Ma trận nghịch đảo

Mọi ý kiến đóng góp xin gửi về email: tam.doduc@hust.edu.vn

Định nghĩa. Cho ma trận A vuông cấp n. Ta nói ma trận A là khả nghịch nếu tồn tại ma trận B sao cho

$$AB = BA = E,$$

trong đó E là ma trận đơn vị cấp n. Khi đó B là ma trận nghịch đảo của ma trận A và kí hiệu là A^{-1}.

Ví dụ 1: Cho ma trận

```
A=[    -2      3      4      3
      9   -10      5     -7
      6      7      5      4
     10      9     -2    -10
]
```

```
A = 4×4
    -2     3     4     3
     9   -10     5    -7
     6     7     5     4
    10     9    -2   -10
```

Ma trận nghịch đảo của A là

```
format rat
Ainv = eye(4)/A
```

```
Ainv =
   -1363/8069       121/8069       414/3635      -126/8069
     556/8069      -298/8069         4/8069       377/8069
     247/1082       141/2452      -277/8069       117/8069
    -155/1016      -240/8069       491/4051      -617/8069
```

Kiểm tra lại

```
disp(A*Ainv)%các dấu * hiện ra do sai số tính toán
```

```
       1              *              *              *
       0              1              *              *
       *              *              1              *
       *              *              0              1
```

Tính chất.

1) Ma trận A khả nghịch khi và chỉ khi định thức của nó khác 0.

2) Cho A, B là các ma trận khả nghịch cùng cấp và $k \neq 0$ là hằng số. Khi đó AB, kA và A^{-1} là các ma trận khả nghịch và

$(AB)^{-1} = B^{-1}A^{-1}$;

$(kA)^{-1} = k^{-1}A^{-1}$;

$(A^{-1})^{-1} = A$.

Phương pháp tìm A^{-1} qua phần bù đại số

Định nghĩa 2. Phần bù đại số của phần tử a_{ij} của ma trận A là một số, ký hiệu bởi A_{ij}, và tính bởi công thức

$$A_{ij} = (-1)^{i+j}M_{ij},$$

trong đó M_{ij} là định thức cấp $n-1$ có được từ ma trận A khi bỏ đi hàng thứ i và cột thứ j.

Ví dụ 2. Tính phần bù đại số A_{13} của ma trận A

```
clc;clear all
A=[    -2      3      4      3
      9    -10      5     -7
      6      7      5      4
     10      9     -2    -10
];
disp(A)
```

```
    -2     3     4     3
     9   -10     5    -7
     6     7     5     4
    10     9    -2   -10
```

Định thức M_{13} là

```
A1 = A; A1(1,:)=[];A1(:,3)=[]; disp(A1)
```

```
     9   -10    -7
     6     7     4
    10     9   -10
```

```
M13 = det(A1)
```

```
M13 =
   -1842
```

Phần bù đại số A_{13} là

```
A13 = (-1)^(1+3)*M13
```

```
A13 =
   -1842
```

Công thức tìm A^{-1} thông qua phần bù đại số

$$A^{-1} = \frac{1}{det(A)}\mathscr{A}^T, \ (1)$$

trong đó $\mathscr{A} = (A_{ij})_{n\times n}$ là ma trận phụ hợp của ma trận A.

Ví dụ 3. Tìm ma trận nghịch đảo của ma trận A sử dụng phần bù đại số

```
clc
clear all
A=[    -2      3      4      3
      9    -10      5     -7
      6      7      5      4
     10      9     -2    -10
];
disp(A)
```

```
    -2     3     4     3
     9   -10     5    -7
     6     7     5     4
    10     9    -2   -10
```

Bước 1. Tính định thức của A

```
detA = det(A)
```

```
detA =
   -8069
```

Bước 2. Tính ma trận phụ hợp $\mathscr{A}$

```
Aph = A;
for i = 1:size(A,1)
    for j = 1:size(A,1)
        B = A([1:i-1,i+1:end],[1:j-1,j+1:end]);
        Aph(i,j)=(-1)^(i+j)*det(B);
    end
end
disp(Aph)
```

```
     1363          -556         -1842          1231
     -121           298          -464           240
     -919            -4           277          -978
      126          -377          -117           617
```

Bước 3. Tính ma trận nghịch đảo A^{-1} theo công thức (1).

```
Ainv1 = 1/detA*Aph';
disp(Ainv1)
```

```
  -1363/8069      121/8069      414/3635     -126/8069
    556/8069     -298/8069        4/8069      377/8069
    247/1082      141/2452     -277/8069      117/8069
   -155/1016     -240/8069      491/4051     -617/8069
```

```
disp(A*Ainv1)
```

```
       1             *             *             0
       *             1             *             *
       *             *             1             *
       0             *             0             1
```

