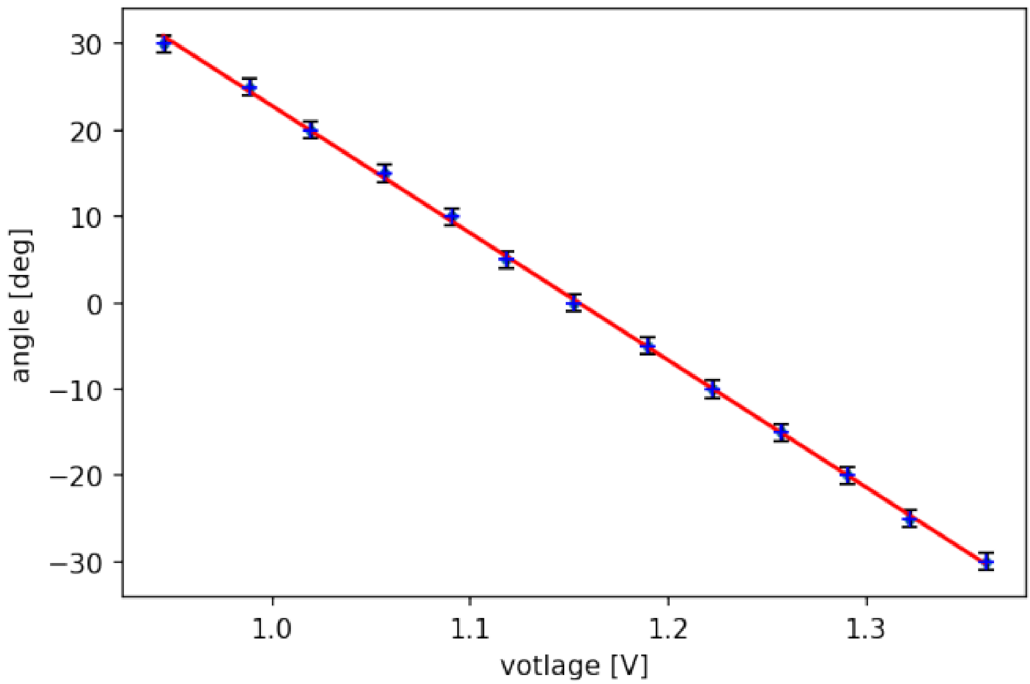
Analysis Appendix

Voltage vs angle raw data

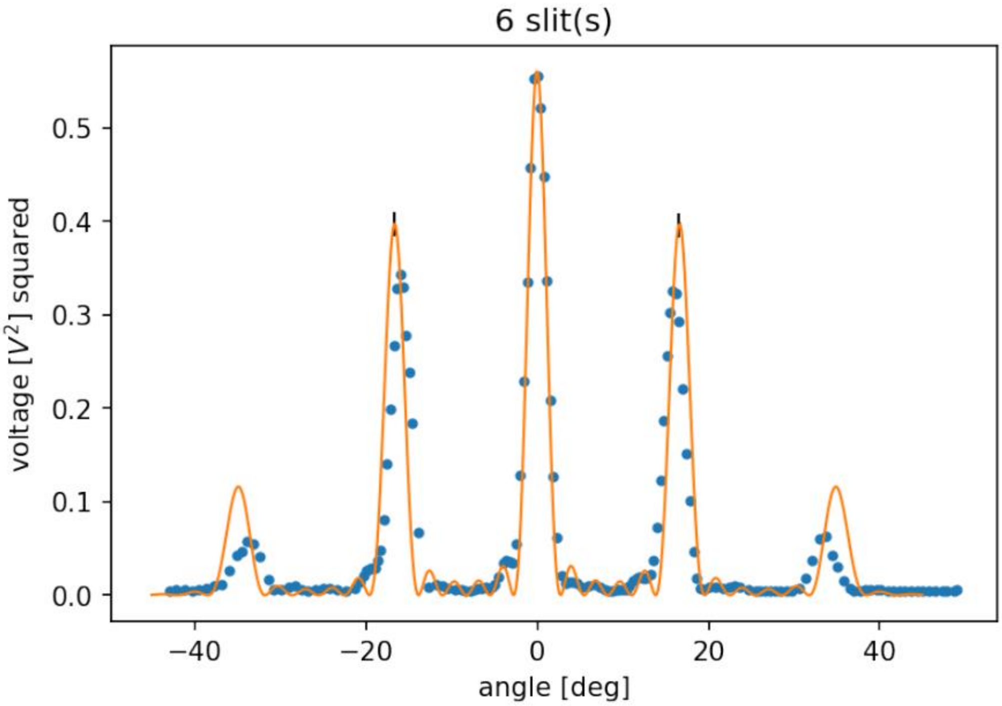
voltage [V] +-5e-5	angle [deg] +-1
1.3602	-30.0
1.3218	-25.0
1.2905	-20.0
1.2567	-15.0
1.2225	-10.0
1.1894	-5.0
1.1523	0.0
1.118	5.0
1.0909	10.0
1.0563	15.0
1.0194	20.0
0.9886	25.0
0.945	30.0

