Numerical experiments for verified numerical computations for linear systems

Tomoki Aoyama Katsuhisa Ozaki

Department of Mathematical Sciences, Shibaura Institute of Technology

The 41st JSST Annual International Conference on Simulation Technology

31 August 2022

Contents

1. Introduction [1, Chapter12]
2. Preparation
3. Strategy of iterative refinement with mixed-precision
4. The result for numerical experiments
5. Conclusion(Future Works)

[1] N. J. Higham, Accuracy and Stability of Numerical Algorithms, 2nd ed: SIAM, Philadelphia, 2002.

Introduction [1, Chapter12]

- Iterative refinement for linear systems has been researched over several decades (see [1, 2, 3]).
- We focus on linear systems,

$$Ax = b, \quad A \in \mathbb{R}^{n \times n}, \quad x \in \mathbb{R}^n, \quad b \in \mathbb{R}^n.$$

⇒ In particular, we set,

$$\frac{\max |x_i|}{\min |x_i|} \geq \mathsf{u}^{-1}, \quad \min |x_i| \neq 0,$$

where u is the unit roundoff. Then the research data couldn't be found.

[1] N. J. Higham, Accuracy and Stability of Numerical Algorithms, 2nd ed: SIAM, Philadelphia, 2002.

[2] R. D. Skeel, Iterative refinement implies numerical stability for Gaussian elimination: Math. Comp., **35** (1980), 817–832.

[3] E. Carson, N. J. Higham, Accelerating the Solution of Linear Systems by Iterative Refinement in Three Precisions: SIAM J. Sci. Comput., **40** A817–A847, (2018).

What is iterative refinement?

- Iterative refinement ••• One of the effective methods to improve the accuracy of the approximate solution.
- The procedure is as follows:

  ※ k : kth iteration

  ※ $\widehat{x}^{(k)}$: the approximate solution for kth iteration

  ※ $\widehat{x}^{(0)}$: the initial approximate solution
  (For example, use LU factorization for A.)

  1. Compute $\widehat{r} \approx b - A\widehat{x}^{(k)}$.
     (Let $\widehat{r}$ denote the computed vector.)
  2. Solve $Ay = \widehat{r}$.
     (Let the approximate solution be $\widehat{y}$.)
  3. Update $\widehat{x}^{(k+1)} \leftarrow \widehat{x}^{(k)} + \widehat{y}$.

The purpose of this presentation

- In this presentation, we will explain the following.
  - Monitoring the relative error using the iterative refinement with/without mixed-precision.
  - Checking the overestimation of the error bound by using verified numerical computations.
  - Checking the proper digits in high precision arithmetic for the computation of the residual.

Environment for numerical experiments

Table 1: Environment for numerical experiments

CPU	Intel(R) Core(TM)i5-11400
OS	Windows10 Education
MATLAB	R2022a
high-precision numerical calculation	Advanpix Multiprecision Computing Toolbox [4]

※ Advanpix Multiprecision Computing Toolbox ••• design ”mp” in MATLAB

[4] Multiprecision Computing Toolbox. Advanpix, http://www.advanpix.com.

Test data

- Test data

  ⇒ In this study, we used $A \in \mathbb{F}^{n\times n}$, $x \in \mathbb{F}^n$, $b \in \mathbb{F}^n$ $\quad (n = 3000)$.

  ※ $\mathbb{F}$: a set of binary floating-point numbers defined in IEEE 754-2019 [5]

  - $A = \text{gallery}('\text{randsvd}', n, \|A\|_2 \cdot \|A^{-1}\|_2, \text{mode}, n, n, 1)$

    ※ $'randsvd'$: Giving a random matrix that has the specified singular values.

    ※ $\|A\|_2 \cdot \|A^{-1}\|_2$: expected condition number

    ⇒ In the study, this is about $10^3, 10^6, 10^9, 10^{12}$.

    ※ mode = 3 : Assigning a random matrix with geometrically distributed singular values.

  - $x = \text{randperm}((-1)^i \times \text{u}^{(1+\text{rem}(i,4))}) \quad (1 \le i \le n)$

    ⇒ Permuting the components randomly.

[5] ANSI/IEEE Std 754-2019, IEEE Standard for Floating-Point Arithmetic, IEEE, 2019.