Phương pháp Gauss-Jordan

Giả sử $X = (x_{ij})_{n\times n}$. Khi đó đẳng thức $AX = E$ được viết lại

$$\begin{pmatrix} a_{11} & \dots & a_{1j} & \dots & a_{1n} \\ a_{21} & \dots & a_{2j} & \dots & a_{2n} \\ a_{i1} & \dots & a_{ij} & \dots & a_{in} \\ a_{n1} & \cdots & a_{nj} & \cdots & a_{nn} \end{pmatrix} \begin{pmatrix} x_{11} & \cdots & x_{1j} & \cdots & x_{1n} \\ x_{21} & \cdots & x_{2j} & \cdots & x_{2n} \\ x_{i1} & \cdots & x_{ij} & \cdots & x_{in} \\ x_{n1} & \cdots & x_{nj} & \cdots & x_{nn} \end{pmatrix} = \begin{pmatrix} 1 & 0 & \cdots & 0 & \cdots & 0 \\ 0 & 1 & \cdots & 0 & \cdots & 0 \\ 0 & 0 & \cdots & 1 & \cdots & 0 \\ 0 & 0 & \cdots & 0 & \cdots & 1 \end{pmatrix}.(2)$$

Đặt X_j, $j = \overline{1, n}$ là cột thứ j của ma trận X

$$X_j = (x_{1j}, x_{2j}, \cdots, x_{nj})^T$$

và E_j, $j = \overline{1, n}$ là cột thứ j của ma trận E

$$E_j = (0, \cdots, 1, \cdots, 0)^T.$$

Khi đó hệ (2) tương đương với n hệ phương trình tuyến tính sau

$$AX_j = E_j, \quad j = \overline{1, n}.$$

Do $det(A) \neq 0$ nên mỗi hệ đều có nghiệm duy nhất. Do n hệ có cùng ma trận hệ số nên ta có thể giải đồng thời n hệ này bằng phương pháp Gauss-Jordan bằng cách lập ma trận mở rộng $\overline{A} = (A|E)$.

Ví dụ 4. Tìm ma trận nghịch đảo của ma trận

```
clc
clear all
A=[    -2     3     4     3
      9   -10     5    -7
      6     7     5     4
     10     9    -2   -10
];
A
```

```
A =
    -2     3     4     3
     9   -10     5    -7
     6     7     5     4
    10     9    -2   -10
```

Ma trận bổ sung $\overline{A}$ là

```
A_bs = [A,eye(size(A,1))];
disp(A_bs)
```

```
    -2     3     4     3     1     0
     9   -10     5    -7     0     1
     6     7     5     4     0     0
    10     9    -2   -10     0     0
```

```
%code cho Gauss-Jordan
tol=1e-5;
[m,n] = size(A)
```

```
m =
     4
n =
     4
```

```
% Loop over the entire matrix.
i = 1; j = 1;
%create empty row vector
jb = zeros(1,0);
while i <= m && j <= n
   % Find value and index of largest element in the remainder of column j.
   [p, k] = max(abs(A_bs(i:m,j)));
   k = k+i-1;
   if p < tol
      % The column is negligible, zero it out.
      A_bs(i:m,j) = 0;
      j = j + 1;
   else
      % Remember column index
      jb = [jb j]; %#ok<AGROW>
      % Swap i-th and k-th rows.
      A_bs([i k],j:end) = A_bs([k i],j:end);
      % Divide the pivot row by the pivot element.
      A_bs(i,j:end) = A_bs(i,j:end)./A_bs(i,j);
      % Subtract multiples of the pivot row from all the other rows.
      for k = [1:i-1 i+1:m]
         A_bs(k,j:end) = A_bs(k,j:end) - A_bs(k,j).*A_bs(i,j:end)
      end
      i = i + 1;       j = j + 1;
   end
end
```

```
A_bs =
     1      9/10    -1/5     -1     0     0
     0   -181/10    34/5      2     0     1
     6        7       5       4     0     0
    -2        3       4       3     1     0
A_bs =
     1      9/10    -1/5     -1     0     0
     0   -181/10    34/5      2     0     1
```

```
       0          8/5          31/5         10            0            0
      -2          3            4            3             1            0
A_bs =
       1          9/10        -1/5         -1             0            0
       0        -181/10        34/5          2             0            1
       0          8/5          31/5         10            0            0
       0         24/5          18/5          1             1            0
A_bs =
       1          0           25/181      -163/181        0           9/181
       0          1          -68/181       -20/181        0         -10/181
       0          8/5          31/5         10            0            0
       0         24/5          18/5          1             1            0
A_bs =
       1          0           25/181      -163/181        0           9/181
       0          1          -68/181       -20/181        0         -10/181
       0          0         1231/181      1842/181        0          16/181
       0         24/5          18/5          1             1            0
A_bs =
       1          0           25/181      -163/181        0           9/181
       0          1          -68/181       -20/181        0         -10/181
       0          0         1231/181      1842/181        0          16/181
       0          0          978/181       277/181        1          48/181
A_bs =
       1          0            0         -1363/1231       0          59/1231
       0          1          -68/181       -20/181        0         -10/181
       0          0            1          1842/1231       0          16/1231
       0          0          978/181       277/181        1          48/181
A_bs =
       1          0            0         -1363/1231       0          59/1231
       0          1            0           556/1231       0         -62/1231
       0          0            1          1842/1231       0          16/1231
       0          0          978/181       277/181        1          48/181
A_bs =
       1          0            0         -1363/1231       0          59/1231
       0          1            0           556/1231       0         -62/1231
       0          0            1          1842/1231       0          16/1231
       0          0            0         -1016/155        1         240/1231
A_bs =
       1          0            0            0        -1363/8069     121/8069
       0          1            0           556/1231       0         -62/1231
       0          0            1          1842/1231       0          16/1231
       0          0            0            1         -155/1016    -240/8069
A_bs =
       1          0            0            0        -1363/8069     121/8069
       0          1            0            0          556/8069    -298/8069
       0          0            1          1842/1231       0          16/1231
       0          0            0            1         -155/1016    -240/8069
A_bs =
       1          0            0            0        -1363/8069     121/8069
       0          1            0            0          556/8069    -298/8069
       0          0            1            0          247/1082     141/2452
       0          0            0            1         -155/1016    -240/8069
```

```
disp(A_bs)
```

```
       1          0            0            0        -1363/8069     121/8069
       0          1            0            0          556/8069    -298/8069
       0          0            1            0          247/1082     141/2452
       0          0            0            1         -155/1016    -240/8069
```

```
Ainv2 = A_bs(:,5:end);
```

