UNIVERSITY OF BRITISH COLUMBIA

Exploration of Resonant Frequency Perturbations due to Slotted and Solid Spheres

by

Ryan Kaufmann and Brittany Crooks

in the
Faculty of Science
Physics and Astronomy Department

April 2018

Abstract

Resonators are very common objects in today's society. They are used in many objects from electronics to musical instruments. One type of resonator, called a Helmholtz resonator, are easy to replicate and are the basis of many types of string instruments. The ease of making them is thus repeated in the ease of experimentation.

When looking at these resonators, it is often first nature to look at resonant frequencies. As part of this experiment, we will explore where resonant frequencies are and how they can be perturbed.

We found in our research that the resonant frequencies can be described mathematically. Furthermore, they can be derived from the partial differential equation known as the wave equation, although the derivation is too detailed to be put here. However, the experimental resonant frequencies that we found agreed with that of the analytic solutions. Once we had an idea of the resonant frequencies were located, we could move on to the perturbations of the solid sphere and slotted sphere.

Starting with the solid sphere, our data shows the resonant frequency has a sine dependency with respect to the depth of the solid sphere, such that:

$$f_r(x) = -3.347\sin(0.2084x + 0.1622) + 585.5$$

We associated this behavior to be the alternating between nodes and anti-nodes.

The slotted sphere data shows a similar sine dependency with different parameters, such that:

$$f_r(\theta) = 1.890\sin(0.03616\theta - 11.22) + 584.9$$

which again is associated with the slotted sphere being located at a point that's not a node.

Contents

Section 1

Introduction

The word "resonator" comes from the Latin verb "resonare" meaning "to resound". Resonators are objects characterized by an increased amplitude of certain waves that exist in the resonator. This can produce effects such as echoes and are often exploited when it comes to musical instruments, noise cancellation, or electric circuits. A cavity resonator is one where the waves that propagate in the resonator run through a hollow space. Furthermore, a Helmholtz resonator is a specific cavity resonator that looks at acoustic waves. Helmholtz resonators are used in musical instruments. As a result, their properties are interesting to both physicists and musicians alike.

The resonant frequencies are simply a property of the Helmholtz resonator. The overall shape and dimensions of the resonator dictates where the resonant frequencies are. These frequencies can both be solved for analytically or found experimentally.

The concern about an object in the resonator is different. Not only does it depend on the Helmholtz resonator itself, but also the physical properties of the object. This makes it much harder to show mathematically what the exact behavior of the wave is. Instead, we can look at the behavior of the resonant frequencies. We expect that the resonant frequencies would fluctuate somehow with the movement of the object.

These two properties are what we will be examining. They will give an understanding of the behavior of standing waves and resonant frequencies in Helmholtz resonators. However, the ideas could possibly be extended to resonators of other types.

Section 2

Background Theory

Waves can generally be written mathematically as a partial differential equation. In one dimension, we can write this differential equation as such:

$$\frac{\partial^2 u}{\partial t^2} = c^2 * \frac{\partial^2 u}{\partial x^2} \tag{2.1}$$

Or we can simplify this to be:

$$u_{tt} = c^2 * u_{xx} \tag{2.2}$$

The one dimensional wave equation can be expanded into three dimensions by adding two extra turns. The resulting equation is as follows:

$$u_{tt} = c^2 * (u_{xx} + u_{yy} + u_{zz}) \tag{2.3}$$

where x, y, and z are perpendicular directions. [1]

The wave equation can be solved for various properties such as frequency, wave speed, amplitude. When we apply our boundaries and speed, we can receive an equation for the resonant frequencies, which is as follows:

$$f = c_s \sqrt{(m/2a)^2 + (n/2b)^2 + (q/2d)^2} \tag{2.4}$$

where f is the resonant frequency; c_s is the speed of sound; a, b, and c are the boundaries of the resonator; and m, n, and q are either positive integers or zero. [3]

By using equation 2.4, we can apply our constraints for the Helmholtz resonator and use various integers for a, b, and c to find resonant frequencies. Several of these values are tabulated in table B.1.

