第四节 线性代数数值方法

一、线性代数方程组的数值解

线性代数方程组在实际问题中应用非常广泛,而且常常有几十个甚至几百个变量,这时利用计算机求解代数方程组是十分必要的.最常用的两种求解线性代数方程组的数值方法是直接法和迭代法.

1. 解线性代数方程组的直接法

若系数矩阵为三角阵,可用下述函数求解

```
function y = sovtri(A,b)
% 解系数为三角矩阵的方程组
% A 的下三角将被忽略
n = length(A);
if length(b) ~ = n
   error('系数矩阵有误。');
   return;
end
y(n) = b(n)/A(n,n);
for i = n - 1: - 1:1
   y(i) = b(i);
   for j = i + 1:n
     y(i) = y(i) - A(i,j) * y(j);
   end
   y(i) = y(i)/A(i,i);
end
y = y';
```

例如

```
A =   2    2    3              b =   3
      0    3    1                   - 5
      0    0    6                     6
x = sovtri(A,b)
x' =   2    - 2    1
```

对于一般的系数矩阵,最常用的是 Gauss 列主元消去法解方程组.在 MATLAB 中提供有函数 rref ($\boldsymbol{B}$),其中 $\boldsymbol{B}$ 为增广矩阵($\boldsymbol{A}$,$\boldsymbol{b}$),该函数返回消去后的

结果,如:

```
B =2  2  3    3
   4  7  7    1
  -2  4  5  -7
y=rref(B)
y =   1  0  0    2
      0  1  0  -2
      0  0  1    1
```

显然,最后一列即为方程组的解.值得注意的是,MATLAB中很多函数及运算都是采用这一方法来实现的.另外,MATLAB还提供了一个演示这一消元过程的函数rrefmovie(**A**);若省略参数**A**,则用一个默认的矩阵进行演示;若提供参数**A**,则该函数用所提供的矩阵**A**演示这一消元过程.同学们可以用这一函数演示各种线性方程组(恰定、超定、不定等)的求解过程.

求解线性方程组还可以使用矩阵分解的办法,如可用函数lu(**A**)将矩阵**A**分解成一个上三角阵和一个下三角阵,然后通过求解两个三角矩阵方程组:$\boldsymbol{Ly}=\boldsymbol{b}$,$\boldsymbol{Ux}=\boldsymbol{y}$,可求得方程组的解.

2. 解线性方程组的迭代法

解线性方程组的直接法虽然能通过有限步求出方程组的精确解(舍入误差忽略),但对于阶数很高且系数矩阵稀疏的线性方程组却存在困难.因而,用迭代法解线性方程组是必要的.迭代法是一种近似方法,且必须判定解序列的收敛性.若方程组是严格对角占优阵,则解序列收敛.下面两个函数分别用简单迭代法与Seidel迭代法求解方程组,并给出一个简单例子,比较一下程序与执行结果,看看两种迭代方法有何不同?

```
function [y,k]=simiter(A,b,c,tol)
 % 简单迭代法求解方程组 Ax=b
 % A 为严格对角占优系数矩阵,b 为常数项
 % c 为初值向量,tol 为容许误差。
n=length(A);
if length(b)~=n
  error('系数矩阵有误');
  return;
end
for i=1:n
  b(i)=b(i)/A(i,i);
  A(i,:)=A(i,:)/A(i,i);
```

```
end
k = 1;
while 1
  y = b;
  for i = 1:n
    for j = 1:n
      if j~ = i
        y(i) = y(i) - A(i,j) * c(j);
      end
    end
  end
  if all(abs(y - c)<tol)
    return;
  end
  c = y;
  k = k + 1;
end
function [y,k] = seiditer(A,b,c,tol)
% Seidel 迭代法求解方程组 Ax = b
% A 为严格对角占优系数矩阵,b 为常数项
% c 为初值向量,tol 为容许误差。
n = length(A);
if length(b)~ = n
  error('系数矩阵有误');
  return;
end
for i = 1:n
  b(i) = b(i)/A(i,i);
  A(i,:) = A(i,:)/A(i,i);
end
k = 1;
while 1
  tol1 = tol/10;
  y = b;
  for i = 1:n
```

```
    for j = 1:n
      if j~ = i
        y(i) = y(i) - A(i,j) * c(j);
      end
    end
    if abs(c(i) - y(i)) > tol1
      tol1 = abs(c(i) - y(i));
    end
    c(i) = y(i);
  end
  if tol1 < tol
    return;
  end
  k = k + 1;
end
```

例如:

```
A =     9   -1   -1      b =   7       c =   0
       -1    8    0            7             0
       -1    0    9            8             0
tol = 0.001
[x k] = simiter(A,b,c,tol)
x =  0.9998
     0.9999
     0.9999
k =      5
[x k] = seiditer(A,b,c,tol)
x =    1.0000
       1.0000
       1.0000
k =      4
```

二、方阵求特征问题的数值方法

前面已经述及用函数 eig(**A**)求矩阵 **A** 的特征值与特征向量的方法,但一般在数值计算中,对于一般矩阵的特征值问题,常采用乘幂法,为了加快收敛速度,用带平移原点的加速收敛法.乘幂法一般只能求绝对值最大的特征值,对于

实对称矩阵的特征值问题，一般常用 Jacbi 法，Jacbi 方法是利用旋转变换的方法计算实对称矩阵的全部特征值与特征向量.

