AN ITERATIVE CRITERION FOR H-MATRICES

(Linear Algebra and Its Applications, 271:179-190, 1998)

Bishan Li
Department of Mathematics and Statistics
University of Regina, Regina, SK, Canada S4S 0A2

Lei Li
Department of Information and Systems Engineering
Aomori University, Aomori 030, Japan

Masunori Harada and Hiroshi Niki
Department of Applied Mathematics
Okayama University of Science, Okayama 700, Japan

Michael J. Tsatsomeros *
Department of Mathematics and Statistics
University of Regina, Regina, SK, Canada S4S 0A2

May 1997

Abstract

We provide an algorithmic characterization of H-matrices. When A is an H-matrix, this algorithm determines a positive diagonal matrix D such that AD is strictly row diagonally dominant. In effect, D is produced iteratively by quantifying and re-distributing the diagonal dominance present in some rows of A to the non-diagonally dominant rows.

*Work supported by a Natural Sciences and Engineering Research Council grant

1 Introduction

The H-matrices, defined below, arise in several applications of the mathematical sciences. The class of H-matrices generalizes the widely studied classes of strictly diagonally dominant matrices and of nonsingular M-matrices. In this note, we will introduce a simple algorithmic characterization of H-matrices. We first need to recall the following definitions.

Let $A = [a_{ij}] \in \mathbf{C}^{n,n}$. We call A *generalized* (row) *diagonally dominant* if there exists an entrywise positive vector $x = [x_k] \in \mathbf{C}^n$ such that

$$|a_{ii}|x_i \geq \sum_{k \neq i} |a_{ik}|x_k \quad (i \in \{1, 2, \ldots, n\}). \tag{1.1}$$

This notion generalizes the notion of (row) *diagonal dominance*, in which $x = e$ (i.e., the all ones vector). In fact, if A satisfies (1.1) and if $D = \mathrm{diag}(x)$ (i.e., the diagonal matrix whose diagonal entries are the entries of x in their natural order), it follows that AD is a diagonally dominant matrix. If the inequality in (1.1) is strict for all $i \in \{1, 2, \ldots, n\}$, then we refer to the dominance as *strict*.

We define next the *comparison matrix* of A, $\mathcal{M}(A) = [\alpha_{ij}]$, by

$$\alpha_{ij} = \begin{cases} |a_{ii}| & \text{if } i = j \\ -|a_{ij}| & \text{if } i \neq j. \end{cases}$$

If $A = \mathcal{M}(A)$ and the eigenvalues of A have positive real parts, we call A a (nonsingular) *M-matrix*. We say that A is an *H-matrix* if $\mathcal{M}(A)$ is an M-matrix. For details and numerous conditions equivalent to being an M-matrix, the reader is referred to [4] and [1].

There are two further remarks we need to make, originating from well known facts about M-matrices found in the aforementioned references. First, every H-matrix, as defined above, is nonsingular. (We caution the reader that some authors define an H-matrix by what amounts to requiring that the eigenvalues of $\mathcal{M}(A)$ have nonnegative real parts, thus allowing for singular "H-matrices".) Second, A is an H-matrix if and only if A is strictly generalized diagonally dominant. Therefore, whether a given matrix A is an H-matrix or not is equivalent to whether there exists or not a positive diagonal matrix D so that AD is strictly diagonally dominant. Let us denote the set of all such positive diagonal matrices by $\mathcal{D}_A$ so that

$$A \text{ is an H-matrix if and only if } \mathcal{D}_A \neq \emptyset.$$

Suppose for a moment that A is an H-matrix and let $B = \mathcal{M}(A)$, $x \in \mathbf{C}^n$ be an entrywise positive vector, and $y = B^{-1}x$. Then, as B^{-1} is an entrywise nonnegative matrix (see e.g., [1, Theorem 6.2.3]), y is also entrywise positive. It follows that $D_y = \mathrm{diag}(y) \in \mathcal{D}_A$. However, the computation of such a vector y can be a relatively intense numerical exercise since B^{-1} is involved.

