

```

# fichier créé avec python 3.9.1

import numpy as np

#####
#####
# prédéfini pour la "compilation", mais doit être redéfini par l'utilisateur
# selon le modèle étudié (ici bidon)

def fonc(NPar: int, p, x: float) -> float: # p devrait être un array(30) float

    # définit une fonction annexe utilisée par FCN

    # les paramètres utilisés dans l'expression ajustée sont ici notés p[i]
    # des noms plus explicites sont définis dans le programme appelant minimi,
    # mais non utilisés ici
    # (ceci n'a rien d'obligatoire, fonc n'est utilisé que par FCN qui peut être
    # écrit autrement)

    # le nombre maximum de paramètres est fixé à 30, ceci est imposé dans minimi
    # le nombre de paramètres effectivement actifs est NPar, les suivants sont
    # ignorés
    # (NPar n'est transmis ici que pour vérification éventuelle)

    # x représente par exemple une abscisse

    f: float = np.nan
    f = p[0] + x # l'indexation python commence à 0
    return f

#####
#####
# ici bidon pour compilation

autres = [0.]

#####
#####
# prédéfini pour la "compilation"...
# ...mais peut être redéfini par l'utilisateur

def FCN(NPar: int, xp, autres) -> float: # xp devrait être un array(30) float
    # autres est une structure contenant ce dont FCN peut dépendre en plus

    # définit la fonction minimisée (généralement un chi2)

    # le nom FCN est réservé par minimi
    # le nom myFCN est réservé par minimi

    # ici la structure autres est inutilisée (bidon)

```

```
f: float = 0.  
# on peut définir des fonctions annexes  
# par exemple fonc(NPar, params, abscisse)
```

```
f : fonc(NPar,xp,3)
```

```
return f
```