Iterative Refinement for Ill-Conditioned Linear Systems

Shin'ichi Oishi[1,4], Takeshi Ogita[2,1,4] and Siegfried M. Rump[3,1]

[1] *Department of Applied Mathematics*
Faculty of Science and Engineering, Waseda University
3–4–1 Okubo, Shinjuku-ku, Tokyo 169–8555, Japan
E-mail: oishi@waseda.jp

[2] *Department of Mathematics, College of Arts and Sciences*
Tokyo Woman's Christian University
2–6–1 Zempukuji, Suginami-ku, Tokyo 167–8585, Japan
E-mail: ogita@lab.twcu.ac.jp

[3] *Institute for Reliable Computing*
Hamburg University of Technology
Schwarzenbergstr. 95, 21071 Hamburg, Germany
E-mail: rump@tu-harburg.de

[4] *CREST, JST, Japan*

Received April 24, 2008
Revised December 22, 2008

This paper treats a linear equation

$$Av = b,$$

where $A \in \mathbb{F}^{n\times n}$ and $b \in \mathbb{F}^n$. Here, $\mathbb{F}$ is a set of floating point numbers. Let $\mathbf{u}$ be the unit round-off of the working precision and $\kappa(A) = \|A\|_\infty \|A^{-1}\|_\infty$ be the condition number of the problem. In this paper, ill-conditioned problems with

$$1 < \mathbf{u}\kappa(A) < \infty$$

are considered and an iterative refinement algorithm for the problems is proposed. In this paper, the forward and backward stability will be shown for this iterative refinement algorithm.

Key words: iterative refinement, verified numerical computation, ill-conditioned linear systems

1. Introduction

In this paper, we will consider the convergence of an iterative refinement for a linear equation

$$Av = b, \tag{1}$$

where $A \in \mathbb{F}^{n\times n}$ and $b \in \mathbb{F}^n$. Here, $\mathbb{F}$ is a set of floating point numbers. Nowadays, usually the double precision floating point number system defined by IEEE 754 standard is used for $\mathbb{F}$. For this case, the normalized floating point number has 64 bit length and its mantissa has 53 bit length. In this case, we call the double precision is the working precision and $2^{-53} \approx 10^{-16}$ is the unit round-off. Usually, numerical computations are done in working precision. Sometimes, higher precision calculations are used auxiliary. For example, a floating point number system with 128 bit length is used in some cases. In this case, we call a calculation by the floating point number system with 128 bit length is called done by an extended precision. In this paper, we treat a general case of $\mathbb{F}$. Namely, let $\mathbf{u}$ be the unit round-off of the working precision. Let further $\kappa(A) = \|A\|_\infty \|A^{-1}\|_\infty$ be the condition number of the problem. Here, $\|\cdot\|_\infty$ is the maximum norm defined by

$$\|v\|_\infty = \max_{1 \leqq i \leqq n} |v_i|, \quad \text{for } v = (v_1, v_2, \ldots, v_n)^{\mathrm{T}} \in \mathbb{R}^n \tag{2}$$

and

$$\|A\|_\infty = \max_{1 \leqq i \leqq n} \sum_{j=1}^{n} |A_{ij}|, \quad \text{for } A = (A_{ij}) \in \mathbb{R}^{n \times n}. \tag{3}$$

The superscript T denotes the transpose and $\mathbb{R}$ is the set of real numbers. For well posed problems, i.e., in case of $\mathbf{u}\kappa(A) < 1$, it has been shown [1]–[4] that the iterative refinement improves the forward and backward errors of computed solutions provided that the residuals are evaluated by extended precision, in which the unit round off $\bar{\mathbf{u}}$ is, for example, the order of $\mathbf{u}^2$, before rounding back to the working precision. Skeel [5] showed that iterative refinement with one iteration in working precision (not extended precision) is backward stable.

