published in J. Comput. Appl. Math., 205(1):533–544, 2007

Convergence of Rump's Method for Inverting Arbitrarily Ill-Conditioned Matrices

Shin'ichi Oishi [a,b] Kunio Tanabe [a] Takeshi Ogita [b,a]
Siegfried M. Rump [c,a]

[a] *Faculty of Science and Engineering,
Waseda University, 3–4–1 Okubo, Tokyo 169–8555 Japan
{oishi, tanabe.kunio, ogita}@waseda.jp*

[b] *CREST, Japan Science and Technology Agency*

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

Abstract

In this paper, the problem of inverting regular matrices with arbitrarily large condition number is treated in double precision defined by IEEE 754 floating point standard. In about 1984, Rump derived a method for inverting arbitrarily ill-conditioned matrices. The method requires the possibility to calculate a dot product in higher precision. Rump's method is of theoretical interest. Rump made it clear that inverting an arbitrarily ill-conditioned matrix in single or double precision does not produce meaningless numbers, but contains a lot of information in it. Rump's method uses such inverses as preconditioners. Numerical experiments exhibit that Rump's method converges rapidly for various matrices with large condition numbers. Why Rump's method is so efficient for inverting arbitrarily ill-conditioned matrices is a little mysterious. Thus, to prove its convergency is an interesting problem in numerical error analysis. In this article, a convergence theorem is presented for a variant of Rump's method.

Key words: matrix inversion, ill-conditioned matrix, accurate dot product, precondition

1 Introduction

In this paper, we will treat the problem of inverting regular matrices $A \in \mathbb{F}^{n\times n}$ with arbitrarily large condition number. Here, $\mathbb{F}$ is the set of double precision floating point numbers defined by IEEE 754 standard [1]. We shall consider a method which only uses ordinary floating point arithmetic $\{+, -, *, /\}$ in working precision (*i.e.* IEEE 754's double precision) and a dot product with k-fold accuracy. Let $\|A\|_\infty$ denote a maximum matrix norm of A and $\kappa(A) := \|A\|_\infty \|A^{-1}\|_\infty$ be its condition number. Let $\mathbf{u}$ be a unit round-off. For doubles defined by IEEE 754 standard, $\mathbf{u} = 2^{-53} \approx 1.1 \times 10^{-16}$.

In about 1984, Rump derived a method for inverting arbitrarily ill-conditioned matrices. The method, which he never published, requires the possibility to calculate a dot product $x^T y$ in k-fold precision and store into working precision. In 1990, Rump [10] reported some numerical experiments exhibiting good convergence of his method.

Fortunately, a very efficient method for calculating a dot product in k-fold precision was just recently developed in [7]. It uses only floating point operations in working precision, has no branches and is very fast. For $k = 2$, which is quadruple precision if working precision is double precision, the method is about 40% faster than corresponding routine of XBLAS [6], the state-of-the-art numerical library for this purpose. In [7], we considered a dot product calculation algorithm executable in working precision with a result as if computed in k-fold *precision*. In the new paper [11], we considered how to compute dot products using only working precision with k-fold *accuracy*[1] . In Rump's original proposal, a dot product in k-fold precision is assumed. Recently, Ohta, Ogita, Rump and Oishi [8] have reformulated Rump's method using a dot product calculation algorithm in k-fold accuracy such as proposed in [5,11].

Rump's method is of theoretical interest. Rump made it clear that inverting an *arbitrarily* ill-conditioned matrix A in single or double precision does not produce meaningless numbers (what one might expect), but contains a lot of information in it. Rump's method uses such inverses as preconditioners for A. As shown in [8], numerical experiments exhibit that Rump's method converges rapidly for almost all matrices with extremely large condition number. Why Rump's method is so efficient for inverting arbitrarily ill-conditioned matrices is a little mysterious. Thus, to prove its convergency is an interesting problem in numerical error analysis. In this article, we shall present a convergence theorem for a variant of Rump's method. Numerical experiments are presented for illustrating the validity of our numerical error analyses.

In the present paper as well as in the previous paper [8], Rump's method is employed in a special manner in which computational precision is adaptively

[1] Both algorithms, computing a dot products in k-fold precision [7] as well as in k-fold accuracy [11] are available in INTLAB Version 5.2, the Matlab toolbox for verified computations. Since all code is written in Matlab it is easy to use, but also suffers from interpretation overhead.

increased according to the unknown condition number of the coefficient matrix. One might suspect that computing an approximate inverse in k-fold precision with a choice of sufficiently large k is adequate. However, since the condition number is rarely known a priori, an appropriate choice of k is not possible in general, hence it would lead to a time consuming repetition of trials and errors. We would like to emphasize the inherently adaptive nature of our method which does not waste any intermediate computations in inverting process. We also emphasize that computing the inverse of a coefficient matrix is a necessary measure for giving a rigorous error bound for a numerical solution of a system of linear equations

$$Ax = b \tag{1}$$

with $b \in \mathbb{F}^n$, although it is widely held that computing an inverse is not an efficient strategy for solving (1). Besides, there are various situations which call for the inverse itself (cf. [4, Chapter 14]). In fact, by using inverses generated from Rump's method, a method [8] was given for obtaining a numerical solution with its rigorous error bound to (1) in case of $\kappa(A)\mathbf{u} > 1$.

Very recently, Tanabe has shown that Rump's method can be extended to obtain other numerically important decompositions such as LU and QR decomposition for regular matrices with arbitrarily large condition number [12].

2 Convergence Theorem

We assume that the dimension of the problem, n, satisfies $n\mathbf{u} \ll 1$ and $C_i\sqrt{\mathbf{u}} \ll 1$. In this paper, $C_i,\ i = 0, 1, 2, \cdots$ denote numbers of $\mathcal{O}(1)$ satisfying $C_i\mathbf{u} \ll 1$ and $C_i\sqrt{\mathbf{u}} \ll 1$. Moreover, c_n denotes a numbers of $\mathcal{O}(n)$ satisfying $c_n\mathbf{u} \ll 1$ and $c_n\sqrt{\mathbf{u}} \ll 1$.

