

Beer-Lambert smartphone CuSO₄

April 4, 2019

1 Utilisation de 2 smartphones pour analyser la composition d'une solution - MPS

1.1 On cherche à déterminer la concentration en masse C_m d'une solution colorée de sulfate de cuivre.

1.1.1 Pré-requis :

- Connaître la définition de la concentration en masse C_m ;
- Savoir réaliser une solution par dilution ;
- Connaître la "formule de la dilution" : $C_m \times V_m = C_f \times V_f$;
- Savoir qu'une image est un ensemble de pixels composés de 3 sous-pixels, chacun codé sur 8 bits, correspondant à $2^8 = 256$ teintes comprises entre 0 et 255.

1.1.2 Capacités mises en uvre :

- Réaliser une dilution ;
- Concevoir et réaliser un protocole expérimental ;
- Appliquer un programme Python pour tracer et exploiter une courbe d'étalonnage ;
- Comprendre et exploiter les composantes RGB d'un signal lumineux.

1.1.3 Matériel par groupe de 4 :

- 1 solution mère de sulfate de cuivre (Cu^{2+}, SO_4^{2-}) concentration en masse $C_m = 100 \text{ g} \cdot L^{-1}$
- 3 pipettes jaugées de 5, 10 et 20 mL ;
- 1 fiole jaugée de 50 mL ;
- 1 ratelier de 8 tubes à essai ;
- 2 béchers de 50 mL ;
- 2 smartphones avec les applications :

– pour émettre une lumière colorée :

- * **Physics Toolbox Suite** sur Android : <https://play.google.com/store/apps/details?id=com.chryst>
- * **Physics Toolbox Color Gen** sur iOS : <https://itunes.apple.com/us/app/physics-toolbox-color-gen/id1372521401>

– pour lire les composante RGB en un point :

- * **Color Grab** sur Android : <https://play.google.com/store/apps/details?id=com.loomatix.colorgra>
- * **Physics Toolbox Color Gen** sur iOS : <https://itunes.apple.com/us/app/pixel-picker-image-color-picker/id930804327#?platform=iphone>

1.1.4 Marche à suivre :

Vous allez travailler par groupes de 4. La solution est une solution de sulfate de cuivre.

Préparez d'abord des solutions de concentrations 0 - 10,0 - 20,0 - 40,0 - 60,0 - 80,0 - 100,0 $g \cdot L^{-1}$ et les placer dans un tube à essai.

Identifier la couleur de travail, puis placer un par un les tubes devant le smartphone émetteur de couleur et relever la composante R, V ou B.

Vous devez ensuite exécuter un à un les blocs du programme ci-dessous en sélectionnant le bloc puis en cliquant sur lecture (▶)

On vous demande d'abord d'entrer : - les valeurs des différentes concentrations des solutions constituants l'échelle de teinte ; - la valeur de la composante R, V ou B correspondante.

```
In [1]: import numpy as np
```

```
C=[]
RGB=[]
N = int(input('Entrer le nombre total de solutions de l'échelle de teinte (eau comprise !) : '))
for i in range(N):
    Ci = np.float(input('Entrer la valeur de C en g/L : '))
    RGB_i = np.float(input('Entrer la valeur de la composante choisie du signal RVB : '))
    RGB.append(RGB_i)
    C.append(Ci)
```

```
Entrer le nombre total de solutions de l'échelle de teinte (eau comprise !) : 7
Entrer la valeur de C en g/L : 0
Entrer la valeur de la composante choisie du signal RVB : 230
Entrer la valeur de C en g/L : 10
Entrer la valeur de la composante choisie du signal RVB : 159
Entrer la valeur de C en g/L : 20
Entrer la valeur de la composante choisie du signal RVB : 104
Entrer la valeur de C en g/L : 40
Entrer la valeur de la composante choisie du signal RVB : 52
Entrer la valeur de C en g/L : 60
Entrer la valeur de la composante choisie du signal RVB : 23
Entrer la valeur de C en g/L : 80
Entrer la valeur de la composante choisie du signal RVB : 11
Entrer la valeur de C en g/L : 100
Entrer la valeur de la composante choisie du signal RVB : 5
```

On calcule alors la proportion de lumière traversant la solution T en % et la proportion de lumière absorbée P en %.

