第七章

线性方程组的数值解法

$$
\begin{cases}
a_{11}x_1 + a_{12}x_x + \cdots + a_{1n}x_n = b_1 \\
a_{21}x_1 + a_{22}x_x + \cdots + a_{2n}x_n = b_2 \\
\qquad\qquad\cdots\cdots \\
a_{n1}x_1 + a_{n2}x_x + \cdots + a_{nn}x_n = b_n
\end{cases}
$$

n阶线性代数方程组的一般形式为：

$$\begin{cases} a_{11}x_1 + a_{12}x_x + \cdots + a_{1n}x_n = b_1 \\ a_{21}x_1 + a_{22}x_x + \cdots + a_{2n}x_n = b_2 \\ \cdots\cdots \\ a_{n1}x_1 + a_{n2}x_x + \cdots + a_{nn}x_n = b_n \end{cases}$$

写成矩阵-向量形式 $Ax = b$

其中 A 为系数矩阵，x 为解向量，b 为右端常向量。

$$A = \begin{bmatrix} a_{11} & \cdots & a_{1n} \\ \vdots & & \vdots \\ a_{n1} & \cdots & a_{nn} \end{bmatrix} \quad x = \begin{bmatrix} x_1 \\ \vdots \\ x_n \end{bmatrix} \quad b = \begin{bmatrix} b_1 \\ \vdots \\ b_n \end{bmatrix}$$

若矩阵A非奇异，即A的行列式 $\det A \neq 0$，根据克莱姆（Gramer）法则，方程组有唯一 解：

$$x_i = \frac{D_i}{D} \qquad i = 1,2,\cdots,n$$

其中D表示$\det A$，D_i表示 D 中第i列换成 b 后所得的行列式。

当阶数较高时用这种方法求解是不现实的。n阶行列式有 $n!$项，每项又是n个数的乘积。对较大的n，其计算量之大，是一般计算机难以完成的。

例：用克莱姆法则求解一个20阶线性方程组时，所需乘法次数为

20！ =2432 9020 0817 6640 000

20*20*20！ =9731 6080 3270 6560 000 00

若采用10亿/秒的计算机，要花费计算时间为

20*20*20！ /(365*24*60*60*1 000 000 000)

=30 858.73(年)

线性方程组的计算机解法：

- **直接法**：经过有限步算术运算，可求得方程组精确解的方法

  ----Gauss消元法、矩阵三角分解

- **迭代法**：用某种极限过程去逐步逼近方程组精确解的方法

Gauss消元法

——按自然顺序进行的消元法

基本思想： 是通过将一个方程乘或除以某个常数，以及将两个方程相加减，逐步减少方程中的变元数，最终使每个方程只含一个变元，从而得出所求的解。

Gauss消元法由“**消元过程**”和“**回代过程**”两部分组成。

例 1 用Gauss消元法求解方程组

$$\begin{cases} 2x_1 + 8x_2 + 2x_3 = 14 \\ x_1 + 6x_2 - x_3 = 13 \\ 2x_1 + x_2 + 2x_3 = 5 \end{cases}$$

解: 用第一个方程消去后两个方程中的 x_1，得

$$\begin{cases} 2x_1 + 8x_2 + 2x_3 = 14 \\ \qquad 2x_2 - 2x_3 = 6 \\ \quad -9x_2 \qquad = -9 \end{cases}$$

再用第2个方程消去第3个方程中的 x_2,得

$$\begin{cases} 2x_1 + 8x_2 + 2x_3 = 14 \\ \qquad 2x_2 - 2x_3 = 6 \\ \qquad\qquad -9x_3 = 18 \end{cases}$$

最后，经过回代求得原方程组的解为

$$x_3 = \frac{18}{-9} = -2$$

$$x_2 = \frac{6 + 2x_3}{2} = 1$$

$$x_1 = \frac{14 - 8x_2 - 2x_3}{2} = 5$$

例 2 解方程组

$$\begin{cases} 2x_1 + x_2 + x_3 = 7 \\ 4x_1 + 5x_2 - x_3 = 11 \\ x_1 - 2x_2 + x_3 = 0 \end{cases}$$