In this paper, we will treat ill-conditioned problems with

$$1 < \mathbf{u}\kappa(A) < \infty. \tag{4}$$

We can assume without loss of generality that for a certain positive integer k the following is satisfied:

$$\mathbf{u}^k \kappa(A) \leqq \beta < 1. \tag{5}$$

In [6], Rump has shown that for arbitrary ill-conditioned matrices A, we can have good approximate inverses $R_{1:k}$ satisfying

$$\|R_{1:k}A - I\|_\infty \leqq \alpha < 1. \tag{6}$$

Here, $R_{1:k}$ is obtained as

$$R_{1:k} = R_1 + R_2 + \cdots + R_k \tag{7}$$

with $R_i \in \mathbb{F}^{n \times n}$ and I is the n-dimensional unit matrix. In [7], we have partially clarified the mechanism of the convergence of Rump's method. Very recently, one of the authors (S.M. Rump) has further developed the convergence analysis of Rump's method [8].

Let $A, B, C \in \mathbb{F}^{n \times n}$. In floating point calculation, usually $AB - C$ cannot be calculated correctly because of the existence of rounding errors. In this paper, we assume that for any positive integer k satisfying $k < K$ with K being a certain sufficiently large positive integer, we can calculate $D_1, D_2, \ldots, D_k \in \mathbb{F}^{n \times n}$ satisfying

$$\left\| \sum_{i=1}^{k} D_i - (AB - C) \right\|_\infty \leqq cu^k \|AB - C\|_\infty. \tag{8}$$

Such algorithms have been proposed, for instance, by the authors [9]–[10]. We also denote

$$D_{1:k} = D_1 + D_2 + \cdots + D_k. \tag{9}$$

We further use a notation like

$$D_{1:k} = [AB - C]_k, \tag{10}$$

which means that $D_{1:k} = D_1 + D_2 + \cdots + D_k$ satisfies (8).

Similarly, for $A \in \mathbb{F}^{n\times n}$ and $b, c \in \mathbb{F}^n$ we assume that for any positive integer k satisfying $k < K$ with K being a certain sufficiently large positive integer, we can calculate $d_1, d_2, \ldots, d_k \in \mathbb{F}^n$ satisfying

$$\left\| \sum_{i=1}^{k} d_i - (Ab - c) \right\|_\infty \leqq cu^k \|Ab - c\|_\infty. \tag{11}$$

Such algorithms have also been proposed, for instance, by the authors [9]–[11]. We also introduce a notation

$$d_{1:k} = d_1 + d_2 + \cdots + d_k. \tag{12}$$

We further use a notation like

$$d_{1:k} = [Ab - c]_k, \tag{13}$$

which means that $d_{1:k} = d_1 + d_2 + \cdots + d_k$ satisfies (11). It should be noted that $[AB - C]_1$ coincides with $[AB - C]_k$ with $k = 1$. Similarly, $[Ab - c]_1$ coincides with $[Ab - c]_k$ with $k = 1$.

Now we propose the following iterative refinement algorithm:

$$v' = [v - R_{1:k}[Av - b]_k]_1. \tag{14}$$

Put $r_k = [Av - b]_k$ and let $\Phi(v) = [v - R_{1:k} r_k]_1$. Then, we can write

$$v' = \Phi(v). \tag{15}$$

The following holds:

$$v' = v - R_{1:k}[(Av - b) + e_r] + e_m, \tag{16}$$

where $e_r = r_k - (Av - b)$ and $e_m \in \mathbb{R}^n$ satisfying

$$\|e_r\|_\infty \leqq c\mathbf{u}^k \|Av - b\|_\infty \tag{17}$$

and

$$\|e_m\|_\infty \leqq c\mathbf{u} \|v - R_{1:k} r_k\|_\infty. \tag{18}$$

In this paper, we will show the forward and backward stability of the iterative algorithm (14). Furthermore, numerical examples are also given for illustrating the forward and backward stability of the iterative refinement algorithm (14). The forward stability of the algorithm guarantees that approximate solutions generated by the algorithm converge, while the backward stability means the stability of the algorithm against the rounding errors.