Let $A = (a_{ij})$ be a real $n \times n$ matrix and $\Pi = (\pi_{ij})$ be an approximate inverse of A. Let $b \in \mathbb{R}^n$ and $\tilde{x}$ be an approximate solution of $Ax = b$. It is known that if

$$\|\Pi A - I\| < 1 \tag{2}$$

is satisfied, A becomes regular. Here, I is the $n \times n$ identity matrix and $\|\cdot\|$ is a subordinate matrix norm. Further,

$$\|A^{-1}\| \leqq \frac{\|\Pi\|}{1 - \|\Pi A - I\|} \tag{3}$$

and

$$\|\tilde{x} - A^{-1}b\| \leqq \frac{\|\Pi(A\tilde{x} - b)\|}{1 - \|\Pi A - I\|} \tag{4}$$

hold. Rump's method is an algorithm to produce Π. Thus, from the above mentioned fact, we set a purpose of this paper to show that Π generated by Rump's method eventually satisfies (2).

For the purpose, we introduce an accurate dot product calculation algorithm. Let $A, B \in \mathbb{F}^{n\times n}$. Let us assume that we have an accurate dot product algorithm which calculates $D_i \in \mathbb{F}^{n\times n}$, $i = 1, 2, \ldots, k$, satisfying

$$|\sum_{i=1}^{k} D_i - AB| \leqq C_0 \mathbf{u}^k |AB|. \tag{5}$$

Here, AB is the usual (error free) matrix multiplication and C_0 is a constant satisfying $C_0 = \mathcal{O}(1)$. We denote such an algorithm as

$$D_{1:k} = [AB]_k \quad \text{with} \quad D_{1:k} := D_1 + D_2 + \cdots + D_k, \quad D_i \in \mathbb{F}^{n\times n}.$$

A very efficient method for calculating such a dot product in k-fold accuracy was just developed in [11]. It uses only floating point operations in working precision, has no branches and is very fast.

In this paper, to simplify the life, working precision is assumed to be the double precision defined by IEEE 754 floating point standard. In the following, we use a variant of Rump's method as given by the following Algorithm 1, which is written in Matlab-like:

Algorithm 1 *Modified Rump's Method I*

```
$\tilde{S}_0 = A + \Delta A;$                    % perturbation for $A$
$X_0 = \mathrm{inv}(\tilde{S}_0);$   $\Pi_1 = X_0;$
for  $k = 1 : k_{\max}$
    $C = [\Pi_{1:k} A]_1;$
    $\tilde{S}_k = C + \Delta C;$                % perturbation for $[\Pi_{1:k} A]_1$
    $X_k = \mathrm{inv}(\tilde{S}_k);$           % inversion of $\tilde{S}_k$ in working precision
    $\Pi_{1:k+1} = [X_k \Pi_{1:k}]_{k+1};$       % $(k+1)$-fold accuracy
end
```

Here, $\mathrm{inv}(B)$ is a built-in function in Matlab for inverting $B \in \mathbb{F}^{n\times n}$. Matrices $\Delta A \in \mathbb{F}^{n\times n}$ and $\Delta C \in \mathbb{F}^{n\times n}$ are defined by $(\Delta A)_{ij} = r_{ij}\sqrt{\mathbf{u}}|A_{ij}|$ and $(\Delta C)_{ij} = s_{ij}\sqrt{\mathbf{u}}|C_{ij}|$, respectively for all (i, j), where r_{ij} and s_{ij} are pseudo-random numbers distributed uniformly in $[-1, 1]$. Note that the perturbation ΔA and ΔC regularize A and $[\Pi_{1:k} A]_1$, respectively.

To simplify the notation, we will write Π_m instead of $\Pi_{1:m}$ throughout the paper except in algorithms. We assume that all numerical calculation is done under IEEE 754's double precision arithmetic in the nearest rounding mode.

Let $S_k := \Pi_k A$. We now show that

$$|\tilde{S}_k - S_k| \leqq C_1\sqrt{\mathbf{u}}|\tilde{S}_k|, \tag{6}$$

where $C_1 = \mathcal{O}(1)$.

From (5), we have

$$|S_k - [S_k]_1| \leqq C_0\mathbf{u}|S_k|. \tag{7}$$

From the definition of ΔC, it follows that

$$|S_k - \tilde{S}_k| \leqq |S_k - [S_k]_1| + |[S_k]_1 - \tilde{S}_k| \leqq C_0\mathbf{u}|S_k| + \sqrt{\mathbf{u}}|[S_k]_1|. \tag{8}$$

From

$$|S_k| \leqq |[S_k]_1| + |S_k - [S_k]_1| \leqq |[S_k]_1| + C_0\mathbf{u}|S_k|, \tag{9}$$

we have

$$|S_k| \leqq \frac{1}{1 - C_0\mathbf{u}}|[S_k]_1|. \tag{10}$$

Moreover, from

$$|[S_k]_1| \leqq |\tilde{S}_k| + |[S_k]_1 - \tilde{S}_k| \leqq |\tilde{S}_k| + \sqrt{\mathbf{u}}|[S_k]_1|, \tag{11}$$

it follows that

$$|[S_k]_1| \leqq \frac{1}{1 - \sqrt{\mathbf{u}}}|\tilde{S}_k|. \tag{12}$$

Substituting (10) and (12) into (8), it is seen that (6) holds with

$$C_1 = \frac{1}{1 - \sqrt{\mathbf{u}}}(1 + \frac{C_0\sqrt{\mathbf{u}}}{1 - C_0\mathbf{u}}). \tag{13}$$

Using (6), we also have

$$|\tilde{S}_k| \leqq \frac{1}{1 - C_1\sqrt{\mathbf{u}}}|S_k|. \tag{14}$$

Since $\tilde{S}_k \in \mathbb{F}^{n\times n}$, X_k can be computed by a standard inversion algorithm using Gaussian elimination in working precision.

2.1 Decrease of Condition Number

The target of this subsection is to show that

$$\kappa(S_{k+1}) = O(\sqrt{\mathbf{u}})\kappa(S_k) + O(1) \tag{15}$$

provided that $\kappa(S_k) \geqq \mathbf{u}^{-1}$.