- We use Miyajima-Ogita-Oishi's method [6], then extend $A \in \mathbb{F}^{n\times n}$, $x \in \mathbb{F}^{n}$, $b \in \mathbb{F}^{n}$ as below.
  ※ $\alpha \in \{0, \mathbb{N}\}$

$$A'x' = b', \quad A' \in \mathbb{F}^{(n+\alpha)\times(n+\alpha)}, \quad x' \in \mathbb{F}^{(n+\alpha)}, \quad b' \in \mathbb{F}^{(n+\alpha)}$$

- A', x' are generated as below.

$$A' = \begin{pmatrix} A & \bullet \\ \boldsymbol{O} & I \end{pmatrix}, \quad x' = \begin{pmatrix} x \\ 1 \\ \vdots \\ 1 \end{pmatrix}$$

※ I : the α identity matrix

※ $\boldsymbol{O}$: the zero matrix

[6] S. Miyajima, T. Ogita, S. Oishi, A method of generating linear systems with an arbitrarily ill-conditioned matrix and an arbitrary solution: *Proc. 2005 International Symposium on Nonlinear Theory and its Applications (NOLTA 2005)*, (2005), 741–744.

The evaluation for the error bounds on the relative errors

- We use the following theorem [7] to compute the error bounds on the relative errors.
- And divide (1) by $|x|_i \neq 0 \ (1 \leq i \leq n)$ as the relative errors.

Theorem (Yamamoto, 1984)

Set $A \in \mathbb{R}^{n\times n} \ (det(A) \neq 0)$, $R \approx \widehat{A}^{-1} \in \mathbb{R}^{n\times n}$, $b, \widehat{x} \in \mathbb{R}^n$, $G \approx I - RA$. *If* $||G||_\infty < 1$, *then* (1) *satisfies.*

$$|A^{-1}b - \widehat{x}| \leq |R(b - A\widehat{x})| + \frac{||R(b - A\widehat{x})||_\infty}{1 - ||G||_\infty}|G|e. \qquad (1)$$

※ $e \, : \, (1, 1, \cdots, 1)^T \in \mathbb{R}^n$

※ $|\cdot|$: componentwise absolute value for a matrix and a vector

[7] T. Yamamoto, Error bounds for approximate solutions of systems of equations: *Japan J. Appl. Math.*, **1**(1984), 157–171.

The feature of mp

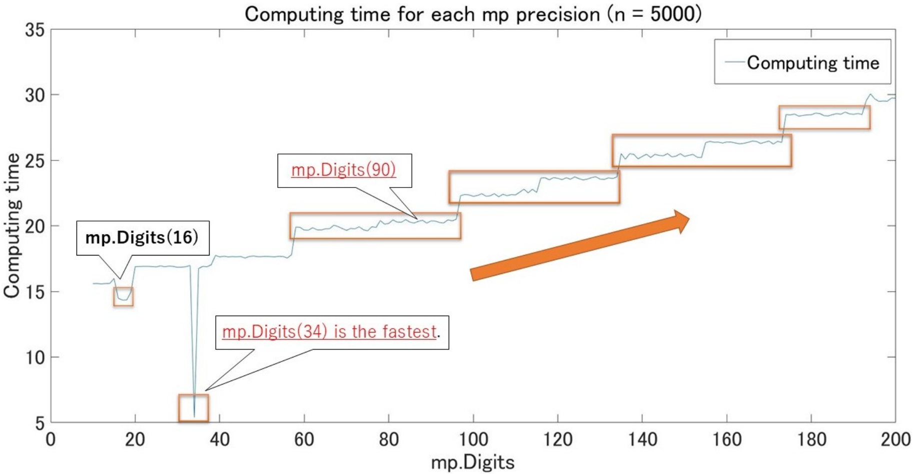

Figure 1: Computing time for each mp precision $(n = 5000)$

※ computing time ••• the sum of times for five runs "mp(b) − mp$(A) *$ mp(x)"

Three models

- Model 1

※ not using iterative refinement

  - $\widehat{x} = \mathsf{mp}(A) \backslash \mathsf{mp}(b)$.

