Capacité numérique : méthode dichotomique

MPSI 1

Année 2025-2026

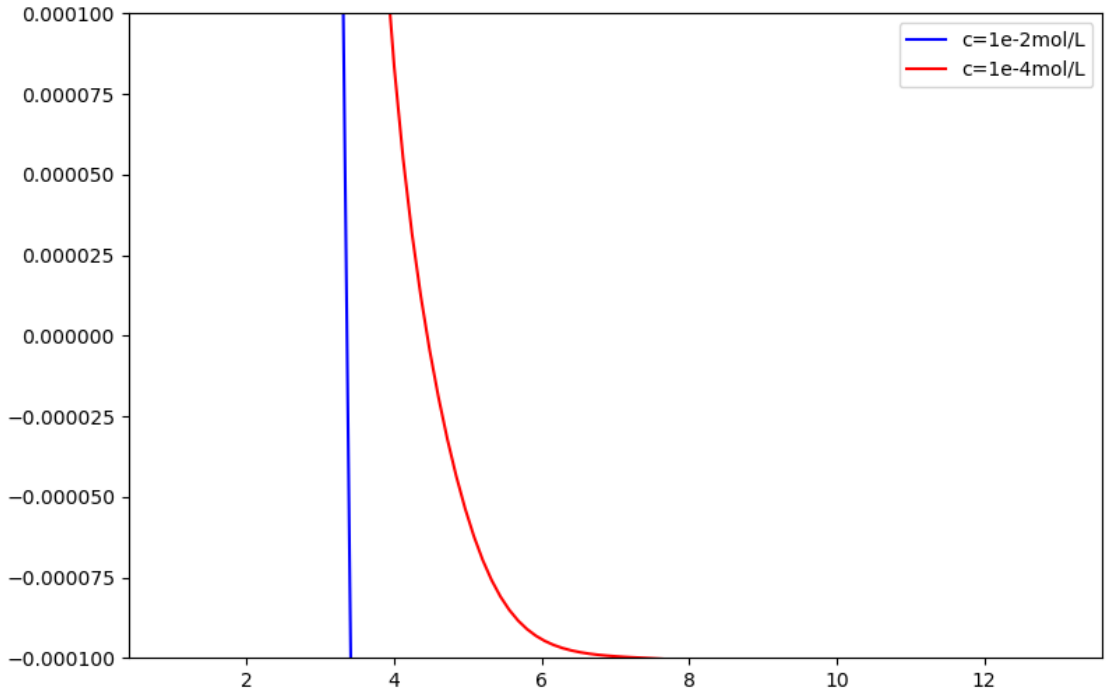
[1]:
```
import numpy as np
import matplotlib.pyplot as plt
import scipy.optimize as opt
```

[11]:
```
pKA=4.7
pKe=14
N=1001
```

[15]:
```
def f(pH,c) :
    return np.power(10.,-pH)-np.power(10.,pH-pKe)-c/(1+np.power(10.,pKA-pH))

datapH=np.linspace(1,pKe-1,101)
dataf1=f(datapH,1e-2)
dataf2=f(datapH,1e-4)

fig,ax=plt.subplots(1,figsize=(9,6))
ax.plot(datapH,dataf1,'b',label='c=1e-2mol/L')
ax.plot(datapH,dataf2,'r',label='c=1e-4mol/L')
ax.set_ylim(-1e-4,1e-4)
ax.legend(loc='best')
plt.show()
```



[16]:
```
pHb=0
pHh=14
eps=1e-3
c=1e-2
while (pHh-pHb)>eps :
    pHm=0.5*(pHh+pHb)
    if f(pHm,c)==0 :
        pHb=pHm
        pHh=pHm
    if f(pHm,c)*f(pHb,c)<0 :
        pHh=pHm
    else :
        pHb=pHm
print("pH=",0.5*(pHb+pHh))
```

```
pH= 3.35943603515625
```

[17]:
```
pHb=0
pHh=14
eps=1e-3
c=1e-4
while (pHh-pHb)>eps :
    pHm=0.5*(pHh+pHb)
    if f(pHm,c)==0 :
```

```
        pHb=pHm
        pHh=pHm
    if f(pHm,c)*f(pHb,c)<0 :
        pHh=pHm
    else :
        pHb=pHm
print("pH=",0.5*(pHb+pHh))
```

```
pH= 4.44635009765625
```