ma trận nghịch đảo của A là

```
disp(Ainv2)
```

```
  -1363/8069      121/8069      414/3635     -126/8069
    556/8069     -298/8069        4/8069      377/8069
    247/1082      141/2452     -277/8069      117/8069
   -155/1016     -240/8069      491/4051     -617/8069
```

Phương pháp Cholesky tìm ma trận nghịch đảo của ma trận đối xứng

Giả sử ma trận $A = (a_{ij})_{n\times n}$ là ma trận đối xứng $a_{ij} = a_{ji}$. Ta đã biết nếu $det(A) \neq 0$ thì có thể phân tích $A = Q^tQ$ trong đó Q là ma trận tam giác trên. Khi đó

$$A^{-1} = (Q^tQ)^{-1} = Q^{-1}(Q^{-1})^t.$$

Bài toán tìm A^{-1} quy về tìm Q^{-1} trong đó

$$Q = \begin{pmatrix} q_{11} & q_{12} & .. & q_{1n} \\ 0 & q_{22} & .. & q_{2n} \\ \cdots & & & \\ 0 & 0 & .. & q_{nn} \end{pmatrix}.$$

Giả sử $Q^{-1} = (\alpha_{ij})_n$. Khi đó đẳng thức $QQ^{-1} = E$ có thể viết lại

$$\begin{pmatrix} q_{11} & q_{12} & .. & q_{1n} \\ 0 & q_{22} & .. & q_{2n} \\ \cdots & & & \\ 0 & 0 & .. & q_{nn} \end{pmatrix}\begin{pmatrix} \alpha_{11} & \alpha_{12} & \cdots & \alpha_{1n} \\ \alpha_{21} & \alpha_{22} & \cdots & \alpha_{2n} \\ \cdots & & & \\ \alpha_{n1} & \alpha_{n2} & \cdots & \alpha_{nn} \end{pmatrix} = \begin{pmatrix} 1 & 0 & \cdots & 0 \\ 0 & 1 & \cdots & 0 \\ \cdots & & & \\ 0 & 0 & \cdots & 1 \end{pmatrix}.(3)$$

Nhân 2 ma trận của vế phải của (3) và so sánh với ma trận đơn vị ở vế trái ta có

Lấy dòng n của ma trận Q nhân với các cột $n, n-1, ..., 1$ của ma trận Q^{-1}, nhận được

$$\alpha_{nn} = \frac{1}{q_{nn}}, \alpha_{n,j} = 0 \forall j < n;$$

Lấy dòng $n-1$ của ma trận Q nhân với các cột $n-1, n-2, ..., 1$ của ma trận Q^{-1}, nhận được

$$\alpha_{n-1,n-1} = \frac{1}{q_{n-1,n-1}}, \alpha_{n-1,j} = 0 \forall j < n-1;$$