Inside the Helmholtz resonator, these particular resonant frequencies act as standing waves with fixed boundaries. As such, the space inside the resonator can be divided into nodes, anti-nodes, or neither. A node is defined as the point along the wave

with the minimum amplitude at all times. These points are where the waves experience destructive interference. Anti-nodes are the opposite, where the wave experiences the maximum amplitude at any time. These form the points of constructive interference. [2]

When something is placed at a node, it doesn't affect the wave at all. The wave still resonates at the same frequency and the amplitude feels the same. However, if moved away from the node, this is not the case. As the object moves toward an anti-node, it cause more interference with the wave. The object becomes a source of destructive interference. It can be thought of as a new node. This changes the frequency of the wave to match the node. As the object moves closer to the anti-node, it changes the frequency more and more, until the anti-node is reached.

In this experiment, we will be observing the behavior of both the resonant frequencies and nodes.

Section 3

Experimental Setup

We set up our Helmholtz resonator such that we have a square-based rectangular prism of dimensions 1.04 meters by 0.437 meters by 0.437 meters. Placed in the bottom of the resonator is a microphone and speaker. A computer was then hooked up to both the microphone and speaker in order to record and emit desired frequencies (see Figure A.1). SigGen and TrueRTA were installed on the computer to send frequencies to the speaker and to check the amplitude of the waves through the microphone.

The first part of the experiment deals with locating resonant frequencies. In other words, we sought out any large amplitudes when a specific frequency was being sent to the speakers. From the approximations done in the background theory, we justify that a search of frequencies between 0 hertz and 700 hertz. To get as full coverage as we can, we investigate every 50 hertz. Then any larger than normal amplitudes detected by TrueRTA reveals a resonant frequency near the one we are looking at. This warrants investigation until the resonant frequency can be detected to the nearest Hertz.

With the information gained regarding the resonant frequencies, we can see how they fluctuate with respect to certain types of perturbations. Our experiment looks at two specific ones. The first is the perturbations caused by a glass sphere lowered into the Helmholtz resonator (see Figure A.2). The second is caused by a slotted sphere rotated at a constant height. We look at the two of these individually (see Figure A.3).

The second part of the experiment concerns the glass sphere. We inserted a glass sphere attached to a metal rod to the top of the Helmholtz resonator. The metal rod was extended down to the 14 centimeter marking on the rod, the smallest possible depth for the rod to be lowered. The glass sphere was then lowered in alternating increments of 3 and 4 centimeters until the glass sphere reached the speaker at approximately 79 centimeters. Centimeter markings on the rod added to the accuracy of the depth measurements. Care was taken to not hit the speaker with the glass sphere. At each depth, we searched for the resonant frequency. We can get the resonant frequency by searching for a frequency that is close. We record this frequency and its output voltage.

Then on either side of this frequency we search for a voltage that is half of our record output voltage. The resonant frequency is then the average of these frequencies.

The third and final part of the experiment concerns the slotted sphere. We inserted a slotted sphere attached to a metal rod to the side of the Helmholtz resonator approximately one sixth from the top of the resonator. The slotted sphere was set so that it was directly above the speaker in the resonator. We set zero degrees to be when the slits in the sphere were perpendicular to the horizon. The sphere was rotated at intervals of 30 degrees from 0 to 330 degrees. Markings and a temporary protractor on the side of the resonator were used to help guide where the degree measurements were. At each angle, we again searched for the resonant frequency using the same technique as the glass sphere.

Section 4

Results and Discussion

4.1 Resonant Frequencies

The first part of the experiment focuses on the finding of resonant frequencies for our Helmholtz resonator. Recall that our resonator has dimensions of 1.04 meters by 0.437 meters by 0.437 meters and follows the diagram shown in Figure A.1. After searching for the resonant frequencies, we tabulated the frequencies we tested with their respective output voltages in Table B.2.

From the table, we can notice a couple of resonant frequencies, namely around 165, 295, 303, 428, and 587 hertz. The frequency with the highest reaction is the 587 Hz resonant frequency. The shape of the output voltage with respect to frequency can be modeled using some sort of distribution. After looking at the data and comparing it to various functions, we fit the data to Lorentz distributions in sections, each focused on a resonant frequency. The Lorentz distribution can be modeled as:

$$L(f, f_0, \gamma) = \frac{1}{\pi\gamma} \frac{\gamma^2}{(f - f_0)^2 + \gamma^2}$$

where f is variable, f_0 is the center of the distribution, and γ describes the width of the distribution.

The data was fitted with four Lorentz distributions, each centered at one of the resonant frequencies with the four highest output voltages. The fitted graph can be seen in Figure C.1, with the residuals in Figure C.2. The data was fitted using the code in Code D.1.

