高等数值算法与应用(5)

－ 特征值与奇异值 (chap10) －

喻文健

Outline

- 矩阵的特征值与奇异值分解
- 特征值计算问题的敏感性
- 奇异值分解的存在性与性质
- 计算特征值、奇异值的算法
- 低秩近似与主分量分析
- 圆生成器

特征值与奇异值分解

- 特征值相关概念
- 奇异值相关概念
- 例子与Matlab

特征值与奇异值

- 方阵A的特征值

$$Ax = \lambda x$$

λ为特征值(eigenvalue)，
非零向量x为(右)特征向量(eigenvector)

  - 若矩阵表示向量空间到自身的变换, 则特征值很重要
  - 特征向量有无穷多. 实对称阵, 一般设 $\|x\|_2 = 1$ 的为标准

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

  - 特征方程: $\det(\lambda I - A) = \lambda^n - \left(\sum_{j=1}^{n} a_{jj}\right)\lambda^{n-1} + \cdots = 0$ (共轭成对)
  - $\Longrightarrow$ n阶方阵有n个特征值(含重根). 一般, 可能是虚数

$$AX = X\Lambda \xrightarrow{\text{若n个特征向量线性无关}} A = X\Lambda X^{-1}$$

(特征值分解)

注意前提条件!

  - 相似变换 $B = T^{-1}AT$ 保特征值不变
  - 实对称阵A：特征值均为实数，可正交对角化 $A = Q\Lambda Q^T$
  - 实矩阵A: 实特征值⇔实特征向量，虚特征值⇔非实特征向量
  - 所有特征值集合称为特征值谱$\lambda(A)$, 则$\lambda(A) = \lambda\left(A^T\right)$

特征值与奇异值

- $m\times n$矩阵的奇异值

与矩阵的奇异程度有关

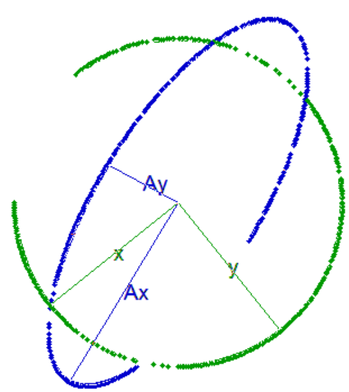
  - $\begin{cases} Av=\sigma u \\ A^H u=\sigma v \end{cases}$ 非负实数σ为奇异值(singular value)
    非零向量u, v为左/右奇异向量(singular vector)
  - $A^H=\overline{A}^T$，即A的厄米特转置 (Matlab中用A'计算)
  - 若矩阵表示一个空间到另一个空间的变换，奇异值很重要 (如超定/欠定方程组)
  - 奇异向量也无穷多，但总标准化为$\|u\|_2=\|v\|_2=1$
  - 设m≥n，若有n个正交的向量$\{v_i\}$，使$\{Av_i\}$也两两正交

即u_i, v_i为左/右奇异向量对

$\Rightarrow AV=U\Sigma$，Σ为对角阵，U的列两两正交 $\Rightarrow A^T U=V\Sigma$

对角元为奇异值

  - eigshow演示程序
  - 2阶方阵的特征向量/奇异向量
  - 找正交的两个向量{x, y}，使它们是奇异向量



思考:
Ax, Ay (u_1, u_2)总是沿椭圆的轴方向?

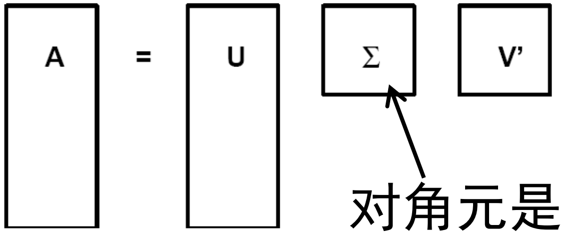
特征值与奇异值

(对一般复矩阵，酉正交)

- 设m≥n，能找到n个正交向量$\{v_i\}$，使$\{Av_i\}$也两两正交 ➡ $AV=U\Sigma$，$A^HU=V\Sigma$

  确实总能找到！（后面证明）
- 简化奇异值分解 $A=U\Sigma V^H$
- V为酉阵($VV^H=I$); U为mxn阵，酉阵前n列
- 可将U扩充列为酉阵(Σ加零行)，得

  $A=U\Sigma V^H$ (**SVD**: singular value decompositon)



调整酉阵U，V列的顺序，使：$\sigma_1 \ge \sigma_2 \ge \sigma_3 \ge \cdots \ge 0$ (A实矩阵：正交阵)

- 例子与Matlab (特征值)

$$A=\begin{pmatrix} -149 & -50 & -154 \\ 537 & 180 & 546 \\ -27 & -9 & -25 \end{pmatrix}$$

  - D为对角阵，V为单位长度特征向量组成的矩阵 (可能奇异!)

```
>> A = gallery(3)
>> eig(A)
ans =
    1.0000
    2.0000
    3.0000
>>[V, D]=eig(A)  % AV=VD
```

例子与Matlab

- 例子与Matlab (奇异值)
  - s= svd(A);   [U,S,V]= svd(A)
  - 注意奇异值与特征值的不同
  - 如果A为实对称呢?
  - 简化SVD: [U,S,V]= svd(A, 0)
  - 即去掉U的后m-n列

```
>> A = gallery(3)
>> [U,S,V]= svd(A)
```

$$A = \begin{pmatrix} -149 & -50 & -154 \\ 537 & 180 & 546 \\ -27 & -9 & -25 \end{pmatrix}$$

```
S =

  817.7597         0         0
         0    2.4750         0
         0         0    0.0030
```

特征值计算问题的敏感性

- A为非亏损阵
- A为亏损阵
- 特征值敏感性的例子

特征值计算问题的敏感性

- A为非亏损矩阵

不存在特征值分解的矩阵为亏损阵，否则为非亏损阵

  - 设 $X^{-1}AX = \Lambda$, 矩阵A变化δA, 则

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

  - 取范数得: $\|\delta\Lambda\| \le \|X^{-1}\|\|X\|\|\delta A\| = \kappa(X)\|\delta A\|$
  - 注意，这里的$\delta\Lambda$往往不是对角阵
  - 上述分析虽不严格, 但结论有普遍性: **特征值的敏感性可通过特征向量构成矩阵的条件数估计**
  - 若特征向量接近相关(对应矩阵很病态), 则特征值计算很敏感

```
>> A = gallery(3)
>> [X,lambda] = eig(A);
>> condest(X)
ans=
        1.2002e+003
```

此特征值计算问题比较病态

特征值计算问题的敏感性

- A为非亏损矩阵
  - 如何分析某个特征值λ的敏感程度？
  - 左特征向量y^H满足 $y^H A = \lambda y^H$

注意:λ的左特征向量是行向量，且一定存在!

