An Analysis of ML(n)BiCGSTAB

Man-Chung Yeung
Mathematics, Wyoming
Feb., 2008

Index Functions (Boley)

- Let k, n be positive integers. Define

$$g_n(k) = \lfloor (k-1)/n \rfloor, \qquad r_n(k) = k - n\, g_n(k),$$

where $\lfloor \cdot \rfloor$ rounds its argument to the nearest integer towards minus infinity.

- If write $k = jn + i$ with $j \geq 0$ and $1 \leq i \leq n$, then

$$g_n(jn+i) = j \quad \text{and} \quad r_n(jn+i) = i.$$

ML(n)BiCGSTAB

- ML(n)BiCGSTAB is a Krylov subspace method (1997).
- Properties:
  1. Built on multiple starting Lanczos procedure
  2. A natural generalization of BiCGSTAB. In fact, ML(1)BiCGSATB = BiCGSTAB.
  3. No A^T is used in implementation.
  4. On average, $1 + \dfrac{1}{n}$ matrix-vector multiplication per iteration.
  5. At iteration step k, it searches an approx. solution

$$\mathbf{x}_k \in \mathbf{x}_0 + span\{\mathbf{r}_0, \mathbf{A}\mathbf{r}_0, \mathbf{A}^2\mathbf{r}_0, \cdots, \mathbf{A}^{\lfloor (k-1)/n \rfloor + k}\mathbf{r}_0\}.$$

  6. Significantly faster than BiCGSTAB with appropriate n.

Derivation of ML(n)BiCGSTAB (1997 Version)

- Recall that BiCGSTAB was derived from Bi-CG.

- At iteration step k, BiCGSTAB searches

$$\mathbf{x}_k \in \mathbf{x}_0 + span\{\mathbf{r}_0, \mathbf{A}\mathbf{r}_0, \mathbf{A}^2\mathbf{r}_0, \cdots, \mathbf{A}^{2k-1}\mathbf{r}_0\},$$

$$\mathbf{r}_k^{BiCGSTAB} \in span\{\mathbf{r}_0, \mathbf{A}\mathbf{r}_0, \mathbf{A}^2\mathbf{r}_0, \cdots, \mathbf{A}^{2k}\mathbf{r}_0\}.$$

- $$\mathbf{r}_k^{BiCGSTAB} = \phi_k(\mathbf{A})\mathbf{r}_k^{BiCG}$$

  where $\phi_k(\lambda)$ is a polynomial of degree k, given by

$$\phi_0(\lambda) = 1 \quad \text{and} \quad \phi_k(\lambda) = (\rho_k\lambda + 1)\,\phi_{k-1}(\lambda).$$

- Similarly, to derive ML(n)BiCGSTAB
  1. First develope ML(n)BiCGS — a Bi-CG like method — based on multiple Lanczos
  2. Then, define

$$\mathbf{r}_k^{MLBiCGSTAB} = \phi_{g_n(k)+1}(\mathbf{A})\mathbf{r}_k^{MLBiCG}. \tag{1}$$

For example, if $n = 3$, then

$$\begin{aligned}
\mathbf{r}_1^{MLBiCGSTAB} &= \phi_1(\mathbf{A})\,\mathbf{r}_1^{MLBiCG} \\
\mathbf{r}_2^{MLBiCGSTAB} &= \phi_1(\mathbf{A})\,\mathbf{r}_2^{MLBiCG} \\
\mathbf{r}_3^{MLBiCGSTAB} &= \phi_1(\mathbf{A})\,\mathbf{r}_3^{MLBiCG} \\
\mathbf{r}_4^{MLBiCGSTAB} &= \phi_2(\mathbf{A})\,\mathbf{r}_4^{MLBiCG} \\
\mathbf{r}_5^{MLBiCGSTAB} &= \phi_2(\mathbf{A})\,\mathbf{r}_5^{MLBiCG} \\
\mathbf{r}_6^{MLBiCGSTAB} &= \phi_2(\mathbf{A})\,\mathbf{r}_6^{MLBiCG}
\end{aligned}$$

- We call every 3 consecutive iterations a cycle.