On trace les courbes $T = f(C)$ et $P = f(C)$:

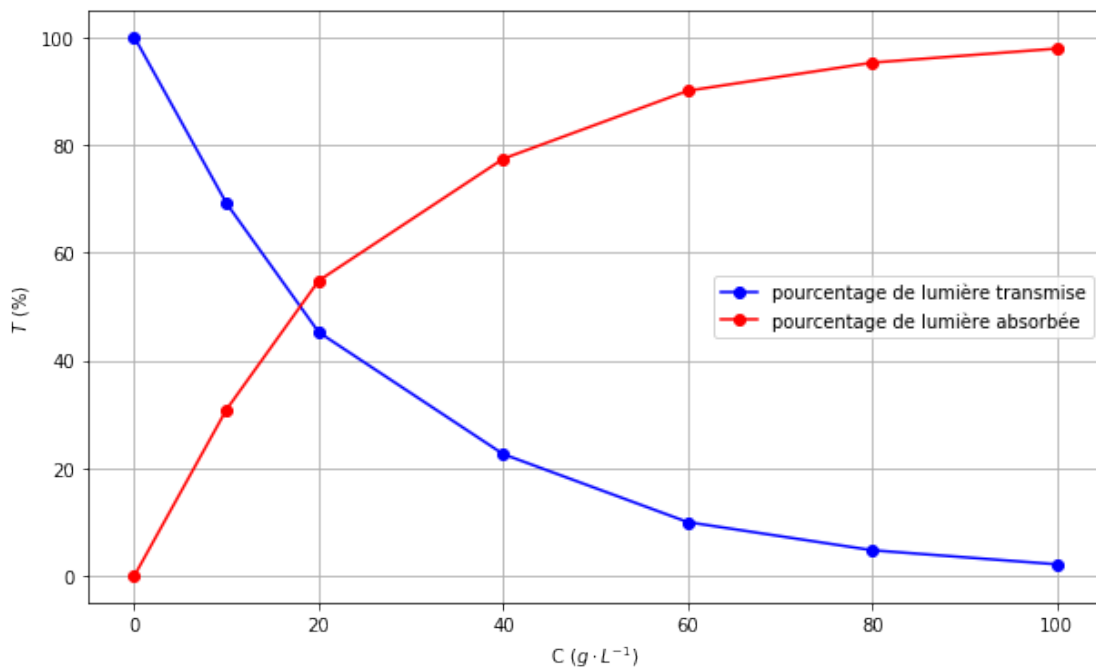
```

In [3]: T = []
        P = []
        for i in range(len(RGB)):
            T_i = 100*(RGB[i])/max(RGB)
            T.append(T_i)
            P.append(100-T_i)

import matplotlib.pyplot as plt

plt.rcParams['figure.figsize'] = [10,6]
plt.xlabel('C ( $\text{g}\cdot\text{L}^{-1}$ )')
plt.ylabel('$T$ (%)')
plt.grid()
plt.plot(C,T,'bo-',label='pourcentage de lumière transmise')
plt.plot(C,P,'ro-',label='pourcentage de lumière absorbée')
plt.legend()
plt.show()

```



```

In [4]: c=0
        while c==0:
            Q1 = input('Peut-on dire que la lumière est absorbée de manière proportionnelle à la concentration? (o/n)')
            if Q1=='o':
                print('Attention : si c'était le cas, on observerait une droite passant par l'origine')
                c=0
            elif Q1=='n':
                c=1

```

```

    print('En effet : la courbe rouge nest pas une droite.')
    break
else:
    print('choisir la réponse o ou n')
    c==0

```

Peut-on dire que la lumière est absorbée de manière proportionnelle à la concentration ? oui : o

Attention : si c'était le cas, on observerait une droite passant par l'origine !

