Revision of "Measuring the beam profile"

Corinna Wegner, Garen Gregorian

August 22, 2022

1 Abstract

Warum neu gemacht

2 Measuring the beam profile

In this experiment we measured the intensity of the laser light that is emitted by the fibre. We cut off some of the beam with a razor blade to see how much voltage is still measured by the photodiode. Thereby we obtained a profile of the beam cross section. After that we put a lens ($f = 100mm$) behind the fibre end so that the beam was focused at the focal point. We measured the profile of the beam at different positions between the two lenses (the second lens ($f = 50mm$) focuses the beam into the photodiode). Near the focal point of the first lens, where the waist of the gaussian beam lies, we measured three times. To eliminate fluctuations, we normalized the voltage signal with the other photodiode.

The total power the photodiode is detecting depends on the position of the razor blade x and is given by:

$$P(x) = \int_x^\infty \mathrm{d}x' I_0 \mathrm{e}^{-\frac{2(x'-x_0)^2}{\omega^2}}. \tag{1}$$

Measuring the cross section profile without focusing the beam To calculate the beam radius $\omega(z)$ we fitted the data of our measurement a (see appendix) to eq. 1. In order to do this we calculated the power from the voltage using eq. **??**.

The output parameters of the fit are:
$8,3cm : I_0 : 0.008777106036129446 Strahltaille : 0.8825862418640391 11cm : I_0 : 0.007991921587422922 Strahltaille : 1.0121807477924682 5,0cm : I_0 : 0.010111526698723994 Strahltaille : 0.7755253848409015 waist : (0.8290558133524704 + -0.053530428511568806) mm rayleighlength : (3.4265554046505233 + -0.4406541724660673) mm$

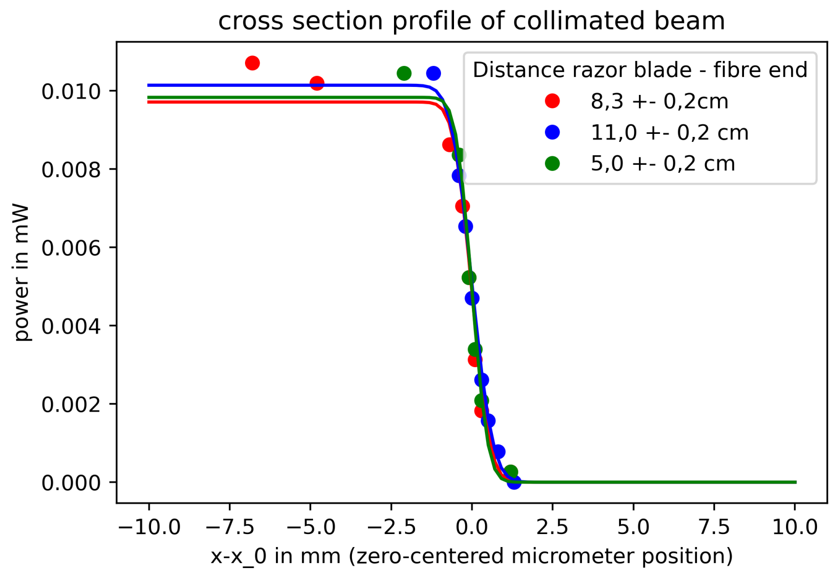

Figure 1: cross section profile of collimated beam

Label	I_0	waist
$8.3cm$		
$11.0cm$		
$5.0cm$		

Überarbeiten, neu interpretieren Neue Werte eingeben The average waist is then $(\pm)mm$. From that we also calculated the rayleigh length

$$z_R = \frac{\pi \omega_0^2 n}{\lambda} \tag{2}$$

at which the beam radius is $\sqrt{2}\omega_0$. In our case, $n = 1$ is the refraction index of the medium (air) and the wavelength of the laser is $\lambda = 632.8nm$. The resulting rayleigh length is: $(\pm)mm$.

Überarbeiten The first two measurements of the series with Distance razor blade - fibre end $= 8.3cm$ seemed to fall out of line. When doing the fit, the curve (red dashed) also looked inappropiate. Presumably we measured these points too early, namely not at the point right before the power falls off (i.e. the maximum). Therefore we decided to leave them out of the fitting, which lead to a much better result (see figure 1).

