LP30_Laser

November 28, 2021

[1]:
```
import os
from pathlib import Path

DataFolder="C:/Users/julie/OneDrive/Bureau/Lessons/LP30_Laser"

os.chdir(Path(DataFolder))
```

[2]:
```
import math
import numpy as np
import scipy as sc
import scipy.special as sp
import scipy.constants as pyc
import scipy.optimize as scopt

import matplotlib
#%matplotlib notebook
import matplotlib.pyplot as plt
from mpl_toolkits.mplot3d import Axes3D
```

0.0.1 paramètre laser

On utilise un laser He/Ne

[3]:
```
lambda_laser=632.8e-9
```

0.0.2 paramètre figures

[4]:
```
figsize=(10,8)        #taille des figures
fontsize=20           #taille de police
colordata='steelblue'#couleur data
colorfit='darkred'    #couleur pour les fits
```

1 Expression de l'intensité et de l'extension radiale

[5]:
```
def I(r,I0,w,r0):
    'Intensité en fonction de r la coordonnée radiale'
    return(I0*np.exp(-2*((r-r0)**2/w**2)))

def w(z,w0,z0):
    'équation de w en fonction de z'
    return(w0*(1+(z**2/z0**2))**0.5)
```

2 Importation données

2.0.1 Tableaux pour premiers tests

[6]:
```
x163=np.genfromtxt("x163.txt",delimiter=';')
```

2.0.2 Array contenant toutes les données

[7]:
```
'Tableau contenant la partie du nom qui change dans les fichiers'

z_string=np.
 ↪array(['154p5','155','156','157','158','159','160','161','162','163','163p5','164','164p2','1
```

```
'Création du tableau'

BigData=np.zeros((np.size(z_string),2048,2))
print(np.shape(BigData[:,0,:]))
file="x{}.txt".format(z_string[0])
for i in range(np.size(z_string)):
    file="x{}.txt".format(z_string[i])
    BigData[i,:,:]=np.genfromtxt(file,delimiter=';')[:,1:]
```