angle vs voltage = -147.23923 * x + 170.02281



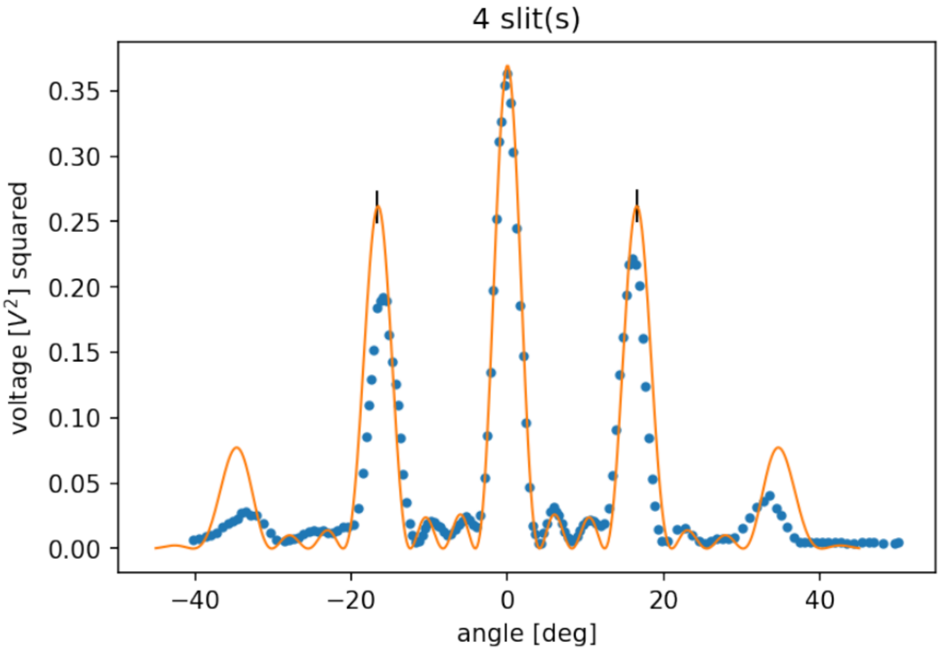
```
Speed of waves =  345.5256835409697 +- 0.5804227843792537 m/s
Wavelength of soundwaves at  40081 +- 0.5 Hz and 24.5 +- 0.5°C =  0.008620685200992233 +- 1.4481644393039123e-05 m
Distance to receiver = 302.4cm +- 0.5cm
Slit:  0.009574999999999998 +- 0.0001 Slat:  0.020454999999999997 +- 0.0001 d:  0.030029999999999994 +- 0.0002020120181555267
```

```
Speed of waves =  345.5256835409697 +- 0.5804227843792537 m/s
A_ 6  =  0.12493283333333333
16.6825686683772 0.11566999579836156
```



