矩阵代数与应用

上海大学计算机工程与科学学院

2023年3月

每种特殊矩阵都与<u>某一类特殊问题相关</u>.

- 置换矩阵(permutation matrix)
- 互换矩阵(exchange matrix)
- 位移矩阵(shift matrix)
- 选择矩阵(selective matrix)
- 正交矩阵与酉矩阵(orthogonal/unitary matrix)
- 三角阵(upper/lower triangular matrix)
- 范德蒙矩阵（Vandermonde matrix)
- 傅里叶矩阵(Fourier matrix)
- 哈达玛矩阵(Hadamard matrix)
- 拓普利兹矩阵(Toeplitz matrix)
- 汉克矩阵(Hankel matrix)

几个特殊运算:

(1)直和　　(2)Hadamard积　　(3)Kronecker积

第二章 特殊矩阵

2.1 特殊矩阵

2.2 特殊矩阵运算

2.1 特殊矩阵

◊对角矩阵与反(斜) 对角矩阵

$$D = \begin{bmatrix} d_1 & & & \\ & d_2 & & \\ & & \ddots & \\ & & & d_n \end{bmatrix} = diag(d_1, d_2, \ldots, d_n) \quad \overline{D} = \begin{bmatrix} & & & d_1 \\ & & d_2 & \\ & \iddots & & \\ d_n & & & \end{bmatrix}$$

◊上三角矩阵与下三角矩阵

$$R = \begin{bmatrix} * & * & \cdots & * \\ & * & \cdots & * \\ & & \ddots & \vdots \\ & & & * \end{bmatrix} \qquad L = \begin{bmatrix} * & & & \\ * & * & \ddots & \\ \vdots & \vdots & \ddots & \\ * & * & \cdots & * \end{bmatrix}$$

*当对角元素为1时，成为**单位上(下)三角矩阵**.

上(下)三角矩阵的性质

运算的封闭性!

(1) 上(下)三角矩阵的和、差、乘积仍为上(下)三角矩阵的.

(2) 上(下)三角矩阵的k次幂仍为上(下)三角矩阵的，且其第i个对角线元素为$r_{ii}^k\,(l_{ii}^k)$.

(3) 上(下)三角矩阵的转置为下(上)三角矩阵.

(4) 上(下)三角矩阵的的逆仍为上(下)三角矩阵的.

(5) 上(下)三角矩阵的行列式等于其对角线元素之积，即

$$\det(R) = r_{11} r_{22} \ldots r_{nn} = \prod_{i=1}^{n} r_{ii}$$

(6) 上(下)三角矩阵的特征值等于其各对角线元素.

(7) **若矩阵** $A_{n\times n} > 0$ **(正定)，则A可以分解为一个下三角矩阵与其复共轭转置之积** $A = LL^H$.

(Cholesky分解(乔里斯基分解)，暂略，下一章讲解)

三角阵的生成

```
>> help chol
chol - Cholesky 分解
   此 MATLAB 函数 基于矩阵 A 的对角线和上三角形生成上三角矩阵 R，满足方程 R'*R=A.chol 函数假定 A 是复数 Hermitian对称矩阵.如果不是对称的，则 chol 使用上三角的（复共轭）转置作为下三角.矩阵 A 必须是正定矩阵.
   R = chol(A)
   L = chol(A,'lower')
   R = chol(A,'upper')
```

```
>> help diag
diag - 创建对角矩阵或获取矩阵的对角元素
   此 MATLAB 函数 返回包含主对角线上向量 v 的元素的对角矩阵.
   D = diag(v)
   D = diag(v,k)
   x = diag(A)
   x = diag(A,k)
```

```
>> help triu
triu - 矩阵的上三角形部分
   此 MATLAB 函数 返回 X 的上三角形部分.
   U = triu(X)
   U = triu(X,k)
```

其它特殊矩阵的生成(1)

```
>> help hadamard
hadamard - Hadamard 矩阵
   此 MATLAB 函数 返回阶次为 n 的Hadamard矩阵.
   H = hadamard(n)
   H = hadamard(n,classname)
```

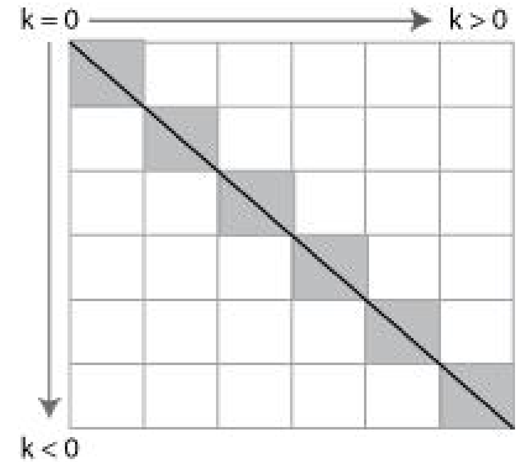
```
>> help hankel
hankel - Hankel 矩阵
   此 MATLAB 函数 返回其第一列是 c 并且其第一个反对角线下方的元素为零的 Hankel 方阵.
   H = hankel(c)
   H = hankel(c,r)
>> B=hankel([1,2 3,4])
B =1     2     3     4
     2     3     4     0
     3     4     0     0
     4     0     0     0
```

```
>> help toeplitz
toeplitz - 托普利茨矩阵
   此 MATLAB 函数 返回非对称托普利茨矩阵，其中 c 作为第一列，r 作为第一行.如果 c 和 r 的首个元素不同，toeplitz 将发出警告并使用列元素作为对角线.
   T = toeplitz(c,r)
   T = toeplitz(r)
```

```
>> diag([1,2,3,5])
ans =
     1     0     0     0
     0     2     0     0
     0     0     3     0
     0     0     0     5
```

U = triu(A) 返回矩阵 A 的上三角部分.

U = triu(A,k) 返回位于 A 的第 k 条对角线上以及该对角线上方的元素.




```
>> A=rand(6,6)
A =
    0.8147    0.2785    0.9572    0.7922    0.6787    0.7060
    0.9058    0.5469    0.4854    0.9595    0.7577    0.0318
    0.1270    0.9575    0.8003    0.6557    0.7431    0.2769
    0.9134    0.9649    0.1419    0.0357    0.3922    0.0462
    0.6324    0.1576    0.4218    0.8491    0.6555    0.0971
    0.0975    0.9706    0.9157    0.9340    0.1712    0.8235
>> U = triu(A)
U =
    0.8147    0.2785    0.9572    0.7922    0.6787    0.7060
         0    0.5469    0.4854    0.9595    0.7577    0.0318
         0         0    0.8003    0.6557    0.7431    0.2769
         0         0         0    0.0357    0.3922    0.0462
         0         0         0         0    0.6555    0.0971
         0         0         0         0         0    0.8235
```

```
>> B=rand(3,3)
B =
    0.6948    0.0344    0.7655
    0.3171    0.4387    0.7952
    0.9502    0.3816    0.1869
>> C=triu(B,1)
C =

         0    0.0344    0.7655
         0         0    0.7952
         0         0         0
>> D=triu(B,-1)
D =
    0.6948    0.0344    0.7655
    0.3171    0.4387    0.7952
         0    0.3816    0.1869
```

```
>> D=rand(3,3)
D =
    0.4898    0.7094    0.6797
    0.4456    0.7547    0.6551
    0.6463    0.2760    0.1626

>> E=D*D'
E =
    1.2051    1.1989    0.6229
    1.1989    1.1973    0.6028
    0.6229    0.6028    0.5204

>> R = chol(E)
R =
    1.0978    1.0921    0.5674
         0    0.0677   -0.2488
         0         0    0.3695

>> R'*R
ans =

    1.2051    1.1989    0.6229
    1.1989    1.1973    0.6028
    0.6229    0.6028    0.5204
```

方阵各元素的代数余子式构成的矩阵称为该矩阵的**伴随矩阵.**

$$A=(a_{ij})_{n\times n},\quad A^*=\begin{pmatrix} A_{11} & A_{21} & \cdots & A_{n1} \\ A_{12} & A_{22} & \cdots & A_{n2} \\ \cdots & \cdots & \cdots & \cdots \\ A_{1n} & A_{2n} & \cdots & A_{nn} \end{pmatrix}$$

```
>> Q=rand(5,5)
Q =
    0.1190    0.2238    0.8909    0.2575    0.9293
    0.4984    0.7513    0.9593    0.8407    0.3500
    0.9597    0.2551    0.5472    0.2543    0.1966
    0.3404    0.5060    0.1386    0.8143    0.2511
    0.5853    0.6991    0.1493    0.2435    0.6160

>> P=triu(Q)
P =
    0.1190    0.2238    0.8909    0.2575    0.9293
         0    0.7513    0.9593    0.8407    0.3500
         0         0    0.5472    0.2543    0.1966
         0         0         0    0.8143    0.2511
         0         0         0         0    0.6160
```

◊基本矩阵

$$E_{ij}^{(m\times n)} = e_i^{(m)}(e_j^{(n)})^T$$

性质:

(1) $E_{ij}^{(m\times n)}E_{kl}^{(n\times r)} = \delta_{jk}E_{il}^{(m\times r)}$

(2) $\left(E_{ij}^{(m\times n)}\right)^T = E_{ji}^{(n\times m)}$

(3) $A = \sum_{i=1}^{m}\sum_{j=1}^{n} a_{ij}E_{ij}^{(m\times n)}$

(4) $E_{ij}^{(s\times m)}AE_{kl}^{(n\times r)} = a_{jk}E_{il}^{(s\times r)}$

(5) $\det(E_{ij}^{(m\times n)}) = 0, (m = n > 1)$

两个向量(列向量)的外积:

$$a \wedge b = ab^T$$

```
>> a=[0 1 0 0 0]
a =
     0     1     0     0     0

>> b=[0 0 0 0 1 0 0]'
b =
     0
     0
     0
     0
     1
     0
     0

>> b*a
ans =
     0     0     0     0     0
     0     0     0     0     0
     0     0     0     0     0
     0     0     0     0     0
     0     1     0     0     0
     0     0     0     0     0
     0     0     0     0     0
```

◊正交矩阵

$$QQ^T = Q^T Q = I$$

半正交(semi-orthogonal)矩阵

$$QQ^T = I_m \text{ 或 } Q^T Q = I_n$$

◊酉矩阵

$$UU^H = U^H U = I$$

仿酉(para-unitary)矩阵

$$UU^H = I_m \quad \text{或} \quad U^H U = I_n$$

```
>> hadamard(4)
ans =
     1     1     1     1
     1    -1     1    -1
     1     1    -1    -1
     1    -1    -1     1
```

性质：

（1） U 为酉矩阵 $\Leftrightarrow U^{-1} = U^H$

（2） $U \in R^{m\times m}$ 为酉矩阵 $\Leftrightarrow$ U为正交矩阵.

（3） U为酉矩阵 $\Leftrightarrow$ U的列(行)是标准正交的向量.

```
>> H=hadamard(4)*1i/2
H =
   0.0000 + 0.5000i   0.0000 + 0.5000i   0.0000 + 0.5000i   0.0000 + 0.5000i
   0.0000 + 0.5000i   0.0000 - 0.5000i   0.0000 + 0.5000i   0.0000 - 0.5000i
   0.0000 + 0.5000i   0.0000 + 0.5000i   0.0000 - 0.5000i   0.0000 - 0.5000i
   0.0000 + 0.5000i   0.0000 - 0.5000i   0.0000 - 0.5000i   0.0000 + 0.5000i
```

```
>> H*H.'
ans =
    -1     0     0     0
     0    -1     0     0
     0     0    -1     0
     0     0     0    -1
```

```
>> H*H'
ans =
     1     0     0     0
     0     1     0     0
     0     0     1     0
     0     0     0     1
```

```
>> H=hadamard(8)
H =
     1     1     1     1     1     1     1     1
     1    -1     1    -1     1    -1     1    -1
     1     1    -1    -1     1     1    -1    -1
     1    -1    -1     1     1    -1    -1     1
     1     1     1     1    -1    -1    -1    -1
     1    -1     1    -1    -1     1    -1     1
     1     1    -1    -1    -1    -1     1     1
     1    -1    -1     1    -1     1     1    -1
```

(4) $U_{m\times m}$ 为酉矩阵，则

$U^T, U^H, U^*, U^{-1}, U^i$ 均为酉矩阵.