2. Convergence theorem: forward stability

Let us consider

$$Av = b, \tag{19}$$

where $A \in \mathbb{F}^{n \times n}$ and $b \in \mathbb{F}^n$. Let

$$1 < \mathbf{u}\kappa(A) < \infty. \tag{20}$$

We assume that we have a good approximate inverses $R_{1:k}$ satisfying (6). Here, $R_{1:k}$ is defined as

$$R_{1:k} = R_1 + R_2 + \cdots + R_k \tag{21}$$

with $R_i \in \mathbb{F}^{n \times n}$. As mentioned in the previous section in [6], Rump has proposed a method of calculating such approximate inverses and in [7], we have partially clarified the mechanism of the convergence of Rump's method. Further, we assume also that the following is satisfied:

$$\mathbf{u}^k\kappa(A) \leqq \beta < 1. \tag{22}$$

We propose the following iterative refinement algorithm:

$$v_n = \Phi(v_{n-1}), \quad \Phi(v) = [v - R_{1:k}r_k]_1, \quad r_k = [Av - b]_k \quad (n = 1, 2, \dots) \tag{23}$$

with any starting vector $v_0 \in \mathbb{F}^n$. The aim of this section is to show the following theorem:

Theorem 1. *Let v_n be generated from* (23) *with any starting vector $v_0 \in \mathbb{F}^n$. We assume that assumptions* (6) *and* (22) *hold. Let $v^* = A^{-1}b$. If*

$$\gamma = (\alpha + c\beta + c\alpha\beta)(1 + c\mathbf{u}) < 1, \tag{24}$$

the relative forward error $\|v_n - v^*\|_\infty / \|v^*\|_\infty$ reduces until

$$\frac{\|v_n - v^*\|_\infty}{\|v^*\|_\infty} \approx \mathbf{u} + \frac{c\mathbf{u}}{1 - \gamma}. \tag{25}$$

Here, for real numbers a and b, $a \approx b$ means that a is approximately equal to b.

This implies the forward stability of the iterative refinement algorithm (23).

Proof. Let $v \in \mathbb{F}^n$ and $v' = \Phi(v)$. The assumption (6) implies the existence of A^{-1}. Thus, from (23), we have for $v^* = A^{-1}b$

$$\begin{aligned}\|v' - v^*\|_\infty &= \|v - v^* - R_{1:k}r_k + e_m\|_\infty \\ &\leqq \|v - v^* - R_{1:k}r_k\|_\infty + \|e_m\|_\infty. \end{aligned} \tag{26}$$

We first note that

$$\begin{aligned}\|v - v^* - R_{1:k}r_k\|_\infty &= \|v - v^* - R_{1:k}[(Av-b) + e_r]\|_\infty \\ &= \|(I - R_{1:k}A)(v - v^*) - R_{1:k}e_r\|_\infty \\ &\leqq \|I - R_{1:k}A\|_\infty \|v - v^*\|_\infty + \|R_{1:k}\|_\infty \|e_r\|_\infty \\ &\leqq \alpha\|v - v^*\|_\infty + \|R_{1:k}\|_\infty \|e_r\|_\infty. \end{aligned} \tag{27}$$

On the other hand, from (17) and (22), we can estimate $\|R_{1:k}\|_\infty \|e_r\|_\infty$ as

$$\begin{aligned}\|R_{1:k}\|_\infty \|e_r\|_\infty &\leqq c\mathbf{u}^k \|R_{1:k}\|_\infty \|Av - b\|_\infty \\ &\leqq c\mathbf{u}^k \|R_{1:k}\|_\infty \|A\|_\infty \|v - v^*\|_\infty \\ &\leqq c'\mathbf{u}^k \kappa(A) \|v - v^*\|_\infty \\ &\leqq c'\beta\|v - v^*\|_\infty. \end{aligned} \tag{28}$$

Here, $c' = c\|R_{1:k}\|_\infty \|A\|_\infty / \kappa(A)$. Further, from

$$\begin{aligned}\|R_{1:k}\|_\infty &\leqq \|A^{-1}\|_\infty + \|A^{-1} - R_{1:k}\|_\infty \\ &= \|A^{-1}\|_\infty + \|(I - R_{1:k}A)A^{-1}\|_\infty \\ &\leqq \|A^{-1}\|_\infty (1 + \|I - R_{1:k}A\|_\infty) \\ &\leqq (1 + \alpha)\|A^{-1}\|_\infty, \end{aligned} \tag{29}$$

it turns out that

$$c' \leqq c(1 + \alpha). \tag{30}$$

Thus, from (27), (28) and (30), it follows that

$$\|v - v^* - R_{1:k}r_k\|_\infty \leqq (\alpha + c\beta + c\alpha\beta)\|v - v^*\|_\infty. \tag{31}$$