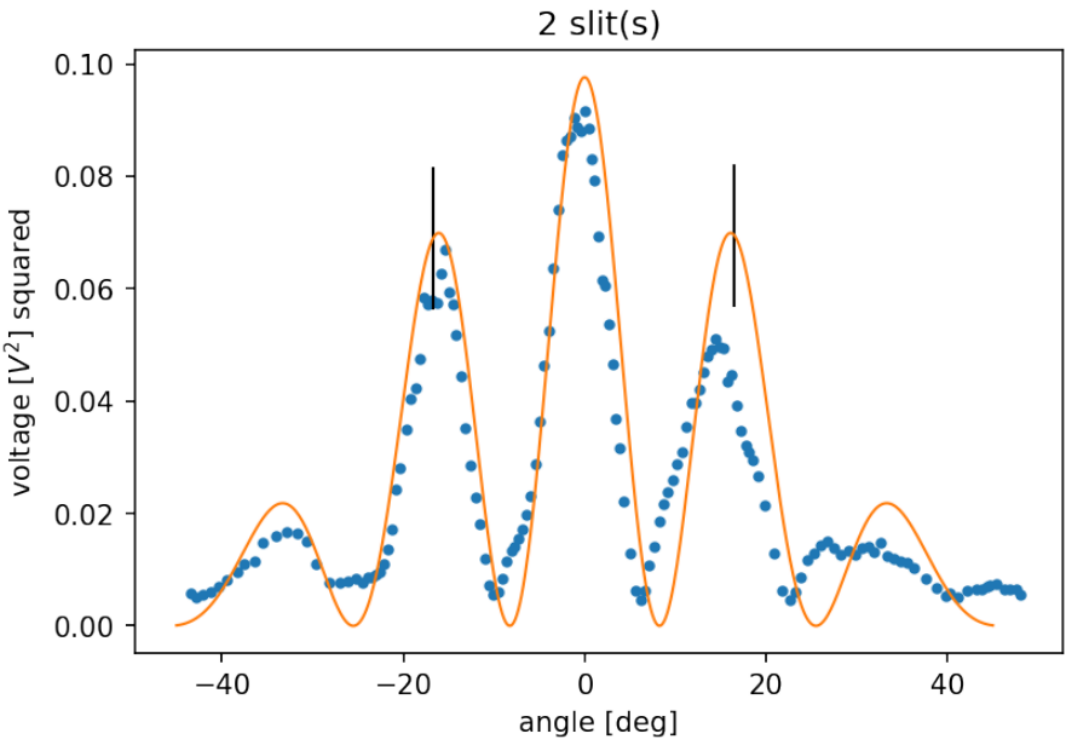
V at 0 = 1.18213

```
A_ 4  =  0.1519425
16.6825686683772 0.11566999579836156
```



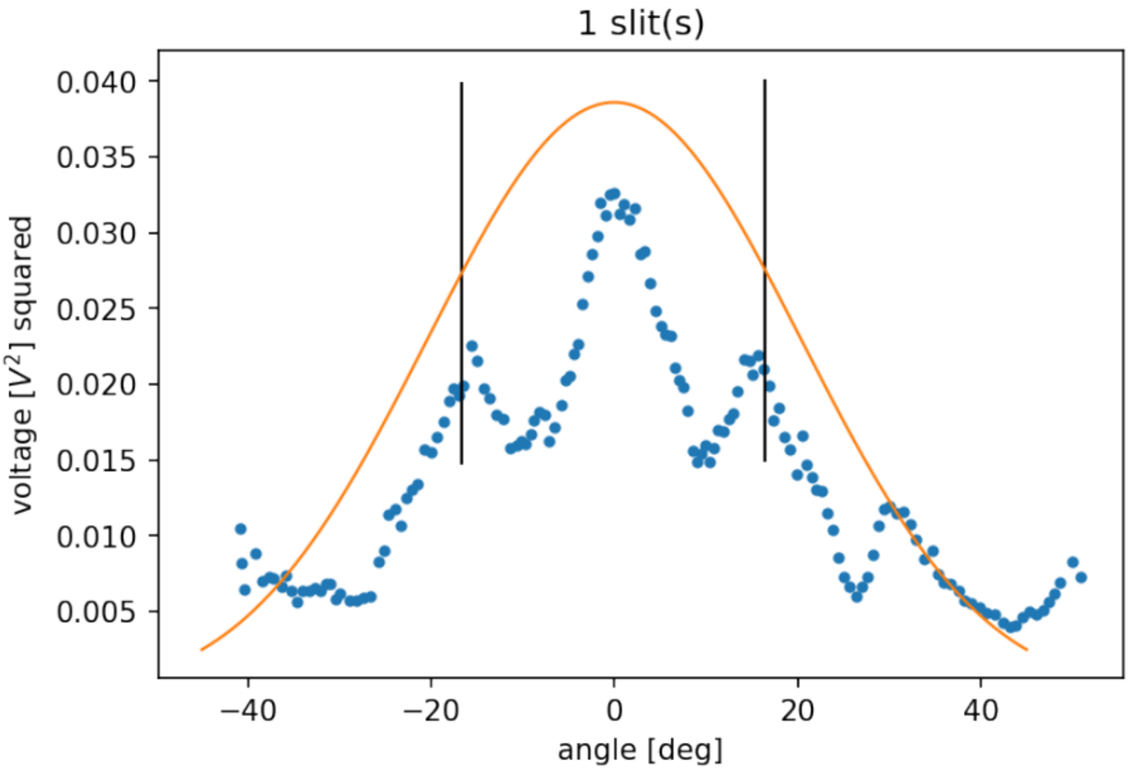
V at 0 = 1.18186

```
A_ 2  =  0.156248
16.6825686683772 0.11566999579836156
```

