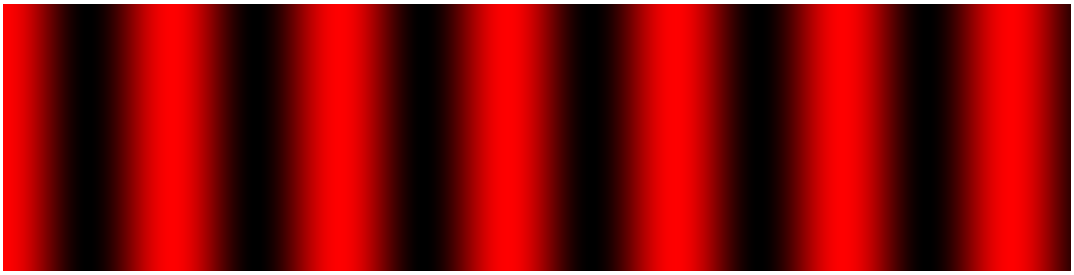
Interférences et images

Quelques exemples de l'utilisation de la bibliothèque PIL pour l'exploitation d'images. A prendre comme une piste de réflexion.

In [2]:

```
from PIL import Image
from math import cos
def monochromatique():
    img = Image.new("RGB",(400, 100))
    for j in range(100):
        for i in range(400):
            img.putpixel((i, j), (int(255*cos(i/20)**2), 0,0))
    return img

monochromatique()
```

Out [2]:



In [3]:

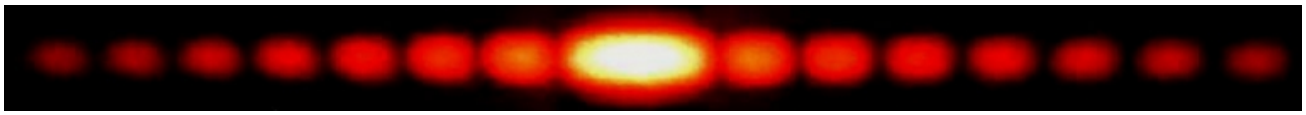
```
import matplotlib.pyplot as plt
import numpy as np

NBVALS = 200

def get_values(image, nb):
    h = image.height
    w = image.width
    out = list()
    for i in range(nb):
        pixel = image.getpixel((
            int(i*w/nb),
            int(h/2)
        ))
        out.append(pixel[0]+pixel[1]+pixel[2])
    return out
```

In [4]:

```
# Limage utilisée ici nest vraiment pas idéale (png dillustration récupéré
# en ligne - saturation - image censée être monochromatique...)
```

