MATLAB中的向量和矩阵运算

# 向量与矩阵

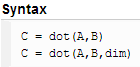
向量和矩阵在内存上都是以数组的方式存储的，从数组结构的角度讲，向量是一维数组；矩阵是二维数组。

MATLAB语言是专门用于矩阵运算的语言，其最基本、最重要的功能就是进行矩阵运算。

# 向量的运算

## 向量的点乘(数量积)（又称内积）

利用dot函数来计算内积。

 **dim 对某一个维度的向量进行求内积。**

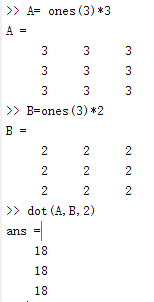
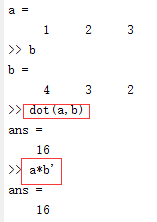
两个向量的模和两个向量之间的夹角余弦三者的乘积。



在二维三维下，有



**只适用于二维、三维空间**。



## 向量的叉乘（向量积、外积）

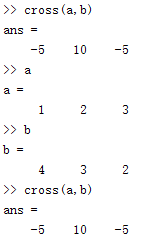
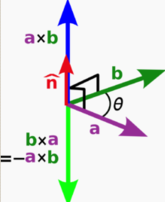
向量积的大小：



向量积|c|=|a×b|=|a| |b|sin<a,b>，即c的长度在数值上等于以a，b，夹角为θ组成的平行四边形的面积。

**向量积的方向**：a向量与b向量的向量积的方向与这两个向量所在平面垂直，且**遵守右手定则**。

c的方向垂直于a与b所决定的平面，c的指向按右手定则从a转向b来确定。

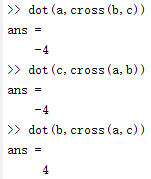


## 向量的混合积

几何意义：混合积的绝对值表示以3个向量为棱的平行六面体的体积，符号由右手定则确定。

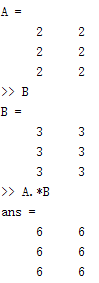
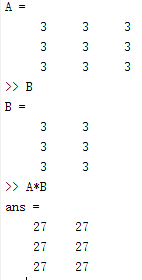
**向量积的模值的意义是以两个向量为边的平行四边形的面积**。

**混合积： dot(a,cross(b,c))、 dot（b,cross(a,b)）、dot(c,cross(a,b))**

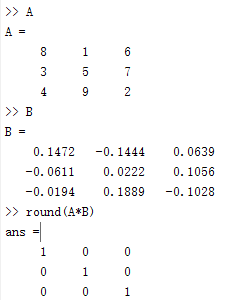


# 矩阵的基本运算

## 矩阵乘法与矩阵点乘



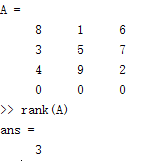
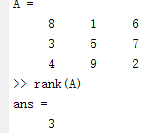
## 矩阵求逆：逆矩阵inv(A)



## 矩阵的秩：rank(A)

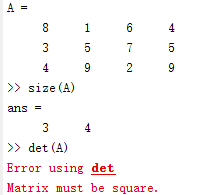
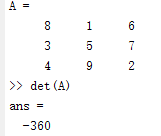
矩阵的秩反映了矩阵各行向量或各列向量之间的线性相关程度。

行满秩或列满秩矩阵，其各行向量或各列向量之间都是线性无关的。



## 矩阵的行列式:det(A)

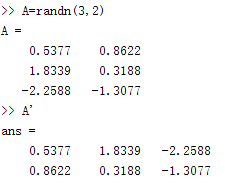
**矩阵A必须是方阵**，才具有行列式。矩阵的行列式是一个数值，用来表明该矩阵是否奇异。

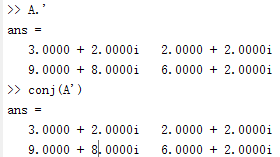
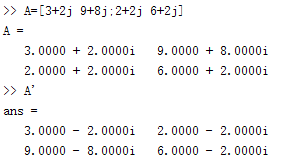