  ※ mp.Digits($\underline{90}$)

  ※ straightforward use of high-precision

- Model 2

※ using iterative refinement

※ $k = 0, 1, \cdots$

Table 2: Computing the residual and updating the approximate solution

Using precision	Computing the residual	Updating the approximate solution
double	$r = b - A * \widehat{x}^{(k)}$	$\widehat{x}^{(k+1)} \leftarrow \widehat{x}^{(k)} + \widehat{y}$
quadruple	$r = \mathrm{mp}(b) - \mathrm{mp}(A) * \mathrm{mp}(\widehat{x}^{(k)})$	$\widehat{x}^{(k+1)} \leftarrow \mathrm{mp}(\widehat{x}^{(k)}) + \mathrm{mp}(\widehat{y})$
more high-precision		

※ double ••• no use of mp

※ quadruple(quad) ••• mp.Digits(34)

※ more high precision(more) ••• mp.Digits(90)

- Forward and backward substitutions are computed by "double".
- For each precision, mp.Digits is fixed.

- Model 3

※ using iterative refinement with adaptive precision

- The procedure follows.

1. Computing the residual and updating the approximate solution with quad(mp.Digits(34)).
2. If $\min($double(abs$((\widehat{x}^{(k)} - \widehat{x}^{(k+1)})./\widehat{x}^{(k)})$))$\leq 10^{-23}$, then change more high precision(mp.Digits(90)).

※ We set 10^{-23} heuristically.

- Note: For three models, if the following condition is satisfied, then iterative refinement is finished.
  (※ In this study, we set $\epsilon = 10^{-30}$ heuristically.)

$$\text{If } \min(\text{double}(\text{abs}((\widehat{x}^{(k)} - \widehat{x}^{(k+1)})./\widehat{x}^{(k)}))) \leq \epsilon \tag{2}$$

whereas

- the maximum of k (iteration number) is 20.

The result for numerical experiments

- 2-norm condition number

  ⇒ We introduce $\underline{\|A\|_2 \cdot \|A^{-1}\|_2 \approx 10^3}$.

- We use the following notations.
  - x_i : the ith component of the solution vector satisfied (3)
  - $\widehat{x}_i$: the ith component of the approximate solution vector satisfied (3)
  - Max(error) : the maximum of the relative errors

$$\max_{1 \leq i \leq n} \frac{|x - \widehat{x}|_i}{|x|_i}, \quad |x|_i \neq 0 \tag{3}$$

  - med(error) : the median of the relative errors
  - error_bound : the error bound on the relative errors (1)

$\|A\|_2 \cdot \|A^{-1}\|_2 = 10^3$

※ For double and quad, the condition (2) was not satisfied (all cases).

Table 3: The maximum and the median of the relative errors and the error bounds

Model		x	$\widehat{x}$	Max- (error)	med- (error)	error_ bound
Model 1		−1.5 e−64	−1.5 e−64	7.8 e−39	6.3 e−71	3.6 e−32
Model 2	double	−1.5 e−64	−1.3 e−29	8.2 e+34	7.5 e+02	8.2 e+34
	quad	−1.5 e−64	−7.4 e−48	4.9 e+16	6.2 e−16	9.3 e+23
	more	−1.5 e−64	−1.5 e−64	5.1 e−40	6.7 e−72	4.0 e−32
Model 3		−1.5 e−64	−1.5 e−64	6.3 e−40	7.5 e−72	3.8 e−32

※ The ideal value for the relative error is under $\mathcal{O}(10^{-16})$.

