

```

from tkinter import *
from tkinter.filedialog import *
import tkinter.filedialog
import numpy as np
import os
import sys
import analyse
import script_final
import matplotlib.pyplot as plt

def ecrire_fichiers():
N=int(N_sb.get())
lref=float(lref_sb.get())
lEu=float(lEu_sb.get())
lTb=float(lTb_sb.get())
l=[lref, lEu, lTb]
script_final.script(N,l,basename[:-9])

def plot_2D():
global fenetre_tau
global Mafenetre
a1 = scale3.get()
a2 = scale4.get()
a3 = scale5.get()
seuil1 = scale1.get()
seuil2 = scale2.get()
normalize = scale7.get()
print(seuil1,seuil2,a1,a2,a3)
[(a_nm,b_nm,c_nm,d_nm,Phi_nm,Tau_nm),(a_std,b_std,c_std,d_std,Phi_std,Tau_std)] = analyse.fct_analyse(seuil1,
seuil2,
chemin,
fileref.get(),
filesignal.get(),
a1,a2,a3,normalize)

print (Tau_nm,Tau_std)
str_tau = "seuil zeros = " + str(seuil1) + " seuil max = "+str(seuil2)+ "\n Tau=" + str(Tau_nm) + '+/-' + str(Tau_std)
if fenetre_tau!=1:
fenetre_tau=1
Frame4 = Frame(Mafenetre,borderwidth=2,relief=GROOVE)
Frame4.pack(padx=10,pady=10,anchor=NW)
Tau = Label(Frame4,text=str_tau)
Tau.pack(padx=10,pady=10)
Tau.config(text = str_tau)

def plot_3D():
N=int(N_sb.get())

```

```
script_final.script(N,Lambda,basename[: -9],w=0,plot3D = 1,v=1)
plt.show()
```

```
def set_reference():
    fileref.set(askopenfilename(filetypes=[('Text', '.txt')],initialdir=chemin))
    print(fileref.get())
```

```
def set_signal():
    filesignal.set(askopenfilename(filetypes=[('Text', '.txt')],initialdir=chemin))
    print(filesignal.get())
```

```
#####
#####
#####
#####
#####
root = Tk()
fenetre_tau = 0
current=os.popen("echo $PWD", "r").read().rstrip()
chemin = tkinter.filedialog.askdirectory(parent=root,initialdir=current,
title='Please select a directory',
)
```

```
if len(chemin) > 0:
    print("You chose directory %s" % chemin)
```

```
basename = os.popen("ls "+ chemin +'/'+ "*00000.txt", "r").read().rstrip()
```

```
print(basename)
```

```
try:
```

```
    fichier = np.loadtxt(basename)
```

```
    print(int("22"))
```

```
except IOError:
```

```
    print("Erreur lors de l'ouverture du fichier !")
```

```
except ValueError:
```

```
    print('Replacing characters...')