Lấy dòng $n-1$ của ma trận Q nhân với các cột n của ma trận Q^{-1}, nhận được

$$\alpha_{n-1,n} = \frac{-q_{n-1,n}\alpha_{n,n}}{q_{n-1,n-1}}.$$

Tiếp tục quá trình như trên, ta nhận được:

$$\begin{aligned} &\alpha_{ii} = \frac{1}{q_{ii}}, \forall i = \overline{1,n}; \\ &\alpha_{ij} = 0 \qquad \forall i > j; \\ &\alpha_{ik} = -\frac{1}{q_{ii}}\sum_{j=i+1}^{k} q_{ij}\alpha_{jk} \qquad \forall i = \overline{1,n}, \forall j = i+1, \cdots, n. \end{aligned}$$

Nhận xét: Q^{-1} là ma trận tam giác trên có các phần tử trên đường chéo chính là nghịch đảo của các phần tử trên đường chéo chính của Q.

Ví dụ 5. Tìm ma trận nghịch đảo của

```
clc
clear all
%A=[1 2 1;2 5 1;1 1 3]
%A=[1 3 -2;3 4 -5;-2 -5 3]

A=[    -2     3     4     3
     9   -10     5    -7
     6     7     5     4
    10     9    -2   -10
];
A=A*A'
```

```
A =

    38         -49          41         -31
   -49         255         -19          60
    41         -19         126          73
   -31          60          73         285
```

Bước 1. Ta phân tích ma trận A thành dạng Q^TQ sử dụng phân tích Cholesky

```
Q = zeros(size(A));
n=size(A,1);
for i=1:n
    Q(i,i)=sqrt(A(i,i)-sum(Q(1:i-1,i).^2));
    Q(i,i+1:end) = 1/Q(i,i)*(A(i,i+1:end)-Q(1:i-1,i).'*Q(1:i-1,i+1:end));
end
format short
disp(Q)
```

```
    6.1644   -7.9488    6.6511   -5.0289
         0   13.8498    2.4454    1.4460
         0         0    8.7053   11.8216
         0         0         0   10.8567
```

```
disp(Q.'*Q)
```

```
   38.0000  -49.0000   41.0000  -31.0000
  -49.0000  255.0000  -19.0000   60.0000
   41.0000  -19.0000  126.0000   73.0000
  -31.0000   60.0000   73.0000  285.0000
```

Bước 2. Tìm ma trận Q^{-1}

```
Qinv = diag(1./diag(Q));
for i=n:-1:1
    Qinv(i,i+1:end)=-1./Q(i,i)*Q(i,i+1:end)*Qinv(i+1:end,i+1:end);
end
```

Ma trận Q^{-1} là

```
Qinv
```

```
Qinv = 4×4
    0.1622    0.0931   -0.1501    0.2262
         0    0.0722   -0.0203    0.0125
         0         0    0.1149   -0.1251
         0         0         0    0.0921
```

Kiểm tra QQ^{-1}

```
Q*Qinv
```

```
ans = 4×4
    1.0000         0   -0.0000         0
         0    1.0000         0         0
         0         0    1.0000         0
         0         0         0    1.0000
```

Bước 3. Tính ma trận $A^{-1} = Q^{-1}(Q^{-1})^T$

```
Ainv3 = Qinv*Qinv.';
disp(Ainv3)
```

```
    0.1087    0.0126   -0.0455    0.0208
    0.0126    0.0058   -0.0039    0.0011
   -0.0455   -0.0039    0.0288   -0.0115
    0.0208    0.0011   -0.0115    0.0085
```

```
disp(A*Ainv3)
```

```
    1.0000    0.0000   -0.0000    0.0000
    0.0000    1.0000    0.0000         0
    0.0000    0.0000    1.0000         0
   -0.0000   -0.0000    0.0000    1.0000
```

Kiểm tra bằng câu lệnh MATLAB thấy kết quả trùng nhau.

```
inv(A)
```

```
ans = 4×4
    0.1087    0.0126   -0.0455    0.0208
    0.0126    0.0058   -0.0039    0.0011
   -0.0455   -0.0039    0.0288   -0.0115
    0.0208    0.0011   -0.0115    0.0085
```

Chú ý: Khi $det(A)$ gần bằng 0 hoặc rất lớn thì quá trình tính không ổn định.

Phương pháp viền quanh

Ma trận khối

$$A_n = \begin{pmatrix} A_{n-1} & \alpha_{n-1,1} \\ \alpha_{1,n-1} & a_{n,n} \end{pmatrix}$$

trong đó A_{n-1} là ma trận cấp $n-1$. Gọi ma trận nghịch đảo của A_n là

$$A_n^{-1} = \begin{pmatrix} B_{n-1} & \beta_{n-1,1} \\ \beta_{1,n-1} & b_{n,n} \end{pmatrix}$$

trong đó B_{n-1} là ma trận cấp $n-1$.

Để nhân hai ma trận khối ta coi các khối như phần tử của ma trận và nhân theo qui tắc thông thường (chú ý kích cỡ của các khối phải phù hợp để nhân).