Moreover, from (31) and (18), we have

$$\begin{aligned}\|e_m\|_\infty &\leqq c\mathbf{u}\|v - R_{1:k}r_k\|_\infty \\ &\leqq c\mathbf{u}(\|v - v^* - R_{1:k}r_k\|_\infty + \|v^*\|_\infty) \\ &\leqq c\mathbf{u}((\alpha + c\beta + c\alpha\beta)\|v - v^*\|_\infty + \|v^*\|_\infty). \end{aligned} \tag{32}$$

Therefore, from (26), (31) and (32), we have finally

$$\begin{aligned}\|v' - v^*\|_\infty &\leqq \|v - v^* - R_{1:k}r_k\|_\infty + \|e_m\|_\infty \\ &\leqq (\alpha + c\beta + c\alpha\beta)(1 + c\mathbf{u})\|v - v^*\|_\infty + c\mathbf{u}\|v^*\|_\infty. \end{aligned} \tag{33}$$

Summing up the above mentioned arguments, for

$$v_n = \Phi(v_{n-1}) \quad (n = 1, 2, \dots), \tag{34}$$

with some starting vector $v_0 \in \mathbb{F}^n$, we have

$$\begin{aligned} \|v_n - v^*\|_\infty &\leqq \gamma\|v_{n-1} - v^*\|_\infty + c\mathbf{u}\|v^*\|_\infty \\ &\leqq \gamma^2\|v_{n-2} - v^*\|_\infty + c\mathbf{u}(1+\gamma)\|v^*\|_\infty \\ &\quad \cdots \\ &\leqq \gamma^n\|v_0 - v^*\|_\infty + \frac{c\mathbf{u}}{1-\gamma}\|v^*\|_\infty \end{aligned} \tag{35}$$

provided that

$$\gamma = (\alpha + c\beta + c\alpha\beta)(1 + c\mathbf{u}) < 1. \tag{36}$$

This implies that if $\gamma < 1$, the relative forward error reduces until

$$\frac{\|v_n - v^*\|_\infty}{\|v^*\|_\infty} \approx \mathbf{u} + \frac{c\mathbf{u}}{1-\gamma}. \tag{37}$$

□

3. Backward stability

In this section, we will show the backward stability of the iterative refinement algorithm (23).

A normwise backward error of an approximation v is defined by

$$\eta(v) = \min\{\varepsilon\colon (A + \Delta A)v = b + \Delta b,\ \|\Delta A\|_\infty \leqq \varepsilon\|A\|_\infty,\ \|\Delta b\|_\infty \leqq \varepsilon\|b\|_\infty\}. \tag{38}$$

It is known [14] that

$$\eta(v) = \frac{\|r\|_\infty}{\|A\|_\infty\|v\|_\infty + \|b\|_\infty}. \tag{39}$$

Here, $r = Av - b$.

The next theorem shows the backward stability of the iterative refinement algorithm (23):

Theorem 2. *Let v_n be generated from* (23) *with any starting vector $v_0 \in \mathbb{F}^n$. We assume the assumptions* (6) *and* (22)*. If*

$$\gamma = (\alpha + c\beta + c\alpha\beta)(1 + c\mathbf{u}) < 1, \tag{40}$$

the backward error $\eta(v_n)$ reduces until

$$\eta(v_n) \lessapprox c_2\mathbf{u}, \tag{41}$$

where c_2 is a certain constant. Here, for real numbers a and b, $a \lessapprox b$ means that a is approximately equal to b or a is less than b.

This implies the backward stability of the iterative refinement algorithm (23).

