Doublet achromatique

1 Lentille simple

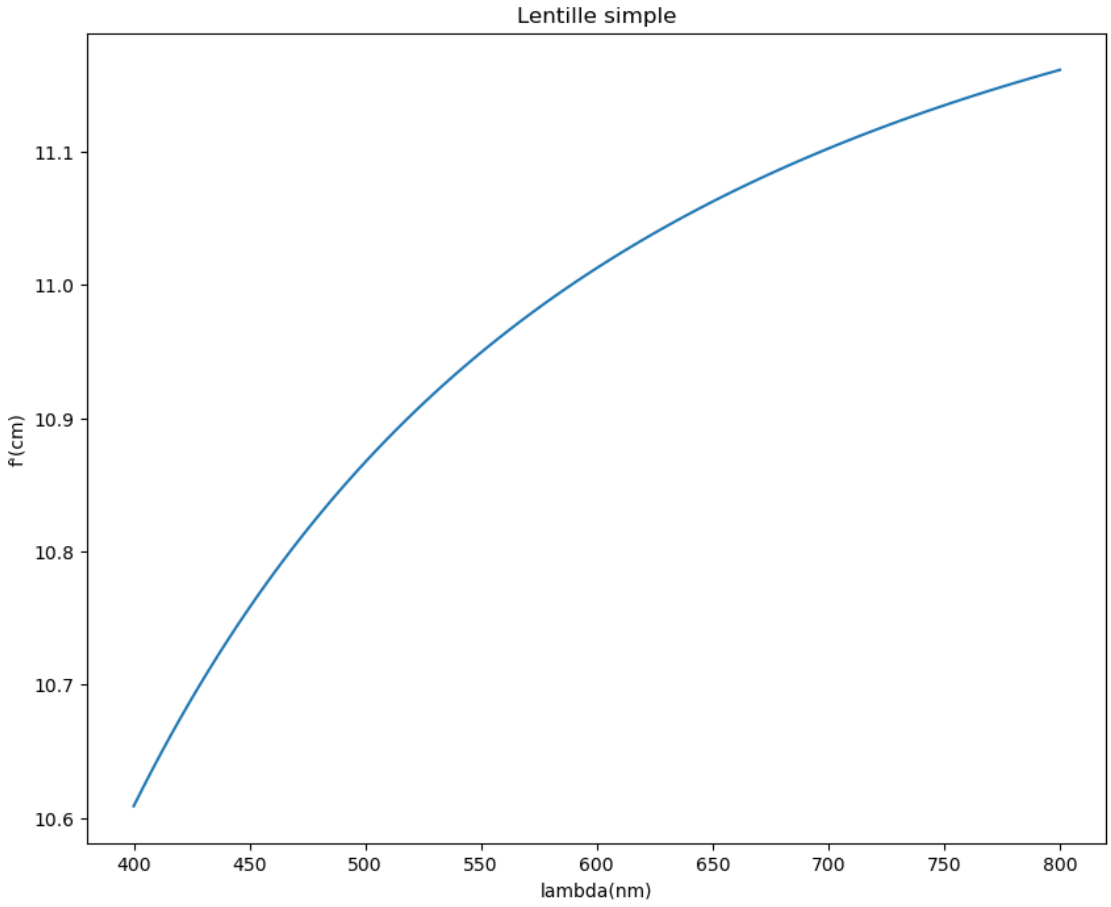
[3]:
```
def fprime(fpD,nD,n):
    """ calcule la distance focale d'une lentille à l'indice n
    connaissant sa distance focale à la raie D d'indice nD """
    return fpD*(nD-1)/(n-1)
```

[4]:
```
def nC(l):
    """ Calcule l'indice du verre Croum à la longueur d'onde l en nm """
    return 1.46+5200/l**2
def nF(l):
    """ Calcule l'indice du verre Flint à la longueur d'onde l en nm """
    return 1.62+18000/l**2
```

[5]:
```
import numpy as np
tab_l=np.linspace(400,800,100)
```

[6]:
```
tab_nC=nC(tab_l)
lD=589
nCD=nC(lD)
tab_fp=fprime(11,nCD,tab_nC)
```

[7]:
```
import matplotlib.pyplot as plt
plt.figure(figsize=(10,8))
plt.plot(tab_l,tab_fp)
plt.xlabel("lambda(nm)")
plt.ylabel("f'(cm)")
plt.title("Lentille simple")
plt.show()
```