解：消元

$$\begin{pmatrix} 2 & 1 & 1 & 7 \\ 4 & 5 & -1 & 11 \\ 1 & -2 & 1 & 0 \end{pmatrix} \xrightarrow[-\frac{1}{2}r_1+r_3]{-\frac{4}{2}r_1+r_2} \begin{pmatrix} 2 & 1 & 1 & 7 \\ 0 & 3 & -3 & -3 \\ 0 & -2.5 & 0.5 & -3.5 \end{pmatrix}$$

$$\xrightarrow{-\frac{-2.5}{3}r_2+r_3} \begin{pmatrix} 2 & 1 & 1 & 7 \\ 0 & 3 & -3 & -3 \\ 0 & 0 & -2 & -6 \end{pmatrix}$$

回代得 $x_3 = \dfrac{-6}{-2} = 3,\qquad x_2 = \dfrac{-3+3x_3}{3} = 2,$

$$x_1 = \frac{7 - x_2 - x_3}{2} = 1$$

Matlab求解

当系数矩阵A为N*N的方阵时，MATLAB会自行用Gauss消元法求解线性代数方程组。

在MATLAB中的求解命令有:

$$x = A \backslash b$$

$$x = inv(A) * b$$

$$x = A \wedge (-1) * b$$

$$x = b' / A'$$

$$x = b' * inv(A')$$

如：求解 $\begin{pmatrix} 3 & 2 \\ 1 & -1 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} = \begin{pmatrix} -1 \\ 1 \end{pmatrix}$

设：$A=[3,2;1,-1];$

$b=[-1,1]';$

则：$x = A \backslash b$

得：$x =$

0.2000

-0.8000

例3　求解下列线性方程组。

$$\begin{cases} 2x_1 + x_2 - 5x_3 + x_4 = 13 \\ x_1 - 5x_2 + 7x_4 = -9 \\ 2x_2 + x_3 - x_4 = 6 \\ x_1 + 6x_2 - x_3 - 4x_4 = 0 \end{cases}$$

命令如下：

```
>>A=[2,1,-5,1;1,-5,0,7;0,2,1,-1;1,6,-1,-4];
>>b=[13,-9,6,0]';
>>x=A\b              % x=inv(A)*b
x =
  -66.5556
   25.6667
  -18.7778
   26.5556
```

例4：用Gauss消元法求解方程组：

$$\begin{cases} 5x_1+6x_2 & =1 \\ x_1+5x_2+6x_3 & =0 \\ x_2+5x_3+6x_4 & =0 \\ x_3+5x_4+6x_5 & =0 \\ x_4+5x_5 & =1 \end{cases}$$

程序如下：

```
A=[5  6  0  0  0
   1  5  6  0  0
   0  1  5  6  0
   0  0  1  5  6
   0  0  0  1  5];
B=[1 0 0 0 1]';
R_A=rank(A)        %求秩
X=A\B              %求解
```

```
>> A=[5   6   0   0   0
      1   5   6   0   0
      0   1   5   6   0
      0   0   1   5   6
      0   0   0   1   5];     B=[1 0 0 0 1]';
R_A=rank(A)        %求秩
X=A\B              %求解

R_A =

     5


X =

    2.2662
   -1.7218
    1.0571
   -0.5940
    0.3188
```

Matlab求解

若系数矩阵的秩r<n，则可能有无穷解；

线性方程组的无穷解 = 对应齐次方程组的通解 +非齐次方程组的一个特解

Matlab将矩阵化成行最简形的命令是rref，其格式为：

R = rref(A)

R 是A的行最简行矩阵

[R,jb] = rref(A)

jb是一个向量，jb中元素表示基向量所在的列

例　求方程组 $\begin{cases} x_1 + x_2 - 3x_3 - x_4 = 1 \\ 3x_1 - x_2 - 3x_3 + 4x_4 = 4 \\ x_1 + 5x_2 - 9x_3 - 8x_4 = 0 \end{cases}$ 的一个特解