1. 乘幂法求一般矩阵的绝对值最大的特征值及其对应的特征向量

函数 Poweig( )和 Powdeig( )分别是利用一般乘幂法和带原点平移的乘幂法求矩阵 $\boldsymbol{A}$ 的最大特征值及其对应的特征向量.

```
function [y,p,k] = poweig(A,x,tol)
% 乘幂法求绝对值最大特征值及其对应特征向量
% A 为方阵，x 为初值向量
if nargin~ = 3
   tol = 0.00001;
end
x0 = x;
k = 1;
while 1
   y = max(x0);
   x = x0/y;
   x0 = A * x;
   if all(abs(x0 - x * y)<tol)
     p = x;
     return;
  end
  k = k + 1;
end
function [y,p,k] = powdeig(A,x,d,tol)
% 带原点平移的乘幂法求绝对值最大特征值及其对应特征向量
% A 为方阵，x 为初值向量
if nargin~ = 4
   tol = 0.00001;
end
A = A - d * eye(size(A));
x0 = x;
k = 1;
while 1
   y = max(x0);
   x = x0/y;
```

```
    x0 = A * x;
    if all(abs(x0 - x * y)<tol)
      p = x;
      y = y + d;
      return;
    end
    k = k + 1;
  end
```

例如：

```
A =   2    4    9                 x0 =  1
      3    9   15                       1
      4   16   36                       1
[y1, p1, k1] = poweig(A, x0)
y1 =   44.1918
p1 = 0.2558
     0.4480
     1.0000
k1 =      7
[y2, p2, k2] = powdeig(A, x0, 2)
y2 =   44.1918
p2 = 0.2558
     0.4480
     1.0000
k2 =      6
```

再如：

```
B =   8    0    -8                x0 =  1
     18    0   -17                      1
     18    1   -18                      1
[y1, p1, k1] = poweig(B, x0)
y1 =   - 8.0000
p1 = 1.0000
     2.0000
     2.0000
k1 =     10
[y2, p2, k2] = powdeig(B, x0, -1)
```

```
y2 =      -8
p2 =   1
       2
       2
k2 =      4
[y3,p3,k3] = powdeig(B,x0,-2)
y3 =    -8.0000
p3 = 1.0000
     2.0000
     2.0000
k3 =     12
```

由上述结果知，对于矩阵 $\boldsymbol{A}$ 来讲，用带原点平移和不带原点平移的乘幂法效果差异不大.这是因为最大特征值44.191 8与次大特征值2.717 5差异很大，因而迭代次数从7降到6意义不大.但对于矩阵 $\boldsymbol{B}$ 来讲，效果就有所不同，用 $d=-1$ 的带原点平移要比不带原点平移收敛要快得多.从10次迭代降为4次迭代，这是因为绝对值最大的特征值 -8 与绝对值次大的特征值 -1 相对差异要小些，但也同时看到若 d 选择不当，如 d 取 -2，反而会使迭代次数增加，由10增加为12.

2. 实对称矩阵的 Jacbi 法求全部特征值与特征向量

用下述函数 jacbieig( ) 可求得矩阵 $\boldsymbol{A}$ 的全部特征值与特征向量，特征向量是按列存贮的，$\boldsymbol{D}$ 为特征值，$\boldsymbol{P}$ 为特征向量，k 为迭代次数.

```
function [D,P,k] = jacbieig(A,tol)
% Jacbi 法求实对称矩阵 A 的全部特征值与特征向量
if nargin~ = 2
   tol = 0.00001;
end
P = eye(size(A));
k = 1;
while 1
   [m,n] = find(abs(A - diag(diag(A))) = = max(max(abs(A - diag(diag
           (A))))));
   if A(m(1),m(1)) = = A(n(1),n(1))
     phi = pi/4;
   else
     phi = atan(2 * A(m(1),n(1))/(A(n(1),n(1)) - A(m(1),m(1))))/2;
```

```
    end
    Q = eye(size(A));
    Q(m(1), m(1)) = cos(phi);
    Q(n(1), n(1)) = cos(phi);
    Q(m(1), n(1)) = sin(phi);
    Q(n(1), m(1)) = - sin(phi);
    A = Q' * A * Q;
    P = P * Q;
    if max(max(abs(A - diag(diag(A)))))< tol
      D = diag(A);
      return;
    end
    k = k + 1;
  end
```

例如：

```
C =   4    3    2
      3    4   -1
      2   -1    3
[y, p, k] = jacbieig(C)
y =   7.1464
     -0.3649
      4.2184
p =    0.7401   -0.6305    0.2338
       0.6414    0.5572   -0.5275
       0.2023    0.5404    0.8168
k =      7
```

第五节　线性代数应用举例

一、动物的繁殖问题

1. 问题

某农场饲养的某种动物所能达到的最大年龄为 15 岁，将其分成三个年龄组，第一组为 0～5 岁，第二组为 6～10 岁，第三组为 11～15 岁. 动物从第二年龄组起开始繁殖后代，经过长期统计，第二年龄组的动物在其年龄段平均繁殖了 4