Linear Systems with the Exact Solution for Numerical Tests

Katsuhisa Ozaki

Department of Mathematical Sciences,
Shibaura Institute of Technology

Joint work with Takeshi Ogita

SCAN2016, Uppsala, Sweden, Sep. 26th, 2016.

Introduction

We discuss the accuracy of approximate solution of

$$Ax = b, \quad A \in \mathbb{F}^{n\times n}, \quad b \in \mathbb{F}^n, \quad \det(A) \neq 0.$$

$\mathbb{F}$: set of floating-point numbers as defined in IEEE 754.

$\tilde{x}$: approximate solution.

Introduction

The residual $A\tilde{x} - b$ is used for checking the accuracy of the approximation $\tilde{x}$.

However,

$$\|\tilde{x} - A^{-1}b\| \leq \|A^{-1}\| \, \|A\tilde{x} - b\|$$

is satisfied.

It indicates that there may be a case;

the residual is small but the error is big

Introduction

Let $x = (1, 1)^T$ be the exact solution of the linear system $Ax = b$:

$$A = \begin{pmatrix} 1 & 1 \\ 1 & 1.0001 \end{pmatrix}, \quad b = \begin{pmatrix} 2 \\ 2.0001 \end{pmatrix}$$

$$\tilde{x} = (2, 0)^T \implies \|A\tilde{x} - b\|_2 = 1.0 * 10^{-4}$$

$$\tilde{x} = (1.0001, 1)^T \implies \|A\tilde{x} - b\|_2 = 1.414 * 10^{-4}$$

Introduction

- Difference between $\|A\tilde{x} - b\|$ and $\|\tilde{x} - x\|$

  $A \in \mathbb{F}^{n\times n}$ and $x, b \in \mathbb{F}^n$ are nice for numerical tests

- is the vector b obtained by $b := Ax$?
  $\longrightarrow$ rounding error may occur in the evaluation of Ax
  $\longrightarrow Ax \notin \mathbb{F}^n$

Notation

- $\mathbf{u}$: roundoff unit ($\mathbf{u} = 2^{-53}$ for binary64)
- $\mathrm{fl}(\cdots)$: result of numerical computations
- $A \in \mathbb{F}^{n\times n}$ and $x \in \mathbb{F}^n$ are given
- $\tilde{x}$ is an approximate solution

Related Work

For given A and x, Miyajima, Ogita, Oishi (2005) proposed a method which produces $A'x' = b$.

For $a, b \in \mathbb{F}$, we can obtain $x, y \in \mathbb{F}$ such that

$$ab = x + y, \ \ x = \mathrm{fl}(ab).$$

$$Ax = \sum_{i=1}^{k} b^{(i)}, \quad b^{(i)} \in \mathbb{F}^n, \quad k \in \mathbb{N}. \tag{1}$$

Related Work

Let $I \in \mathbb{F}^{(k-1)\times(k-1)}$ be the identity matrix and $O \in \mathbb{F}^{(k-1)\times(k-1)}$ be the zero matrix. Setting

$$B = [-b^{(2)}, \ldots, -b^{(k)}] \in \mathbb{F}^{n\times(k-1)}, \quad e = (1, \ldots, 1)^T \in \mathbb{F}^{k-1},$$

$A' \in \mathbb{F}^{(n+k-1)\times(n+k-1)}$, $x' \in \mathbb{F}^{n+k-1}$ and $b' \in \mathbb{F}^{n+k-1}$ are obtained by

$$A' = \begin{pmatrix} A & B \\ O & I \end{pmatrix}, \; x' = \begin{pmatrix} x \\ e \end{pmatrix}, \; b' = \begin{pmatrix} b^{(1)} \\ e \end{pmatrix}.$$

Advantage and Disadvantage

- Ill-conditioned matrix is treatable
- Setting arbitrary x
- The size is increased
- Structure of the matrix is changed

$$A' = \begin{pmatrix} A & B \\ O & I \end{pmatrix}$$

Proposed Method

- Ill-conditioned matrix is not treatable
- Setting arbitrary x is not possible
- The size is not changed
- Structure of the matrix is not changed
  $(a_{ij} = a_{kl},\ a_{ij} = 0)$

$$A = A' + \Delta$$

$$A' \approx A$$

Strategy

A is divided into

$$A = A' + \Delta, \quad A', \Delta \in \mathbb{F}^{n \times n}$$

in order to satisfy

$$A'x = \mathrm{fl}(A'x), \quad A' \approx A.$$

The Proposed Method

For given $A \in \mathbb{F}^{n\times n}$ and $x \in \mathbb{F}^n$, we explain how to obtain A' and b such that

$$A'x = b, \quad A' \in \mathbb{F}^{n\times n}, \quad b \in \mathbb{F}^n \tag{2}$$

Assume that A is non-singular and x is not zero vector.

