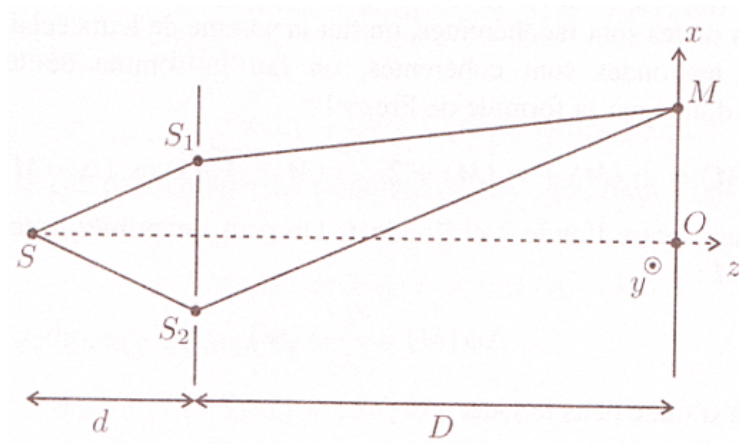


# 1 # Etude du phénomène de battement avec le doublet du sodium 1f55-1603542

On considère deux trous sources  $S_1$  et  $S_2$  identiques, distants de  $a = 1\text{ mm}$ . Les distances  $D = 2\text{ m}$  et  $d$  sont très grandes devant  $a$ . L'indice de l'air sera considéré comme égale à 1. La source de lumière est supposée ponctuelle. Soit  $M(x, y, 0)$  un point quelconque de l'écran où sont observées les interférences et tel que  $x \ll D$  et  $y \ll D$ .



1) La source  $S$  est monochromatique et de longueur d'onde  $\lambda$  et le milieu assimilé à du vide. Déterminer l'expression de l'éclairement en un point  $M$  de l'écran en précisant les hypothèses et définitions nécessaires.

Vous avez à disposition un programme permettant de tracer l'éclairement pour différentes sources.

In [11]:

```

1
2 import matplotlib.pyplot as plt
3 import numpy as np
4 import pylab as py
5
6
7 sources=[[404.7e-9],[589e-9,589.6e-9],[650e-9]]
8 D=1 # distance trous-écran
9 a=1e-3 # distance des 2 trous d'Young
10
11 def enregistrement(num_source):
12     D=1 # distance trous-écran
13     a=1e-3 # distance des 2 trous d'Young
14     if num_source==0:
15         taille=400 # Largeur en pixels de l'image
16         lim=2e-3 # Largeur maximale de l'image
17         x=np.linspace(-lim,lim,taille) # L'axe des x
18         y=x # Le même en y
19         N=len(x)
20         X,Y=np.meshgrid(x,y,indexing='xy')
21         intensite=(1+np.cos(2*np.pi*a*x/(sources[0]*D)))/2
22         fig=py.figure(figsize=[16,8],facecolor='w') # Taille globale
23         intensiteRGB=np.zeros([N,N,3])
24         intensiteRGB[:, :, 2]=intensite
25         image=py.imshow(intensiteRGB, extent = [-lim,lim,-lim,lim])
26         plt.show()
27     elif num_source==1:
28         taille=400 # Largeur en pixels de l'image
29         lim=0.5 # Largeur maximale de l'image
30         x=np.linspace(-lim,lim,taille) # L'axe des x
31         y=x # Le même en y
32         N=len(x)
33         X,Y=np.meshgrid(x,y,indexing='xy')
34         intensite=np.zeros([N,N])
35         for i in sources[num_source]: # pour toutes les longueurs d'onde
36             intensite=intensite+(1+np.cos(2*np.pi*a*x/(i*D)))/2 # on somme les intensités
37         intensite=intensite/2
38         fig=py.figure(figsize=[16,8],facecolor='w')
39         intensiteRGB=np.zeros([N,N,3])
40         intensiteRGB[:, :, 0]=intensite
41         intensiteRGB[:, :, 1]=intensite
42         image=py.imshow(intensiteRGB, extent = [-lim,lim,-lim,lim])
43         plt.show()
44     elif num_source==2:
45         taille=400 # Largeur en pixels de l'image
46         lim=6.5e-3 # Largeur maximale de l'image
47         x=np.linspace(-lim,lim,taille) # L'axe des x
48         y=x # Le même en y
49         N=len(x)
50         X,Y=np.meshgrid(x,y,indexing='xy')
51         intensite=(1+np.cos(2*np.pi*a*x/(sources[2]*D)))/2
52         fig=py.figure(figsize=[16,8],facecolor='w') # Taille globale
53         intensiteRGB=np.zeros([N,N,3])
54         intensiteRGB[:, :, 0]=intensite
55         image=py.imshow(intensiteRGB, extent = [-lim,lim,-lim,lim])
56         plt.show()
57
58     else:
59         print("ce numéro de source n'existe pas")
60

```

2) Exprimer l'interfrange et tester la validité de votre expression pour la source monochromatique rouge en utilisant le programme précédent.

3) On considère maintenant le doublet du sodium. On suppose que les deux longueurs d'onde  $\lambda_1 = \frac{c}{f_1}$  et  $\lambda_2 = \frac{c}{f_2}$  sont de même intensité avec  $f_2 - f_1 = \Delta f$ . Montrer que l'éclairement résultant peut se mettre sous la forme :

$$\mathcal{E} = 2\mathcal{E}_0 \left(1 + \cos\left(2\pi \frac{\Delta f}{c} \delta\right) \cos \Delta \phi\right)$$

4) Mettre en évidence le phénomène de battement à l'aide du programme fourni et proposer un protocole permettant d'estimer alors  $\Delta f$