Fast Verified Solutions of Sparse Linear Systems with *H*-matrices*

A. Minamihata

Graduate School of Fundamental Science and Engineering, Waseda University, Tokyo, Japan
aminamihata@moegi.waseda.jp

K. Sekine

Graduate School of Fundamental Science and Engineering, Waseda University, Tokyo, Japan

T. Ogita

Division of Mathematical Sciences, School of Arts and Sciences, Tokyo Woman's Christian University, Tokyo, Japan
ogita@lab.twcu.ac.jp

S. Oishi

Department of Applied Mathematics, Faculty of Science and Engineering, Waseda University, Tokyo, Japan

Abstract

This paper is concerned with the problem of verifying the accuracy of an approximate solution of a sparse linear system whose coefficient matrix is an *H*-matrix. Fast and efficient methods of calculating componentwise error bounds of the computed solution are proposed. The methods are based on the verified criterion for an *M*-matrix. The main point of this article is that the proposed methods can be applied with any iterative solution methods such as the Gauss-Seidel method and Krylov subspace methods. Therefore, the sparsity of the coefficient matrix is preserved in

*Submitted: February 25, 2013; Revised: September 23, 2013; Accepted: December 5, 2013.

the verification process. Numerical results are presented, illustrating the performance of the proposed methods.

Keywords: verified numerical computations, sparse linear systems, H-matrix
AMS subject classifications: 65G20, 65F30, 65F50

1 Introduction

Let $\mathbb{R}$ be the set of real numbers. For a linear system

$$Ax = b, \quad A \in \mathbb{R}^{n\times n},\ b \in \mathbb{R}^n, \tag{1}$$

we can obtain an approximate solution efficiently by some numerical algorithm. In general, however, we do not know how accurate the computed solution is.

In this paper, we deal with the case where A is an H-matrix, which is also known as a generalized diagonally dominant matrix. H-matrices arise in numerical computations for various applications including fluid dynamics, electromagnetics, and so forth [1, 2]. The purpose of this article is to show how to verify the accuracy of a computed solution $\tilde{x}$ of a sparse linear system $Ax = b$ in the case of A being an H-matrix. In particular, we compute a rigorous error bound of the computed solution while preserving the sparsity of the coefficient matrix A. Thus, we never calculate the inverse of A nor a complete LU factorization explicitly.

To our knowledge, few verification methods for general sparse linear systems are known except methods by Rump [13]. In fact, the verification for sparse systems of linear (interval) equations is known as one of the important open problems posed by Neumaier in *Grand Challenges and Scientific Standards in Interval Analysis* [10]. Moreover, Rump [15] states the problems more concretely. See [15] for details.

On the other hand, some verification methods suited for general dense linear systems using the property of H-matrices have been proposed (cf. e.g., [9, 16]), which rely on the fact that a preconditioned matrix RA, where R is normally taken as an approximate (full) inverse of A, is expected to be an H-matrix. Such methods also can be used for sparse linear systems if A is an H-matrix, since the preconditioner R is not necessary.

In this paper, we propose fast and efficient verification methods for sparse linear systems with H-matrices, i.e., we aim to calculate componentwise error bounds of a computed solution $\tilde{x}$ of $Ax = b$ to the exact solution $x^* := A^{-1}b$ such that

$$|x_i^* - \tilde{x}_i| \le \epsilon_i \quad \text{for } 1 \le i \le n$$

by the use of verified numerical computations.

The rest of the paper is organized as follows. In Section 2, we explain the notation and state definitions used in this paper. In Section 3, we introduce some methods of determining whether a given matrix is an M-matrix. In Section 4, we review the existing theorems which can be used for verified solutions of sparse linear systems with H-matrices. In Section 5, we propose several verification methods for sparse linear systems with H-matrices by componentwise error estimation. Finally, in Section 6, we present some numerical results showing the performance of the proposed methods.

2 Notation and Definitions

Let I and O denote the $n \times n$ identity matrix and the $n \times n$ matrix of all zeros, respectively. Moreover, let e and $\mathbf{0}$ denote the n-vector of all ones and that of all zeros, respectively. Inequalities for matrices are understood componentwise, e.g., for real $n \times n$ matrices $A = (a_{ij})$ and $B = (b_{ij})$, the notation $A \leq B$ means $a_{ij} \leq b_{ij}$ for all (i, j). In particular, the notation $A \geq O$ (or $A > O$) means that all the elements of A are nonnegative (or positive). Moreover, the notation $|A|$ means $|A| = (|a_{ij}|) \in \mathbb{R}^{n \times n}$, the nonnegative matrix consisting of componentwise absolute values of A. Similar notations are applied to real vectors.

Let $A = (a_{ij}) \in \mathbb{R}^{n \times n}$. The spectral radius of A is denoted by $\rho(A)$, which is the largest absolute value of the eigenvalues of A. The comparison matrix $\langle A \rangle = (\widehat{a}_{ij})$ of A is defined by

$$\widehat{a}_{ij} = \begin{cases} |a_{ij}|, & \text{if } \ i = j, \\ -|a_{ij}|, & \text{if } \ i \neq j. \end{cases}$$

This can also be written by $\langle A \rangle = |D| - |E|$, where D and E correspond to diagonal and off-diagonal parts of A, respectively.

Definition 2.1 (strict diagonal dominance) *A real $n \times n$ matrix $A = (a_{ij})$ is called strictly diagonally dominant if $|a_{ii}| > \sum_{j \neq i} |a_{ij}|$ for $i = 1, 2, \ldots, n$.*

Definition 2.2 (monotonicity) *A real $n \times n$ matrix A is called monotone if $Ax \geq \mathbf{0}$ implies $x \geq \mathbf{0}$ for $x \in \mathbb{R}^n$.*

Definition 2.3 (Z-matrix) *Let $A = (a_{ij})$ be a real $n \times n$ matrix with $a_{ij} \leq 0$ for $i \neq j$. Then A is called a Z-matrix.*

Definition 2.4 (L-matrix) *If all the diagonal entries of a Z-matrix A are positive, then A is called an L-matrix.*

Definition 2.5 (M-matrix) *If a Z-matrix A is monotone, then A is called an M-matrix.*

Throughout the paper, we say a matrix A has the "M-property" if A is an M-matrix.