  For example,

  iterations $1, 2, 3$ is a cycle,

  iterations $4, 5, 6$ is another cycle, and so on.

- The degree of ϕ is increased by 1 at the beginning of a cycle.

- With Definition (1), ML(n)BiCGSTAB searches

$$\mathbf{x}_k^{MLBiCGSTAB} \in \mathbf{x}_0 + span\{\mathbf{r}_0, \mathbf{A}\mathbf{r}_0, \cdots, \mathbf{A}^{g_n(k)+k}\mathbf{r}_0\}$$
$$\mathbf{r}_k^{MLBiCGSTAB} \in span\{\mathbf{r}_0, \mathbf{A}\mathbf{r}_0, \cdots, \mathbf{A}^{g_n(k)+k+1}\mathbf{r}_0\}$$

  The dimension of the Krylov subspace in which $\mathbf{x}_k^{MLBiCGSTAB}$ is searched is

$$g_n(k) + k + 1.$$

- With Definition (1), the resulting ML(n)BiCGSTAB algorithm has the following:

Table 1: Average cost per iteration step and storage requirement.

Matvec	$1 + \frac{1}{n}$
dot product	$n + 1 + \frac{2}{n}$
Vector addition	$2 - \frac{2}{n}$
Saxpy	$\max(2.5n + 3.5 - \frac{1}{n}, 6)$
Storage	$\mathbf{A} + \mathbf{M} + (4n-1)N$ $+O(n) + O(N)$

Other Definitions

- In Definition (1), we increased the degree of ϕ by 1 at the beginning of a cycle.
- It turns out that we can increase the degree of ϕ anywhere within a cycle.

  For example, if $n = 3$, then

$$
\begin{aligned}
\mathbf{r}_1^{MLBiCGSTAB} &= \phi_0(\mathbf{A})\,\mathbf{r}_1^{MLBiCG} \\
\mathbf{r}_2^{MLBiCGSTAB} &= \phi_1(\mathbf{A})\,\mathbf{r}_2^{MLBiCG} \\
\mathbf{r}_3^{MLBiCGSTAB} &= \phi_1(\mathbf{A})\,\mathbf{r}_3^{MLBiCG} \\
\mathbf{r}_4^{MLBiCGSTAB} &= \phi_1(\mathbf{A})\,\mathbf{r}_4^{MLBiCG} \\
\mathbf{r}_5^{MLBiCGSTAB} &= \phi_2(\mathbf{A})\,\mathbf{r}_5^{MLBiCG} \\
\mathbf{r}_6^{MLBiCGSTAB} &= \phi_2(\mathbf{A})\,\mathbf{r}_6^{MLBiCG}
\end{aligned}
$$

- As an illustration, let us increase the degree of ϕ by 1 at the end of a cycle by defining

$$\mathbf{r}_k^{MLBiCGSTAB} = \phi_{g_n(k+1)}(\mathbf{A})\mathbf{r}_k^{MLBiCG}. \tag{2}$$

For example, if $n = 3$, then

$$\begin{aligned}
\mathbf{r}_1^{MLBiCGSTAB} &= \phi_0(\mathbf{A})\,\mathbf{r}_1^{MLBiCG} \\
\mathbf{r}_2^{MLBiCGSTAB} &= \phi_0(\mathbf{A})\,\mathbf{r}_2^{MLBiCG} \\
\mathbf{r}_3^{MLBiCGSTAB} &= \phi_1(\mathbf{A})\,\mathbf{r}_3^{MLBiCG} \\
\mathbf{r}_4^{MLBiCGSTAB} &= \phi_1(\mathbf{A})\,\mathbf{r}_4^{MLBiCG} \\
\mathbf{r}_5^{MLBiCGSTAB} &= \phi_1(\mathbf{A})\,\mathbf{r}_5^{MLBiCG} \\
\mathbf{r}_6^{MLBiCGSTAB} &= \phi_2(\mathbf{A})\,\mathbf{r}_6^{MLBiCG}
\end{aligned}$$