The relatively high errors can be explained by the strong fluctuations of the multimeter display, making it hard to measure the voltage. This can be seen when looking at the next figure, where the measurement points differ quite a lot from the fit curves. However, one can see that the shapes of the fit curves look similar. Another problem of the experiment was that there could be scattering light from the ambient or laser, which influences the photodiode. Furthermore the razor did not really fit the radial intensity profile of the beam. It only cuts off the beam from one direction, leaving the other direction always open. Therefore there is a systematic error in the experiment. Using an apperture would have been a better way to measure the beam profile. Also, in order to measure at certain distances from the fibre end, we had to put the razor mount into a place between the threads where you can fix the mount on the table. This could lead to a non orthogonal angle between razor and beam, which would mean that you have to turn the micrometer screw more to get the same decrease of intensity. Besides, at the edge of the razor there is diffraction happening, as we have examined in a previous practical. This could also skew the measurements. Finally it was hard to measure the distances in the z-direction having only a ruler. The experiment was very barred by the coupler and other optical instruments, so a caliper would have helped to increase the quality of those measurements.

Measuring the cross section profile with lens ($f = 100mm$) We measured the beam profile using the razor blade technique at seven distances from the lens. Three measurements were taken near the focal point because here we expected the waist ω_0, i.e. the minimum of the beam radius $\omega(z)$. They are related by the equation:

$$\omega(z) = \omega_0 \sqrt{1 + \frac{z^2}{z_R^2}} \tag{3}$$

The razorblade positions z from which we measured the beam profile are $-(7.0 \pm 0.2)cm$, $-(4.0 \pm 0.2)cm$, $-(1.0 \pm 0.2)cm$, $(0.0 \pm 0.2)cm$, $(1.0 \pm 0.2)cm$, $(4.3 \pm 0.2)cm$ and $(7.0 \pm 0.2)cm$, where $z = 0$ is the focal point. For illustration purposes we plotted the data near focal point in an extra plot. The plots show the calculated powers (equation **??**) from the measuring data along with the corresponding fit to the gaussian integral (equation 1). From the fit we obtained I_0 and $\omega(z)$:

$Rot : I_0 : 0.0314677004624013 Strahltaille : 0.21386318820116362 Blau : I_0 : 0.07270445083092966 Strahltaille : 0.0774080737868041 Grun : I_0 : 0.029874937606379274 Strahltaille : 0.15819545705252225 Gelb : I_0 : 0.015763387087890192 Strahltaille : 0.37413622790241946 turkis : I_0 : 0.06136075469487003 Strahltaille : 0.08303635608624456 magenta : I_0 : 0.06379644469561196 Strahltaille : 0.08349636759598762 schwarz : I_0 : 0.004440113074039632 Strahltaille : 1.549232366989364$

Label	I_0	waist
$-7.0cm$		
$-4.0cm$		
$-1.0cm$		
$0.0cm$		
$1.0cm$		
$4.3cm$		
$7.0cm$		

Überarbeiten, interpretieren, warum ein Wert extrem...

Again, we fitted the measurement series for each razor-lens-distance z to eq. 1 and obtained the local $\omega(z)$. Then we fitted these together with the corresponding z to eq. 3. From that we obtained the waist
$\omega_0 = (1 \pm 6)mm$
The resulting rayleigh length is
$z_R = (1 \pm 7)mm$