In [2, Theorem 1], a sufficient condition is given for strict generalized diagonal dominance of $A \in \mathbf{C}^{n,n}$. The proof of that result proceeds with the construction of a matrix $D \in \mathcal{D}_A$. However, the condition in [2] is not necessary. Moreover, the construction of D depends on knowing a partition of $\{1, 2, \ldots, n\}$ for which the sufficient condition is satisfied, making the computational complexity prohibitive. Similar remarks are valid for the sufficient conditions for H-matrices presented in [6] and [3].

In view of the preceding comments, we find ourselves in pursuit of another method for computing a matrix in $\mathcal{D}_A$. Ideally, we want this method to be computationally convenient, and we also want the possible failure of the algorithm to produce a matrix in $\mathcal{D}_A$ to signify that the input matrix A is not an H-matrix. In other words, we are in pursuit of an algorithmic characterization of an H-matrix, which can be effectively implemented on a computer. The algorithm described in the following section has these features.

2 The algorithm

Henceforth, given a positive integer n we denote $\{1, 2, \ldots, n\}$ by $\langle n \rangle$. Also, given a matrix $X = [x_{ij}] \in \mathbf{C}^{n,n}$ we use the notation

$$R_i(X) = \sum_{k \neq i} |x_{ik}| \quad (i \in \langle n \rangle)$$

and

$$\mathbf{N}_1(X) = \{i \in \langle n \rangle \ : \ |x_{ii}| > R_i(X)\}, \quad \text{and} \quad \mathbf{N}_2(X) = \langle n \rangle \setminus \mathbf{N}_1(X).$$

An algorithmic approach to computing a matrix in $\mathcal{D}_A$ was proposed in [5], where the columns of the m-th iterate, $A^{(m)}$, are scaled by post-multiplication with a suitable diagonal matrix $\text{diag}(d)$. The entries of $d \in \mathbf{C}^n$ satisfy

$$d_i = \begin{cases} 1 - \epsilon & \text{if } i \in \mathbf{N}_1(A^{(m)}) \\ 1 & \text{if } i \in \mathbf{N}_2(A^{(m)}). \end{cases}$$

Assuming that $\epsilon > 0$ is sufficiently small, and that A is an H-matrix, the algorithm produces a diagonally dominant matrix. Thus the product of the intermediate diagonal matrices yields a matrix in $\mathcal{D}_A$. The main drawback of this method is that the choice of ϵ may lead to a large number of required iterations. Moreover, when it is not a priori known whether A is an H-matrix, a possible failure of the algorithm to produce a matrix in $\mathcal{D}_A$ after a large number of iterations cannot necessarily be attributed to the choice of ϵ.

We will next introduce a different algorithmic procedure for the computation of a matrix in $\mathcal{D}_A$, in which the above drawbacks are addressed.

There are two cases where A is easily seen not to be an H-matrix. First, if A has no diagonally dominant rows, then all the entries of $\mathcal{M}(A)e$ are nonpositive, violating the monotonicity condition for M-matrices (see e.g., [1, Theorem 6.2.3]). It follows that A is not an H-matrix. Second, if a diagonal entry of A is zero, then A is not an H-matrix since $\mathcal{D}_A = \emptyset$. Consequently, the algorithm below is designed to terminate (at step 1 - before any iterations take place) if either of these cases occurs. Otherwise, it quantifies the diagonal dominance in certain rows of the m-th iterate, $A^{(m)}$, by computing the ratios $R_i(A^{(m)})/|a_{ii}^{(m)}|$. Then the algorithm proceeds to re-distribute the (collective) diagonal dominance among all rows by rescaling the columns of $A^{(m)}$, thus producing $A^{(m+1)}$.

<u>`Algorithm` $\mathbb{H}$</u>

`INPUT: a matrix` $A = [a_{ij}] \in \mathbf{C}^{n,n}$ `and` <u>`any`</u> $\epsilon > 0$.