## 矩阵转置

利用 ‘ 实现转置操作。

对于实数矩阵，’ 表示矩阵转置；对于复数矩阵，就是共轭转置。如果非共轭转置，可以使用 **.’ 也可以使用conj()再取个共轭。**





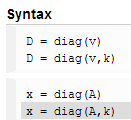
## 特殊矩阵的生成

魔术矩阵magic(N)、全1矩阵ones、全0矩阵zeros、eye单位矩阵

**rand 服从均匀分布的随机矩阵。**

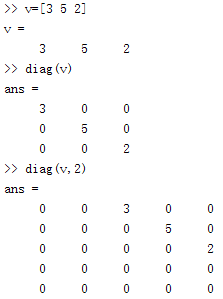
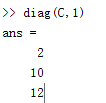
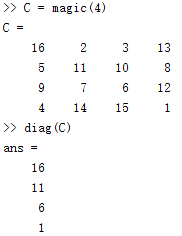
**randn 服从正态分布的随机矩阵**。

### 对角矩阵：



1. **V表示向量，k为以个整数**，表示矩阵的第k个对角线为V。默认k为0，即主对角线，k大于0表示主对角线的上方，k小于0表示主对角线的下方。

②　**A 为一个矩阵，x=diag（A，k）**表示取出矩阵A的第k个对角线的值组成一个向量，默认k=0。

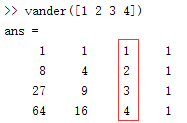


### compan 生成伴随矩阵。



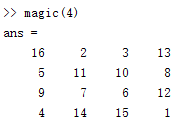
### vander 范德蒙矩阵





### 魔术矩阵magic

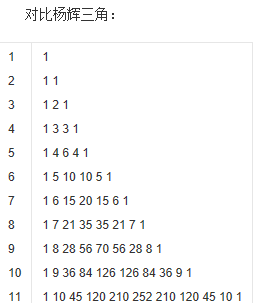
每行、每列及对角线之和相同。 **magic(N) N>2**



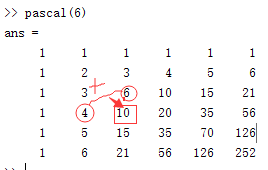
### Pascal矩阵：帕斯卡矩阵

**帕斯卡矩阵：**Pascal矩阵的第一行元素和第一列元素都为1，其余位置处的元素是**该元素的左边元素加起上一行对应位置相加而得**，如元素**Ai,j=Ai,j-1+Ai-1,j**。Ai,j表示第i行，第j列位置上的元素。**帕斯卡矩阵一定是正定矩阵**，可以Cholesky分解。

**对比杨辉三角形表，杨辉三角形表**是二项式 (x+y)^n 展开后的系数随自然数 n 的增大组成的一个三角形表。

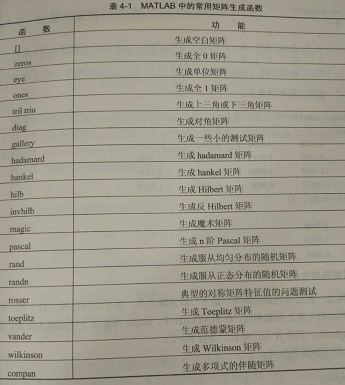


帕斯卡pascal矩阵：



其他的一些函数生成特殊矩阵。

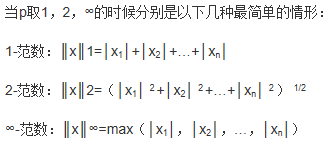
如tril triu 上三角或下三角 、compan 伴随矩阵、vander 范德蒙矩阵



