

Menthonecorr.py

```
01| ##Isomérisation de la menthone
02|
03| import numpy as np
04| import matplotlib.pyplot as plt
05| from scipy.optimize import bisect
06|
07| ##Données
08| kd= 12.3e-3
09| ki=23.0e-3
10| PRSm=-29.6
11| PRSp=91.7
12| alpha0=-10.3
13|
14| ##Avancement volumique en fonction du temps
15| C0=alpha0/PRSm
16| k=kd+ki
17| #Résolution analytique
18| def Xana(t):
19|     return kd*C0*(1-np.exp(-k*t))/k
20|
21| ## Pouvoir rotatoire en fonction du temps
22| T=np.linspace ( 0, 100 )
23| X1= Xana(T)
24| ALPHA = alpha0+ X1*(PRSp-PRSm) #tableau des
valeurs de alpha
25|
26|
27| plt.figure(0)
28| plt.plot(T,ALPHA,'+b')
29| plt.grid()
30| plt.legend()
31| plt.xlabel("t(min)")
32| plt.ylabel(r"$\alpha$")
33| plt.show()
34|
35|
36| ## résolution avec la fonction odeint
37| from scipy.integrate import odeint
38|
39| def F(X,t):
40|     return (kd*C0-k*X)
41|
42| #Comparaison des deux méthodes de résolution
43| X2 = odeint ( F, [0],T)
44|
45| plt.figure(1)
46| plt.plot(T,X1,'+b',label="resolution analytique")
47| plt.plot(T,X2,'xr',label="résolution numérique")
48| plt.grid()
49| plt.legend()
50| plt.xlabel("t(min)")
51| plt.ylabel("x")
52| plt.show()
```

