第11章 特征值与奇异值

苏 芮
srhello@zju.edu.cn
开物苑4-202

10.1 特征值与奇异值分解

方阵 A 的一组特征值(eigenvalue)和特征向量(eigenvector)是一个标量 λ 和一个非零向量 x，它们满足

$$Ax = \lambda x$$

方阵的特征值-特征向量方程可以写成

$$(A - \lambda I)x = 0, \quad x \neq 0$$

这意味着 $A - \lambda I$ 是奇异的，因此有

$$\det(A - \lambda I) = 0$$

特征值的这种定义就是矩阵 A 的特征方程(characeristic equation)或者特征多项式(characteristic polynomial)，不直接涉及对应的特征向量。特征多项式的次数等于矩阵的阶数。这表示一个 $n \times n$ 的矩阵有 n 个特征值，其中重复的特征值也要计数。和行列式本身一样，特征多项式在理论研究和手算中都是非常有用的，但是不能为鲁棒的数值计算软件提供一个坚实的基础。

令 λ_1，λ_2，……，λ_n 为矩阵 A 的特征值，x_1，x_2，……，x_n 为对应的特征向量组，又令 Λ 是以 λ_n 为对角元的 nxn 对角阵，再令矩阵 X 是由 x_1 为其第 j 列的 nxn 矩阵，那么则有：

$$AX = X\Lambda$$

对应矩阵 **A=[1,5;2,4]，**其特征值为 λ **$_1$=-1，** λ **$_2$=6，**特征向量 **x_1=[-0.928476690885259; 0.371390676354104], x_2=[-0.707106781186547;-0.707106781186547],**

X=[-0.928476690885259, -0.707106781186547; 0.371390676354104, -0.707106781186547]

Lambda=[-1,0;0,6];

上述公式就变成了：

$$\begin{bmatrix} 1 & 5 \\ 2 & 4 \end{bmatrix}\begin{bmatrix} -0.9284 & -0.7071 \\ 0.37139 & -0.7071 \end{bmatrix}=\begin{bmatrix} -0.9284 & -0.7071 \\ 0.37139 & -0.7071 \end{bmatrix}\begin{bmatrix} 1 & 0 \\ 0 & -6 \end{bmatrix}$$

A*X=X*Lambda

上述公式就变成了：

A*X

ans =

```
    0.9285    -4.2426
   -0.3714    -4.2426
```

X*Lambda

ans =

```
    0.9285    -4.2426
   -0.3714    -4.2426
```

10.1 特征值与奇异值分解

奇异向量总可以被选择彼此正交，以使得其列为规范向量矩阵 U 和 V 满足$U^H U = I$ 和

$V^H V = I$，换句话说，U 和 V 均为实矩阵时，他们都是正交矩阵(orthogonal matrix)，为复矩阵时，它们都是酉矩阵(unitary matrix)，于是有：

$$AV = U\Sigma \rightarrow AVV^{-1} = U\Sigma V^{-1} \rightarrow A = U\Sigma V^{-1} = U\Sigma V^H$$

这称为矩阵 A 的奇异值分解（Singular Value Decomposition),记作 SVD。

若 A 是一个 mxn 矩阵，并且 m>n，完整型 SVD 分解中，U 是 mxm 矩阵，Σ是 mxn 主对角阵，V'是 nxn 矩阵。节省内存的 SVD 形式称为紧凑型(economy-sized) SVD，在紧凑型的 SVD 中，只计算前 n 列和Σ的前 n 行，V'是 nxn 矩阵。图中显示了两种形式的形状。

10.1 特征值与奇异值分解

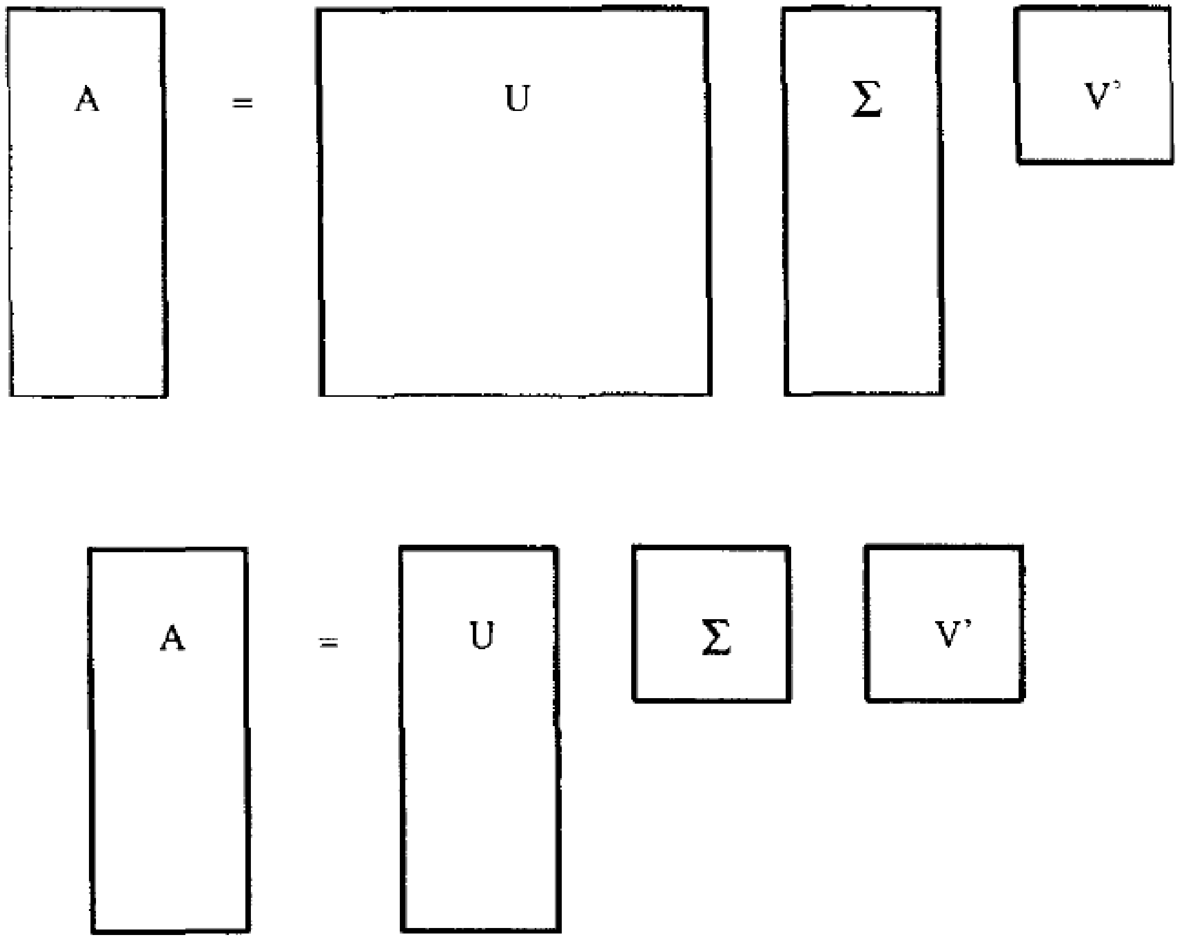


图 10-1 完全和简化的 SVD

10.2 一个简单的例子

$$\det(A-\lambda I)=\lambda^3-6\lambda^2+11\lambda-6=(\lambda-1)(\lambda-2)(\lambda-3)$$

$$\Lambda=\begin{pmatrix}1&0&0\\0&2&0\\0&0&3\end{pmatrix}$$

使用测试矩阵，
运行：

A=gallery(3)

A =

```
 -149   -50  -154
  537   180   546
  -27    -9   -25
```

特征向量矩阵被规范化使其元素均为整数(第一列向量乘 3.16，第二列向量乘以大约 10，第三列向量乘以大约-50）

$$X=\begin{pmatrix}1&-4&7\\-3&9&-49\\0&1&9\end{pmatrix}$$

$$X^{-1}=\begin{pmatrix}130&43&133\\27&9&28\\-3&-1&-3\end{pmatrix}$$

10.2 一个简单的例子

$$\det(A-\lambda I)=\lambda^3-6\lambda^2+11\lambda-6=(\lambda-1)(\lambda-2)(\lambda-3)$$

使用测试矩阵，运行：

```
A=gallery(3)
A =
 -149   -50  -154
  537   180   546
  -27    -9   -25
```

$$\Lambda=\begin{pmatrix}1&0&0\\0&2&0\\0&0&3\end{pmatrix}$$

特征向量矩阵被规范化使其元素均为整数(第一列向量乘 3.16，第二列向量乘以大约 10，第三列向量乘以大约-50）

$$X=\begin{pmatrix}1&-4&7\\-3&9&-49\\0&1&9\end{pmatrix}$$

$$X^{-1}=\begin{pmatrix}130&43&133\\27&9&28\\-3&-1&-3\end{pmatrix}$$

以上这些矩阵给出了这个例子的特征值分解：

$$A=X\Lambda X^{-1}$$

这个矩阵的 SVD 不能通过一些小整数简洁的表示出来，它的奇异值是下面这个方程的所有正根：

$$\sigma^6-668737\sigma^4+4096316\sigma^2-36=0$$

10.2 一个简单的例子

我们直接使用SVD命令从矩阵A中求奇异值。

```
A=gallery(3)
A =
  -149   -50  -154
   537   180   546
   -27    -9   -25

[U,S,V]=svd(A)
U =
   -0.2691   -0.6798    0.6822
    0.9620   -0.1557    0.2243
   -0.0463    0.7167    0.6959

S =
  817.7597         0         0
         0    2.4750         0
         0         0    0.0030

V =
    0.6823   -0.6671    0.2990
    0.2287   -0.1937   -0.9540
    0.6944    0.7193    0.0204
```

注意gallery(3)这个矩阵的特征值1,2,3和奇异值817,2.47,0.03之间存在巨大的差别，这种差别的产生，主要是由于该矩阵的对称性很差导致。

10.3 eigshow

Eigshow 是一个演示程序，演示矩阵及其特征值的关系：

默认的矩阵是 A=[1/4 3/4; 1 1/2]

我们直接求它的特征值和特征向量

[V,D]=eig(A)

V =

-0.7071 -0.6000

0.7071 -0.8000

D =

-0.5000 0

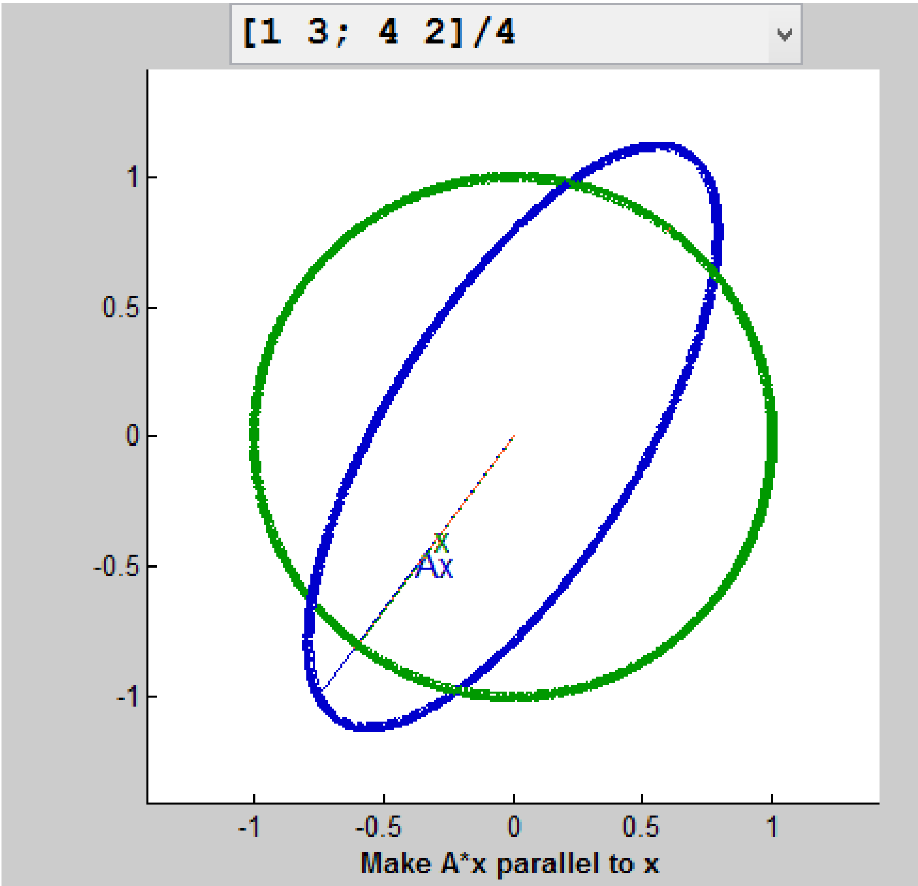
0 1.2500

也就是一个特性向量 $x_1 = \begin{bmatrix} -0.7071 \\ 0.7071 \end{bmatrix}$，特征值 $\lambda_1 = -0.5$

另一个特性向量 $x_2 = \begin{bmatrix} -0.6000 \\ -0.8000 \end{bmatrix}$，特征值 $\lambda_1 = 1.25$

而我们要求的是 $[A]\{X\} = \lambda\{X\}$

这里假设我有一个初始的 x=[1,0]′，反复选择 x，然后按照运算规则 Ax 向量也在旋转，其实我们要求的就是看当 x 取到什么值的时候，Ax 和 x 本身平行



我们可以看到当 x=[-0.6;-0.8]的时候，它们是平行的，这个时候 x 就是其特征向量，特征值就是 Ax 和 x 长度的比值，大约等于 1.25，和前面求出来的特征值相符合。

10.4 特征多项式

令 A 是一个 20x20 的对角矩阵，其对角线上的元素分别为 1,2，...,20,显然，A 的特征值就是其对角线上的元素，但是，它的特征多项式 det(A- λ I)却是：

$$
\begin{gathered}
\lambda^{20}-210\lambda^{19}+20615\lambda^{18}-1256850\lambda^{17}+53327946\lambda^{16}\\
-1672280820\lambda^{15}+40171771630\lambda^{14}-756111184500\lambda^{13}\\
+11310276995381\lambda^{12}-135585182899530\lambda^{11}\\
+1307535010540395\lambda^{10}-10142299865511450\lambda^{9}\\
+63030812099294896\lambda^{8}-311333643161390640\lambda^{7}\\
+1206647803780373360\lambda^{6}-3599979517947607200\lambda^{5}\\
+8037811822645051776\lambda^{4}-12870931245150988800\lambda^{3}\\
+13803759753640704000\lambda^{2}-8752948036761600000\lambda\\
+2432902008176640000
\end{gathered}
$$

其中 $-\lambda^{19}$ 的系数为 210，是所有特征值的和，λ^{0} 的系数为 20!，也就是所有特征值的积，其他的系数都是特征值积的各种和。其实就是 $(x-a)(x-b)=x^2-(a+b)x+ab$ 的扩展而已。

10.4 特征多项式

我们给出多项式的所有系数是为了强调，用它们进行任何浮点运算都可能引入很大的舍入误差，使用 IEEE 的浮点数会改变其中的五个值，例如 λ^4 系数的最后三位数字就会从 776 变到 392，对于 16 位有效数字，用浮点表示系数得到的该多项式根的准确值如下：

```
1.00000000000000
2.00000000000096
2.99999999986640
4.00000000495944
4.99999991473414
6.00000084571661
6.99999455544845
8.00002443256894
8.99992001186835
10.00019696490537
10.99962843024064
12.00054374363591
12.99938073455790
14.00054798867380
14.99962658217055
16.00019208303847
16.99992773461773
18.00001875170604
18.99999699774389
20.00000022354640
```

我们发现，使用双精度浮点数来存储这个特征多项式的系数，改变的某些特征值的计算值在第五位有效数字之内。

这个特殊的多项式是由J.H.Wilkinson在1960年提出，他对这个多项式设置的扰动与我们不同，但是他的观点和我们相同，就是通过幂形式来表示多项式的方法，对于通过多项式求根或者对应矩阵的特征值是不能令人满意的。

10.5 对称矩阵和埃米尔特(Hermitian)矩阵

若实矩阵与其自身的转置相等，即 **A=A^T^**，那么该矩阵是对称的。若复矩阵与其自身的共轭转置相等，即 **A=A^H^**，则称它是埃尔米特矩阵。实对称（**Symmetric Matrix**）的特征值和特征向量都是实的，特征向量矩阵可被选择为正交阵，若实矩阵 **A=A^T^**，那么其特征值分解为：

$$A = X\Lambda X^{\mathrm{T}}$$

其中有 $XX^{\mathrm{T}} = I = X^{\mathrm{T}}X$，尽管 **Hermit** 矩阵的特征向量一定是复数向量，但是其特征值可以证明是实数，此外，特征向量矩阵可以选择为酉矩阵，若复矩阵 **A** 有 **A=A^H^**,那么其特征值分解为：

$$A = X\Lambda X^{\mathrm{H}}$$

10.5 对称矩阵和埃米尔特(Hermitian)矩阵

这里我们假定一个实对称矩阵 A，有 $A=A^T$

A=[1,2;2,1]

A =

1	2
2	1

[V,G]=eig(A)

V =

-0.7071	0.7071
0.7071	0.7071

G =

-1	0
0	3

这里的V是特征向量矩阵，也就是上式中的X，可以验证，特征向量矩阵V就是一个正交阵：

V*V

ans =

1.0000	0
0	1.0000

我们再看$A=X\Lambda X^T$公式是否成立

V*G*V'

ans =

1.0000	2.0000
2.0000	1.0000

10.6 特征值的灵敏度和精度

有的矩阵的特征值对扰动（**Perturbation**）很敏感，矩阵元素的微小变化就可能导致特征值的巨大变化，由浮点算法计算特征值的舍入误差也可能是特征值的变化被放大。

为了简单的理解灵敏度(**Sensitivity**)，假如矩阵 **A** 具有线性无关特征向量完备集（**full set of eigenvectors**）,并且有特征值分解：

$$A = X\Lambda X^{-1}$$

也可以重新表达成：

$$\Lambda = X^{-1}AX$$

加上由于Λ的变动δΛ导致 **A** 的变动δ**A:**

$$\Lambda + \delta\Lambda = X^{-1}(A+\delta A)X$$

则有：

$$\delta\Lambda = X^{-1}\delta AX$$

$$\|\delta\Lambda\| \le \|X^{-1}\|\|\delta A\|\|X\| = \|X^{-1}\|\|X\|\|\delta A\| = \kappa(X)\|\delta A\|$$

$\kappa(X)$就是第二章线性方程组中引入的矩阵条件数，关键因素是特征向量矩阵 **X** 的条件，

而非 **A** 自身的条件。扰动$\|\delta A\|$将被放大$\kappa(X)$表征在$\|\delta\Lambda\|$中，由此可以推出大体上正确的

结论：特征值的灵敏度可以用特征向量矩阵的条件数来进行估计的。

10.6 特征值的灵敏度和精度

我们可以利用函数condest来估计特征值向量的条件数。

```
A=gallery(3)
[X,lambda]=eig(A)
condest(X)

A=gallery(3)

A =
  -149   -50  -154
   537   180   546
   -27    -9   -25
```

```
[X,lambda]=eig(A)
X =
    0.3162   -0.4041   -0.1391
   -0.9487    0.9091    0.9740
   -0.0000    0.1010   -0.1789
lambda =
    1.0000         0         0
         0    2.0000         0
         0         0    3.0000
condest(X)
ans =
  1.2002e+03
```

这就是意味着gallery(3)中一个扰动可能会被放大到1.2E3倍后形成特征值扰动。这说明，gallery(3)的特征值为轻微坏条件（badly conditioned）。