Each element of the vector θ satisfies

$$\theta_j = 2^k,\ k \in \mathbb{Z}, \quad x_j \in \theta_j\mathbb{Z}, \quad x_j \notin 2\theta_j\mathbb{Z}.$$

The Proposed Method

The vector σ is defined as

$$\sigma_i := 2^{\beta} \cdot 2^{g_i}, \quad \beta = \lceil \log_2 n \rceil.$$

We set the vector g

$$g_i := \begin{cases} \lceil \log_2 \alpha_i \rceil, & \alpha_i = 0 \\ 1 & \text{otherwise} \end{cases}, \qquad \alpha_i := \max_{1 \le j \le n} |a_{ij} x_j|$$

The Proposed Method

The vector σ is defined as

$$\sigma_i := 2^{\beta} \cdot 2^{g_i}, \quad \beta = \lceil \log_2 n \rceil.$$

If the structure is preserved, then we set the vector g

$$g_i := \begin{cases} \lceil \log_2 \alpha \rceil, & \alpha = 0 \\ 1 & \text{otherwise} \end{cases}, \quad \alpha := \max_{1 \le i,j \le n} |a_{ij} x_j|$$

The Proposed Method

We set the σ'

$$\sigma' = \frac{2}{\min\limits_{\theta_j \neq 0} \theta_j} \cdot \sigma. \tag{3}$$

The matrix A' is obtained by

$$a'_{ij} := \mathrm{fl}((a_{ij} + \sigma'_i) - \sigma'_i). \tag{4}$$

Here, $\mathrm{fl}(A'x) = A'x$ is satisfied.

MATLAB Code, $x = (1, \ldots, 1)^T$

```
function [A2, b] = str_one(A)
  n = length(A);
  y = max(abs(A(:)));
  σ = 2^ceil(log2(n)) * 2.^ceil(log2(y));
  T = repmat(σ, n, n);
  A2 = (A + T) - T;
  b = A2 * ones(n, 1);    %no rounding error
end
```

MATLAB Code for Sparse

```
function [A2, b] = str_one(A)
  n = length(A);
  y = max(abs(A(:)));
  σ = 2^ceil(log2(n)) * 2.^ceil(log2(y));
  T = σ * spones(A);
  A2 = (A + T) - T;
  b = A2 * ones(n, 1);     %no rounding error
end
```

Iterative Refinement

If we set smaller σ', the $Ax = b$ may be satisfied.

If we take small σ' until

$$\mathrm{fl}_{\nabla}(A'x) \neq \mathrm{fl}_{\triangle}(A'x)$$

- fl_{∇}: numerical results with rounding downward mode
- $\mathrm{fl}_{\triangle}$: numerical results with rounding upward mode

Iterative Refinement

n	original A'	after the refinement
100	2.2552e-13	2.0687e-14
500	1.1593e-12	7.9412e-14
1000	2.9518e-12	9.2558e-14
5000	2.8430e-11	4.1641e-13
10000	6.2564e-11	4.7524e-13

Table 1: The maximum of $\dfrac{|a_{ij} - a'_{ij}|}{|a_{ij}|}$

Tests for VNC

We check the tightness of the verified numerical computations.

$$\|\tilde{x} - A^{-1}b\|_\infty \leq \frac{\|R(A\tilde{x} - b)\|_\infty}{1 - \|RA - I\|_\infty} =: \alpha$$

is well known. We set $\tilde{x}$ and check $\dfrac{\alpha}{\|\tilde{x} - A^{-1}b\|_\infty}$.