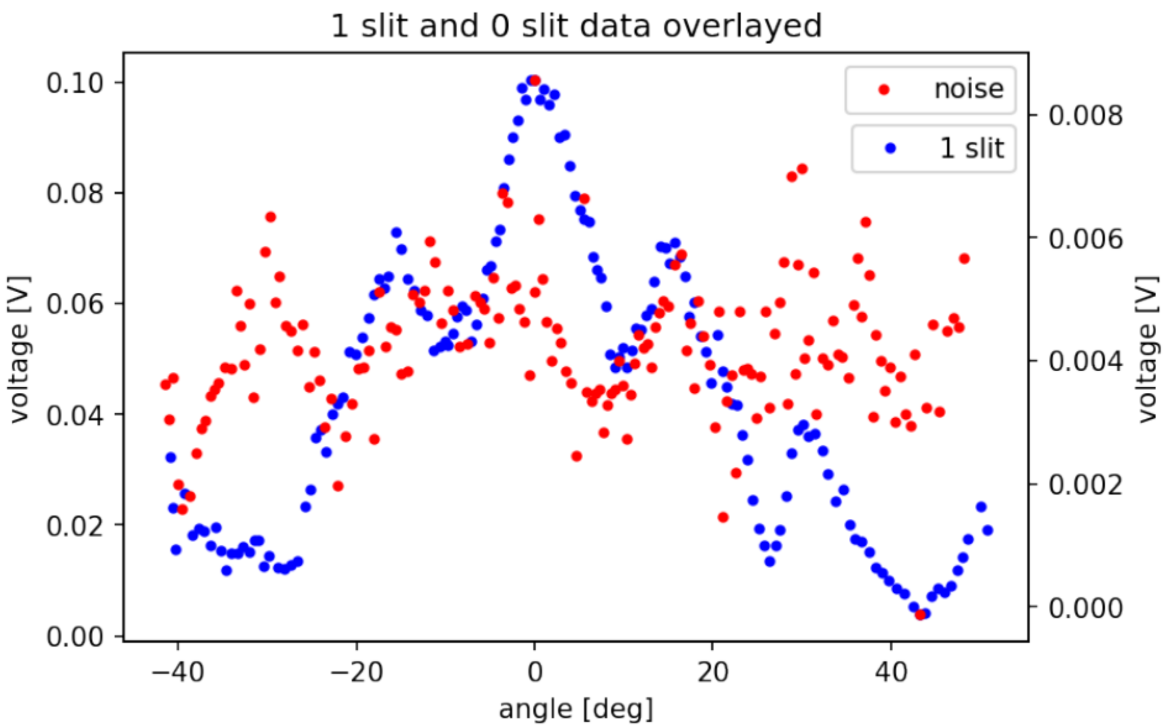


V at 0 = 1.16986

```
A_ 1  =  0.196469
16.6825686683772 0.11566999579836156
```



V at 0 = 1.18965



$$\Delta z = [(\Delta x)^2 + (\Delta y)^2]^{1/2} \quad (1)$$

$$\frac{\Delta z}{z} = [\frac{\Delta x^2}{x^2} + \frac{\Delta y^2}{y^2}]^{1/2} \quad (2)$$

$$A^2 = A_N^2 * \frac{\sin\left(\frac{\phi}{2}\right)^2}{\left(\frac{\phi}{2}\right)^2} * \frac{\sin\,(\frac{N\beta}{2})^2}{\sin\,(\frac{\beta}{2})^2} \quad (3)$$