Ví dụ 6. Cho các ma trận $A_2 = \begin{pmatrix} 1 & 2 & 5 \\ -3 & 1 & -2 \end{pmatrix}, B_1 = \begin{pmatrix} 1 & 2 & 6 \end{pmatrix}$ và $A_3 = \begin{pmatrix} 1 & 3 & 2 \\ 9 & -3 & 1 \\ 0 & 0 & 1 \end{pmatrix}$. Xét phép nhân

$\begin{pmatrix} A_2 \\ B_1 \end{pmatrix} A_3 = \begin{pmatrix} 1 & 2 & 5 \\ -3 & 1 & -2 \\ 1 & 2 & 6 \end{pmatrix} \begin{pmatrix} 1 & 3 & 2 \\ 9 & -3 & 1 \\ 0 & 0 & 1 \end{pmatrix}$.

```
A2=[1 2 5; -3 1 -2]; B1=[1 2 6]; A3 =[1 3 2;9 -3 1;0 0 1];
```

Thực hiện nhân thông thường

```
[A2;B1]*A3
```

```
ans = 3×3
    19    -3     9
     6   -12    -7
    19    -3    10
```

Thực hiện nhân theo khối $\begin{pmatrix} A_2 \\ B_1 \end{pmatrix} A_3 = \begin{pmatrix} A_2 * A_3 \\ B_1 * A_3 \end{pmatrix}$

```
A2*A3
```

```
ans = 2×3
    19    -3     9
     6   -12    -7
```

```
B1*A3
```

```
ans = 1×3
    19    -3    10
```

Ghép hai ma trận kết quả ta cũng được ma trận tích như cách nhân thông thường.

Bây giờ ta nhân AA^{-1}

$$A \cdot A^{-1} = E \Leftrightarrow \begin{cases} A_{n-1}B_{n-1} + \alpha_{n-1,1}\beta_{1,n-1} = E_{n-1} & (a) \\ A_{n-1}\beta_{n-1,1} + \alpha_{n-1,1}b_{n,n} = 0_{n-1,1} & (b) \\ \alpha_{1,n-1}B_{n-1} + a_{n,n}\beta_{1,n-1} = 0_{1,n-1} & (c) \\ \alpha_{1,n-1}\beta_{n-1,1} + a_{n,n}b_{n,n} = 1 & (d) \end{cases}.$$

- Lấy (b) nhân với $\alpha_{1,n-1}A_{n-1}^{-1}$ từ bên trái rồi trừ đi (d), ta tìm được

$$b_{n,n} = \frac{1}{a_{n,n} - \alpha_{1,n-1}A_{n-1}^{-1}\alpha_{n-1,1}}.$$

- Thay vào (b) ta có

$$\beta_{n-1,1} = \frac{-A_{n-1}^{-1}\alpha_{n-1,1}}{a_{n,n} - \alpha_{1,n-1}A_{n-1}^{-1}\alpha_{n-1,1}}.$$

- Làm tương tự với cặp phương trình (a) và (c) ta thu được:

$$\begin{cases} \beta_{1,n-1} = -\dfrac{\alpha_{1,n-1}A_{n-1}^{-1}}{a_{n,n} - \alpha_{1,n-1}A_{n-1}^{-1}\alpha_{n-1,1}}, \\ B_{n-1} = A_{n-1}^{-1}\left(E_{n-1} + \dfrac{\alpha_{n-1,1}\alpha_{1,n-1}A_{n-1}^{-1}}{a_{n,n} - \alpha_{1,n-1}A_{n-1}^{-1}\alpha_{n-1,1}}\right). \end{cases}$$

Để tìm ma trận nghịch đảo của ma trận A ta áp dụng phương pháp viền quanh để tìm lần lượt ma trận con góc trên bên trái của A.

Điều kiện áp dụng: Tồn tại A_k^{-1} với mọi $k = \overline{1,n}$ trong đó A_k là ma trận con cấp k góc trên bên trái của A.

Trong trường hợp A khả nghịch bất kỳ thì ta đi tìm ma trận nghịch đảo của ma trận $M = A^T A$. Để ý

$$x^t(M)x = x^T(A^T A)x = \langle x, A^T Ax\rangle = \langle Ax, Ax\rangle \geq 0 \forall x \in \mathbb{R}^n,$$

điều này chứng tỏ dạng toàn phương ứng với ma trận đối xứng M luôn dương, do đó các ma trận M_k, $\forall k = \overline{1,n}$ lớn hơn), hay tồn tại M_k^{-1}, $\forall k = \overline{1,n}$. Lại có $M^{-1} = A^{-1}(A^{-1})^T \Leftrightarrow A^{-1} = M^{-1}A^T$.