Proof. Let $v' = \Phi(v)$, i.e.,

$$v' = [v - R_{1:k}[Av - b]_k]_1. \tag{42}$$

Put $r = Av - b$. We have

$$v' = v - R_{1:k}(r + e_r) + e_m, \tag{43}$$

where $e_r = r_k - r$ and $e_m \in \mathbb{R}^n$ satisfy

$$\|e_r\|_\infty \leqq c\mathbf{u}^k \|Av - b\|_\infty, \quad \|e_m\|_\infty \leqq c\mathbf{u}\|v - R_{1:k}r_k\|_\infty. \tag{44}$$

Put $r' = Av' - b$. Then, it follows from (43) that

$$\begin{aligned} r' &= r + A(v' - v) \\ &= r + A(-R_{1:k}(r + e_r) + e_m) \\ &= r + A[(-A^{-1} + A^{-1} - R_{1:k})(r + e_r) + e_m] \\ &= -e_r + A[(I - R_{1:k}A)A^{-1}(r + e_r) + e_m] \\ &= -e_r + A[(I - R_{1:k}A)(v - v^* + A^{-1}e_r) + e_m]. \end{aligned} \tag{45}$$

Here, $v^* = A^{-1}b$. Thus, from (45) together with (17), (6) and (22), we have

$$\begin{aligned} \|r'\|_\infty &\leqq \|e_r\|_\infty + \|A\|_\infty\|I - R_{1:k}A\|_\infty\|v - v^*\|_\infty \\ &\quad + \|A\|_\infty\|I - R_{1:k}A\|_\infty\|A^{-1}\|_\infty\|e_r\|_\infty + \|A\|_\infty\|e_m\|_\infty \\ &\leqq c\mathbf{u}^k\|r\|_\infty + \alpha\|A\|_\infty\|v - v^*\|_\infty + c\alpha\beta\|r\|_\infty + \|A\|_\infty\|e_m\|_\infty \\ &= c(\alpha\beta + \mathbf{u}^k)\|r\|_\infty + \alpha\|A\|_\infty\|v - v^*\|_\infty + \|A\|_\infty\|e_m\|_\infty. \end{aligned} \tag{46}$$

We now recall (18):

$$\|e_m\|_\infty \leqq c\mathbf{u}\|v - R_{1:k}r_k\|_\infty. \tag{47}$$

It is note here that

$$\begin{aligned} v - R_{1:k}r_k &= v - R_{1:k}r - R_{1:k}e_r \\ &= v - [A^{-1} - (A^{-1} - R_{1:k})]r - R_{1:k}e_r \\ &= v - A^{-1}r + (A^{-1} - R_{1:k})r - R_{1:k}e_r \\ &= v^* + (I - R_{1:k}A)A^{-1}r - R_{1:k}e_r. \end{aligned} \tag{48}$$

Substituting (48) into (47) and noticing (29), it follows

$$\begin{aligned} \|e_m\|_\infty &\leqq c\mathbf{u}\|v^* + (I - R_{1:k}A)(v - v^*) - R_{1:k}e_r\|_\infty \\ &\leqq c\mathbf{u}(\|v^*\|_\infty + \alpha\|v - v^*\|_\infty + (1 + \alpha)\|A^{-1}\|_\infty\|e_r\|_\infty). \end{aligned} \tag{49}$$

Thus, from (46) and (49), we have

$$\begin{aligned}\|r'\|_\infty \leqq\ & [c(\alpha\beta + \mathbf{u}^k) + c^2(1+\alpha)\beta\mathbf{u}]\|r\|_\infty \\ & + \alpha(1 + c\mathbf{u})\|A\|_\infty\|v - v^*\|_\infty + c\mathbf{u}\|A\|_\infty\|v^*\|_\infty. \qquad (50)\end{aligned}$$

From the condition $\gamma < 1$, if v is an approximate solution sufficiently refined by the iterative refinement (23), then Theorem 1 implies that $\|v - v^*\|_\infty = c''\mathbf{u}\|v^*\|_\infty$ and $v' \approx v$, i.e., $r' \approx r$. Here, $c'' = 1 + c/(1-\gamma)$. Thus, in this case, we have

$$\|r\|_\infty \lessapprox \omega\|r\|_\infty + c_1\mathbf{u}\|A\|_\infty\|v^*\|_\infty, \qquad (51)$$

where $c_1 = c''\alpha(1 + c\mathbf{u}) + c$ and $\omega = c(\alpha\beta + \mathbf{u}^k) + c^2(1 + \alpha)\beta\mathbf{u}$.