Peut-on dire que la lumière est absorbée de manière proportionnelle à la concentration ? oui : o

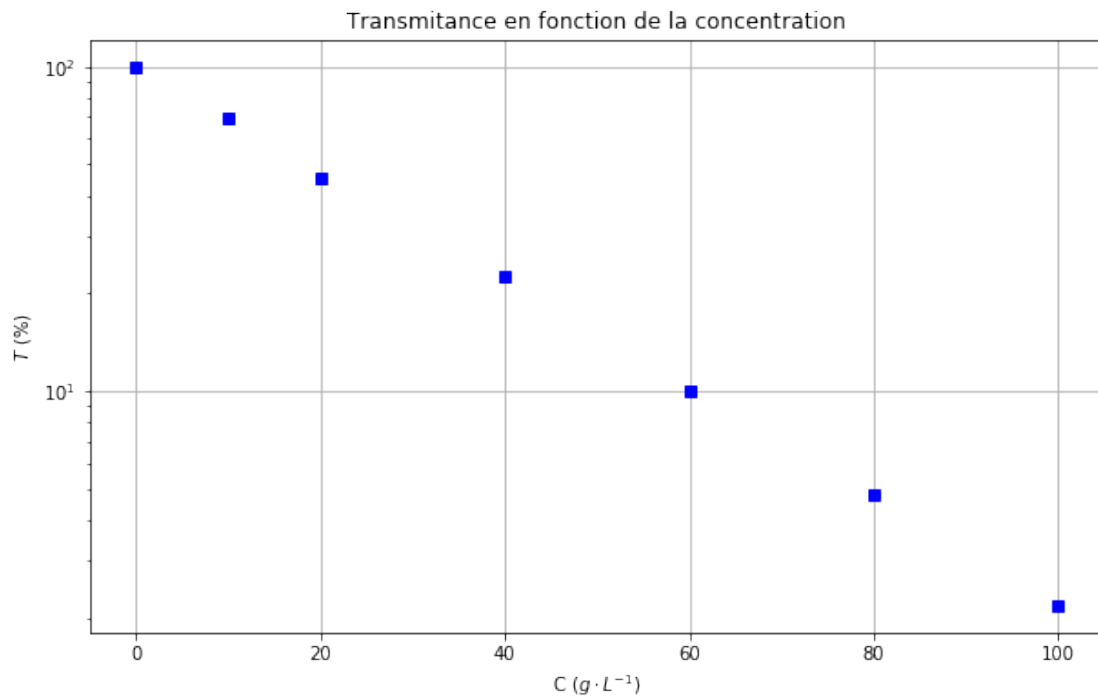
En effet : la courbe rouge nest pas une droite.

Traçons la courbe $T = f(C)$ avec un axe vertical gradué en puissances de 10 :

```

In [5]: plt.rcParams['figure.figsize'] = [10,6]
plt.title("Transmittance en fonction de la concentration")
plt.xlabel('C ( $\text{g}\cdot\text{L}^{-1}$ )')
plt.ylabel('$T\ (\%)$')
plt.grid()
plt.yscale('log')
plt.plot(C,T, 'bs')
plt.show()