(y是A^H的对应$\bar{\lambda}$的右特征向量)

  - 将矩阵A看成随某个扰动因素变化的函数，$\dot{A}$为对应的微分

$$Ax = \lambda x \xrightarrow{\text{两边微分}} \dot{A}x + A\dot{x} = \dot{\lambda}x + \lambda\dot{x}$$

  - 乘以左特征向量 $y^H\dot{A}x + \cancel{y^H A\dot{x}} = y^H\dot{\lambda}x + \cancel{y^H\lambda\dot{x}}$
  - 得：$\dot{\lambda} = \dfrac{y^H\dot{A}x}{y^H x}$ 即 $|\dot{\lambda}| \le \dfrac{\|y\|\|x\|}{|y^H x|}\|\dot{A}\|$

>> condeig(A)

  - 定义特征值条件数 $\kappa(\lambda, A) = \dfrac{\|y\|\|x\|}{|y^H x|}$ 则$|\dot{\lambda}| \le \kappa(\lambda, A)\|\dot{A}\|$
  - 因为

$\boldsymbol{X}^{-1}\boldsymbol{A}\boldsymbol{X} = \boldsymbol{\Lambda} \Longrightarrow \boldsymbol{X}^{-1}\boldsymbol{A} = \boldsymbol{\Lambda}\boldsymbol{X}^{-1} \Longrightarrow \boldsymbol{y}^H$为$\boldsymbol{X}^{-1}$的行向量，则 $\boldsymbol{y}^H\boldsymbol{x} = 1$

$$\kappa(\lambda, A) = \frac{\|y\|\|x\|}{|y^H x|} = \|y\|\|x\| \le \kappa(X)$$

特征值计算问题的敏感性

- A为非亏损矩阵
  - 若A为实对称矩阵，或复Hermite阵 $A = A^H$ (特征值全为实数)
  - 左特征向量y满足$y^H A = \lambda y^H$ ⟹ $Ay = \lambda y$ ⟹ x, y同方向

$|y^H x| = \|y\| \|x\|$ ⟹ $\kappa(\lambda, A) = \dfrac{\|y\|\|x\|}{|y^H x|} = 1$

实对称阵的特征值计算不敏感！

- A为亏损阵
  - 设λ_k为重特征值，$p(\lambda) = \det(A - \lambda I) = (\lambda - \lambda_k)^m q(\lambda)$
  - 矩阵某元素扰动δ后，特征多项式变为$p(\lambda) - O(\delta)$
  - 重特征值对扰动很敏感！ ⟹ $\lambda = \lambda_k + O(\delta^{1/m})$

例：

$$A_{16\times16} = \begin{pmatrix} 2 & 1 & & & \\ & 2 & 1 & & \\ & & \ddots & \ddots & \\ & & & 2 & 1 \\ \delta & & & & 2 \end{pmatrix}$$

$\delta = 10^{-16}$，则特征值的扰动为0.1

```
A=diag(ones(16,1)*2)+diag(ones(15,1), 1);
A(16,1)= 1e-16;
eig(A)
```

特征值计算问题的敏感性

- 另一个例子

```
A = gallery(5)
e = eig(A)
A^5
```

A =

-9	11	-21	63	-252
70	-69	141	-421	1684
-575	575	-1149	3451	-13801
3891	-3891	7782		
1024	-1024	2048		

e =

-0.0405
-0.0118 + 0.0383i
-0.0118 - 0.0383i
0.0320 + 0.0228i
0.0320 - 0.0228i

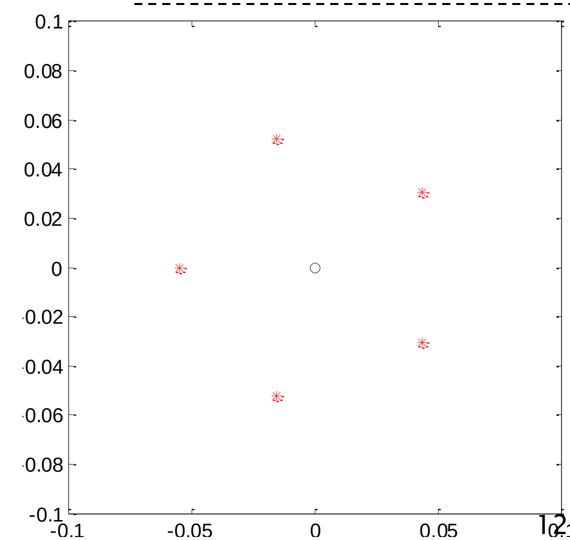
  - 此矩阵的特征值应该全为0！ (A^k特征值与A的特征值的关系)

```
plot(real(e),imag(e),'r*',0,0,'ko');
axis(.1*[-1 1 -1 1]); axis square;
```

  - 特征值计算的命令有错，还是误差？

```
e = eig(A + eps*randn(5,5).*A);
```

  - 重复执行，看复平面上五个特征值
  - 理论分析：0是特征多项式的五重根
  - 若特征多项式扰动δ，则$\lambda \approx O(\delta^{\frac{1}{5}})$



奇异值分解的存在性与性质

- 分解存在性的证明
- 奇异值分解的性质
- 计算奇异值的敏感性

奇异值分解(SVD)

- 奇异值分解的存在性
  - A为m×n矩阵，奇异值分解为 $A = U\Sigma V^H$
  - A^H的分解: $A^H = V\Sigma^H U^H$

    实对称阵或复Hermite阵
  - 不妨仅讨论m≥n的情况 $\underbrace{A^H A}_{n\times n} = V\Sigma^H\Sigma V^H = V\begin{bmatrix}\sigma_1^2 & & \\ & \ddots & \\ & & \sigma_n^2\end{bmatrix}V^H$
  - 奇异值是 $A^H A$特征值的算术平方根
  - 为了方便，仅证明A为实数矩阵的情况
- 存在性的证明
  - $A^T A$ 为对称半正定阵，设其非零特征值为$\sigma_1^2, \sigma_2^2, \cdots, \sigma_r^2$,形成$\Sigma_r^2$