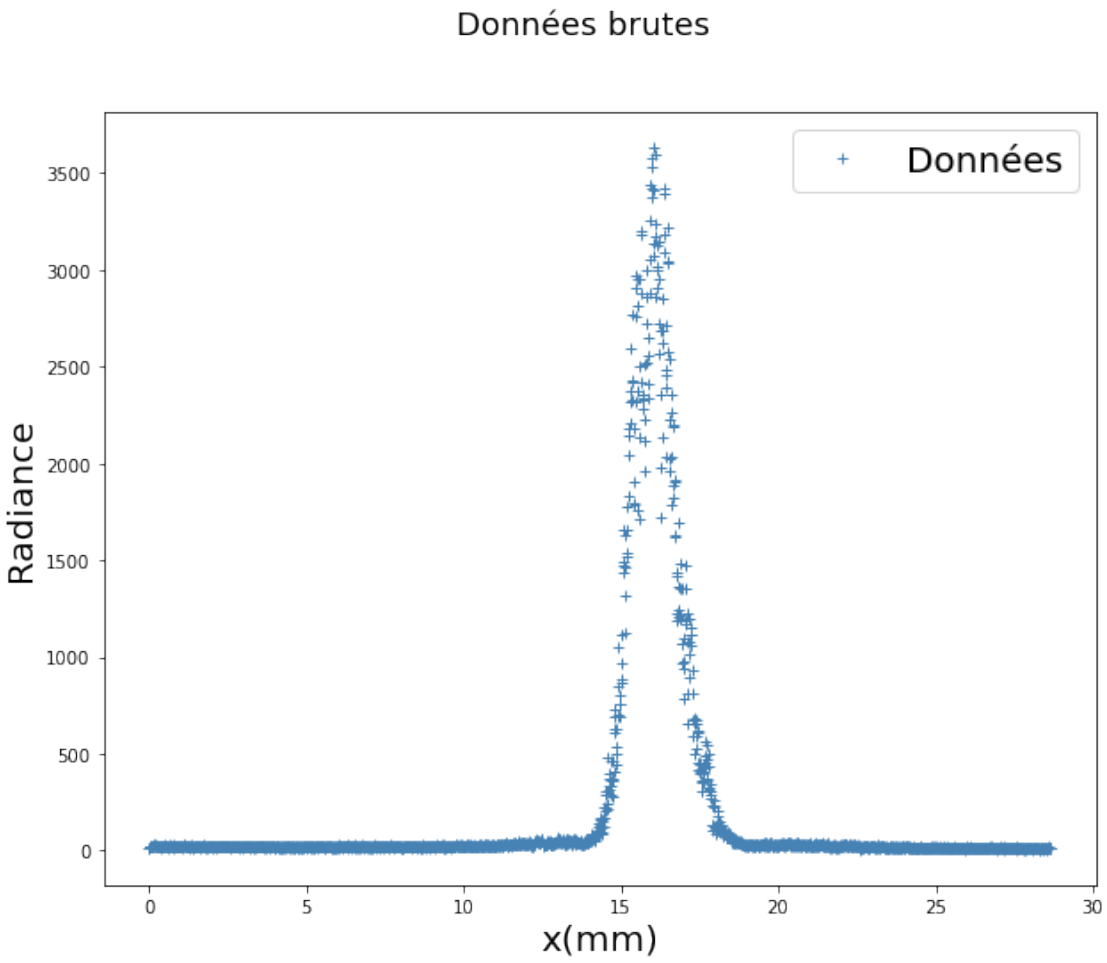
```
(21, 2)
```

3 Premiers tracés

[8]:
```
'indice dans BigData'

i=9
#%matplotlib notebook
```

[9]:
```
linestyle='None'
Data=BigData[i,:,:]
```

```
title='Données brutes'
marker='+'
xlabel='x(mm)'
ylabel='Radiance'
fig,ax=plt.subplots(num=title, figsize=figsize)
im=ax.plot(Data[:,0],Data[:,1],marker=marker,␣
 ↪linestyle=linestyle,label='Données',color=colordata)

fig.suptitle(title,fontsize=18)
plt.xlabel(xlabel,fontsize=fontsize)
plt.ylabel(ylabel,fontsize=fontsize)
ax.legend(fontsize=fontsize)
plt.show()
```