$$\phi = kb * \sin(\theta) \quad (4)$$

$$\beta = \left(\frac{2\pi}{\lambda}\right) * d * \sin(\theta) \quad (5)$$

$$v_w = 331 * \sqrt{\frac{T}{273.15}} \quad (6)$$

Where b is the slit width, d is slit + slat, k is the wavenumber, T is in Kelvin and N is the number of slits.

```python
import numpy as np
import matplotlib.pyplot as plt
plt.rcParams['figure.dpi'] = 150
from scipy.optimize import curve_fit
import pandas as pd

angle = np.array([-30, -25, -20, -15, -10, -5, 0, 5, 10, 15, 20, 25, 30])
voltage = np.array([1.3602, 1.3218, 1.2905, 1.2567, 1.2225, 1.1894, 1.1523, 1.
 ↪1180, 1.0909, 1.0563, 1.0194, 0.9886, 0.9450])
data1 = np.column_stack((voltage, angle))
#print(data1)

# making a table to nicely read raw data
fig, ax = plt.subplots(1,1)
column_labels = ["voltage [V] +-5e-5", "angle [deg] +-1"]
df =pd.DataFrame(data1, columns=column_labels)
ax.axis('tight')
ax.axis('off')
ax.table(cellText=df.values, colLabels=df.columns, loc="center", cellLoc =␣
 ↪"center")
plt.show()

plt.plot(voltage, angle, '.')
def func(x, m, b):
    return m*x + b
popt, pcov = curve_fit(func, voltage, angle)
a, b = popt # used for conversion later from voltage to degrees
print('angle vs voltage = %.5f * x + %.5f' % (a, b))
plt.errorbar(voltage, angle, yerr=1, xerr=5e-5, fmt='b+', ecolor='k',capsize=3)
plt.plot(voltage, func(voltage, *popt), 'r'); plt.ylabel('angle [deg]'); plt.
 ↪xlabel('votlage [V]')
plt.show()
```

```python
slats_measured = np.array([20.52, 20.50, 20.40, 20.40]) # Slat measurements in␣
↪mm with error +-0.1mm for all
slat_error = 0.1 /1000 # in mm converted to m
slat = np.average(slats_measured)/1000 # average slat width converted to m
slits_measured = np.array([9.60, 9.52, 9.66, 9.52]) # Slit measurements in mm␣
↪with error +-0.1mm for all
slit_error = 0.1 / 1000 # in mm converted to m
slit = np.average(slits_measured)/1000 # average slit width converted to m

# standard deviation for slats and slits
std_slat = np.std(slats_measured)/1000
std_slit = np.std(slits_measured)/1000

d_rel = np.sqrt((std_slat / slat)**2 + (std_slit / slit)**2) # error in d


T = 24.5+273.15 # Room temperature in °C converted to Kelvin
T_error = 0.5 # K or °C
v_w = 331*np.sqrt(T/273.15) # Speed of soundwave at specific T
v_w_err_rel = T_error/T
v_w_error = v_w*v_w_err_rel

print("Speed of waves = ",v_w,"+-" ,v_w_error, "m/s")
```

```python
freq = 40081 # Frequency of soundwaves in Hz
freq_err = 0.5 # in Hz
wavelength = v_w/freq
w_error_rel = np.sqrt((v_w_error/v_w)**2 + (freq_err/freq)**2)
w_error = wavelength*w_error_rel

k = (2*np.pi)/wavelength # wavenumber
d = slit + slat # middle distance betweens slits
d_error = d*d_rel

#print(a, b)
# making garphing function to not repeat myself
def graph(file, N):
    array_in = np.loadtxt(file, delimiter=',')
    x = array_in[:,0] # taking second element of each row (second column)
    yy = array_in[:,1] # taking third element of each row (third column)
    y = yy**2
    h_N = np.amax(yy) # height of experimental central peak (or max voltage␣
↪reading from data)
    ind = np.argmax(yy) # index of peak
    shift = x[ind] # getting the 0 angle voltage at the peak value from data to␣
↪shift points accordingly
    A_N = h_N/N
```

```
    print("A_",N ," = ", A_N)
    xa = np.linspace(-45, 45, 500) # angles to plug into phi
    xa_rad = np.deg2rad(xa)