`OUTPUT:` $D = D^{(1)} D^{(2)} \ldots D^{(m)} \in \mathcal{D}_A$ `if` A `is an H-matrix.`

1. if $\mathbf{N}_1(A) = \emptyset$ or $a_{ii} = 0$ for some $i \in \langle n \rangle$, 'A is not an H-matrix', STOP; otherwise
2. set $A^{(0)} = A$, $D^{(0)} = I$, $m = 1$
3. compute $A^{(m)} = A^{(m-1)} D^{(m-1)} = [a_{ij}^{(m)}]$
4. if $\mathbf{N}_1(A^{(m)}) = \langle n \rangle$, '$A$ is an H-matrix', STOP; otherwise
5. set $d = [d_i]$, where

$$d_i = \begin{cases} 1 - \frac{|a_{ii}^{(m)}| - R_i(A^{(m)})}{|a_{ii}|^{(m)} + \epsilon} & \texttt{if } i \in \mathbf{N}_1(A^{(m)}) \\ 1 & \texttt{if } i \in \mathbf{N}_2(A^{(m)}) \end{cases}$$

6. set $D^{(m)} = \mathtt{diag}(d)$, $m = m + 1$; go to step 3

The theoretical basis for the functionality of Algorithm $\mathbb{H}$ as a criterion for H-matrices is provided by the following theorem and the two lemmata that precede its proof.

Theorem 2.1 *The matrix $A = [a_{ij}] \in \mathbf{C}^{n,n}$ is an H-matrix if and only if Algorithm $\mathbb{H}$ terminates after a finite number of iterations by producing a strictly diagonally dominant matrix.*

Lemma 2.2 *The Algorithm $\mathbb{H}$ either terminates or it produces an infinite sequence of distinct matrices $\{A^{(m)} = [a_{ij}^{(m)}]\}$ such that $\lim_{m\to\infty} |a_{ij}^{(m)}|$ exists for all $i, j \in \langle n \rangle$.*

Proof: Suppose that Algorithm $\mathbb{H}$ does not terminate, that is, it produces an infinite sequence of matrices. Recall that this means $\mathbf{N}_1(A) \neq \emptyset$ and $a_{ii} \neq 0$ for all $i \in \langle n \rangle$. For notational convenience, we can assume that $A = \mathcal{M}(A)$ and that

$$A = \begin{pmatrix} a_{11} & -a_{12} & \dots & -a_{1n} \\ -a_{21} & a_{22} & \dots & -a_{2n} \\ \dots & \dots & \dots & \dots \\ -a_{n1} & -a_{n2} & \dots & a_{nn} \end{pmatrix},$$

where $a_{ii} > 0$ and $a_{ij} \geq 0$ for all $i, j \in \langle n \rangle$. By the definition of d_i in step 5, it readily follows that for all $i \in \mathbf{N}_1(A^{(m)})$, $d_i \in (0, 1)$ and also that

$$d_i = \frac{R_i(A^{(m)}) + \epsilon}{a_{ii}^{(m)} + \epsilon}.$$

Hence, since d_i is an increasing function of $\epsilon \in [0, \infty)$, we have that for any $m = 1, 2, \dots$,

$$\begin{aligned} a_{ii}^{(m+1)} &= d_i a_{ii}^{(m)} > \frac{R_i(A^{(m)})}{a_{ii}^{(m)}} a_{ii}^{(m)} \\ &= R_i(A^{(m)}) \geq R_i(A^{(m+1)}). \end{aligned}$$

In other words, we have shown that

$$\mathbf{N}_1(A) = \mathbf{N}_1(A^{(1)}) \subseteq \mathbf{N}_1(A^{(2)}) \subseteq \dots \subseteq \mathbf{N}_1(A^{(m)}) \subseteq \dots .$$

Consequently, there exists a smallest integer ℓ such that $\mathbf{N}_1(A^{(\ell)}) = \mathbf{N}_1(A^{(\ell+p)})$ for all $p = 1, 2, \ldots$. Since Algorithm $\mathbb{H}$ terminates for the input matrix A if and only if it terminates for the input matrix $A^{(\ell)}$, we may without loss of generality assume that $\ell = 1$. Further, we may suppose that

$$\mathbf{N}_1(A) = \mathbf{N}_1(A^{(1)}) = \{1, 2, \ldots, k\} \quad \text{for some} \quad k < n$$

