

Modélisation des couleurs en informatique

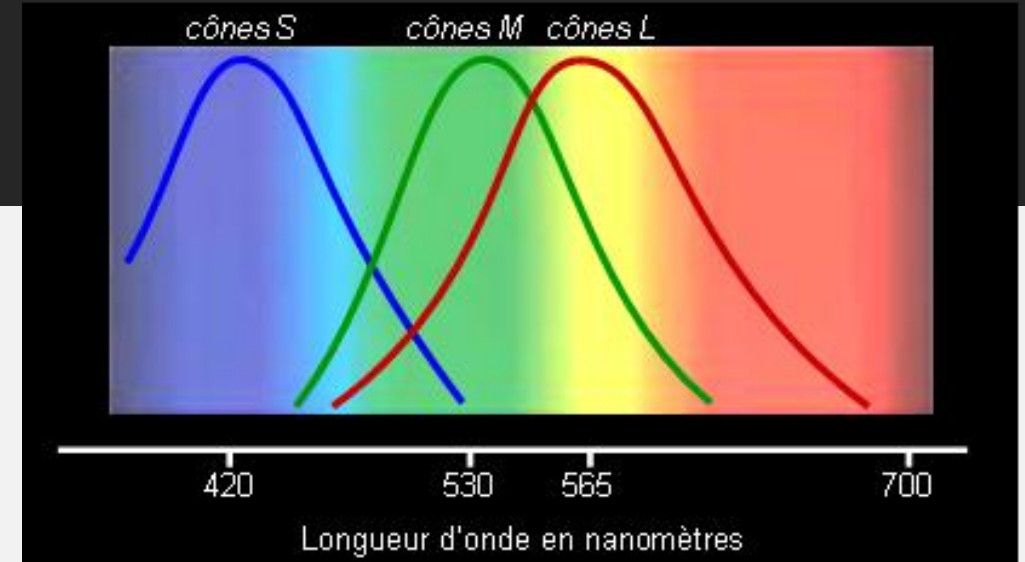
Problématique : Comment modéliser numériquement l'expérience des trous de Young polychromatique pour imiter au mieux la vision réelle ?

Plan

-
- Problème de la représentation des couleurs
 - Reprise de l'expérience de J. Guild pour trouver les valeurs RGB
 - Résultats et norme par la CIE
 - Modélisation de l'expérience des trous de Young et résultats intéressants

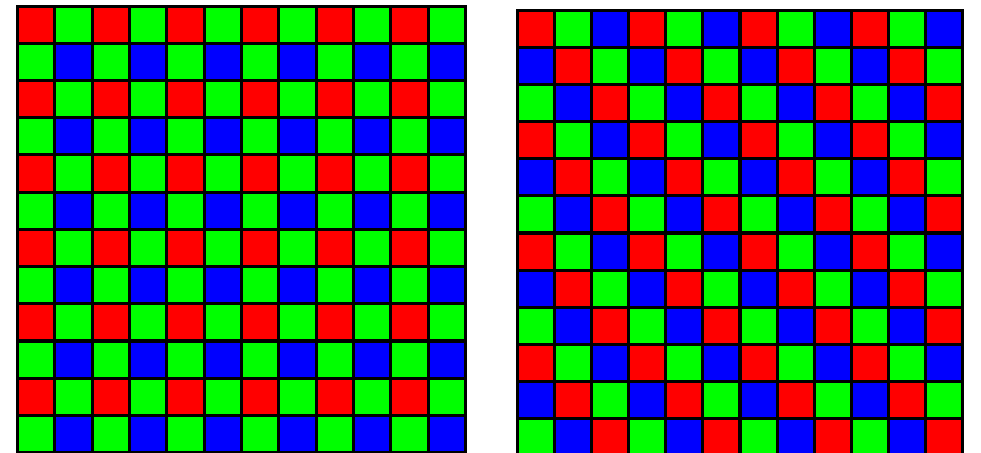
Couleurs pour l'œil et pour l'ordinateur