```

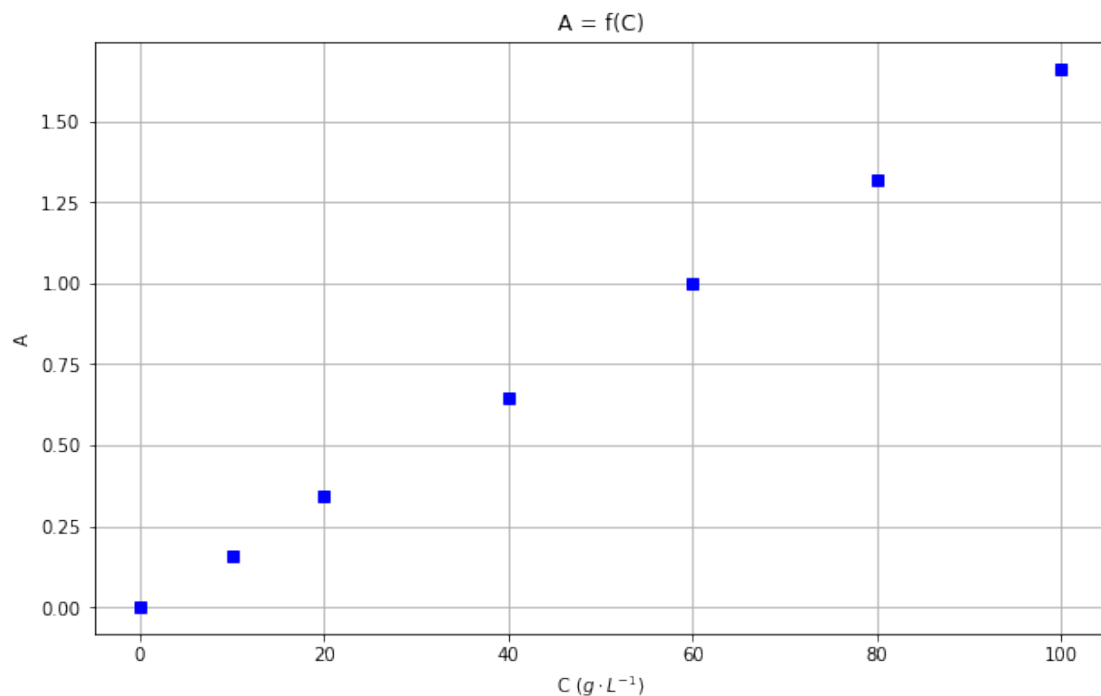


On définit une nouvelle grandeur A que l'on appelle l'absorbance par la formule suivante :

$$A = -\log T$$

On calcule A pour chaque solution et on trace $A = f(C)$:

```
In [6]: A = []
        for x in T:
            Ai = -np.log10(x/100)
            A.append(Ai)
        plt.rcParams['figure.figsize'] = [10,6]
        plt.title("A = f(C)")
        plt.xlabel('C ( $\text{g}\cdot\text{L}^{-1}$ )')
        plt.ylabel('A')
        plt.grid()
        plt.plot(C,A, 'bs')
        plt.show()
```



On observe que cette nouvelle grandeur A semble proportionnelle à la concentration C .
On modélise donc le nuage de points par une droite passant par l'origine :

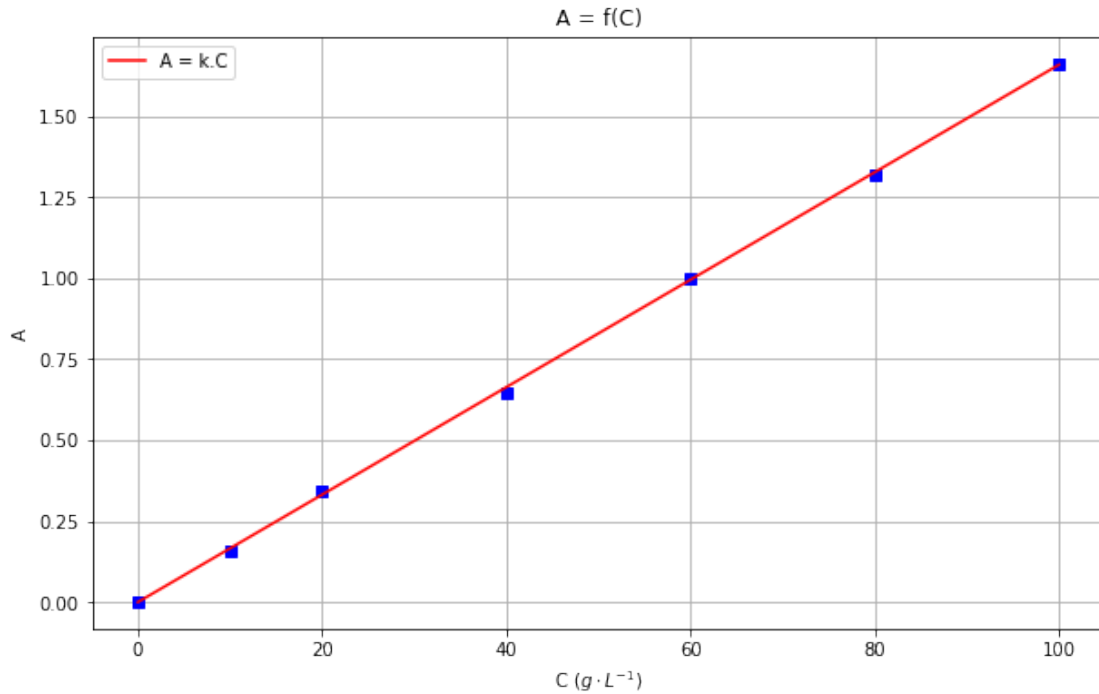
```
In [7]: Afit = np.polyfit(C,A,1)
        k = Afit[0]
        Amod = k * np.array(C)

        plt.rcParams['figure.figsize'] = [10,6]
```

```

plt.title("A = f(C)")
plt.xlabel('C ($g \cdot L^{-1}$)')
plt.ylabel('A')
plt.grid()
plt.plot(C,A,'bs')
plt.plot(C,Amod,'r-',label='A = k.C')
plt.legend()
plt.show()
print('L'équation du modèle de la droite est : A = k x C, avec k =',round(k,5),'L/mg')