(otherwise we can consider a permutation similarity of A). Under this assumption, the algorithm yields

$$A^{(m+1)} = A^{(m)} D^{(m)} \quad (m = 1, 2, \ldots),$$

where

$$D^{(m)} = \text{diag}(d_m), \quad d_m = [d_1^{(m)}, d_2^{(m)}, \ldots, d_k^{(m)}, 1, 1, \ldots, 1]^T,$$

and $d_i^{(m)} \in (0, 1)$ for all $i \in \langle k \rangle$. Thus,

$$a_{st}^{(m+1)} = \begin{cases} d_t^{(m)} a_{st}^{(m)} & \text{if } s \in \langle n \rangle \text{ and } t \in \mathbf{N}_1(A^{(1)}) \\ a_{st} & \text{if } s \in \langle n \rangle \text{ and } t \in \mathbf{N}_2(A^{(1)}). \end{cases}$$

It follows that for any $s, t \in \langle n \rangle$, $\{a_{st}^{(m)}\}$ is a non-increasing and bounded sequence. Thus $\lim_{m\to\infty} a_{st}^{(m)}$ exists for all $s, t \in \langle n \rangle$. ■

Lemma 2.3 *If Algorithm* $\mathbb{H}$ *produces the infinite sequence* $\{A^{(m)} = [a_{ij}^{(m)}]\}$*, then for all* $i \in \mathbf{N}_1(A)$*,*

$$\lim_{m\to\infty} \left(|a_{ii}^{(m)}| - R_i^{(m)}(A) \right) = 0.$$

Proof: Assume that A is as in the proof of Lemma 2.2 and suppose, by way of contradiction, that for some $i \in \mathbf{N}_1(A)$, $\lim_{m\to\infty} \left(a_{ii}^{(m)} - R_i^{(m)}(A) \right) \neq 0$. Notice that $a_{ii}^{(m)} > R_i(A^{(m)})$ and recall that, from Lemma 2.2, both sequences $\{a_{ii}^{(m)}\}$ and $\{R_i(A^{(m)}\}$ converge. We can therefore conclude that there exists $\epsilon_0 > 0$ such that

$$a_{ii}^{(m)} - R_i(A^{(m)}) > \epsilon_0 \quad (m = 1, 2, \ldots). \tag{2.2}$$

In particular, $a_{ii}^{(m)} > \epsilon_0 + R_i(A^{(m)}) \geq \epsilon_0$. From Algorithm $\mathbb{H}$ we then obtain

$$\begin{aligned} 0 < a_{ii}^{(m+1)} &= d_i^{(m)} a_{ii}^{(m)} \\ &= a_{ii}^{(m)} - \frac{a_{ii}^{(m)}}{a_{ii}^{(m)} + \epsilon} \left(a_{ii}^{(m)} - R_i(A^{(m)}) \right) \\ &\leq a_{ii}^{(m)} - \frac{a_{ii}^{(m)}}{a_{ii}^{(m)} + \epsilon} \epsilon_0 \quad \text{(by (2.2))} \\ &\leq a_{ii}^{(m)} - \frac{\epsilon_0}{\epsilon_0 + \epsilon} \epsilon_0 \ = a_{ii}^{(m)} - \theta, \end{aligned}$$

where $\theta = \frac{\epsilon_0^2}{\epsilon_0 + \epsilon}$. Note that θ is positive and therefore, as

$$a_{11} \geq a_{11}^{(1)} + \theta \geq \ldots \geq a_{11}^{(m)} + m\theta \geq m\theta,$$

by letting $m \to \infty$ we obtain a contradiction. ■

We are now able to prove our main result.

Proof of Theorem 2.1:

Sufficiency: Suppose that Algorithm $\mathbb{H}$ terminates after k iterations. That is, we have obtained a strictly diagonally dominant matrix $A^{(k)} = AD$, where $D = D^{(1)}D^{(2)}\dots D^{(k-1)}$ is by construction a positive diagonal matrix. By our introductory remarks, it follows that A is an H-matrix.