- 3 types de cônes
- Codage en RGB
- Sensibilité plus importante au vert
- Pixels avec deux verts/sur plus d'octets



Université Virtuelle Environnement et Développement durable

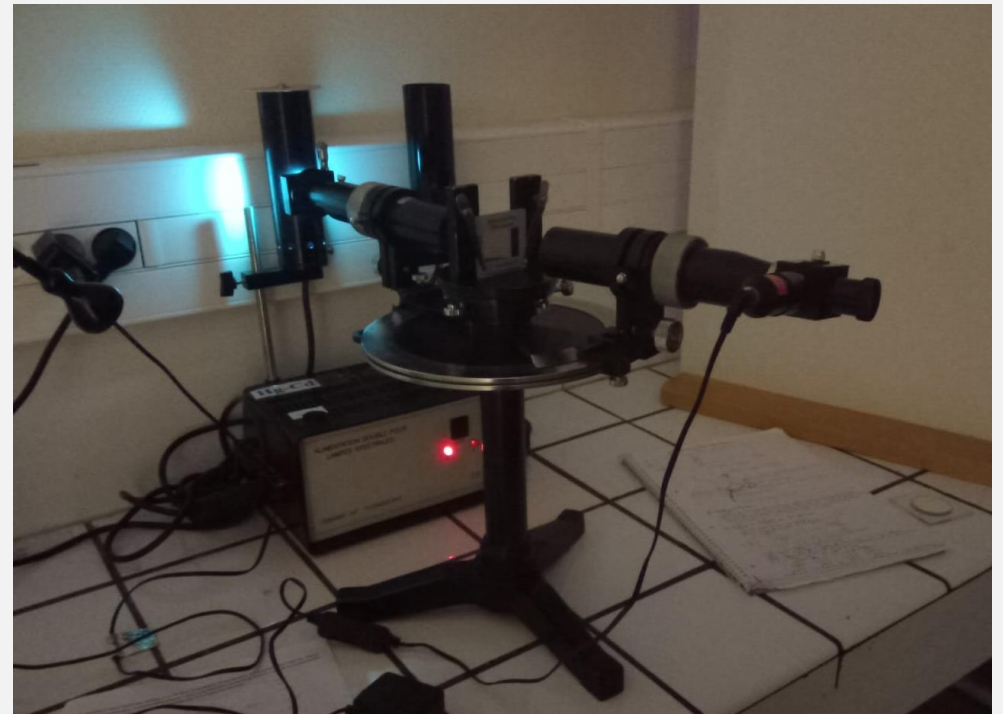
Quadrillage de Bayer et exemple de matrice RGB



Color Filter Array Designs, Quadiblock

Manipulation

-
- **Objectif** : trouver les valeurs RGB des couleurs des longueurs d'onde d'une lampe spectrale mercure cadmium
 - **Protocole initial** :
 - Diffracter la lampe spectrale avec un réseau
 - Comparer les couleurs au goniomètre à une couleur donnée par ordinateur
 - **Difficultés** :
 - Intensité lumineuse
 - Température de couleur de l'écran