10.7 奇异值的灵敏度和精度

奇异值灵敏度比特征值灵敏度容易表征得多，奇异值问题总是有较好的条件，扰动分析涉及下列所示的方程：

$$\Sigma + \delta\Sigma = U^H (A + \delta A) V$$

由于 U 和 V 是正交阵或者酉矩阵，所以它们的范数不变，这是正交矩阵的保范性，结果为

$\|\delta\Sigma\| = \|\delta A\|$。任何矩阵中的任何大小的扰动会引起奇异值中“大小相仿”的扰动，没有必要为奇异值定义条件数，原因是它们总是等于 1。MATLAB 的 SVD 计算出的奇异值总能达到满精度（full accuracy）水平。

我们必须小心翼翼的理解大小相仿和满精度的含义，扰动和精度是相对于矩阵范数或如下最大奇异值测度的：

$$\|A\|_2 = \sigma_1$$

那些较小奇异值的精度都是相对于最大奇异值测度的，若奇异值大小的变化幅度有几个量级，那么较小奇异值就不会有相对于自身的满精度。

10.8 约当型和舒尔型

特征值分解是力图找到一个对角矩阵 **Λ** 和一个非奇异矩阵 **X**,使得：

$$A = X\Lambda X^{-1}$$

在此，特征值分解存在两个困难，理论困难是：这种分解并非始终存在。数值困难是：即使这种分解存在，也可能找不到可供稳健计算的基。

分解不存在的那种困难的解决办法是，找尽可能接近对角阵的矩阵替代，这就引出了约当标准型（**Jordan canonical form, JCF**)。稳健性困难的解决方法是，用三角阵（**triangular matrix**）替代对角阵，并使用正交或酉变换，这就引出了舒尔型（**Schur form**）。

退化矩阵（**defective matrix**）是：至少包含一个线性无关特征向量集不完备的多重特征值的矩阵，**Gallery(5)**是人为构造的一个退化矩阵，它的五个特征值都是 **0**，然而，当我们使用特征值计算命令计算时，由于浮点运算而产生了很大误差，特征值并不为 **0.**

10.8 约当型和舒尔型

```
>> gallery(5)

A =

    -9       11      -21       63     -252
    70      -69      141     -421     1684
  -575      575    -1149     3451   -13801
  3891    -3891     7782   -23345    93365
  1024    -1024     2048    -6144    24572
```

```
>> eig(A)
ans =
  -0.0347 + 0.0258i
  -0.0347 - 0.0258i
   0.0138 + 0.0401i
   0.0138 - 0.0401i
   0.0419 + 0.0000i
```

约当标准型 JCF

$$A = XJX^{-1}$$

JCF 的满意的数值替代是舒尔型，任何矩阵都可以通过酉相似变换成为上三角形式：

$$B = T^H AT$$

矩阵 A 的特征值位于舒尔型 B 的对角线上，由于酉矩阵具有最完美的条件，所以它们不会放大任何误差。

10.8 约当型和舒尔型

例如：

```
A=gallery(3)
[T,B]=schur(A)

T =
   -0.3162    0.6529    0.6882
    0.9487    0.2176    0.2294
    0.0000   -0.7255    0.6882


B =
    1.0000   -7.1119  815.8706
         0    2.0000   55.0236
         0         0    3.0000
```

B的对角线上元素就是A的特征值，若A为对称矩阵，那么B应该是对角阵，因此，舒尔型B中非对角线上的大值元素可衡量A矩阵对称性的缺失程度。

```
A=gallery(5);
[T,B]=schur(A)
T =
    0.0000   -0.3485    0.6881   -0.6362    0.0147
   -0.0206   -0.7780    0.1658    0.6049   -0.0299
    0.1397   -0.5191   -0.6973   -0.4681    0.0755
   -0.9574   -0.0607   -0.1109   -0.0924   -0.2427
   -0.2519    0.0065    0.0213    0.0418    0.9666
B =
   1.0e+05 *
   -0.0000    0.0036   -0.0123    0.0252   -1.0099
   -0.0000   -0.0000    0.0001   -0.0001    0.0045
         0         0   -0.0000    0.0000   -0.0011
         0         0         0    0.0000    0.0004
         0         0         0   -0.0000    0.0000
```

这个时候我们看到gallery(5)的舒尔矩阵的对角线的值为5个0，另外，右上角的值非常大，可见对称性严重缺失。

10.9 QR算法

QR 算法是重要性最高、应用最广泛、最成功的工程计算工具，在 MATLAB 的数学内核中，包含了不同实施方法的 QR 算法的几个变种，它们分别用于计算实数对称、非对称矩阵的特征值，计算复数矩阵的 Q 酉矩阵和 R 上三角阵，用于计算一般矩阵的奇异值，这些 M 函数通常用于多项式的求根、特殊线性方程组求解、稳定性评估中。

QR 算法的基础在于反复使用 QR 分解，Q 表示正交或酉矩阵，R 表示右（或上）三角矩阵，MATLAB 中的函数能把任何矩阵，实矩阵或者复矩阵，方阵或非方阵分解成为正交规范矩阵 Q 和上三角矩阵 R 的乘积。

利用 QR 函数，把 QR 算法最简单变种“单步位移法”编写成 MATLAB 的单行码，先运行：

```
A=gallery(3)
n=size(A,1)
I=eye(n,n)
```

此后，单步位移 QR 迭代的每步可执行如下：

```
s=A(n,n); [Q,R]=qr(A-s*I); A=R*Q+s*I
```

10.9 QR算法

假如你把上述命令写在一行中，就可以反复迭代，s 就是位移量，可以加速收敛，**QR** 分解是矩阵三角化：

$$A-sI=QR$$

两边分别乘以：Q^{-1}

$$Q^{-1}(A-sI)=Q^{-1}QR=R$$

存在有：

$$Q^{-1}=Q^{T}$$

代入得到：

$$R=Q^{T}(A-sI)$$

然后，进行 **QR** 反序相乘就可以恢复出特征值，原因是：

$$RQ+sI=Q^{T}(A-sI)Q+sI=Q^{T}AQ-Q^{T}\cdot sI\cdot Q+sI$$

所以新 **A** 阵正交相似于原 **A** 阵，每次迭代都能有效的把下三角的一些质量转移到上三角，同时保持特征值不变，随着迭代的反复进行，该矩阵就不断逼近上三角矩阵，而使得特征值恰好显示于对角线上。

10.9 QR算法

A=gallery(3)

A =

-149	-50	-154
537	180	546
-27	-9	-25

A的QR分解结果是：

[Q,R] = qr(A)