For the purpose, in the first place, we estimate $\|S_{k+1}\|_\infty$ assuming that $\kappa(S_k) \geqq \mathbf{u}^{-1}$. Let $\Gamma := \tilde{S}_k - S_k$. Then, from (6) and (14) we have

$$\|\Gamma\|_\infty \le C_1\sqrt{\mathbf{u}}\|\tilde{S}_k\|_\infty \le C_1'\sqrt{\mathbf{u}}\|S_k\|_\infty, \tag{16}$$

where $C_1' := C_1/(1-C_1\sqrt{\mathbf{u}})$. We note here that (16) states that the difference between $\tilde{S}_k$ and S_k, which is almost singular, is of order $\sqrt{\mathbf{u}}\|\tilde{S}_k\|$. Thus, usually a distance between $\tilde{S}_k$ and the nearest singularity, which lies very near to S_k, becomes about $C_1\sqrt{\mathbf{u}}$. This implies (cf. [3,2])

$$\kappa(\tilde{S}_k) = C_2\mathbf{u}^{-1/2}. \tag{17}$$

Here, we assume

Assumption 1 $C_2 = \mathcal{O}(1)$.

This implies $\kappa(\tilde{S}_k) = C_2\mathbf{u}^{-1/2} \ll \mathbf{u}^{-1}$. Examples in the next section show that Assumption 1 is satisfied in many instances. Since a good approximate inverse of a matrix in $\mathbb{F}^{n\times n}$ with a condition number much less than $\mathbf{u}^{-1}$ can be obtained in working precision, under Assumption 1 we can expect that X_k becomes a good approximate inverse of $\tilde{S}_k$ satisfying

Assumption 2 $\|I - X_k\tilde{S}_k\|_\infty = \varepsilon \ll 1$.

We assume that Assumption 2 also holds. It follows from Assumption 2, $\tilde{S}_k^{-1}$ exists. Then, we note that

$$\begin{aligned}\|X_k - \tilde{S}_k^{-1}\|_\infty &= \|(I - X_k\tilde{S}_k)\tilde{S}_k^{-1}\|_\infty \leqq \|\tilde{S}_k^{-1}\|_\infty\|I - X_k\tilde{S}_k\|_\infty \\ &\leqq \frac{\|X_k\|_\infty}{1 - \|I - X_k\tilde{S}_k\|_\infty}\|I - X_k\tilde{S}_k\|_\infty \\ &= \frac{\varepsilon}{1-\varepsilon}\|X_k\|_\infty. \end{aligned} \tag{18}$$

From (18), it follows

$$\|X_k\|_\infty \leqq \|\tilde{S}_k^{-1}\|_\infty + \|X_k - \tilde{S}_k^{-1}\|_\infty \leqq \|\tilde{S}_k^{-1}\|_\infty + \frac{\varepsilon}{1-\varepsilon}\|X_k\|_\infty. \tag{19}$$

This and Assumption 2 imply that

$$\|X_k\|_\infty \leqq \frac{\|\tilde{S}_k^{-1}\|_\infty}{1 - \dfrac{\varepsilon}{1-\varepsilon}} = \frac{1-\varepsilon}{1-2\varepsilon}\|\tilde{S}_k^{-1}\|_\infty = C_3\|\tilde{S}_k^{-1}\|_\infty. \tag{20}$$

Here, $C_3 := (1-\varepsilon)/(1-2\varepsilon) = \mathcal{O}(1)$. Let L and U be computed LU factors of $\tilde{S}_k$. Then, since we have used Matlab's 'inv' function, we have from [4, p. 268, (14.18)]

$$\|I - X_k\tilde{S}_k\|_\infty \leqq c_n\mathbf{u}\|X_k\|_\infty\|L\|_\infty\|U\|_\infty, \tag{21}$$

where $c_n = \mathcal{O}(n)$. Here, we introduce a constant g_k satisfying $\|L\|_\infty \|U\|_\infty \leqq g_k \|\tilde{S}_k\|_\infty$. Then, we have

$$\|I - X_k \tilde{S}_k\|_\infty \leqq c_n g_k \mathbf{u} \|X_k\|_\infty \|\tilde{S}_k\|_\infty. \tag{22}$$

From (17), (20) and (22), it follows that

$$\|I - X_k \tilde{S}_k\|_\infty \leqq c_n g_k C_3 \mathbf{u} \kappa(\tilde{S}_k) = c_n C_4 \sqrt{\mathbf{u}}, \tag{23}$$

where $C_4 := g_k C_2 C_3$. Under Assumption 2, which states $\|I - X_k \tilde{S}_k\|_\infty \ll 1$, (23) asserts $\|I - X_k \tilde{S}_k\|_\infty$ can be estimated as $\mathcal{O}(n\sqrt{\mathbf{u}})$ provided that $C_4 = \mathcal{O}(1)$. Thus, it turns out that Assumption 2 is equivalent to

Assumption 3 $C_4 = \mathcal{O}(1)$ *satisfying* $c_n C_4 \sqrt{\mathbf{u}} \ll 1$.

Under this assumtion, we now show X_k is the exact inverse of $\tilde{S}_k + \Delta$, where $\|\Delta\|_\infty \leqq c_n C_5 \sqrt{\mathbf{u}} \|S_k\|_\infty$. Here, C_5 is the constant defined below. From (23), we have for $\Delta = X_k^{-1} - \tilde{S}_k$

$$\begin{aligned}
\|\Delta\|_\infty &= \|X_k^{-1} - \tilde{S}_k\|_\infty = \|X_k^{-1}(I - X_k \tilde{S}_k)\|_\infty \\
&\leqq \|X_k^{-1}\|_\infty \|I - X_k \tilde{S}_k\|_\infty \\
&\leqq \frac{\|\tilde{S}_k\|_\infty}{1 - \|I - X_k \tilde{S}_k\|_\infty} \|I - X_k \tilde{S}_k\|_\infty \\
&\leqq \frac{c_n C_4 \sqrt{\mathbf{u}}}{1 - c_n C_4 \sqrt{\mathbf{u}}} \|\tilde{S}_k\|_\infty \leqq c_n C_5 \sqrt{\mathbf{u}} \|S_k\|_\infty.
\end{aligned} \tag{24}$$