The resonant frequencies that we found matched mostly with the resonant frequencies that we found analytically in Section 3.

4.2 Solid Glass Sphere

Now that we have some resonant frequencies, we can begin to observe the second part of the experiment regarding objects in the resonator. Recall our set up diagram shown in Figure A.2. The glass sphere moves up and down vertically directly above the speaker. We took the data alternating between 3 and 4 centimeter changes. At each measurement, we found the resonant frequency using the method described in the previous chapter. This data is shown in Table B.3.

When plotted, this data shows a somewhat sinusoidal shape. We can then fit the data to a sine wave of frequency as a function of depth. The data is fitted using the code in Code D.2. After using the code, we get a resulting fitted graph that can be seen in Figure C.3 with residuals in Figure C.4. The data gives us a chi-squared value of 1.48. This value, along with the residuals, show us that the fit function agrees with the data of the table. The fit function gives us the resulting equation:

$$f_r(x) = -3.347\sin(0.2084x + 0.1622) + 585.5$$

where x is the depth in centimeters.

This agrees with what we had predicted with our theorems. The resonant frequency oscillates as the solid sphere is lowered into the resonator. It must also be then that when $\sin(0.2084x+0.1622) = 0$, the sphere must be located at a node. Furthermore, if $|\sin(0.2084x+0.1622)| = 1$, the sphere must be located at an anti-node. By placing an object at the anti-node, we can reduce or increase the resonant frequency by a maximum of 3.347 Hz. We do not, however, know how or even if the four parameters (amplitude, frequency, phase shift, and offset) are related to each other and can be determined physically. Albeit, we have shown that the resonant frequency alternates as a sinusoidal with respect to the depth of an object in the resonator.

4.3 Slotted Sphere

We can now look to see how the resonant frequency changes with a rotating slotted sphere, and if it is distinguishable to the solid sphere. Recall our set up diagram shown in Figure A.3. The slotted sphere rotates above the speaker at approximately one sixth of the height from the top of the resonator. We took data at every 30 degrees. At each measurement, we found the resonant frequency using the method described in the previous chapter. This data is shown in table B.4.

Based on the geometry of the sphere, we can shift all the data above 180 degrees down by 180 degrees. This makes sense at 0 degrees corresponds to the exact shape of the sphere at 180 degrees, and 30 degrees to 210 degrees, and so forth. Thus, our data set between 0 and 330 degrees is adjusted to be between 0 and 180 degrees.

When plotted, this data shows a shape that is either sinusoidal or quadratic. Assuming that this data has the shape of a sine wave, we fit the data using the code in Code D.3. After using the code, we get a resulting fitted graph that can be seen in Figure C.5 with residuals in Figure C.6. The fit function gets a chi-squared value of 2.62. While this value doesn't give us the best goodness of fit, when combined with the residuals, it is adequate enough. The data had received a fit function matching the following:

$$f_r(\theta) = 1.890\sin(0.03616\theta - 11.22) + 584.9$$

where θ is measured in degrees.

Once again, this agrees with our predictions. The resonant frequency again fluctuates as the slotted sphere is turned. It experiences no interference when the slots are parallel to the direction the speaker faces, while it experiences maximum interference when the slots are perpendicular to this direction. Furthermore, since there is some amount of fluctuation in the frequency, the slotted sphere must not be located at a node, or there would be absolutely no fluctuation in the resonant frequency. Once again, similarly to the solid sphere, the relationship between the parameters is unknown. Finally, there seems to be some difference between the two sinusoidal waves. At the very least, the parameters, barring the offset value, are too different between the slotted sphere and solid sphere.

Section 5

Conclusion

This experiment focused on the identification of resonant frequencies and their properties, namely how they react with some perturbation within the resonator. With the resonator set up, we began to look through 10 Hz and 700 Hz for any resonant frequencies. We managed to find several resonant frequencies that matched our analytic solution to the three dimensional wave equation. The matching of these frequencies shows that there is both some merit to our experiment and the mathematical equation.