Ví dụ 7. Tìm ma trận nghịch đảo của A

```
clc
clear all
A=[    -2      3      4      3
        9    -10      5     -7
        6      7      5      4
       10      9     -2    -10
]
```

```
A = 4×4
    -2     3     4     3
     9   -10     5    -7
     6     7     5     4
    10     9    -2   -10
```

Trước hết ta tìm ma trận nghịch đảo của M

```
M=A'*A
```

```
M = 4×4
   221    36    47  -145
    36   239   -21    17
    47   -21    70    17
  -145    17    17   174
```

```
A_k_inv=1/M(1,1);
for i = 1:size(M,1)-1
    alpha21=M(1:i,i+1);%cột
    alpha12=M(i+1,1:i);%dòng
    ann = M(i+1,i+1);
    D=(ann-alpha12*A_k_inv*alpha21);
    bnn=1/(ann-alpha12*A_k_inv*alpha21);
    beta21=-A_k_inv*alpha21/D;%cột
    beta12=-alpha12*A_k_inv/D;%dòng
    B=A_k_inv*(eye(i)+alpha21*alpha12*A_k_inv/D);
    A_k_inv=[B,beta21;beta12,bnn]
end
```

```
A_k_inv = 2×2
    0.0046   -0.0007
   -0.0007    0.0043
A_k_inv = 3×3
    0.0056   -0.0012   -0.0041
   -0.0012    0.0046    0.0022
   -0.0041    0.0022    0.0177
A_k_inv = 4×4
    0.0420   -0.0129   -0.0418    0.0403
   -0.0129    0.0083    0.0143   -0.0129
   -0.0418    0.0143    0.0568   -0.0418
    0.0403   -0.0129   -0.0418    0.0447
```

```
Minv=A_k_inv;
inv(M)
```

```
ans = 4×4
    0.0420   -0.0129   -0.0418    0.0403
   -0.0129    0.0083    0.0143   -0.0129
   -0.0418    0.0143    0.0568   -0.0418
    0.0403   -0.0129   -0.0418    0.0447
```

Tìm ma trận $A^{-1} = M^{-1}A^T$

```
Ainv = Minv*A';
disp(Ainv)
```

```
   -0.1689    0.0150    0.1139   -0.0156
    0.0689   -0.0369    0.0005    0.0467
    0.2283    0.0575   -0.0343    0.0145
   -0.1526   -0.0297    0.1212   -0.0765
```

Kiểm tra lại bằng câu lệnh có sẵn của MATLAB

```
inv(A)
```

```
ans = 4×4
   -0.1689    0.0150    0.1139   -0.0156
    0.0689   -0.0369    0.0005    0.0467
    0.2283    0.0575   -0.0343    0.0145
   -0.1526   -0.0297    0.1212   -0.0765
```

Phương pháp lặp JACOBI

Nếu ma trận A là chéo trội hàng hoặc cột thì có thể dùng công thức lặp

$$X = (E - TA)X + T$$

trong đó $T = diag(1/a_{11}, 1/a_{22}, \cdots, 1/a_{nn})$, I là ma trận đơn vị cấp n để tìm gần đúng ma trận nghịch đảo của A. Bản chất chúng ta đi tìm nghiệm gần đúng của phương trình ma trận $AX = E$.

Ví dụ 8. Tìm ma trận nghịch đảo của A

```
clc
clear all
A=[   50     3     4     3
     9   30     5    -7
     6     7    -60    4
    10     9    -2   100
]
```

```
A = 4×4
    50     3     4     3
     9    30     5    -7
     6     7   -60     4
    10     9    -2   100
```

Trước hết tạo ma trận T

```
T = diag(1./diag(A));
format rational
disp(T)
```

```
       1/50           0              0              0
       0              1/30           0              0
       0              0             -1/60           0
       0              0              0              1/100
```

```
alpha = eye(size(A,1))-T*A;
beta = T;
```

Ma trận lặp là α

```
disp(alpha)
```

```
       0             -3/50          -2/25          -3/50
      -3/10           0             -1/6            7/30
       1/10           7/60           0              1/15
      -1/10          -9/100          1/50           0
```

Chọn xấp xỉ đầu $X0$

```
x0= randi([-10,10],size(A))
```

```
x0 =
      -9           9          -6           0
      -5           5          -1          -6
       6           0          10           0
     -10           2           1           3
```

Thực hiện lặp $n = 50$ lần

```
n=50;
for i=1:n
    x1=alpha*x0+T;
    x0=x1;
end
```

Ma trận nghịch đảo của A là

```
disp(x1)
```

```
      23/1128      -38/18267      63/51944      -20/24819
     -50/7527       21/641        25/11327       13/5407
      46/39333      57/16589     -30/1837        65/75698
      -3/2116      -53/19837     -19/29387      374/37849
```