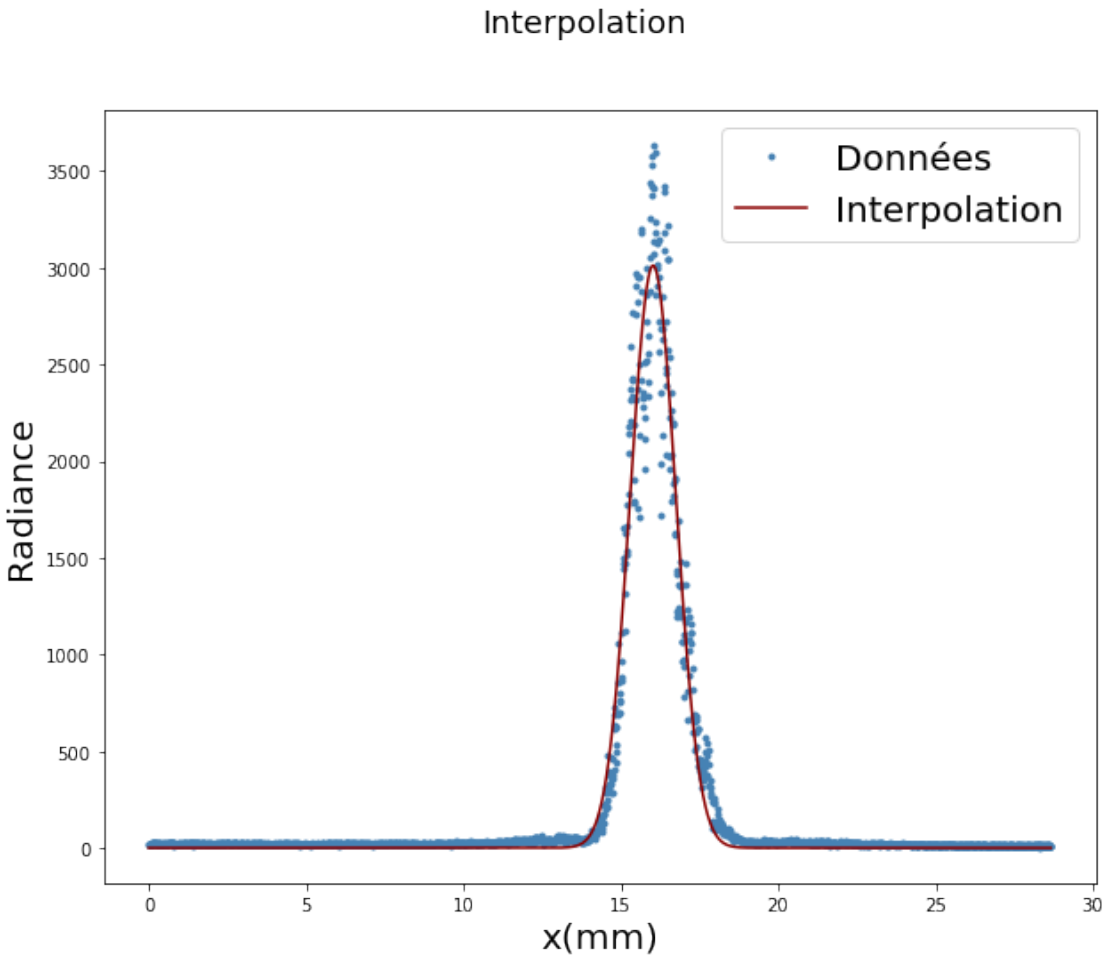


[10]:
```
'Interpolation'

p0=np.array([3000,1,15])
```

```
a=scopt.curve_fit(I,Data[:,0],Data[:,1])
print(a)
```

```
(array([3.01176438e+03, 1.44774918e+00, 1.60238307e+01]), array([[
2.19707164e+02, -7.04088500e-02,  1.15627961e-08],
       [-7.04088500e-02,  6.76910957e-05, -5.54417867e-12],
       [ 1.15627961e-08, -5.54417867e-12,  1.69227735e-05]]))
```

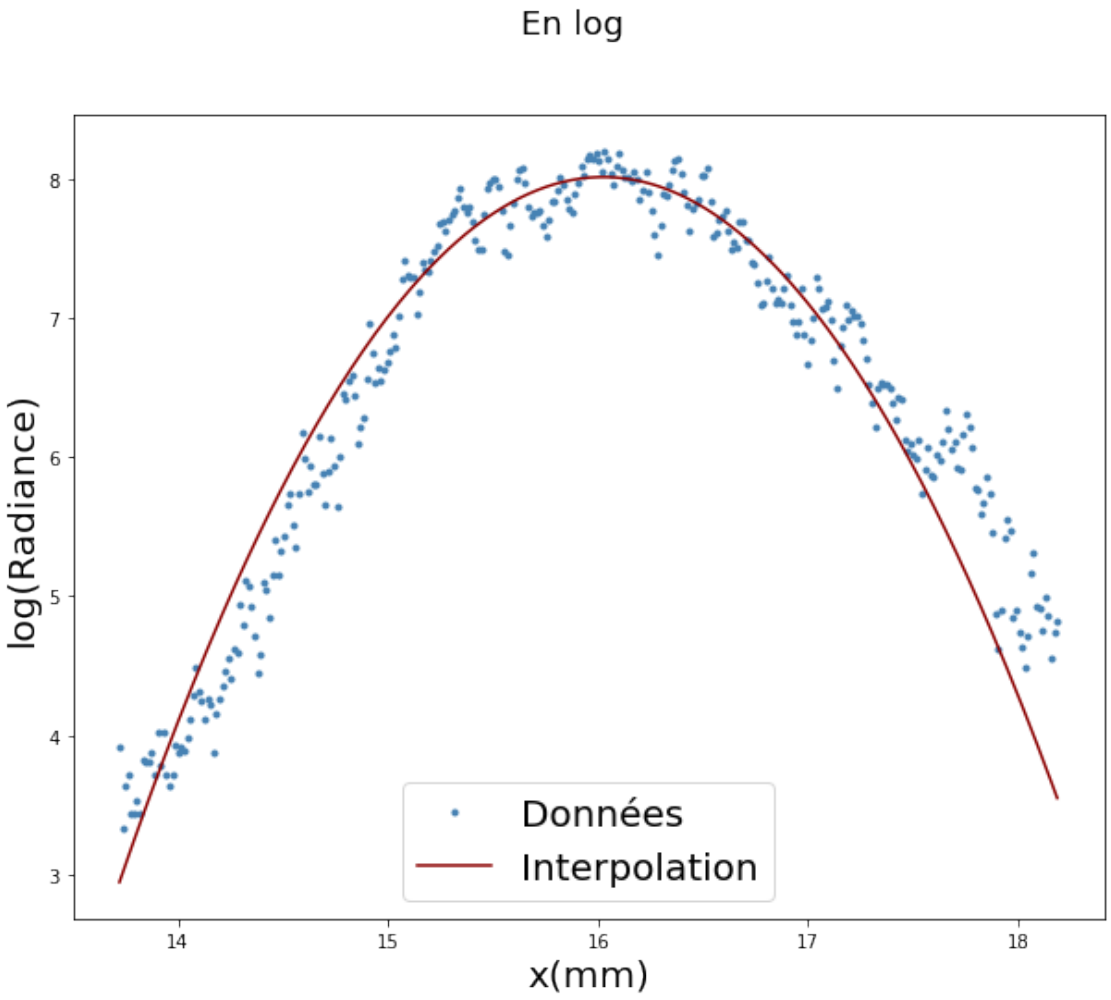
[11]:
```
x=Data[:,0]
y=I(x,a[0][0],a[0][1],a[0][2])
Fit=np.zeros((np.size(x),np.size(x)))
Fit[:,0]=x
Fit[:,1]=y
title='Interpolation'
fig,ax=plt.subplots(num=title, figsize=figsize)
im=ax.plot(Data[:,0],Data[:,1],marker='.', linestyle='␣
 ↪',color=colordata,label='Données')
im_fit=ax.plot(Fit[:,0],Fit[:,1], linestyle='-',c=colorfit,␣
 ↪label='Interpolation')

fig.suptitle(title,fontsize=18)
plt.xlabel(xlabel,fontsize=fontsize)
plt.ylabel(ylabel,fontsize=fontsize)
ax.legend(fontsize=fontsize)
plt.show()
```