- With Definition (2), ML(n)BiCGSTAB searches

$$\mathbf{x}_k^{MLBiCGSTAB} \in \mathbf{x}_0 + span\{\mathbf{r}_0, \mathbf{A}\mathbf{r}_0, \cdots, \mathbf{A}^{g_n(k+1)+k-1}\mathbf{r}_0\}$$
$$\mathbf{r}_k^{MLBiCGSTAB} \in span\{\mathbf{r}_0, \mathbf{A}\mathbf{r}_0, \cdots, \mathbf{A}^{g_n(k+1)+k}\mathbf{r}_0\}$$

  The dimension of the Krylov subspace in which $\mathbf{x}_k^{MLBiCGSTAB}$ is searched is

$$g_n(k+1) + k.$$

- Compared to Def. (1), Def. (2) has advantages:
  1. At each step, ML(n)BiCGSTAB with Def. (2) searches for an approx. solution in a smaller space.
  2. The resulting algorithm with Def. (2) requires about 25% less in storage.
  3. Slightly cheaper in terms of computational cost.
- Def. (2) has disadvantages:

  The computed $\mathbf{r}_k$ easily diverges from the exact $\mathbf{b} - \mathbf{A}\mathbf{x}_k$.

Numerical Experiments

- Run

  ML(n)BiCGSTAB associated with Def. (1)

  ML(n)BiCGSTAB associated with Def. (2)

  for $n = 1, 2, \cdots, 100$.

- Example #1:
  1. Test data labeled *e20r0100* from SPARSKIT.
  2. $\mathbf{A}$ is 4241×4241 real unsymmetric.
  3. Preconditioner: $ILU(0)$.
  4. Initial: $\mathbf{x}_0 = \mathbf{0}$.
  5. Stop: $\|\mathbf{r}_k\|/\|\mathbf{r}_0\| < 10^{-7}$