```



L'équation du modèle de la droite est : $A = k \times C$, avec $k = 0.01659$ L/mg

On peut alors déterminer la concentration en masse C_m de la solution inconnue :

```

In [9]: RGBm = np.float(input('Entrer la valeur de la composante RGB du signal transmis par la s
Am = -np.log10(RGBm/max(RGB))

```

```

yh=[Am,Am]
xh=[0,Am/k]
xv=[Am/k,Am/k]
yv=[0,Am]

```

```

plt.rcParams['figure.figsize'] = [10,6]
plt.title("A = f(C)")

```

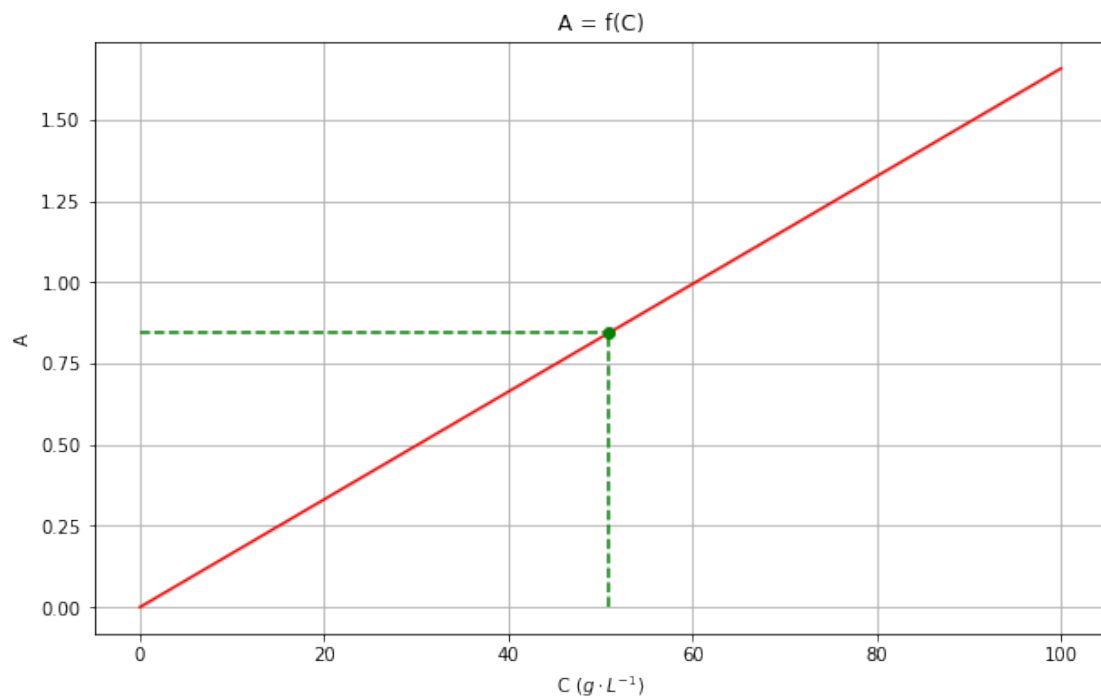
```

plt.xlabel('C ($g \cdot L^{-1}$)')
plt.ylabel('A')
plt.grid()
plt.plot(C,Amod,'r-')
plt.plot(Am/k,Am,'go')
plt.plot(xv,yv,'g--')
plt.plot(xh,yh,'g--')
plt.show()

print('La concentration en soluté de la solution inconnue est Cm =',round(Am/k,2),' g/L')

```

Entrer la valeur de la composante RGB du signal transmis par la solution inconnue : 33



La concentration en soluté de la solution inconnue est Cm = 50.83 g/L