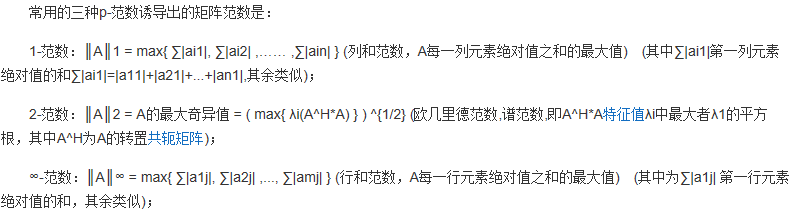
## 向量与矩阵的范数

一般都是求１、２、无穷级范数。

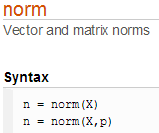
向量范数：

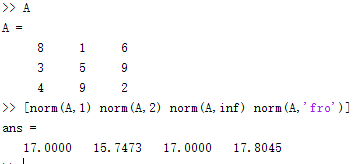


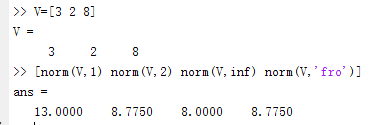
矩阵范数：



利用函数norm 函数求解。P通常取1、2 、inf 和‘fro’；‘fro’ 为求解矩阵A的Frobenius范数。**p值默认为2**。





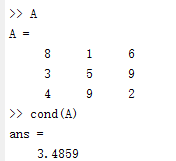


## 矩阵的条件数

矩阵的条件数：在矩阵的逆和矩阵范数的基础上定义的，是用于衡量矩阵病态程度的关键量。一个矩阵的条件数越大，表明该矩阵的病态程度越严重。

利用**cond函数**求解矩阵的条件数。

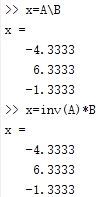
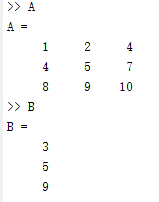
对于秩为零或接近零的奇异矩阵，其条件数会比较大。



# 矩阵与线性代数

## 求解线性方程组

Ax=B 求x x=inv(A)\*B = A\B



利用左除符号和右除符号，求解线性方程组，避免了矩阵求逆操作。

如果系数矩阵A的维度为mXn：

m = n，此方程组为方阵系统，matlab给出精确解；

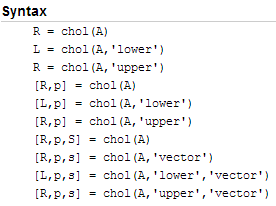
# 矩阵分解

## Cholesky分解

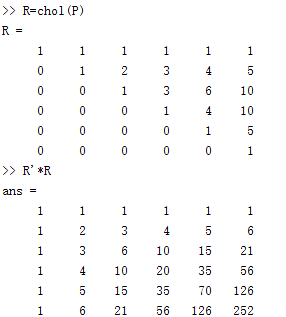
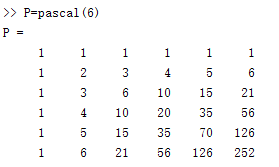
Cholesky分解将对称矩阵表示为一个三角矩阵与其转置的乘积形式，即**A=R’R**，其中A为对称矩阵，R为上三角矩阵。

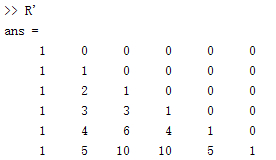
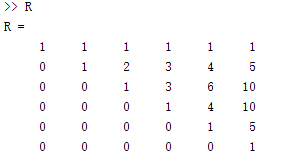
并非所有的对称矩阵都能进行Cholesky分解，只有正定矩阵能够进行Cholesky分解，如**Pascal矩阵**。

在MATLAB中，Cholesky分解由函数chol实现**，chol(A)** 对矩阵Ａ进行Cholesky分解，返回其对应的上三角矩阵。



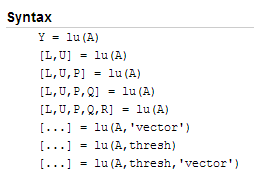
**Pascal矩阵**：





## LU分解

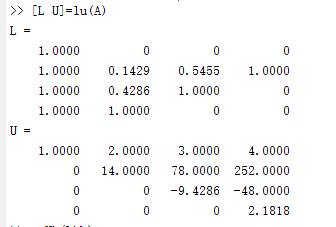
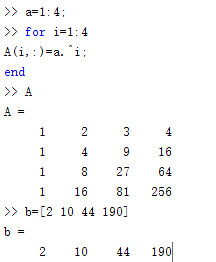
矩阵的LU分解将一个方阵表示为一个下三角置换矩阵和一个上三角矩阵乘积的形式。如A=LU。其中L为下三角置换矩阵，U为上三角矩阵。MATLAB中通过lu函数可以实现矩阵的LU分解。