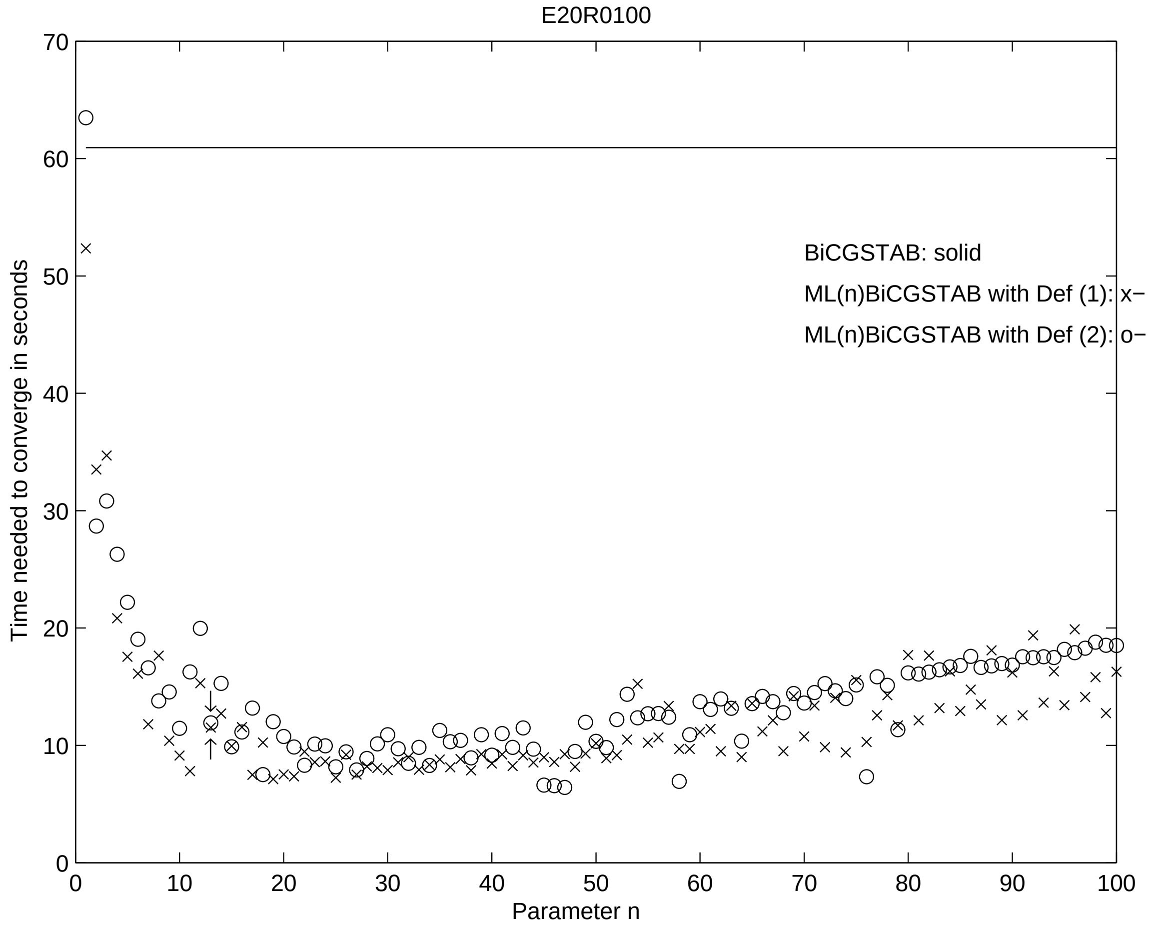

Figure 1: Example #1. Time needed to converge

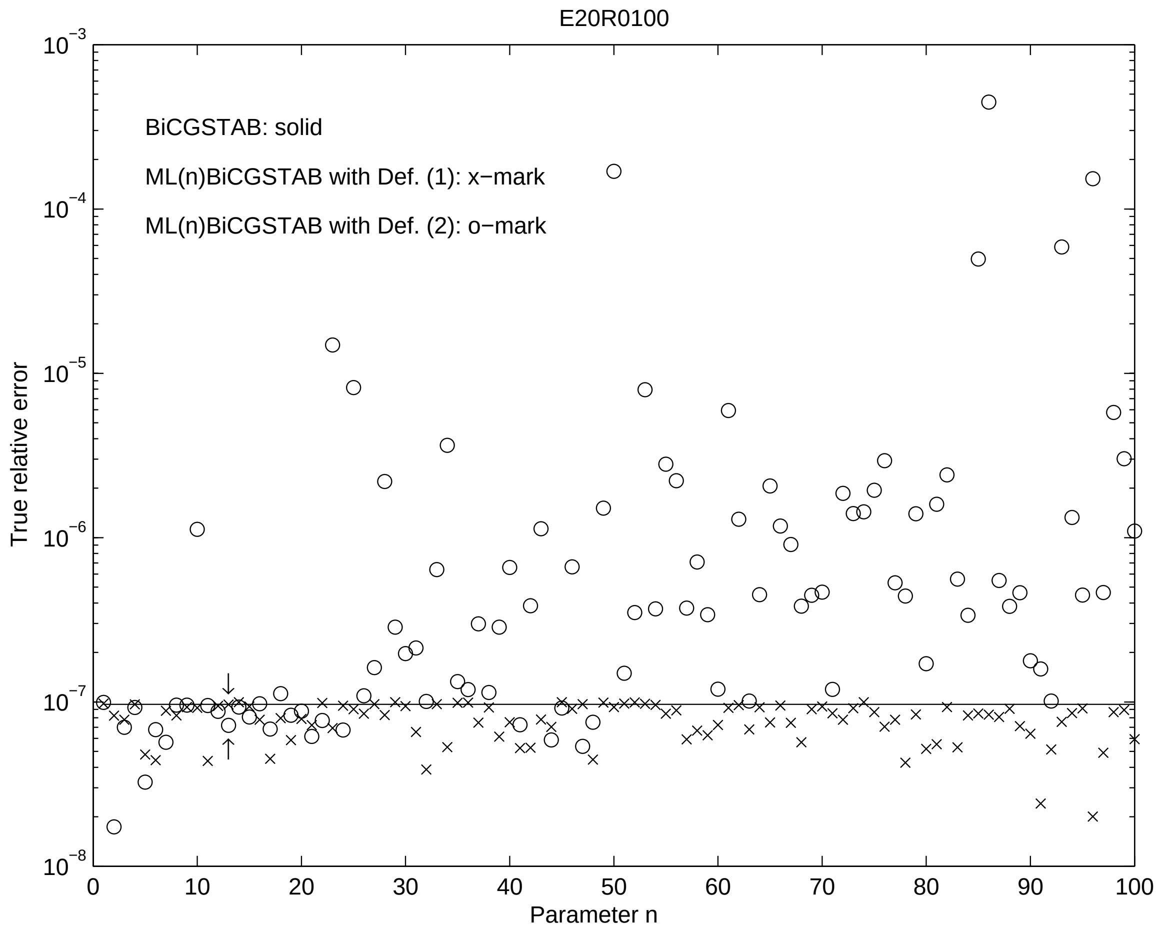

Figure 2: Example #1. True relative error $\|b - Ax\|/\|b\|$.

- Example #2:

  1. Test data labeled *utm5940*.
  2. $\mathbf{A}$ is 5940×5940 real unsymmetric.
  3. $ILU(0)$ preconditioner.
  4. Initial guess $\mathbf{x}_0 = \mathbf{0}$
  5. stopping is $\|r_k\|_2/\|b\|_2 < 10^{-7}$.

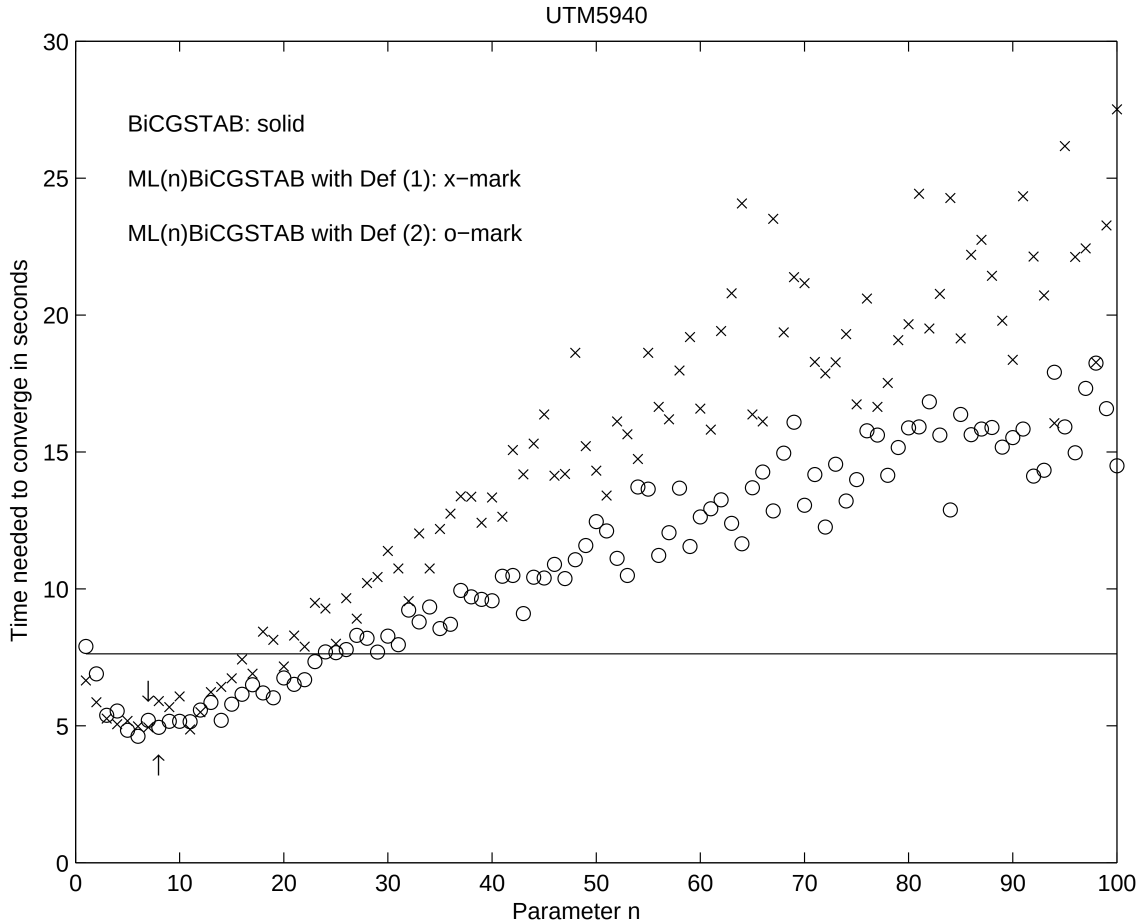

Figure 3: Example #2. Time needed to converge

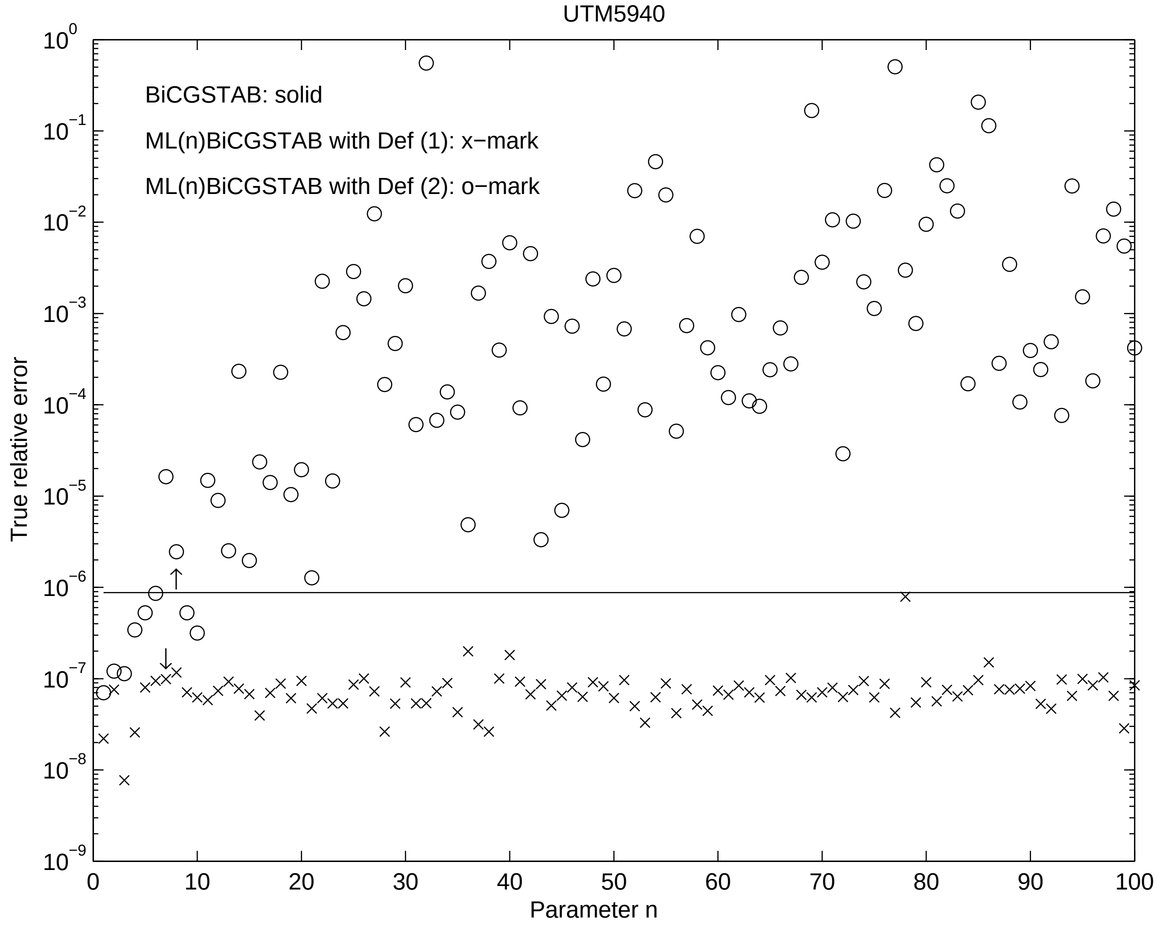

Figure 4: Example #2. True relative error $\|b - Ax\|/\|b\|$.

- Example #3:

1. The test data labeled *QC2534*
2. A is 2534×2534 complex symmetric indefinite
3. $ILU(0)$ preconditioner
4. Initial guess $x_0 = 0$
5. stopping is $\|r_k\|_2 / \|b\|_2 < 10^{-7}$

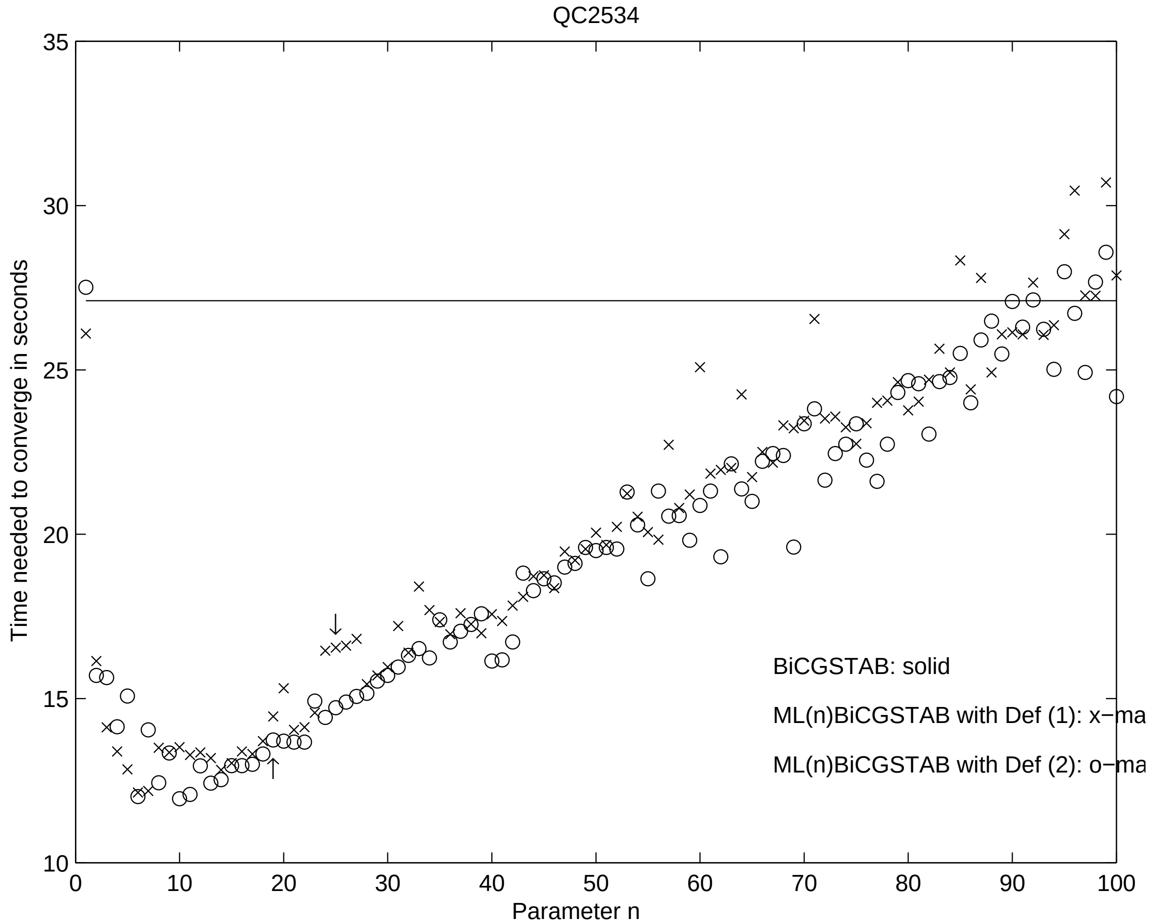

Figure 5: Example #3. Time needed to converge

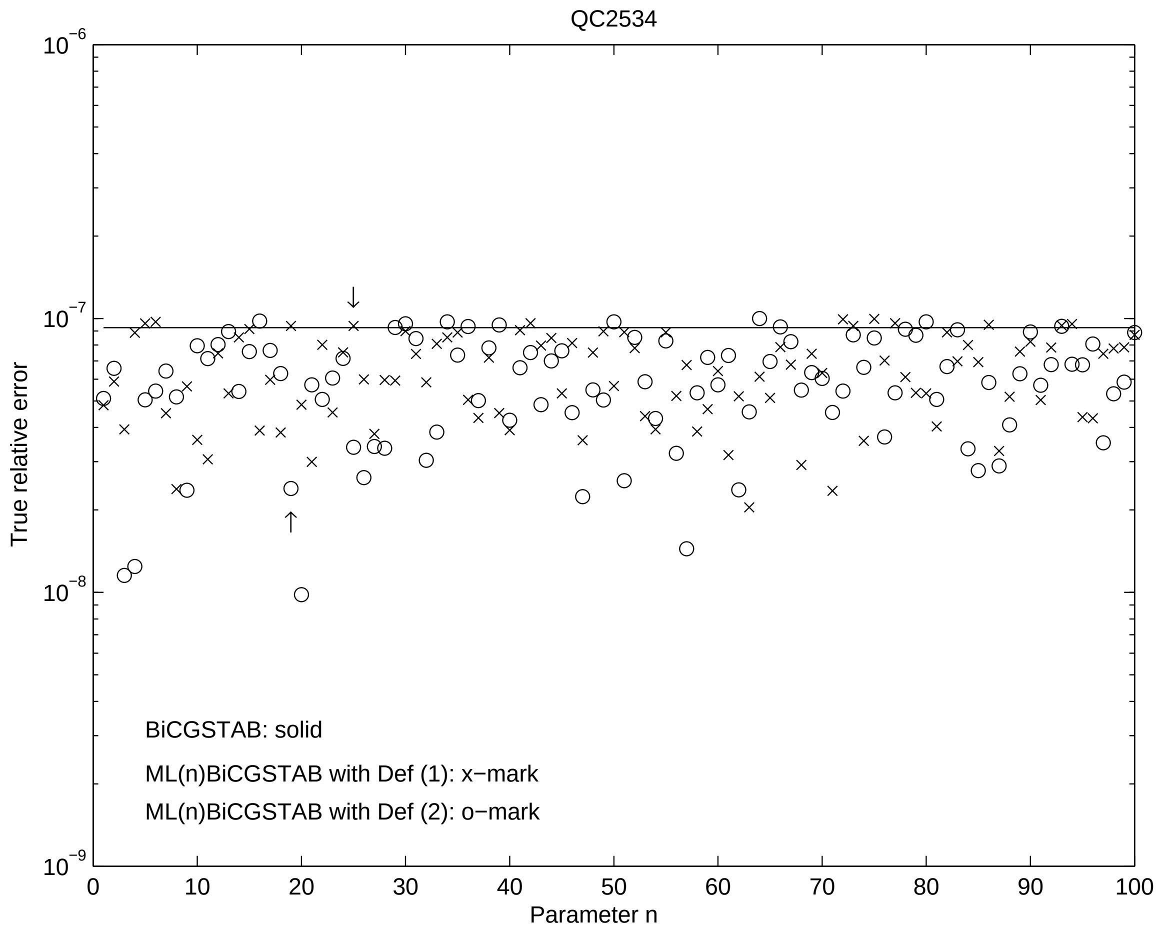

Figure 6: Example #3. True relative error $\|b - Ax\|/\|b\|$

Conclusions

- For ML(n)BiCGSTAB, there are some other ways to define its residual r_k besides the definition in the 1997 paper. One definition will result in one algorithm.

- We presented two definitions and derived the associated algorithms, namely,
  1. Def. (1) which increases the degree of ϕ at the beginning of an iteration cycle;
  2. Def. (2) with the degree of ϕ increased at the end of a cycle.

- ML(n)BiCGSTAB with Def. (2) is cheaper in storage and computational cost, but the computed r_k can easily diverge from its exact residual.

- For a definition of r_k that increases the degree of ϕ somewhere within an iteration cycle, we expect that the performance of the resulting algorithm would lie between those associated with (1) and (2).