Test for VNC

S. Oishi, T. Ogita, T. Ohta, Numerical Verification Method for Systems of Linear Equations Using Accurate Computation of Dot Product, Simulation, 25(3), 2006 (in Japanese).

$$\|\tilde{x} - A^{-1}b\|_\infty \leq \frac{\|R(A\tilde{x} - b)\|_\infty}{1 - \|RA - I\|_\infty} =: \alpha$$

We check $\dfrac{\alpha}{\|\tilde{x} - A^{-1}b\|_\infty}$ for $x = (1, 1, \ldots, 1)^T$ and

$\tilde{x} = (c, c, \ldots, c)^T, c \in \mathbb{F}$.

Test for VNC

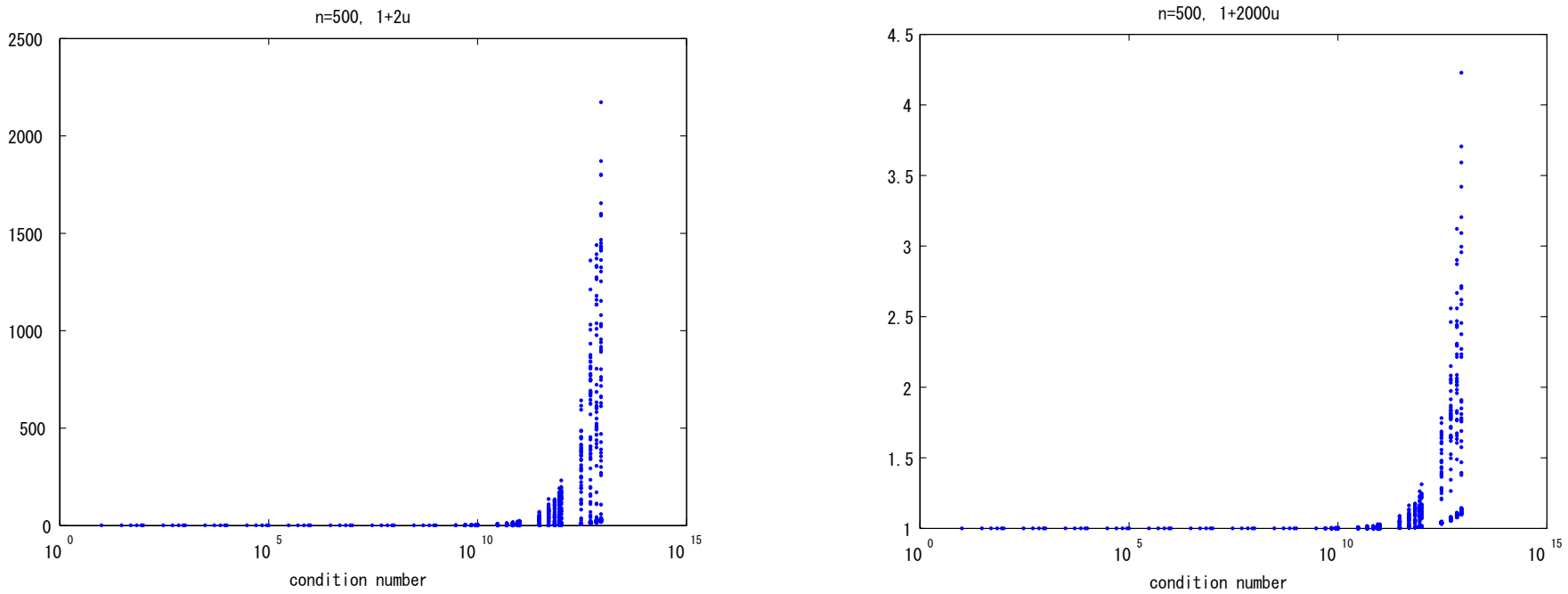


Figure 1: $c = 1 + 2\mathbf{u}$ (left) and $c = 1 + 2000\mathbf{u}$ (right)

Test for CG method

If the exact solution is known in advance, we check the behavior of iterative methods exactly.

We check

$$\|A\tilde{x} - b\|_2, \quad \|r_{k+1}\|_2 = \|r_k - \alpha_k A p_k\|_2, \quad \|x - \tilde{x}\|_2$$

using conjugate gradient (CG) method.

