Fitting the Longitudinal Profile of a Gaussian Beam

Fitting parameters:

- waist = w0
- waist position = offset
- M-square = M2

In [1]:
```
import numpy as np
import matplotlib.pyplot as plt
from scipy.optimize import curve_fit
import seaborn as sns

#Style setting
sns.set(style="darkgrid")

#Quick Conversions to SI units
nm = 1e-9
um = 1e-6
mm = 1e-3
```

Global Variables

In [2]:
```
n = 100
z = np.linspace(-1,1,n)
sW = np.random.normal(0,scale = 5*um, size=n)
ri = 1 # refractive index medium
wavelength = 1064 *nm
```

Define Fitting Fuction

In [3]:
```
#Beam Profile
def beamProfile( z, w0=1*mm, M2 =1,offset=.5):

    z_R = (np.pi * w0**2 ) / (wavelength* ri* M2)
    w = w0*np.sqrt(1+((z+offset)/z_R)**2)

    return w
```

Define Fit

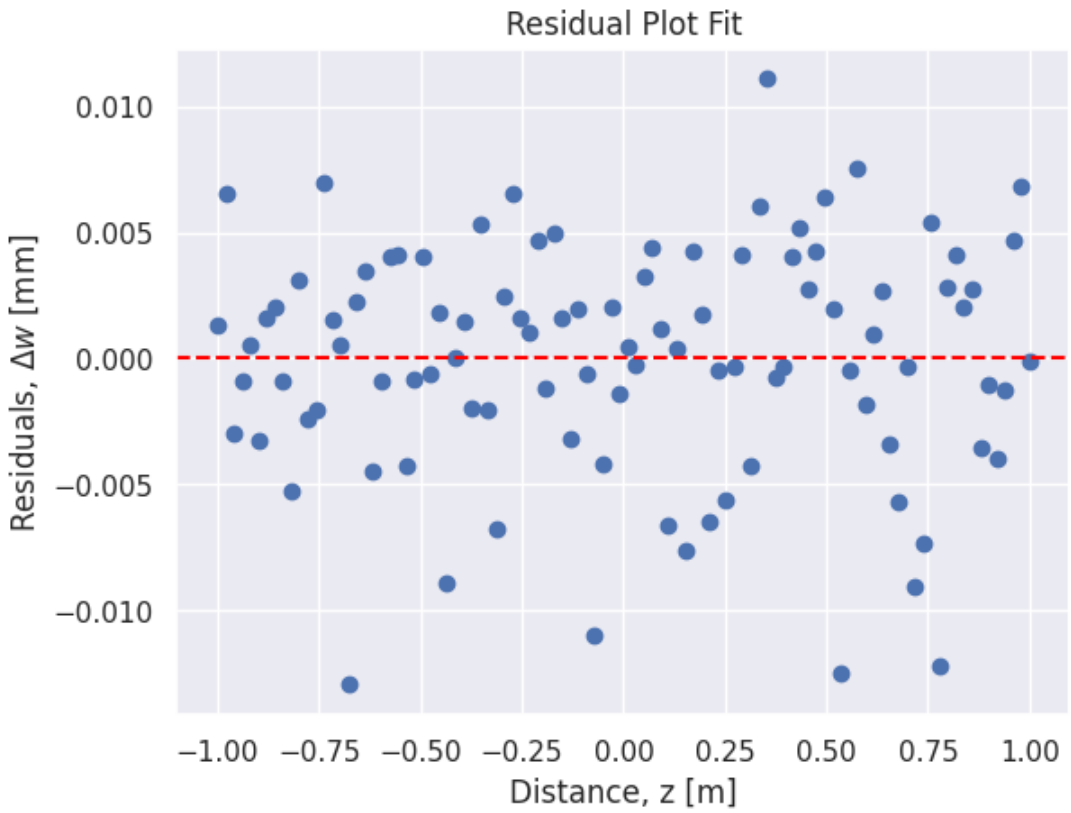
In [4]:
```
def fitBeamProfile(sW,z,n):

    w = beamProfile(z)+ sW
    params, covariance = curve_fit(beamProfile, z, w)
    print("Fitted parameters:", params)
    x_fit = z
    y_fit = beamProfile(x_fit,params[0], params[1], params[2])

    return x_fit,y_fit, w,z
```