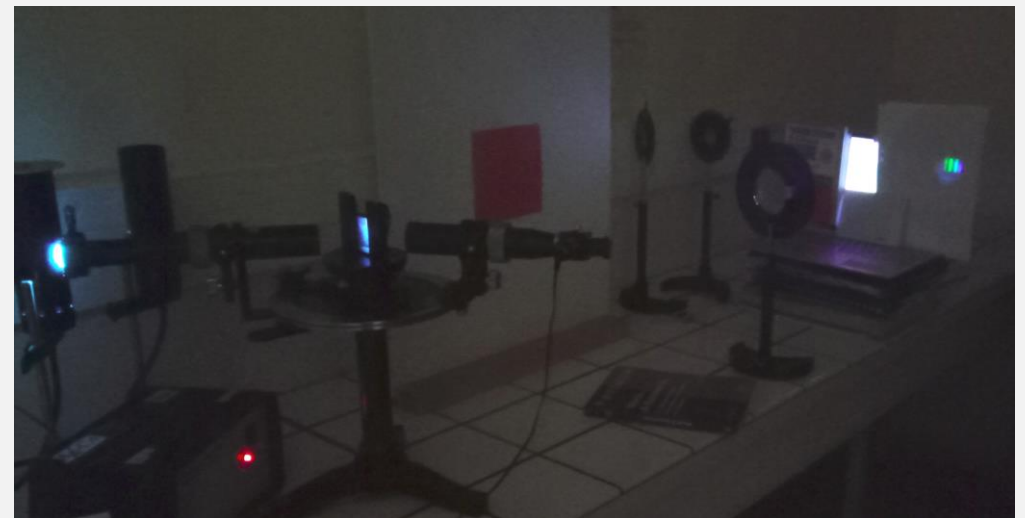
Manipulation

- **Modification des protocoles et difficultés**

- Projection sur papier blanc des différentes couleurs diffractées
- Dissimulation des lumières de l'écran avec des caches

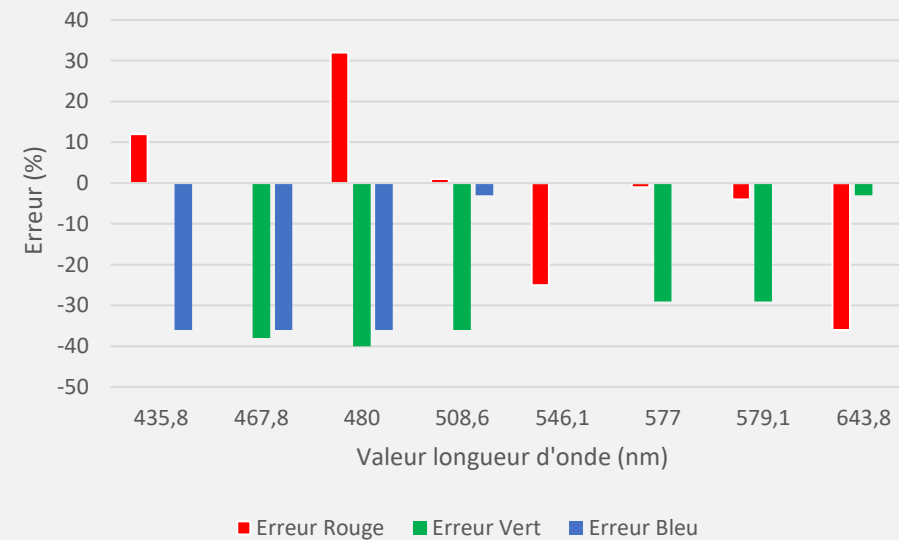
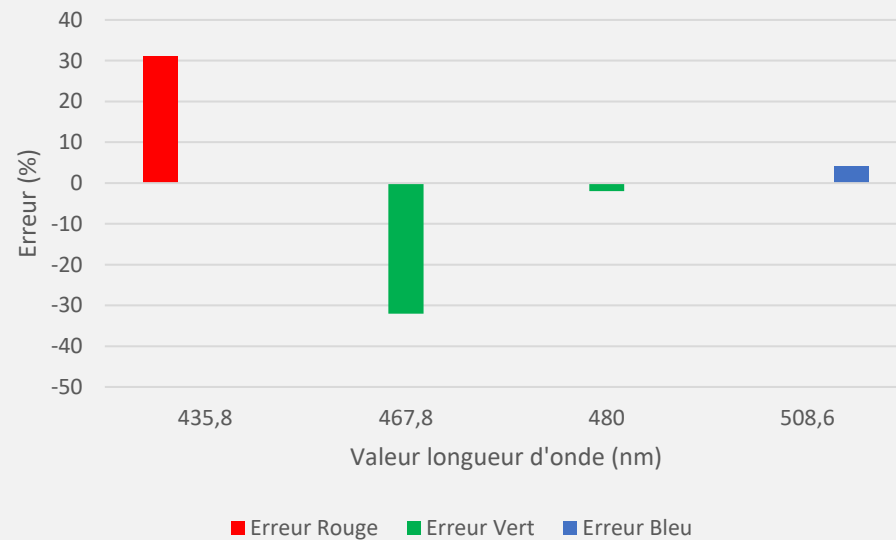
- **Difficultés**