(5) U和V均为酉矩阵 $\Rightarrow UV$ 为酉矩阵

(6) 若 $U_{m\times m}$ $V_{n\times n}$ 为酉矩阵，则

① $U\oplus V$ 为酉矩阵

② $U\otimes V$ 为酉矩阵

(7) 若 $U_{m\times m}$ 为酉矩阵，则

① $\det(U)=\pm 1$

② $rank(U)=m$

③ $UU^H=U^HU$

④ λ 为U的特征值 $\Rightarrow |\lambda|=1$

⑤ $x_{m\times 1}\Rightarrow \|Ux\|_2=\|x\|_2$

⑥ $A_{m\times n}\Rightarrow \|UA\|_F=\|A\|_F$

⑦ $A_{n\times m}=\|AU\|_F=\|A\|_F$

(等距变换)

Block diagonal concatenation of matrix input arguments.

矩阵的直和: blkdiag(A,B)

$$A\oplus B=\begin{bmatrix} A & O_{m\times n} \\ O_{n\times m} & B \end{bmatrix}$$

矩阵的Kronecker积:kron(A,B)

the Kronecker tensor product of X and Y.

$$A\otimes B=[a_{ij}B]=\begin{bmatrix} a_{11}B & a_{12}B & \cdots & a_{1n}B \\ a_{21}B & a_{22}B & \cdots & a_{2n}B \\ \vdots & \vdots & & \vdots \\ a_{m1}B & a_{m2}B & \cdots & a_{mn}B \end{bmatrix}_{mp\times nq}$$

表1. 实向量、实矩阵与复向量、复矩阵的性质比较

实向量、实矩阵	复向量、复矩阵
范数: $\|x\| = \sqrt{x_1^2 + \cdots + x_n^2}$	范数: $\|x\| = \sqrt{\|x_1\|^2 + \cdots + \|x_n\|^2}$
转置: $A^T = [a_{ij}], (AB)^T = B^T A^T$	转置: $A^T = [a_{ji}^*], (AB)^H = B^H A^H$
内积: $\langle x, y \rangle = x^T y$	内积: $\langle x, y \rangle = x^H y$
正交性: $x^T y = 0$	正交性: $x^H y = 0$
对称矩阵: $A^T = A$	对称矩阵: $A^H = A$
正交矩阵: $Q^T = Q^{-1}$	正交矩阵: $U^H = U^{-1}$
特征值分解: $A = Q\Sigma Q^T = Q\Sigma Q^{-1}$	特征值分解: $A = U\Sigma U^H = U\Sigma U^{-1}$
范数的正交不变性: $\|Qx\| = \|x\|$	范数的正交不变性: $\|Ux\| = \|x\|$
内积的正交不变性: $\langle Qx, Qy \rangle = \langle x, y \rangle$	内积的正交不变性: $\langle Ux, Uy \rangle = \langle x, y \rangle$

酉变换：若U为酉矩阵，则称线性变换 Ux 称为x的酉变换.

酉等价：若U为酉矩阵，则称矩阵 $B=U^H AU$ 与A酉等价.

特别地，如果U取实数(因而是实正交的)，则称B与A正交等价.

◊正规矩阵（normal matrix）：满足 $A^H A = AA^H$ 的 $A \in C^{n\times n}$ 矩阵.

◊ **Hadamard矩阵**：所有元素取+1或-1，并且满足

$$H_n H_n^T = H_n^T H_n = nI_n$$

的 $n\times n$ 正方矩阵 H_n 称为 n 阶**Hadamard矩阵**.

性质：

(1)Hadamard矩阵的每一行或列均由+1或-1构成，且两两正交.特别地，$\frac{1}{\sqrt{n}}H_n$ 为标准正交矩阵.

（2）用-1乘Hadamard矩阵的任意一行(列),所得结果仍为一 Hadamard矩阵. 于是，可以得到第一列和第一行的所有元素为+1的Hadamard矩阵，并称之为**规范化Hadamard矩阵**.

（3）当Hadamard矩阵的阶数 $n>2$ 时，$n=4k$，$k\in Z$

（4）$\det|H_n| = \pm n^{n/2}$

定理：令 $n=2^k, k=1,2,\ldots$，则

$$H_{2n} = \frac{1}{\sqrt{2}}\begin{bmatrix} H_n & H_n \\ H_n & -H_n \end{bmatrix}，\quad 其中\ H_2 = \frac{1}{\sqrt{2}}\begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}$$

◊**相似矩阵**：若存在非奇异矩阵 $S\in C^{n\times n}$，使得 $B=S^{-1}AS$，则称矩阵 $B\in C^{n\times n}$ 与 $A\in C^{n\times n}$ **相似**，并简记为B∽A.

★**相似变换：** $A\mapsto S^{-1}AS$ **（代表相同的线性变换）**

相似矩阵的性质：

（1）自反性：A∽A，即任意矩阵与他自己相似.

（2）对称性：若A相似于B，则B相似于A.

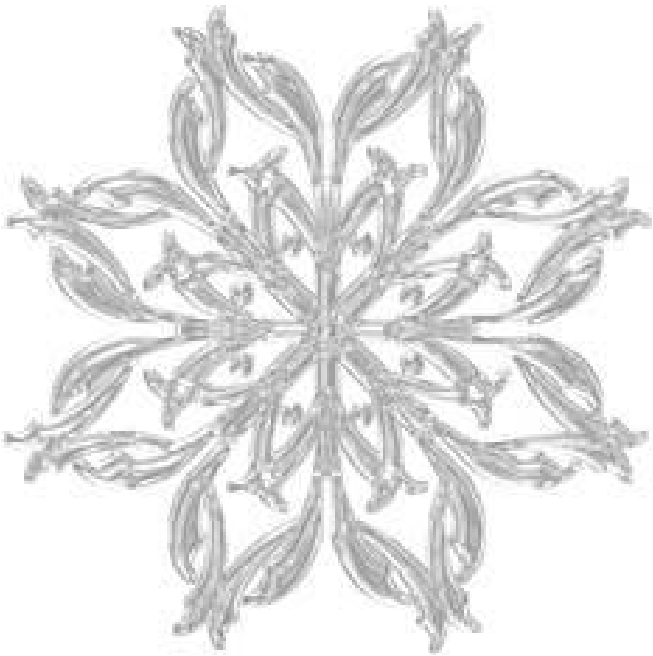
（3）传递性：若A∽B和B∽C，则A∽C.

（4）若B∽A，则B^k∽A^k.

（5）若B∽A，且均可逆，则B^{-1}∽A^{-1}.

（6）若B∽A，则 $\det(B)=\det(A)$.

（7）若B∽A，则 $tr(B)=tr(A)$.



◊**相合矩阵：** 若 $A,B,C\in C^{n\times n}$，且C非奇异，则称矩阵 $B=C^H AC$ 与A **相合(congruent)**.

双线性映射：Z=XAY
X:n行m列矩阵，
A:m行p列矩阵，
Y: p行q列矩阵
Z:n行q列矩阵.
当n=q=1时，也称为双线性函数.
二次型可以看成一类特殊的双线性映射.

★**相合变换：** $A\mapsto C^H AC$

性质：

（1）自反性：A相合与A，即任一矩阵与自身相合.

（2）对称性：若A相合于B，则B相合于A.

（3）传递性：若A相合于B，而B又相合于D，则A相合于D.

思考： 相似变换，相合变换与酉变换之间的关系？

合同：<u>一个二次型在不同坐标变换下的表示</u>.

◊**初等矩阵：**对 $n \times n$ 单位矩阵 I_n 进行初等行(或列)变换得到的矩阵.

Ⅰ 型初等矩阵 $E_{(p,q)}$：互换单位矩阵 I_n 的第p行和第q行得到的矩阵,或互换单位矩阵的第p列和第q列得到的矩阵.

$$r_p \leftrightarrow r_q \text{ 或 } c_p \leftrightarrow c_q$$

Ⅱ型初等矩阵 $E_{\alpha(p)}$：用一个非零常数α乘以单位矩阵 I_n 的第p行（或列）得到的矩阵.

$$r_p \leftarrow \alpha r_p \text{ 或 } c_p \leftarrow \alpha c_p$$

Ⅲ型初等矩阵 $E_{(p)+\alpha(q)}$：用一个非零常数α乘以单位矩阵 I_n 的第q行（或列），再加上 I_n 的第p行（或列）得到的矩阵.

$$r_p \leftarrow r_p + \alpha r_p \quad \text{或} \quad c_p \leftarrow c_p + \alpha c_p$$

初等矩阵左乘：

（1） Ⅰ型初等矩阵左乘矩阵A，即 $E_{(p,q)}A$ ，表示互换矩阵A的第p行和第q行.

（2） Ⅱ型初等矩阵左乘矩阵A，即 $E_{\alpha(p)}A$ ，表示矩阵A的第p行元素乘一个非零常数α.

（3） Ⅲ型初等矩阵左乘矩阵A，即 $E_{(p)+\alpha(q)}A$，表示矩阵A的第q行乘以非零常数α后，再与第p行相加.

初等矩阵右乘：

（1） Ⅰ型初等矩阵右乘A，即 $AE_{(p,q)}$ ，表示互换矩阵A的第p列和第q列.

（2） Ⅱ型初等矩阵右乘A，即 $AE_{\alpha(p)}$ ，表示矩阵A的第p列元素乘一个非零常数α.

（3） Ⅲ型初等矩阵右乘A，即 $AE_{(p)+\alpha(q)}$，表示矩阵A的第q列乘以非零常数α后，再与第p列相加.

◊**置换矩阵(permutation matrix)**：每一行和每一列有且仅有一个非零元素1的正方矩阵.

性质：(1) $P^T P = PP^T = I$ (2) $P^{-1} = P^T$

- **交换矩阵(commutation matrix)：** 对向量（矩阵）的各行、列进行等间隔有规律的重排. $K_{mn} = \sum_{j=1}^{n} (e_j^T \otimes I_m \otimes e_j)$ $K_{mn} vec(A) = vec(A^T)$

- **互换矩阵(exchange matrix):**

性质：(1) $J^T = J$ (2) $J^2 = I$

$$J = \begin{bmatrix} 0 & & 1 \\ & & 1 & \\ & \iddots & & \\ 1 & & & 0 \end{bmatrix}$$

矩阵的向量化和反向量化通过reshape函数来实现

```
vec_A = reshape(A,[],1);
A = reshape(A,line,row);
```

```
A =1     2     3
   4     5     6
>> B=reshape(A,1,[])
B =     1     4     2     5     3     6
>> C=reshape(B,2,3)
C =1     2     3
   4     5     6
```

```
>> help reshape
 reshape Reshape array.
   reshape(X,M,N) or reshape(X,[M,N]) returns the M-by-N matrix  whose elements are taken columnwise from X.
An error results   if X does not have M*N elements.
```

按列重排！！！！

问题：按行重排怎么办？（向量化和反向量化）

●移位矩阵(shift matrix):

$$P=\begin{bmatrix} 0 & 1 & 0 & \cdots & 0 \\ 0 & 0 & 1 & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \cdots & 1 \\ 1 & 0 & 0 & \cdots & 0 \end{bmatrix}$$

```
>> help circshift
 circshift Shift positions of elements circularly.
    Y = circshift(X,K) where K is an integer scalar circularly
shifts the elements in the array X by K positions. If X is a
vector and K is  positive, then the values of X are
circularly shifted from the beginning  to the end. If K is
negative, they are shifted from the end to the beginning.
If X is a matrix, circshift shifts along columns.
```

A=eye(3);B=circshift(A,-1);C=B*[10,11,12]'

●**广义置换矩阵(g矩阵):** 每一行和每一列有且仅有一个非零元素的正方矩阵.

★ 一个正方矩阵是g矩阵，当且仅当它可以分解为一个置换矩阵P和一个非奇异的对角矩阵D之积

$$G = PD$$

选择矩阵(selective matrix): 可以抽取（选择）给定向量（矩阵）的某些元素（行/列）的矩阵.

求和矩阵(summing vector): $l=[1,\quad 1,\quad \cdots \quad 1]^T$

◊**Fourier矩阵：** Fourier变换（DFT）

$$\hat{x}(k)=\sum_{n=0}^{N-1}x(n)e^{-j2\pi nk/N}=\sum_{n=0}^{N-1}x(n)w^{nk},k=0,1,\ldots,N-1$$

$$\begin{bmatrix}\hat{x}(0)\\ \hat{x}(1)\\ \vdots\\ \hat{x}(N-1)\end{bmatrix}=\begin{bmatrix}1&1&1&\cdots&1\\ 1&w&w^2&\cdots&w^{N-1}\\ \vdots&\vdots&\vdots&&\vdots\\ 1&w^{N-1}&w^{2(N-1)}&\cdots&w^{(N-1)(N-1)}\end{bmatrix}\begin{bmatrix}x(0)\\ x(1)\\ \vdots\\ x(N-1)\end{bmatrix}$$

$$F=\begin{bmatrix}1&1&1&\cdots&1\\ 1&w&w^2&\cdots&w^{N-1}\\ \vdots&\vdots&\vdots&&\vdots\\ 1&w^{N-1}&w^{2(N-1)}&\cdots&w^{(N-1)(N-1)}\end{bmatrix},w=e^{-j\frac{2\pi}{N}}$$

简记作 $\hat{\mathbf{x}}=F\mathbf{x}$，其中F 称为**Fourier矩阵**. 由 $F^HF=NI$，可得

$$\mathbf{x}=\frac{1}{N}F^H\hat{\mathbf{x}}$$

（**非对称Fourier变换对**）

```
clear all
N=8;
w=exp(-1i*2*pi/N);
for ii=1:N
    V(ii)=w^(ii-1);
end
A=vander(V');
real(A)
imag(A)
```

范得蒙(Vandermonde)矩阵： 最后一列全为1，倒数第二列为一个指定的向量，其他各列是其后列与倒数第二列的点乘积. 可以用一个指定向量生成一个范得蒙矩阵. MATLAB函数vander(V)生成以向量V为基础向量的范得蒙矩阵.

```
>> help vander
 vander Vandermonde matrix.
    A = vander(V), for a vector of length n, .returns the n-by-n
    Vandermonde matrix A. The columns of A are powers of the vector V,
    such that the j-th column is A(:,j) = V(:).^(n-j).
```

ans =

1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000
0.7071	-0.0000	-0.7071	-1.0000	-0.7071	0.0000	0.7071	1.0000
-0.0000	-1.0000	0.0000	1.0000	-0.0000	-1.0000	0.0000	1.0000
-0.7071	0.0000	0.7071	-1.0000	0.7071	-0.0000	-0.7071	1.0000
-1.0000	1.0000	-1.0000	1.0000	-1.0000	1.0000	-1.0000	1.0000
-0.7071	-0.0000	0.7071	-1.0000	0.7071	0.0000	-0.7071	1.0000
0.0000	-1.0000	-0.0000	1.0000	0.0000	-1.0000	-0.0000	1.0000
0.7071	0.0000	-0.7071	-1.0000	-0.7071	-0.0000	0.7071	1.0000

ans =

0	0	0	0	0	0	0	0
0.7071	1.0000	0.7071	-0.0000	-0.7071	-1.0000	-0.7071	0
1.0000	-0.0000	-1.0000	0.0000	1.0000	-0.0000	-1.0000	0
0.7071	-1.0000	0.7071	-0.0000	-0.7071	1.0000	-0.7071	0
-0.0000	0.0000	-0.0000	0.0000	-0.0000	0.0000	-0.0000	0
-0.7071	1.0000	-0.7071	-0.0000	0.7071	-1.0000	0.7071	0
-1.0000	-0.0000	1.0000	0.0000	-1.0000	-0.0000	1.0000	0
-0.7071	-1.0000	-0.7071	-0.0000	0.7071	1.0000	0.7071	0

```
V=zeros(1,8);
X=zeros(8,8);
for ii=1:8
   VV=V;
   VV(ii)=1;
   X(ii,:)=fft2(VV);
end
disp(real(X));
disp(imag(X));
```

1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000
1.0000	0.7071	0	-0.7071	-1.0000	-0.7071	0	0.7071
1.0000	0	-1.0000	0	1.0000	0	-1.0000	0
1.0000	-0.7071	0	0.7071	-1.0000	0.7071	0	-0.7071
1.0000	-1.0000	1.0000	-1.0000	1.0000	-1.0000	1.0000	-1.0000
1.0000	-0.7071	0	0.7071	-1.0000	0.7071	0	-0.7071
1.0000	0	-1.0000	0	1.0000	0	-1.0000	0
1.0000	0.7071	0	-0.7071	-1.0000	-0.7071	0	0.7071

0	0	0	0	0	0	0	0
0	-0.7071	-1.0000	-0.7071	0	0.7071	1.0000	0.7071
0	-1.0000	0	1.0000	0	-1.0000	0	1.0000
0	-0.7071	1.0000	-0.7071	0	0.7071	-1.0000	0.7071
0	0	0	0	0	0	0	0
0	0.7071	-1.0000	0.7071	0	-0.7071	1.0000	-0.7071
0	1.0000	0	-1.0000	0	1.0000	0	-1.0000
0	0.7071	1.0000	0.7071	0	-0.7071	-1.0000	-0.7071

```
clear all
N=8;
w=exp(-1i*2*pi/N);
V=zeros(1,N);
for ii=1:N
    V(ii)=w^(ii-1);
end
A=vander(V);
A0=real(A)
B=imag(A)
C=flipud(B)
```

```
A0 = 1.0000    1.0000    1.0000    1.0000    1.0000    1.0000    1.0000    1.0000
     0.7071   -0.0000   -0.7071   -1.0000   -0.7071    0.0000    0.7071    1.0000
    -0.0000   -1.0000    0.0000    1.0000   -0.0000   -1.0000    0.0000    1.0000
    -0.7071    0.0000    0.7071   -1.0000    0.7071   -0.0000   -0.7071    1.0000
    -1.0000    1.0000   -1.0000    1.0000   -1.0000    1.0000   -1.0000    1.0000
    -0.7071   -0.0000    0.7071   -1.0000    0.7071    0.0000   -0.7071    1.0000
     0.0000   -1.0000   -0.0000    1.0000    0.0000   -1.0000   -0.0000    1.0000
     0.7071    0.0000   -0.7071   -1.0000   -0.7071   -0.0000    0.7071    1.0000
B =0         0         0         0         0         0         0         0
     0.7071    1.0000    0.7071   -0.0000   -0.7071   -1.0000   -0.7071         0
     1.0000   -0.0000   -1.0000    0.0000    1.0000   -0.0000   -1.0000         0
     0.7071   -1.0000    0.7071   -0.0000   -0.7071    1.0000   -0.7071         0
    -0.0000    0.0000   -0.0000    0.0000   -0.0000    0.0000   -0.0000         0
    -0.7071    1.0000   -0.7071   -0.0000    0.7071   -1.0000    0.7071         0
    -1.0000   -0.0000    1.0000    0.0000   -1.0000   -0.0000    1.0000         0
    -0.7071   -1.0000   -0.7071   -0.0000    0.7071    1.0000    0.7071         0
C =-0.7071   -1.0000   -0.7071   -0.0000    0.7071    1.0000    0.7071         0
    -1.0000   -0.0000    1.0000    0.0000   -1.0000   -0.0000    1.0000         0
    -0.7071    1.0000   -0.7071   -0.0000    0.7071   -1.0000    0.7071         0
    -0.0000    0.0000   -0.0000    0.0000   -0.0000    0.0000   -0.0000         0
     0.7071   -1.0000    0.7071   -0.0000   -0.7071    1.0000   -0.7071         0
     1.0000   -0.0000   -1.0000    0.0000    1.0000   -0.0000   -1.0000         0
     0.7071    1.0000    0.7071   -0.0000   -0.7071   -1.0000   -0.7071         0
     0         0         0         0         0         0         0         0
```

%flipud函数和fliplr函数（上下翻转，左右翻转）

```
clear all
N=8;
w=exp(-1i*2*pi/N);
V=zeros(1,N);
for ii=1:N
    V(ii)=w^(ii-1);
end
A=vander(V);
A0=real(A)
B=imag(A)
%flipud函数和fliplr函数
C=flipud(B)
J=zeros(N,N);
ii=1;
for jj=N:-1:1
    J(ii,jj)=1;
    ii=ii+1;
end
disp(J);
D=C*J
E=J*C
```

```
     0     0     0     0     0     0     0     1
     0     0     0     0     0     0     1     0
     0     0     0     0     0     1     0     0
     0     0     0     0     1     0     0     0
     0     0     0     1     0     0     0     0
     0     0     1     0     0     0     0     0
     0     1     0     0     0     0     0     0
     1     0     0     0     0     0     0     0
```

J=flipud(eye(8))

```
D =  0    0.7071    1.0000    0.7071   -0.0000   -0.7071   -1.0000   -0.7071
     0    1.0000   -0.0000   -1.0000    0.0000    1.0000   -0.0000   -1.0000
     0    0.7071   -1.0000    0.7071   -0.0000   -0.7071    1.0000   -0.7071
     0   -0.0000    0.0000   -0.0000    0.0000   -0.0000    0.0000   -0.0000
     0   -0.7071    1.0000   -0.7071   -0.0000    0.7071   -1.0000    0.7071
     0   -1.0000   -0.0000    1.0000    0.0000   -1.0000   -0.0000    1.0000
     0   -0.7071   -1.0000   -0.7071   -0.0000    0.7071    1.0000    0.7071
     0         0         0         0         0         0         0         0
```

```
E = 0          0          0          0          0          0          0          0
    0.7071    1.0000    0.7071   -0.0000   -0.7071   -1.0000   -0.7071         0
    1.0000   -0.0000   -1.0000    0.0000    1.0000   -0.0000   -1.0000         0
    0.7071   -1.0000    0.7071   -0.0000   -0.7071    1.0000   -0.7071         0
   -0.0000    0.0000   -0.0000    0.0000   -0.0000    0.0000   -0.0000         0
   -0.7071    1.0000   -0.7071   -0.0000    0.7071   -1.0000    0.7071         0
   -1.0000   -0.0000    1.0000    0.0000   -1.0000   -0.0000    1.0000         0
   -0.7071   -1.0000   -0.7071   -0.0000    0.7071    1.0000    0.7071         0
```