Definition 2.6 (H-matrix) *If $\langle A \rangle$ is an M-matrix, then A is called an H-matrix.*

Lemma 2.1 (e.g., [8, p.113]) *If A is an H-matrix, then $|A^{-1}| \leq \langle A \rangle^{-1}$.*

By Definition 2.6, we find that verifying the M-property is essential for confirming whether a given matrix is an H-matrix. Therefore, we focus on the verification of the M-property in the next section.

3 Verification of M-property

In this section, we consider how to verify whether a given matrix has the M-property. We first introduce a theorem, which plays an important role for this purpose.

Theorem 3.1 (e.g. Fiedler-Pták [4]) *Let an L-matrix $A \in \mathbb{R}^{n \times n}$ be given. Then the following conditions are equivalent:*

1. *A is nonsingular, and $A^{-1} \geq O$ (i.e., A is an M-matrix).*
2. *There exists $v \in \mathbb{R}^n$ with $v > \mathbf{0}$ satisfying $Av > \mathbf{0}$.*
3. *A can be expressed as $A = \mu I - B$ using some $\mu \in \mathbb{R}$ and a nonnegative matrix $B \in \mathbb{R}^{n \times n}$ satisfying $\rho(B) < \mu$.*

The point is that it is sufficient to find some $v > \mathbf{0}$ satisfying $Av > \mathbf{0}$ to verify that A is an M-matrix from condition 2. For condition 3, it is known that $\rho(B) < \mu$ holds for any μ satisfying $\max_{1 \leq i \leq n} a_{ii} \leq \mu$ if A is an M-matrix.

Lemma 3.1 *Let $A \in \mathbb{R}^{n \times n}$ and $v \in \mathbb{R}^n$ with $v > \mathbf{0}$ be given. If $\langle A \rangle v > \mathbf{0}$, then $|A^{-1}| \leq \langle A \rangle^{-1}$.*

Proof: Theorem 3.1 and Lemma 2.1 prove the lemma. ■

In the following, we introduce the existing verification methods for the M-property. We then present an alternative method using the eigenvector associated with the smallest eigenvalue.

3.1 Existing Verification Methods for the M-property

We first introduce a simple verification method for the M-property suggested by Neumaier [9]. Let A be an $n \times n$ L-matrix. Let $b \in \mathbb{R}^n$ with $b > \mathbf{0}$. If A has the M-property, then $A^{-1}b > \mathbf{0}$. Let $\tilde{y}$ be a computed solution of a linear system $Ay = b$. If A is not ill-conditioned, then we can expect that $\tilde{y} > \mathbf{0}$ and $A\tilde{y} > \mathbf{0}$ are satisfied. The choice of b is arbitrary, and a natural one is $b = e$.

Next, we introduce a verification method for the M-property by Rump [16], based on the Perron-Frobenius theorem. We split an L-matrix A as follows:

$$A = D - E \quad \text{with} \quad D > O,\ E \geq O,$$

where D and E correspond to diagonal and off-diagonal parts of A, respectively. The matrix $A = D - E$ has the M-property if and only if $\beta := \rho(D^{-1}E) < 1$. Since $D^{-1}E \geq O$, β corresponds to the Perron root of $D^{-1}E$. Usually, $D^{-1}E$ is at least nonnegative irreducible if not positive, so that the corresponding eigenvector is positive. Then a power iteration

$$v^{(k+1)} = D^{-1}Ev^{(k)} \quad \text{for} \quad v^{(0)} := D^{-1}e$$

implies that $\max_i v_i^{(k+1)} / v_i^{(k)}$ decreases monotonically to the Perron root of $D^{-1}E$. Assume $D^{-1}Ev^{(k)} \approx \beta v^{(k)}$ for $\beta < 1$, then

$$Av^{(k)} = D(I - D^{-1}E)v^{(k)} \approx (1 - \beta)Dv^{(k)}.$$

Therefore, starting with $v := D^{-1}e$, we perform power iterations to find a positive vector v satisfying $Av > \mathbf{0}$.

Apart from the above mentioned methods, several criteria for H-matrix have been proposed. See [5, 6, 7] and the literature cited there for details.

3.2 Alternative Method

Suppose A is an irreducible M-matrix. From the Perron-Frobenius theorem, there exists the positive eigenvector of A^{-1} associated with the largest eigenvalue of A^{-1}, which corresponds to the smallest eigenvalue $\lambda_{\min} > 0$ of A. If we compute an approximation v of the Perron vector of A^{-1}, then we can expect v to satisfy $Av \approx \lambda_{\min} v$ and $Av > \mathbf{0}$. Below, we show how to compute v. We assume $\max_{1 \le i \le n} a_{ii} \le \mu$. Then the Perron vector of A^{-1} corresponds to that of $B := \mu I - A \ge O$. From Theorem 3.1, it holds that $\gamma := \rho(B) < \mu$. An eigenvalue problem

$$Ax = \lambda x$$

can be rewritten as a new eigenvalue problem

$$Bx = (\mu - \lambda)x.$$

Then $\gamma = \mu - \lambda_{\min}$, where $\lambda_{\min}$ is equal to the reciprocal of the Perron root of A^{-1}, so that the Perron vector of B corresponds to that of A^{-1}. Therefore, a power iteration

$$v^{(k+1)} = Bv^{(k)} \quad \text{for} \quad v^{(0)} := e$$

implies that $\max_i v_i^{(k+1)}/v_i^{(k)}$ decreases monotonically to the Perron root of B. Assume $Bv^{(k)} \approx \gamma v^{(k)}$ for $\gamma < \mu$, then

$$Av^{(k)} = (\mu I - B)v^{(k)} \approx (\mu - \gamma)v^{(k)}.$$

Therefore, if A is an irreducible M-matrix, we perform power iterations starting with $v := e$ to find a positive vector v satisfying $Av > \mathbf{0}$.

Although this method is similar to that of Rump [16] from the previous subsection, its behavior is a little different. We can see that in the results of numerical experiments in Section 6.

4 Verification Theory for Linear Systems

We first cite a theorem for bounding $\|A^{-1}\|_\infty$ and obtaining a normwise error bound of a linear system $Ax = b$ in the case of A being monotone.