解：

用rref求解：

```
» A=[1 1 -3 -1;3 -1 -3 4;1 5 -9 -8];
» B=[1  4  0]';
» C=[A,B];   %构成增广矩阵
» R=rref(C)
```

```
>> C=[A,B];     %构成增广矩阵
R=rref(C)

R =

    1.0000         0   -1.5000    0.7500    1.2500
         0    1.0000   -1.5000   -1.7500   -0.2500
         0         0         0         0         0
```

由此得解向量X=[1.2500 –0.2500 0 0]'(一个特解)。

在Matlab中，函数null用来求解零空间，即满足A·X=0的解空间，实际上是求出解空间的一组基(基础解系)。其格式为：

z = null(A)

z的列向量为方程组AX=0的正交规范基，满足$z' \times z = I$

设 $A=\begin{bmatrix} 1 & 1 & -1 & -1 \\ 2 & -5 & 3 & 2 \\ 7 & -7 & 3 & 11 \end{bmatrix}$ $B=\begin{bmatrix} 5 \\ -4 \\ 7 \end{bmatrix}$

求线性方程组Ax=B的通解。

解法1：　利用除法 \ 和 null 函数

在命令窗口输入以下命令：　　(注意：这里给出的 A不 是方阵）

```
A=[1 1 -1 -1;2 -5 3 2;7 -7 3 1];
B=[5; -4; 7];
format rat
x1=A\B            %求得非齐次方程组Ax=B的一个特解x1
Y=null(A,'r')     %求得齐次方程组Ax=0 的基础解系Y
```

输出结果：

```
x1 =
     3
     2
     0
     0
```

```
Y =
     2/7        3/7
     5/7        4/7
      1          0
      0          1
```

则方程组Ax=B的通解为：x=x1+k1*Y(:,1)+k2*Y(:,2)

解法2：利用 rref 函数

在命令窗口输入以下命令：

```
format rat
A=[1 1 -1 -1;2 -5 3 2;7 -7 3 1];
B=[5; -4; 7];
%用初等行变换将增广矩阵 [A B] 化成最简行阶梯形T
T=rref([A B])
```

输出结果：

```
T =
     1       0      -2/7     -3/7      3
     0       1      -5/7     -4/7      2
     0       0        0        0       0
```

于是可得方程组Ax=B的通解为:

$$X = \begin{pmatrix} 3 \\ 2 \\ 0 \\ 0 \end{pmatrix} + k_1 \begin{pmatrix} 2/7 \\ 5/7 \\ 1 \\ 0 \end{pmatrix} + k_2 \begin{pmatrix} 3/7 \\ 4/7 \\ 0 \\ 1 \end{pmatrix}$$

2．利用矩阵的分解求解线性方程组

矩阵分解是指根据一定的原理用某种算法将一个矩阵分解成若干个矩阵的乘积。常见的矩阵分解有LU分解、QR分解、Cholesky分解，以及Schur分解、Hessenberg分解、奇异值分解等。

利用矩阵的分解求解线性方程组优点：

运算速度快，节省存储空间。

LU分解

矩阵的LU分解就是将一个矩阵表示为一个下三角矩阵和一个上三角矩阵的乘积形式。

MATLAB提供的lu函数用于对矩阵进行LU分解，其调用格式为(矩阵A必须是方阵):

(1) [L,U]=lu(A)

产生一个上三角阵U和一个矩阵L(行交换后，可变成下三角阵)，使之满足A=LU。

注:

1. 对奇异矩阵，Matlab中也可进行LU分解;
2. L往往不是下三角阵;
3. A=L*U;

```
>> A=[1,2,3;2,4,6;7,8,9]
A =
     1     2     3
     2     4     6
     7     8     9
>> [L,U]=lu(A)
L =
    0.1429    0.5000    1.0000
    0.2857    1.0000         0
    1.0000         0         0
%注意L不是下三角阵；
U =
    7.0000    8.0000    9.0000
         0    1.7143    3.4286
         0         0         0
%U为上三角阵。
```

```
>> L*U
ans =
     1     2     3
     2     4     6
     7     8     9
```