Necessity: Let A be an H-matrix and assume that A is as in the proof of Lemma 2.2. Furthermore, by way of contradiction, assume that Algorithm $\mathbb{H}$ yields the infinite sequences

$$\{A^{(m)}\},\ \{a_{ii}^{(m)}\},\ \{R_i(A^{(m)})\},\ \{\mathbf{N}_1(A^{(m)})\}.$$

As in the proof of Lemma 2.2, we can without loss of generality assume that $\mathbf{N}_1(A^{(m)}) = \mathbf{N}_1(A) = \{1, 2, \dots, k\}$ for some $k < n$ and all $m = 1, 2, \dots$. Notice that

$$A^{(m+1)} = A^{(m)}D^{(m)} = AD^{(1)}D^{(2)}\dots D^{(m)} = AF^{(m)},$$

where $F^{(m)}$ is a positive diagonal matrix $\text{diag}(d_m)$ with $d_m = [f_1^{(m)}, f_2^{(m)}, \dots, f_k^{(m)}, 1, 1, \dots, 1]^T$. From Lemma 2.2, it follows that $\lim_{m\to\infty} A^{(m)}$ exists and so $\lim_{m\to\infty} F^{(m)}$ also exists. Say these limits are B and $F = \text{diag}(d)$, respectively, where $d = [f_1, f_2, \dots, f_k, 1, 1, \dots, 1]^T$. We thus have $AF = B$. Now notice that B is of the form

$$\begin{pmatrix} b_{11} & -b_{12} & \dots & -b_{1k} & -a_{1,k+1} & \dots & -a_{1n} \\ \dots & \dots & \dots & \dots & \dots & \dots & \dots \\ -b_{k1} & -b_{k2} & \dots & b_{kk} & -a_{k,k+1} & \dots & -a_{kn} \\ \dots & \dots & \dots & \dots & \dots & \dots & \dots \\ -b_{n1} & -b_{n2} & \dots & -b_{nk} & -a_{n,k+1} & \dots & a_{nn} \end{pmatrix},$$

where, by Lemma 2.3, $b_{ii} = R_i(B)$ for all $i \in \mathbf{N}_1(A)$, and $b_{ii} = a_{ii} \le R_i(B)$ for all $i \in \mathbf{N}_2(A)$. Hence $\mathbf{N}_1(B) = \emptyset$, implying that B is not an H-matrix.

Claim: $f_1 = f_2 = \dots = f_k = 0$.

Proof of claim: First, note that if all $f_i > 0$, then $B = AF$ would be an H-matrix, a contradiction. So at least one of the f_i's equals zero. Without loss of generality, assume that $f_1 = f_2 = \dots = f_p = 0$ for some $p < k$ and that $f_q > 0$ for all $q = p+1, p+2, \dots, k$ (otherwise we can consider a permutation similarity of A that symmetrically permutes the first p rows and columns of A, leaving $\mathbf{N}_1(A)$ invariant). Then $B = AF$ has the block form

$$AF = \begin{pmatrix} 0 & * \\ 0 & \tilde{A}_{n-p} \end{pmatrix} = \begin{pmatrix} 0 & * \\ 0 & B_{n-p} \end{pmatrix} = B,$$

where $\tilde{A}_{n-p}$ and B_{n-p} are $(n-p)\times(n-p)$. As $\tilde{A}_{n-p}$ is an H-matrix, so is B_{n-p}. This is a contradiction, because $b_{ii} \le R_i(B_{n-p})$ for all $i \in \langle n\rangle \setminus \langle p\rangle$. This completes the proof of the claim.

We now have that

$$AF = A\begin{pmatrix} 0 & 0 \\ 0 & I_{n-k} \end{pmatrix} = \begin{pmatrix} 0 & * \\ 0 & \tilde{A}_{n-k} \end{pmatrix} = B = \begin{pmatrix} 0 & * \\ 0 & B_{n-k} \end{pmatrix}.$$

Once again, we have a contradiction because $\tilde{A}_{n-k}$ is an H-matrix but B_{n-k} is not. This shows that Algorithm $\mathbb{H}$ must terminate after a finite number of iterations, completing the proof of the theorem. ■

3 Comments and a MATLAB function

We begin by noticing that the proofs of Lemma 2.2, Lemma 2.3, and Theorem 2.1 can be easily adapted to the case where the parameter ϵ is not constant throughout the iterations, but instead it is independently chosen at the m-th iteration of Algorithm $\mathbb{H}$ to form a bounded sequence $\{\epsilon^{(m)}\}$.