Theorem 4.1 (Ogita-Oishi-Ushiro [11]) *Let $A \in \mathbb{R}^{n \times n}$ be monotone, and $\tilde{y} \in \mathbb{R}^n$ be given. If $\|e - A\tilde{y}\|_\infty < 1$, then*

$$\|A^{-1}\|_\infty \le \frac{\|\tilde{y}\|_\infty}{1 - \|e - A\tilde{y}\|_\infty}.$$

Using this theorem, if $\|e - A\tilde{y}\|_\infty < 1$, then it holds for b and $\tilde{x} \in \mathbb{R}^n$ that

$$\begin{aligned} \|A^{-1}b - \tilde{x}\|_\infty &\le \|A^{-1}\|_\infty \|b - A\tilde{x}\|_\infty \\ &\le \frac{\|\tilde{y}\|_\infty \|b - A\tilde{x}\|_\infty}{1 - \|e - A\tilde{y}\|_\infty}. \end{aligned} \tag{2}$$

In practice, $\tilde{x}$ and $\tilde{y}$ are taken as computed solutions of the linear systems $Ax = b$ and $Ay = e$, respectively. Note that it is difficult in general to confirm whether a given matrix is monotone.

In addition, Ogita, Oishi and Ushiro [11] also introduced a different type of componentwise error bound as below.

Theorem 4.2 (Ogita-Oishi-Ushiro [11]) *Let $A \in \mathbb{R}^{n\times n}$ and b, $\tilde{x}$, $\tilde{z} \in \mathbb{R}^n$ be given. Suppose A is nonsingular and $\|A^{-1}\|_p \le \tau$ for any $p \in \{1, 2, \infty\}$. Then*

$$|A^{-1}b - \tilde{x}| \le |\tilde{z}| + \tau\|b - A(\tilde{x} + \tilde{z})\|_p e. \tag{3}$$

Here $\tilde{z}$ is also known as the staggered correction whose concept has already appeared in [12]. If the correction $\tilde{z}$ is of high quality and $|\tilde{z}| \gg \tau\|b - A(\tilde{x} + \tilde{z})\|_p e$, then (3) gives a tight error bound of $\tilde{x}$.

In [10], Neumaier presented a simple method of obtaining an upper bound on the inverse of an H-matrix as follows. Let A and $B \in \mathbb{R}^{n\times n}$ be given. Assume $v \in \mathbb{R}^n$ with $v > \mathbf{0}$ satisfies $u := \langle A\rangle v > \mathbf{0}$. Then A is an H-matrix. Define $w \in \mathbb{R}^n$ by

$$w_k := \max_{1\le i\le n} \frac{G_{ik}}{u_i} \quad \text{for } 1 \le k \le n,$$

where $G := I - \langle A\rangle B$. Then

$$|A^{-1}| \le \langle A\rangle^{-1} \le B + vw^T. \tag{4}$$

In practice, B is taken as an approximate inverse of $\langle A\rangle$.

In [16], Rump applied Neumaier's method to calculating an error bound of a computed solution of a linear system by replacing A by RA, where R is an approximate inverse of A and putting $B = D^{-1}$, where D is the diagonal part of A.

Theorem 4.3 (Rump [16]) *Let A, $R \in \mathbb{R}^{n\times n}$ and b and $\tilde{x} \in \mathbb{R}^n$ be given. Assume $v \in \mathbb{R}^n$ with $v > \mathbf{0}$ satisfies $u := \langle RA\rangle v > \mathbf{0}$. Denote by $\langle RA\rangle := D - E$ the splitting of $\langle RA\rangle$ into diagonal and off-diagonal parts, and define $w \in \mathbb{R}^n$ by*

$$w_k := \max_{1\le i\le n} \frac{G_{ik}}{u_i} \quad \textit{for } 1 \le k \le n,$$

where $G := I - \langle RA\rangle D^{-1} = ED^{-1} \ge O$. Then A and R are nonsingular, and

$$|A^{-1}b - \tilde{x}| \le (D^{-1} + vw^T)|R(b - A\tilde{x})|. \tag{5}$$

Moreover, $|A^{-1}b - \tilde{x}| \le \epsilon$ implies

$$|A^{-1}b - \tilde{x}| \le D^{-1}(|R(b - A\tilde{x})| + E\epsilon).$$

If A is an H-matrix, then we can use Theorem 4.3 by putting $R = I$. However, the error bounds obtained by the theorem may become very pessimistic, since it implicitly assumes that an approximate inverse R of A is used as a preconditioner and $RA \approx I$. We will improve this point in the next section.

5 Componentwise Error Estimation

Based on the results of the above sections, we present theorems for calculating a componentwise error bound for a computed solution of a linear system $Ax = b$ in the case of A being an H-matrix. All the methods in this section are based on the simple estimates

$$|A^{-1}b - \tilde{x}| \le |A^{-1}|\ |b - A\tilde{x}| \tag{6}$$

and

$$|A^{-1}b - \tilde{x}| \le |\tilde{z}| + |A^{-1}|\ |b - A(\tilde{x} + \tilde{z})|. \tag{7}$$

Although (7) can easily be deduced from (6), an error bound obtained by (7) often becomes much better than that by (6) provided that $\tilde{z}$ is sufficiently accurate. Note that this technique has already been used in Theorem 4.2.

Remark 1 *Let D be the diagonal part of an H-matrix A. Then $\rho(|I - D^{-1}A|) < 1$. Thus, we can easily compute a correction vector $\tilde{z}$ by solving a linear system $Az = r$, where $r := b - A\tilde{x}$ using some Jacobi or Gauss-Seidel iterations.*

Remark 2 *If either A or b is an interval quantity, then there is a limit to the correction for $\tilde{x}$ by $\tilde{z}$. For example, if b is an interval vector, then neither $|b - A\tilde{x}|$ nor $|b - A(\tilde{x} + \tilde{z})|$ can become less than the radius of b. In this case, there is almost no difference between the estimates* (6) *and* (7) *for the best possible $\tilde{x}$.*

We first present a lemma to derive a theorem for componentwise error bounds.

Lemma 5.1 *For $A \in \mathbb{R}^{n\times n}$ and $x \in \mathbb{R}^n$, it holds that*

$$|A|\ |x| \le \|x\|_\infty |A|e.$$

Proof: It holds that $|x| \le \|x\|_\infty e$, and the result follows. ■

The following theorem is a modification of Theorem 4.1.