说明A=L*U。

(2) [L,U,P]=lu(A)

产生一个上三角阵U和一个单位下三角阵L以及一个置换矩阵P，使之满足PA=LU。

注：

1. 此时L是下三角阵；
2. 但是A≠L*U；
3. A=inv(P)*L*U。

```
>> A=[1,2,3;2,4,6;7,8,9]
A =
     1     2     3
     2     4     6
     7     8     9
>> [L,U,P]=lu(A)
L =
    1.0000         0         0
    0.2857    1.0000         0
    0.1429    0.5000    1.0000
% L单位下三角阵
U =
    7.0000    8.0000    9.0000
         0    1.7143    3.4286
         0         0         0
% U上三角阵
P =
     0     0     1
     0     1     0
     1     0     0
```

```
>> P*A
ans =
     7     8     9
     2     4     6
     1     2     3
>> L*U
ans =
     7     8     9
     2     4     6
     1     2     3
```

说明A ≠L*U；

但PA=LU。

实现LU分解后，求线性方程组Ax=b的解命令为:

x=U\(L\b)　(对应分解(1)式 [L,U]=lu(A))

or　x=U\(L\Pb). (对应分解(2)式 [L,U,P]=lu(A))

推导过程

$Ax=b \rightarrow L(Ux)=b$

$\rightarrow Ux=y,\ Ly=b$

$\rightarrow y=L^{-1}b$

$\rightarrow x=U^{-1}L^{-1}b$

$PA=LU,\ A=P^{-1}LU$

$Ax=b \rightarrow P^{-1}LUx=b$

$\rightarrow LUx=Pb$

$\rightarrow Ux=y,\ Ly=Pb$

$\rightarrow y=L^{-1}Pb,\ x=U^{-1}L^{-1}Pb$

例5 用LU分解求解线性方程组。

命令如下：

```
A=[2,1,-5,1;1,-5,0,7;0,2,1,-1;1,6,-1,-4];
b=[13,-9,6,0]';
[L,U]=lu(A)
L =
    1.0000         0         0         0
    0.5000    1.0000         0         0
         0   -0.3636    0.4773    1.0000
    0.5000   -1.0000    1.0000         0
U =
    2.0000    1.0000   -5.0000    1.0000
         0   -5.5000    2.5000    6.5000
         0         0    4.0000    2.0000
         0         0         0    0.4091
x=U\(L\b)
x =
  -66.5556
   25.6667
  -18.7778
   26.5556
```

或采用LU分解的第2种格式，命令如下：

```
[L,U ,P]=lu(A)
L =
    1.0000         0         0         0
    0.5000    1.0000         0         0
    0.5000   -1.0000    1.0000         0
         0   -0.3636    0.4773    1.0000
U =
    2.0000    1.0000   -5.0000    1.0000
         0   -5.5000    2.5000    6.5000
         0         0    4.0000    2.0000
         0         0         0    0.4091
P =
     1     0     0     0
     0     1     0     0
     0     0     0     1
     0     0     1     0
x=U\(L\P*b)

x =
  -66.5556
   25.6667
  -18.7778
   26.5556
```

QR分解

- A=QR：A为方阵，Q为正交矩阵， R为上三角矩阵
- [Q，R]=qr(A)：产生一个一个正交矩阵Q和一个上三角矩阵R，满足A=QR
- [Q，R，P]=qr(A)：产生一个一个正交矩阵Q、一个上三角矩阵R、置换矩阵P，满足AP=QR

