English Series | NUMERICAL MATHEMATICS A Journal of Chinese Universities | Vol.14, No.1 Feb. 2005

A DIRECT ALGORITHM FOR DISTINGUISHING NONSINGULAR M-MATRIX AND H-MATRIX*

Li Yaotang(李耀堂)   Zhu Yan(朱　艳)

Abstract *A direct algorithm is proposed by which one can distinguish whether a matrix is an M-matrix (or H-matrix) or not quickly and effectively. Numerical examples show that it is effective and convincible to distinguish M-matrix (or H-matrix) by using the algorithm.*

Key words *nonsingular M-matrix, nonsingular H-matrix, direct algorithm.*

AMS(2000)subject classifications 15A48

1 Introduction

For many kinds of applications of M-matrices and H-matrices, the problem how to determine whether a matrix is an M-matrix (or H-matrix) or not arouses many researchers interesting. Recently, some iterative methods have been proposed for distinguishing H-matrices (see [2-5]). However, these methods have a common drawback, that is, it is not possible to determine the number of steps of iteration, and when A is not an H-matrix, a wasteful computation is necessary. A direct algorithm has been proposed in [6], but it is only useful when matrices are symmetrical. In this paper, to conquer these drawbacks, we propose a new direct algorithm.

2 A direct algorithm for distinguishing M-matrix

Let $R^{n\times n}$ denote the set of all $n \times n$ real matrices. $A = (a_{ij}) \in R^{n\times n}$ is said to be an M-matrix if $a_{ij} \leq 0$, for $i \neq j$, and $A^{-1} \geq 0$.

Lemma 1[1] Let $A = (a_{ij}) \in R^{n\times n}$ be an M-matrix, then any principle submatrix of A is an M-matrix.

Lemma 2[1] Let $A = (a_{ij}) \in R^{n\times n}$, its off-diagonal entries are all non-positive, then A is

* Foundation item: This work is supported by the Science Foundations of the Education Department of Yunnan Province (03Z169A) and the Science Foundatons of Yunnan University (2003Z013B).
Received: Sep. 11, 2004.

an M-matrix if and only if successive principle minor of $A, D_K > 0, k = 1, \cdots, n$.

From Lemma 2, we can immediately obtain the following lemma.

Lemma 3 Let $A = (a_{ij}) \in R^{2\times 2}$, and $a_{ij} \leq 0, i \neq j, a_{ii} > 0$, then A is an M-matrix if and only if determinant of A, det $A > 0$.

Theorem 1 Let

$$B = \begin{bmatrix} B_{11} & B_{12} \\ B_{21} & B_{22} \end{bmatrix} \in R^{n\times n},$$

where $B_{12} \leq 0, B_{21} \leq 0, B_{11}$ is a 2×2 square matrix and B_{22} is an $(n-2)\times(n-2)$ square matrix, in which their diagonal entries are all positive and off-diagonal entries are all non-positive. Then B is an M-matrix if and only if $\det B_{11} > 0$ and $B_{22} - B_{21}B_{11}^{-1}B_{12}$ is an M-matrix.

Proof Necessity: Suppose B is an M-matrix, then

$$B^{-1} = \begin{bmatrix} B_{11}^{-1} + B_{11}^{-1}B_{12}(B_{22} - B_{21}B_{11}^{-1}B_{12})^{-1}B_{21}B_{11}^{-1} & -B_{11}^{-1}B_{12}(B_{22} - B_{21}B_{11}^{-1}B_{12})^{-1} \\ -(B_{22} - B_{21}B_{11}^{-1}B_{12})^{-1}B_{21}B_{11}^{-1} & (B_{22} - B_{21}B_{11}^{-1}B_{12})^{-1} \end{bmatrix} \geq 0,$$

and B_{11} and B_{22} are M-matrices by Lemma 1. Hence, $\det B_{11} > 0$ by Lemma 3, and $B_{11}^{-1} \geq 0, (B_{22} - B_{21}B_{11}^{-1}B_{12})^{-1} \geq 0$. For $B_{12} \leq 0, B_{21} \leq 0$, we have $B_{21}B_{11}^{-1}B_{12} \geq 0$, and off-diagonal entries of matrix $B_{22} - B_{21}B_{11}^{-1}B_{12}$ are all non-positive. So, $B_{22} - B_2B_{11}^{-1}B_{12}$ is an M-matrix.