```

```
os.system('sed -i "s/,./g" '+ basename[:9].replace(' ','\ ') + '*.txt')
```

```
fichier = np.loadtxt(basename)
Lambda = list(fichier[:,0])
```

```
Mafenetre = Tk()
Mafenetre.title('Programme')
Mafenetre['bg']='bisque' # couleur de fond
```

```
Frame1 = Frame(Mafenetre,borderwidth=2,relief=GROOVE)
Frame1.pack(side=LEFT,padx=10,pady=10)
```

```
Frame2 = Frame(Mafenetre,borderwidth=2,relief=GROOVE)
Frame2.pack(side=LEFT,padx=10,pady=10)
```

```
Frame3 = Frame(Mafenetre,borderwidth=2,relief=GROOVE)
Frame3.pack(side=LEFT,padx=10,pady=10)
```

```
# création d'un widget Label et d'un widget Button dans un widget Frame
```

```
Label(Frame1,text="Création fichiers").pack(padx=10,pady=10)
```

```
#####
```

```
#####Frame1
```

```
N_sb = Spinbox(Frame1, values=tuple(np.arange(1,5000)))
```

```
N_sb.delete(0,"end")
```

```
N_sb.insert(0,'1000')
```

```
N_sb.pack()
```

```
lref_sb = Spinbox(Frame1, values=tuple(Lambda))
```

```
lref_sb.delete(0,"end")
```

```
lref_sb.insert(0,str(Lambda[1605]))
```

```
lref_sb.pack()
```

```
lEu_sb = Spinbox(Frame1, values=tuple(Lambda))
```

```
lEu_sb.delete(0,"end")
```

```
lEu_sb.insert(0,str(Lambda[810]))
```

```
lEu_sb.pack()
```

```
lTb_sb = Spinbox(Frame1, values=tuple(Lambda))
```

```
lTb_sb.delete(0,"end")
```

```
ITb_sb.insert(0,str(Lambda[1169]))
ITb_sb.pack()
```

```
Button(Frame1,text="Créer !",fg='navy',command=ecrire_fichiers).pack(padx=10,pady=10)
```

```
#####
#####Frame2
```

```
Label(Frame2,text="Plot").pack(padx=20,pady=10)
```

```
fileref = StringVar(Frame2)
filesignal = StringVar(Frame2)
Entry(Frame2, textvariable=fileref).pack(padx=18,side="top")
Button(Frame2, text="Fichier ref", command=set_reference).pack(padx=10,pady=10,side="top")
Entry(Frame2, textvariable=filesignal).pack(padx=18,side="top")
Button(Frame2, text="Fichier signal", command=set_signal).pack(padx=10,pady=10,side="top")
```

```
#browseButton2 = Button(Frame2, text="Fichier signal", command=set_signal())
#browseButton2.pack(side="top")
normaliser = IntVar()
scale7 = Scale(Frame2,
variable=normaliser,
from_=0, to_=1,
orient = HORIZONTAL,
label = 'Normaliser ?' )
scale7.pack()
scale7.set(1)
```

```
remplace_carac = IntVar()
scale6 = Scale(Frame2,
variable=remplace_carac,
from_=0, to_=1,
orient = HORIZONTAL,
label = 'Remplacer "," par "." ?' )
scale6.pack()
```

```
afffit = IntVar()
scale3 = Scale(Frame2,
variable=afffit,
from_=0, to_=1,
orient = HORIZONTAL,
label = 'afficher fit' )
scale3.pack()
scale3.set(1)
```

```
affabcd = IntVar()
scale4 = Scale(Frame2,
```

```
variable=affabcd,  
from_=0, to_=1,  
orient = HORIZONTAL,  
label = 'afficher a,b,c,d' )  
scale4.pack()  
scale4.set(1)
```

```
affseuils = IntVar()  
scale5 = Scale(Frame2,  
variable=affseuils,  
from_=0, to_=1,  
orient = HORIZONTAL,  
label = 'afficher seuils' )  
scale5.pack()  
scale5.set(1)
```

```
seuil_bande = DoubleVar()  
scale1 = Scale(Frame2,  
variable=seuil_bande,  
from_=0, to_=1,  
digits=3,  
resolution=0.01,  
orient = HORIZONTAL,  
label = 'seuil_bande' )  
scale1.pack()  
scale1.set(0.15)
```

```
seuil_maxs = DoubleVar()  
scale2 = Scale(Frame2,  
variable=seuil_maxs,  
from_=0, to_=1,  
digits=3,  
resolution=0.01,  
orient = HORIZONTAL,  
label = 'seuil_max' )  
scale2.pack()  
scale2.set(0.85)
```

```
Button(Frame2,text="Plotter 2D!",fg='navy',command=plot_2D).pack(padx=10,pady=10)
```

```
#####  
#####Frame2
```

```
Label(Frame3,text="Plot 3D").pack(padx=10,pady=10)  
Button(Frame3,text="Plotter !",fg='navy',command=plot_3D).pack(padx=10,pady=10)
```

Mafenetre.mainloop()