- **实现QR分解后，**
  若采用[Q，R]=qr(X)，则Ax=b的解为 x=R\(Q\b)
  若采用[Q，R，P]=qr(X)，则Ax=b的解为 x=P(R\(Q\b))

$$Ax = b \to QRx = b \to Rx = y, Qy = b \to y = Q^{-1}b, x = R^{-1}Q^{-1}b$$

$$AP = QR \to A = QRP^{-1}$$

$$Ax = b \to QRP^{-1}x = b \to P^{-1}x = y, x = Py, QRy = b$$

$$\to Ry = z, Qz = b \to z = Q^{-1}b, y = R^{-1}Q^{-1}b, x = PR^{-1}Q^{-1}b$$

例6

A =[1 2 3;4 5 6; 7 8 9; 10 11 12];

[Q,R] = qr(A)

```
>> A =[ 1  2  3;4  5  6; 7  8  9; 10  11  12];
[Q,R] = qr(A)

Q =

   -0.0776   -0.8331    0.5405   -0.0885
   -0.3105   -0.4512   -0.6547    0.5209
   -0.5433   -0.0694   -0.3121   -0.7763
   -0.7762    0.3124    0.4263    0.3439


R =

  -12.8841  -14.5916  -16.2992
         0   -1.0413   -2.0826
         0         0   -0.0000
         0         0         0
```

例7：用QR分解求解线性方程组

```
>> A=[3，1，-4，1;1，-3，0，2;0，2，1，-1;
        1，6，-  1，-3];
>> b=[12，-6，4，0]';
>> [Q，R]=qr(A);
>> x=R\(Q\b)
```

或采用QR分解的第2种格式，命令如下：

```
>> [Q，R，P]=qr(A);
>>x=P*(R\(Q\b))
```

两种得到结果都是

```
x =-16.4444
    20.6667
    -1.1111
    36.2222
```

迭代解法

迭代解法非常适合求解大型系数矩阵的方程组，主要包括 Jacobi迭代法、Gauss-Seidel迭代法、超松弛迭代法和两步迭代法。

1．Jacobi迭代法

对于线性方程组Ax=b，如果A为非奇异方阵，即$a_{ii}\neq 0$ (i=1,2,…,n)，则可将A分解为A=D-L-U，其中D为对角阵，其元素为A的对角元素，L与U为A的下三角阵和上三角阵，于是Ax=b化为：

$$x=D^{-1}(L+U)x+D^{-1}b$$

与之对应的迭代公式为：

$$x^{(k+1)}=D^{-1}(L+U)x^{(k)}+D^{-1}b$$

这就是Jacobi迭代公式。如果序列$\{x^{(k+1)}\}$收敛于x，则x必是方程组Ax=b的解。

Jacobi迭代法的MATLAB函数文件jacobi.m如下：

```
function [y,n]=jacobi(A,b,x0,eps)
if nargin==3      %函数输入参数个数
   eps=1.0e-6;
elseif nargin<3  %至少得输入前3个参数
   error
   return
end
D=diag(diag(A));    %求A的对角矩阵
L=-tril(A,-1);      %求A的 不包括主对角线下三角阵
U=-triu(A,1);       %求A的不包括主对角线上三角阵
B=D\(L+U);
f=D\b;
y=B*x0+f;
n=1;                 %迭代次数
while norm(y-x0)>=eps
   x0=y;
   y=B*x0+f;
   n=n+1;
end
```

例8　用Jacobi迭代法求解线性方程组。设迭代初值为0，迭代精度为10^{-6}。

$$\begin{cases} 10x_1 - x_2 = 9 \\ -x_1 + 10x_2 - 2x_3 = 7 \\ -2x_2 + 10x_3 = 6 \end{cases}$$

在命令中调用函数文件jacobi.m，命令如下：

```
A=[10,-1,0;-1,10,-2;0,-2,10];
b=[9,7,6]';
[x,n]=jacobi(A,b,[0,0,0]',1.0e-6)
        % or [x,n]=jacobi(A,b,[0,0,0]')
        % or x=jacobi(A,b,[0,0,0]')
x =
  0.9958
  0.9579
  0.7916
n =   11
```

2．Gauss-Seidel迭代法

Jacobi迭代公式$Dx^{(k+1)}=(L+U)x^{(k)}+b$可以改进为$Dx^{(k+1)}=Lx^{(k+1)}+Ux^{(k)}+b$，于是得到：

$$x^{(k+1)}=(D-L)^{-1}Ux^{(k)}+(D-L)^{-1}b$$

该式即为Gauss-Seidel迭代公式。

和Jacobi迭代相比，Gauss-Seidel迭代用新分量代替旧分量，精度会高些。

Gauss-Seidel迭代法的MATLAB函数文件gauseidel.m如下：

```
function [y,n]=gauseidel(A,b,x0,eps)
if nargin==3
   eps=1.0e-6;
elseif nargin<3
   error
   return
end
D=diag(diag(A));    %求A的对角矩阵
L=-tril(A,-1);      %求A的下三角阵
U=-triu(A,1);       %求A的上三角阵
G=(D-L)\U;
f=(D-L)\b;
y=G*x0+f;
n=1;                %迭代次数
while norm(y-x0)>=eps
   x0=y;
   y=G*x0+f;
   n=n+1;
end
```

例9　用Gauss-Seidel迭代法求解下列线性方程组。设迭代初值为0，迭代精度为10^{-6}。

$$\begin{cases} 10x_1 - x_2 = 9 \\ -x_1 + 10x_2 - 2x_3 = 7 \\ -2x_2 + 10x_3 = 6 \end{cases}$$

在命令中调用函数文件gauseidel.m，命令如下：

```
A=[10,-1,0;-1,10,-2;0,-2,10];
b=[9,7,6]';
[x,n]=gauseidel(A,b,[0,0,0]',1.0e-6)
                    % or [x,n]=gauseidel(A,b,[0,0,0]')
                    % or x=gauseidel(A,b,[0,0,0]')
x =
   0.9958
   0.9579
   0.7916
n =
   7
```

达到同样的精度，Gauss-Seidel 迭代法比Jacobi迭代法，迭代的次数少，即收敛快----一定条件下正确

例10 分别用Jacobi迭代和Gauss-Seidel迭代法求解下列线性方程组，看是否收敛。

$$\begin{cases} x_1 + 2x_2 - 2x_3 = 9 \\ x_1 + x_2 + x_3 = 7 \\ 2x_1 + 2x_2 + x_3 = 6 \end{cases}$$

命令如下：

```
a=[1,2,-2;1,1,1;2,2,1];
b=[9;7;6];
[x,n]=jacobi(a,b,[0;0;0])
[x,n]=gauseidel(a,b,[0;0;0])
x =
  -27
   26
    8
n =
    4
x =
  NaN
  NaN
  NaN
n = 1012
```

Jacobi迭代法收敛，
Gauss-Seidel迭代法不收敛

有时Gauss-Seidel 迭代法收敛
而Jacobi迭代法发散

实验练习题

分别用Gauss消元法、LU分解、QR分解、Jacobi迭代、Gauss-Seidel迭代法求解下列方程组：

1、求解方程组 $\begin{cases} 4x_1 + 2x_2 - x_3 = 2 \\ 3x_1 - x_2 + 2x_3 = 10 \\ 11x_1 + 3x_2 = 8 \end{cases}$。

2、求解方程组 $\begin{cases} 34x_1 + 17x_2 + 47x_3 + 47x_4 = -2 \\ 42x_1 - 32x_2 + 47x_3 + 4x_4 = 43 \\ -29x_1 - 15x_2 - 26x_3 + 32x_4 = 32 \\ 43x_1 + 9x_2 + 48x_3 - 28x_4 = 47 \end{cases}$。

3、求解方程组 $\begin{cases} 8x_1 + 10x_2 + 6x_3 + 11x_4 = 1 \\ 10x_1 + 16x_2 + 9x_3 + 15x_4 = 8 \\ 6x_1 + 9x_2 + 20x_3 + 13x_4 = 7 \\ 11x_1 + 15x_2 + 13x_3 + 20x_4 = 9 \end{cases}$。

Q & A

- 有什么问题吗？