It is clear from the definition of Algorithm $\mathbb{H}$ and Theorem 2.1 that the termination or not of Algorithm $\mathbb{H}$ is irrespective of the choice of the positive parameter ϵ, or of the bounded sequence $\{\epsilon^{(m)}\}$ as remarked in the previous paragraph. However, the column scalings and the re-distribution of the diagonal dominance at each iteration is done according to the ratios

$$\frac{R_i(A^{(m)}) + \epsilon^{(m)}}{|a_{ii}^{(m)}| + \epsilon^{(m)}}.$$

Also, for $0 < b < a$, $(b+\epsilon)/(a+\epsilon)$ is an increasing function of $\epsilon > 0$. Hence, smaller choices of the parameter $\epsilon^{(m)}$ result in at least as large a set $\mathbf{N}_1(A^{(m+1)})$. Nevertheless, it is <u>not</u> generally true that by choosing $\epsilon^{(m)}$ small enough the number of further iterations required for the termination of the algorithm is 1, even if A is an H-matrix. To see this formally, let $A \in \mathbf{C}^{n,n}$ be an H-matrix and suppose that $\ell \in \mathbf{N}_2(A^{(m)})$ for some positive integer m. Observe then that

$$R_\ell(A^{(m+1)}) = \sum_{k \in \mathbf{N}_1(A^{(m)})} \frac{R_k(A^{(m)}) + \epsilon}{|a_{kk}^{(m)}| + \epsilon} \, |a_{\ell k}^{(m)}| + \sum_{k \in \mathbf{N}_2(A^{(m)}),\ k \neq \ell} |a_{\ell k}^{(m)}|.$$

So, if the entries of $A^{(m)}$ satisfy

$$\sum_{k \in \mathbf{N}_1(A^{(m)})} \frac{R_k(A^{(m)})}{|a_{kk}^{(m)}|} \, |a_{\ell k}^{(m)}| + \sum_{k \in \mathbf{N}_2(A^{(m)}),\ k \neq \ell} |a_{\ell k}^{(m)}| > |a_{\ell\ell}^{(m)}|,$$

then at least 2 more iterations of Algorithm $\mathbb{H}$ are required, regardless of the choice of $\epsilon > 0$. We illustrate this situation with the following example.

Example 3.1 Consider the H-matrix

$$A = \begin{pmatrix} 4 & 1 & -1 \\ -1 & 3 & 1 \\ 1 & 1 & -1 \end{pmatrix}$$

and notice that $\mathbf{N}_1(A) = \{1,2\}$, $\mathbf{N}_2(A) = \{3\}$. As

$$\lim_{\epsilon \to 0+} \sum_{k \in \mathbf{N}_1(A)} \frac{R_k(A) + \epsilon}{|a_{kk}| + \epsilon} \, |a_{3k}| = \lim_{\epsilon \to 0+} \left(\frac{2+\epsilon}{4+\epsilon} + \frac{2+\epsilon}{3+\epsilon} \right) > 1 = |a_{33}|,$$

it follows that a first pass of Algorithm $\mathbb{H}$ will not result in a strictly diagonally dominant third row. That is, at least 2 iterations are needed for the algorithm to terminate by producing $D \in \mathcal{D}_A$, regardless of the choice of $\epsilon > 0$. In fact, for $\epsilon = 0.1$ exactly 2 iterations are needed.

Let us now consider some practical aspects of choosing the parameters $\epsilon^{(m)}$. As is done for example in MATLAB, let E denote the floating point relative accuracy, namely, the distance from 1.0 to the next largest floating point number ($\mathrm{E} = 2^{-52} = 2.2204e - 16$ on machines with IEEE floating point arithmetic). Choosing $\epsilon^{(m)}$ optimally does not mean that we should choose $\epsilon^{(m)} = \mathrm{E}$, because we must ensure that when a diagonal entry is scaled down, the corresponding row remains strictly diagonally dominant within the machine's working precision. Instead, we must choose $\epsilon^{(m)}$ so that for all $k \in \mathbf{N}_1(A^{(m)})$,

$$\frac{R_k(A^{(m)}) + \epsilon^{(m)}}{|a_{kk}^{(m)}| + \epsilon^{(m)}}\ |a_{kk}^{(m)}|\ -\ R_k(A^{(m)})\ >\ \mathrm{E},$$

or equivalently,

$$\epsilon^{(m)}\ >\ \frac{\mathrm{E}\ (|a_{kk}^{(m)}|)}{|a_{kk}^{(m)}| - R_k(A^{(m)}) - \mathrm{E}}. \tag{3.3}$$

For example, we may choose "optimal" parameters $\epsilon^{(m)}$ by

$$\epsilon^{(m)}\ =\ \max_{k \in \mathbf{N}_1(A^{(m)})} \left(\frac{\mathrm{E}\ (|a_{kk}^{(m)}| + 1)}{|a_{kk}^{(m)}| - R_k(A^{(m)})} \right) \tag{3.4}$$

so that (3.3) is satisfied.