Q =

-0.2671	-0.7088	0.6529
0.9625	-0.1621	0.2176
-0.0484	0.6865	0.7255

R =

557.9418	187.0321	567.8424
0	0.0741	3.4577
0	0	0.1451

利用了QR分解算法，将矩阵A-s*I的进行QR分解，不断逼近上三角矩阵，特征值就恰好显示于上三角矩阵。

分步迭代，第一次迭代：

A =

28.8263	-259.8671	773.9292
1.0353	-8.6686	33.1759
-0.5973	5.5786	-14.1578

第二次迭代：

A =

11.8153	-128.8377	-807.3201
0.0629	0.2505	-13.9212
0.1140	-1.3579	-6.0659

第三次迭代：

A =

6.2154	-64.6615	815.1030
-0.0491	1.6089	10.8801
-0.0271	0.3363	-1.8244

第五次迭代：

A =

2.7137	-10.5427	-814.0932
-0.0767	1.4719	-76.5847
0.0006	-0.0039	1.8144

第十次迭代

A =

3.0716	-7.6952	802.1201
0.0193	0.9284	158.9554
-0.0000	0.0000	2.0000

其中一个特征值已经满精度算出，并且对角线下紧挨的元素已经为0，所以，只用再计算左上的2x2的矩阵了。

10.10 QR算法演示界面eigsvdgui

eigsvdgui Demonstrate computation of matrix eigenvalues and singular values. eigsvdgui shows three variants of the QR algorithm.

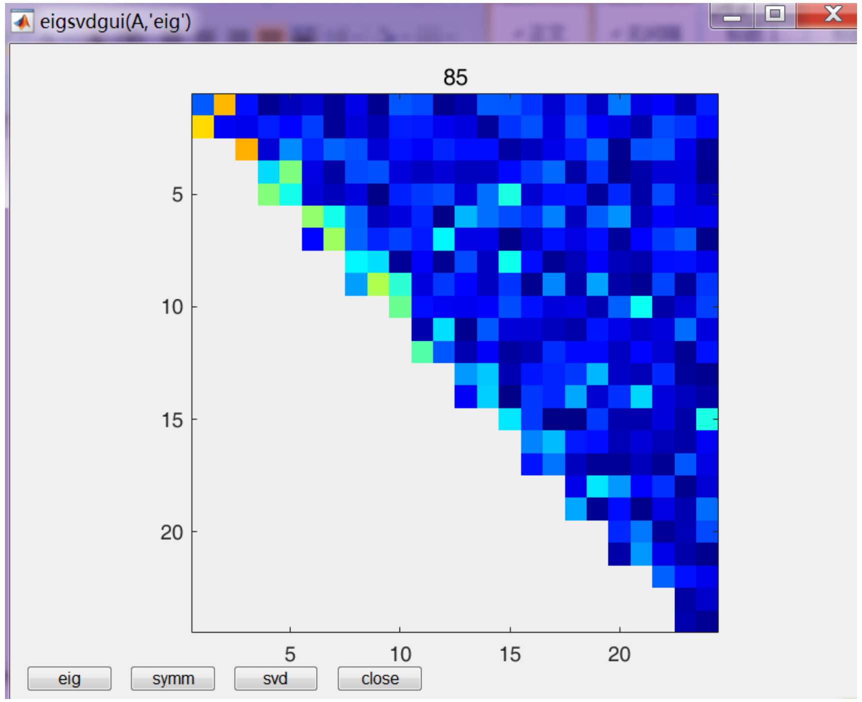
eigsvdgui(A) for square, nonsymmetric A, or eigsvdgui(A,'eig'), reduces A to Hessenberg form, then applies a double-shift, eigenvalue-preserving QR algorithm. The result is the real Schur block upper triangular form, with one-by-one diagonal blocks for real eigenvalues and two-by-two diagonal blocks for pairs of complex eigenvalues.

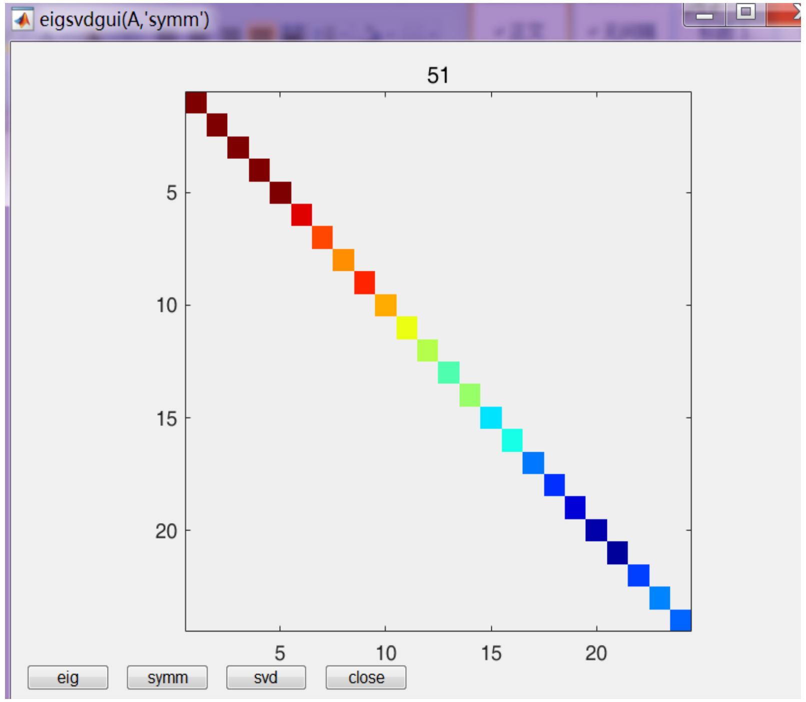
eigsvdgui(A) for square, symmetric A, or eigsvdgui(A,'symm'), reduces the symmetric part, (A+A')/2, to tridiagonal form, then applies a single-shift, eigenvalue-preserving QR algorithm. The result is a diagonal matrix containing the eigenvalues, which are all real.