$(※\ \|A\|_2 \cdot \|A^{-1}\|_2 = 10^3)$

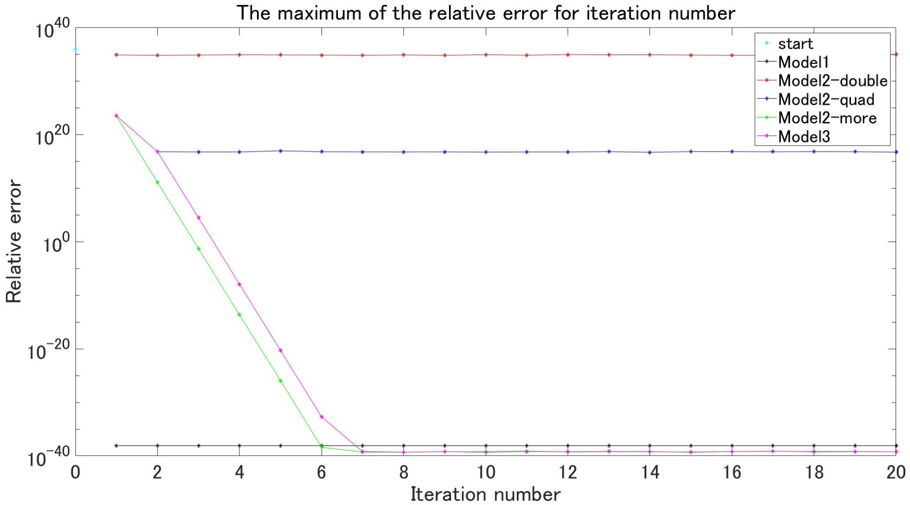

Figure 2: The maximum of the relative errors for each iteration number

Table 4: Comparing three models in terms of computing time ("b")

Model		Computing time
Model 1		2.1628 e+02
Model 2	double	1.1654 e−01
	quad	7.9263 e+00
	more	1.0425 e+01
Model 3		8.2769 e+00
acceleration rate		about 21 %

- Compare three models to check the computing time "$b \rightarrow B$".

※ B is generated by $(b, b, \cdots, b) \in \mathbb{F}^{(n+\alpha)\times j}$ $(j \in \mathbb{N})$.

Table 5: Comparing three models in terms of computing time

j		5	10	100
Model 1		2.1727 e+02	2.1756 e+02	2.4064 e+02
Model 2	double	2.3249 e−01	3.0443 e−01	6.1378 e−01
	quad	1.5176 e+01	2.3417 e+01	1.5495 e+02
	more	1.7075 e+01	2.5173 e+01	1.4695 e+02
Model 3		1.3673 e+01	2.0221 e+01	1.2151 e+02
acceleration rate		about 20 %		about 17 %

Conclusion(Future Works)

- Conclusion
  - ▶ Model 3 ••• the advantage of the computing time
  - ▶ In this experiment, by extending b to B, the more increasing the size of B is, the larger the time difference between Model 2 with more high precision and Model 3 is.

- Future Works
  - ▶ Consider stopping criterion with theoretically reason (now heuristic).

  $\Rightarrow$ For example, $10^{-23}, \epsilon$ (see (2)).
  - ▶ Develop more faster algorithm for this linear systems.
  - ▶ Analyzes the relative errors and the error bounds on the relative errors.

References I

[1] N. J. Higham, Accuracy and Stability of Numerical Algorithms, 2nd ed: *SIAM, Philadelphia*, 2002.

[2] R. D. Skeel, Iterative refinement implies numerical stability for Gaussian elimination: *Math. Comp.*, **35** (1980), 817–832.

[3] E. Carson, N. J. Higham, Accelerating the Solution of Linear Systems by Iterative Refinement in Three Precisions: *SIAM J. Sci. Comput.*, **40** (2018), A817–A847.

[4] Multiprecision Computing Toolbox. Advanpix, `http://www.advanpix.com.`

[5] ANSI/IEEE Std 754-2019, *IEEE Standard for Floating-Point Arithmetic*, IEEE, 2019.

References II

[6] S. Miyajima, T. Ogita, S. Oishi, A method of generating linear systems with an arbitrarily ill-conditioned matrix and an arbitrary solution: *Proc. 2005 International Symposium on Nonlinear Theory and its Applications (NOLTA 2005)*, (2005), 741–744.

[7] T. Yamamoto, Error bounds for approximate solutions of systems of equations: *Japan J. Appl. Math.*, **1**(1984), 157–171.

These are appendix slides.

A supplement of iterative refinement

- the procedure
  1. Compute $\widehat{r} \approx b - A\widehat{x}^{(k)}$.
     (Let $\widehat{r}$ denote the computed vector.)
  2. Solve $Ay = \widehat{r}$.
     (Let the approximate solution be $\widehat{y}$.)
  3. Update $\widehat{x}^{(k+1)} \leftarrow \widehat{x}^{(k)} + \widehat{y}$.
- (1), if $\widehat{x} = x$, then $\widehat{r} = 0$.
- (2), this equation is called "correction equation".
  Let define Δx be $|x - \widehat{x}|$. Then,

$$Ax = b \Leftrightarrow A(\widehat{x} + \Delta x) = b \Leftrightarrow A\Delta x = b - A\widehat{x} \approx \widehat{r},$$

  so, $\widehat{x}$ is improved by (3) ($\Delta x \approx \widehat{y}$).