- Lumières parasites
- Ecran
- Intensité lumineuse



Manipulation

Analyse des résultats



Norme CIE

- Objectif : déterminer des fonctions colorimétriques

$$\{C\} = \int_0^\infty f(\lambda) c_1(\lambda) d\lambda \cdot \{P_1\} + \int_0^\infty f(\lambda) c_2(\lambda) d\lambda \cdot \{P_2\} + \int_0^\infty f(\lambda) c_3(\lambda) d\lambda \cdot \{P_3\}.$$

- État présent : formalisation par un changement de base (x, y, z)
- Changement de base :

$$M = \begin{pmatrix} 2,768\,892 & 1,751\,748 & 1,130\,160 \\ 1,000\,0 & 4,590\,7 & 0,060\,100 \\ 0,000\,0 & 0,056\,508 & 5,594\,292 \end{pmatrix}$$

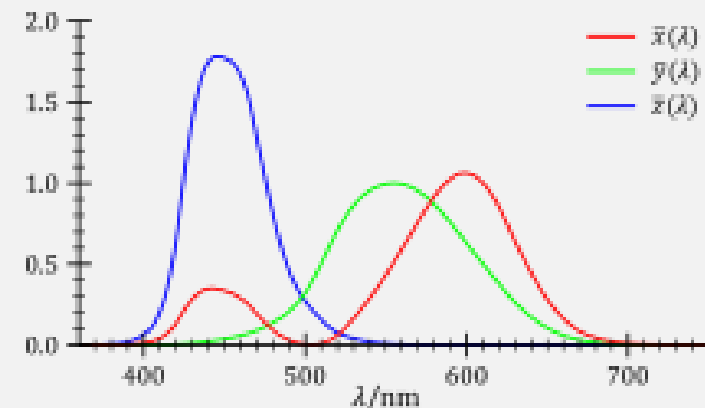
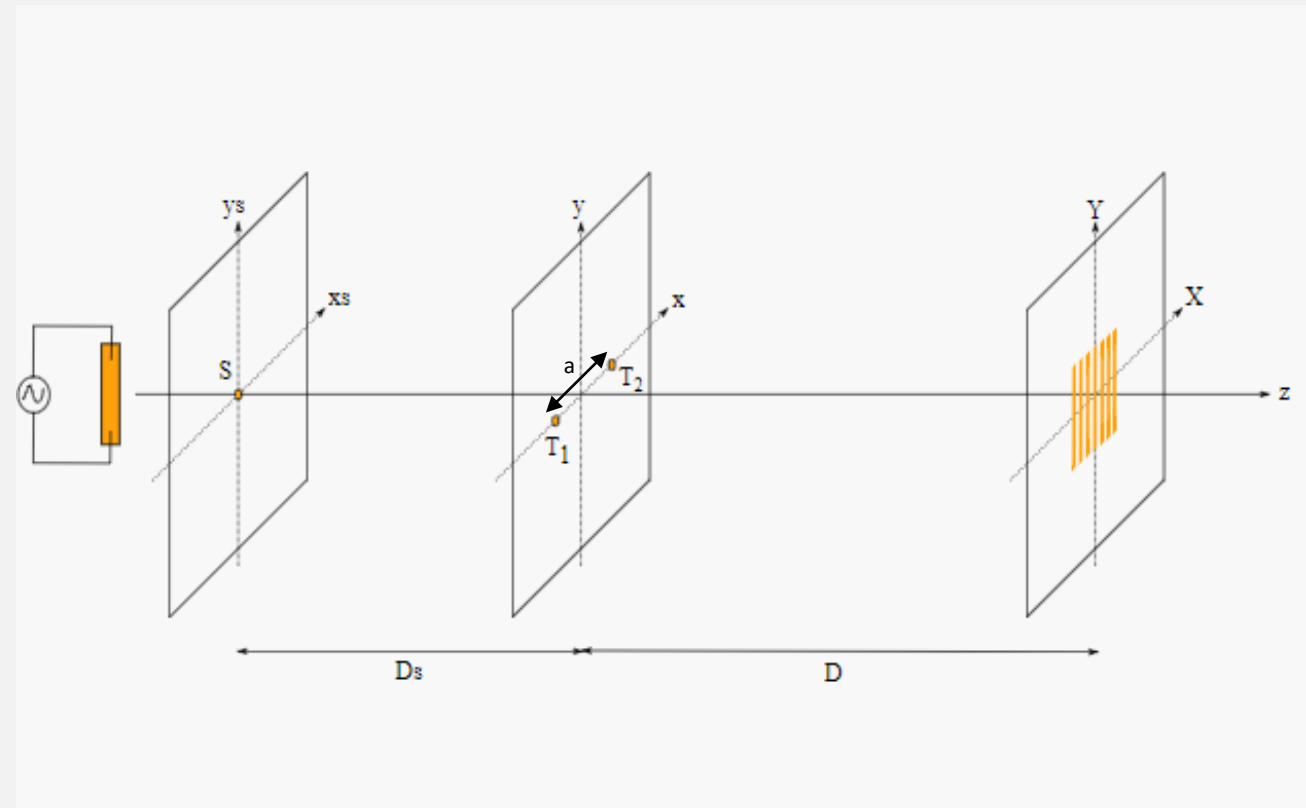