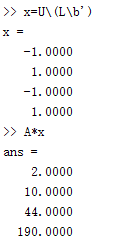


**矩阵的LU分解可以实现线性方程组的快速求解**。

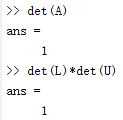
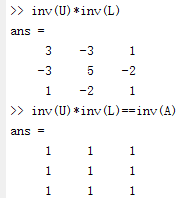
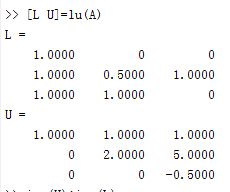
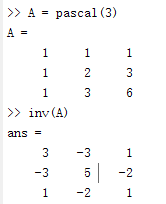
**矩阵的LU分解可以用来快速求解矩阵的A的逆矩阵和其行列式**，即det(A)=det(L)\*det(U) 和 inv(A)=inv(U)\*inv(L) 注：（AB）’=B’A’

Ax=b





求逆：



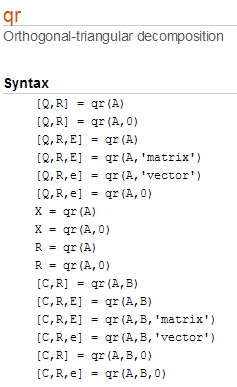
结果都是一致的，但是利用LU分解法可以节省计算时间。

## QR分解（正交分解）

如果ＱＱ＇＝Ｉ,则矩阵Q为正交矩阵。

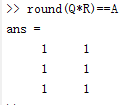
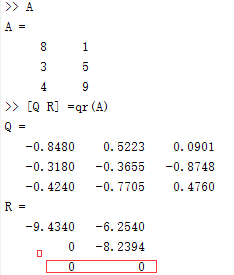
正交分解： 将矩阵表示为正交矩阵（或酉矩阵）和上三角矩阵的乘积。

如A=QR或AP=QR。其中Q为正交矩阵或酉矩阵，R为上三角矩阵，P为转置矩阵。

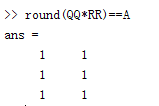
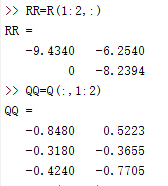
正交分解有4中形式：包括完全分解、简化分解、带置换矩阵的分解和不带置换矩阵的分解。  


### 完全分解

一矩阵AmXn，其中m>n，完全正交分解产生一个mXm的正交矩阵和一个mXn的上三角矩阵R，满足A=QR。 利用函数qr实现矩阵的完全分解。

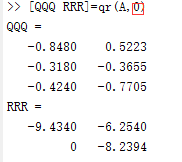


在QR分解中，Q的最后m-n列及R的最后m-n行都没有意思，可以省略，省略后就是简化正交分解。



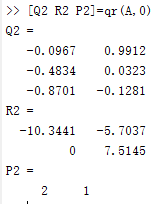
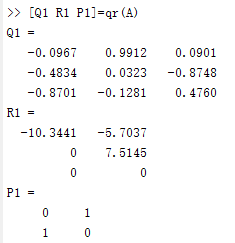
### 简化分解

简化正交分解可以节省存储空间和运算时间。正交分解通过在qr函数中添加0参数即可。



对于[Q R P]=qr(C) : 包含置换的正交分解。

如果对矩阵进行置换，可以避免由于矩阵奇异造成的误差。选择置换后，在分解的每一步，选择剩下列中范数最大的一列作为分解的基。

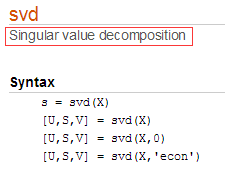


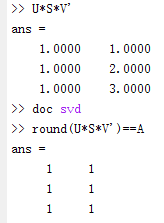
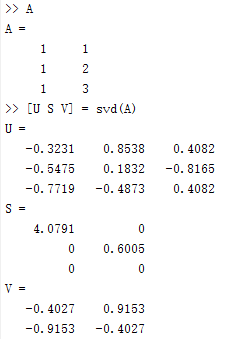
## 矩阵的奇异值分解

**矩阵的奇异值分解**：

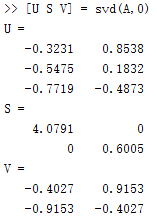
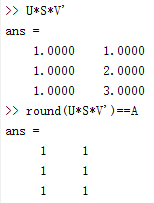
对于矩阵A，如果存在数σ和向量u、v，满足**Av=σu和ATu=σv**，则称σ为A的奇异值，u、v为A的奇异向量。