[8]:
```
print(max(tab_fp)-min(tab_fp))
```

```
0.5523988869046015
```

1.1 Doublet achromatique

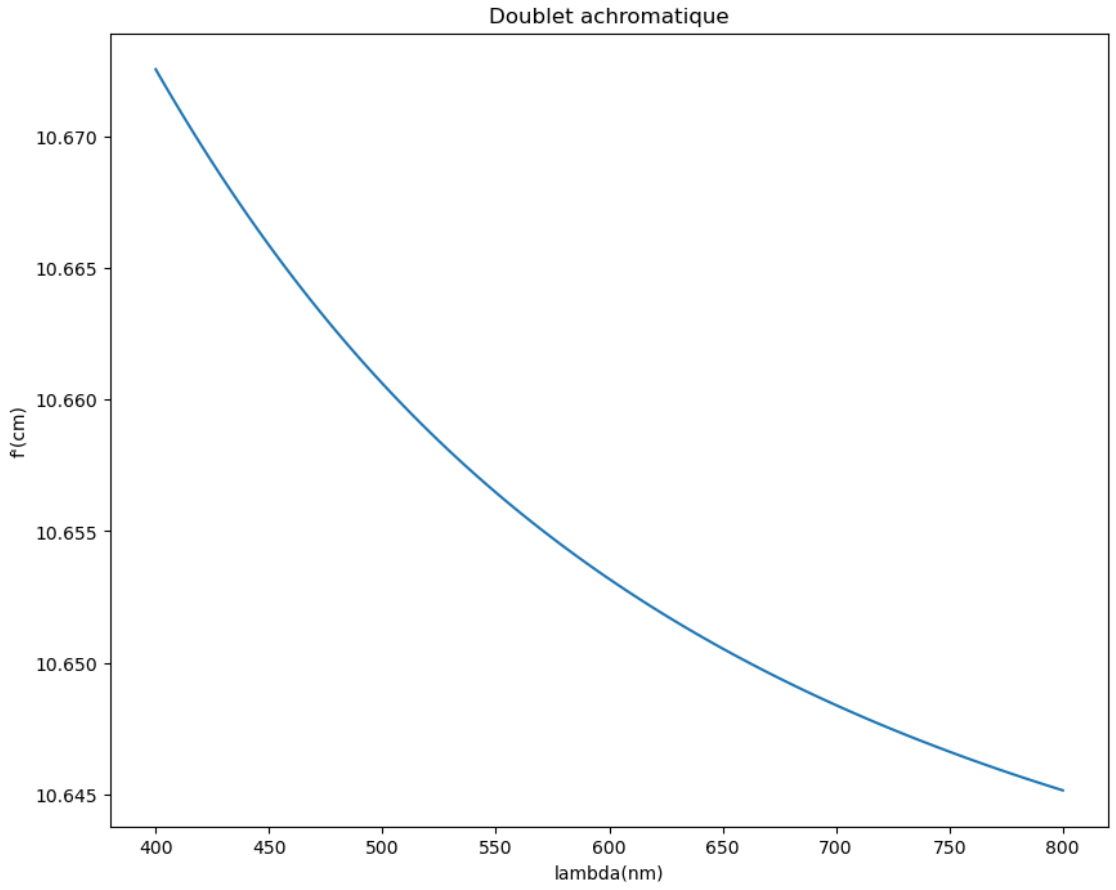
[12]:
```
import numpy as np
tab_l=np.linspace(400,800,100)
tab_nC=nC(tab_l)
tab_nF=nF(tab_l)
lD=589
tab_fp1=fprime(6,nC(lD),tab_nC)
tab_fp2=fprime(-12,nF(lD),tab_nF)
```

[13]:
```
def xImg(xObj,xLen,fp):
    """ calcule la position de l'objet, connaissant
    la position de l'objet (xObj)
    la position de la lentille (xLen)
    la focale de la lentille (fp) """
```

```
    return xLen+((xObj-xLen)*fp)/(xObj-xLen+fp)
```

[16]:
```
tab_x1=tab_fp1
tab_x2=xImg(tab_x1,0.5,tab_fp2)
```

[17]:
```
plt.figure(figsize=(10,8))
plt.plot(tab_l,tab_x2)
plt.xlabel("lambda(nm)")
plt.ylabel("f'(cm)")
plt.title("Doublet achromatique")
plt.show()
```