y-Achsenbeschriftung noch von W in mW ändern oder so ähnlich

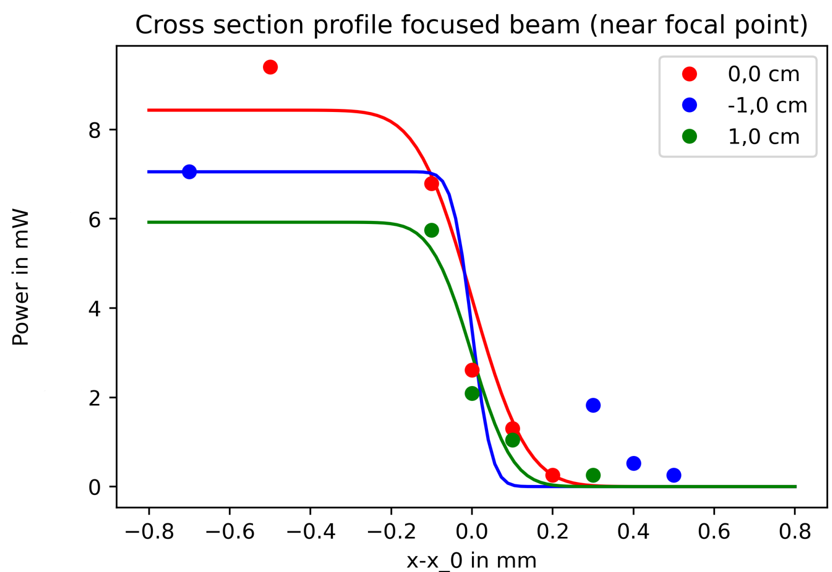

Figure 2: Cross section profile focused beam (near focal point)

Überarbeiten As in the previous part, we see that the standard deviations of ω_0 and z_R are relatively high. This time for the rayleigh length is is even higher than the value itself, which is illogical because it includes negative values within the error range. Since the experiments are very similar, the discussed error sources apply as well here. Furthermore, we observed extreme fluctuations on

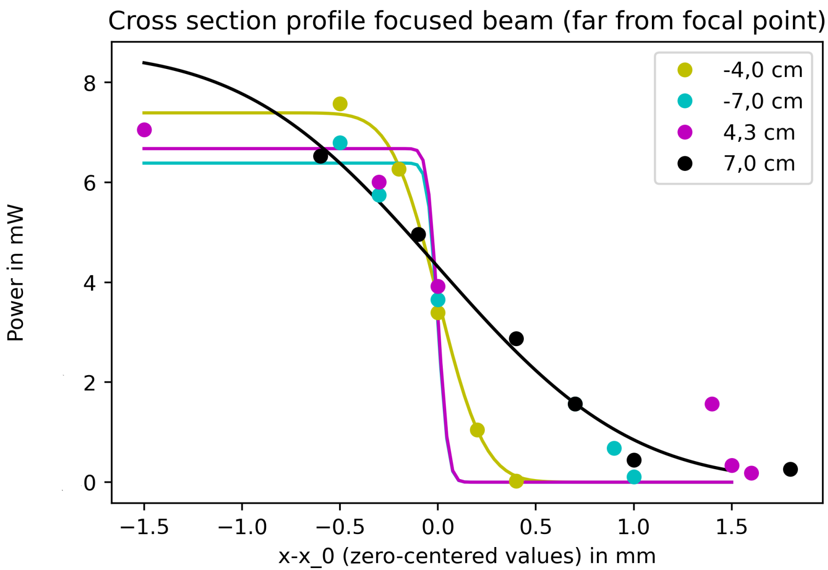

Figure 3: Cross section profile focused beam (far from focal point)

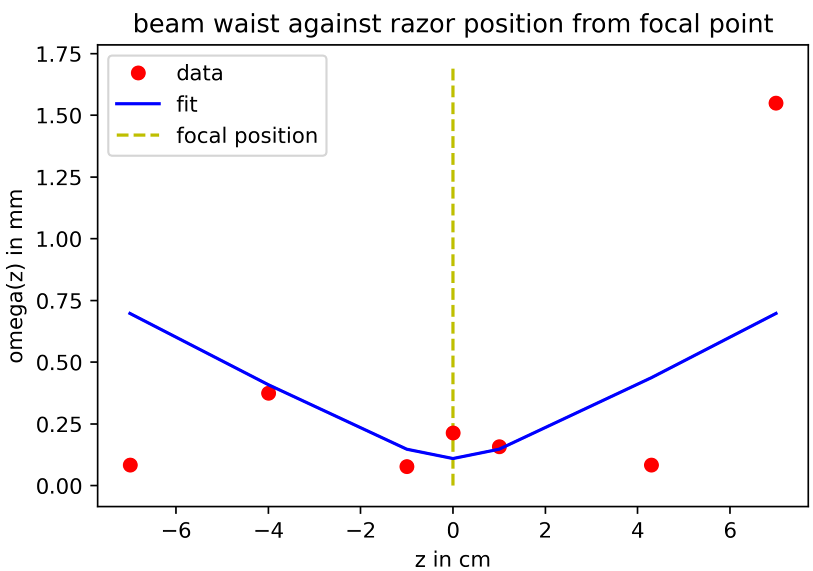

Figure 4: Beam waist against razor position from focal point

the multimeter display especially when measuring near the focal point. A reason for this could be that the waist is so small such that even the tiniest micrometer screw turn makes a lot of difference. In fact, sometimes only contact with the table changed the displayed value by about 20 mV. To avoid abberations we made sure the beam enters the plano-convex lens on its curved surface, such that the refraction takes place on both surfaces of the lens. Nevertheless we could only approximately determine the center of the lens and thereby abberations can not be excluded totally.