The method of Miyajima-Ogita-Oishi [6]

$$A'x' = b', \quad A' \in \mathbb{F}^{(n+\alpha \times n+\alpha)}, \quad x' \in \mathbb{F}^{(n+\alpha)}, \quad b' \in \mathbb{F}^{(n+\alpha)}$$

⇒ where $\alpha \in \{0, \mathbb{N}\}$. These are generated on the next page. Note,

- $A = (a_{ij}) \in \mathbb{F}^{n \times n}$ $(1 \leq i \leq n,\ 1 \leq j \leq n)$,
- $b_l^{(k)}$ $(1 \leq l \leq n, 1 \leq k \leq \alpha)$,

⇒ the lth component of the kth right-hand side vector $b^{(k)}$ which is satisfied $Ax^* = \displaystyle\sum_{k=1}^{\alpha} b^{(k)}$.

[6] S. Miyajima, T. Ogita, S. Oishi, A method of generating linear systems with an arbitrarily ill-conditioned matrix and an arbitrary solution: *Proc. 2005 International Symposium on Nonlinear Theory and its Applications (NOLTA 2005)*, (2005), 741–744.

$$A' = \begin{pmatrix} a_{11} & a_{12} & \cdots & a_{n1} & -b_1^{(2)} & -b_1^{(3)} & \cdots & -b_1^{(\alpha)} \\ a_{21} & a_{22} & \cdots & a_{n2} & -b_2^{(2)} & -b_2^{(3)} & \cdots & -b_2^{(\alpha)} \\ \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \ddots & \vdots \\ a_{n1} & a_{n2} & \cdots & a_{nn} & -b_n^{(2)} & -b_n^{(3)} & \cdots & -b_n^{(\alpha)} \\ 0 & 0 & \cdots & 0 & 1 & 0 & \cdots & 0 \\ 0 & 0 & \cdots & 0 & 0 & 1 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & 0 & 0 & 0 & \cdots & 1 \end{pmatrix} \in \mathbb{F}^{(n+\alpha \times n+\alpha)},$$

$$x' = \begin{pmatrix} x_1^* \\ x_2^* \\ \vdots \\ x_n^* \\ 1 \\ \vdots \\ 1 \end{pmatrix} \in \mathbb{F}^{(n+\alpha)}, \quad b' = \begin{pmatrix} b_1^{(1)} \\ b_2^{(1)} \\ \vdots \\ b_n^{(1)} \\ 1 \\ \vdots \\ 1 \end{pmatrix} \in \mathbb{F}^{(n+\alpha)}.$$

About the feature of mp

- $A \in \mathbb{F}^{5000\times 5000}, x \in \mathbb{F}^{5000}, b \in \mathbb{F}^{5000}$ are generated as below.

$$A = \text{randn}(5000), \quad x = \text{mp}(\text{randn}(5000,1)), \quad b = \text{randn}(5000,1)$$

- The data in Figure 1 were obtained as follows:

Algorithm

```
index $= 1$;
for i $= 10 : 200$
    i
    mp.Digits(i);
    tic
    for j $= 1 : 5$
        C = mp(b) - mp(A) * mp(x);
    end for
    t(index) = toc;
    index = index + 1;
end for
```

stopping criterion (2)

- In this experiment, we give the solution vector...

however

- In practice, we don't know the solution vector regarding numerical experiments in many cases.

therefore

- We have to consider the stopping criterion used the approximate solution vector $\widehat{x}$ (not used the solution vector x)!!!

$\|A\|_2 \cdot \|A^{-1}\|_2 = 10^6$

Table 6: The maximum and the median of the relative errors and the error bounds

Model		x	$\widehat{x}$	Max- (error)	med- (error)	error_ bound
Model 1		-1.5 e-64	-1.5 e-64	1.8 e-36	1.5 e-68	1.8 e-29
Model 2	double	-1.5 e-64	-5 e-27	3.3 e$+37$	3.9 e$+05$	3.3 e$+37$
	quad	-1.5 e-64	5.6 e-45	3.7 e$+19$	3.6 e-13	4.3 e$+26$
	more	-1.5 e-64	-1.5 e-64	2.5 e-37	3.6 e-69	1.7 e-29
Model 3		-1.5 e-64	-1.5 e-64	3.2 e-37	3.3 e-69	1.8 e-29

(※ The ideal value for the relative error is under $\mathcal{O}(10^{-16})$).