Theorem 5.1 *Let $A \in \mathbb{R}^{n\times n}$ be monotone, and b, $\tilde{x}$, $\tilde{y}$, $\tilde{z} \in \mathbb{R}^n$ be given. If $\|e - A\tilde{y}\|_\infty < 1$, then*

$$|A^{-1}b - \tilde{x}| \le \frac{\|b - A\tilde{x}\|_\infty}{1 - \|e - A\tilde{y}\|_\infty}\ |\tilde{y}| \tag{8}$$

and

$$|A^{-1}b - \tilde{x}| \le |\tilde{z}| + \frac{\|b - A(\tilde{x} + \tilde{z})\|_\infty}{1 - \|e - A\tilde{y}\|_\infty}\ |\tilde{y}|. \tag{9}$$

Proof: Since A is monotone, $|A^{-1}| = A^{-1}$. By Lemma 5.1, it follows that

$$\begin{aligned} |A^{-1}e - \tilde{y}| &= |A^{-1}(e - A\tilde{y})| \le |A^{-1}|\ |e - A\tilde{y}| \\ &\le \|e - A\tilde{y}\|_\infty |A^{-1}|e = \|e - A\tilde{y}\|_\infty A^{-1}e \end{aligned}$$

and

$$(1 - \|e - A\tilde{y}\|_\infty)A^{-1}e \le |\tilde{y}|.$$

If $\|e - A\tilde{y}\|_\infty < 1$, then

$$|A^{-1}|e = A^{-1}e \le \frac{1}{1 - \|e - A\tilde{y}\|_\infty}|\tilde{y}|. \tag{10}$$

By Lemma 5.1 and (6), we have

$$|A^{-1}b - \tilde{x}| \le \|b - A\tilde{x}\|_\infty |A^{-1}|e. \tag{11}$$

Inserting (10) into (11) yields (8). Moreover, by Lemma 5.1 and (7), we have

$$|A^{-1}b - \tilde{x}| \le |\tilde{z}| + \|b - A(\tilde{x} + \tilde{z})\|_\infty |A^{-1}|e. \tag{12}$$

Inserting (10) into (12) yields (9). ■

In addition, we can derive componentwise error bounds for H-matrices in a similar way to the derivation of Theorem 5.1 as follows:

Corollary 5.1 *Let $A \in \mathbb{R}^{n\times n}$ and b, $\tilde{x}$, $\tilde{y}$, $\tilde{z} \in \mathbb{R}^n$ be given with $\tilde{y} > \mathbf{0}$. If*

$$\|e - \langle A\rangle\tilde{y}\|_\infty < 1, \tag{13}$$

then

$$|A^{-1}b - \tilde{x}| \leq \frac{\|b - A\tilde{x}\|_\infty}{1 - \|e - \langle A\rangle\tilde{y}\|_\infty}\ |\tilde{y}| \tag{14}$$

and

$$|A^{-1}b - \tilde{x}| \leq |\tilde{z}| + \frac{\|b - A(\tilde{x} + \tilde{z})\|_\infty}{1 - \|e - \langle A\rangle\tilde{y}\|_\infty}\ |\tilde{y}|. \tag{15}$$

Proof: Since (13) implies $\langle A\rangle\tilde{y} > \mathbf{0}$, $|A^{-1}| \leq \langle A\rangle^{-1}$ holds from Lemma 3.1. The rest can easily be deduced from the proof of Theorem 5.1. ■

The following is a simple modification of Theorem 4.3 suited for the case where A is an H-matrix, which is based on an approach similar to Theorem 4.2.

Corollary 5.2 *Let $A \in \mathbb{R}^{n\times n}$ and b, $\tilde{x}$, $\tilde{z} \in \mathbb{R}^n$ be given. Assume $v \in \mathbb{R}^n$ with $v > \mathbf{0}$ satisfies $u := \langle A\rangle v > \mathbf{0}$. Denote by $\langle A\rangle := D - E$ the splitting of $\langle A\rangle$ into diagonal and off-diagonal parts, and define $w \in \mathbb{R}^n$ by*

$$w_k := \max_{1\leq i\leq n} \frac{G_{ik}}{u_i} \quad \text{for } 1 \leq k \leq n,$$

where $G := I - \langle A\rangle D^{-1} = ED^{-1} \geq O$. Then

$$|A^{-1}b - \tilde{x}| \leq \left(D^{-1} + vw^T\right)\ |b - A\tilde{x}| \tag{16}$$

and

$$|A^{-1}b - \tilde{x}| \leq |\tilde{z}| + \left(D^{-1} + vw^T\right)|b - A(\tilde{x} + \tilde{z})|\,. \tag{17}$$

Proof: In (4), putting $B = D^{-1}$ yields

$$|A^{-1}| \leq \langle A\rangle^{-1} \leq D^{-1} + vw^T,$$

and the results follow. ■

Next, we present the following theorem as an alternative verification method, whose derivation is similar to that of (4).

Theorem 5.2 *Let $A \in \mathbb{R}^{n\times n}$ and b, $\tilde{x}$, $\tilde{z} \in \mathbb{R}^n$ be given. Assume $v \in \mathbb{R}^n$ with $v > \mathbf{0}$ satisfies $\langle A\rangle v > \mathbf{0}$. If*

$$\mathbf{0} < |b - A\tilde{x}| \leq \alpha\langle A\rangle v \tag{18}$$

is satisfied for some positive $\alpha \in \mathbb{R}$, then

$$|A^{-1}b - \tilde{x}| \leq \alpha v. \tag{19}$$

Moreover, if

$$\mathbf{0} < |b - A(\tilde{x} + \tilde{z})| \leq \beta\langle A\rangle v \tag{20}$$

is satisfied for some positive $\beta \in \mathbb{R}$, then

$$|A^{-1}b - \tilde{x}| \leq |\tilde{z}| + \beta v. \tag{21}$$

Proof: By the assumption, $|A^{-1}| \leq \langle A \rangle^{-1}$ stems from Lemma 3.1. From (18), we have

$$\langle A \rangle^{-1} \ |b - A\tilde{x}| \leq \alpha v. \tag{22}$$

Here (6) implies

$$|A^{-1}b - \tilde{x}| \leq \langle A \rangle^{-1} \ |b - A\tilde{x}|. \tag{23}$$

Combining (22) and (23) proves (19). In a similar way, (21) can also be proved. ■

The following corollary can be applied to the case where A is an M-matrix, which gives better error bounds than Theorem 5.2.