We note here that from (22) it is seen that

$$\gamma - \omega = \alpha(1 + c\mathbf{u}) + \mathbf{u}^k(c\kappa(A) - 1) > 0, \qquad (52)$$

which implies

$$\omega < 1. \qquad (53)$$

Thus, we have

$$\|r\|_\infty \lessapprox \frac{c_1\mathbf{u}\|A\|_\infty\|v^*\|_\infty}{1-\omega} = c_2\mathbf{u}(\|A\|_\infty\|v^*\|_\infty + \|b\|_\infty) \qquad (54)$$

with c_2 being a suitable constant, which gives a small backward error

$$\eta(v) \lessapprox c_2\mathbf{u} \qquad (55)$$

provided that v is an approximate solution sufficiently refined by the iterative refinement (23). □

4. Numerical examples illustrating forward and backward stability

In this section, we will present numerical examples illustrating the forward and the backward stability of the iterative refinement algorithm (23).

We have used the IEEE 754 double precision floating point number system in these numerical calculations. Thus, in the following calculations, the unit round-off $\mathbf{u}$ is given as

$$1.11 \times 10^{-16} < \mathbf{u} = 2^{-53} < 1.12 \times 10^{-16}. \qquad (56)$$

4.1. Hilbert matrix

Let H be the $n \times n$ Hilbert matrix. Let further $A = sH$. Here, s is the minimum common multiplier of $1, 2, 3, \ldots, n-1$. Furthermore,

$$b = Az, \qquad (57)$$

where, $z = (1, 1, \dots, 1)^{\mathrm{T}} \in \mathbb{F}^n$. We have solved $Ax = b$ for $n = 20$. In this example, $1.92 \times 10^{16} < \|A\|_\infty < 1.93 \times 10^{16}$, $1.92 \times 10^{16} < \|b\|_\infty < 1.93 \times 10^{16}$ and $2.44 \times 10^{28} < \kappa(A) < 2.45 \times 10^{28}$.

In this case, a good approximate inverse can be constructed with $k = 2$ such that

$$\|R_{1:2}A - I\|_\infty < \alpha = 4.16 \times 10^{-4}, \tag{58}$$

where

$$R_{1:2} = R_1 + R_2 \tag{59}$$

with suitable $R_1, R_2 \in \mathbb{F}$. The iterative refinement algorithm (23) converges with 3 iterations. We finally have an approximate solution with the relative maximum error about 1.92×10^{-16}. Furthermore, it is seen that

$$\beta = \mathbf{u}^2\kappa(A) < (1.2 \times 10^{-16})^2 \times 2.45 \times 10^{28} < 3.08 \times 10^{-4}. \tag{60}$$

Table 1 shows the relative errors

$$\frac{\|v^* - v_i\|_\infty}{\|v^*\|_\infty} \tag{61}$$

and the backward errors $\eta(v_i)$ of approximate solutions obtained by the iterative refinement calculations (23). These calculations are done by MATLAB on Intel core 2 duo CPU. The initial approximation v_0 is given by

$$v_0 = [R_{1:2}b]_1 = [(R_1 + R_2)b]_1. \tag{62}$$

Since $R_{1:2}$ is a good approximate inverse satisfying (6), the first a few digits of v_0 already coincide with those for v^*. This is consistent with the result shown in Table 1, which shows that the first a few digits of $\|v_i\|_\infty$ are the same. According to the progress of iterative refinement, the forward errors

$$\frac{\|v_n - v^*\|_\infty}{\|v^*\|_\infty} \tag{63}$$

decrease up to the unit round-off $\mathbf{u}$.

Table 1. Hilbert matrix ($n = 20$)

i	$\|v^* - v_i\|_\infty/\|v^*\|_\infty$	$\|v_i\|_\infty$	$\|r_i\|_\infty$	$\eta(v_i)$
0	3.50×10^{-4}	1.16	1.89×10^{11}	4.55×10^{-6}
1	4.03×10^{-9}	1.16	1.71×10^{6}	4.12×10^{-11}
2	5.10×10^{-14}	1.16	2.09×10	5.04×10^{-16}
3	1.91×10^{-16}	1.16	7.37×10^{-2}	1.77×10^{-18}
4	1.91×10^{-16}	1.16	7.37×10^{-2}	1.77×10^{-18}

On the other hand, the backward errors are proportional to the residuals $\|r_i\|_\infty$. Since, the residuals are decreasing, the backward errors also decrease up to around the unit round-off $\mathbf{u}$.