$r_0 = b - Ax_0;$

$p_o = r_0;$

`while` 1

$\alpha_k = (r_k^T r_k)/(p_k^T A p_k);$

$x_{k+1} = x_k + \alpha_k p_k;$

$r_{k+1} = r_k - \alpha_k A p_k;$

`if` $\|r_{k+1}\|_2 < 1e-10,$ `break;` , `end`

$\beta_k = (r_{k+1}^T r_{k+1})/(r_k^T r_k);$

$p_{k+1} = r_{k+1} + \beta_k p_k;$

`end`

Test for CG method

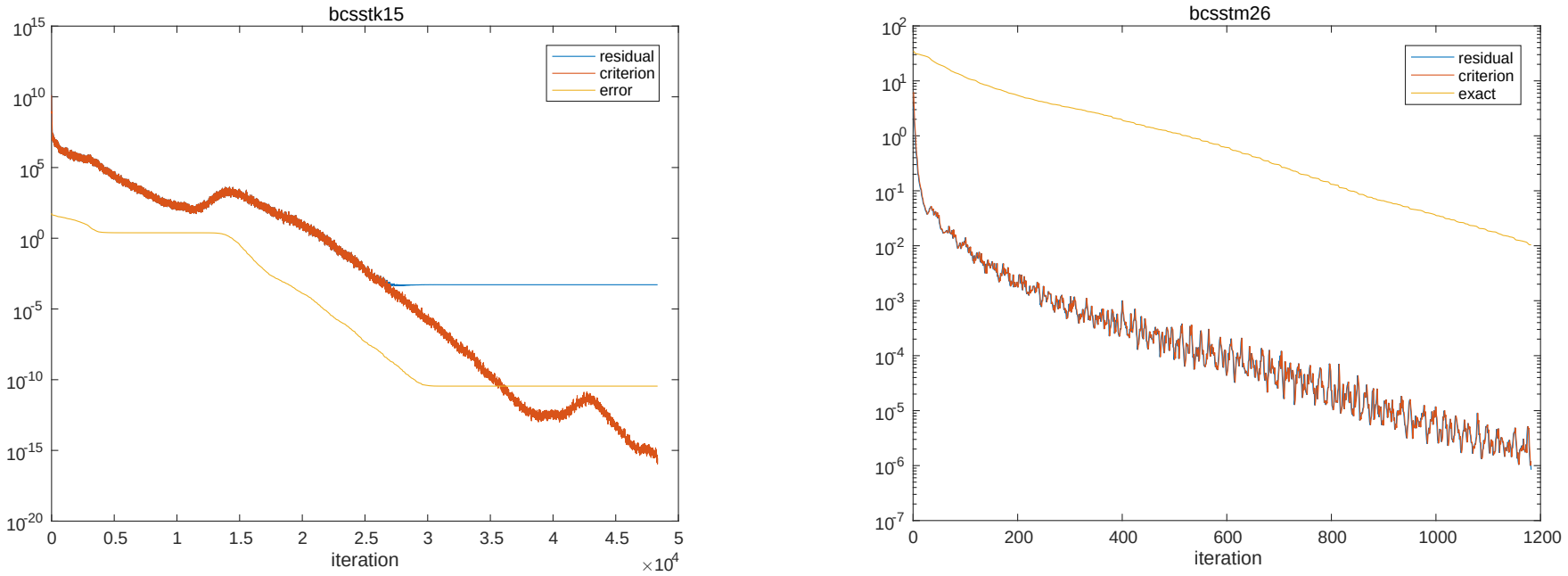


Figure 2: bcsstk15 and bcsstm26 from Matrix Market

Test for CG method

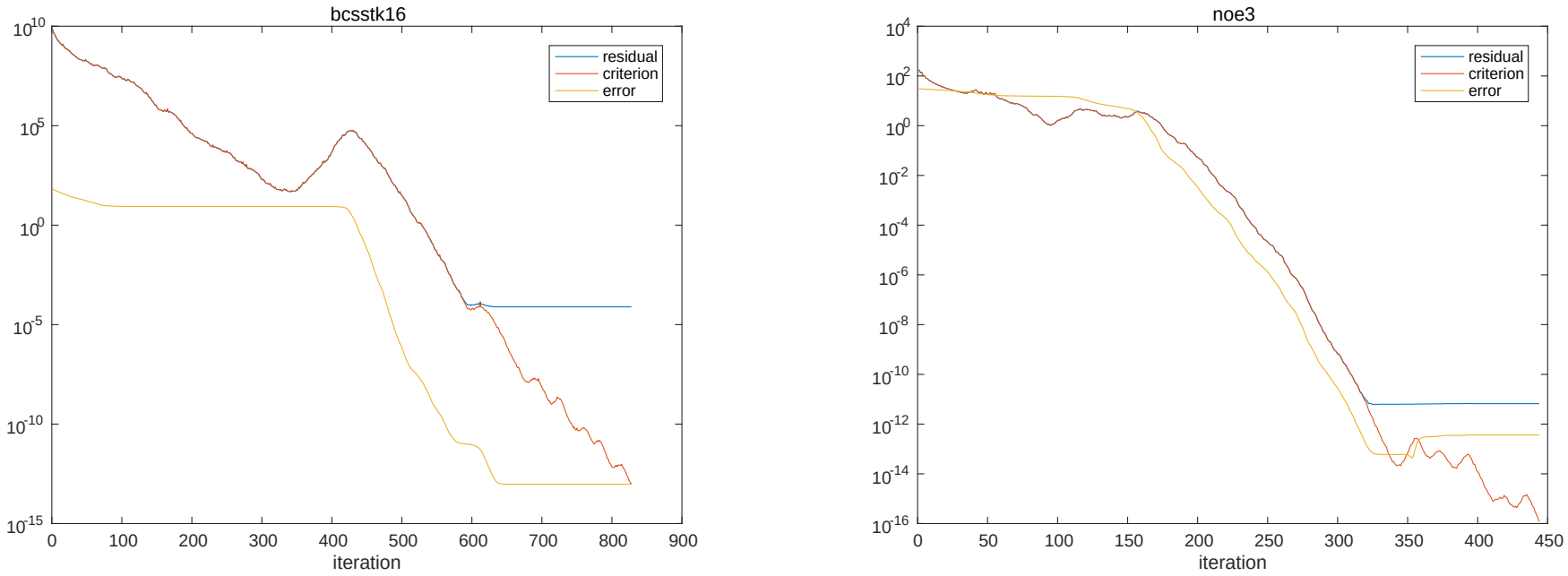
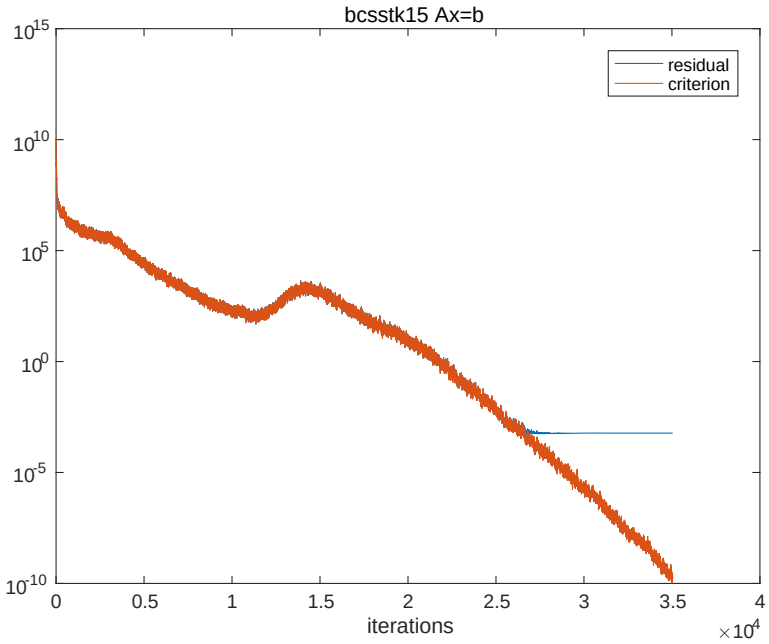


Figure 3: bcsstk16 and nos3 from Matrix Market

Test for CG method (bcsstk15)



