good tools?

SILC: Simple Interface for Library Collections

- Basic ideas
  - **Data transfer** and **a request for computation**
  - **Mathematical expressions** for the request
  - **A separate memory space** for the computation



Solving a system of $Ax=b$

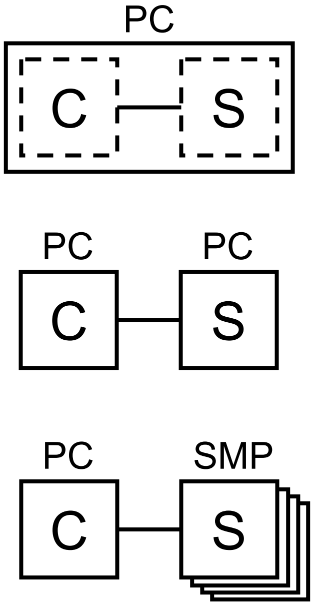
- In the traditional way (using LAPACK in C)

```
double *A, *b;
int kl, ku, lda, ldb, nrhs, info, *ipiv;

dgbtrf (N, N, kl, ku, A, lda, ipiv, &info);  /* LU factorization */
if (info == 0)
    dgbtrs ('N', N, kl, ku, nrhs, A, lda, ipiv, b, ldb, &info);  /* solve */
```

- In SILC

```
silc_envelope_t A, b, x;

SILC_PUT ("A", &A);
SILC_PUT ("b", &b);
SILC_EXEC ("x = A \ b");  /* call a solver (e.g., dgbtrf & dgbtrs) */
SILC_GET (&x, "x");
```

Benefits of using SILC



- Source-level independence between user programs and matrix computation libraries
  - Easy access to alternative solvers and matrix storage formats, possibly in other libraries
  - Instant porting to other computing environments without any modification in user programs
- You need to prepare only the smallest amount of data
  - Temporary buffers are automatically allocated
- Language-independent mathematical expressions
  - Applicable in many programming languages (C, Fortran, Python)

SILC Code for this project

```
SILC_PUT("A", &A);
SILC_PUT("b", &b);

SILC_EXEC("prefer leq_lis");

SILC_EXEC("lis_options[LIS_OPTIONS_SOLVER] = LIS_SOLVER_BICG");
SILC_EXEC("start = time()");
SILC_EXEC("x = A ¥¥ b");
SILC_EXEC("end = time()");

SILC_EXEC("lis_options[LIS_OPTIONS_SOLVER] = LIS_SOLVER_CG");
SILC_EXEC("start = time()");
SILC_EXEC("AT = A'");
SILC_EXEC("ATA = AT * A");
SILC_EXEC("ATb = AT * b");
SILC_EXEC("x = ATA ¥¥ ATb");
SILC_EXEC("end = time()");

SILC_EXEC("c1 = 1.0 / rcond(A)");
SILC_EXEC("c2 = 1.0 / rcond(A' * A)");

 SILC_EXEC("h = lis_get_residual_history()");
```

Advertisement

- SILC version 1.1 is freely available
  - For UNIX environments
  - Sample programs in C, Fortran, and Python
  - Documentations
- Visit the SILC home page for more info

http://ssi.is.s.u-tokyo.ac.jp/silc/

Functionalities

- Data structures
  - Data types: scalar, vector, matrix, cubic array
  - Precisions: integer, real, complex (single or double)
  - Matrix storage formats: dense, banded, CRS
- Mathematical expressions
  - Binary arithmetic operators (+, −, *, /, %)
  - Solutions of systems of linear equations (A \ b)
  - Conjugate transposes (A'), complex conjugates (A~)
  - Built-in functions
    - Ex. "sqrt(b' * b)" is the 2-norm of vector b
  - Subscript
    - Ex. "A[1:5,1:5]" is a 5×5 submatrix of A