特征值分解

$$A^T A = \underbrace{[V_1\ \ V_2]}_{\text{(正交阵)}}\begin{bmatrix}\Sigma_r^2 & O \\ O & O\end{bmatrix}\begin{bmatrix}V_1^T \\ V_2^T\end{bmatrix} \Rightarrow \begin{bmatrix}V_1^T \\ V_2^T\end{bmatrix}A^T A[V_1\ \ V_2] = \begin{bmatrix}\Sigma_r^2 & O \\ O & O\end{bmatrix}$$

$$\begin{bmatrix}V_1^T A^T A V_1 & \cdots \\ \cdots & V_2^T A^T A V_2\end{bmatrix}$$

$$U_1^T U_1 = I_r \Leftarrow \Sigma_r^{-1}V_1^T A^T \boxed{A V_1 \Sigma_r^{-1}} = I_r \Leftarrow$$

$$A V_2 = O \Leftarrow$$

奇异值分解(SVD)

- 奇异值分解的存在性
  - 存在性的证明 $A = U\Sigma V^T$

U₁为正交阵r列

$$\begin{bmatrix} V_1^T \\ V_2^T \end{bmatrix} A^T A [V_1 \ \ V_2] = \begin{bmatrix} \Sigma_r^2 & O \\ O & O \end{bmatrix} \Rightarrow \begin{array}{l} AV_2 = O \\ \text{设} U_1 = AV_1\Sigma_r^{-1}，\ U_1^T U_1 = I_r \end{array}$$

根据 U_1 补齐正交单位向量得到正交阵 $U = [U_1 \ \ U_2]$

$$U^T AV = \begin{bmatrix} U_1^T \\ U_2^T \end{bmatrix} A[V_1 \ \ V_2] = \begin{bmatrix} U_1^T AV_1 & U_1^T AV_2 \\ U_2^T AV_1 & U_2^T AV_2 \end{bmatrix} = \begin{bmatrix} U_1^T AV_1 & O \\ U_2^T AV_1 & O \end{bmatrix}$$

$$U_1 = AV_1\Sigma_r^{-1} \Rightarrow \begin{array}{l} U_1^T AV_1 = \Sigma_r^{-1} V_1^T A^T AV_1 = \Sigma_r \\ U_2^T AV_1 = U_2^T U_1 \Sigma_r = O \end{array}$$

所以，$U^T AV = \begin{bmatrix} \Sigma_r & O \\ O & O \end{bmatrix} = \Sigma$ **SVD的存在性得证!**

奇异值分解(SVD)

- 奇异向量的意义
  - $A = U\Sigma V^T \Rightarrow AV = U\Sigma$ 设有r个非零对角元
  - U的前r列, u_1, …, u_r是span(A)的单位正交基$\{Ax : \forall x \in \mathbb{R}^n\}$
  - 而$AV_2 = O$, 说明v_{r+1}, …, v_n是A的核空间$\{x \in \mathbb{R}^n : Ax = 0\}$的单位正交基
- 与特征值分解比较(m=n) $A = X\Lambda X^{-1} \Rightarrow AX = X\Lambda$
  - 对非亏损阵存在, 非零特征值对应的矩阵X的列向量为span(A)的基, 但一般不正交
  - 当A为实对称或复Hermite，可正交对角化 $A = X\Lambda X^T$ $A = X\Lambda X^H$
  - 非负特征值就是奇异值, 且$u=v=x$
  - 负特征值, 其相反数是奇异值, 且$u=-v=x$

实对称半正定阵呢?

奇异值分解(SVD)

- 奇异值与矩阵的2-范数、条件数
  - $A = U\Sigma V^T$，其中$\Sigma=\mathrm{diag}(\sigma_1,\sigma_2,\cdots,\sigma_n)$

最大奇异值

$$\|A\|_2 = \max_{x\neq O}\frac{\|Ax\|_2}{\|x\|_2} = \max_{x\neq O}\frac{\|U\Sigma V^Tx\|_2}{\|x\|_2} = \max_{x\neq O}\frac{\|\Sigma V^Tx\|_2}{\|V^Tx\|_2} = \max_{y\neq O}\frac{\|\Sigma y\|_2}{\|y\|_2} = \sigma_1$$

若 A 非奇异(m=n)，其奇异值均>0

$$\|A^{-1}\|_2 = \left(\min_{x\neq O}\frac{\|Ax\|_2}{\|x\|_2}\right)^{-1} = \left(\min_{y\neq O}\frac{\|\Sigma y\|_2}{\|y\|_2}\right)^{-1} = 1/\sigma_n$$

$$\Rightarrow \mathrm{cond}(A)_2 = \sigma_1/\sigma_n$$

若列满秩(奇异值均大于0)，

$$A^+ = (A^TA)^{-1}A^T = (V\Sigma^T\Sigma V^T)^{-1}V\Sigma^TU^T = V\underline{(\Sigma^T\Sigma)^{-1}\Sigma^T}U^T$$

$$\begin{bmatrix}\sigma_1^{-1} & & \\ & \ddots & \\ & & \sigma_n^{-1}\end{bmatrix}$$

$$\Rightarrow \|A^+\|_2 = 1/\sigma_n \Rightarrow \mathrm{cond}(A)_2 = \sigma_1/\sigma_n$$

SVD！

在其他情况下 $\mathrm{cond}(A)_2 = \infty$

奇异值计算问题的敏感性

- 敏感性分析
  - $\Sigma = U^H A V \Longrightarrow \Sigma + \delta\Sigma = U^H(A+\delta A)V$ (不严格分析)
  - U, V为正交阵或酉阵，则 $\|\delta\Sigma\|_2 = \|\delta A\|_2$
  - 矩阵2–范数是其最大奇异值, 这说明最大奇异值不敏感
  - 求小奇异值(可能是0)时, 仍可能很敏感
  - 例子 (奇异阵，特征值全0)

```
>> A = gallery(5)
>> format long e
>> svd(A)
```

$$\begin{bmatrix} -9 & 11 & -21 & 63 & -252 \\ 70 & -69 & 141 & -421 & 1684 \\ -575 & 575 & -1149 & 3451 & -13801 \\ 3891 & -3891 & 7782 & -23345 & 93365 \\ 1024 & -1024 & 2048 & -6144 & 24572 \end{bmatrix}$$

反复执行：

```
svd(A+eps*randn(5,5).*A)
```

结果：

```
1.010353607103610e+005
1.67945738406****e+000
1.46283872808****e+000
1.08016906998****e+000
*.****************-0**
```