Pour être sûr qu'on a bien la forme de la gaussienne et pas la diffraction on trace en log : (dans l'idéal il faudrait faire tout le raisonnement en log)

[12]:
```
x=Data[:,0]
y=I(x,a[0][0],a[0][1],a[0][2])
Fit=np.zeros((np.size(x),np.size(x)))
Fit[:,0]=x
Fit[:,1]=y
p=980
q=1300

title='En log'
fig,ax=plt.subplots(num=title, figsize=figsize)
im=ax.plot(Data[p:q,0],np.log(Data[p:q,1]),marker='.', linestyle='␣
 ↪',color=colordata,label='Données')
im_fit=ax.plot(Fit[p:q,0],np.log(Fit[p:q,1]),␣
 ↪linestyle='-',c=colorfit,label='Interpolation')
```

```
fig.suptitle(title,fontsize=18)
plt.xlabel(xlabel,fontsize=fontsize)
plt.ylabel('log(Radiance)',fontsize=fontsize)
ax.legend(fontsize=fontsize)
plt.show()
```



4 Forme du faisceau

4.0.1 Fonction

[13]:
```
def waist(BigData,zi):
    'Fonction qui récupère la valeur de w pour tous les z'
    W=np.zeros((np.size(zi)))
    W=[]
```

```
    for i in range(np.size(BigData[:,0,0])):
        Data=BigData[i,:,:]
        p0=np.array([np.max(Data[:,1]),1,15])
        w=scopt.curve_fit(I,Data[:,0],Data[:,1],p0)[0][1]
        W.append(w)
    return(np.array(W))
```