(※ $\|A\|_2 \cdot \|A^{-1}\|_2 = 10^6$)

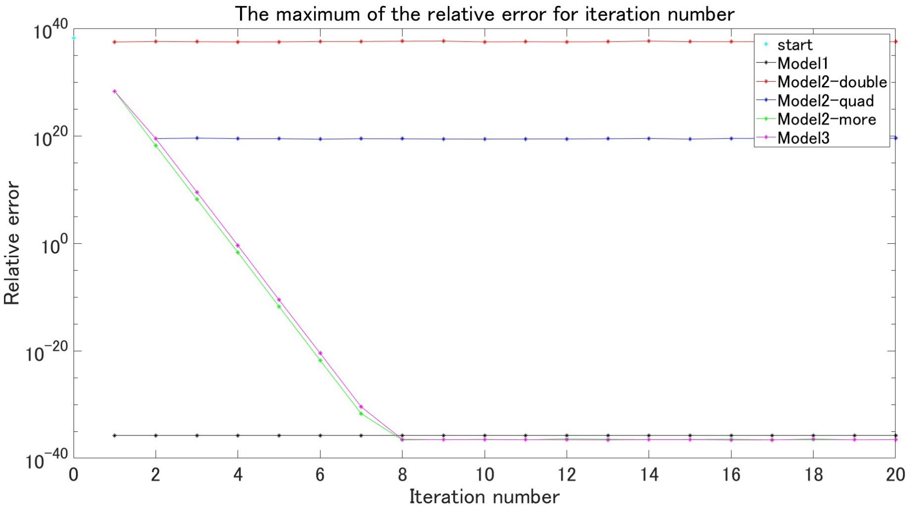

Figure 3: The maximum of the relative errors for each iteration number

Table 7: Comparing three models in terms of computing time ($''b''$)

Model		Computing time
Model 1		2.1652 e+02
Model 2	double	1.8426 e−01
	quad	7.7814 e+00
	more	1.1984 e+01
Model 3		9.7827 e+00
acceleration rate		about 18 %

- Compare three models to check the computing time "$b \to B$".

※ B is generated by $(b, b, \cdots, b) \in \mathbb{F}^{(n+\alpha)\times j}$ $(j \in \mathbb{N})$.

Table 8: Comparing three models in terms of computing time

j		5	10	100
Model 1		2.0353 e+02	2.1798 e+02	2.4129 e+02
Model 2	double	1.2895 e−01	1.3824 e−01	5.1497 e−01
	quad	1.4647 e+01	2.3554 e+01	1.5533 e+02
	more	1.9641 e+01	2.8822 e+01	1.7107 e+02
Model 3		1.6091 e+01	2.5019 e+01	1.5082 e+02
acceleration rate		about 18 %	about 13 %	about 12 %

$\|A\|_2 \cdot \|A^{-1}\|_2 = 10^9$

Table 9: The maximum and the median of the relative errors and the error bounds

Model		x	$\widehat{x}$	Max- (error)	med- (error)	error_ bound
Model 1		−1.5 e−64	−1.5 e−64	6.5 e−34	7.7 e−66	1.1 e−26
Model 2	double	−1.5 e−64	3.2 e−24	2.1 e+40	2.4 e+08	2.1 e+40
	quad	−1.5 e−64	−2.6 e−42	1.7 e+22	2.0 e−10	3.0 e+29
	more	−1.5 e−64	−1.5 e−64	1.7 e−34	1.9 e−66	9.7 e−27
Model 3		−1.5 e−64	−1.5 e−64	1.8 e−34	2.0 e−66	1.1 e−26

(※ The ideal value for the relative error is under $\mathcal{O}(10^{-16})$).