Sufficiency: Suppose $\det B_{11} > 0$ and $B_{22} - B_{21}B_{11}^{-1}B_{12}$ is an M-matrix, then by Lemma 3, we have that B_{11} is an M-matrix, so

$$(B_{22} - B_{21}B_{11}^{-1}B_{12})^{-1} \geq 0, \qquad B_{11}^{-1} \geq 0.$$

Therefore

$$\begin{aligned} &B_{11}^{-1} + B_{11}^{-1}B_{12}(B_{22} - B_{21}B_{11}^{-1}B_{12})^{-1}B_{21}B_{11}^{-1} \geq 0, \\ &-(B_{22} - B_{21}B_{11}^{-1}B_{12})^{-1}B_{21}B_{11}^{-1} \geq 0, \\ &-B_{11}^{-1}B_{12}(B_{22} - B_{21}B_{11}^{-1}B_{12})^{-1} \geq 0. \end{aligned}$$

From these inequalities, we have

$$B^{-1} = \begin{bmatrix} B_{11}^{-1} + B_{11}^{-1}B_{12}(B_{22} - B_{21}B_{11}^{-1}B_{12})^{-1}B_{21}B_{11}^{-1} & -B_{11}^{-1}B_{12}(B_{22} - B_{21}B_{11}^{-1}B_{12})^{-1} \\ -(B_{22} - B_{21}B_{11}^{-1}B_{12})^{-1}B_{21}B_{11}^{-1} & (B_{22} - B_{21}B_{11}^{-1}B_{12})^{-1} \end{bmatrix} \geq 0.$$

Thus B is an M-matrix.

From Theorem 1, we propose the following algorithm A.

Algorithm A

Input The given matrix $B = (b_{ij}) \in R^{n\times n}$.

Step 1 Set $B = B^{(m)}$, and $m = 0$.

Step 2 Partition $B^{(m)}$ into a 2×2 block matrix

$$B^{(m)} = \begin{bmatrix} B_{11}^{(m)} & B_{12}^{(m)} \\ B_{21}^{(m)} & B_{22}^{(m)} \end{bmatrix},$$

where $B_{11}^{(m)}$ is a 2×2 square matrix, and $B_{22}^{(m)}$ is an $(n-2) \times (n-2)$ square matrix.

Step 3 Set $B^{(m+1)} = B_{22}^{(m)} - B_{21}^{(m)}(B_{11}^{(m)})^{-1}B_{12}^{(m)}$. If $\det B_{11}^{(m)} \le 0$ or $B^{(m+1)}$ satisfies that diagonal entries of $B^{(m+1)}$ are not all positive or off-diagonal entries of $B^{(m+1)}$ are not all non-positive, then from Theorem 1, B is not an M-matrix, stop and output "B is not an M-matrix"; otherwise.

Step 4 Set $m = m+1$, when $m < \left[\dfrac{n+1}{2}\right] - 2$, go to step 2; otherwise.

Step 5 $B^{(m+1)}$ is a real numbers or a 2×2 square matrix, if $B^{(m+1)} > 0$ or $\det B^{(m+1)} > 0$, then B is an M-matrix, stop and output "B is an M-matrix"; if $B^{(m+1)} \le 0$ or $\det B^{(m+1)} \le 0$, then B is not an M-matrix, stop and output "B is not an M-matrix".

Note: it is estimated that the total cost of the algorithm is $O\left(\dfrac{n^3}{3}\right)$.

3 The use of Algorithm A for distinguishing H-matrix

Let $C^{n\times n}$ denote the set of all $n \times n$ complex matrices, $A = (a_{ij}) \in C^{n\times n}$, the comparison matrix $\mathcal{M}(A) = (b_{ij})$,

$$b_{ij} = \begin{cases} |a_{ij}|, i = j, \\ -|a_{ij}|, i \ne j, \end{cases} \qquad i, j = 1, 2, \cdots, n.$$

If $\mathcal{M}(A)$ is an nonsingular M-matrix, then A is called an H-matrix.

From the definition of H-matrix, we can distinguish whether a given matrix is an H-matrix or not by using Algorithm A on the comparison matrix $\mathcal{M}(A)$.