计算特征值、奇异值的算法

- 计算矩阵特征值的算法
  QR算法及相关技术

特征值分解及其计算

- 特征值计算的方法
  - 求解特征方程？
    - 通过算矩阵行列式得到多项式系数，计算量大且很敏感
    - 高阶多项式方程求根：工作量大，且得到的解不准确
    - 例：$A = \begin{bmatrix} 1 & \epsilon \\ \epsilon & 1 \end{bmatrix}$, ϵ 比$\sqrt{\epsilon_{\text{mach}}}$ 小一点，准确特征值为$1 \pm \epsilon$，
      但 $\det(A-\lambda I) = \lambda^2 - 2\lambda + 1$，算出两个根均为1　(pp.232的例子)
    - 相反，求矩阵特征值的方法可用于解多项式方程　roots命令
      $$p(\lambda) = c_0 + c_1\lambda + \cdots + c_{n-1}\lambda^{n-1} + \lambda^n$$
      是伴随矩阵C_n的特征多项式。求出特征值即得多项式方程的所有根
      $$C_n = \begin{bmatrix} 0 & 0 & \cdots & 0 & -c_0 \\ 1 & 0 & \cdots & 0 & -c_1 \\ 0 & 1 & \cdots & 0 & -c_2 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & \cdots & 1 & -c_{n-1} \end{bmatrix}$$
  - 幂法 (power iteration) $x_k = Ax_{k-1}$
    - 求最大的特征值(主特征值)、及对应的特征向量
    - 条件：主特征值是唯一的，对实矩阵它一定是实的

特征值、特征向量的计算

- 幂法与反幂法
  - 幂法的应用：Google的PageRank算法
  - 加速幂法的收敛(瑞利商、位移技术、二次外推...)
  - 反幂法
    - 求最小的特征值，对A^{-1}应用幂法(解方程组)
    - 与位移技术结合(对$A-\lambda I$用), 很适合求某特征值的特征向量
- 计算非单个特征值
  - 若已知某个特征值及特征向量，可用收缩技术降维
  - QR迭代算法
    - 将矩阵正交相似变换为易求的形式 $\boldsymbol{Q}^T\boldsymbol{A}\boldsymbol{Q}$
  - Krylov子空间迭代法: Lanczos算法，Arnoldi算法
    - 适合于大规模稀疏阵，求最大的若干个特征值

特征值、特征向量的计算

- 收缩技术 （已知$\boldsymbol{A}$的一个特征值λ_1，特征向量$\boldsymbol{x}_1$）
  - 构造$\boldsymbol{H}$矩阵对$\boldsymbol{x}_1$实现消元：$\boldsymbol{H}\boldsymbol{x}_1=\sigma\boldsymbol{e}_1$，再对$\boldsymbol{A}$做正交相似变换

$$\boldsymbol{HAH}^T\boldsymbol{e}_1=\boldsymbol{HA}\left(\frac{1}{\sigma}\boldsymbol{x}_1\right)=\frac{1}{\sigma}\boldsymbol{HA}\boldsymbol{x}_1=\frac{1}{\sigma}\boldsymbol{H}\lambda_1\boldsymbol{x}_1=\frac{\lambda_1}{\sigma}(\sigma\boldsymbol{e}_1)=\lambda_1\boldsymbol{e}_1$$

$$\boldsymbol{HAH}^T=\begin{bmatrix}\lambda_1 & \boldsymbol{r}_1^T\\ \boldsymbol{0} & \boldsymbol{A}_1\end{bmatrix}$$

  - 例子：

$$\boldsymbol{A}=\begin{bmatrix}3 & -1 & 1\\ 2 & 0 & 1\\ 1 & -1 & 2\end{bmatrix}$$

$\lambda_1=2$，对应特征向量为$\boldsymbol{x}_1=[1,1,0]^T$，求其他特征值

Householder变换处理$\boldsymbol{x}_1$，$\boldsymbol{v}=\boldsymbol{x}_1+\sigma\boldsymbol{e}_1=\begin{bmatrix}2.4142\\ 1\\ 0\end{bmatrix}$

$$\Longrightarrow \boldsymbol{H}=\begin{bmatrix}-0.7072 & -0.7072 & 0\\ -0.7072 & 0.7072 & 0\\ 0 & 0 & 1\end{bmatrix},\quad \boldsymbol{HAH}^T=\begin{bmatrix}2 & 3 & -1.4142\\ 0 & 1 & 0\\ 0 & -1.4142 & 2\end{bmatrix}$$

$\boldsymbol{A}_1$

易知$\boldsymbol{A}$的其他两个特征值为1和2

计算所有特征值的方法

- QR算法

J. G.F. Francis和V. N. Kublanovskaya各自发明，见网络学堂"资源"文章

  - "二十世纪十大算法"之一，计算中小规模矩阵所有特征值的稳定、有效的方法
  - 实Schur分解： $A = QSQ^T \;\Rightarrow\; Q^T A Q = S$ (Q为正交阵)
  - S为1阶或2阶对角块形成的分块上三角阵(实Schur型)

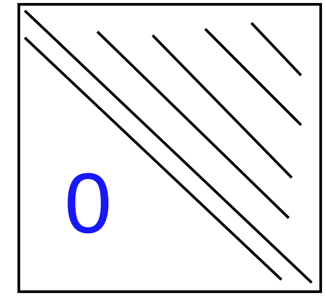
复Schur分解 $A = QUQ^H$

  - QR算法实现了Schur分解
    - $A_0 = A,\ Q_k R_k = A_k,\ A_{k+1} = R_k Q_k,\ k=0, 1, 2, \cdots$
    - $A_{k+1} = Q_k^T A_k Q_k$, 所以A_k的特征值总与A的相同

基本QR算法

    - 可证明：序列$\{A_k\}$"基本收敛"到S (实Schur型), 其对角块为A的特征值 (2阶对角块 ~ 一对共轭复特征值)
    - 条件：实矩阵A为非亏损阵，等模特征值只有实重特征值，或多重复的共轭特征值两种情况
  - 怎么减少每步迭代的计算量？怎么保证收敛？