如果将矩阵的奇异值写为对角矩阵的格式，记为S（不足的部分记为0）；以奇异向量为列并扩充为正交矩阵U和V，则有**AV=US和ATU=VS**。U和V为正交矩阵，则得到**A=USVT**，即为矩阵A的奇异值分解。





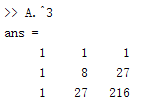
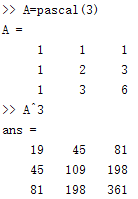
简化奇异值分解，与简化正交分解一样，输入参数中以0标志。

# 矩阵指数函数和幂函数

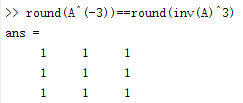
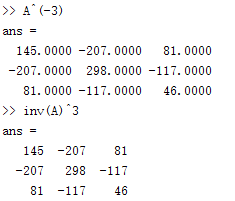
## 矩阵的正整数幂

如果A为方阵，p为正整数，则A^p表示p个A相乘。

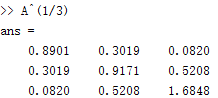
点对点运算

## 矩阵的负数幂与分数幂

如果A为非奇异方阵，则A^(-p)等价于inv(A)^p。



矩阵的分数幂运算，其结果依赖于矩阵特征值的分布情况。

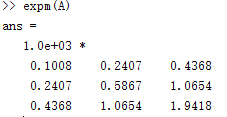
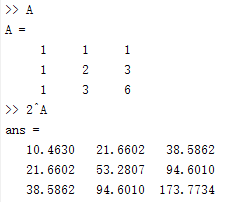


## 矩阵指数运算

矩阵作为指数。

线性系统dx/dt = Ax ，则解为**x(t)=etAx(0)。**

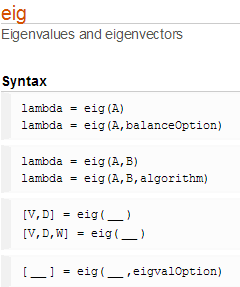
**eA可以通过函数expm(A)完成。其他的，如2A 直接写即可。**



# 矩阵特征值、特征向量及特征多项式

对于矩阵Ａ，如果存在向量ｘ和数值λ，使得Ax=λx，则ｘ称为矩阵Ａ的特征向量，λ为矩阵Ａ的特征值。｜Ａ－λＥ｜为特征多项式。

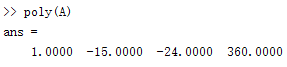
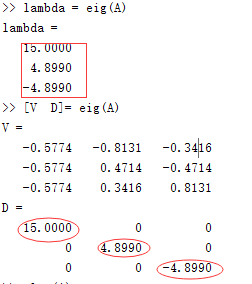
利用eig函数获取矩阵A的特征值和特征向量，利用ploy获取矩阵A的特征多项式的系数。



**lambda=eig(A)** 只是获取矩阵A的特征值；

**[V D]=eig(A)** 其中矩阵V是以矩阵A的特征向量为列的矩阵，矩阵D是以矩阵A的特征值为对角元素的矩阵。

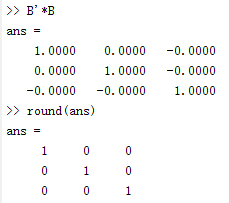
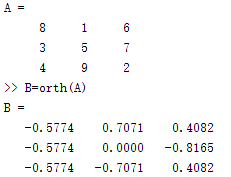
**利用poly函数获取矩阵A的特征多项式，特征多项式的根就是矩阵的特征值。**



# 矩阵的标准正交基

MATLAB中通过orth函数获得矩阵A的一组标准正交基，其语法为B = orth（A）。

B的列向量就是矩阵A的标准正交基，满足**BTB=eye(rank(A))。**



# 稀疏型矩阵

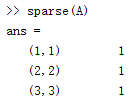
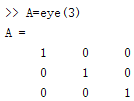
**稀疏型矩阵： 只有少数非0元素的矩阵。**