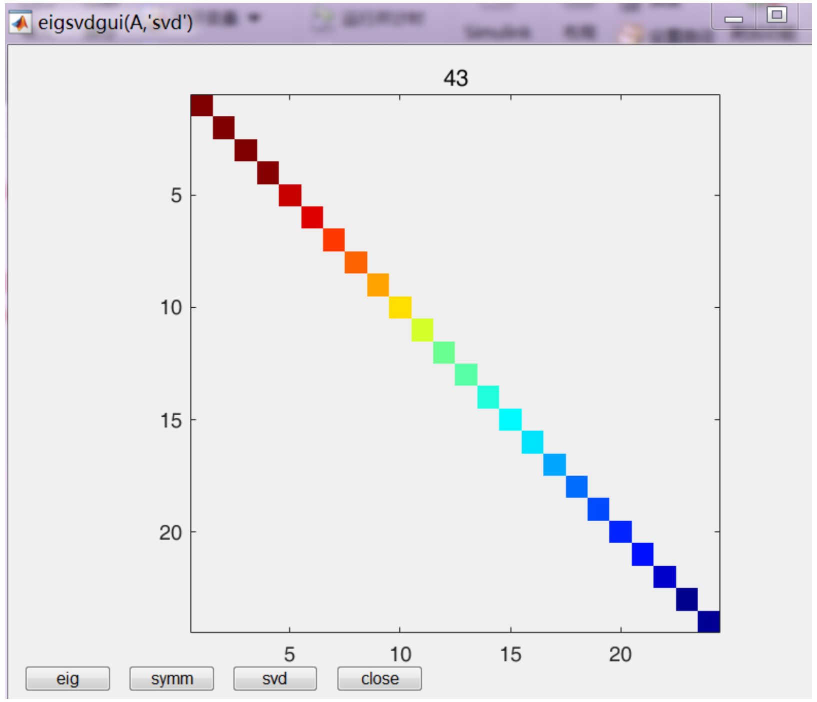
eigsvdgui(A) for rectangular A, or eigsvdgui(A,'svd'), reduces A to bidiagonal form, then applies a single-shift QR algorithm that preserves the singular values. The result is a diagonal matrix containing the singular values.

If A is symmetric and positive definite, the three variants compute the same final diagonal matrix by three different algorithms.

10.10 QR算法演示界面eigsvdgui







一直做保持特征值不变的相似变换，这是第一阶段，对角线下方的元素均变换到0，形成上三角矩阵。

第二个阶段是利用QR算法在第一次对角线上引入0，实非对称矩阵一般都有一些复特征值，所以不可能被变换成上三角舒尔型

SVD

10.11 主成分分析

主成分（或主分量）分析（**Principal Component Analysis, PCA**）,通过一系列简单矩阵的计算来近似一个一般矩阵，这里是指秩为一的矩阵，所有的行都是差一个倍数，所有的列也是，令 **A** 是一个任意实 **mxn** 矩阵，简化的 **SVD** 为：

$$A = U\Sigma V^T$$

也可以重写成：

$$A = E_1 + E_2 + \cdots + E_p$$

其中 **p=min(m,n)**，分量矩阵 E_k 是秩一外积：

$$E_k = \sigma_k u_k v_k^T$$

E_k 的每一列都是矩阵 **U** 第 **k** 列 u_k 的倍数，每一行是矩阵 **V** 第 **k** 列的转置 v_k^T 的倍数，这些分量彼此还是正交的。

$$E_j E_k^T = 0，\ j \neq k$$

每个分量矩阵的范数是对应的奇异值

$$\|E_k\| = \sigma_k$$

10.11 主成分分析

每个分量矩阵 E_k 在重建矩阵 A 时所起的作用由奇异值 σ_k 决定

如果只取前面 r<p 项求和，

$$A_r = E_1 + E_2 + \cdots + E_r$$

结果是原始矩阵 A 称为 r 的近似矩阵，事实上，A_r 是所有秩为 r 的矩阵中与 A 最接近，可以证明这种近似误差是：

$$\|A - A_r\| = \sigma_{r+1}$$

因为奇异值是降序的，所以近似的精度随着秩的增加而增加。

PCA 的应用非常广泛，它的描述有很多种，可能最常见的是用交叉乘积矩阵 $A^T A$ 的特征值和特征向量来描述，因为：

$$A^T AV = V\Sigma^2$$

矩阵 **V** 的列就是 $A^T A$ 的特征向量，矩阵 **U** 的列向量，经过奇异值变换后，从下式获得：

$$U\Sigma = AV$$

矩阵 **A** 通常通过减去列的平均值然后除以它们的标准差来标准化(**standardized**)，这样交叉乘积矩阵就是相关矩阵了。

10.11 主成分分析

PCA的简单例子，一个没有经过修改的矩阵A，假设我们度量六个物体的高度和重量并得到下面的数据：

A=[47,15;93,35;53,15;45,10;67,27;42,10]

A =

47	15
93	35
53	15
45	10
67	27
42	10

我们期待发现高度和重量的关联，认为存在一个潜在的分量，比如“尺寸”，它能够同时预测高度和重量。

[U,S,V]=svd(A,0)

U =

-0.3153	0.1056
-0.6349	-0.3656
-0.3516	0.3259
-0.2929	0.5722
-0.4611	-0.4562
-0.2748	0.4620

S =

156.4358	0
0	8.7658

V =

-0.9468	0.3219
-0.3219	-0.9468

sigma=diag(S)

sigma =

156.4358
8.7658

注意到σ 1=156.4，比σ2=8.7要大很多，A的秩一近似为：

E1=sigma(1)*U(:,1)*V(:,1)'

E1 =

46.7021	15.8762
94.0315	31.9657
52.0806	17.7046
43.3857	14.7488
68.2871	23.2139
40.6964	13.8346

E2=sigma(2)*U(:,2)*V(:,2)'

E2 =

0.2979	-0.8762
-1.0315	3.0343
0.9194	-2.7046
1.6143	-4.7488
-1.2871	3.7861
1.3036	-3.8346

B=E1+E2

B =

47.0000	15.0000
93.0000	35.0000
53.0000	15.0000
45.0000	10.0000
67.0000	27.0000
42.0000	10.0000

可以发现A=B=E1+E2

换句话说，潜在的单一主分量是

size=sigma(1)*U(:,1)

size =

-49.3269
-99.3163
-55.0076
-45.8240
-72.1250
-42.9837

这个分量就是降维成一维后的判断标准。

10. 11 主成分分析

由下式，这两个被测量就可以很好的近似

height=size*V(1,1)

height =
   46.7021
   94.0315
   52.0806
   43.3857
   68.2871
   40.6964

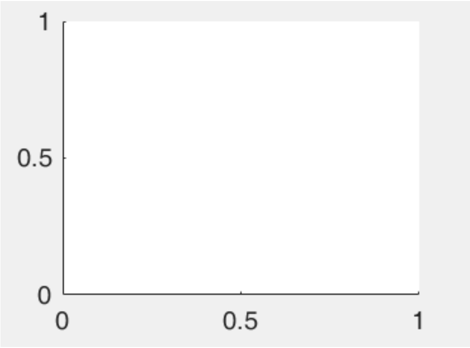
weight=size*V(2,1)