若

$$\bar{F}=\frac{1}{\sqrt{N}}\begin{bmatrix}1 & 1 & 1 & \cdots & 1\\ 1 & w & w^2 & \cdots & w^{N-1}\\ \vdots & \vdots & \vdots & & \vdots\\ 1 & w^{N-1} & w^{2(N-1)} & \cdots & w^{(N-1)(N-1)}\end{bmatrix}, w=e^{-j\frac{2\pi}{N}}$$

则

$$\hat{x}(k)=\frac{1}{\sqrt{N}}\sum_{n=0}^{N-1}x(n)e^{-j2\pi nk/N}, k=0,1,\dots, N-1$$

$$x(n)=\frac{1}{\sqrt{N}}\sum_{k=0}^{N-1}\hat{x}(k)e^{j2\pi nk/N}, n=0,1,\dots, N-1$$

即有 $\hat{\mathbf{x}}=\bar{F}\mathbf{x}$ 和 $\mathbf{x}=\bar{F}^H\hat{\mathbf{x}}$ （对称Fourier变换对）

注：此处“对称”是指正变换和反变换的数学表达式形式上几乎一样。“对称”概念，一般指针对某种特定操作，操作前后，其数学形式一样。命题的前提是“操作”……

$$F=\begin{bmatrix}1 & 1 & 1 & \cdots & 1\\ 1 & w & w^2 & \cdots & w^{N-1}\\ \vdots & \vdots & \vdots & & \vdots\\ 1 & w^{N-1} & w^{2(N-1)} & \cdots & w^{(N-1)(N-1)}\end{bmatrix}, w=e^{-j\frac{2\pi}{N}}$$

$$\overline{F} = \frac{1}{\sqrt{N}} \begin{bmatrix} 1 & 1 & 1 & \cdots & 1 \\ 1 & w & w^{2} & \cdots & w^{N-1} \\ \vdots & \vdots & \vdots & & \vdots \\ 1 & w^{N-1} & w^{2(N-1)} & \cdots & w^{(N-1)(N-1)} \end{bmatrix},$$

$$w = e^{-j\frac{2\pi}{N}} = \cos(-\frac{2\pi}{N}) + j\sin(-\frac{2\pi}{N})$$

对称Fourier矩阵的性质

当说明是对称傅里叶矩阵时，可以将记号 $\overline{F}$ 简写为F.记号使用的灵活性.

(1) $F^T = F$

(2) $F^{-1} = F^*$

(3) $F^2 = P = [e_1, e_n, e_{n-1}, \ldots, e_2]$，其中$e_k$是基本向量

(4) $F^4 = I$

(5) $\sqrt{n}F = C + jS$，其中矩阵C和S的元素分别为

$$C_{ij} = \cos\left(\frac{2\pi}{n}(i-1)(j-1)\right)$$

$$S_{ij} = \sin\left(\frac{2\pi}{n}(i-1)(j-1)\right)$$

式中i，j=1，2，...,n.

(6) $CS = SC$ 和 $C^2 + S^2 = nI$

(7) 归一化Fourier矩阵 $\overline{F}$ 是对称的Vandermonde矩阵，也是正交矩阵.

练习题：请用程序验证本页所述性质。

```
clear all
N=4;
w=exp(-1i*2*pi/N);
V=zeros(1,N);
for ii=1:N
    V(ii)=w^(ii-1);
end
A=vander(V);
B=fliplr(A);
F=B;
for ii=1:N
    for jj=1:N
        k=((ii-1)*(jj-1));
        FF(ii,jj)=w^k;
    end
end
F-FF
C=real(F);
S=imag(F);
C*C+S*S
F*F
```

```
>> m11

ans =

     0     0     0     0
     0     0     0     0
     0     0     0     0
     0     0     0     0


ans =

    4.0000   -0.0000         0    0.0000
   -0.0000    4.0000   -0.0000         0
         0   -0.0000    4.0000   -0.0000
    0.0000         0   -0.0000    4.0000
ans =

   4.0000 + 0.0000i  -0.0000 - 0.0000i   0.0000 - 0.0000i   0.0000 - 0.0000i
  -0.0000 - 0.0000i   0.0000 - 0.0000i   0.0000 - 0.0000i   4.0000 + 0.0000i
   0.0000 - 0.0000i   0.0000 - 0.0000i   4.0000 + 0.0000i  -0.0000 - 0.0000i
   0.0000 - 0.0000i   4.0000 + 0.0000i  -0.0000 - 0.0000i   0.0000 - 0.0000i
```

➢一维离散傅立叶变换

对有限长序列$f(x)$, x=0,1,2,…,N-1,可定义一维离散傅立叶变换对如下：

$$F(u)=\sum_{x=0}^{N-1} f(x)\exp(-2\pi ux/N),\ u=0,1,\cdots,N-1 \quad \text{(A)}$$

$$f(x)=\frac{1}{N}\sum_{x=0}^{N-1} F(u)\exp(2\pi ux/N),\ x=0,1,\cdots,N-1 \quad \text{(B)}$$

令 $W=e^{-j\frac{2\pi}{N}}$ 则以上两式可写为：

$$F(u)=\sum_{x=0}^{N-1} f(x)W^{xu},\ u=0,1,\cdots,N-1$$

$$f(x)=\frac{1}{N}\sum_{x=0}^{N-1} F(u)W^{-ux},\ x=0,1,\cdots,N-1$$

傅立叶变换对也可以简记为： $f(x)\Leftrightarrow F(u)$

注意到(A)和(B)的结构，傅立叶变换也可以写成矩阵形式：

$$\begin{pmatrix} f(0) \\ f(1) \\ \cdots \\ f(N-1) \end{pmatrix} = \begin{pmatrix} W^0 & W^0 & W^0 & \cdots & W^0 \\ W^0 & W^{-1\times 1} & W^{-2\times 1} & \cdots & W^{-(N-1)\times 1} \\ \cdots & \cdots & \cdots & \cdots & \cdots \\ W^0 & W^{-(N-1)\times 1} & W^{-(N-1)\times 2} & \cdots & W^{-(N-1)\times(N-1)} \end{pmatrix} \begin{pmatrix} F(0) \\ F(1) \\ \cdots \\ F(N-1) \end{pmatrix}$$

$$\begin{pmatrix} F(0) \\ F(1) \\ \cdots \\ F(N-1) \end{pmatrix} = \begin{pmatrix} W^0 & W^0 & W^0 & \cdots & W^0 \\ W^0 & W^{1\times 1} & W^{2\times 1} & \cdots & W^{(N-1)\times 1} \\ \cdots & \cdots & \cdots & \cdots & \cdots \\ W^0 & W^{1\times(N-1)} & W^{2\times(N-1)} & \cdots & W^{(N-1)\times(N-1)} \end{pmatrix} \begin{pmatrix} f(0) \\ f(1) \\ \cdots \\ f(N-1) \end{pmatrix}$$

➢二维离散傅立叶变换

将连续傅立叶变换离散化，可得离散傅立叶变换.（详细推导过程请参考相关文献）

设有离散函数$f(x,y)$，则可定义其傅立叶变换对：

$$F(u,v)=\sum_{x=0}^{M-1}\sum_{y=0}^{N-1}f(x,y)\exp(-2j\pi(\frac{ux}{M}+\frac{vy}{N}))$$

$$u=0,1,\cdots M-1;v=0,1,\cdots N-1$$

$$f(x,y)=\frac{1}{MN}\sum_{u=0}^{M-1}\sum_{v=0}^{N-1}F(u,v)\exp(2j\pi(\frac{ux}{M}+\frac{vy}{N}))$$

$$x=0,1,\cdots M-1;y=0,1,\cdots N-1$$

二维离散傅立叶变换的矩阵表达式

令$P(u,x)=\frac{1}{\sqrt{N}}e^{-j\frac{2\pi}{N}ux}$, $Q(y,v)=\frac{1}{\sqrt{N}}e^{-j\frac{2\pi}{N}yv}$，则

$$F(u,v)=\sum_{x,y}P(u,x)f(x,y)Q(y,v)$$

$$f(x,y)=\sum_{u,v}P^{*}(x,u)F(u,v)Q^{*}(v,y)$$

令$F=[F(u,v)]_{N\times N}, f=[f(x,y)]_{N\times N}$

$P=[P(u,x)]_{N\times N}, Q=[Q(y,v)]_{N\times N}$

注意到矩阵乘法的定义，可知：

$$F=PfQ, f=P^{H}FQ^{H}$$

P是正交矩阵（酉矩阵）

一般地，若P是正交矩阵，则称

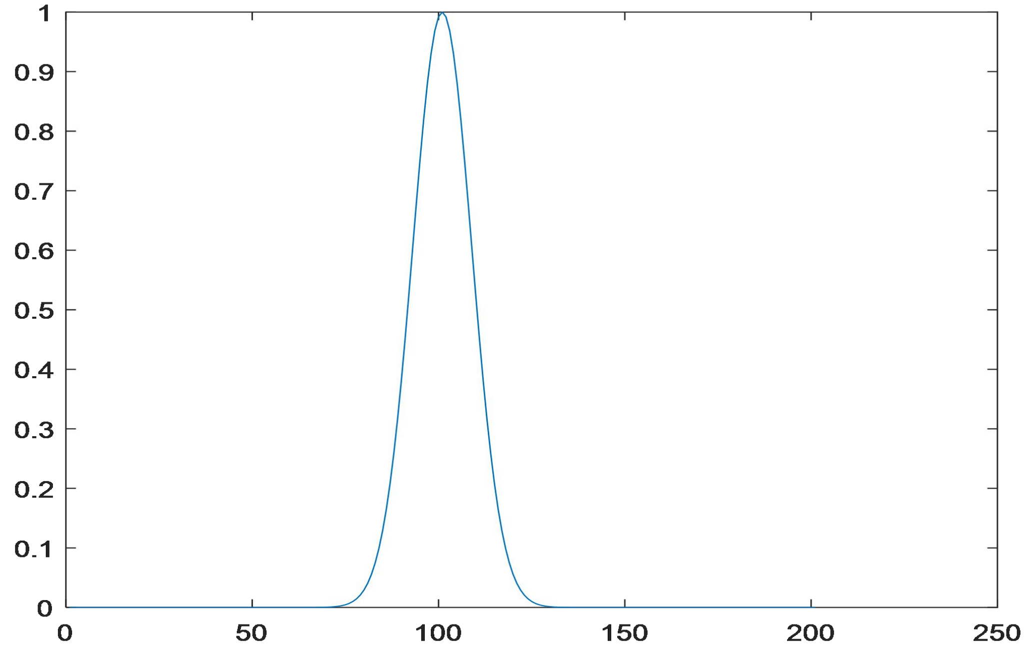
$F=PfP^{T}$是图象的正交变换

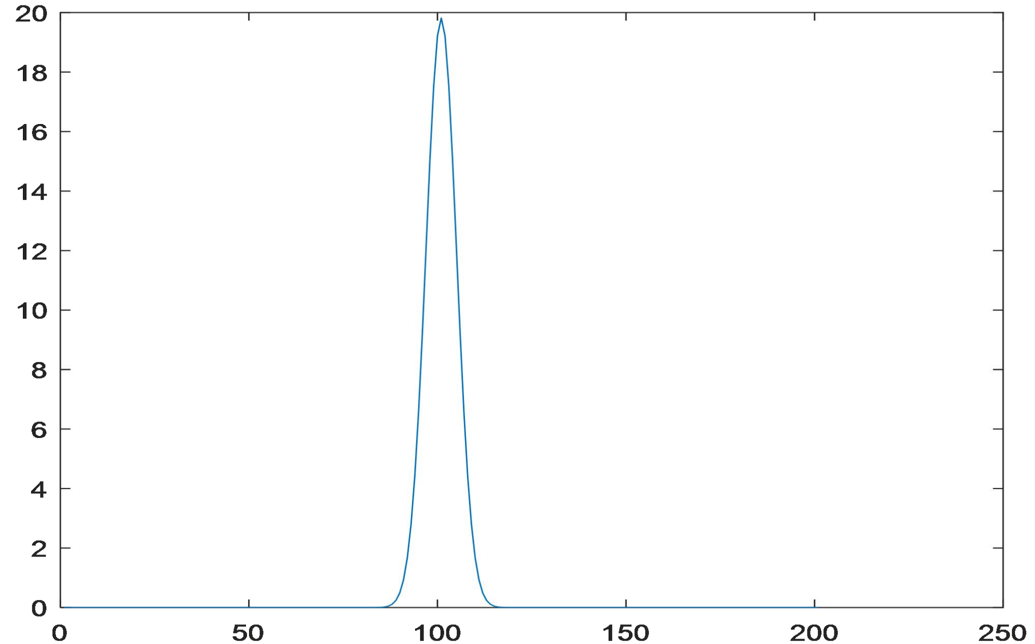
案例1（1） 傅里叶变换（一维）

```
x=zeros(1,201);
z=x;
for ii=-100:100
    x(ii+101)=exp(-(ii/50)^2/.0500);
end
y=fft2(x);
figure(1);
plot(x);
figure(2);
for ii=1:201
    jj=rem(ii+100,201)+1;
    z(ii)=y(jj);
end
plot(abs(z));
```

想说明：高斯函数的傅里叶变换还是高斯函数.

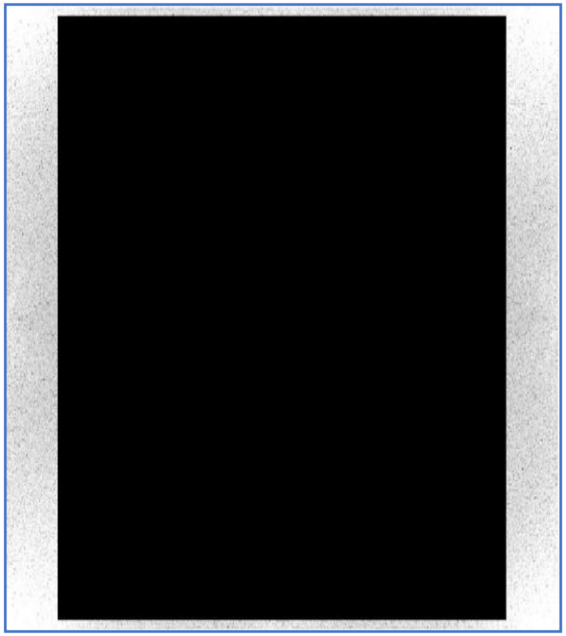
在傅里叶变换下，高斯函数在函数集合中的地位, 相当于整数1在实数中的地位.





案例1 (2) 傅里叶变换(二维)

(a) 原图



(b)剔除原图的傅里叶变换的80%数据

(c) 图(b)数据的反变换

结论：基于傅里叶变换，可以对图像进行数据压缩和模糊化操作

◊**对称矩阵：** $A = A^T$

◊**反对称矩阵：** $A = -A^T$

◊**斜对称(pre-symmetric)矩阵：**

$$A_{n\times n} = (a_{ij}) = (a_{n-j+1,n-i+1}), \forall i, j = 1,2,\dots,n$$

亦称为**交叉对称矩阵**或**次对称矩阵**.

若A是任意矩阵，则基于A可构造对称阵和反对称矩阵：

$$B = (A + A^T)/2,$$

$$C = (A - A^T)/2,$$

$$A = B + C.$$

性质：

（1）若A为斜对称矩阵，则 A^T 亦为斜对称矩阵.

（2）若A为斜对称矩阵，则 A^{-1} 亦为斜对称矩阵.

（3）若A为斜对称矩阵，且J为互换矩阵，则有

$$JA^TJ = A \qquad (JA)^T = JA$$

```
A=rand(5,5)
B=zeros(5,5);
for ii=1:5
    for jj=1:5
        B(ii,jj)=A(5-jj+1,5-ii+1);
    end
end
disp(B);
```

A =

0.8147	0.0975	0.1576	0.1419	0.6557
0.9058	0.2785	0.9706	0.4218	0.0357
0.1270	0.5469	0.9572	0.9157	0.8491
0.9134	0.9575	0.4854	0.7922	0.9340
0.6324	0.9649	0.8003	0.9595	0.6787

0.6787	0.9340	0.8491	0.0357	0.6557
0.9595	0.7922	0.9157	0.4218	0.1419
0.8003	0.4854	0.9572	0.9706	0.1576
0.9649	0.9575	0.5469	0.2785	0.0975
0.6324	0.9134	0.1270	0.9058	0.8147

把矩阵按次对角线进行了翻转.

```
A=rand(5,5);
B=A+A'
C=flipud(A);
disp(C);
```

B =

1.5155	1.4492	1.2157	1.0942	0.6610
1.4492	0.0637	0.9718	0.4277	0.5427
1.2157	0.9718	0.6342	1.7157	0.6808
1.0942	0.4277	1.7157	1.5904	0.8962
0.6610	0.5427	0.6808	0.8962	1.5094

C =

0.6610	0.5427	0.6808	0.8962	1.5094
1.0942	0.4277	1.7157	1.5904	0.8962
1.2157	0.9718	0.6342	1.7157	0.6808
1.4492	0.0637	0.9718	0.4277	0.5427
1.5155	1.4492	1.2157	1.0942	0.6610

◊中心对称（centro-symmetric）矩阵：

$$A_{n\times n}=(a_{ij})=(a_{n-i+1,n-j+1}),\forall i,j=1,2,\ldots,n$$

性质：

（1）若A为中心对称矩阵，则A^T亦为中心对称矩阵.

（2）若A为斜对称矩阵，且J为互换矩阵，则有

$$JA^TJ=A$$

（3）任一2n阶中心对称矩阵A可写作

$$A=\begin{bmatrix} X & YJ \\ JY & JXJ \end{bmatrix}, X,Y\in R^{n\times n}$$

练习题：任意奇数阶的中心对称矩阵如何生成？

```
X=rand(3);
Y=rand(3);
J=flipud(eye(3));
A=[X    Y*J
   J*Y    J*X*J];
disp(A);
```

```
>> m12
    0.6892    0.0838    0.1524    0.7749    0.1067    0.9961
    0.7482    0.2290    0.8258    0.8173    0.9619    0.0782
    0.4505    0.9133    0.5383    0.8687    0.0046    0.4427
    0.4427    0.0046    0.8687    0.5383    0.9133    0.4505
    0.0782    0.9619    0.8173    0.8258    0.2290    0.7482
    0.9961    0.1067    0.7749    0.1524    0.0838    0.6892
```

```
A=rand(5,5)
B=A;
for ii=1:5
    for jj=1:5
        if ii+jj>5
            B(ii,jj)=A(5-jj+1,5-ii+1);
        end
    end
end
C=(A-A')/2;
disp(B);
disp(C);
```

0.1622	0.6020	0.4505	0.8258	0.1067
0.7943	0.2630	0.0838	0.5383	0.9619
0.3112	0.6541	0.2290	0.9961	0.0046
0.5285	0.6892	0.9133	0.0782	0.7749
0.1656	0.7482	0.1524	0.4427	0.8173

0.1622	0.6020	0.4505	0.8258	0.1067
0.7943	0.2630	0.0838	0.5383	0.8258
0.3112	0.6541	0.2290	0.0838	0.4505
0.5285	0.6892	0.6541	0.2630	0.6020
0.1656	0.5285	0.3112	0.7943	0.1622

0	-0.0962	0.0697	0.1486	-0.0295
0.0962	0	-0.2851	-0.0754	0.1069
-0.0697	0.2851	0	0.0414	-0.0739
-0.1486	0.0754	-0.0414	0	0.1661
0.0295	-0.1069	0.0739	-0.1661	0

生成一个斜对称矩阵和一个反对称矩阵!

（4） 对于（3）式的2n阶中心对称矩阵A有

$$\det(A) = \det(X+Y)\det(X-Y)$$

$$A^{-1} = \begin{bmatrix} C & RJ \\ JR & ICJ \end{bmatrix}$$

其中

$$C = \frac{1}{2}\left[(X+Y)^{-1} + (X-Y)^{-1}\right]$$

$$R = \frac{1}{2}[(X+Y)^{-1} - (X-Y)^{-1}]$$

练习题：请编程验证

◊复共轭对称（Hermitian）矩阵：$A = A^H$

◊反Hermitian矩阵：$A = -A^H$

◊斜复共轭对称(Hermitian)矩阵：

$$A_{n\times n} = (a_{ij}) = (a^*_{n-j+1,n-i+1}), \forall i, j = 1,2,\ldots,n$$

◊中央（中心）复共轭对称(Hermitian)矩阵：

$$A_{n\times n} = (a_{ij}) = (a^*_{n-i+1,n-j+1}), \forall i, j = 1,2,\ldots,n$$

◊Vandermonde矩阵：

$$A = \begin{bmatrix} 1 & x_1 & x_1^2 & \cdots & x_1^{n-1} \\ 1 & x_2 & x_2^2 & \cdots & x_2^{n-1} \\ \vdots & \vdots & \vdots & & \vdots \\ 1 & x_n & x_n^2 & \cdots & x_n^{n-1} \end{bmatrix} \text{ 或 } A = \begin{bmatrix} 1 & 1 & \cdots & 1 \\ x_1 & x_2 & \cdots & x_n \\ x_1^2 & x_2^{n-1} & \cdots & x_n^{n-1} \\ \vdots & \vdots & & \vdots \\ x_1^{n-1} & x_2^{n-1} & \cdots & x_n^{n-1} \end{bmatrix}$$

性质：当 $x_1, \cdots, x_n$ 各异时，矩阵A非奇异.

复Vandermonde矩阵的逆矩阵

$$A=\begin{bmatrix} 1 & 1 & \cdots & 1 \\ a_1 & a_2 & \cdots & a_n \\ \vdots & \vdots & & \vdots \\ a_1^{n-1} & a_2^{n-1} & \cdots & a_n^{n-1} \end{bmatrix}, a_k \in C$$

$$A^{-1}=$$

$$\begin{bmatrix} \dfrac{\sigma_{n-1}(a_2,a_3,\cdots,a_n)}{\prod\limits_{k=1,k\neq 1}^{n}(a_k-a_1)} & -\dfrac{\sigma_{n-2}(a_2,a_3,\cdots,a_n)}{\prod\limits_{k=1,k\neq 1}^{n}(a_k-a_1)} & \cdots & \dfrac{(-1)^{n+1}}{\prod\limits_{k=1,k\neq 1}^{n}(a_k-a_1)} \\ \dfrac{\sigma_{n-1}(a_1,a_3,\cdots,a_n)}{\prod\limits_{k=1,k\neq 2}^{n}(a_k-a_2)} & -\dfrac{\sigma_{n-2}(a_1,a_3,\cdots,a_n)}{\prod\limits_{k=1,k\neq 2}^{n}(a_k-a_2)} & \cdots & \dfrac{(-1)^{n+2}}{\prod\limits_{k=1,k\neq 2}^{n}(a_k-a_2)} \\ \vdots & \vdots & \vdots & \vdots \\ \dfrac{\sigma_{n-1}(a_1,a_2,\cdots,a_{n-1})}{\prod\limits_{k=1,k\neq n}^{n}(a_k-a_n)} & -\dfrac{\sigma_{n-2}(a_2,a_3,\cdots,a_n)}{\prod\limits_{k=1,k\neq n}^{n}(a_k-a_n)} & \cdots & \dfrac{(-1)^{n+n}}{\prod\limits_{k=1,k\neq n}^{n}(a_k-a_n)} \end{bmatrix}$$

Permutation and Combination

$\sigma_m(b_1,b_2,\cdots,b_k)=k$个元素$b_1,b_2,\cdots,b_k$的所有$m$个元素组合的元素乘积之和.