如果对稀疏型矩阵的全部元素进行存储和计算，就会导致时间和空间上的极大浪费，因此，为了有效地存储和处理稀疏型矩阵，MATLAB采用了一些优化技术：MATLAB中只存储稀疏矩阵中的非0元素，并用行索引和列索引表明每个非0元素在原矩阵中的位置。同样，MATLAB中采用了一些专门的算法来处理稀疏矩阵，以避免0元素的运算，并且最大限度地减少中间结果中的非0元素。

## 稀疏型矩阵的生成

**MATLAB不会自动生成稀疏型矩阵**。

利用sparse函数，可以将一个满矩阵装换成稀疏型矩阵。



利用特殊函数生成稀疏型矩阵。

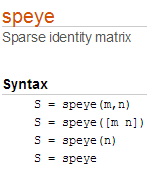
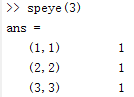
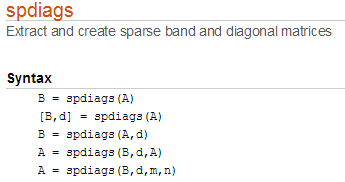
如 speye 生成单位稀疏矩阵

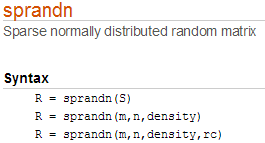
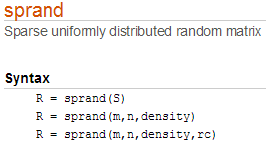
**sprand 均匀分布的随机稀疏矩阵**

**sprandn 正态分布的随机稀疏矩阵**

sprandsym 对称随机稀疏矩阵

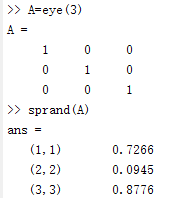
spdiags 对角稀疏矩阵

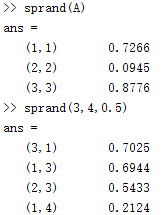


**sprand(n) 不能生成n阶稀疏矩阵**。

### Sprand（S） 生成一个与稀疏矩阵S结构完全相同的稀疏矩阵，矩阵元素服从均匀分布。



### Sprand(m,n,density) 生成mXn阶稀疏矩阵，矩阵非0元素个数大约为m\*n\*density。



### Sprand（m,n,density,rc）: 与②相似。Rc有新的意义。查文档。

sprandn与sprand用法相同。只不过，sprand服从均匀分布，sprandn服从正态分布。

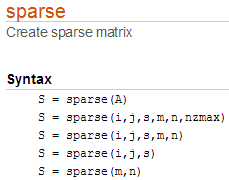
## 稀疏矩阵与满矩阵之间的相互转化

Sparse 满矩阵转化为稀疏矩阵

Full 稀疏矩阵转化为 满矩阵

Find 查找非0 元素的索引

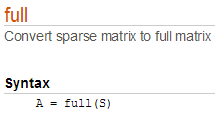
Spconvert 导入稀疏矩阵

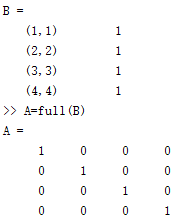
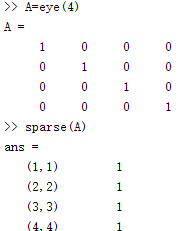


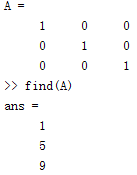
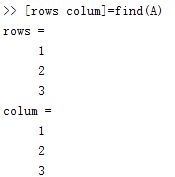
Sparse（m，n）生成一个所有元素都为0的稀疏矩阵。



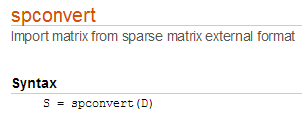
**sparse（A） 转化**。 full（B）。





**spconvert**



## 稀疏矩阵的操作

MATLAB中的大部分函数都可以用于稀疏矩阵，功能和调用格式与满矩阵相同。

下面介绍一些用于系数矩阵的函数：

nnz 返回矩阵非零元素个数

nonzeros 返回矩阵中非零元素构成的向量，以矩阵的列为序

spones 将矩阵所有非零元素置为1

spalloc 为稀疏矩阵分配内存空间

**issparse 判断是否为稀疏矩阵**

spfun 对稀疏矩阵的非零元素进行操作

spy 稀疏矩阵的图形表示