The next practical aspect of Algorithm $\mathbb{H}$ we want to discuss is the situation when the input matrix $A \in \mathbf{C}^{n,n}$ is not (known to be) an H-matrix. When the computed diagonal matrix $D^{(m)}$ is approximately equal to the identity (and the algorithm has not terminated), it means that the present iterate is not diagonally dominant and there is little numerical hope that it will become one. Based on Theorem 2.1, we can then stop and declare that A is not an H-matrix.

We also comment that Algorithm $\mathbb{H}$ can be modified so that step 6 takes place every time an $i \in \mathbf{N}_1(A^{(m)})$ is encountered; then it proceeds by searching for the first index in $\mathbf{N}_1(A^{(m+1)})$. This usually results in fewer iterations until a matrix $D \in \mathcal{D}_A$ is found.

Finally, we provide a MATLAB function implementing Algorithm $\mathbb{H}$ with a fixed parameter ϵ. The termination criteria regarding the computation of a $D \in \mathcal{D}_A$ or the decision that A is not an H-matrix are handled by the default relative accuracy of MATLAB.

```
function [diagonal,m] = hmat(a, epsilon, maxit)
% INPUT: a=square matrix, epsilon=parameter of re-distribution
% maxit=maximum number of iterations allowed

% OUTPUT: m=number of iterations performed,
% diagonal=diagonal matrix d so that ad is strictly diag.  dominant
%          =[ ] if a is not an H-matrix)

n= size(a,1); diagonal=eye(n); m=1; one=ones(1,n); stoppage=0;
if (nargin==1); epsilon=.001; maxit=100; end
if (nargin==2) maxit=100; end

if (1-all(diag(a)))
  stoppage=1; diagonal=[ ]; m=m-1; 'Input is NOT an H-matrix',
end
```

```
while (stoppage==0 & m<maxit+1)
  for i=1:n
   r(i)=sum(abs(a(i,1:n)))-abs(a(i,i));
   if (abs(a(i,i))>r(i))
    d(i)=(r(i)+epsilon)/(abs(a(i,i))+epsilon);
   else
    d(i)=1;
   end
  end
  if (d==one)
   stoppage=1; diagonal=[ ]; 'Input is NOT an H-matrix',
  elseif (d<one)
   stoppage=1; 'Input IS an H-matrix',
  else
   for i=1:n
    diagonal(i,i)=diagonal(i,i)*d(i);
   end
   a=a*diag(d); m=m+1;
  end
end
if (m==maxit+1 & stoppage==0)
  diagonal=[ ]; m=m-1;
  'Inconclusive:  decrease "epsilon" or increase "maxit"',
end
```

References

[1] Abraham Berman and Robert J. Plemmons, *Nonnegative Matrices in the Mathematical Sciences*, Classics in Applied Mathematics, SIAM, 1994.

[2] Yi-ming Gao and Xiao-hui Wang, Criteria for Generalized Diagonally Dominant Matrices and M-matrices, *Linear Algebra and Its Applications*, 169:257-268 (1992).

[3] Yi-ming Gao and Xiao-hui Wang, Criteria for Generalized Diagonally Dominant Matrices and M-matrices II, *Linear Algebra and Its Applications*, 248:339-353 (1996).

[4] Roger A. Horn and Charles R. Johnson, *Topics in Matrix Analysis*, Cambridge Univ. Press, 1991.

[5] M. Harada, H. Niki, M. Usui, An extension of the criteria for generalized diagonally dominant matrices (II), *Proceedings of the Annual Symposium of the Japan SIAM*, pp. 280-281 (1995) (in Japanese).

[6] Huang Tin-Zhu. A Note on Generalized Diagonally Dominant Matrices. *Linear Algebra and Its Applications*, 225:237-242 (1995).