3 Appendix

Measuring the beam profile (a)

```
# -*- coding: utf-8 -*-
"""
Created on Thu Jun  9 22:27:53 2022

@author: corin
"""

import numpy as np
import matplotlib.pyplot as plt
import pandas as pd
from scipy.optimize import curve_fit
import scipy.integrate as integrate
from functools import partial #https://stackoverflow.com/questions/61675014/inte

csv_path_a = r"C:/Users/corin/Gausssche-Strahlenoptik/Strahlprofil_a.csv"
csv_path_b = r"C:/Users/corin/Gausssche-Strahlenoptik/Strahlprofil_b.csv" # Vie

a_72 = pd.read_csv(csv_path_a, delimiter=";", header=None, skiprows=3, nrows=2,
a_45 = pd.read_csv(csv_path_a, delimiter=";", header=None, skiprows=6, nrows=2,
a_105 = pd.read_csv(csv_path_a, delimiter=";", header=None, skiprows=9, nrows=2,
data_a = [a_45, a_72, a_105]

h = 6.62607015*10**(-34)
c = 299792458
wvl = 632.8*10**(-9)
e = 1.602176634*10**(-19)
distLS = 15.5 #Distance lens-fibre end (Source of laser beam)
Rd = 10**4 #Resistance of the photodiode in ohm
x = np.linspace(-10,10, 100)

#the positions must be arranged symmetrically around zero (zero-centered):
v1 = 12.8 #Verschiebung für rot
```

```
v2 = 1.2 #Verschiebung für blau
v3 = 6.9 # Verschiebung für grün
#normalized micrometer positions:
x72 = a_72.values[0]-v1
x45 = a_45.values[0]-v2
x105 = a_105.values[0]-v3

maxints = []
omegas = []

def P(U, R):
    return (h*c*U)/(wvl*R*e*0.75)

def gaussint(x, I0, w):
    inner = lambda xp: np.exp((-2*xp**2)/(w**2))
    #integrate.quad kann keine Integrationsgrenzen als Variablen haben, darum ko
    integral = np.array(list(map(partial(integrate.quad, inner, b=np.inf), x)))[
    return I0*integral

def rayleigh(w):
    return (np.pi*w**2)/wvl

plt.plot(x72, P(a_72.values[1], Rd), 'ro', label="8,3 +- 0,2cm")
plt.plot(x45, P(a_45.values[1], Rd), 'bo', label="11,0 +- 0,2 cm")
plt.plot(x105, P(a_105.values[1], Rd), 'go', label="5,0 +- 0,2 cm")

#8,3cm fit
popt, cov = curve_fit(gaussint, x72, P(a_72.values[1], Rd), bounds=(0, 15))
maxintensity, omega = popt
plt.plot(x, gaussint(x, maxintensity, omega), 'r')
#popt, cov = curve_fit(gaussint, x72new[2:], P((a_72.values[1]), Rd)[2:])
#maxintensity, omega = popt
maxints.append(maxintensity)
omegas.append(omega)
#plt.plot(x72[2:], gaussint(x72new[2:], maxintensity, omega), 'r')
print("8,3cm:\n", "I_0:", maxintensity, "Strahltaille:", omega)

#11cm fit
popt, cov = curve_fit(gaussint, x45, P(a_45.values[1], Rd), bounds=(0, 15))
maxintensity, omega = popt
plt.plot(x, gaussint(x, maxintensity, omega), 'b')
print("11cm:\n", "I_0:", maxintensity, "Strahltaille:", omega)

#5,0cm fit
popt, cov = curve_fit(gaussint, x105, P(a_105.values[1], Rd), bounds=(0, 15))
maxintensity, omega = popt
```

```
maxints.append(maxintensity)
omegas.append(omega)
plt.plot(x, gaussint(x, maxintensity, omega), 'g')
print("5,0cm:\n", "I_0:", maxintensity, "Strahltaille:", omega)

plt.xlabel("x-x_0 in mm (zero-centered micrometer position)")
plt.ylabel("power in mW") #Milli weil die Spannungen in mV angegeben sind
plt.legend(title="Distance razor blade - fibre end")
plt.title("cross section profile of collimated beam")
plt.savefig("cross section profile of collimated beam.png", dpi=400)

print("waist: (", np.mean(omegas), "+-", np.std(omegas), ") mm")
rays = [rayleigh(i)/10**6 for i in omegas] #10**6 weil millimeter umrechnen
print("rayleigh length: (", np.mean(rays), "+-", np.std(rays), ") mm")
```