Image libre de droits, issues des chiffres de Wyszecki, Günter and Stiles, Walter Stanley (2000) Color Science: Concepts and Methods, Quantitative Data and Formulae

Modélisation

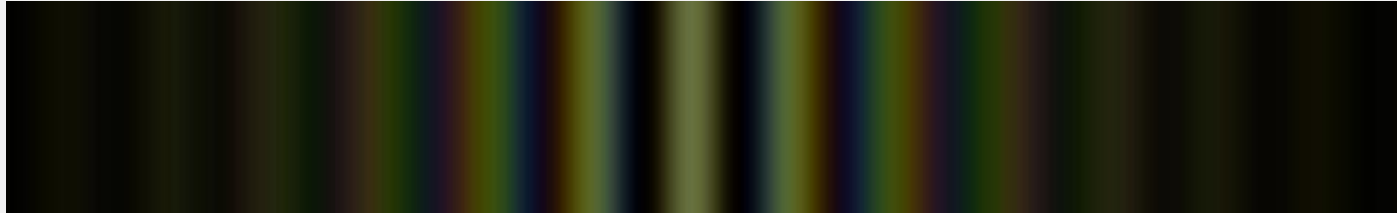
- Expérience des trous de Young
 - Utilisation des données théoriques et expérimentales
 - Calcul sur chaque longueur d'onde
 - Moyenne des intensités en chaque point