Kiểm tra lại bằng câu lệnh MATLAB

```
disp(inv(A))
```

```
      23/1128      -38/18267      63/51944      -20/24819
     -50/7527       21/641        25/11327       13/5407
      46/39333      57/16589     -30/1837        65/75698
      -3/2116      -53/19837     -19/29387      374/37849
```

Phương pháp lặp Gauss-Seidel

Nếu ma trận A là chéo trội hàng hoặc cột thì có thể dùng công thức lặp

$$\begin{cases} x_1^{(k+1)} = b_{12}x_2^{(k)} + \cdots + b_{1n}x_n^{(k)} + d_1 \\ x_2^{(k+1)} = b_{21}x_1^{(k+1)} + \cdots + b_{2n}x_n^{(k)} + d_2 \\ x_3^{(k+1)} = b_{31}x_1^{(k+1)} + b_{32}x_2^{(k+1)} + \cdots + b_{3n}x_n^{(k)} + d_3 \\ \cdots \\ x_n^{(k+1)} = b_{n1}x_1^{(k+1)} + b_{n2}x_2^{(k+1)} + \cdots + b_{n,n-1}x_n^{(k+1)} + d_n \end{cases}$$

trong đó $T = diag(1/a_{11}, 1/a_{22}, \cdots, 1/a_{nn})$,

$$B = E - TA \text{ và } D = T.$$

Trong công thức trên ta kí hiệu x_i là hàng thứ i của ma trận X, d_i là hàng thứ i của ma trận D.

Ví dụ 9. Tìm ma trận nghịch đảo của A

```
clc
clear all
A=[   50     3     4     3
     9   30     5    -7
     6     7     -60    4
    10     9    -2   100
]
```

```
A =

    50              3              4              3
     9             30              5             -7
     6              7            -60              4
    10              9             -2            100
```

```
N=10;
n=size(A,1);
T=diag(1./diag(A));% tạo ma trận đường chéo T
alpha = eye(size(A,1))-T*A;%tạo ma trận lặp alpha
disp(alpha)
```

```
     0             -3/50          -2/25          -3/50
    -3/10           0             -1/6            7/30
     1/10           7/60           0              1/15
    -1/10          -9/100          1/50           0
```

```
L = zeros(size(A));%tạo ma trận tam giác dưới
for i =1:n
    L(i,1:i-1)=alpha(i,1:i-1);
end
U = alpha-L;%tạo ma trận tam giác trên

x0 = randi([-10,10],size(A));
x1 = zeros(size(A));% giữ chỗ cho x1

for k=1:N% lặp theo số lần lặp
    for i=1:n%lặp theo dòng của x1
        x1(i,:)=L(i,:)*x1+U(i,:)*x0+T(i,:);
    end
    x0=x1;
end
format long
disp(x1)
```

```
   0.020390070844222  -0.002080254108296   0.001212844553085  -0.000805833868716
  -0.006642751407978   0.032761327129144   0.002207117356058   0.002404290570587
   0.001169501413150   0.003436011164641  -0.016330988194661   0.000858675202272
  -0.001417769429441  -0.002671773807501  -0.000646544781247   0.009881370739564
```

Kiểm tra lại bằng câu lệnh của MATLAB

```
inv(A)
```

```
ans = 4×4
   0.020390071115562  -0.002080254093637   0.001212844690155  -0.000805833707628
  -0.006642751132218   0.032761327144030   0.002207117495380   0.002404290734233
   0.001169501514173   0.003436011170093  -0.016330988143622   0.000858675262226
  -0.001417769479373  -0.002671773810197  -0.000646544806472   0.009881370709926
```

Phương pháp Newton tìm ma trận nghịch đảo

Ý tưởng: mô phỏng phương pháp tìm nghịch đảo của số thực: Nghịch đảo của số a là nghiệm của phương trình $ax = 1$ hay $x - \frac{1}{a} = 0$. Đặt $f(x) = x - \frac{1}{a}$ thì theo phương pháp Newton nghiệm gần đúng tìm theo công thức lặp

$$x_{n+1} = x_n - \frac{f(x_n)}{f'(x_n)} = x_n - \frac{\frac{1}{x_n} - a}{\frac{-1}{x_n^2}} = x_n - x_n(ax_n - 1).$$

Tương tự, nếu ở vị trí a là ma trận A, số 1 là ma trận đơn vị E thì công thức lặp là

$$X_{n+1} = X_n - X_n(AX_n - E).$$

Sự hội tụ của phương pháp:

$$\begin{aligned}
&X_{n+1} = X_n - X_n(AX_n - E)\\
&G_n = E - AX_n \Rightarrow X_{n+1} = X_n + X_nG_n = X_n(E + G_n)\\
&G_{n+1} = E - AX_{n+1} = E - AX_n(E + G_n)\\
&= E - (E + G_n)(E - G_n)\\
&= G_n^2 = G_{n-1}^{2^2} = \cdots = G_{n-k}^{2^{2+1}} = G_0^{2^{n+1}}\\
&\Rightarrow \|G_{n+1}\| \le \|G_0\|^{2^{n+1}} \longrightarrow 0 \text{ khi } n \longrightarrow \infty \Leftrightarrow \|G_0\| < 1.
\end{aligned}$$

Điều này gợi ý ta phải chọn X_0 sao cho $\|AX_0 - E\| < 1$ hay là X_0 đủ gần A^{-1}.