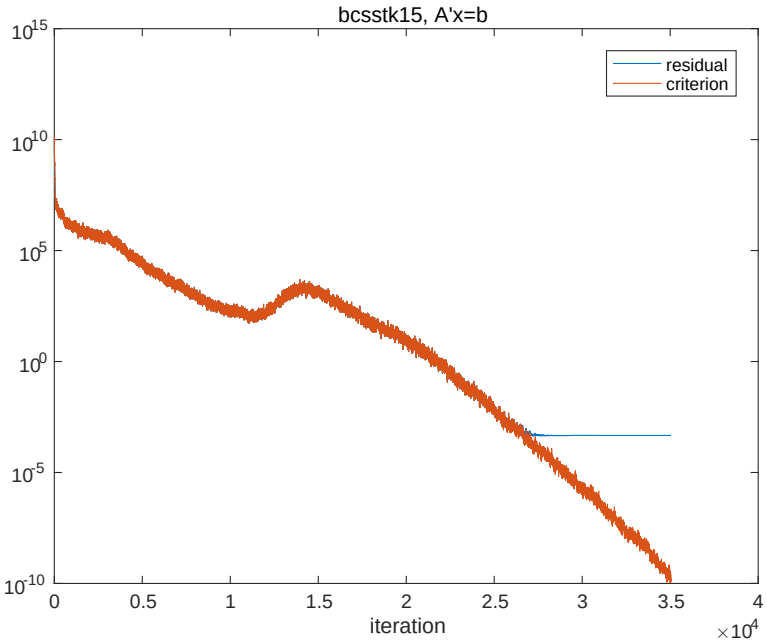


Figure 4: $Ax = b$ (left) and $A'x = b$ (right)

Test for CG method (bcsstm26)

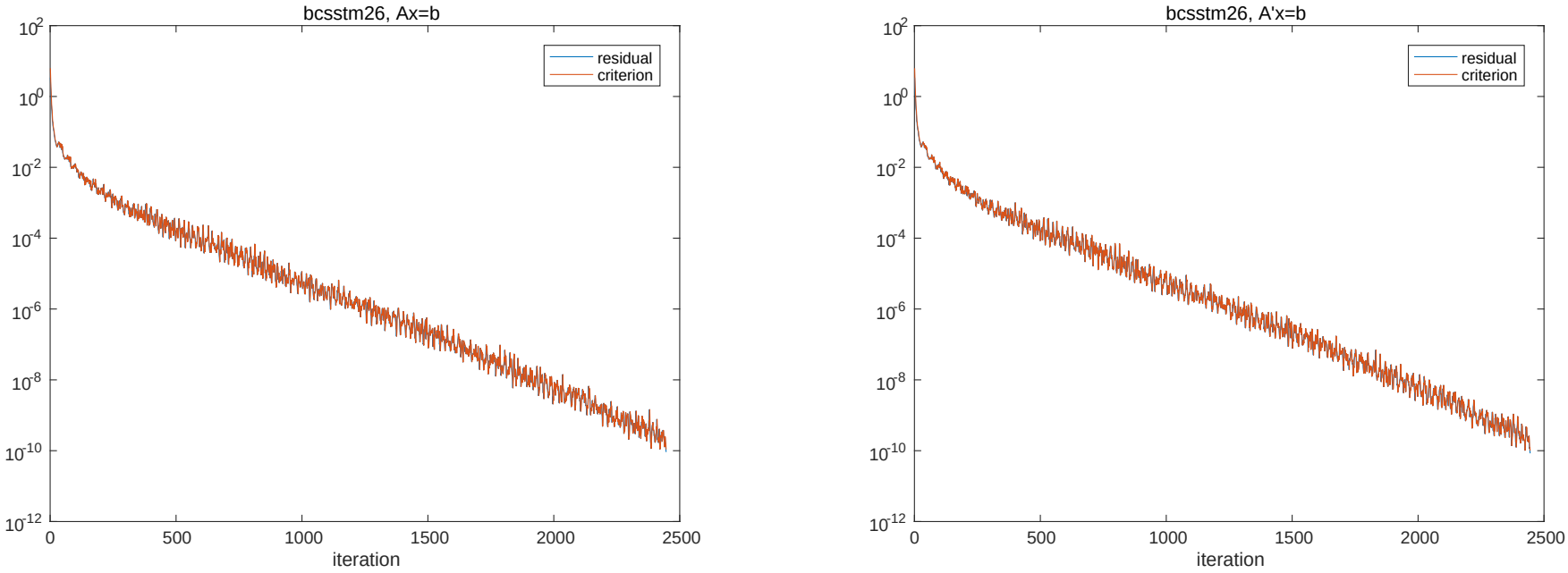


Figure 5: $Ax = b$ (left) and $A'x = b$ (right)

Conclusion

- We proposed a method which produces linear systems with the exact solution.
- We can develop alternative method using bit operations.
- However, the method introduced in this talk based on matrix operation, so that it is easy to implement.

Thank you very much for your attention!