$$I(x) = 2I(1 + \cos \frac{2\pi ax}{\lambda D})$$

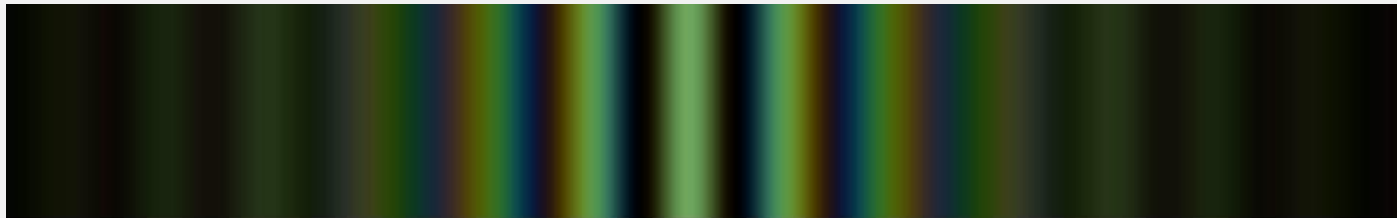


Informatique appliquée aux sciences physiques, Frédéric Legrand

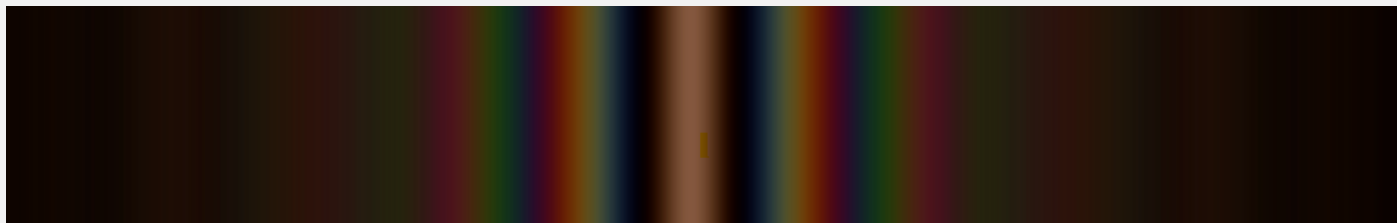
Modélisation



Lampe mercure cadmium
valeurs expérimentales



Lampe mercure cadmium
valeurs théoriques



Lumière avec des longueurs
d'onde équiréparties à la même
intensité



Lumière blanche suivant la loi de
Planck

$$L(\nu, T) = \frac{2h\nu^3}{c^2} \frac{1}{e^{\frac{h\nu}{kT}} - 1}$$

Conclusion

- Les différents espaces de couleur :
 - Physique
 - Mathématique
 - RGB

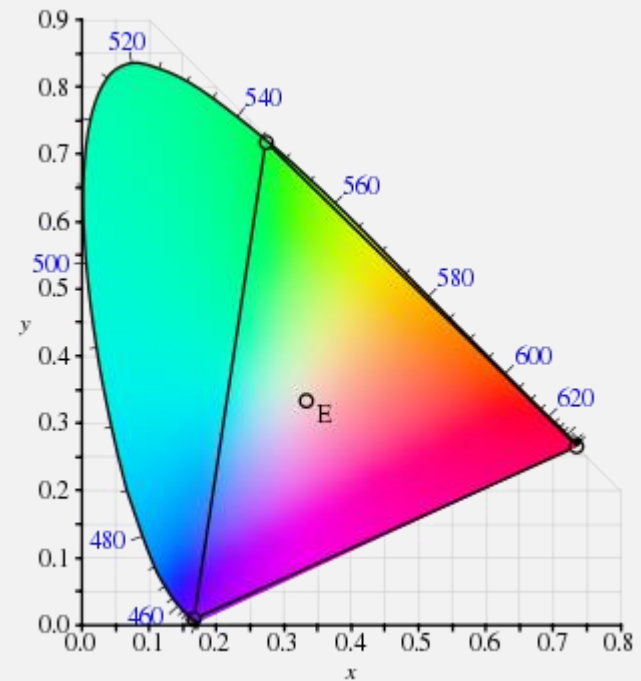


Image libre de droits par BenRG, espace RGB de 1931

Annexe

	R	G	B	Erreur Rouge	Erreur Vert	Erreur Bleu
435,8	100	0	255	31	0	0
467,8	0	60	255	0	-32	0
480	0	200	255	0	-2	0
508,6	0	255	35	0	0	4