Here, using (16) we have put

$$C_5 := \frac{C_1' C_4}{C_1(1 - c_n C_4 \sqrt{\mathbf{u}})} = \mathcal{O}(1). \tag{25}$$

Lemma 1 *Let us assume that Assumptions 1 and 3 are satisfied. Then, the following a priori error estimate holds:*

$$\|I - X_k S_k\|_\infty \leqq C_7, \tag{26}$$

where

$$C_7 := C_2 C_3 (C_1 + c_n g_k \sqrt{\mathbf{u}}). \tag{27}$$

Proof. Using (16), (20) and (22), we have

$$\begin{aligned}\|I - X_kS_k\|_\infty &= \|I - X_k(S_k - \tilde{S}_k + \tilde{S}_k)\|_\infty \\ &\leqq \|X_k(S_k - \tilde{S}_k)\|_\infty + \|I - X_k\tilde{S}_k\|_\infty \\ &\leqq C_1\sqrt{\mathbf{u}}\|X_k\|_\infty\|\tilde{S}_k\|_\infty + c_ng_k\mathbf{u}\|X_k\|_\infty\|\tilde{S}_k\|_\infty \\ &\leqq (C_1 + c_ng_k\sqrt{\mathbf{u}})\sqrt{\mathbf{u}}\|X_k\|_\infty\|\tilde{S}_k\|_\infty \\ &= C_6\sqrt{\mathbf{u}}\kappa(\tilde{S}_k), \end{aligned} \tag{28}$$

where

$$C_6 := C_3(C_1 + c_ng_k\sqrt{\mathbf{u}}). \tag{29}$$

This and (17) prove the lemma. □

From this lemma, we have

$$\begin{aligned}\|X_kS_k\|_\infty &\leqq \|X_kS_k - I\|_\infty + \|I\|_\infty = 1 + \|I - X_kS_k\|_\infty \\ &\leqq 1 + C_7. \end{aligned} \tag{30}$$

Here, we derive a relation between S_{k+1} and X_kS_k:

$$\begin{aligned}|S_{k+1} - X_kS_k| &= |\Pi_{k+1}A - X_k\Pi_kA| = |(\Pi_{k+1} - X_k\Pi_k)A| \\ &\leqq |\Pi_{k+1} - X_k\Pi_k||A| \end{aligned} \tag{31}$$

Since $\Pi_{k+1} = [X_k\Pi_k]_{k+1}$, we have

$$|\Pi_{k+1} - X_k\Pi_k| \leqq C_8\mathbf{u}^{k+1}|X_k\Pi_k|. \tag{32}$$

Here, $C_8 = \mathcal{O}(1)$. Inserting this into (31), we have

$$|S_{k+1} - X_kS_k| \leqq C_8\mathbf{u}^{k+1}|X_k||\Pi_k||A|. \tag{33}$$

Thus, we have

$$\|S_{k+1}\|_\infty \leqq \|X_kS_k\|_\infty + \mathbf{u}^{k+1}\alpha, \tag{34}$$

where

$$\alpha := C_8\||X_k||\Pi_k||A|\|_\infty. \tag{35}$$

Here, we assume

Assumption 4 $\mathbf{u}^{k+1}\alpha \ll 1$.

Remark 1 *Since* $\Pi_{k+1} \approx X_k\Pi_k$*, usually we have*

$$\|\Pi_{k+1}\|_\infty \approx \|\Pi_k\|_\infty\|X_k\|_\infty. \tag{36}$$

Here, Π_k*,* $k = 1, 2, \ldots,$ *work as the preconditioners for* A*, we have* $\|S_k\|_\infty = \|\Pi_kA\|_\infty = \mathcal{O}(1)$ *and therefore* $\|\tilde{S}_k\|_\infty = \mathcal{O}(1)$ *for* $k \geqq 1$*. Thus, from (20)*

$$\|X_k\|_\infty \leqq C_3\|\tilde{S}_k^{-1}\|_\infty = C_3\kappa(\tilde{S}_k)\|\tilde{S}_k\|_\infty^{-1} = \mathcal{O}(\mathbf{u}^{-1/2}) \tag{37}$$

for $k \geqq 1$*. Moreover, it can be expected that* $\tilde{S}_0$ *is not so ill-conditioned and* $\kappa(\tilde{S}_0) = \|\tilde{S}_0\|_\infty \|\tilde{S}_0^{-1}\|_\infty = \mathcal{O}(\mathbf{u}^{-1/2})$*, so that* $\|X_0\|_\infty = \|\mathrm{inv}(\tilde{S}_0)\|_\infty \approx \|\tilde{S}_0^{-1}\|_\infty$*. This and* $\|\tilde{S}_0\|_\infty \approx \|A\|_\infty$ *yield*

$$\|X_0\|_\infty = \mathcal{O}(\mathbf{u}^{-1/2})\|A\|_\infty^{-1}. \tag{38}$$

From (36), (37) and (38), it follows

$$\|\Pi_k\|_\infty \approx \mathcal{O}(\mathbf{u}^{-k/2})\|A\|_\infty^{-1} \tag{39}$$

provided that $\kappa(S_k) > \mathbf{u}^{-1}$*. Thus, from (35) we have*

$$\mathbf{u}^{k+1}\alpha \approx \mathcal{O}(\mathbf{u}^{k+1}\mathbf{u}^{-1/2}(\mathbf{u}^{1/2})^{-k})\|A\|_\infty^{-1}\|A\|_\infty = \mathcal{O}(\mathbf{u}^{(k+1)/2}). \tag{40}$$

From this remark, we can expect Assumption 4 is usually satisfied. If this assumption is not satisfied, we modify Algorithm 1 as follows:

Algorithm 2 *Modified Rump's Method II*

```
$\tilde{S}_0 = A + \Delta A$;
$X_0 = \mathrm{inv}(\tilde{S}_0)$;   $\Pi_1 = X_0$;
for  $k = 1 : k_{\max}$
    $C = [\Pi_{1:(k-1)m+1} A]_1$;                    % $m \geqq 1$
    $\tilde{S}_k = C + \Delta C$;
    $X_k = \mathrm{inv}(\tilde{S}_k)$;
    $\Pi_{1:km+1} = [X_k \Pi_{1:(k-1)m+1}]_{km+1}$;   % $(km+1)$-fold accuracy
end
```

Then, Assumption 4 becomes

Assumption 5 $\mathbf{u}^{km+1}\alpha \ll 1$,

which is satisfied for sufficiently large $m \in \mathbb{N}$. Algorithm 2 is used if needed. Thus, without loss of generality, we can assume that Assumption 4 is satisfied.