[18]:
```
print(max(tab_x2)-min(tab_x2))
```

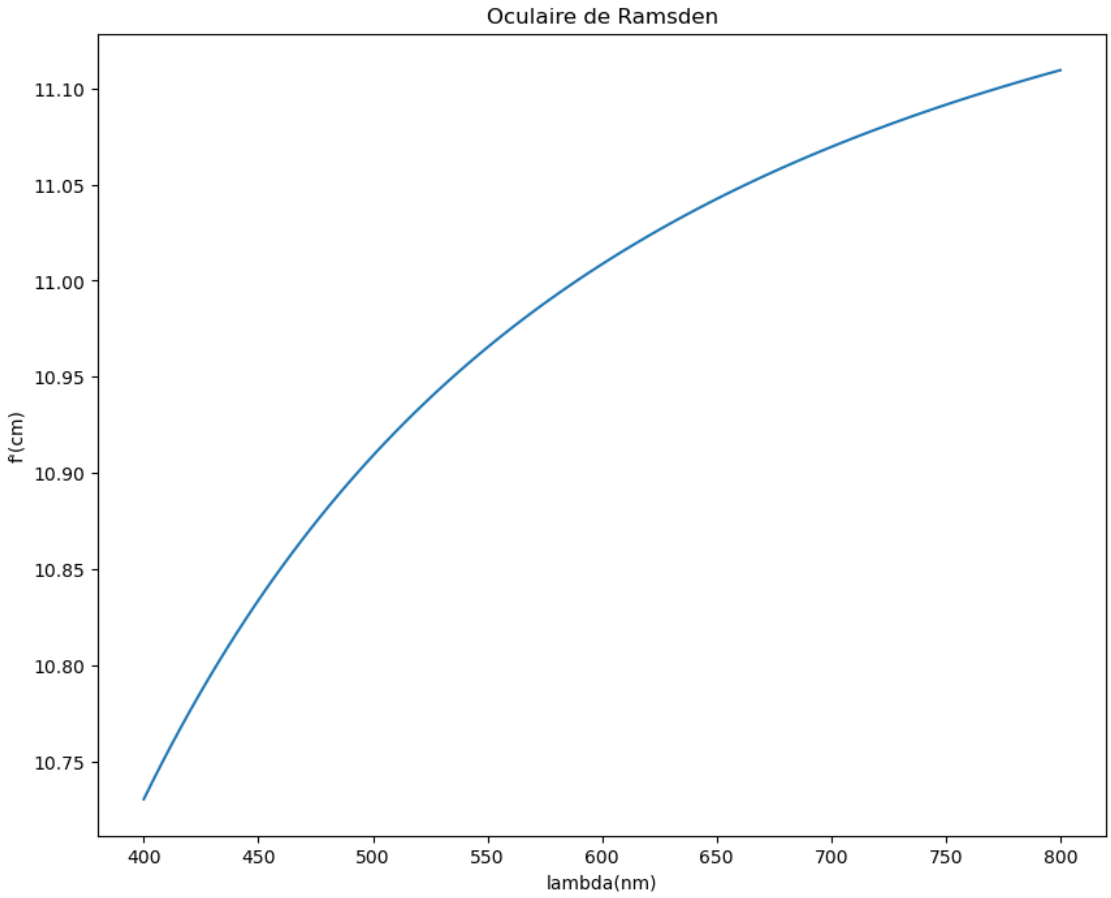
```
0.027406106439107347
```

1.2 Oculaire de Ramsden

[19]:
```
import numpy as np
tab_l=np.linspace(400,800,100)
tab_nC=nC(tab_l)
tab_nF=nF(tab_l)
lD=589
tab_fp1=fprime(12,nC(lD),tab_nC)
tab_fp2=fprime(12,nC(lD),tab_nC)
```

[20]:
```
tab_x1=tab_fp1
tab_x2=xImg(tab_x1,8,tab_fp2)
```

[21]:
```
plt.figure(figsize=(10,8))
plt.plot(tab_l,tab_x2)
plt.xlabel("lambda(nm)")
plt.ylabel("f'(cm)")
plt.title("Oculaire de Ramsden")
plt.show()
```



[22]:
```
print(max(tab_x2)-min(tab_x2))
```

```
0.3791669276897327
```

[ ]:
```

```