计算所有特征值的方法



- 实用的QR算法
  - 把一般矩阵化为上Hessenberg阵再执行QR

(Householder变换)

$$A=\begin{bmatrix} a_{11} & \boldsymbol{r}_1^T \\ \boldsymbol{c}_1 & \boldsymbol{A}_{22}^{(1)} \end{bmatrix} \xrightarrow{\boldsymbol{H}_1=\begin{bmatrix} 1 & \boldsymbol{O}^T \\ \boldsymbol{O} & \boldsymbol{H}_1' \end{bmatrix}} \boldsymbol{H}_1\boldsymbol{A}=\begin{bmatrix} a_{11} & \boldsymbol{r}_1^T \\ \sigma_1\boldsymbol{e}_1 & \boldsymbol{H}_1'\boldsymbol{A}_{22}^{(1)} \end{bmatrix} \longrightarrow \boldsymbol{H}_1\boldsymbol{A}\boldsymbol{H}_1=\begin{bmatrix} a_{11} & \boldsymbol{r}_1^T\boldsymbol{H}_1' \\ \sigma_1\boldsymbol{e}_1 & \boldsymbol{H}_1'\boldsymbol{A}_{22}^{(1)}\boldsymbol{H}_1' \end{bmatrix}$$

$$\boldsymbol{A}^{(2)}=\boldsymbol{H}_1\boldsymbol{A}\boldsymbol{H}_1=\begin{bmatrix} \boldsymbol{A}_{11}^{(2)} & \boldsymbol{A}_{12}^{(2)} \\ \boldsymbol{A}_{21}^{(2)} & \boldsymbol{A}_{22}^{(2)} \end{bmatrix} \xrightarrow{\boldsymbol{H}_2=\begin{bmatrix} \boldsymbol{I}_2 & \boldsymbol{O}^T \\ \boldsymbol{O} & \boldsymbol{H}_2' \end{bmatrix}} \boldsymbol{H}_2\boldsymbol{A}^{(2)}\boldsymbol{H}_2=\left[\begin{array}{cc:cccc} a_{11} & a_{12}^{(2)} & a_{13}^{(3)} & \cdots & \cdots & a_{1n}^{(3)} \\ \sigma_1 & a_{22}^{(2)} & a_{23}^{(3)} & \cdots & \cdots & a_{2n}^{(3)} \\ \hdashline 0 & \sigma_2 & a_{33}^{(3)} & \cdots & \cdots & a_{3n}^{(3)} \\ 0 & 0 & \vdots & \ddots & & \vdots \\ \vdots & \vdots & \vdots & & \ddots & \vdots \\ 0 & 0 & a_{n3}^{(3)} & \cdots & \cdots & a_{nn}^{(3)} \end{array}\right]$$

…… ……

  - 最终 $\boldsymbol{H}_{n-1}\cdots\boldsymbol{H}_1\boldsymbol{A}\boldsymbol{H}_1\cdots\boldsymbol{H}_{n-1}$ 为上Hessenberg阵 (与A相似)
  - QR迭代第一步计算量变小，后续迭代步呢？

计算所有特征值的方法

- 实用的QR算法
  - 对上Hessenberg矩阵A_k执行QR迭代
    - 采用Givens旋转，以4阶矩阵为例 $Q_k^T A_k = R_k$

$$
\underset{A_k}{\begin{bmatrix} \times & \times & \times & \times \\ \times & \times & \times & \times \\ & \times & \times & \times \\ & & \times & \times \end{bmatrix}}
\longrightarrow
\underset{P_1A_k}{\begin{bmatrix} \times & \times & \times & \times \\ & \times & \times & \times \\ & \times & \times & \times \\ & & \times & \times \end{bmatrix}}
\longrightarrow
\underset{P_2P_1A_k}{\begin{bmatrix} \times & \times & \times & \times \\ & \times & \times & \times \\ & & \times & \times \\ & & \times & \times \end{bmatrix}}
\longrightarrow
\underset{P_3P_2P_1A_k}{\begin{bmatrix} \times & \times & \times & \times \\ & \times & \times & \times \\ & & \times & \times \\ & & & \times \end{bmatrix}}
\longrightarrow
$$

$A_{k+1} = R_kQ_k$

$$
\longrightarrow
\underset{P_3P_2P_1A_kP_1^T}{\begin{bmatrix} \times & \times & \times & \times \\ \times & \times & \times & \times \\ & & \times & \times \\ & & & \times \end{bmatrix}}
\longrightarrow
\underset{P_3P_2P_1A_kP_1^TP_2^T}{\begin{bmatrix} \times & \times & \times & \times \\ \times & \times & \times & \times \\ & \times & \times & \times \\ & & & \times \end{bmatrix}}
\longrightarrow
\underset{A_{k+1}}{\begin{bmatrix} \times & \times & \times & \times \\ \times & \times & \times & \times \\ & \times & \times & \times \\ & & \times & \times \end{bmatrix}}
$$

$$A_{k+1} = P_3P_2P_1A_kP_1^TP_2^TP_3^T$$

    - 计算效率很高; A_{k+1}仍是上Hessenberg阵！

计算所有特征值的方法

- 实用的QR算法
  - 单位移技术可加速收敛

$$\begin{cases} Q_k R_k = A_k - s_k I \ , & (\text{作 QR 分解}) \\ A_{k+1} = R_k Q_k + s_k I, & k = 1, 2, \cdots \end{cases} \Longrightarrow A_{k+1} = Q_k^T(A_k - s_k I)Q_k + s_k I = Q_k^T A_k Q_k$$

  - $\{A_k\}$序列仍正交相似，求出原问题特征值
  - 一种<u>单位移</u>策略：$s_k = A_k(n, n)$，使$A_k(n, n)$最先收敛到特征值. 然后，删除第n行，第n列，对低一阶矩阵做QR迭代
  - Wilkinson位移: 右下角2阶矩阵的特征值，其中接近$A_k(n,n)$的
  - 对实对称阵，上Hessenberg约化后为三对角阵

$$\begin{bmatrix} \times & \times & \times & \times & \times \\ \times & \times & \times & \times & \times \\ \times & \times & \times & \times & \times \\ \times & \times & \times & \times & \times \\ \times & \times & \times & \times & \times \end{bmatrix} \to \begin{bmatrix} \times & \times & & & \\ \times & \times & \times & & \\ & \times & \times & \times & \\ & & \times & \times & \times \\ & & & \times & \times \end{bmatrix} \to \begin{bmatrix} \times & & & & \\ & \times & & & \\ & & \times & & \\ & & & \times & \\ & & & & \times \end{bmatrix}$$