◊Toeplitz矩阵：

$$A=\begin{bmatrix} a_0 & a_{-1} & a_{-2} & \cdots & a_{-n} \\ a_1 & a_0 & a_{-1} & \cdots & a_{-n+1} \\ a_2 & a_1 & a_0 & \ddots & \vdots \\ \vdots & \vdots & \ddots & \ddots & a_{-1} \\ a_n & a_{n-1} & \cdots & a_1 & a_0 \end{bmatrix}=(a_{i-j})_{i,j=0}^{n}$$

最多有2n-1个不同的数.n是矩阵的阶数.

★任一Toeplitz矩阵均为斜对称矩阵;

★<u>任一对称Toeplitz矩阵均为对称的中心对称矩阵.</u>

对称Toeplitz矩阵：满足对称关系$a_{-i}=a_i, i=1,2,\dots,n$ 的Toeplitz矩阵.

★对称Toeplitz矩阵可仅由其第一行元素完全描述.因此，常将其简记作

$$A = Toep\,[a_0, a_1, \dots, a_n]$$

```
>> help  toeplitz
toeplitz - Toeplitz matrix
   This MATLAB function returns a nonsymmetric Toeplitz matrix with c as its first column and r as its first row.
   T = toeplitz(c,r)
   T = toeplitz(r)
```

Hermitian Toeplitz 矩阵：

$$A=\begin{bmatrix} a_0 & a_1^* & a_2^* & \cdots & a_n^* \\ a_1 & a_0 & a_1^* & \cdots & a_{n-1}^* \\ a_2 & a_1 & a_0 & \ddots & \vdots \\ \vdots & \vdots & \ddots & \ddots & a_1^* \\ a_n & a_{n-1} & \cdots & a_1 & a_0 \end{bmatrix}$$

反Hermitian Toeplitz 矩阵：

$$A_S=\begin{bmatrix} 0 & -a_1^* & -a_2^* & \cdots & -a_n^* \\ a_1 & 0 & -a_1^* & \cdots & -a_{n-1}^* \\ a_2 & a_1 & 0 & \ddots & \vdots \\ \vdots & \vdots & \ddots & \ddots & -a_1^* \\ a_n & a_{n-1} & \cdots & a_1 & 0 \end{bmatrix}$$

```
>> c=[1 2 3 4]
>> r=[10 11 12 13]
>> toeplitz(c,r)
警告: 输入列的第一个元素与输入行的第一个元素不匹配.在对角线冲突中，列具有更高优先级.
ans =
     1    11    12    13
     2     1    11    12
     3     2     1    11
     4     3     2     1
```

◊**Hankel矩阵：**

$$H=\begin{bmatrix} h_0 & h_1 & h_2 & \cdots & h_n \\ h_1 & h_2 & h_3 & \cdots & h_{n+1} \\ h_2 & h_3 & h_4 & \cdots & h_{n+2} \\ \vdots & \vdots & \vdots & & \vdots \\ h_n & h_{n+1} & h_{n+2} & \cdots & h_{2n} \end{bmatrix}$$

最多有2n-1个不同的数. n是矩阵的阶数.

性质： （1） JH 和 HJ 均为Toeplitz矩阵;

（2） $(JH)^T = HJ.$

无穷Hankel矩阵:

$$S=\begin{bmatrix} s_0 & s_1 & s_2 & \cdots \\ s_1 & s_2 & s_3 & \cdots \\ s_2 & s_3 & s_4 & \cdots \\ \vdots & \vdots & \vdots & \vdots \end{bmatrix}$$

```
>> help hankel
hankel - Hankel 矩阵
   此 MATLAB 函数 返回其第一列是 c 并且其第一个反对角线下方的元素为零的 Hankel 方阵.
   H = hankel(c)
   H = hankel(c,r)
```

```
ans =
    1    2    3    4
    2    3    4   11
    3    4   11   12
    4   11   12   13
```

定理： 无穷阶Hankel矩阵 $s=[s_{i+k}]_0^\infty$ 具有有限秩r，当且仅当存在r个常数 $\alpha_1,\alpha_2,\ldots,\alpha_r$，使得

$$s_l=\sum_{i=1}^{r}\alpha_i s_{l-i}, l=r,r+1,\ldots$$

成立，其中r是具有该性质的最小整数.

应用1：最小二乘拟合

设 $(x_j, y_j), j=1,2,\dots,m$ 为一组观测数据. 试寻找一个n次非零多项式

$$f(x) = a_0 + a_1 x + \cdots + a_n x^n = \sum_{i=0}^{n} a_i x^i \,(n < m)$$

以最小化

$$J = \sum_{j=1}^{m} \left| y_j - f(x_j) \right|^2$$

$$= \sum_{j=1}^{m} \left| y_j - \sum_{i=1}^{n} a_i (x_j)^i \right|^2$$

为此，令

$$\frac{\partial J}{\partial a_k} = -2\sum_{j=1}^{m} (x_j)^k \left[y_j - \sum_{i=0}^{n} a_i (x_j)^i \right] = 0, k = 0,1,\dots, n$$

即

$$\frac{\partial J}{\partial a_k} = -2\sum_{j=1}^{m} (x_j)^k y_j + 2\sum_{i=0}^{n} a_i \sum_{j=1}^{m} (x_j)^{i+k} = 0, \quad k = 0,1,\dots, n$$

若记 $\alpha_{k+i}=\sum_{j=1}^{m}(x_j)^{k+i}, \beta_k=\sum_{j=1}^{m}(x_j)^k y_j,\quad k=0,1,\ldots,n$，则整理上式可得

$$\sum_{j=1}^{m}(x_j)^k y_j=\sum_{i=0}^{n}a_i\sum_{j=1}^{m}(x_j)^{k+i}\Rightarrow \beta_k=\sum_{i=0}^{n}\alpha_{k+i}a_i$$

即 $H_{n+1}a=b$ 其中

$$H_{n+1}=\begin{array}{c}\begin{bmatrix}\alpha_0 & \alpha_1 & \alpha_2 & \cdots & \alpha_n\\ \alpha_1 & \alpha_2 & \alpha_3 & \cdots & \alpha_{n+1}\\ \alpha_2 & \alpha_3 & \alpha_4 & \cdots & \alpha_{n+2}\\ \vdots & \vdots & \vdots & & \vdots\\ \alpha_n & \alpha_{n+1} & \alpha_{n+2} & \cdots & \alpha_{2n}\end{bmatrix}\end{array}\begin{array}{l}\leftarrow k=0\\ \leftarrow k=1\\ \leftarrow k=2\\ \quad\vdots\\ \leftarrow k=n\end{array}$$

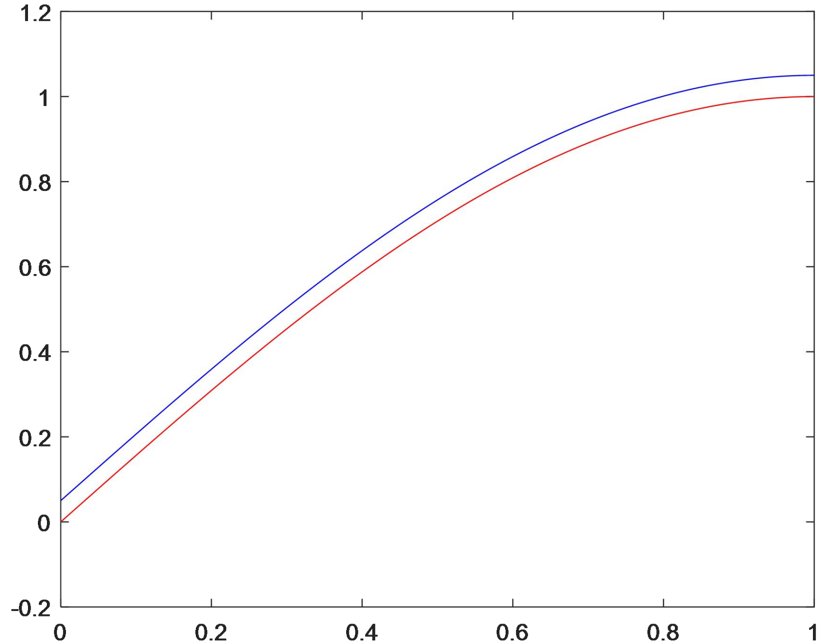
为一Hankel矩阵，

$$a=(a_0,a_1,\cdots,a_n)^T,\qquad b=(\beta_0,\beta_1,\cdots,\beta_n)^T$$

即最小二乘数据拟合问题，可以转化为求解一个以Hankel矩阵为系数矩阵的线性方程的问题.

```
N=7;%阶数，N+1个系数
M=20;%数据采样点个数
x=rand(M,1);
y=sin(pi/2*x);
a=zeros(1,2*N+1);
b=zeros(N+1,1);
for k=1:2*N+1
    for ii=1:M
        a(k)=a(k)+x(ii)^(k-1);
    end
end
for k=1:N+1
    for ii=1:M
        b(k)=b(k)+y(ii)*x(ii)^(k-1);
    end
end
```

```
H=zeros(N+1,N+1);
for ii=1:N+1
    for jj=1:N+1
        H(ii,jj)=a(ii+jj-1);
    end
end
disp(H);
A=H\b;
 disp(A');
 tt=0:0.01:1;
yy=zeros(1,length(tt));
for ii=1:length(tt)
    for jj=1:N+1
        yy(ii)=yy(ii)+A(jj)*tt(ii)^(jj-1);
    end
end
plot(tt,yy,'Color','r');
hold on
plot(tt,0.05+sin(tt*pi/2),'Color','b');
```



>> ttt

20.0000	9.8706	6.4347	4.6450	3.5666	2.8572	2.3589	1.9910
9.8706	6.4347	4.6450	3.5666	2.8572	2.3589	1.9910	1.7085
6.4347	4.6450	3.5666	2.8572	2.3589	1.9910	1.7085	1.4849
4.6450	3.5666	2.8572	2.3589	1.9910	1.7085	1.4849	1.3036
3.5666	2.8572	2.3589	1.9910	1.7085	1.4849	1.3036	1.1537
2.8572	2.3589	1.9910	1.7085	1.4849	1.3036	1.1537	1.0279
2.3589	1.9910	1.7085	1.4849	1.3036	1.1537	1.0279	0.9209
1.9910	1.7085	1.4849	1.3036	1.1537	1.0279	0.9209	0.8289

-0.0000	1.5708	-0.0000	-0.6457	-0.0011	0.0820	-0.0026	-0.0034

```
x=[1 2 3 4 5 6 7];
y=[2 4 6 7 8 9 9];
hankel(x)
%Hankel矩阵：是指每一条副对角线上的元素都相等的方阵.
hankel(x,y)
%返回一个m x n的Hankel矩阵，它的第一列向量为x，最后一行为向量y
```

```
ans =
     1     2     3     4     5     6     7
     2     3     4     5     6     7     0
     3     4     5     6     7     0     0
     4     5     6     7     0     0     0
     5     6     7     0     0     0     0
     6     7     0     0     0     0     0
     7     0     0     0     0     0     0
警告: 输入列的最后一个元素与输入行的第一个元素不匹配.在反对角线冲突中，列具有更高优先级.
> In hankel (line 27)
  In Untitled2 (line 25)


ans =
     1     2     3     4     5     6     7
     2     3     4     5     6     7     4
     3     4     5     6     7     4     6
     4     5     6     7     4     6     7
     5     6     7     4     6     7     8
     6     7     4     6     7     8     9
     7     4     6     7     8     9     9
```

◊右循环矩阵

$$C_R = (c_{ij}), \text{其中 } c_{ij} = \begin{cases} c_{j-i}, & j-i \geq 0 \\ c_{n+j-i}, & j-i < 0 \end{cases}$$

左循环矩阵

$$C_L = (c_{ij}), \text{其中 } c_{ij} = \begin{cases} c_{n+1-i-j}, & j+i \leq n+1 \\ c_{2n+1-i-j}, & j+i > n+1 \end{cases}$$

性质：

（1）右循环矩阵是一特殊的Toeplitz矩阵.