Corollary 5.3 *Let an L-matrix $A \in \mathbb{R}^{n \times n}$ and b and $\tilde{x} \in \mathbb{R}^n$ be given. Assume $v \in \mathbb{R}^n$ with $v > \mathbf{0}$ satisfies $Av > \mathbf{0}$. If*

$$\alpha Av \leq b - A\tilde{x} \leq \beta Av \tag{24}$$

is satisfied for some α and $\beta \in \mathbb{R}$, then

$$\alpha v \leq A^{-1}b - \tilde{x} \leq \beta v. \tag{25}$$

Proof: In view of Theorem 3.1, the assumption implies that A is an M-matrix, so that $A^{-1} \geq O$. By monotonicity, multiplying A^{-1} by (24) yields (25). ■

Note that (25) gives more tight bounds than (19) in Theorem 5.2, because the evaluation of (24) does not require taking the absolute value of the residual, while that of (18) does. In (24), replacing $\tilde{x}$ by $\tilde{x} + \tilde{z}$ leads to

$$\alpha Av \leq b - A(\tilde{x} + \tilde{z}) \leq \beta Av,$$

and

$$\tilde{z} + \alpha v \leq A^{-1}b - \tilde{x} \leq \tilde{z} + \beta v, \tag{26}$$

which is similar to the modification from (19) to (21).

The basic ideas of Theorem 5.2 and Corollaries 5.1 and 5.2 look quite similar. The differences among them are as follows: in Corollary 5.1, we first need to estimate $\langle A \rangle^{-1} e$ and then obtain an error bound of a computed solution $\tilde{x}$ by taking the maximum norm $\|b - A\tilde{x}\|_\infty$, which sometimes causes overestimations. On the other hand, in Theorem 5.2, we estimate $\langle A \rangle^{-1} |b - A\tilde{x}|$ directly, whose result v can also be used for calculating α in (19) while aiming $\alpha \approx 1$. Thus, it is expected that (19) in Theorem 5.2 gives more tight error bounds than (14) in Corollary 5.1. If the correction $\tilde{z}$ is of good quality, then there is almost no difference between the error bounds (15) and (21), since both of their second terms can be ignored anyway. Corollary 5.2 should work effectively only if A is nearly diagonal in the sense that $\rho(ED^{-1}) \ll 1$. Otherwise, the term vw^T in (17) tends to become much larger than D^{-1} in magnitude. Therefore, the use of Theorem 5.2 is expected to be more stable than that of Corollary 5.2.

There remains the problem of how to know whether the correction $\tilde{z}$ is good enough. To calculate $\tilde{z}$, we can suitably solve a linear system $Az = r$, where $r := b - A\tilde{x}$ by some Jacobi or Gauss-Seidel iterations. To avoid overestimations of the error bounds, we need appropriate stopping criteria with a tolerance $0 < \varepsilon_{\mathrm{tol}} < 1$ such as

$$|\tilde{z}| \geq \varepsilon_{\mathrm{tol}} \cdot \frac{\|b - A(\tilde{x} + \tilde{z})\|_\infty}{1 - \|e - \langle A \rangle \tilde{y}\|_\infty} \ |\tilde{y}| \tag{27}$$

for (15) in Corollary 5.1,

$$|\tilde{z}| \geq \varepsilon_{\mathrm{tol}} \cdot \left(D^{-1} + vw^T\right) \ |b - A(\tilde{x} + \tilde{z})| \tag{28}$$

for (17) in Corollary 5.2, and

$$|\tilde{z}| \geq \varepsilon_{\mathrm{tol}} \cdot \beta v \tag{29}$$

for (21) in Theorem 5.2. The main difference among them is that all the quantities in the right-hand side of (27) and (28) are fixed except the residual $b - A(\tilde{x} + \tilde{z})$, while that is not the case with (29), since β in (29) depends on $\tilde{z}$ and v. Namely, if we adopt a verification method based on Theorem 5.2 with (29) as a stopping criterion of some iterative method for solving $Az = r$, then additional computational effort is needed for updating v. Of course, v can be fixed independently of $\tilde{z}$, e.g., by the use of v satisfying (18) instead of (20). In this case, however, the quantity βv via (18) tends to become relatively large compared with that via (20).

Summarizing the above discussions, we now present algorithms based on Theorem 5.2. In the algorithms, $\mathbf{u}$ denotes the relative rounding error unit.

Algorithm 5.1 *For given $A \in \mathbb{R}^{n\times n}$ and b and $\tilde{x} \in \mathbb{R}^n$, the following algorithm calculates an upper bound $d \in \mathbb{R}^n$ of $|A^{-1}b - \tilde{x}|$ while verifying that A is an H-matrix, which is based on* (19) *in Theorem 5.2.*

1. *Calculate an upper bound $\overline{r} \in \mathbb{R}^n$ of $|b - A\tilde{x}|$ (in higher precision arithmetic if possible). If $\overline{r}_i = 0$ for some i, then compute $\gamma = \mathbf{u}\|\overline{r}\|_\infty$ and set $\overline{r}_i = \gamma$.*
2. *Calculate $v \in \mathbb{R}^n$ such that $v \approx \langle A\rangle^{-1}\overline{r}$.*
3. *Check whether $v > \mathbf{0}$. If not, then the algorithm ends in failure.*
4. *Calculate $w \in \mathbb{R}^n$ satisfying $w \leq \langle A\rangle v$ (in higher precision arithmetic if possible).*
5. *Check whether $w > \mathbf{0}$. If not, then the algorithm ends in failure.*
6. *Calculate $\alpha \in \mathbb{R}$ satisfying $\overline{r} \leq \alpha w$.*
7. *Calculate $d \in \mathbb{R}^n$ satisfying $d \geq \alpha v$, and the algorithm ends.*

