[numpy linalg模块](http://www.cnblogs.com/xieshengsen/p/6836430.html)

# 线性代数  
# numpy.linalg模块包含线性代数的函数。使用这个模块，可以计算逆矩阵、求特征值、解线性方程组以及求解行列式等。

import numpy as np

# 1. 计算逆矩阵  
# 创建矩阵  
A = np.mat("0 1 2;1 0 3;4 -3 8")  
print (A)  
#[[ 0 1 2]  
# [ 1 0 3]  
# [ 4 -3 8]]

# 使用inv函数计算逆矩阵  
inv = np.linalg.inv(A)  
print (inv)  
#[[-4.5 7. -1.5]  
# [-2. 4. -1. ]  
# [ 1.5 -2. 0.5]]

# 检查原矩阵和求得的逆矩阵相乘的结果为单位矩阵  
print (A \* inv)  
#[[ 1. 0. 0.]  
# [ 0. 1. 0.]  
# [ 0. 0. 1.]]

# 注：矩阵必须是方阵且可逆，否则会抛出LinAlgError异常。

# 2. 求解线性方程组  
# numpy.linalg中的函数solve可以求解形如 Ax = b 的线性方程组，其中 A 为矩阵，b 为一维或二维的数组，x 是未知变量

import numpy as np

#创建矩阵和数组  
B = np.mat("1 -2 1;0 2 -8;-4 5 9")  
b = np.array([0,8,-9])

# 调用solve函数求解线性方程  
x = np.linalg.solve(B,b)  
print (x)  
#[ 29. 16. 3.]

# 使用dot函数检查求得的解是否正确  
print (np.dot(B , x))  
# [[ 0. 8. -9.]]

# 3. 特征值和特征向量  
# 特征值（eigenvalue）即方程 Ax = ax 的根，是一个标量。其中，A 是一个二维矩阵，x 是一个一维向量。特征向量（eigenvector）是关于特征值的向量  
# numpy.linalg模块中，eigvals函数可以计算矩阵的特征值，而eig函数可以返回一个包含特征值和对应的特征向量的元组

import numpy as np

# 创建一个矩阵  
C = np.mat("3 -2;1 0")

# 调用eigvals函数求解特征值  
c0 = np.linalg.eigvals(C)  
print (c0)  
# [ 2. 1.]

# 使用eig函数求解特征值和特征向量 (该函数将返回一个元组，按列排放着特征值和对应的特征向量，其中第一列为特征值，第二列为特征向量)  
c1,c2 = np.linalg.eig(C)  
print (c1)  
# [ 2. 1.]   
print (c2)  
#[[ 0.89442719 0.70710678]  
# [ 0.4472136 0.70710678]]

# 使用dot函数验证求得的解是否正确  
for i in range(len(c1)):  
print ("left:",np.dot(C,c2[:,i]))  
print ("right:",c1[i] \* c2[:,i])  
#left: [[ 1.78885438]  
# [ 0.89442719]]  
#right: [[ 1.78885438]  
# [ 0.89442719]]  
#left: [[ 0.70710678]  
# [ 0.70710678]]  
#right: [[ 0.70710678]  
# [ 0.70710678]]

# 4.奇异值分解  
# SVD（Singular Value Decomposition，奇异值分解）是一种因子分解运算，将一个矩阵分解为3个矩阵的乘积  
# numpy.linalg模块中的svd函数可以对矩阵进行奇异值分解。该函数返回3个矩阵——U、Sigma和V，其中U和V是正交矩阵，Sigma包含输入矩阵的奇异值。

import numpy as np

# 分解矩阵  
D = np.mat("4 11 14;8 7 -2")  
# 使用svd函数分解矩阵  
U,Sigma,V = np.linalg.svd(D,full\_matrices=False)  
print ("U:",U)  
#U: [[-0.9486833 -0.31622777]  
# [-0.31622777 0.9486833 ]]  
print ("Sigma:",Sigma)  
#Sigma: [ 18.97366596 9.48683298]  
print ("V",V)  
#V [[-0.33333333 -0.66666667 -0.66666667]  
# [ 0.66666667 0.33333333 -0.66666667]]  
# 结果包含等式中左右两端的两个正交矩阵U和V，以及中间的奇异值矩阵Sigma

# 使用diag函数生成完整的奇异值矩阵。将分解出的3个矩阵相乘  
print (U \* np.diag(Sigma) \* V)  
#[[ 4. 11. 14.]  
# [ 8. 7. -2.]]

# 5. 广义逆矩阵  
# 使用numpy.linalg模块中的pinv函数进行求解,  
# 注：inv函数只接受方阵作为输入矩阵，而pinv函数则没有这个限制

import numpy as np

# 创建一个矩阵  
E = np.mat("4 11 14;8 7 -2")  
# 使用pinv函数计算广义逆矩阵  
pseudoinv = np.linalg.pinv(E)  
print (pseudoinv)  
#[[-0.00555556 0.07222222]  
# [ 0.02222222 0.04444444]  
# [ 0.05555556 -0.05555556]]

# 将原矩阵和得到的广义逆矩阵相乘  
print (E \* pseudoinv)  
#[[ 1.00000000e+00 -5.55111512e-16]  
# [ 0.00000000e+00 1.00000000e+00]]

# 6. 行列式  
# numpy.linalg模块中的det函数可以计算矩阵的行列式

import numpy as np

# 计算矩阵的行列式  
F = np.mat("3 4;5 6")  
# 使用det函数计算行列式  
print (np.linalg.det(F))  
# -2.0