  - **结论:** 对实对称阵，使用Wilkinson位移的QR算法一定收敛

计算所有特征值的方法

- 针对实对称阵的单位移实用QR算法

约化为上Hessenberg阵

利用 Householder 变换将矩阵$\boldsymbol{A}$化简为三对角阵;

k: = n;　　　　{对$\boldsymbol{A}$的前 k 行、k 列执行 QR 算法}

While k>1 并且 $a_{k,k-1} \neq 0$ **do**

　$s := \boldsymbol{A}(k-1:k, k-1:k)$的特征值中，接近$a_{kk}$的那个;

Wilkinson位移

　For j=1, 2, ..., k　　　　{计算$\boldsymbol{A}(1:k,1:k) - s\boldsymbol{I}$ }

　　$a_{jj} := a_{jj} - s$;

　End

　用 Givens 旋转将$\boldsymbol{A}(1:k,1:k)$化为上三角阵，得旋转阵$\boldsymbol{G}_j$ ($j = 1, \cdots, k-1$)的参数c_j, s_j ;

　For j=1, 2, ..., k-1　　　　{计算$\boldsymbol{RQ} = \boldsymbol{R}\boldsymbol{G}_1^T \cdots \boldsymbol{G}_{n-1}^T$}

　　$\boldsymbol{A}(1:k,1:k) := \boldsymbol{A}(1:k,1:k)\boldsymbol{G}_j^T$; {执行列的 Givens 旋转}

上Hessenberg阵的QR迭代

　End

　For j=1, 2, ..., k　　　　{计算$\boldsymbol{A}(1:k,1:k) + s\boldsymbol{I}$ }

　　$a_{jj} := a_{jj} + s$;

　End

　If $a_{k,k-1} = 0$, **then**

矩阵第k行已收敛!

　　k: = k-1;

　End

End　　　　{$\boldsymbol{A}$的对角元就是待求的特征值$\lambda_1, \cdots, \lambda_n$}

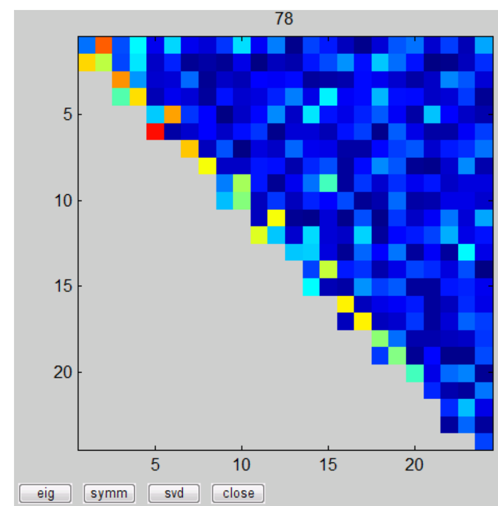
计算所有特征值的方法

- 实用的QR算法
  - 对非对称实矩阵, Wilkinson位移不适用(虚数特征值), 可采用一种双位移策略
  - 计算过程:

$$\begin{bmatrix} \times & \times & \times & \times & \times \\ \times & \times & \times & \times & \times \\ \times & \times & \times & \times & \times \\ \times & \times & \times & \times & \times \\ \times & \times & \times & \times & \times \end{bmatrix} \to \begin{bmatrix} \times & \times & \times & \times & \times \\ \times & \times & \times & \times & \times \\ & \times & \times & \times & \times \\ & & \times & \times & \times \\ & & & \times & \times \end{bmatrix} \to \begin{bmatrix} \times & \times & \times & \times & \times \\ & \times & \times & \times & \times \\ & & \times & \times & \times \\ & & & \times & \times \\ & & & & \times \end{bmatrix}$$

  - 正交约化为上Hessenberg阵, 以及位移技术, 显著扩大了QR算法收敛的适用范围
  - 对实非对称阵,可证明: 利用双位移技术的QR算法一定收敛*

    (课本的说法值得商榷)
  - 演示程序eigsvdgui



*L Elden, Matrix Methods in Data Mining and Pattern Recognition, SIAM, 2007, pp.198

Matlab中有关命令及其他

- eig命令　– 主要对稠密矩阵　LAPACK程序包
  - d=eig(A); 返回所有特征值，使用QR算法
  - [V, D]=eig(A); 返回特征值和特征向量矩阵, QR算法
  - 可能还会对矩阵元素进行比例调整，提高数值稳定性
  - 中小规模稠密阵, 较小规模的对称稀疏阵(只特征值)
- eigs命令
  - 针对一般稀疏矩阵, Lanczos算法，Arnoldi算法
  - d = eigs(A, k); 返回最大的k个特征值(k缺省为6)
  - [V, D]=eigs(A, k); 返回最大的k个特征值与相应特征向量
- hess命令: 对称正交变换生成上Hessenberg阵
- schur命令: 生成矩阵的Schur分解　T = schur(A);

注意与eig的区别　[U, T] = schur(A);

如何计算矩阵奇异值?

- 利用求特征值的QR算法
  - 等效于求A^TA矩阵的特征值　　　这样计算误差大!
  - 注意：对A施加左、右正交变换不改变它的奇异值
  - 先将A化简(Householder变换)为双对角阵B, 再对B^TB执行(隐式)QR迭代算法

$$\boldsymbol{A} = \begin{bmatrix} a_1 & \cdots\cdots & a_n \end{bmatrix} \xrightarrow[\boldsymbol{H}_1\boldsymbol{a}_1 = \alpha_1\boldsymbol{e}_1^{(m)}]{\boldsymbol{H}_1\boldsymbol{A}} \begin{bmatrix} \alpha_1 & \boldsymbol{b}_1^T \\ 0 & \end{bmatrix} \xrightarrow[\boldsymbol{H}_2\boldsymbol{b}_1 = \beta_1\boldsymbol{e}_1^{(n-1)}]{\boldsymbol{H}_1\boldsymbol{A}\begin{bmatrix}1 & 0\\ 0 & \boldsymbol{H}_2\end{bmatrix} = \boldsymbol{H}_1\boldsymbol{A}\boldsymbol{T}_1} \begin{bmatrix} \alpha_1 & \beta_1 \quad 0 \\ 0 & \end{bmatrix}$$

$$\cdots \longrightarrow \begin{bmatrix} \alpha_1 & \beta_1 & & \\ & \ddots & \ddots & \\ & & \ddots & \beta_{n-1} \\ & & & \alpha_n \end{bmatrix} = \boldsymbol{H}_n \cdots \boldsymbol{H}_1\boldsymbol{A}\boldsymbol{T}_1 \cdots \boldsymbol{T}_{n-1} = \boldsymbol{B}$$