Algorithm 5.2 *For given $A \in \mathbb{R}^{n\times n}$ and b and $\tilde{x} \in \mathbb{R}^n$, the following algorithm calculates an upper bound $d \in \mathbb{R}^n$ of $|A^{-1}b - \tilde{x}|$ while verifying that A is an H-matrix, which is based on* (21) *in Theorem 5.2.*

1. *Calculate $r \in \mathbb{R}^n$ such that $r \approx b - A\tilde{x}$ (in higher precision arithmetic if possible).*
2. *Calculate $\tilde{z} \in \mathbb{R}^n$ such that $\tilde{z} \approx A^{-1}r$ by some Jacobi or Gauss-Seidel iterations.*
3. *Calculate an upper bound $\overline{s} \in \mathbb{R}^n$ of $|b - A(\tilde{x} + \tilde{z})|$ (in higher precision arithmetic if possible). If $\overline{s}_i = 0$ for some i, then compute $\gamma = \mathbf{u}\|\overline{s}\|_\infty$ and set $\overline{s}_i = \gamma$.*
4. *Calculate $v \in \mathbb{R}^n$ such that $v \approx \langle A\rangle^{-1}\overline{s}$.*
5. *Check whether $v > \mathbf{0}$. If not, then the algorithm ends in failure.*
6. *Calculate $w \in \mathbb{R}^n$ satisfying $w \leq \langle A\rangle v$ (in higher precision arithmetic if possible).*
7. *Check whether $w > \mathbf{0}$. If not, then the algorithm ends in failure.*
8. *Calculate $\beta \in \mathbb{R}$ satisfying $\overline{s} \leq \beta w$.*
9. *Calculate $d \in \mathbb{R}^n$ satisfying $d \geq |\tilde{z}| + \beta v$, and the algorithm ends.*

Table 1: Test matrices.

Problem	n	nnz	cond	Class
Bourchtein/atmosmodd	1,270,432	8,814,880	$9.02 \cdot 10^3$	SDD/H
Bourchtein/atmosmodl	1,489,752	10,319,760	$1.47 \cdot 10^3$	SDD/H
Hamm/memplus	17,758	99,147	$1.29 \cdot 10^5$	H
HB/1138_bus	1,138	4,054	$8.57 \cdot 10^6$	M
HB/sherman3	5,005	20,033	$5.01 \cdot 10^{17}$	SDD/M
Sandia/ASIC_100ks	99,190	578,890	$9.30 \cdot 10^9$	SDD/H
Simon/raefsky5	6,316	167,178	$3.87 \cdot 10^{14}$	H
Simon/raefsky6	3,402	130,371	$1.41 \cdot 10^{16}$	H
Wang/wang3	26,064	177,168	$6.18 \cdot 10^3$	M
Wang/wang4	26,068	177,196	$4.02 \cdot 10^5$	M
Random10K_10	10,000	109,948	$2.05 \cdot 10^4$	H
Random10K_20	10,000	209,767	$7.58 \cdot 10^4$	H
Random100K_10	100,000	1,099,926	n/a	H
Random100K_20	100,000	2,099,772	n/a	H
Random1M_10	1,000,000	10,999,949	n/a	H
Random1M_20	1,000,000	20,999,785	n/a	H

nnz: the number of nonzero elements, cond: condition number
n/a: not available due to memory limitations
SDD: strictly diagonally dominant, M: M-matrix, H: H-matrix

Remark 3 *For the step 1 in Algorithm 5.1 and the steps 1 and 3 in Algorithm 5.2, higher precision arithmetic is effective for calculating the residual vectors, since such computations often cause heavy cancellations in floating-point arithmetic. Moreover, for the step 4 in Algorithm 5.1 and the step 6 in Algorithm 5.2, higher precision arithmetic is useful for calculating a tight lower bound w of $\langle A \rangle v$ to reduce the effect of rounding errors if $\langle A \rangle$ is ill-conditioned.*

6 Numerical Results

The numerical experiments were carried out using MATLAB R2012b and INTLAB Version 6 [14] on a PC with 2.8 GHz Intel Core i7 CPU and 16 GB of main memory. We took test problems from University of Florida Sparse Matrix Collection [3]. Moreover, we generated sparse random H-matrices based on the MATLAB function `sprandn`. Table 1 displays the list of test matrices with their properties. Note that the condition numbers (cond in the table) are taken from the web site of [3] if available. Otherwise, the MATLAB function `condest` is used. For large test matrices, `condest` is not available due to memory limitations.

We first discuss the numerical behavior of the algorithms for verifying the M-property that are introduced in Section 3. The methods tested are summarized in Table 2. Table 3 displays the computing time for verifying whether the comparison matrix of a test matrix is an M-matrix. Note that the cases of test matrices being strictly diagonally dominant (labeled as "SDD") are omitted because it is obvious

Table 2: Methods for the verification of the M-property.

(i)	Solve $\langle A\rangle y = e$ using PCG method if a test matrix $\langle A\rangle$ is symmetric, otherwise BiCG method (stopping criterion: $\|e - \langle A\rangle\tilde{y}\|_\infty \leq 10^{-3}$).
(ii)	Compute an approx. Perron vector v of $D^{-1}E$ using power method where D and E correspond to diagonal and off-diagonal parts of $\langle A\rangle$, respectively (stopping criterion: $D^{-1}Ev < v$).
(iii)	Compute an approx. Perron vector of $B := \mu I - \langle A\rangle$ using power method where $\mu := \max_i \lvert a_{ii}\rvert$ (stopping criterion: $Bv < \mu v$).
(iv)	Use Algorithm 1 in [7] by L. Li ($\varepsilon = 0.001$).

Table 3: Computing time (sec.) and the number of iterations (inside the parentheses) for the verification of the M-property.