weight =
   15.8762
   31.9657
   17.7046
   14.7488
   23.2139
   13.8346

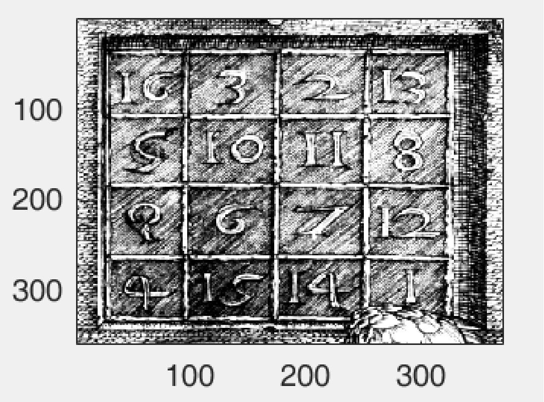
一个更大的示例教学涉及数字图像处理技术（Digital Image Processing）,运行以下语句：

load detail

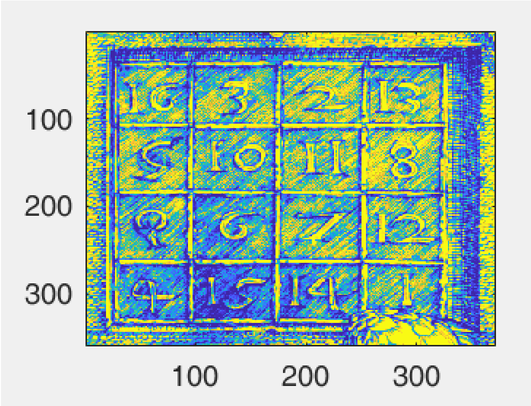
subplot(2,2,1)



colormap(gray(64))



image(X)



axis image, axis off

r=rank(X)

r =

   359

title(['rank=','int2str(r)])

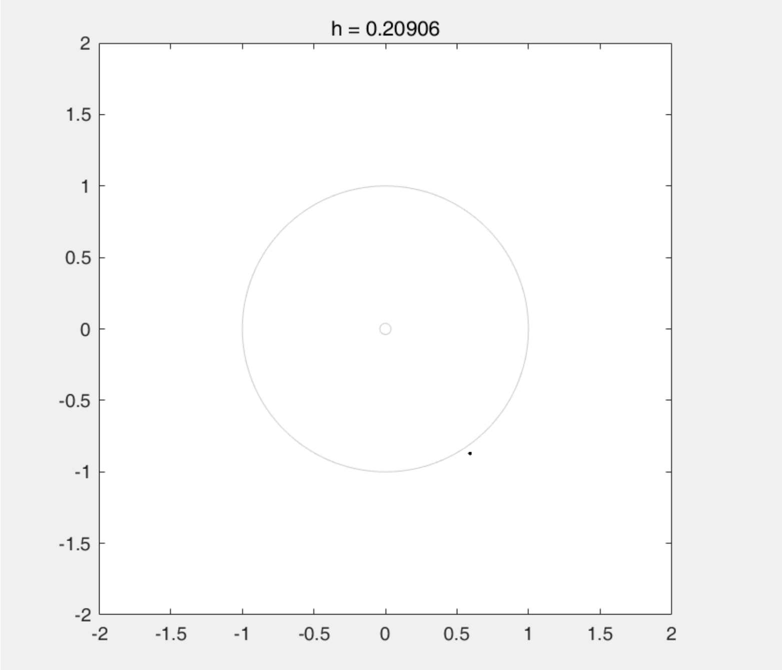
[U,S,V]=svd(X,0)

sigma=diag(S)

10.12 圆生成器

下面的算法曾应用于带图像显示仪的主机画圆

```
x=32768
y=0
L: load y
shift right 5 bits
add x
store in x
change sign
shift right 5 bits
add y
store in y
plot x y
go to L
```

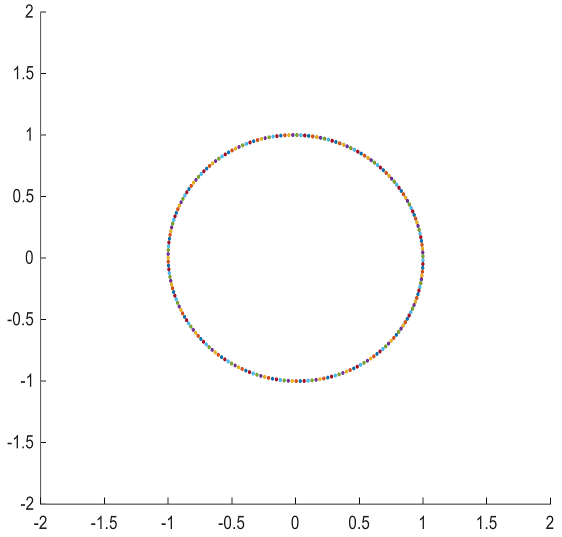


画圆的MATLAB的代码，注意，和教材上的代码有所不同，直接使用教材的代码是跳动的，不能成圆，需要改进，代码主要加了：

1）加了坐标轴的区间限制，set语句

2）加了显示点保留的限制，hold on语句

```
h=1/32;
x=1;
y=0;
while 1
  x=x+h*y;
  y=y-h*x;
  plot(x,y,'.');
  set(gca,'XLim',[-2 2]);
  set(gca,'YLim',[-2 2]);
  drawnow
  hold on
end
```



执行Circlegen能让我们用不同的步长h进行试验，默认h=0.20906，也可以取不同的h，比如circlegen(0.1)

10.12 圆生成器

假如我们用（xn,yn）来表示生成的第 n 个点，那么迭代的过程就是

$$x_{n+1} = x_n + hy_n$$

$$y_{n+1} = y_n - hx_{n+1}$$

如果将第二个方程中的 x_{n+1}，使用第一个方程中的公式代入，得到：

$$x_{n+1} = x_n + hy_n$$

$$y_{n+1} = y_n - hx_{n+1} = y_n - h(x_n + hy_n) = -hx_n + (1-h^2)y_n$$

把它转换成矩阵-向量表示方式，用 x_n 记述第 n 点处的 2 维向量（其实就是用 x_n 代替 $(x_n, y_n)'$），用 A 记述成圆(Circle generator)矩阵：

$$A = \begin{pmatrix} 1 & h \\ -h & 1-h^2 \end{pmatrix}$$

迭代过程记为：

$$x_{n+1} = Ax_n$$

进一步就可以写成：

$$x_{n+1} = A^n x_0$$

对于大多数矩阵 A 而言，A^n 的性状由它的特征值决定。

可生成对角(diagonal)特征值矩阵 A 及其对应的特征向量矩阵 X,使得：

$$AX = X\Lambda$$

若有 X^{-1}存在，则有：

$$A = X\Lambda X^{-1}$$

并且有：

$$A^n = X\Lambda^n X^{-1}$$

因此， A^n 有界的条件是：特征向量矩阵非特异，并且作为 A 阵对角元的特征值$|\lambda_k| \le 1$，

在此，有一个简单的试验，运行下面的单行代码：

```
h=2*rand, A=[1 h; -h 1-h^2], lambda=eig(A), abs(lambda)
```

```
h=2*rand, A=[1 h; -h 1-h^2], lambda=eig(A), abs(lambda)
h =
    1.6294
A =
    1.0000    1.6294
   -1.6294   -1.6551

lambda =
  -0.3275 + 0.9448i
  -0.3275 - 0.9448i
ans =
     1
     1
```