Under Assumption 4, it can be seen from (34) that

$$\|S_{k+1}\|_\infty = \|X_k S_k\|_\infty + \varepsilon, \tag{41}$$

where $\varepsilon \ll 1$.

Now, we estimate $\|S_{k+1}^{-1}\|_\infty$. Using (16), (24) and (25), we have

$$
\begin{aligned}
\|(X_kS_k)^{-1}\|_\infty &= \|((S_k+\Delta+\Gamma)^{-1}S_k)^{-1}\|_\infty = \|I+S_k^{-1}(\Delta+\Gamma)\|_\infty \\
&\leqq 1+\|S_k^{-1}\|_\infty(\|\Delta\|_\infty+\|\Gamma\|_\infty) \\
&\leqq 1+(C_1'+c_nC_5)\sqrt{\mathbf{u}}\|S_k\|_\infty\|S_k^{-1}\|_\infty \\
&\leqq 1+(C_1'+c_nC_5)\sqrt{\mathbf{u}}\kappa(S_k). \qquad (42)
\end{aligned}
$$

Let P and Q be regular $n \times n$ matrices. If $\|P-Q\|_\infty \leqq \delta$, it follows that

$$
\|P^{-1}-Q^{-1}\|_\infty \leqq \|P^{-1}(P-Q)Q^{-1}\|_\infty \leqq \delta\|P^{-1}\|_\infty\|Q^{-1}\|_\infty. \qquad (43)
$$

Then, (33) and (43) yield

$$
\begin{aligned}
\|S_{k+1}^{-1}-(X_kS_k)^{-1}\|_\infty &\leqq \|S_{k+1}-X_kS_k\|_\infty\|S_{k+1}^{-1}\|_\infty\|(X_kS_k)^{-1}\|_\infty \\
&\leqq \mathbf{u}^{k+1}\beta\|S_{k+1}^{-1}\|_\infty \qquad (44)
\end{aligned}
$$

where

$$
\beta := C_8\||X_k||\Pi_{k+1}||A|\|_\infty\|(X_kS_k)^{-1}\|_\infty.
$$

From (44), we have

$$
\begin{aligned}
\|S_{k+1}^{-1}\|_\infty &\leqq \|S_{k+1}^{-1}-(X_kS_k)^{-1}\|_\infty+\|(X_kS_k)^{-1}\|_\infty \\
&\leqq \mathbf{u}^{k+1}\beta\|S_{k+1}^{-1}\|_\infty+\|(X_kS_k)^{-1}\|_\infty. \qquad (45)
\end{aligned}
$$

If it holds that

Assumption 6 $\mathbf{u}^{k+1}\beta \ll 1$,

then we have

$$
\|S_{k+1}^{-1}\|_\infty \leqq (1-\mathbf{u}^{k+1}\beta)^{-1}\|(X_kS_k)^{-1}\|_\infty. \qquad (46)
$$

If Assumption 6 is not satisfied, we use the modified Rump's method II (Algorithm 2). Namely,

Assumption 7 $\mathbf{u}^{km+1}\beta \ll 1$

is satisfied if we choose $m \in \mathbb{N}$ sufficiently large. Then, (46) becomes

$$
\|S_{k+1}^{-1}\|_\infty \leqq (1-\mathbf{u}^{km+1}\beta)^{-1}\|(X_kS_k)^{-1}\|_\infty. \qquad (47)
$$

Thus, without loss of generality, we can assume that Assumption 6 is satisfied. Then, it holds

$$
\|S_{k+1}^{-1}\|_\infty \leqq C_9\|(X_kS_k)^{-1}\|_\infty, \qquad (48)
$$

where $C_9 = O(1)$.

Summarizing the above mentioned estimations (*i.e.*, from (30), (41), (42) and (48)), we have

$$\begin{aligned}\kappa(S_{k+1}) &= \|S_{k+1}\|_\infty \|S_{k+1}^{-1}\|_\infty \\ &\leqq (\|X_k S_k\|_\infty + \epsilon) C_9 \|(X_k S_k)^{-1}\|_\infty \\ &\leqq (1 + C_7 + \epsilon) C_9 (1 + (C_1' + c_n C_5)\sqrt{\mathbf{u}}\kappa(S_k)) \\ &= \mu_k \sqrt{\mathbf{u}}\kappa(S_k) + \mathcal{O}(1). \end{aligned} \tag{49}$$

Here, $\mu_k := C_9(C_1' + c_n C_5)(1 + C_7 + \varepsilon) = \mathcal{O}(n)$.

Summing up the above mentioned discussions, we have the following theorem:

Theorem 1 *Assume that $\kappa(S_k) \geqq \mathbf{u}^{-1}$. Further, let us assume that Assumptions 1, 3, 4 and 6 (or Assumptions 1, 3, 5 and 7) are satisfied. Then, $\kappa(S_{k+1}) \leqq \mu_k \sqrt{\mathbf{u}}\kappa(S_k) + O(1)$ with $\mu_k = \mathcal{O}(n)$.*

If $\mu_k \sqrt{\mathbf{u}} < 1$ holds for $k = 1, 2, \cdots, K$, then $\kappa(S_k)$ decreases as $\mathcal{O}((n\sqrt{\mathbf{u}})^k)\kappa(A)$ during $k \leqq K$ and finally $\kappa(S_k)$ becomes $\mathcal{O}(1)$ as k becomes sufficiently large provided that $k \leqq K$.

2.2 Convergence

The target of this subsection is to show $\|I - S_{k+1}\|_\infty = \mathcal{O}(\sqrt{\mathbf{u}})$ when $\kappa(S_k) = \mathcal{O}(1)$.