Sai số:

$$\begin{aligned}
&G_n = AX_n - E = A\left(X_n - A^{-1}\right) \Rightarrow X_n - A^{-1} = A^{-1}G_n\\
&\|X_n - A^{-1}\| \le \|A^{-1}\|\|G_n\| \le \|A^{-1}\|\|G_0\|^{2^n}.
\end{aligned}$$

Lại có

$$G_0 = E - AX_0 \Leftrightarrow A^{-1} = X_0(E - G_0)^{-1} = X_0\sum_{k=0}^{\infty} G_0^k$$

Suy ra

$$\|A^{-1}\| \le \|X_0\|\sum_{k=0}^{\infty}\|G_0\|^k = \frac{\|X_0\|}{1 - \|G_0\|}$$

Thay vào trên ta có

$$\|X_n - A^{-1}\| \le \frac{\|X_0\|}{1 - \|G_0\|}\|G_0\|^{2^n}.$$

Nhận xét: Trong thực hành thì chọn X_0 thỏa mãn $\|AX_0 - E\| < 1$ là vấn đề khó.

Ví dụ 10. Tìm ma trận nghịch đảo của A

```
clc
clear all
A=[   50     3     4     3
     9   30     5    -7
     6     7     -60    4
    10     9    -2   100
]
```

```
A = 4×4
    50     3     4     3
     9    30     5    -7
     6     7   -60     4
    10     9    -2   100
```

với xấp xỉ đầu X_0

```
x0=     [    0.020538691227195  -0.002491662019002   0.000955676728967   0.000514814763804
  -0.006506991336407   0.032380069130999   0.001945684318962   0.003585295375661
   0.001220078436307   0.003299143629787  -0.016425107243146   0.001300566931064
  -0.001443838333717  -0.002599057147294  -0.000599181406466   0.009651853278431]
```

```
x0 = 4×4
   0.020538691227195  -0.002491662019002   0.000955676728967   0.000514814763804
  -0.006506991336407   0.032380069130999   0.001945684318962   0.003585295375661
   0.001220078436307   0.003299143629787  -0.016425107243146   0.001300566931064
  -0.001443838333717  -0.002599057147294  -0.000599181406466   0.009651853278431
```

Nhận thấy $AX_0 - E$ có chuẩn bằng

```
norm(A*x0-eye(size(A,1)),inf)
```

```
ans =
   0.114537412237921
```

thỏa mãn điều kiện hội tụ.

```
E=eye(size(A))
```

```
E = 4×4
     1     0     0     0
     0     1     0     0
     0     0     1     0
     0     0     0     1
```

```
for i=1:3
    x1 = x0-x0*(A*x0-E);
    x0=x1
end
```

```
x0 = 4×4
   0.020390959220564  -0.002082832287114   0.001211031767166  -0.000798188981288
  -0.006641942408359   0.032758972281206   0.002205467048696   0.002411254528360
   0.001169776834951   0.003435207273009  -0.016331554435251   0.000861042136700
  -0.001417925568605  -0.002671320131137  -0.000646226299804   0.009880026833310
x0 = 4×4
   0.020390071151777  -0.002080254198939   0.001212844616692  -0.000805833395377
  -0.006642751098990   0.032761327047414   0.002207117427974   0.002404291020724
   0.001169501525534   0.003436011137058  -0.016330988166670   0.000858675360187
  -0.001417769485758  -0.002671773791632  -0.000646544793520   0.009881370654876
x0 = 4×4
   0.020390071115562  -0.002080254093637   0.001212844690155  -0.000805833707628
  -0.006642751132218   0.032761327144030   0.002207117495380   0.002404290734233
   0.001169501514173   0.003436011170093  -0.016330988143622   0.000858675262226
  -0.001417769479373  -0.002671773810197  -0.000646544806472   0.009881370709926
```

Kiểm tra trực tiếp

```
A*x0
```

```
ans = 4×4
   1.000000000000000  -0.000000000000000   0.000000000000000   0.000000000000000
  -0.000000000000000   1.000000000000000   0.000000000000000   0.000000000000000
  -0.000000000000000   0.000000000000000   1.000000000000000  -0.000000000000000
  -0.000000000000000                   0  -0.000000000000000   1.000000000000000
```