4.0.2 Tracé

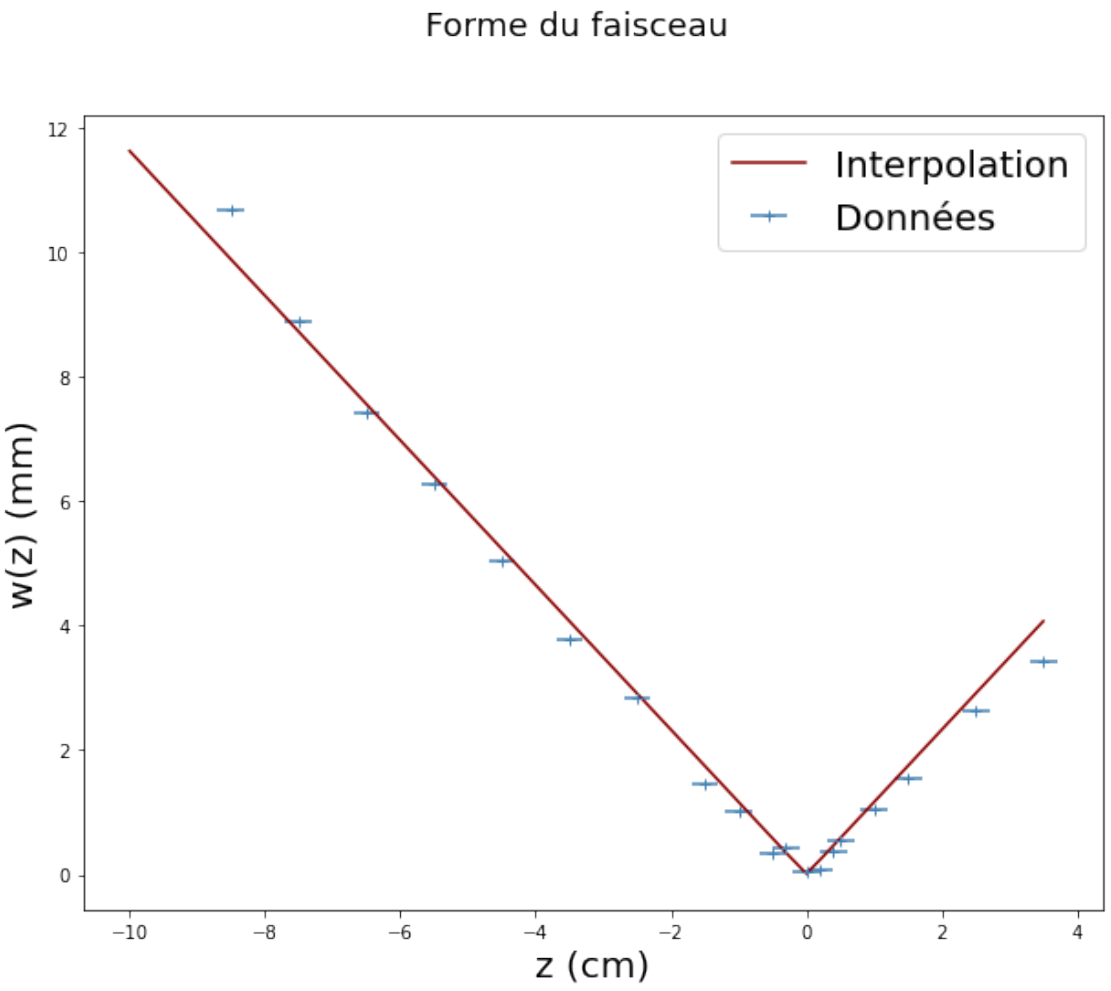
[14]:
```
'Tableau de tous les z'
zi=np.array([154.5,155,156,157,158,159,160,161,162,163,163.5,164,164.2,164.5,164.
 ↪7,164.9,165,165.5,166,167,168])
W=waist(BigData,zi)
#print(W)

'Récupération du minimum, cad du waist'
j=np.argmin(W)
z=zi-zi[j]
w0=W[j]
zr=np.sqrt(w0**2*np.pi/lambda_laser)
print('Le waist est w0={}mm et zr={} mm'.format(np.round(w0,3),np.round(zr,4)))
#print(zi[j])
```

```
Le waist est w0=0.043mm et zr=94.833 mm
```

[15]:
```
p=2
q=np.size(z)

a=scopt.curve_fit(w,z[p:q],W[p:q])
print(a)
title='Forme du faisceau'
Fit=np.zeros((np.size(z),2))
Fit[:,0]=z
Fit[:,1]=w(z,a[0][0],a[0][1])
fig,ax=plt.subplots(num=title, figsize=figsize)
im=ax.errorbar(z[p:q],W[p:q],xerr=0.
 ↪2,marker='+',linestyle='None',color=colordata,label='Données')
plt.xlabel('z (cm)',fontsize=fontsize)
plt.ylabel('w(z) (mm)',fontsize=fontsize)
fig.suptitle(title,fontsize=18)
im_fit=ax.plot(Fit[:,0],Fit[:,1], linestyle='-',c=colorfit,label='Interpolation')
ax.legend(fontsize=fontsize)
```