反反复复操作，可以相信如下结论：

对于 0<h<2 区间内的任何 h，成圆矩阵 A 的特征值都是模为 1 的复数。

10.12 圆生成器

符号工具包能给该结论提供某些帮助。

```
syms h
A=[1 h; -h 1-h^2]
lambda=eig(A)

A =
[ 1,         h]
[ -h, 1 - h^2]

lambda =
  1 - h^2/2 - (h*((h - 2)*(h + 2))^(1/2))/2
  (h*((h - 2)*(h + 2))^(1/2))/2 - h^2/2 + 1

ans =
  abs((h*((h - 2)*(h + 2))^(1/2))/2 + h^2/2 - 1)
  abs((h*((h - 2)*(h + 2))^(1/2))/2 - h^2/2 + 1)
```

若 $|h|<2$，

```
d=det(A)
d =
1

d=simple(prod(lambda)
```

特征值 λ 就是复数并且它们的乘积为 1，所以一定有：

因为有：

$$\lambda = 1 - h^2/2 \pm h\sqrt{-1+h^2/4}$$

这个表达式不是很清楚，似是而非，当我们采用三角函数定义时，有：

$$\cos\theta = 1 - h^2/2$$

$$\sin\theta = h\sqrt{1-h^2/4}$$

则有：

$$\lambda = cos\theta \pm isin\theta$$

总之，以上符号证明了 $|h|<2$ 时，成圆矩阵的特征值为：

$$\lambda = e^{\pm i\theta}$$

这两个特征值各异，因此 X 阵必定非奇异，进而有：

$$A^n = X\begin{pmatrix} e^{i\cdot n\theta} & 0 \\ 0 & e^{-i\cdot n\theta} \end{pmatrix}X^{-1}$$

如果步长 h 取值恰好使得 $\theta = 2\pi/p$， p 为整数，那么该算法只生成 p 个离散点。

成圆代码如何才越来越接近于真圆呢？事实上，成圆代码产生的是椭圆，随着步长 **h** 的增长，椭圆就越来越接近于真圆，椭圆的纵横比(**aspect ratio**)是它的主、次轴的比。

如下 **2x2** 的微分方程组：

$$\dot{x} = Qx$$

其中

$$Q = \begin{pmatrix} 0 & 1 \\ -1 & 0 \end{pmatrix}$$

的解为一个圆

$$x(t) = \begin{pmatrix} \cos t & \sin t \\ -\sin t & \cos t \end{pmatrix} \cdot x(0)$$

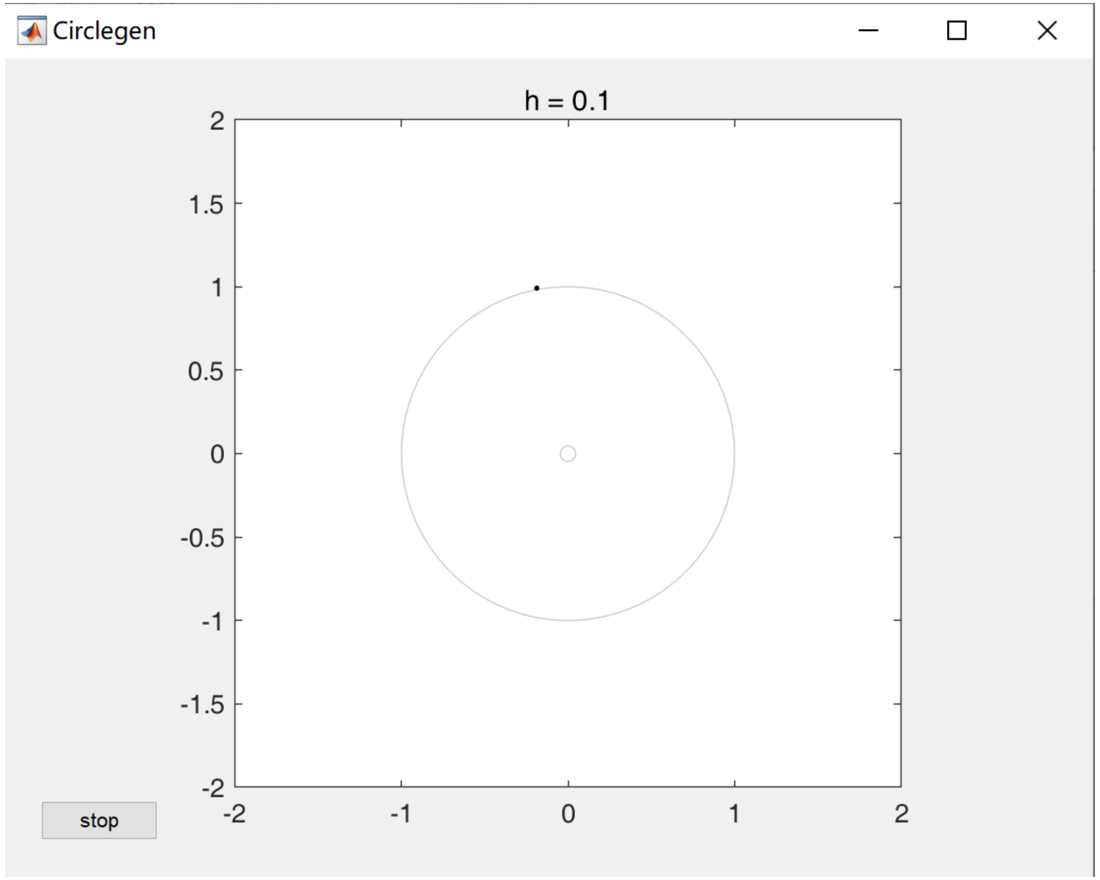
所以迭代矩阵

$$\begin{pmatrix} \cos h & \sin h \\ -\sin h & \cos h \end{pmatrix}$$

可以生成真圆，**cosh** 和 **sinh** 的泰勒级数表明，我们成圆算法的迭代矩阵：

$$A = \begin{pmatrix} 1 & h \\ -h & 1-h^2 \end{pmatrix}$$

会随着 **h** 的减小而逼近于真圆。



感谢聆听,欢迎讨论！