After receiving a small amount of resonant frequencies, we decided to begin to analyze the perturbations. Starting with the solid sphere, we searched for the resonant frequencies whenever we changed the depth of the solid sphere using a technique of averaging with help from the output voltages. As a result, we managed to show that the resonant frequency has a sine wave dependency on the depth of the solid sphere. Specifically, the resonant frequency fluctuates such that:

$$f_r(x) = -3.347\sin(0.2084x + 0.1622) + 585.5$$

Finally, there was the slotted sphere. Using the same method as the solid sphere, we searched for resonant frequencies. From the data collection, we got to show the resonant frequency has a sine dependency on the angle of rotation of the slotted sphere. The resonant frequency fluctuates so that it satisfies:

$$f_r(\theta) = 1.890\sin(0.03616\theta - 11.22) + 584.9$$

Both of the solid sphere and slotted sphere, we had determined, experienced their sinusoidal properties due to the placement of the nodes and anti-nodes. When the solid sphere is at a node or the slotted sphere has its slots parallel to the way the speaker faces, there is no perturbation in the resonant frequency. However, when the solid sphere is at an anti-node or the slotted sphere has its slots perpendicular to the way the speaker faces, there is a maximum perturbation in the resonant frequency.

Overall, the Helmholtz resonator has resonant frequencies that are determined by the properties of the resonator. Meanwhile, the perturbations in the resonant frequencies are brought about by a combination of the physical properties of the Helmholtz resonator and the physical properties and placement of the object in the resonator.