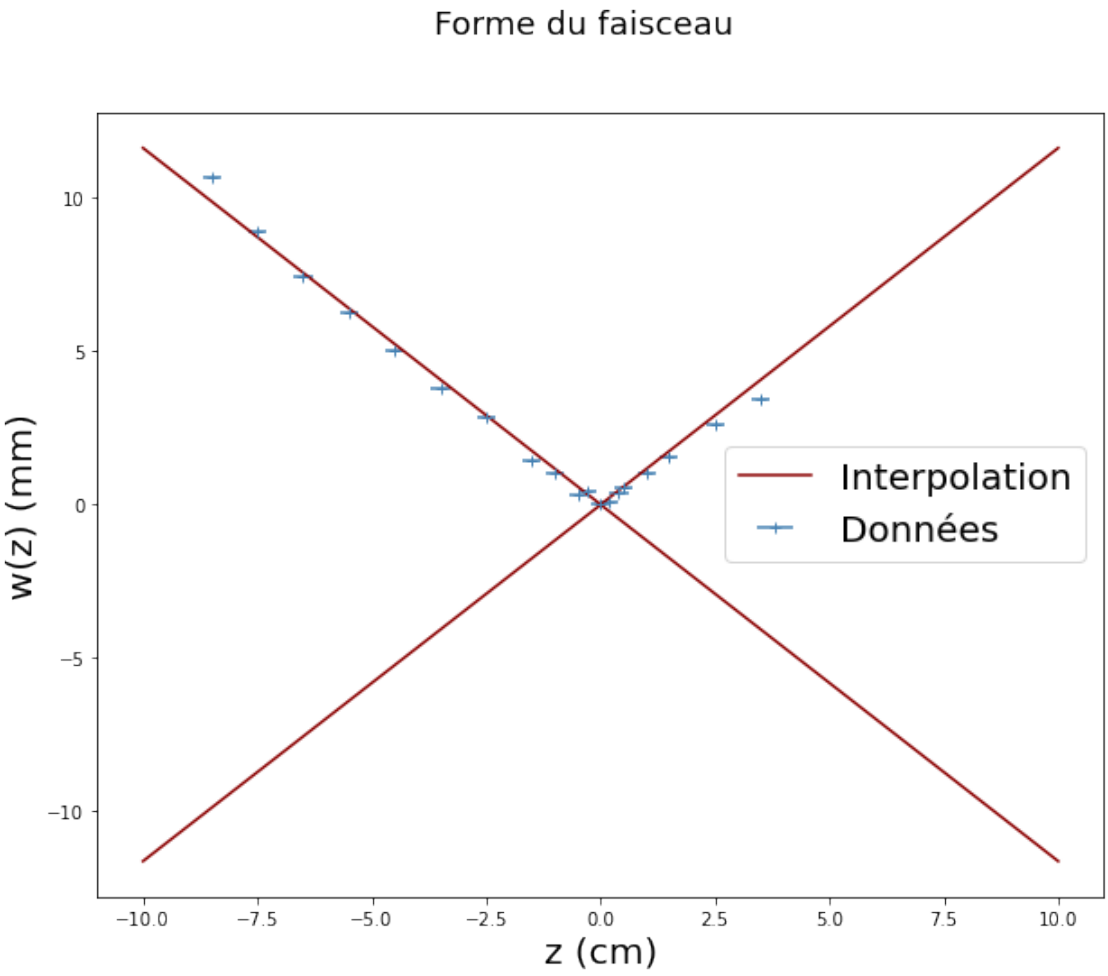
```
(array([0.01372679, 0.01180689]), array([[0.09221196, 0.07931548],
       [0.07931548, 0.06822269]]))
```