4.2. Rump's matrix ($n = 100$)

Let A be $n \times n$ matrix with an anticipated condition number being 10^{100} generated by Rump's algorithm [13]. We choose $n = 100$ and $b = (1, 1, \dots, 1)^{\mathrm{T}} \in \mathbb{F}^n$. In this example, $1.04 \times 10^{16} < \|A\|_\infty < 1.05 \times 10^{16}$, $\|b\|_\infty = 1$ and $1.74 \times 10^{107} < \kappa(A) < 1.75 \times 10^{107}$.

In this case, a good approximate inverse can be constructed with $k = 8$ such that

$$\|R_{1:8}A - I\|_\infty < \alpha = 1.86 \times 10^{-4}, \tag{64}$$

where

$$R_{1:8} = R_1 + R_2 + \cdots + R_8 \tag{65}$$

with suitable $R_1, R_2, \dots, R_8 \in \mathbb{F}$. The iterative refinement algorithm (23) converges with 3 iterations.

Moreover, it is seen that

$$\beta = \mathbf{u}^8 \kappa(A) < (1.12 \times 10^{-16})^8 \times 1.75 \times 10^{107} < 4.34 \times 10^{-21}. \tag{66}$$

Table 2 shows the relative errors and the backward errors of approximate solutions obtained by the iterative refinement calculations (23). The calculations are done by the same computational environment as that for the previous example.

Table 2. Rump's matrix ($n = 100$)

i	$\|v^* - v_i\|_\infty / \|v^*\|_\infty$	$\|v_i\|_\infty$	$\|r_i\|_\infty$	$\eta(v_i)$
0	7.51×10^{-6}	4.44×10^{91}	1.85×10^{94}	3.98×10^{-14}
1	5.98×10^{-11}	4.44×10^{91}	2.61×10^{89}	5.61×10^{-19}
2	4.88×10^{-16}	4.44×10^{91}	1.15×10^{89}	2.46×10^{-19}
3	3.18×10^{-19}	4.44×10^{91}	3.06×10^{89}	6.58×10^{-19}
4	3.18×10^{-19}	4.44×10^{91}	3.06×10^{89}	6.58×10^{-19}

The initial approximation v_0 is given by

$$v_0 = [R_{1:8}b]_1. \tag{67}$$

Since $R_{1:8}$ is a good approximate inverse satisfying (6), the first a few digits of v_0 already coincide with those for v^*. This is consistent with the result shown in Table 2, which shows that the first a few digits of $\|v_i\|_\infty$ are the same. According to the progress of iterative refinement, the forward errors

$$\frac{\|v_n - v^*\|_\infty}{\|v^*\|_\infty} \tag{68}$$

decrease up to the unit round-off $\mathbf{u}$.

On the other hand, the backward errors are proportional to the residuals $\|r_i\|_\infty$. Since, the residuals are decreasing, the backward errors also decrease up to around the unit round-off $\mathbf{u}$.

4.3. Rump's matrix ($n = 300$)

Let A be $n \times n$ matrix with an anticipated condition number being 10^{50} generated by Rump's algorithm [13]. We choose $n = 300$ and $b = (1, 1, \dots, 1)^{\mathrm{T}} \in \mathbb{F}^n$. In this example, $3.10 \times 10^8 < \|A\|_\infty < 3.11 \times 10^8$, $\|b\|_\infty = 1$ and $6.28 \times 10^{59} < \kappa(A) < 6.29 \times 10^{59}$.

In this case, a good approximate inverse can be constructed with $k = 5$ such that

$$\|R_{1:5}A - I\|_\infty < \alpha = 1.16 \times 10^{-9}, \tag{69}$$

where

$$R_{1:5} = R_1 + R_2 + \cdots + R_5 \tag{70}$$

with suitable $R_1, R_2, \dots, R_5 \in \mathbb{F}$. The iterative refinement algorithm converges (23) with 1 iteration.

