高斯列主元消元法计算行列式

PB16030899 - 朱河勤
2018-5-20

> 书上 p186, 附录 1 程序 11

算法描述

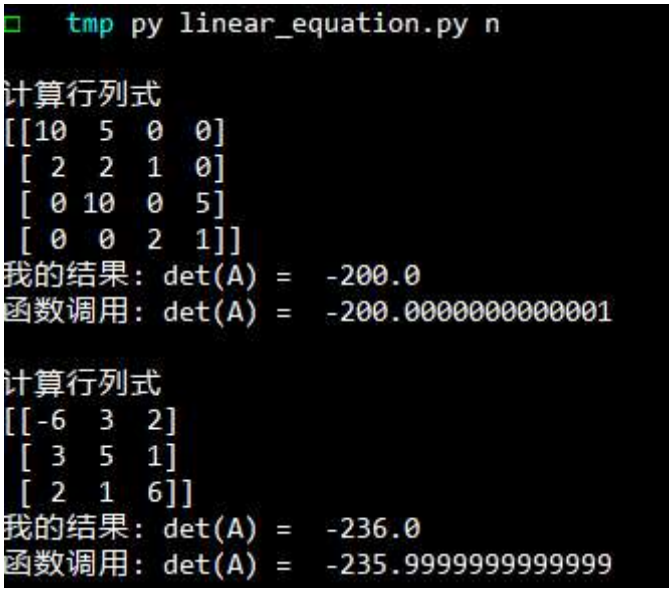
行序从 1 到 m, 当为第 i 行, 找出第 i 列第 i 个数到第 m 个数中最大值的行 k, 然后交换行 i,k, 然后依次用第 i 行, 将 i+1..m 行的第 i 个元素化作 0, 最后得到上三角矩阵, 对角线元素相乘就得到行列式

程序源码

```
import numpy as np
def gauss_prior_elimination(A,b=None):
    '''using guass elimination,get up_trianglge form of A'''
    m,n = A.shape
    res = [0]*m if b is None else b
    if m!=n:raise Exception("[Error]: matrix is not inversable")
    # necessary,otherwise when the dtype of A is int, then it will be wrong
    B = np.matrix(A,dtype=float)
    for  i in range(m-1):
        col = abs(B[i:,i]) #  note using abs value,  return a matrix in (m-i)x1 form
        mx = col.max()
        if mx==0: raise Exception("[Error]: matrix is not inversable")
        pos = i+col.argmax()
        if pos != i :  B[[pos,i],:] = B[[i,pos],:]  #  note how to swap cols/rows
        #B[i,:] = B[i,:]/mx
        #res[i]/=mx
        for j in range(i+1,m):
            if B[j,i]!=0:
                B[j,:] -= B[j,i]/B[i,i] * B[i,:]
                res[j] -= res[i]/B[i,i]*res[j]
    if b is None:return B
    return B,res

def det(A):
    m,n = A.shape
    ret=1
    B = gauss_prior_elimination(A)
    print(B)
    for i in range(m):ret *= B[i,i]
    return ret
```

测试结果


tmp py linear_equation.py n
计算行列式
[[10 5 0 0]
[2 2 1 0]
[0 10 0 5]
[0 0 2 1]]
我的结果: det(A) = -200.0
函数调用: det(A) = -200.0000000000001
计算行列式
[[-6 3 2]
[3 5 1]
[2 1 6]]
我的结果: det(A) = -236.0
函数调用: det(A) = -235.9999999999999