[15]:
```
<matplotlib.legend.Legend at 0x1b5eaba2eb0>
```



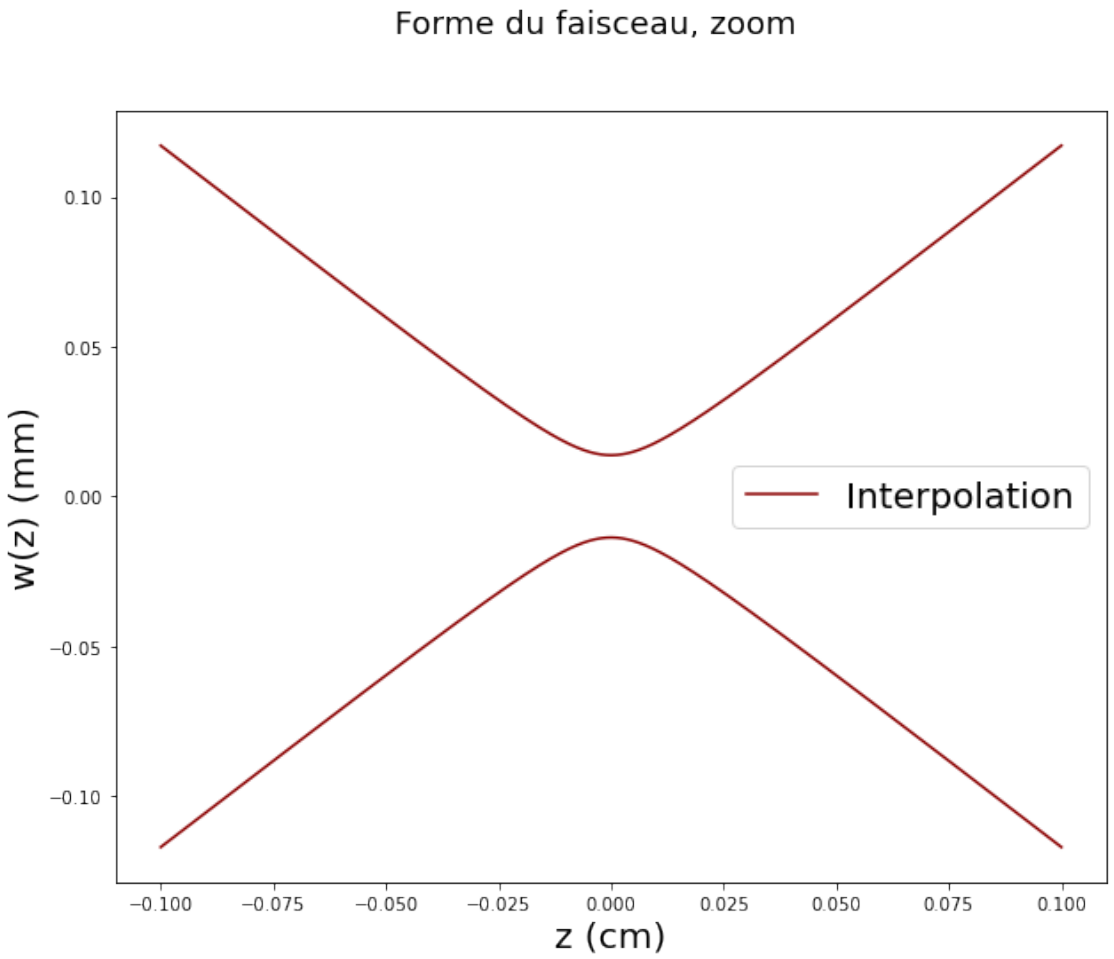
[16]:
```
fig,ax=plt.subplots(num=title, figsize=figsize)
zmod=np.linspace(-10,10,1000)
Fit=np.zeros((1000,2))
Fit[:,0]=zmod
Fit[:,1]=w(zmod,a[0][0],a[0][1])
im=ax.errorbar(z[p:q],W[p:q],xerr=0.
 ↪2,marker='+',linestyle='None',c=colordata,label='Données')

title='Forme du faisceau'
im_fit=ax.plot(Fit[:,0],Fit[:,1], linestyle='-',c=colorfit)
im_fit=ax.plot(Fit[:,0],-Fit[:,1],␣
 ↪linestyle='-',c=colorfit,label='Interpolation')
plt.xlabel('z (cm)',fontsize=fontsize)
plt.ylabel('w(z) (mm)',fontsize=fontsize)
fig.suptitle(title,fontsize=18)
ax.legend(fontsize=fontsize)
```

```
plt.show()
```



[17]:
```
title='Forme du faisceau, zoom'
fig,ax=plt.subplots(num=title, figsize=figsize)
zmod=np.linspace(-0.1,0.1,1000)
Fit=np.zeros((1000,2))
Fit[:,0]=zmod
Fit[:,1]=w(zmod,a[0][0],a[0][1])
im_fit=ax.plot(Fit[:,0],Fit[:,1], linestyle='-',c=colorfit)
im_fit=ax.plot(Fit[:,0],-Fit[:,1],␣
 ↪linestyle='-',c=colorfit,label='Interpolation')

plt.xlabel('z (cm)',fontsize=fontsize)
plt.ylabel('w(z) (mm)',fontsize=fontsize)
fig.suptitle(title,fontsize=18)
ax.legend(fontsize=fontsize)
```

```
plt.show()
```



[18]:
```
print('Le waist est w0={}mm et zr={} mm'.format(np.round(w0,3),np.round(zr,4)))
```

```
Le waist est w0=0.043mm et zr=94.833 mm
```