(※ $\|A\|_2 \cdot \|A^{-1}\|_2 = 10^9$)

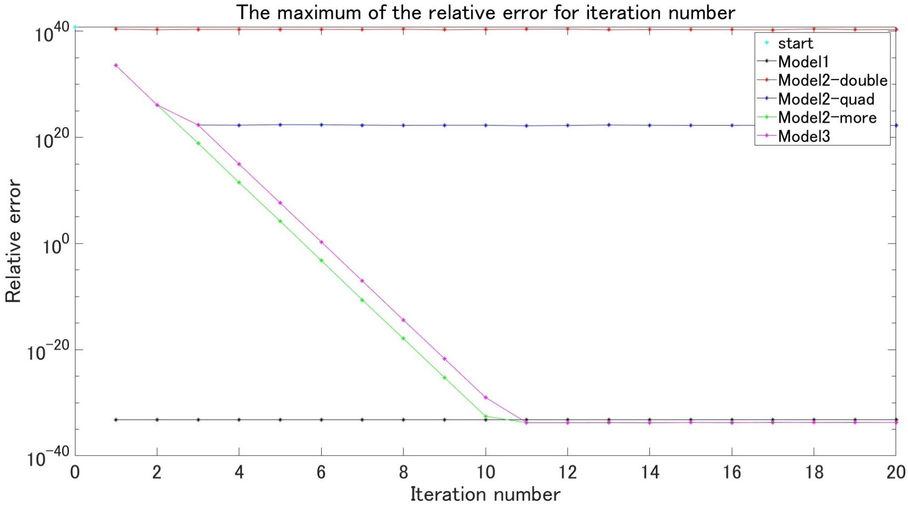

Figure 4: The maximum of the relative errors for each iteration number

Table 10: Comparing three models in terms of computing time ("b")

Model		Computing time
Model 1		2.1584 e+02
Model 2	double	8.0057 e−02
	quad	7.7424 e+00
	more	1.6529 e+01
Model 3		1.4669 e+01
acceleration rate		about 11 %

- Compare three models to check the computing time "$b \rightarrow B$".

※ B is generated by $(b, b, \cdots, b) \in \mathbb{F}^{(n+\alpha)\times j}$ $(j \in \mathbb{N})$.

Table 11: Comparing three models in terms of computing time

j		5	10	100
Model 1		2.1647 e+02	2.1764 e+02	2.4241 e+02
Model 2	double	1.2017 e−01	1.5420 e−01	6.6250 e−01
	quad	1.4919 e+01	2.3438 e+01	1.5526 e+02
	more	2.6555 e+01	3.9402 e+01	2.3358 e+02
Model 3		2.3733 e+01	3.5406 e+01	2.1154 e+02
acceleration rate		about 11 %	about 10 %	

$\|A\|_2 \cdot \|A^{-1}\|_2 = 10^{12}$

Table 12: The maximum and the median of the relative errors and the error bounds

Model		x	$\widehat{x}$	Max-(error)	med-(error)	error_ bound
Model 1		−1.5 e−64	−1.5 e−64	3.2 e−31	4.0 e−63	7.8 e−24
	double	−1.5 e−64	−2.4 e−21	1.6 e+43	1.9 e+11	1.7 e+43
Model 2	quad	−1.5 e−64	2.4 e−39	1.6 e+25	1.6 e−07	2.0 e+32
	more	−1.5 e−64	−1.5 e−64	1.4 e−31	1.5 e−63	8.0 e−24
Model 3		−1.5 e−64	−1.5 e−64	1.5 e−31	1.9 e−63	7.5 e−24

(※ The ideal value for the relative error is under $\mathcal{O}(10^{-16})$).