Measuring the beam profile (b)

```
# -*- coding: utf-8 -*-
"""
Created on Sun Aug 21 12:05:53 2022

@author: corin
"""

# -*- coding: utf-8 -*-
"""
Created on Sat Jun 11 00:53:58 2022

@author: corin
"""

import numpy as np
import matplotlib.pyplot as plt
import pandas as pd
from scipy.optimize import curve_fit
import scipy.integrate as integrate
from functools import partial
import collections

csv_path_b = r"C:/Users/corin/Gausssche-Strahlenoptik/Strahlprofil_b.csv"

b_10 = pd.read_csv(csv_path_b, delimiter=";", header=None, skiprows=3, nrows=2,
b_9 = pd.read_csv(csv_path_b, delimiter=";", header=None, skiprows=6, nrows=2, u
b_11 = pd.read_csv(csv_path_b, delimiter=";", header=None, skiprows=9, nrows=2,
b_6 = pd.read_csv(csv_path_b, delimiter=";", header=None, skiprows=12, nrows=2,
b_3 = pd.read_csv(csv_path_b, delimiter=";", header=None, skiprows=15, nrows=2,
```

```
b_14 = pd.read_csv(csv_path_b, delimiter=";", header=None, skiprows=18, nrows=2,
b_17 = pd.read_csv(csv_path_b, delimiter=";", header=None, skiprows=21, nrows=2,
data = [b_3, b_6, b_9, b_10, b_11, b_14, b_17]

Rd = 10*10**3
#f = 100 #mm

lin = np.linspace(-0.8,0.8,100)
lin2 = np.linspace(-1.5, 1.5, 100)
maxints=[]
omegas= {}
localwaists=[]

def P(U, R):
    h = 6.62607015*10**(-34)
    c = 299792458
    wvl = 632.8*10**(-9)
    e = 1.602176634*10**(-19)
    return (h*c*U)/(wvl*R*e*0.75)

def localwaist(z, omega_0, rayleigh):
    #rayleigh = np.pi*(omega_0**2)/(632.8*10**(-9)
    return omega_0*np.sqrt(1+(z**2)/(rayleigh**2))

#lokaler Strahlradius bestimmen
def gaussint(x, I0, w):
    inner = lambda xp: np.exp((-2*xp**2)/(w**2))
    #integrate.quad kann keine Integrationsgrenzen als Variablen haben, darum ko
    integral = np.array(list(map(partial(integrate.quad, inner, b=np.inf), x)))[
    return I0*integral

#zero-centering
v1 = 5.7 #Verschiebung für rot
v2 = 6.2 #Verschiebung für blau
v3 = 2.3 # Verschiebung für grün
v4 = 5.7 #gelb
v5 = 14 #türkis
v6 = 14 #magenta
v7 = 10 #schwarz

#roter fit:
popt, cov = curve_fit(gaussint, b_10.values[0]-v1 , P(b_10.values[1], Rd), bound
maxintensity, omega = popt
maxints.append(maxintensity)
omegas[0] = omega
print("Rot: I_0:", maxintensity, "Strahltaille:", omega)
```

```
plt.plot(lin, gaussint(lin, maxintensity, omega), 'r')

#roter fit:
popt, cov = curve_fit(gaussint, b_9.values[0]-v2 , P(b_9.values[1], Rd), bounds=
maxintensity, omega = popt
maxints.append(maxintensity)
omegas[-1] = omega
print("Blau: I_0:", maxintensity, "Strahltaille:", omega)
plt.plot(lin, gaussint(lin, maxintensity, omega), 'b')

#grüner fit
popt, cov = curve_fit(gaussint, b_11.values[0]-v3 , P(b_11.values[1], Rd),bounds
maxintensity, omega = popt
maxints.append(maxintensity)
omegas[1] = omega
print("Grün: I_0:", maxintensity, "Strahltaille:", omega)
plt.plot(lin, gaussint(lin, maxintensity, omega), 'g')

#nahe z=0
