TP 6 : Multimètres

Code Capytale : 33d8-7689854

Entrée[1]:
```
import numpy as np
import matplotlib.pyplot as plt
```

Caractéristique d'un résistance

Mesures

Entrée[2]:
```
I=[0.661,0.896,1.415,1.691,2.122,2.573]
U=[3.05,4.16,6.56,7.84,9.85,11.86]
```

Régression linéaire

Entrée[3]:
```
[a,b]=np.polyfit(I,U,1)
```

Tracé

Entrée[4]:
```
Umodel=[a*i+b for i in I]
plt.clf() #pour effecer les tracés précédents
plt.plot(I,U,'o',label='mesures')
plt.plot(I,Umodel,':',label='regression lineaire')
plt.legend()
plt.grid()
plt.axhline()
plt.axvline()
plt.xlabel('I(mA)')
plt.ylabel('U(V)')
plt.show()
```

Affichage de R

Entrée[5]:
```
print("R = {:.5f} kohm".format(a))# 5 chiffres après la virgule, valeur en kohm (car I en mA)
```

```
R = 4.61567 kohm
```

Incertitude sur R : simulation Monte Carlo de N régressions linéaires

N simulations avec précisions fixes

Entrée[6]:
```
N = 10000
A = [] # Listes des valeurs de R
delta_I = 0.012  #unique valeur pour simplifier le code
delta_U = 0.09 #unique valeur pour simplifier le code
for k in range(N):
    Isim = I + np.random.uniform(-delta_I,delta_I,len(I))
    Usim = U + np.random.uniform(-delta_U,delta_U,len(I))
    a,b = np.polyfit(Isim,Usim,1)  #régression linéaire
    A.append(a)
```

Calcul de l'incertitude type sur R

Entrée[7]:
```
u_a=np.std(A,ddof=1)
print("u(a) = {:.3f} kohm ".format(u_a))
```

```
u(a) = 0.038 kohm
```

N simulations avec précisons variables

Précisions des mesures

Entrée[8]:
```
delta_I = [0.5/100*i+5*0.001 for i in I] #en mA
print(delta_I)
delta_U = [0.5/100*u+5*0.01 for u in U] #en V
print(delta_U)
```

```
[0.008305, 0.00948, 0.012075, 0.013455000000000002, 0.015609999999999999, 0.017865]
[0.06525, 0.0708, 0.0828, 0.0892, 0.09925, 0.10930000000000001]
```

N simulations avec précisons variables

Entrée[9]:
```
N=10000
A=[]
for k in range(N):
    Isim=[]
    Usim=[]
    for j in range(len(I)):
        Isim.append(np.random.uniform(I[j]-delta_I[j],I[j]+delta_I[j]))
        Usim.append(np.random.uniform(U[j]-delta_U[j],U[j]+delta_U[j]))
    a,b = np.polyfit(Isim,Usim,1)
    A.append(a)
```

Calcul de l'incertitude type sur R

Entrée[10]:
```
u_R=np.std(A,ddof=1)
print("u(a) = {:.3f} kohm ".format(u_R))
```

```
u(a) = 0.039 kohm
```

Caractéristique d'un dipôle actif

Entrée[11]:
```
I=[0,7.99,15.2,21.3,33.5,65.3]
U=[4.63,4.22,3.84,3.54,2.95,1.4]
```

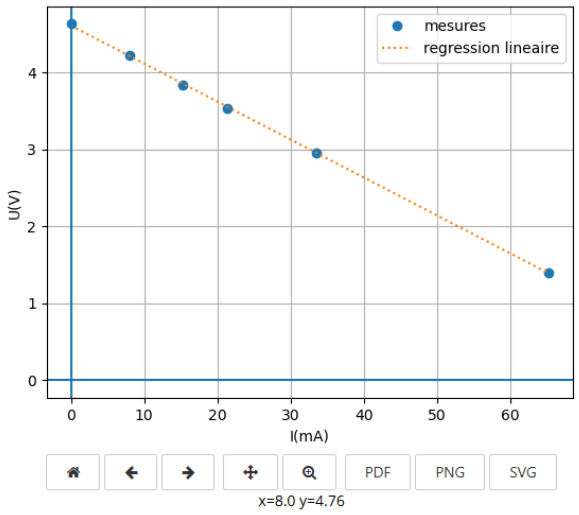
Régression linéaire

Entrée[12]:
```
[a,b]=np.polyfit(I,U,1)
```

Tracés

Entrée[13]:
```
Umodel=[a*i+b for i in I]
plt.clf() #pour effecer les tracés précédents
plt.plot(I,U,'o',label='mesures')
plt.plot(I,Umodel,':',label='regression lineaire')
plt.legend()
plt.grid()
plt.axhline()
plt.axvline()
plt.xlabel('I(mA)')
plt.ylabel('U(V)')
plt.show()
```

Figure 1



Paramètres du modèle équivalent de Thévenin

Entrée[14]:
```
print("E = {:.2f} V".format(b)) # 2 chiffres après la virgule
print("r = {:.2f} ohm".format(-a*1000)) # 2 chiffres après la virgule
```

```
E = 4.61 V
r = 49.30 ohm
```