

```

# fichier créé avec python 3.9.1

##### AVERTISSEMENT #####
# dans python, la protection des types est encore très primitive...
# je décline toute responsabilité associée à ce défaut regrettable !
#####

import numpy as np

import MinimiLib_18 as mn # qui fait lui même appel à myFCN

# pour le calcul du chi2, les points expérimentaux sont dans myFCN

# noms des paramètres (utilisé dans les résultats pour mieux les reconnaître)
# parN devrait être array(range(30), str)...
# ...mais la gestion simpliste des types par python permet de simplifier
parN = ["a", "b", "c"]

# valeurs des paramètres
# par devrait être array(range(30), float) mais on simplifie de même
par = [1., -8., 10.]

# estimation des pas d'optimisation ;
# (si zéro : le paramètre est traité comme une constante et n'est pas optimisé)
# dPar devrait être array(range(30), float) mais on simplifie de même
dPar = [0.1, 0.1, 0.1]

nnPar = 3
# il faut indiquer sérieusement le nombre de paramètres utilisés (les suivants
# sont ignorés)... d'autant plus qu'on a simplifié !

stepp = 1.0
epsii = 0.0001
impp = 0 # on fait comme on veut, mais on laisse ainsi minimi choisir au mieux
errorss = True
minList = mn.minimi(nnPar,par,parN,dPar,stepp,epsii,impp,errorss)
print(minList)

the_a = 0.921678690383195 # on récupère les valeurs ajustées
the_b = -7.807398579117353
the_c = 12.593241467215671

import myFCN as mf # c'est là qu'on a placé les points
import matplotlib.pyplot as plt

xf = [i/200 for i in range(2000)] # on prépare la courbe
yf = [mf.fonc(3,[the_a,the_b,the_c],i/200) for i in range(2000)]
plt.plot(xf, yf, color='black')

plt.errorbar(mf.xPts, mf.yPts, xerr=mf.dxPts, yerr=mf.dyPts,
             fmt='o', markersize=3, color='black',

```

```
    capsize=3, ecolor='green')  
plt.show()
```