（2）左循环矩阵是一特殊的Hankel矩阵.

（3）循环矩阵可由其第一行（列）元素完全确定，并记为

$$C_R = C_R(c_0, c_1, \cdots, c_{n-1})$$
$$C_L = C_L(c_0, c_1, \cdots, c_{n-1})$$

循环矩阵是一种特殊形式的 Toeplitz矩阵，它的行向量的每个元素都是前一个行向量各元素依次右移一个位置得到的结果。由于可以用离散傅立叶变换快速解循环矩阵方程，所以在数值分析中有重要的应用。（百度百科）

```
n=8; A=rand(1,n);
for ii=1:n
    for jj=1:n
        if jj-ii>=0
            B(ii,jj)=A(jj-ii+1);
        else
            B(ii,jj)=A(n+jj-ii+1);
    end
    end
end
disp(B);
```

```
c=rand(1,8);
cc=fliplr(c);
ccc=circshift(cc,1);
A=toeplitz(c,ccc)
```

```
A =
    0.4218    0.9340    0.8491    0.0357    0.6557    0.9595    0.7922    0.9157
    0.9157    0.4218    0.9340    0.8491    0.0357    0.6557    0.9595    0.7922
    0.7922    0.9157    0.4218    0.9340    0.8491    0.0357    0.6557    0.9595
    0.9595    0.7922    0.9157    0.4218    0.9340    0.8491    0.0357    0.6557
    0.6557    0.9595    0.7922    0.9157    0.4218    0.9340    0.8491    0.0357
    0.0357    0.6557    0.9595    0.7922    0.9157    0.4218    0.9340    0.8491
    0.8491    0.0357    0.6557    0.9595    0.7922    0.9157    0.4218    0.9340
    0.9340    0.8491    0.0357    0.6557    0.9595    0.7922    0.9157    0.4218
```

```
>>
    0.9157    0.7922    0.9595    0.6557    0.0357    0.8491    0.9340    0.6787
    0.6787    0.9157    0.7922    0.9595    0.6557    0.0357    0.8491    0.9340
    0.9340    0.6787    0.9157    0.7922    0.9595    0.6557    0.0357    0.8491
    0.8491    0.9340    0.6787    0.9157    0.7922    0.9595    0.6557    0.0357
    0.0357    0.8491    0.9340    0.6787    0.9157    0.7922    0.9595    0.6557
    0.6557    0.0357    0.8491    0.9340    0.6787    0.9157    0.7922    0.9595
    0.9595    0.6557    0.0357    0.8491    0.9340    0.6787    0.9157    0.7922
    0.7922    0.9595    0.6557    0.0357    0.8491    0.9340    0.6787    0.9157
```

```
n=9;
rng(0);
c=rand(1,n);
cc=[c  c];
A=hankel(cc);
B=A(1:n,1:n);
disp(B);
    0.8147    0.9058    0.1270    0.9134    0.6324    0.0975    0.2785    0.5469    0.9575
    0.9058    0.1270    0.9134    0.6324    0.0975    0.2785    0.5469    0.9575    0.8147
    0.1270    0.9134    0.6324    0.0975    0.2785    0.5469    0.9575    0.8147    0.9058
    0.9134    0.6324    0.0975    0.2785    0.5469    0.9575    0.8147    0.9058    0.1270
    0.6324    0.0975    0.2785    0.5469    0.9575    0.8147    0.9058    0.1270    0.9134
    0.0975    0.2785    0.5469    0.9575    0.8147    0.9058    0.1270    0.9134    0.6324
    0.2785    0.5469    0.9575    0.8147    0.9058    0.1270    0.9134    0.6324    0.0975
    0.5469    0.9575    0.8147    0.9058    0.1270    0.9134    0.6324    0.0975    0.2785
    0.9575    0.8147    0.9058    0.1270    0.9134    0.6324    0.0975    0.2785    0.5469
```

```
n=5;
c=rand(1,n);
cc=fliplr(c);
ccc=circshift(cc,1);
A=toeplitz(c,ccc);
d=rand(1,n);
dd=[d  d];
D=hankel(dd);
B=D(1:n,1:n);
disp(A*B-B*A);
c=rand(1,n);
cc=fliplr(c);
ccc=circshift(cc,1);
C=toeplitz(c,ccc);
disp(A*C-C*A);
E=fliplr(C);
disp(B*E-E*B);
```

```
-0.0109   -0.0695    0.1286   -0.0830    0.0348
  -0.0695    0.1286   -0.0830    0.0348   -0.0109
   0.1286   -0.0830    0.0348   -0.0109   -0.0695
  -0.0830    0.0348   -0.0109   -0.0695    0.1286
   0.0348   -0.0109   -0.0695    0.1286   -0.0830
1.0e-15 *
   0.2220   -0.2220   -0.1110   -0.2220    0.2220
        0         0         0         0    0.2220
        0         0         0    0.2220         0
   0.1110         0   -0.2220   -0.2220    0.2220
   0.2220    0.1110         0   -0.2220         0
```

```
        0   -0.1356    0.2368   -0.2368    0.1356
   0.1356         0   -0.1356    0.2368   -0.2368
  -0.2368    0.1356         0   -0.1356    0.2368
   0.2368   -0.2368    0.1356         0   -0.1356
  -0.1356    0.2368   -0.2368    0.1356         0
```

(4) 循环矩阵的线性运算和乘积仍为循环矩阵.（封闭性）

(5) 若A,B同为（右或左）循环矩阵，**则AB为右循环矩阵.**（？？？？）

(6) 若A,B为循环矩阵，则

$$AB = BA$$ （交换律成立！！！）

(7) 若P为（右）循环移位矩阵，则

$$C_R(c_0,\dots,c_{n-1}) = c_0 I + c_1 P + c_2 P^2 + \cdots + c_{n-1} P^{n-1} = f(P)$$

$$P=\begin{bmatrix} 0 & 1 & 0 & \cdots & 0 \\ 0 & 0 & 1 & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \cdots & 1 \\ 1 & 0 & 0 & \cdots & 0 \end{bmatrix}$$

其中 $f(x) = c_0 + c_1 x + c_2 x^2 + \cdots + c_{n-1} x^{n-1}$

(8) $FC_R F^H = diag\,(f(\lambda_0), f(\lambda_1), \cdots, f(\lambda_{n-1}))$

其中 $\lambda_0, \lambda_1, \cdots, \lambda_{n-1}$ 为P的特征值.

```
n=5;A=eye(n);P=circshift(A,-1);
c=rand(1,n);C=zeros(n,n);
for ii=1:n
C=C+c(ii)*P^(ii-1);
end
disp(C);
disp(fliplr(C));
```

0.1299	0.5688	0.4694	0.0119	0.3371
0.3371	0.1299	0.5688	0.4694	0.0119
0.0119	0.3371	0.1299	0.5688	0.4694
0.4694	0.0119	0.3371	0.1299	0.5688
0.5688	0.4694	0.0119	0.3371	0.1299

0.3371	0.0119	0.4694	0.5688	0.1299
0.0119	0.4694	0.5688	0.1299	0.3371
0.4694	0.5688	0.1299	0.3371	0.0119
0.5688	0.1299	0.3371	0.0119	0.4694
0.1299	0.3371	0.0119	0.4694	0.5688

(9) 若A和B为循环矩阵，则 $A \pm B, AB$ 以及 A^{-1} 的特征值分别为

$$\lambda_k(A) \pm \lambda_k(B), \lambda_k(A)\lambda_k(B), \lambda_k^{-1}(A)(k = 0,1,\dots, n-1)$$

(10) 循环矩阵的逆仍为循环矩阵，且

$$C_R^{-1}(c_0, c_1, \dots, c_{n-1}) = C_R(b_0, b_1, \dots, b_{n-1})$$

其中

$$b_0 = \frac{1}{n}\sum_{k=0}^{n-1} \lambda_0^k f^{-1}(\lambda_k)$$

$$b_j = \frac{1}{n}\sum_{k=0}^{n-1} \lambda_{n-j}^k f^{-1}(\lambda_k), j = 1,2,\dots, n-1$$

2.2 特殊矩阵运算

■向量化函数与矩阵化函数

（列）向量化函数（按列拉直）

$$A = \begin{bmatrix} a_{11} & \cdots & a_{1n} \\ \vdots & \vdots & \vdots \\ a_{m1} & \cdots & a_{mn} \end{bmatrix}_{m\times n} \rightarrow vec(A) = \begin{bmatrix} a_{11} \\ \vdots \\ a_{m1} \\ \vdots \\ a_{1n} \\ \vdots \\ a_{mn} \end{bmatrix}$$

行向量化函数 $rvec(A) = [a_{11}, \cdots, a_{1n}, \cdots, a_{m1}, \cdots, a_{mn}]$

矩阵化函数

$$unvec_{m,n}(a) = A_{m\times n} = \begin{bmatrix} a_1 & a_{m+1} & \cdots & a_{m(n-1)+1} \\ a_2 & a_{m+2} & \cdots & a_{m(n-1)+2} \\ \vdots & \vdots & & \vdots \\ a_m & a_{2m} & \cdots & a_{mn} \end{bmatrix}$$

```
function test()
A=rand(3,3);
disp(A);
V=vec(A);
disp(V);
end

function V=vec(A)
s=size(A);
V=zeros(s(1)*s(2),1);
for ii=1:s(2)
    for jj=1:s(1)
        V((ii-1)*s(2)+jj)=A(jj,ii);
    end
end
end
```

```
>> test
    0.9448    0.3377    0.1112
    0.4909    0.9001    0.7803
    0.4893    0.3692    0.3897

    0.9448
    0.4909
    0.4893
    0.3377
    0.9001
    0.3692
    0.1112
    0.7803
    0.3897
```

◊矩阵化算子和向量化算子的关系

$$unvec_{m,n}(a) = A_{m\times n} \Rightarrow vec(A_{m\times n}) = a$$

◊向量化算子和行向量化算子的关系

$$rvec(A) = (vec(A^T))^T, rvec(A^T) = (rvec(A))^T$$

◊ $K_{mn} vec(A) = vec(A^T)$

其中 K_{mn} 为交换矩阵（commutation matrix)

$$K_{mn} = \sum_{j=1}^{n} (e_j^T \otimes I_m \otimes e_j)$$

■Kronecker积

◊ $m\times n$ 矩阵A和 $p\times q$ 矩阵B的**右Kronecker积**

$$A\otimes B=[a_{ij}B]=\begin{bmatrix} a_{11}B & a_{12}B & \cdots & a_{1n}B \\ a_{21}B & a_{22}B & \cdots & a_{2n}B \\ \vdots & \vdots & & \vdots \\ a_{m1}B & a_{m2}B & \cdots & a_{mn}B \end{bmatrix}_{mp\times nq}$$

◊ $m\times n$ 矩阵A和 $p\times q$ 矩阵的**左Kronecker积**

$$[A\otimes B]_{left}=[Ab_{ij}]=\begin{bmatrix} Ab_{11} & Ab_{12} & \cdots & Ab_{1q} \\ Ab_{21} & Ab_{22} & \cdots & Ab_{2q} \\ \vdots & \vdots & & \vdots \\ Ab_{p1} & Ab_{p2} & \cdots & Ab_{pq} \end{bmatrix}_{mp\times nq}$$

矩阵的Kronecker乘法及Matlab运用kron()计算Kronecker乘积.

```
A=rand(3,3)
B=ones(2,2)
C=kron(A,B)
```

```
A =
    0.4854    0.4218    0.9595
    0.8003    0.9157    0.6557
    0.1419    0.7922    0.0357
B =
     1     1
     1     1
C =
    0.4854    0.4854    0.4218    0.4218    0.9595    0.9595
    0.4854    0.4854    0.4218    0.4218    0.9595    0.9595
    0.8003    0.8003    0.9157    0.9157    0.6557    0.6557
    0.8003    0.8003    0.9157    0.9157    0.6557    0.6557
    0.1419    0.1419    0.7922    0.7922    0.0357    0.0357
    0.1419    0.1419    0.7922    0.7922    0.0357    0.0357
```

矩阵的Kronecker积:kron(A,B)

$$A \otimes B = [a_{ij}B] = \begin{bmatrix} a_{11}B & a_{12}B & \cdots & a_{1n}B \\ a_{21}B & a_{22}B & \cdots & a_{2n}B \\ \vdots & \vdots & & \vdots \\ a_{m1}B & a_{m2}B & \cdots & a_{mn}B \end{bmatrix}_{mp\times nq}$$

the Kronecker tensor product of X and Y.