Problem	(i)	(ii)	(iii)	(iv)
Hamm/memplus	0.56	6.52 (11482)	0.12 (194)	0.12 (59)
HB/1138_bus	0.03	Failed	Failed	Failed
Simon/raefsky5	0.03	Failed	Failed	0.04 (10)
Simon/raefsky6	0.03	Failed	Failed	0.03 (8)
Wang/wang3	0.30	0.26 (333)	0.70 (944)	0.59 (237)
Wang/wang4	0.27	0.31 (400)	5.84 (7644)	0.96 (394)
Random10K_10	0.07	0.03 (12)	Failed	0.04 (12)
Random10K_20	0.09	0.05 (9)	Failed	0.05 (9)
Random100K_10	0.80	0.31 (14)	Failed	0.40 (14)
Random100K_20	1.33	0.51 (12)	Failed	0.59 (12)
Random1M_10	18.39	5.43 (15)	Failed	7.70 (24)
Random1M_20	32.57	9.49 (14)	Failed	8.73 (14)

that their comparison matrices are M-matrices. The results of Methods (ii), (iii), and (iv) are labeled as "Failed" in the table when the number of iterations exceeds the dimension of a test matrix.

From Table 3, it can be seen that Methods (ii) and (iii) are less robust than the others. The weakness of Methods (ii) and (iii) comes from the fact that the power method is sensitive to the distance between the first and the second largest magnitude eigenvalues. In contrast, Method (i) worked in all the cases and seems to work better than Method (iv). It turns out that Method (i) is the most stable among the methods tested, while it is also slower than Methods (ii) and (iv) in many cases.

We now move on to the numerical behavior of the algorithms for verified solutions of linear systems that are presented in Section 5. We compare the algorithms summarized in Table 4. If a test matrix A is symmetric, we adopt the MATLAB function `pcg` as a preconditioned conjugate gradient method for solving linear systems $Ax = b$ and $\langle A\rangle v = t$, where $t := \overline{r} \geq |b - A\tilde{x}|$ in Algorithm I, $t := \overline{s} \geq |b - A(\tilde{x} + \tilde{z})|$ in Algorithms II and III, and $t := e$ in Algorithm IV. We use the MATLAB commands:

```
L = ichol(A,struct('type','ict','droptol',1e-3));
```

Table 4: Algorithms for verified solutions of linear systems.

I	Algorithm 5.1: $\epsilon_x = 10^{-10}$, $\epsilon_v = 10^{-10}$
II	Algorithm 5.2: $\epsilon_x = 10^{-10}$, $\epsilon_v = 10^{-10}$, $iter = 30$
III	Algorithm 5.2: $\epsilon_x = 10^{-10}$, $\epsilon_v = 10^{-6}$, $iter = 30$
IV	Use (17) in Corollary 5.2: $\epsilon_x = 10^{-10}$, $\epsilon_v = 10^{-10}$, $iter = 30$

Table 5: Median of relative error bounds of approximate solutions to linear systems.

Problem	I	II	III	IV
Bourchtein/atmosmodd	$1.51 \cdot 10^{-11}$	$4.12 \cdot 10^{-12}$	$4.17 \cdot 10^{-12}$	$2.59 \cdot 10^{-6}$
Bourchtein/atmosmodl	$5.69 \cdot 10^{-11}$	$2.15 \cdot 10^{-11}$	$2.16 \cdot 10^{-11}$	$6.90 \cdot 10^{-6}$
Hamm/memplus	$2.21 \cdot 10^{-11}$	$1.74 \cdot 10^{-11}$	$1.76 \cdot 10^{-11}$	$5.61 \cdot 10^{-7}$
HB/1138_bus	$9.56 \cdot 10^{-11}$	$8.24 \cdot 10^{-11}$	$8.24 \cdot 10^{-11}$	$8.40 \cdot 10^{-8}$
HB/sherman3	$1.74 \cdot 10^{-5}$	$1.48 \cdot 10^{-6}$	$1.48 \cdot 10^{-6}$	$3.54 \cdot 10^{-3}$
Sandia/ASIC_100ks	$2.25 \cdot 10^{-11}$	$9.48 \cdot 10^{-12}$	$9.48 \cdot 10^{-12}$	$3.21 \cdot 10^{-8}$
Simon/raefsky5	$5.07 \cdot 10^{-15}$	$4.39 \cdot 10^{-15}$	$4.42 \cdot 10^{-15}$	$2.62 \cdot 10^{-11}$
Simon/raefsky6	$4.67 \cdot 10^{-15}$	$4.00 \cdot 10^{-15}$	$4.00 \cdot 10^{-15}$	$6.76 \cdot 10^{-10}$
Wang/wang3	$4.44 \cdot 10^{-11}$	$7.20 \cdot 10^{-12}$	$7.21 \cdot 10^{-12}$	$3.44 \cdot 10^{-8}$
Wang/wang4	$3.79 \cdot 10^{-13}$	$8.03 \cdot 10^{-14}$	$8.03 \cdot 10^{-13}$	$6.71 \cdot 10^{-9}$
Random10K_10	$1.96 \cdot 10^{-7}$	$6.16 \cdot 10^{-11}$	$6.16 \cdot 10^{-11}$	$5.83 \cdot 10^{-9}$
Random10K_20	$4.15 \cdot 10^{-8}$	$1.33 \cdot 10^{-11}$	$1.33 \cdot 10^{-11}$	$9.23 \cdot 10^{-9}$
Random100K_10	$1.51 \cdot 10^{-8}$	$9.69 \cdot 10^{-12}$	$9.69 \cdot 10^{-12}$	$2.38 \cdot 10^{-7}$
Random100K_20	$2.97 \cdot 10^{-6}$	$2.15 \cdot 10^{-10}$	$2.15 \cdot 10^{-10}$	$2.93 \cdot 10^{-6}$
Random1M_10	$6.36 \cdot 10^{-7}$	$4.46 \cdot 10^{-11}$	$4.46 \cdot 10^{-11}$	$4.22 \cdot 10^{-6}$
Random1M_20	$2.51 \cdot 10^{-5}$	$2.15 \cdot 10^{-9}$	$2.15 \cdot 10^{-9}$	$3.15 \cdot 10^{-4}$

```
[x,flag,relres,iter] = pcg(A,b,1e-10,1000,L,L');
```

Otherwise, we use the MATLAB function `bicg` as a preconditioned biconjugate gradient method for solving the linear systems. We use the following MATLAB commands:

```
[L,U] = ilu(A);
[x,flag,relres,iter] = bicg(A,b,1e-10,1000,L,U);
```

We set a stopping criterion of the iterations for solving $Ax = b$ as $\|b - A\tilde{x}\|/\|b\| \leq \epsilon_x$, where $\epsilon_x := 10^{-10}$ in all the algorithms. Moreover, we also set a stopping criterion of the iterations for solving $\langle A \rangle v = t$ as $\|t - \langle A \rangle \tilde{v}\|/\|t\| \leq \epsilon_v$, where $\epsilon_v := 10^{-10}$ in Algorithms I, II and IV, and $\epsilon_v := 10^{-6}$ in Algorithm III. In Algorithms II, III and IV, we fix the number of Jacobi iterations as $iter := 30$ for solving $Az = r$ with $r := b - A\tilde{x}$. Table 5 displays the median of the relative error bound obtained by the algorithms in Table 4. Table 6 displays the computing time for calculating $\tilde{x}$ by solving $Ax = b$ and that for verifying its error bound.