4 Numerical example

Example 1 First we consider the matrix

$$B = \begin{bmatrix} 1 & -\frac{1}{2} & 0 & -1 \\ -\frac{1}{2} & 1 & -\frac{1}{3} & 0 \\ 0 & -\frac{1}{3} & 1 & -\frac{1}{4} \\ -1 & 0 & -\frac{1}{4} & 1 \end{bmatrix}.$$

We get a 2×2 matrix

$$B^{(1)} = B_{22} - B_{21}B_{11}^{-1}B_{12} = \begin{bmatrix} \frac{23}{27} & -\frac{17}{36} \\ -\frac{17}{36} & -\frac{1}{3} \end{bmatrix}$$

by using Algorithm A. For $B^{(1)}$ has an negative diagonal entry $-\dfrac{1}{3}$, so B is not an M-matrix from Algorithm A. In fact, we know principle submatrix of $B, B(1,4) = \begin{bmatrix} 1 & -1 \\ -1 & 1 \end{bmatrix}$, $\det B(1,4) = 0$, by Lemma 2, B is really not an M-matrix.

Example 2 The second example is

$$B=\begin{bmatrix} 1 & -1 & & & & \\ -1 & 3 & -2 & & O & \\ & -1 & 4 & -3 & & \\ & & & \ddots & \ddots & \ddots \\ & O & & & & -(n-1) \\ & & & & -1 & (n+1) \end{bmatrix}.$$

When $n=10$ and $n=99$, we get real numbers $90>0$ and $9506>0$, respectively, by using Algorithm A, so B is an M-matrix from Algorithm A. In fact, we can proof that successive principal minor of $A, D_K=K!^{[1]}, k=1,2,\cdots,n$, so B is really an M-matrix.

Example 3 The third example is

$$B=\begin{bmatrix} 1 & -\frac{1}{4} & -\frac{1}{4} & 0 \\ -\frac{1}{4} & 1 & 0 & -\frac{1}{4} \\ -\frac{1}{4} & 0 & 1 & -\frac{1}{4} \\ 0 & -\frac{1}{4} & -\frac{1}{4} & 1 \end{bmatrix}.$$

We get a real number $0.8>0$ by using Algorithm A, so B is an M-matrix from Algorithm A. In fact,

$$B^{-1}=\begin{bmatrix} \frac{7}{2} & \frac{2}{6} & \frac{2}{6} & \frac{1}{6} \\ \frac{2}{6} & \frac{7}{6} & \frac{1}{6} & \frac{2}{6} \\ \frac{2}{6} & \frac{1}{6} & \frac{7}{6} & \frac{2}{6} \\ \frac{1}{6} & \frac{2}{6} & \frac{2}{6} & \frac{7}{6} \end{bmatrix}>0,$$

so B is really an M-matrix.

Of course, as to H-matrix, such as

$$B_1=\begin{bmatrix} 8 & -2 & -2 & -3 \\ -3 & 6 & -1 & -1 \\ -2 & -\frac{5}{2} & 4 & 0 \\ -4 & -2 & -1 & 6 \end{bmatrix}, \qquad B_2=\begin{bmatrix} 0.9 & 0.1 & 0.2 & 0.1 \\ 0.9 & 0.9 & 0.7 & 0.8 \\ 0.1 & 0.1 & 0.9 & 0.1 \\ 0.3 & 0.1 & 0.2 & \frac{23}{30} \end{bmatrix}.$$

We get two real numbers $3.59693>0$ and $0.11824>0$ by using Algorithm A on the comparison matrix $\mathcal{M}(B_1)$ and $\mathcal{M}(B_2)$, respectively, so these two matrices are H-matrices.