Since $\|S_k - \tilde{S}_k\|_\infty \leqq C_1\sqrt{\mathbf{u}}\|S_k\|_\infty$, the distance between $\tilde{S}_k$ and the nearest singularity is the same order with that between S_k and the nearest singularity. This means that $\kappa(\tilde{S}_k) \approx \kappa(S_k)$. Thus, we have $\kappa(\tilde{S}_k) = \mathcal{O}(1)$. Then, we can expect that X_k becomes a good approximate inverse of $\tilde{S}_k$ satisfying

$$\|I - X_k \tilde{S}_k\|_\infty \ll 1. \tag{50}$$

This implies that there exist $C_3' = \mathcal{O}(1)$ such that

$$\|X_k\|_\infty \leqq C_3' \|\tilde{S}_k^{-1}\|_\infty. \tag{51}$$

Let L and U be computed LU factors of $\tilde{S}_k$. From $\kappa(\tilde{S}_k) = \mathcal{O}(1)$, we can assume that

Assumption 8 $\|L\|_\infty = \mathcal{O}(1)$, $\|U\|_\infty = \mathcal{O}(1)$ *and* $\|X_k\|_\infty = \mathcal{O}(1)$.

Then, from (28) we have

$$\|I - X_k S_k\|_\infty \leqq C_6' \sqrt{\mathbf{u}}\kappa(\tilde{S}_k) = C_{11}\sqrt{\mathbf{u}}, \tag{52}$$

where C_6' is the constant obtained from C_6 by replacing C_3 with C_3' and $C_{11} := C_6'\kappa(\tilde{S}_k)$. Thus, from (33) we have

$$\begin{aligned}\|I - S_{k+1}\|_\infty &\leqq \|I - X_k S_k\|_\infty + \|X_k S_k - S_{k+1}\|_\infty \\ &\leqq C_{11}\sqrt{\mathbf{u}} + \mathbf{u}^{k+1}\alpha, \end{aligned} \tag{53}$$

where α is defined in (35). Since $\kappa(\tilde{S}_k) = \mathcal{O}(1)$, we assume that

Assumption 9 *$C_{11} = \mathcal{O}(1)$.*

Furthermore, we assume that k is so large such that

Assumption 10 *$\mathbf{u}^{k+1}\alpha \ll 1$.*

If this assumption does not hold, we use the modified Rump's method II (Algorithm 2). Then,

$$\|I - S_{k+1}\|_\infty \leqq C_{11}\sqrt{\mathbf{u}} + \mathbf{u}^{km+1}\alpha \tag{54}$$

holds. Thus, if m is large enough, it holds that

Assumption 11 *$\mathbf{u}^{km+1}\alpha \ll 1$.*

Thus, without loss of genelarity, we can assume that Assumption 10 is satisfied and

$$C_{11}\sqrt{\mathbf{u}} + \varepsilon' \tag{55}$$

holds, where $\varepsilon' \ll 1$.

Summing up the above mentioned discussions, we have the following theorem:

Theorem 2 *Let $\kappa(S_k) = \mathcal{O}(1)$. We assume that Assumptions 8, 9 and 10 (or Assumptions 8, 9 and 11) are satisfied. Then, $\|I - S_{k+1}\|_\infty = \|I - \Pi_{k+1}A\|_\infty = C_{11}\sqrt{\mathbf{u}} + \varepsilon' \ll 1$ holds.*

3 Numerical Experiments

3.1 Numerical Examples

We now show the following numerical examples.

Example 1 In the first place, we consider Rump's random matrices with the prescribed condition number [9] as a coefficient matrix A. In this example, we take $n = 20$ and $\kappa(A) \approx 2.0 \times 10^{30}$. In this example, we have $\|A\|_\infty \approx 1.5 \times 10^7$. The result of a numerical experiment is shown in Table 1. In the table, for example, $3.7e + 09 = 3.7 \times 10^9$.

Table 1
Example 1: Rump's random matrix ($n = 20$, $\kappa(A) \approx 2.0 \times 10^{30}$)

k	$\|\tilde{S}_k\|_\infty$	$\|L\|_\infty$	$\|U\|_\infty$	$\|X_k\|_\infty$	$\|I - X_k\tilde{S}_k\|_\infty$	$\|\Pi_k\|_\infty$
1	$8.0e+01$	$8.6e+00$	$8.0e+01$	$3.7e+09$	$1.5e-06$	$1.1e+04$
2	$2.2e+02$	$5.0e+00$	$2.2e+02$	$1.3e+08$	$2.8e-07$	$4.4e+11$
3	$1.5e+01$	$4.3e+00$	$1.5e+01$	$9.6e+06$	$2.7e-09$	$1.9e+17$
4	$1.1e+00$	$1.1e+00$	$1.0e+00$	$1.1e+00$	$4.7e-16$	$1.3e+23$

Example 2 In this example, we also consider Rump's random matrices as a coefficient matrix A. We take $n = 100$ and $\kappa(A) \approx 1.4 \times 10^{113}$. In this case, we have $\|A\|_\infty \approx 1.8 \times 10^{16}$. The result of a numerical experiment is shown in Table 2.

Table 2
Example 2: Rump's random matrix ($n = 100$, $\kappa(A) \approx 1.4 \times 10^{113}$)