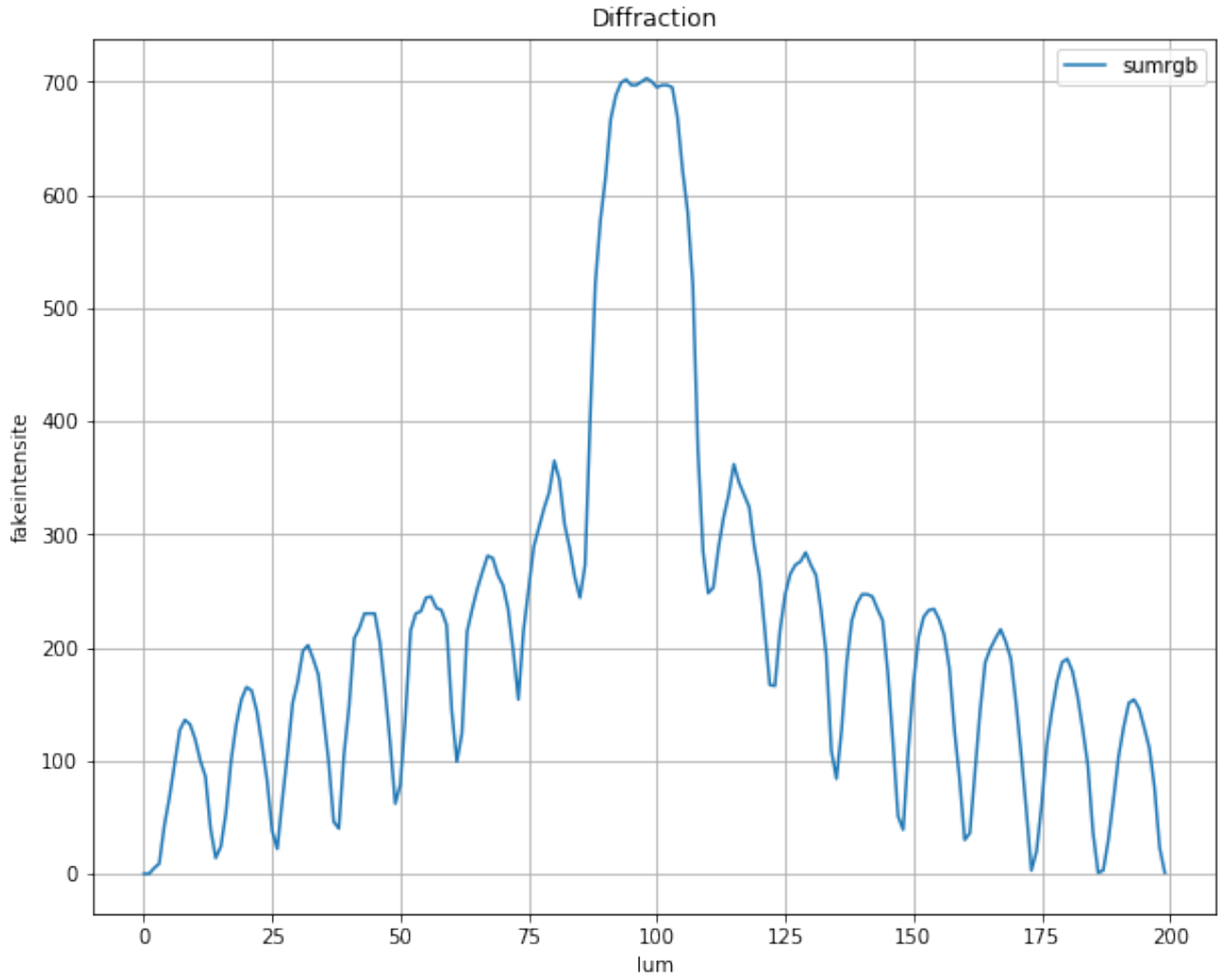
```
# bref : à revoir avec une meilleure image - mas pas eu le temps de faire la manip sorry
img = Image.open("./images/diffraction.png")
img
```

Out [4]:



In [5]:

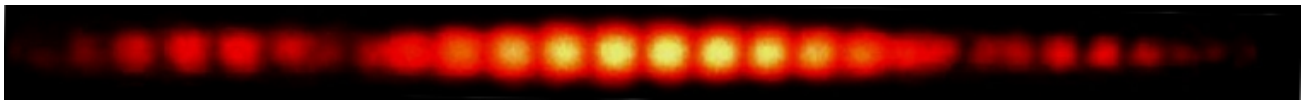
```
values = get_values(img, NBVALS)
plt.figure (figsize = (10,8))
plt.plot(range(NBVALS), values,label="sumrgb")
plt.xlabel("lum")
plt.ylabel("fakeintensite")
plt.legend()
plt.grid()
plt.title ("Diffraction")
plt.show()
```



In [6]:

```
# Limage ci-dessous nest pas dune grande qualité non plus, désolé...
img = Image.open("./images/interferences.png")
img
```

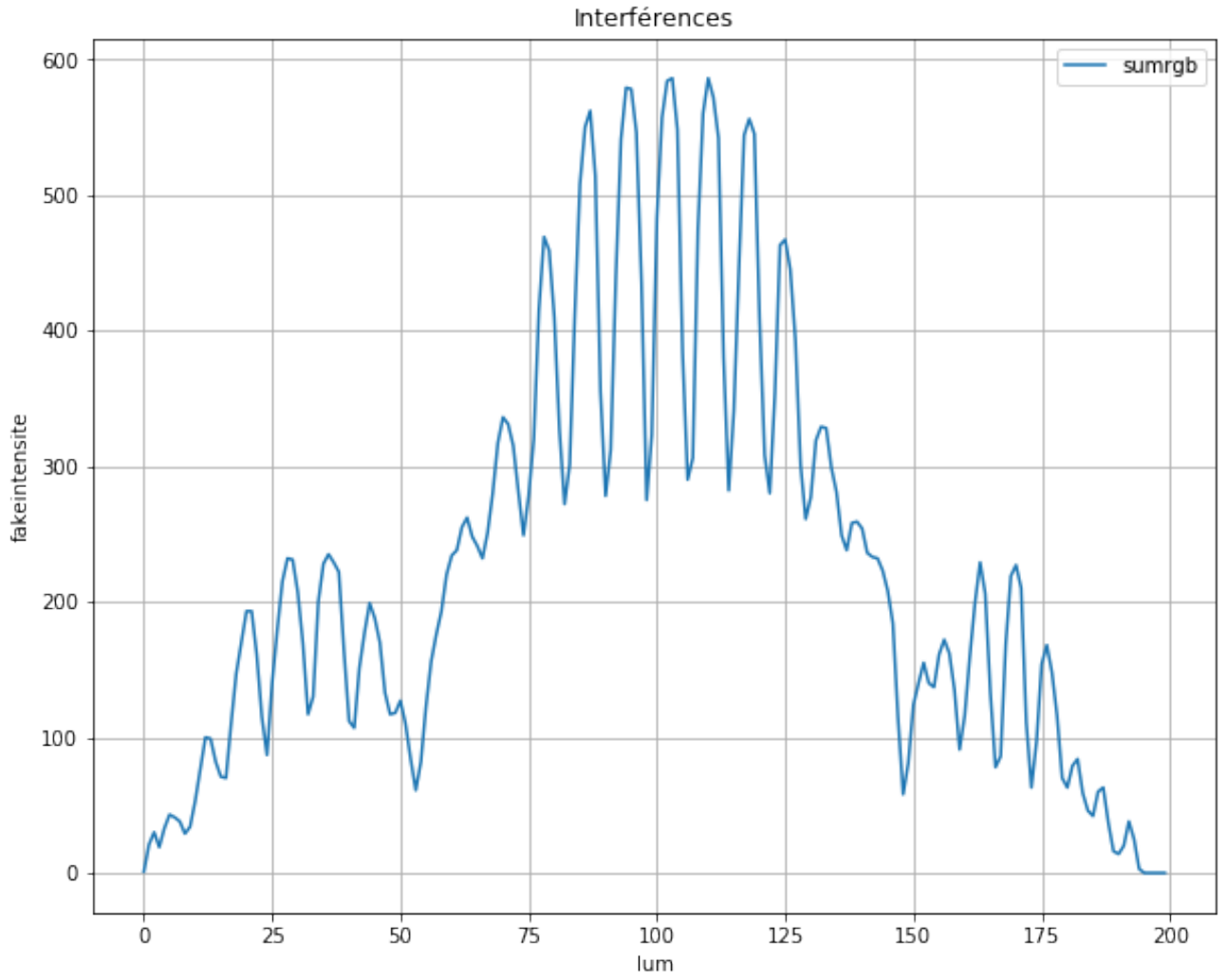
Out [6]:



In [7]:

```
values = get_values(img, NBVALS)
plt.figure (figsize = (10,8))
plt.plot(range(NBVALS), values,label="sumrgb")
plt.xlabel("lum")
plt.ylabel("fakeintensite")
```

```
plt.legend()
plt.grid()
plt.title ("Interférences")
plt.show()
```



In [40]:

```
# A RETRAVAILLER :)
```

In [ ]:

```

```