plt.plot(b_10.values[0]-v1, P(b_10.values[1], Rd), 'ro', label="0,0 cm")
plt.plot(b_9.values[0]-v2, P(b_9.values[1], Rd), 'bo', label="-1,0 cm")
plt.plot(b_11.values[0]-v3, P(b_11.values[1], Rd), 'go', label="1,0 cm")
plt.legend()
plt.xlabel("x-x_0 in mm")
plt.ylabel("Power in mW")
plt.title("Cross section profile focused beam (near focal point)")
plt.savefig("Cross section profile focused beam (near focal point).png", dpi=400
plt.clf()

#gelber fit
popt, cov = curve_fit(gaussint, b_6.values[0]-v4 , P(b_6.values[1], Rd), bounds=
maxintensity, omega = popt
maxints.append(maxintensity)
omegas[-4] = omega
print("Gelb: I_0:", maxintensity, "Strahltaille:", omega)
plt.plot(lin2, gaussint(lin2, maxintensity, omega), 'y')

#türkiser fit
popt, cov = curve_fit(gaussint, b_3.values[0]-v5 , P(b_3.values[1], Rd), bounds=
maxintensity, omega = popt
maxints.append(maxintensity)
omegas[-7] = omega
print("türkis: I_0:", maxintensity, "Strahltaille:", omega)
plt.plot(lin2, gaussint(lin2, maxintensity, omega), 'c')

#magenta fit
```

```
popt, cov = curve_fit(gaussint, b_14.values[0]-v6 , P(b_14.values[1], Rd), bound
maxintensity, omega = popt
maxints.append(maxintensity)
omegas[4.3] = omega
print("magenta: I_0:", maxintensity, "Strahltaille:", omega)
plt.plot(lin2, gaussint(lin2, maxintensity, omega), 'm')

#schwarzer fit
popt, cov = curve_fit(gaussint, b_17.values[0]-v7 , P(b_17.values[1], Rd), bound
maxintensity, omega = popt
maxints.append(maxintensity)
omegas[7] = omega
print("schwarz: I_0:", maxintensity, "Strahltaille:", omega)
plt.plot(lin2, gaussint(lin2, maxintensity, omega), 'k')

plt.plot(b_6.values[0]-v4, P(b_6.values[1], Rd), 'yo', label="-4,0 cm")
plt.plot(b_3.values[0]-v5, P(b_3.values[1], Rd), 'co', label="-7,0 cm")
plt.plot(b_14.values[0]-v6, P(b_14.values[1], Rd), 'mo', label="4,3 cm")
plt.plot(b_17.values[0]-v7, P(b_17.values[1], Rd), 'ko', label="7,0 cm")
plt.legend()
plt.xlabel("x-x_0 (zero-centered values) in mm")
plt.ylabel("Power in mW")
plt.title("Cross section profile focused beam (far from focal point)")
plt.savefig("Cross section profile focused beam (far from focal point).png", dpi
plt.clf()

##waist bestimmen
z = sorted(omegas)
o = [omegas[i] for i in z]
popt, err = curve_fit(localwaist, z, o, absolute_sigma="True") #fitted localwais
waist, zR = popt
plt.plot(z, o, 'ro', label="data")
#localwaist(z, omega_0, rayleigh)
for i in z:
    localwaists.append(localwaist(i, waist, zR))
#plt.plot(z, localwaist(z, o))
plt.plot(z, localwaists, 'b', label = "fit")
plt.vlines(0, 0, 1.7, 'y', '--', label="focal position")
plt.xlabel("z in cm")
plt.ylabel("omega(z) in mm")
plt.legend()
title = "beam waist against razor position from focal point"
plt.title(title)
plt.savefig("beam waist against razor position from focal point.png", dpi=400)
print("waist: (", waist, "+-", np.sqrt(err[0,0]), ") mm")
print("rayleigh length: (", zR, "+-", np.sqrt(err[1,1]), ") mm")
```