k	$\|\tilde{S}_k\|_\infty$	$\|L\|_\infty$	$\|U\|_\infty$	$\|X_k\|_\infty$	$\|I - X_k\tilde{S}_k\|_\infty$	$\|\Pi_k\|_\infty$
1	$2.3e+03$	$2.7e+01$	$2.3e+03$	$2.5e+10$	$7.2e-05$	$5.7e-04$
2	$3.7e+03$	$2.3e+01$	$3.7e+03$	$3.4e+09$	$1.6e-05$	$4.4e+03$
3	$4.2e+02$	$2.1e+01$	$3.7e+02$	$1.1e+10$	$2.1e-05$	$2.5e+09$
4	$4.8e+02$	$2.1e+01$	$4.4e+02$	$1.4e+11$	$2.4e-04$	$4.8e+16$
5	$6.1e+03$	$1.3e+01$	$6.1e+03$	$1.8e+09$	$9.7e-06$	$2.2e+25$
6	$5.8e+02$	$1.5e+01$	$7.3e+02$	$1.1e+10$	$2.3e-05$	$5.4e+30$
7	$4.8e+02$	$9.7e+00$	$4.2e+02$	$6.2e+10$	$5.5e-05$	$1.2e+38$
8	$2.8e+03$	$1.2e+01$	$2.8e+03$	$2.8e+11$	$3.8e-04$	$9.9e+45$
9	$1.9e+04$	$9.2e+00$	$1.9e+04$	$1.9e+10$	$2.2e-05$	$1.4e+54$
10	$2.7e+03$	$8.3e+00$	$2.7e+03$	$2.9e+11$	$9.0e-04$	$1.3e+60$
11	$1.5e+04$	$5.6e+00$	$1.5e+04$	$1.0e+10$	$2.4e-05$	$3.3e+68$
12	$1.3e+03$	$7.1e+00$	$1.3e+03$	$2.0e+11$	$1.3e-04$	$7.8e+73$
13	$4.1e+03$	$8.0e+00$	$4.1e+03$	$9.7e+10$	$9.7e-05$	$5.1e+82$
14	$2.7e+03$	$7.0e+00$	$2.4e+03$	$1.8e+10$	$1.8e-05$	$1.2e+90$
15	$9.9e+02$	$5.0e+00$	$9.9e+02$	$1.3e+03$	$7.7e-13$	$5.9e+96$

Example 3 In this example, we further consider Rump's random matrices as a coefficient matrix A. We take $n = 500$ and $\kappa(A) \approx 1.1 \times 10^{61}$. In this case, we have $\|A\|_\infty \approx 5.7 \times 10^{8}$. The result of a numerical experiment is shown in Table 3.

Table 3
Example 3: Rump's random matrix ($n = 500$, $\kappa(A) \approx 1.1 \times 10^{61}$)

k	$\|\tilde{S}_k\|_\infty$	$\|L\|_\infty$	$\|U\|_\infty$	$\|X_k\|_\infty$	$\|I - X_k\tilde{S}_k\|_\infty$	$\|\Pi_k\|_\infty$
1	$5.2e+03$	$1.1e+02$	$4.3e+03$	$1.6e+10$	$2.2e-04$	$7.7e+04$
2	$4.5e+03$	$9.2e+01$	$4.5e+03$	$7.2e+10$	$1.0e-03$	$2.3e+11$
3	$9.4e+03$	$7.9e+01$	$9.6e+03$	$4.9e+10$	$7.2e-04$	$3.4e+18$
4	$1.4e+04$	$6.3e+01$	$1.4e+04$	$1.7e+10$	$1.6e-04$	$1.6e+25$
5	$3.2e+03$	$3.2e+01$	$2.6e+03$	$6.7e+10$	$2.5e-04$	$1.6e+31$
6	$3.5e+03$	$2.2e+01$	$3.5e+03$	$4.7e+10$	$2.8e-04$	$3.3e+38$
7	$2.6e+03$	$2.5e+01$	$2.4e+03$	$1.4e+10$	$1.3e-04$	$4.9e+45$
8	$3.8e+02$	$1.1e+01$	$3.7e+02$	$3.1e+02$	$8.8e-13$	$2.4e+52$

Example 4 In this example, we consider 20×20 Hilbert matrix H. To avoid expression error, we consider

$$A = \text{const.} \times H.$$

Here, const. is some common multiplier of $2, 3, \dots, 39$. In this example, we have $\kappa(A) \approx 6.3 \times 10^{28}$ and $\|A\|_\infty \approx 1.9 \times 10^{16}$. The result of a numerical experiment is shown in Table 4.

Table 4
Example 4: Hilbert matrix ($n = 20$, $\kappa(A) \approx 6.3 \times 10^{28}$)

k	$\|\tilde{S}_k\|_\infty$	$\|L\|_\infty$	$\|U\|_\infty$	$\|X_k\|_\infty$	$\|I - X_k\tilde{S}_k\|_\infty$	$\|\Pi_k\|_\infty$
1	$5.1e+01$	$7.4e+00$	$4.8e+01$	$6.5e+08$	$3.8e-07$	$4.1e-06$
2	$3.8e+01$	$8.3e+00$	$3.8e+01$	$4.1e+08$	$2.6e-07$	$5.4e+01$
3	$1.5e+01$	$5.6e+00$	$1.3e+01$	$2.9e+05$	$1.0e-10$	$3.4e+08$

3.2 Summary of Numerical Experiments

Results of numerical experiments shown in Eamples 1 to 4 satisfy all assumtions mentioned in this paper. Thus, based on Theorem 1, $\kappa(S_k)$ decrease until $\kappa(S_k)$ becomes $\mathcal{O}(1)$. Once $\kappa(S_k)$ becomes $\mathcal{O}(1)$, based on Theorem 2, $\|I - \Pi_{k+1}A\|_\infty \ll 1$ holds.

Other numerical experiments exhibit similar behaviors.

4 Conjecture

It should be noted that the original Rump's method has the following form:

Algorithm 3 *The Original Rump's Method*

```
$\tilde{S}_0 = A$;
$X_0 = \mathrm{inv}(\tilde{S}_0)$;
$(*)$   while error,  $X_0 = \mathrm{inv}(\tilde{S}_0 + \Delta)$;  end          % $|\Delta| \approx \mathbf{u}|\tilde{S}_0|$
$\Pi_1 = X_0$;
for  $k = 1 : k_{\max}$
    $\tilde{S}_k = [\Pi_{1:k} A]_1$;
    $X_k = \mathrm{inv}(\tilde{S}_k)$;
    $(*)$   while error,  $X_k = \mathrm{inv}(\tilde{S}_k + \Delta)$;  end   % $|\Delta| \approx \mathbf{u}|\tilde{S}_k|$
    $\Pi_{1:k+1} = [X_k \Pi_{1:k}]_{k+1}$;
end
```

Here, the line $(*)$ works as a regularization of $\tilde{S}_k$, which is done similarly in the proposed algorithms. For example, one may set $\Delta_{ij} = d_{ij}\mathbf{u}|\tilde{S}_k|_{ij}$, where d_{ij} is a pseudo-random number distributed uniformly in $[-1, 1]$. Since $\tilde{S}_k$ are extremely ill-conditioned, it may happen that the function 'inv', *i.e.* Gaussian elimination, ends prematurely. This perturbation ensures ending of the algorithm.