In [5]:
```
x_fit, y_fit, w, z= fitBeamProfile(sW,z,n);
```

```
Fitted parameters: [1.00007767e-03 1.03163727e+00 4.77309253e-01]
```

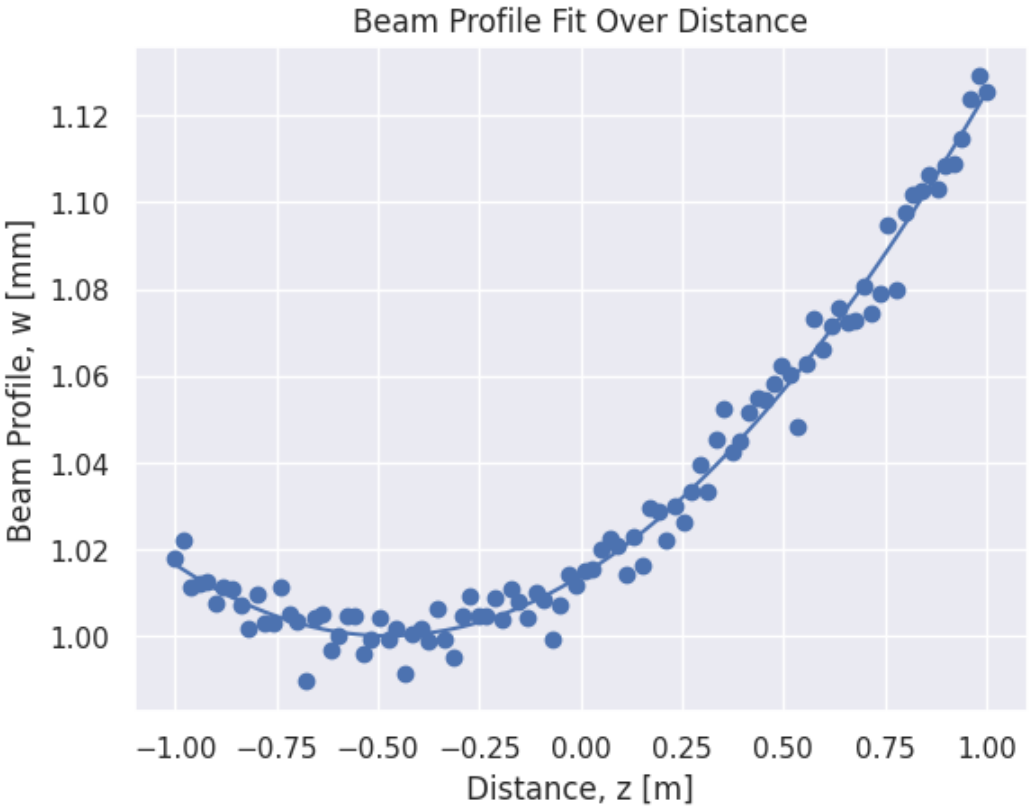
Plot residuals

In [6]:
```
residuals = w - y_fit
plt.scatter(z, residuals/mm)
plt.axhline(0, color='red', linestyle='--')
plt.xlabel('Distance, z [m]')
plt.ylabel('Residuals, $\Delta w$ [mm]')
plt.title('Residual Plot Fit')
plt.show()
```



Plot Fit

In [7]:
```
plt.title("Beam Profile Fit Over Distance")
plt.xlabel("Distance, z [m]")
plt.ylabel("Beam Profile, w [mm]")
plt.scatter(z,w/mm)
plt.plot(x_fit,y_fit/mm)
plt.show()
```


Beam Profile Fit Over Distance
Beam Profile, w [mm]
1.12
1.10
1.08
1.06
1.04
1.02
1.00
−1.00
−0.75
−0.50
−0.25
0.00
0.25
0.50
0.75
1.00
Distance, z [m]