    phi = k*slit*np.sin(xa_rad) # calculating all phi values
    beta = k*d*np.sin(xa_rad)
    #A2 = (A_N**2)*((np.sin(phi/2)**2)/(phi/2)**2)*((np.sin((N*beta)/2)**2)/np.
↪sin(beta/2)**2)

    A2 = ((A_N**2)*((np.sin((phi/2))**2)/((phi/2)**2)))*((np.sin((N*beta)/
↪2)**2)/(np.sin((beta/2))**2))

    theta_rad = np.arcsin(1*wavelength/(d))
    theta = np.rad2deg(theta_rad)
    theta_err_rel = np.sqrt((w_error/wavelength)**2 + (d_error/d)**2)
    theta_err = theta*theta_err_rel

    x_err = np.zeros(500)
    x_err[342] = theta_err
    x_err[156] = theta_err
    print(theta, theta_err)

    phi_error = np.sqrt((slit_error/slit)**2 + (w_error_rel)**2)
    amp_error = np.sqrt((phi_error)**2 + (theta_err_rel)**2)

    y_err = np.zeros(500)
    y_err[341] = amp_error
    y_err[157] = amp_error
```

```
    # offsetting also y with 0.006V which works nicely with all graphs
    plt.plot((x*a + b)-(shift*a+b), y-0.006, '.'); plt.xlabel('angle [deg]');␣
↪plt.ylabel('voltage [$V^2$] squared') # plotting the data and giving it a␣
↪red colour
    plt.errorbar(xa, A2, xerr = x_err, yerr = y_err, ecolor='k', lw=1)
    title = str(N) + " slit(s)"
    plt.title(title)


#print(1.1613*a+b)

# 6 slits
graph("6slitdata13.csv", 6)
plt.show()

# 4 slits
graph("4slitdata1.csv", 4)
plt.show()
```

```
# 2 slits
graph("2slitdata13.csv", 2)
plt.show()

# 1 slit
graph("1slitdata11.csv", 1)
plt.show()


# 0 slits
array_in = np.loadtxt("data0.csv", delimiter=',')
x = array_in[:,0] # taking first column
y = array_in[:,1] # taking second column
# excluding last four data points here to remove anomaly
x_adj = x[:-4]
y_adj = y[:-4]
# offestting here too and and adjusting y so last point is exactly 0
ind = np.argmax(y_adj) # index of peak
shift = x[ind] # getting the 0 angle voltage at the peak value from data to␣
↪shift points accordingly
plt.plot((x_adj*a + b)-(shift*a+b), y_adj-0.096, '.'); plt.xlabel('angle␣
↪[deg]'); plt.ylabel('voltage [V]') # plotting the data and giving it a red␣
↪colour
plt.title('0 slits')
plt.show()
```

```
fig,ax = plt.subplots()
# 1 slit
array_in = np.loadtxt("1slitdata11.csv", delimiter=',')
x = array_in[:,0] # taking first column
y = array_in[:,1] # taking second column

ind = np.argmax(y) # index of peak
shift = x[ind] # getting the 0 angle voltage at the peak value from data to shift points accordingly
# offestting here too and and adjusting y so last point is exactly 0
ax.plot((x*a + b)-(shift*a+b), y-0.096, '.', color='blue', label='1 slit'); ax.set_xlabel('angle [deg]'); ax.set_ylabel('voltage [V]') # plotting the data and giving it a red colour

# 0 slit data displayed on separate axis

# 0 slits
ax2 = ax.twinx()
array_in = np.loadtxt("data0.csv", delimiter=',')
x = array_in[:,0] # taking first column
y = array_in[:,1] # taking second column
# excluding last four data points here to remove anomaly
x_adj = x[:-4]
y_adj = y[:-4]
ind = np.argmax(y_adj) # index of peak
shift = x[ind] # getting the 0 angle voltage at the peak value from data to shift points accordingly
# offestting here too and and adjusting y so last point is exactly 0
ax2.plot((x_adj*a + b)-(shift*a+b), y_adj-0.096, '.', color = 'red', label='noise'); ax2.set_ylabel('voltage [V]') # plotting the data and giving it a red colour
plt.title('1 slit and 0 slit data overlayed')
ax2.legend()
ax.legend(loc='best', bbox_to_anchor=(0.5, 0., 0.5, 0.9))
plt.show()
```