Moreover, it is seen that

$$\beta = \mathbf{u}^5\kappa(A) < (1.12 \times 10^{-16})^5 \times 6.29 \times 10^{59} < 1.11 \times 10^{-20}. \tag{71}$$

Table 3 shows the relative errors and the backward errors of approximate solutions obtained by the iterative refinement calculations (23). The calculations are done by the same computational environment as that for the previous example.

Table 3. Rump's matrix ($n = 300$)

i	$\|v^* - v_i\|_\infty / \|v^*\|_\infty$	$\|v_i\|_\infty$	$\|r_i\|_\infty$	$\eta(v_i)$
0	8.02×10^{-12}	4.89×10^{50}	2.15×10^{42}	1.42×10^{-17}
1	8.10×10^{-23}	4.89×10^{50}	6.17×10^{40}	4.07×10^{-19}
2	8.10×10^{-23}	4.89×10^{50}	6.17×10^{40}	4.07×10^{-19}

The initial approximation v_0 is given by

$$v_0 = [R_{1:5}b]_1. \tag{72}$$

Since $R_{1:5}$ is a good approximate inverse satisfying (6), the first a few digits of v_0 already coincide with those of v^*. This is consistent with the result shown in Table 3, which shows that the first a few digits of $\|v_i\|_\infty$ are the same. According to the progress of iterative refinement, the forward errors

$$\frac{\|v_n - v^*\|_\infty}{\|v^*\|_\infty} \tag{73}$$

decrease up to the unit round-off $\mathbf{u}$.

On the other hand, the backward errors are proportional to the residuals $\|r_i\|_\infty$. Since, the residuals are decreasing, the backward errors also decrease up to around the unit round-off $\mathbf{u}$.

Acknowledgement. The authors are supported by the Grant-in-Aid for Specially Promoted Research from the MEXT, Japan: "Establishment of Verified Numerical Computation" (No. 17002012).

References

[1] G.B. Moler, Iterative refinement in floating point. J. Assoc. Comput. Mach., **14** (1967), 316–321.

[2] N.J. Higham, Iterative refinement for linear systems and LAPACK. IMA J. Numer. Anal., **17** (1997), 495–509.

[3] M. Jankowsky and H. Woznlakowski, Iterative refinement implies numerical stability. BIT, **17** (1997), 303–311.

[4] F. Tisseur, Newton's method in floating point arithmetic and iterative refinement of generalized eigenvalue problems. SIAM J. Matrix Anal. Appl., **22** (2001), 1038–1057.

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

[6] S.M. Rump, Approximate inverses of almost singular matrices still contain useful information. Technical Report 90.1, Faculty of Information and Communication Sciences, Hamburg University of Technology, 1990.

[7] S. Oishi, K. Tanabe, T. Ogita and S.M. Rump, Convergence of Rump's method for inverting arbitrary ill-conditioned matrices. J. Comp. and Appl. Math., **205** (2007), 533–544.

[8] S.M. Rump, Inversion of extremely ill-conditioned matrices in floating point. Submitted for publication in JJIAM, March 15, 2008.

[9] T. Ogita, S.M. Rump and S. Oishi, Accurate sum and dot product. SIAM Journal on Scientific Computing, **26** (2005), 1955–1988.

[10] S.M. Rump, T. Ogita and S. Oishi, Accurate floating-point summation I: Faithful rounding. Accepted for publication in SIAM Journal on Scientific Computing. Preprint is available from `http://www.ti3.tu-harburg.de/publications/rump`.

[11] S.M. Rump, T. Ogita and S. Oishi, Accurate floating-point summation II: Sign, K-fold faithful and rounding to nearest. Accepted for publication in SIAM Journal on Scientific Computing. Preprint is available from `http://www.ti3.tu-harburg.de/publications/rump`.

[12] T. Ohta, T. Ogita, S.M. Rump and S. Oishi, A method of verified numerical computation for ill-conditioned linear system of equations. Journal of JSIAM, **15** (2005), 269–287, in Japanese.

[13] S.M. Rump, A class of arbitrarily ill-conditioned floating-point matrices. SIAM J. Matrix Anal. Appl., **12** (1991), 645–653.

[14] J.D. Rigal and J. Gaches, On the compativility of a given solution with the data of a linear equation. J. Assoc. Comput. Mach., **14** (1967), 543–548.