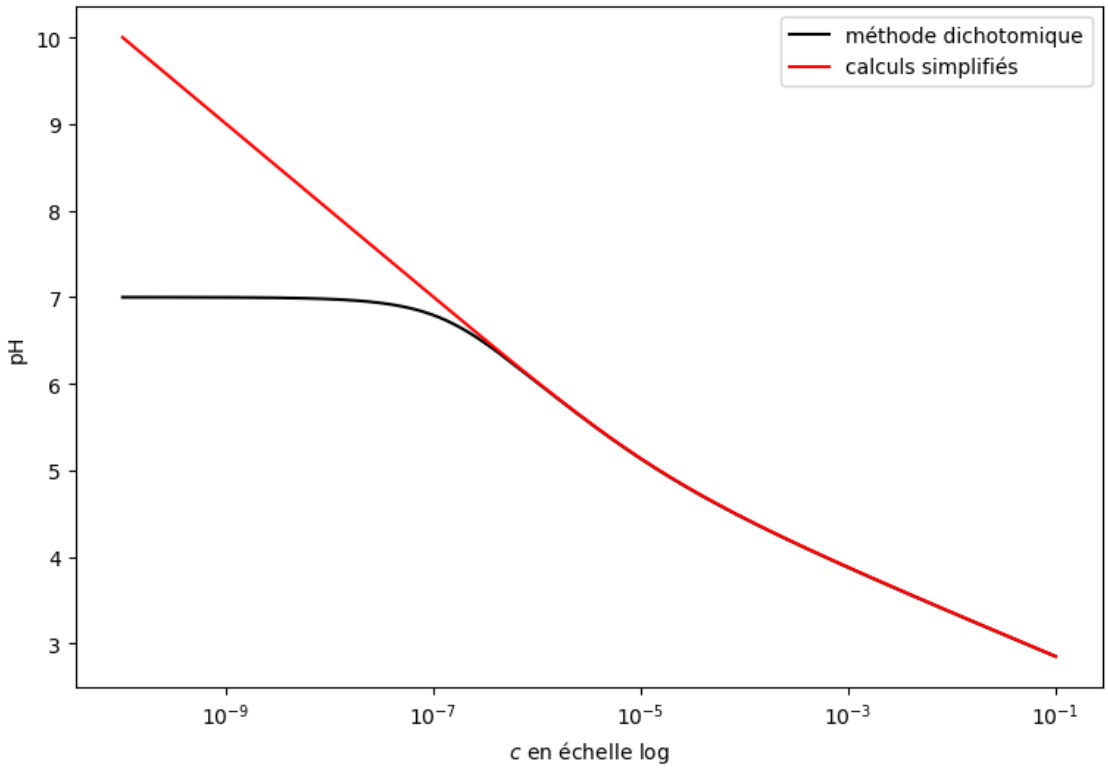
[21]:
```
opt.bisect(f,0,14,args=(1e-2))
```

[21]:
```
3.3596987931202875
```

[19]:
```
opt.bisect(f,0,14,args=(1e-4))
```

[19]:
```
4.446205239865549
```

[33]:
```
data_c=np.logspace(-1,-10,N)
pH_s=np.zeros(N)
pH_e=np.zeros(N)
for n in range(N) :
    c=data_c[n]
    pH_s[n]=-np.log10(np.sqrt(0.25*np.power(10,-2*pKA)+np.power(10.,-pKA)*c)-0.
 ↪5*np.power(10.,-pKA))
    pH_e[n]=opt.bisect(f,0,14,args=(c))
fig,ax=plt.subplots(1,figsize=(9,6))
ax.semilogx(data_c,pH_e,'k',label="méthode dichotomique")
ax.semilogx(data_c,pH_s,'r',label="calculs simplifiés")
ax.set_xlabel("$c$ en échelle log")
ax.set_ylabel("pH")
ax.legend(loc="best")
plt.show()
```



[23]:

[23]:
```
array([1.00000000e-01, 9.79489985e-02, 9.59400632e-02, ...,
       1.04231743e-10, 1.02093948e-10, 1.00000000e-10])
```

[ ]: