```
import matplotlib.pyplot as plt
import scipy
from scipy.special import pro_cv

def fctl(deuxk,m,n):
    if deuxk == 0:
        return(n*(n+1))
    elif deuxk == 2:
        return(0.5*(1-(2*m-1)*(2*m+1)/((2*n-1)*(2*n+3))))
    elif deuxk == 4:
        return(-(n-m+1)*(n-m+2)*(n+m+1)*(n+m+2)/(2*(2*n+1)*((2*n+3)**3)*(2*n+5))
+((n-m-1)*(n-m)*(n+m-1)*(n+m)/(2*(2*n-3)*((2*n-1)**3)*(2*n+1))))
    elif deuxk == 6:

return((4*m*m-1)*((n-m+1)*(n-m+2)*(n+m+1)*(n+m+2)/((2*n-1)*(2*n+1)*((2*n+3)**6)*
(2*n+5)*(2*n+7))-((n-m-1)*(n-m)*(n+m-1)*(n+m)/
((2*n-5)*(2*n-3)*((2*n-1)**5)*(2*n+1)*(2*n+3)))))
    elif deuxk == 8:
        A
=((n-m-1)*(n-m)*(n+m-1)*(n+m)/(((2*n-5)**2)*(2*n-3)*((2*n-1)**7)*(2*n+1)*((2*n+3
)**2)))-((n-m+1)*(n-m+2)*(n+m+1)*(n+m+2)/
(((2*n-1)**2)*(2*n+1)*((2*n+3)**7)*(2*n+5)*((2*n+7)**2)))
        B =
((n-m-3)*(n-m-2)*(n-m-1)*(n-m)*(n+m-3)*(n+m-2)*(n+m-1)*(n+m)/((2*n-7)*((2*n-5)**
2)*((2*n-3)**3)*((2*n-1)**4)*(2*n+1)))-((n-m+1)*(n-m+2)*(n-m+3)*(n-
m+4)*(n+m+1)*(n+m+2)*(n+m+3)*(n+m+4)/
((2*n+1)*((2*n+3)**4)*((2*n+5)**3)*((2*n+7)**2)*(2*n+9)))
        C =
(((n-m+1)**2)*((n-m+2)**2)*((n+m+1)**2)*((n+m+2)**2)/(((2*n+1)**2)*((2*n+3)**7)*
((2*n+5)**2)))-(((n-m-1)**2)*((n-m)**2)*((n+m-1)**2)*((n+m)**2))/
(((2*n-3)**2)*((2*n-1)**7)*((2*n+1)**2))
        D =
((n-m-1)*(n-m)*(n-m+1)*(n-m+2)*(n+m-1)*(n+m)*(n+m+1)*(n+m+2))/((2*n-3)*((2*n-1)*
*4)*((2*n+1)**2)*((2*n+3)**4)*(2*n+5))
        return(2*((4*m*m-1)**2)*A+(B/16)+(C/8)+(D/2))

def lmb(m,n,c):
    somme = 0
    for deuxk in [0,2,4,6,8]:
        somme = somme+fctl(deuxk,m,n)*(c**deuxk)
    return(somme)

for c in [5.5,6.5,7.5,8.5,9.5,10.5]:
    combien = 2*round(c/2)+1
    for n in range(1,combien+1):
        print('c = ',c,' ---- n = ',n)
        lmbpython=[pro_cv(m,n,c) for m in range(1,n+1)]
        for m in range(n+1):
            print('lmb abramovitz ',lmb(m,n,c),'  lmbpython ',pro_cv(m,n,c))
        print('===========================')
```