From Table 5, it can be seen that Algorithms II and III give more tight error bounds than the others, as expected. Although the quality of the results by Algorithms II and III is almost the same, Algorithm III is considerably faster than Algorithm II, which can

Table 6: Computing time (sec.) for verified solutions of linear systems.

Problem	Solve $Ax = b$	I	II	III	IV
Bourchtein/atmosmodd	80.48	76.40	77.70	49.54	84.47
Bourchtein/atmosmodl	54.73	57.21	60.25	38.33	61.36
Hamm/memplus	1.65	1.65	1.67	1.05	3.83
HB/1138_bus	0.03	0.01	0.02	0.01	0.02
HB/sherman3	0.13	0.15	0.15	0.13	0.13
Sandia/ASIC_100ks	0.42	0.47	0.58	0.31	0.67
Simon/raefsky5	0.03	0.03	0.05	0.05	0.07
Simon/raefsky6	0.04	0.03	0.05	0.04	0.06
Wang/wang3	0.55	0.55	0.59	0.43	0.56
Wang/wang4	0.47	0.43	0.45	0.36	0.53
Random10K_10	0.08	0.10	0.11	0.10	0.14
Random10K_20	0.12	0.14	0.16	0.13	0.18
Random100K_10	0.99	1.21	1.54	1.12	1.64
Random100K_20	1.36	1.96	2.38	1.84	2.44
Random1M_10	20.70	28.53	32.51	26.45	37.00
Random1M_20	31.69	59.08	62.39	44.75	73.31

be confirmed from Table 6. Algorithm III is faster because when solving $\langle A \rangle v = t$, the accuracy of its approximate solution $\tilde{v}$ in Algorithm III is relatively worse than that in Algorithm II due to the difference between the stopping criteria of the iterations. However, it does not strongly affect the quality of a final error bound of an approximate (but sufficiently accurate) solution $\tilde{x}$ of $Ax = b$, because $\tilde{v}$ affects the second term of (21), which is expected to be less than the first term $|\tilde{z}|$. In fact, by the Jacobi iterations in Algorithms II and III, the magnitude of the residual $|b - A(\tilde{x} + \tilde{z})|$ decreases to a certain extent. Therefore, the difference of the quality of $\tilde{v}$ is successfully hidden while reducing the computational cost in Algorithm 5.2.

Acknowledgements

The authors would like to express their sincere thanks to Prof. Siegfried M. Rump from Hamburg University of Technology for his stimulating discussions. Moreover, the authors wish to thank the two anonymous referees for their valuable comments. This research was partially supported by CREST, Japan Science and Technology Agency (JST).

References

[1] A. Berman and R. J. Plemmons. *Nonnegative Matrices in the Mathematical Sciences*, volume 9 of *Classics Appl. Math.* SIAM, Philadelphia, 1994.

[2] L. Cvetkovic and V. Kostic. Application of generalized diagonal dominance in wireless sensor network optimization problems. *Appl. Math. Comput.*, 218(9):4798–4805, 2012.

[3] T. A. Davis and Y. Hu. The University of Florida Sparse Matrix Collection. *ACM Trans. Math. Software*, 38(1):1:1–25, 2011. `http://www.cise.ufl.edu/research/sparse/matrices`.

[4] M. Fiedler and V. Pták. On matrices with non-positive off-diagonal elements and positive principal minors. *Czech. Math. J.*, 12(3):382–400, 1962.

[5] A. Hadjidimos. An extended compact profile iterative method criterion for sparse H-matrix. *Linear Alg. Appl.*, 389:329–345, 2004.

[6] T. Kohno, H. Niki, H. Sawami, and Y. Gao. An iterative test for H-matrix. *J. Comp. Appl. Math.*, 115(1–2):349–355, 2000.

[7] L. Li. On the iterative criterion for generalized diagonally dominant matrices. *SIAM J. Matrix Anal. Appl.*, 24(1):17–24, 2002.

[8] A. Neumaier. *Interval Methods for Systems of Equations*. Encyclopedia of Mathematics and its Applications. Cambridge University Press, Cambridge, 1990.

[9] A. Neumaier. A simple derivation of the Hansen-Bliek-Rohn-Ning-Kearfott enclosure for linear interval equations. *Reliable Computing*, 5(2):131–136, 1999.

[10] A. Neumaier. Grand challenges and scientific standards in interval analysis. *Reliable Computing*, 8(4):313–320, 2002.

[11] T. Ogita, S. Oishi, and Y. Ushiro. Computation of sharp rigorous componentwise error bounds for the approximate solutions of systems of linear equations. *Reliable Computing*, 9(3):229–239, 2003.

[12] S. M. Rump. *Kleine Fehlerschranken bei Matrixproblemen*. PhD thesis, Universität Karlsruhe, 1980.

[13] S. M. Rump. Verification Methods for Dense and Sparse Systems of Equations. In J. Herzberger, editor, *Topics in Validated Computations - Studies in Computational Mathematics*, pages 63–136. Elsevier, Amsterdam, 1994.

[14] S. M. Rump. INTLAB – INTerval LABoratory. In Tibor Csendes, editor, *Developments in Reliable Computing*, pages 77–104. Kluwer Academic Publishers, Dordrecht, 1999. `http://www.ti3.tuhh.de/rump/`.

[15] S. M. Rump. Verification methods: Rigorous results using floating-point arithmetic. *Acta Numerica*, 19:287–449, 2010.

[16] S. M. Rump. Accurate solution of dense linear systems, Part II: Algorithms using directed rounding. *J. Comp. Appl. Math.*, 242:185–212, 2013.