Kronecker积的性质

（1）$A \otimes B \neq B \otimes A$

（2）$AB \otimes CD = (A \otimes C)(B \otimes D)$

（3）$A \otimes (B \pm C) = A \otimes B \pm A \otimes C$

$$(B \pm C) \otimes A = B \otimes A \pm C \otimes A$$

（4）$(A \otimes B)^{\dagger} = A^{\dagger} \otimes B^{\dagger}$

特别地，若A和B是可逆的正方矩阵，则

$$(A \otimes B)^{-1} = A^{-1} \otimes B^{-1}$$

(5) $(A \otimes B)^T = A^T \otimes B^T$; $(A \otimes B)^H = A^H \otimes B^H$

(6) $rank(A \otimes B) = rank(A) rank(B)$

(7) 对于 $A_{m\times m}, B_{n\times n}$，有 $\det(A \otimes B) = (\det(A))^m (\det(B))^n$

(8) $tr(A \otimes B) = tr(A) tr(B)$

(9) 对于矩阵 $A_{m\times n}, B_{m\times n}, C_{p\times q}, D_{p\times q}$，有

$$(A+B)\otimes(C+D) = A\otimes C + A\otimes D + B\otimes C + B\otimes D$$

更一般地 $$\left[\sum_{i=1}^{M} A(i)\right] \otimes \left[\sum_{j=1}^{N} B(j)\right] = \sum_{i=1}^{M}\sum_{j=1}^{N}[A(i)\otimes B(j)]$$

(10) 对于矩阵 $A_{m\times n}, B_{p\times q}, C_{n\times r}, D_{q\times s}$，有

$$(A \otimes B)(C \otimes D) = AC \otimes BD$$

更一般地

$$\prod_{i=1}^{N}[A(i)\otimes B(i)] = \left[\prod_{i=1}^{N} A(i)\right] \otimes \left[\prod_{i=1}^{N} B(i)\right]$$

定理 令 $A_{m\times p}, B_{p\times q}, C_{q\times n}$, 则

$$vec(ABC) = (C^T \otimes A)vec(B)$$

特别地，当 $A = I_m$，而 $B \in R^{m\times q}, C \in R^{q\times n}$ 时，有

$$vec(BC) = (C^T \otimes I_m)vec(B)$$

$$= (C^T \otimes B)vec(I_q)$$

$$= (I_n \otimes B)vec(C)$$

应用1： 矩阵方程 $AXB = D$ 的求解.

应用2：矩阵方程 $AX + XB = C$的求解.

矩阵方程 $AXB = D$ 的求解.

$$AXB = D$$
$$vec(AXB) = vec(D)$$
$$(B^T \otimes A)vec(X) = vec(D)$$
$$vec(X) = (B^T \otimes A) \backslash vec(D)$$
$$X = unvec((B^T \otimes A) \backslash vec(D))$$

```
n=5;
A=rand(n,n);
X=rand(n,n);
B=rand(n,n);
D=A*X*B;
vD=vec(D);
H=kron(B',A);
vX=H\vD;
XX=rvec(vX,n,n);
disp(A*XX*B-D);
function V=vec(A) %向量化
  V=reshape(A,[],1);
end
function V=unvec(A,m,n)%反向量化
  V=reshape(A,m,n);
end
```

矩阵方程 $AX + XB = D$ 的求解.

```
n=5;
I=eye(n);
A=rand(n,n);
X=rand(n,n);
B=rand(n,n);
D=A*X+X*B;
vD=vec(D);
H=kron(I',A)+kron(B',I);
vX=H\vD;
XX=uunvec(vX,n,n);
disp(A*XX+XX*B-D);
function V=vec(A) %向量化
  V=reshape(A,[],1);
end
function V=unvec(A,m,n)%反向量化
  V=reshape(A,m,n);
end
```

```
1.0e-14 *
        0    0.1776    0.0888    0.0888    0.0444
  -0.0444         0    0.0444   -0.0444   -0.0888
        0         0   -0.0444         0   -0.0222
   0.0444    0.0444    0.0444         0    0.0222
  -0.0444         0         0   -0.0444         0
```

广义Kronecker积

给定N个$m \times r$ 矩阵 $A_i, i = 1,2,\ldots, N$,它们组成矩阵组 $\{A\}_N$.该矩阵组与 $N \times l$ 矩阵B的Kronecker积称为**广义Kronecker积**，定义为

$$\{A\}_N \otimes B = \begin{bmatrix} A_1 \otimes b_1 \\ A_2 \otimes b_2 \\ \vdots \\ A_N \otimes b_N \end{bmatrix}$$

式中, b_i 是矩阵的第i个行向量.

广义Kronecker积在滤波器组的分析、Haar变换和Hadamard变换的快速算法的推导中有着重要的作用.

■Khatri-Rao积

定义： $m \times n$ 矩阵A与$m \times n$矩阵B的**Khatri-Rao积**记为$A*B$

并定义为

$$A*B=[a_1\otimes b_1, a_2\otimes b_2,\ldots,a_n\otimes b_n]_{m^2\times n}$$

其中 a_i和b_i 分别为矩阵A与矩阵B的第i列.

性质：

（1）结合律 $A*(B*C)=(A*B)*C$

（2）交换律 $A*B=K_{nn}(B*A)$

$$K_{mn}=\sum_{j=1}^{n}(e_j^T\otimes I_m\otimes e_j)$$

其中K_{nn}为交换矩阵

（3）Kronecker积与Khatri-Rao积的乘积

$$(A\otimes B)(C*D)=AC*BD$$

■Hadamard积

定义：$m \times n$ 矩阵$A=[a_{ij}]$ 与 $m \times n$ 矩阵 $B=[b_{ij}]$ 的**Hadamard积**记作 A ⊙ B，并定义为

$$A \odot B = [a_{ij} b_{ij}]_{m \times n}$$

◊Hadamard积也称为**Schur积**或**对应元素乘积(elementwise product)**.

性质：

(1) 若A，B均为 $m \times n$ 矩阵，则

$$A \odot B = B \odot A$$

$$(A \odot B)^T = A^T \odot B^T$$

$$(A \odot B)^H = A^H \odot B^H$$

$$(A \odot B)^* = A^* \odot B^*$$

(2) 若 $A \in C^{m\times n}$，则 $A \odot O_{m\times n} = O_{m\times n} \odot A = O_{m\times n}$

(3) 若c为常数，则 $c(A \odot B) = (cA) \odot B = A \odot (cB)$

(4) $\mathrm{A}\odot I_{\mathrm{m}}=I_{\mathrm{m}}\odot \mathrm{A}=diag(A)=diag(a_{11},a_{22},\ldots,a_{mm})$

(5) 若A，B，C，D均为 $m\times n$ 矩阵，则

$$\mathrm{A}\odot(BC)=(A\odot B)\odot \mathrm{C}=A\odot B\odot C$$

$$(A\pm B)\odot C=A\odot C\pm B\odot C$$

$$(A+B)\odot(C+D)=A\odot C+A\odot D+B\odot C+B\odot D$$

(6) 若A,C为 $m\times m$ 矩阵，B，D为 $n\times n$ 矩阵，则

$$(A\oplus B)\odot(C\oplus D)=(A\odot C)\oplus(B\odot D)$$

(7) 若A，B，C为 $m\times n$ 矩阵，则

$$tr(A^T(\mathrm{B}\oplus\mathrm{C}))=tr((A^T\odot B^T)C)$$

(8) 若 A，B，D为 $m\times m$ 矩阵，则

D为对角矩阵 $\Rightarrow (DA)\odot(BD)=D(A\odot B)D$

(9) 若 $m\times m$ 矩阵A,B是(半)正定的，则A ⊙ B也是(半)正定的.

■矩阵直和

定义：$m \times m$ 矩阵与 $n \times n$ 矩阵的**直和**定义为

$$A \oplus B = \begin{bmatrix} A & O_{m\times n} \\ O_{n\times m} & B \end{bmatrix}$$

性质：

（1）若c为常数，则 $c(A \oplus B) = cA \oplus cB$.

（2）一般情况下，$A \oplus B \neq B \oplus A$

（3）

$$(A \oplus B)^* = A^* \oplus B^*$$

$$(A \oplus B)^T = A^T \oplus B^T$$

$$(A \oplus B)^H = A^H \oplus B^H$$

$$(A \oplus B)^{-1} = A^{-1} \oplus B^{-1}$$

Block diagonal concatenation of matrix input arguments.

矩阵的直和: blkdiag(A,B)

$$A \oplus B = \begin{bmatrix} A & O_{m\times n} \\ O_{n\times m} & B \end{bmatrix}$$

(4) $A \oplus (B \oplus C) = (A \oplus B) \oplus C = A \oplus B \oplus C$

(5) $(A \pm B) \oplus (C \pm D) = (A \oplus C) \pm (B \oplus D)$

$(A \oplus C)(B \oplus D) = AB \oplus CD$

(6)

$$\det(\oplus_{i=1}^{N} A_i) = \prod_{i=1}^{N} \det(A_i)$$

$$tr(\oplus_{i=1}^{N} A_i) = \sum_{i=1}^{N} tr(A_i)$$

$$rank(\oplus_{i=1}^{N} A_i) = \sum_{i=1}^{N} rank(A_i)$$

（7）若A,B分别为$m \times m$ 和$n \times n$ 正交矩阵，则 $A \oplus B$ 为 $(m+n)\times(m+n)$ 正交矩阵.

数据实验(一)

1.编程验证Kronecker积与Khatri-Rao积的乘积的关系 $(A\otimes B)(C*D)=AC*BD$

2.编写程序，求矩阵方程 $AX+XB=C$

3.编程实现rvec

4.编写程序，求一组数据的多项式**最小二乘拟合，并估算其精度**

提交截止日期：4月16日

作业2

(1) 2.11，2.12，2.13

注：可以用演绎法证明，也可以用数据实验验证（请自我设计验证方案）

(2)编写求**左循环矩阵的程序**

(3)请写出矩阵方程$AXB + CXD = F$ 的形式解**（用Kronecker积和vec,rvec函数表示）**

提交截止日期：4月16日