5 Program

```
#include <stdio.h>
 #include <iostream.h>
```

```
#include <math.h>
main ( )
{     int n, i, j, k = 0, h = 0, s = 1;
//m = B;
float m[150][150];
//b11 = |B11|; b = B22; c = 1/B11; b21 = B21; b12 = B12; d = B21 * B11'; e = d * B12;
float b11, b[150][150], c[3][3], b21[150][3], b12[3][150];
float d[150][3], e[150][150];
//input//input function: there you input your matrix data
cout<<"Program to judge an M-matrix:"<<endl;
cout<<"please input the rank(n) of the matrix:"<<endl;
cout<< "n = ";
cin>> n;
if(n > 149)
{cout<<"the rank(n) of the matrix should be less than 149"<<endl;
return 0;}
cout<<"please input the element of the matrix:"<<endl;
for (i = 1; i < n + 1; i++)
{   for (j = 1; j < n + 1; j++)
    {   k++;
        cout<< "m[" << i << "][" << j << "] = ";
        cin>> m[i][j];
        if (k == n)
        {   k = 0;
            cout<<endl;}}}
//the end of init//
//display the matrix data//
cout<<"the matrix you input is:"<<endl;
for (i = 1; i < n + 1; i++)
```

```
{   for (j = 1; j < n + 1; j++)
    cout<< m[i][j] <<' ';
    cout<<endl;}
//the begin of the function to judge an M-matrix //
k = 0;
int l = n;
for (l = l; l > 0; l-= 2)
{   //b11 = |B11|; b = B22; c = 1/B11; b21 = B21; b12 = B12;
    //b11 = |B11|;// and judge whether b11 is positive or not;
    b11 = m[1][1] * m[2][2] - m[1][2] * m[2][1];
    if(i == 2)
        break;
    else if(i == 1)
   {    b11 = m[1][1];
         break;}
    if (b11 < 0)
    {   k = 1;
        break;}
    c[1][1] = m[2][2]/(m[1][1] * m[2][2] - m[1][2] * m[2][1]);
    c[1][2] = m[1][2]/(m[2][1] * m[1][2] - m[1][1] * m[2][2]);
    c[2][1] = m[2][1]/(m[1][2] * m[2][1] - m[2][2] * m[1][1]);
    c[2][2] = m[1][1]/(m[2][2] * m[1][1] - m[1][2] * m[2][1]);
    //b21 = B21;
    for(i = 1; i < n - 1; i++)
   {    for(j = 1; j < 3; j++)
        {   b21[i][j] = m[i + 2][j]; }}
    //b12 = B12;
    for(i = 1; i < 3; i++)
    {   for(j = 1; j < n - 1; j++)
```

```
        {b12[i][j] = m[i][j + 2];}}
    //d = B21 * B11'
    for(i = 1; i < n - 1; i + +)
    {   for(j = 1; j < 3; j + +)
        {   d[i][j] = 0;
            for(h = 1; h < 3; h + +)
            {d[i][j]+ = b21[i][h] * c[h][j];}}}
    //q = d * B12 = B21 * B11' * B12;
    for(i = 1; i < n - 1; i + +)
    {   for(j = 1; j < n - 1; j + +)
        {   e[i][j] = 0;
            for(h = 1; h < 3; h + +)
            { e[i][j]+ = d[i][h] * b12[h][j];}}}
    //e = B22 - B21 * B11' * B12;
    for(i = 1; i < n - 1; i + +)
    {for(j = 1; j < n - 1; j + +)
        {   b[i][j]- = e[i][j];}}
  //b = B22;/ and judge whether diagonal entries are positive or not and
off-diagonal entries are non-positive or not;
for (i = 1; i < n - 1; i + +)
{ for(j = 1; j < n - 1; j + +)
    { b[i][j] = m[i + 2][j + 2];
        if (i == j)
        {   if (b[i][j] < 0)
            {   k = 1;
                break;}}
        else if(b[i][j] > 0)
        {     k = 1;
            break;}}}
```

```
//m = b; n = n - 2
n -= 2;
for(i = 1; i < n + 1; i++)
{ for(j = 1; j < n + 1; j++)
    {   m[i][j] = b[i][j];}}}
//result
if(k == 0)
{     cout<<endl<<"b11="<<b11<<endl;
      cout<<"the matrix you input is an M-matrix!"<<endl;}
else
{     cout<<endl<<"b11="<<b11<<endl;
      cout<<"the matrix you input is not an M-matrix!"<<endl;}
the end of the program
```

References

1 You Z Y. Nonsingular M-matrix. Wuhan: Huazhong Institute of Technology Press, 1981

2 Ojiro K, Niki H, Usui M. A new criterion for the H-matrix property. J. Comput. Appl. Math., 2003, 150: 293-302

3 Li L. On the iterative criterion for generalized diagonally dominant matrices. Japan Soc. Indust. Appl. Math., Tokyo, 2002,17-24

4 Li B, Li L, Harada Niki H. An iterative criterion for H-matrices. Linear Algebra and Its Applications, 1998, 271: 179-190

5 Kohno T, Niki H, Sawami H, Gao Y M. An iterative test for-matrix. J. Comput. Math., 2000,115: 345-355

6 Guo X J, Wang C W, Wang Y M, Kong L F. A method for judging nonsingular M-matrix and its parallel algorithm. Numeri.Math.J.Chinese Univ. (Chinese Series). 2001, 23(4): 357-362

Li Yaotang Department of Mathematics, University of Yunnan, Kunming 650091, PRC.

Zhu Yan Department of Mathematics, University of Yunnan, Kunming 650091, PRC.