Numerical experiments show that this form of Rump's method works much more efficient than Algorithm 1. For example, we again treat Example 2 in Section 3. The result of a numerical experiment by Algorithm 3 is shown in Table 5.

In this example, Algorithm 3 requires only 8 iterations until convergence, while Algorithm 1 required 15 iterations. This means that the convergence speed of the original Rump's method is almost double compared with that for modified Rump's method proposed in this paper. This fact is confirmed by a number of numerical experiments done by the authors.

In the original Rump's method, a distance between $\tilde{S}_k$ and the nearest singularity is usually about $C_{12}\mathbf{u}$. This implies $\kappa(\tilde{S}_k) \approx (C_{12}\mathbf{u})^{-1}$ (cf. [2]). In this case, even if $C_{12} = \mathcal{O}(1)$, $\kappa(\tilde{S}_k)$ becomes an order of $\mathbf{u}^{-1}$. Thus, usually we have

$$\|I - X_k\tilde{S}_k\|_\infty = \mathcal{O}(n) > 1 \tag{56}$$

Table 5
Numerical Result by Original Rump's Method (Example 2)

k	$\|\tilde{S}_k\|_\infty$	$\|L\|_\infty$	$\|U\|_\infty$	$\|X_k\|_\infty$	$\|I - X_k\tilde{S}_k\|_\infty$	$\|\Pi_k\|_\infty$
1	$2.1e+03$	$3.0e+01$	$2.1e+03$	$7.9e+16$	$1.1e+03$	$2.1e+04$
2	$1.1e+03$	$2.5e+01$	$1.1e+03$	$7.5e+16$	$3.7e+02$	$1.6e+18$
3	$3.7e+02$	$2.9e+01$	$4.6e+02$	$7.5e+16$	$3.8e+02$	$1.5e+32$
4	$3.8e+02$	$2.7e+01$	$3.4e+02$	$8.3e+17$	$4.6e+03$	$2.2e+46$
5	$4.6e+03$	$1.9e+01$	$5.6e+03$	$2.9e+16$	$7.0e+02$	$1.7e+61$
6	$7.0e+02$	$2.5e+01$	$1.3e+03$	$1.5e+17$	$9.6e+02$	$5.4e+73$
7	$9.5e+02$	$1.7e+01$	$1.3e+03$	$1.8e+15$	$8.0e+00$	$7.4e+87$
8	$8.8e+00$	$8.9e+00$	$1.9e+00$	$1.0e+01$	$4.3e-15$	$6.6e+96$

provided that $\kappa(S_k) > \mathbf{u}^{-1}$. This is also confirmed from numerical experiments as shown in Table 4. Thus, the arguments in Section 2 cannot be applied to this case.

However, even in this case, the following conjecture might be held:

Conjecture 1 *X_k is the exact inverse of $\tilde{S}_k + \Delta$, where $\|\Delta\|_\infty \leqq c_n \mathbf{u} \|\tilde{S}_k\|_\infty$ with $c_n = \mathcal{O}(n)$.*

If we can prove this conjecture, the convergence of the original Rump's method follows by the similar arguments in Section 2. If this is the case, its convergence speed is like $\mathcal{O}((n\mathbf{u})^k)\kappa(A)$ during $\kappa(S_k) > \mathbf{u}^{-1}$, which is consistent with our numerical experiments.

However, until now we cannot prove Conjecture 1. Thus, saying fairly, the convergence proof of the original Rump's method is still open. However, the authors think their arguments clarify at least a part of a mechanism of the convergence of Rump's method.

Acknowledgements

This research was partially supported by Grant-in-Aid for Specially Promoted Research (No. 17002012: Establishment of Verified Numerical Computation) from the Ministry of Education, Science, Sports and Culture of Japan.

References

[1] ANSI/IEEE, IEEE Standard for Binary Floating Point Arithmetic, Std 754–1985 edition, IEEE, New York, 1985.

[2] J. B. Demmel, Condition numbers and the distance to the nearest ill-posed problem, Numer. Math., 51 (1987), 251–289.

[3] C. Eckhart and G. Young, The approximatoin of one matrix by another of lower rank, Psychometrika, 1 (1936), 211–218

[4] N. J. Higham, Accuracy and Stability of Numerical Algorithms, 2nd ed., SIAM Publications, Philadelphia, PA, 2002.

[5] U. W. Kulisch, W. L. Miranker, The arithmetic of the digital computer: A new approach, SIAM Review, 28 (1986), 1–40.

[6] X. Li, J. B. Demmel, D. Bailey, G. Henry, Y. Hida, J. Iskandar, W. Kahan, S. Kang, A. Kapur, M. Martin, B. Thompson, T. Tung, D. Yoo, Design, implementation and testing of extended and mixed precision BLAS, ACM Trans. Math. Softw., 28 (2002), 152–205.

[7] T. Ogita, S. M. Rump, S. Oishi, Accurate sum and dot product, SIAM J. Sci. Comput., 26:6 (2005), 1955–1988.

[8] T. Ohta, T. Ogita, S. M. Rump, S. Oishi, Numerical verification method for arbitrarily ill-conditioned linear systems, Trans. JSIAM, 15:3 (2005), 269–287. [in Japanese]

[9] S. M. Rump, A class of arbitrarily ill-conditioned floating-point matrices, SIAM J. Matrix Anal. Appl., 12:4 (1991), 645–653.

[10] S. M. Rump, Approximate inverses of almost singular matrices still contain useful information, Forschungsschwerpunktes Informations- und Kommunikationstechnik, Technical Report 90.1, Hamburg University of Technology, 1990.

[11] S. M. Rump, T. Ogita, S. Oishi, Accurate floating-point summation, 41 pages, submitted for publication, 2006. Preprint is available from `http://www.ti3.tu-harburg.de/publications/rump` .

[12] K. Tanabe, private communication.