QR迭代过程中：

$\boldsymbol{B}_k^T\boldsymbol{B}_k$为三对角矩阵，迭代中保持其形状

演示程序: eigsvdgui

低秩近似与主分量分析

- 矩阵的低秩近似与PCA
一个图像处理问题

低秩近似与主分量分析 (PCA)

- 利用SVD做矩阵的低秩近似

$$A = [u_1 \cdots u_m] \begin{bmatrix} \sigma_1 & & \\ & \ddots & \\ & & \sigma_n \end{bmatrix} \begin{bmatrix} v_1^T \\ \vdots \\ v_n^T \end{bmatrix} = [\sigma_1 u_1 \cdots \sigma_n u_n] \begin{bmatrix} v_1^T \\ \vdots \\ v_n^T \end{bmatrix} = \sum_{i=1}^{n} \sigma_i E_i$$

  - $E_i = u_i v_i^T$, $i = 1, \cdots, n$, 为秩1矩阵, 它有个奇异值为1, 其余0
  - 存储 E_i 只要m+n个单元, 计算 $E_i x$ 只需m+n次乘法 $\|E_i\|_2 = 1$
  - 忽略小奇异值有关项, 得到A的低秩近似 $A_r = \sum_{i=1}^{r} \sigma_i E_i$
  - 可证明, 在所有秩为r的矩阵中A_r最接近A

(无论2范数还是F范数) $\|A - A_r\|_2 = \sigma_{r+1}$

- 主分量分析
  - 统计分析、数据处理中的一个方法
  - 用较少的独立数据(主分量)表示存在相关性的所有数据

$$A \approx [\sigma_1 u_1 \cdots \sigma_r u_r] \begin{bmatrix} v_1^T \\ \vdots \\ v_r^T \end{bmatrix}$$

主分量分析 (PCA)

- 一组高度与重量的数据
  - 两列数据是高度相关联的

```
>> [U,S,V] = svd(A,0);
>> sigma = diag(S)
sigma =
        156.4358
        8.7658
>> E1=sigma(1)*U(:,1)*V(:,1)'
```

  - 近似表示高度与重量

有一个潜在的主要分量

单一主分量是:

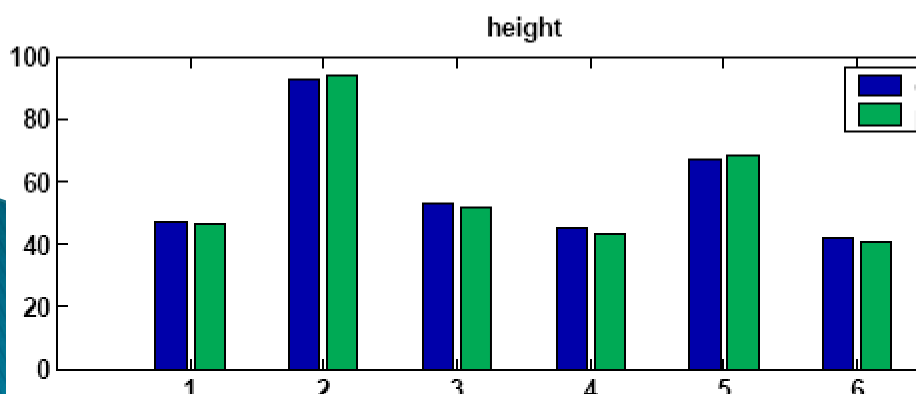
sigma(1)*U(:,1)

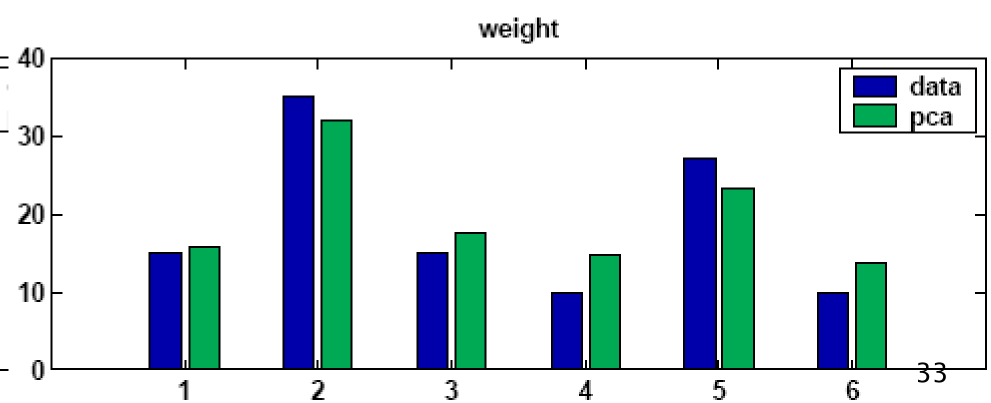
49.3269
99.3163
55.0076
45.8240
72.1250
42.9837

A =

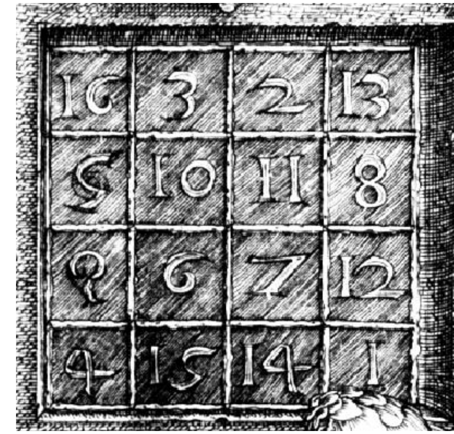
47	15
93	35
53	15
45	10
67	27
42	10

V(:,1)’=[0.9468, 0.3219]





主分量分析 (PCA)

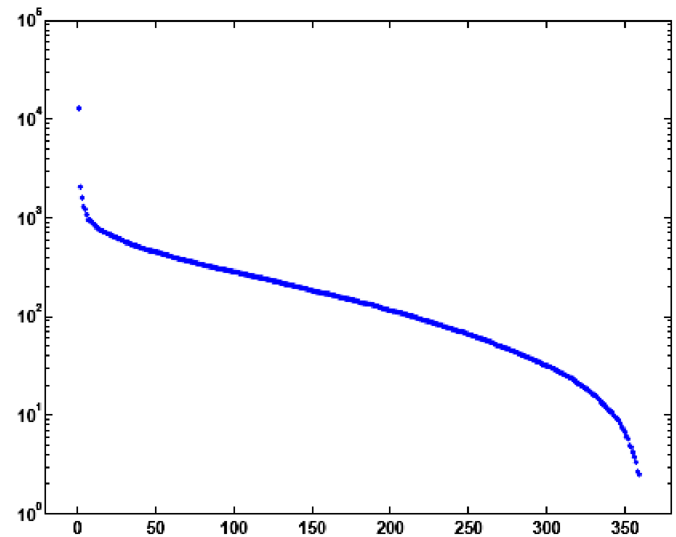


- 数字图像处理

演示imagesvd.m

  - Albrecht Durer(丢勒)的版画Melancolia II
  - 图像数据为359x371的矩阵, 进行低秩近似

```
>> load detail;
>> image(X); colormap(gray(64))
>> [U, S, V]= svd(X, 0);
>> sigma= diag(S);
>> semilogy(sigma, '.')
```