	R	G	B	Erreur Rouge	Erreur Vert	Erreur Bleu
435,8	52	0	163	12	0	-36
467,8	0	46	163	0	-38	-36
480	81	101	163	32	-40	-36
508,6	3	163	17	1	-36	-3
546,1	67	255	0	-25	0	0
577	242	180	0	-1	-29	0
579,1	242	181	0	-4	-29	0
643,8	163	0	0	-36	-3	0

```
7
8 import tkinter as tk
9 from math import cos, pi, exp
10 import numpy as np
11
12 def _from_rgb(rgb):
13     r, g, b = rgb
14     return f'#{r:02x}{g:02x}{b:02x}'
15
16 def affichage(long_onde, a, d, rb, expe=False):
17     fenetre = tk.Tk()
18     canvas = tk.Canvas(fenetre, width=1000, height=120, background='black')
19     lst_couleurs = [(0, 0, 0) for i in range(1002)]
20     somme = somme2(long_onde)
21     for lo in long_onde :
22         l, i = lo
23         trous_young(lst_couleurs, l, i, a, d, rb, somme, expe)
24         trans_entier(lst_couleurs)
25     for i in range(1002) :
26         canvas.create_line(i, 0, i, 122, fill = _from_rgb(lst_couleurs[i]))
27     canvas.pack()
28     fenetre.mainloop()
29
30 def trous_young(lst, lo, i, a, d, rb, nb_lo, expe):
31     for x in range(len(lst)):
32         nlo = lo*10E-5
```

```
31     for x in range(len(lst)):
32         nlo = lo*10E-5
33         L = 2*d*nlo/rb
34         xut = x-500
35         intens = 0.5*i*(1 + cos(2*pi*a*xut/(nlo*d)))
36         intens *= exp(-(xut)*(xut)/((L/2)*(L/2)))
37         if expe :
38             r, g, b = conv_exp_lo_to_rgb(lo)
39         else :
40             r, g, b = conv_lo_to_rgb(lo)
41         lst[x] = (lst[x][0]+(intens*r)/nb_lo, lst[x][1]+(intens*g)/nb_lo, lst[x][2]+(intens*b)/nb_lo)
42
43 def trans_entier(lst):
44     for i in range(len(lst)):
45         lst[i] = int(lst[i][0]),int(lst[i][1]),int(lst[i][2])
46
47 def somme2(lst):
48     acc = 0
49     for lo in lst :
50         l, i = lo
51         acc += i
52     return acc
53
54 def ouvrir_fichier(file):
55     lst = np.loadtxt(file, delimiter = ' ')
56     return lst
```



```

55     lst = np.loadtxt(file, delimiter = ' ')
56     return lst
57
58 def conv_lo_to_rgb(lo):
59     lst = file1[lo-380]
60     return int(lst[1]*255),int(lst[2]*255),int(lst[3]*255)
61
62 def conv_exp_lo_to_rgb(lo):
63     for lst in file2 :
64         if lst[0] == lo :
65             return lst[1],lst[2],lst[3]
66
67 def f(i):
68     h = 6.62e-34
69     c = 299792458
70     nu = c/i
71     k = 1.38e-23
72     T = 5000
73     return (2*h*nu**3/c**2)/(exp(h*nu/k*T)-1)
74
75 file1 = ouvrir_fichier("Valeur theoriques.txt")
76 file2 = ouvrir_fichier("ValeursExperimentales.txt")
77 affichage([(436,1),(468,1),(480,1),(509,1),(546,1),(577,1),(579,1),(644,1)],10,10000,2,True)
78 affichage([(436,1),(468,1),(480,1),(509,1),(546,1),(577,1),(579,1),(644,1)],10,10000,2) #merc
79 affichage([(i,1) for i in range(380,780,20)],10,10000,2,False) #lumière blanche équirépartie
80 affichage([(i,f(i)) for i in range(380,780,20)],10,10000,2,False) #lumière blanche avec loi de

```