$(\text{※}\ \|A\|_2 \cdot \|A^{-1}\|_2 = 10^{12})$

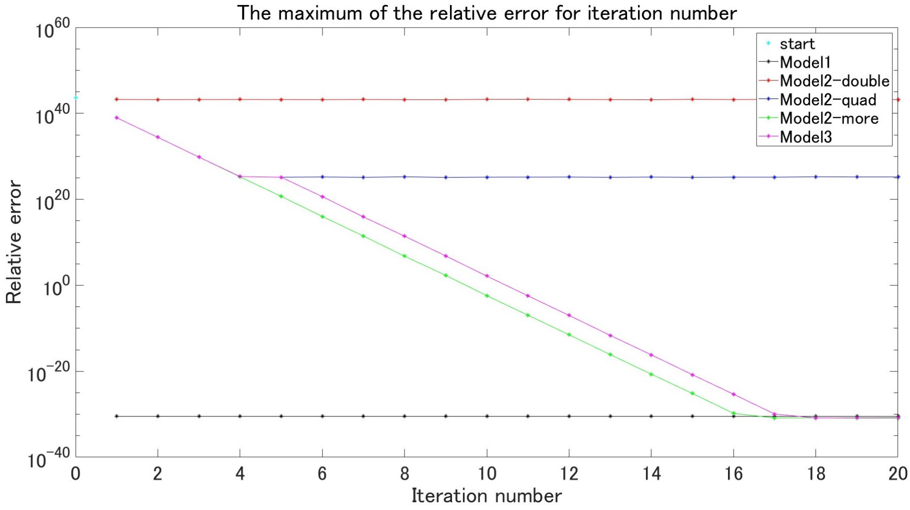

Figure 5: The maximum of the relative errors for each iteration number

Table 13: Comparing three models in terms of computing time ($''b''$)

Model		Computing time
Model 1		2.1679 e+02
Model 2	double	8.8011 e−02
	quad	7.8422 e+00
	more	2.7070 e+01
Model 3		2.2996 e+01
acceleration rate		about 15 %

- Compare three models to check the computing time "$b \to B$".

※ B is generated by $(b, b, \cdots, b) \in \mathbb{F}^{(n+\alpha)\times j}$ $(j \in \mathbb{N})$.

Table 14: Comparing three models in terms of computing time

j		5	10	100
Model 1		2.1601 e+02	2.1725 e+02	2.4325 e+02
Model 2	double	1.2903 e−01	1.9633 e−01	9.1305 e−01
	quad	1.5047 e+01	2.3394 e+01	1.5513 e+02
	more	4.3616 e+01	6.4391 e+01	3.8097 e+02
Model 3		3.7299 e+01	5.5521 e+01	3.3253 e+02
acceleration rate		about 14 %	about 14 %	about 13 %

These are preliminary slides.

Part1

- Question 1 : Why do you use Miyajima-Ogita-Oishi's method?

⇒ to avoid errors due to $A * x$

⇒ error ••• It's likely that $A * x \neq b$.

- Question 2 : Why do you use mp.Digits(90)?

⇒ According to Figure 1, from about mp.Digits(60) to about mp.Digits(100), the computation times are approximately the same.

⇒ Because around about mp.Digits(100) is subtle, so we set digits that are easy to understand just before that.

- Question 3 : What are all cases?

⇒ $\|A\|_2 \cdot \|A^{-1}\|_2 = 10^3, 10^6, 10^9, 10^{12}$

Part2

- Question 4 :
  If $\mathcal{O}(*)$ is double, it seems that error_bound is the same as Max(error)...

$\Rightarrow$ Strictly speaking, these are not the same.

$\Rightarrow$ In fact, error/error_bound $\approx 9.99 \cdots 95 * 10^{-9}$. The result is rounded by double, so it seems that...

- Question 5 :
  Why do you adapt the maximum of iteration number k is 20?

$\Rightarrow$ The answer is no specific meaning.

$\Rightarrow$ We adapt a relatively large number that we think is just right.

$\Rightarrow$ So, for example, 30, 50, and so on..., it's OK.

Part3

- Question 6 : Why do you think the acceleration rate is about 20%?

⇒ We use "if" statement at three times.

1. To change mp.Digits.

$$\text{If}\min(\text{double}(\text{abs}((\widehat{x}^{(k)}-\widehat{x}^{(k+1)})./\widehat{x}^{(k)}))) \leq 10^{-23}$$

2. To finish iterative refinement.

$$\text{If}\min(\text{double}(\text{abs}((\widehat{x}^{(k)}-\widehat{x}^{(k+1)})./\widehat{x}^{(k)}))) \leq \epsilon$$

3. To check kth iteration is the maximum 20?

Part4

- Question 7 : Why don't you try using scaling?

⇒ We don't think it yet. The reason is to focus on iterative refinement.

⇒ Digression : Iterative refinement eliminates the effects of poor scaling [2].

- Question 8 : Is there an application?

⇒ It's difficult to answer it.

⇒ The purpose of this study is thinking an effective algorithm in the case where the absolute values of the solution between components are varied and focusing on numerical calculations.

[2] R. D. Skeel, Iterative refinement implies numerical stability for Gaussian elimination: Math. Comp., **35** (1980), 817–832.