  - 取若干主要奇异值重建矩阵

```
>> Xr= U(:,1:r)*S(1:r,1:r)*V(:,1:r)'
```

  - 节省存储量; 目前主要用于特征识别

程序svd_appl.m

- 若A实对称,$A = Q\Lambda Q^T$左右奇异向量很接近, 低秩近似更简单

圆生成器

- 用整数+,-,*就能画圆?
  其内在原因与特征值的关系

圆生成器

(在只有整数的计算机上)

```
    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
```

- 一个简单的画圆的程序
  - 整数计算；没有乘除法/平方根/三角函数
  - 为什么能画圆？ cirgen.m

```
h = 1/32;
x = 1;  y = 0;
h1=plot(x, y, '.', 'erasemode','none');
axis([-1.1, 1.1, -1.1, 1.1]); axis square;
while 1
   x = x + h*y;
   y = y - h*x;
   set(h1, 'xdata', x, 'ydata', y); drawnow
end
```

$$x_{n+1} = \begin{pmatrix} 1 & h \\ -h & 1-h^2 \end{pmatrix} x_n$$

  - 设第n个点为$\boldsymbol{x}_n$，则$\boldsymbol{x}_0=[1,0]^{\mathrm{T}}$，$\boldsymbol{x}_{n+1}(1)=\boldsymbol{x}_n(1)+h\boldsymbol{x}_n(2)$

$$\boldsymbol{x}_{n+1}(2)=\boldsymbol{x}_n(2)-h\boldsymbol{x}_{n+1}(1)=-h\boldsymbol{x}_n(1)+(1-h^2)\boldsymbol{x}_n(2)$$

圆生成器

- 一个简单的画圆的程序
  - 生成一系列二维点$\{\boldsymbol{x}_n\}$，满足 $x_{n+1} = \begin{pmatrix} 1 & h \\ -h & 1-h^2 \end{pmatrix} x_n$ A
  - $\boldsymbol{x}_n = \boldsymbol{A}^n \boldsymbol{x}_0$，要求通式需求$\boldsymbol{A}$特征值

$$\lambda = \frac{2-h^2 \pm \sqrt{h^2(h^2-4)}}{2} \xrightarrow{\text{h很小}} \lambda = \frac{2-h^2}{2} \pm i\frac{\sqrt{h^2(4-h^2)}}{2}$$

  - 可证明：$|\lambda| = 1.$ $\quad \boldsymbol{AX} = \boldsymbol{X\Lambda} \Longrightarrow A^n = X\Lambda^n X^{-1}$ 假设0<h<2
  - 设$\lambda_1 = e^{i\theta}, \lambda_2 = e^{-i\theta}$，$\boldsymbol{\Lambda}^n = \begin{bmatrix} e^{in\theta} & 0 \\ 0 & e^{-in\theta} \end{bmatrix}$
  - 求解特征向量，得 再求$\boldsymbol{X}^{-1}$，整理得

$$\boldsymbol{X} = \begin{bmatrix} \frac{h}{2} + i\sqrt{1-\frac{h^2}{4}} & 1 \\ -1 & -\frac{h}{2} - i\sqrt{1-\frac{h^2}{4}} \end{bmatrix}$$

是椭圆，但h→0,θ→0

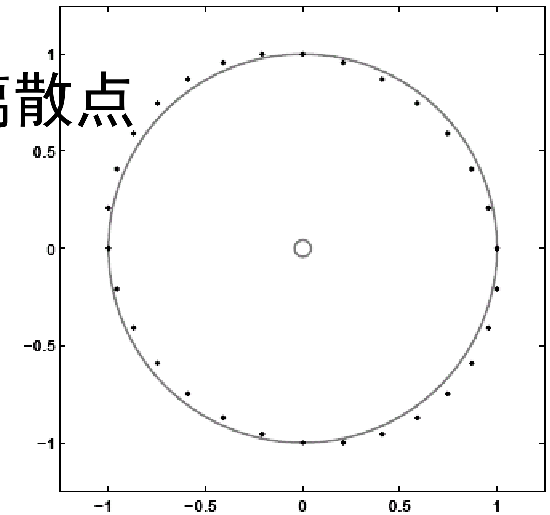
$$\boldsymbol{x}_n = \frac{1}{\cos(\frac{\theta}{2})} \begin{bmatrix} \cos(n\theta - \frac{\theta}{2}) \\ \sin(n\theta) \end{bmatrix}$$

满足$[x - y\sin\left(\frac{\theta}{2}\right)]^2 + [y\cos\left(\frac{\theta}{2}\right)]^2 = 1$

圆生成器

$$\boldsymbol{x}_n = \frac{1}{\cos(\frac{\theta}{2})}\begin{bmatrix}\cos(n\theta - \frac{\theta}{2}) \\ \sin(n\theta)\end{bmatrix}$$

- 画圆的程序
  - h=1/32, 足够小, 画圆效果很好
  - h=0.20906, 对应的$\theta=\pi/15$, 生成30个离散点



- 更多讨论
  - 平面旋转矩阵 $\begin{pmatrix}\cos h & \sin h \\ -\sin h & \cos h\end{pmatrix}$
  - 若它为A, 则$\boldsymbol{x}_{n+1} = \boldsymbol{A}\boldsymbol{x}_n$, 一定生成圆
  - 事实上 $\begin{pmatrix}\cos h & \sin h \\ -\sin h & \cos h\end{pmatrix} \approx \begin{pmatrix}1 & h \\ -h & 1-h^2\end{pmatrix}$
  - 常微分方程 $\dot{x} = Qx$, $Q = \begin{pmatrix}0 & 1 \\ -1 & 0\end{pmatrix}$ 的准确解也得到圆

称为"圆"常微分方程 (与第7章和习题10.16有关)