Appendix A

Figures

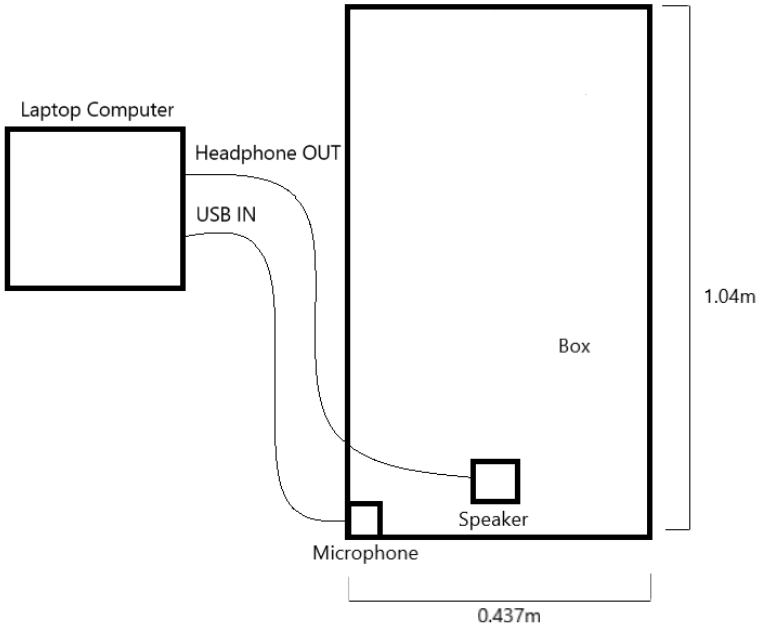

FIGURE A.1: Set up of Helmholtz Resonator with Laptop Computer

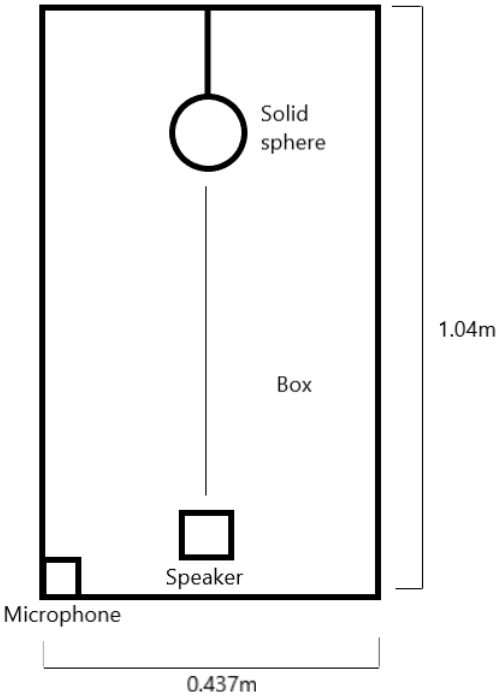

FIGURE A.2: Set up of Helmholtz Resonator Using Glass Sphere

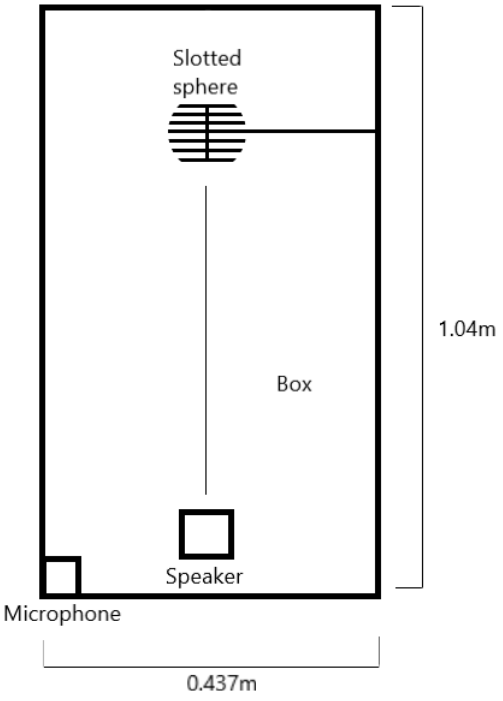

FIGURE A.3: Set up of Helmholtz Resonator Using Slotted Sphere

Appendix B

Tables

m (Z-direction)	n (Y-direction)	q (X-direction)	Resonant Frequency (Hz)
0	0	1	389.3
0	1	0	389.3
1	0	0	163.6
1	1	0	422.3
1	0	1	422.3
0	1	1	550.6
1	1	1	574.4

TABLE B.1: Resonant Frequencies Determined by the Analytical Solution

Frequency (Hz)	Output Voltage (mV)
10	3.5
50	3.5
100	4.5
125	4.4
150	7.7
155	6.6
160	12
162	11.3
165	12.1
170	9.8
175	7.3
200	5.2
250	5.7
290	10.5
295	15
300	3

301	45.6
302	53.6
303	60.6
305	33.6
310	15.8
350	6.8
400	18.6
425	35.5
428	37.7
429	37.5
432	34.7
435	33
450	21
500	23
550	12.1
580	91
585	181
586	206.5
587	218.7
588	207
590	157.9
600	49.6
610	26.5
650	5.4
675	6.2
700	14.4
750	20.8

TABLE B.2: Data Received from Searching for Resonance Frequency

Depth (cm)	Resonant Frequency (Hz)
14	585.25
17	586.85
21	589.125
24	588.15
28	586.75
31	583.875

35	582.625
38	582.125
42	584.05
45	585.775
49	588.55
52	588.7
56	588.1
59	585.075
63	583.625
66	582.425
70	583.15
73	583.3
77	587.625

TABLE B.3: Data of Resonant Frequency Based on Depth of Glass Sphere

Angle (Degrees)	Resonant Frequency (Hz)
0	586.825
30	585.6
60	584
90	580.5
120	583.75
150	585.55
180	586.775
210	585.7
240	583.95
270	583.1
300	584.05
330	586.35

TABLE B.4: Data of Resonant Frequency Based on Rotation of Slotted Sphere

Appendix C

Graphs

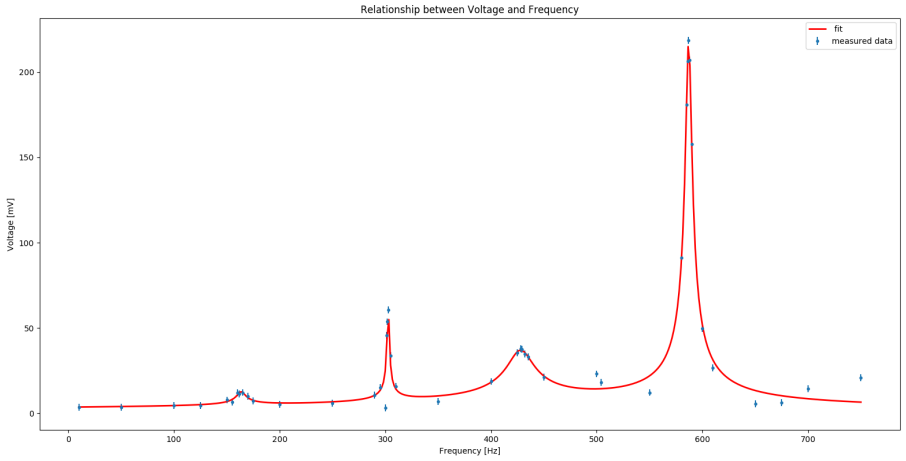

FIGURE C.1: Fitted Lorentz Distribution to Resonant Frequency Data

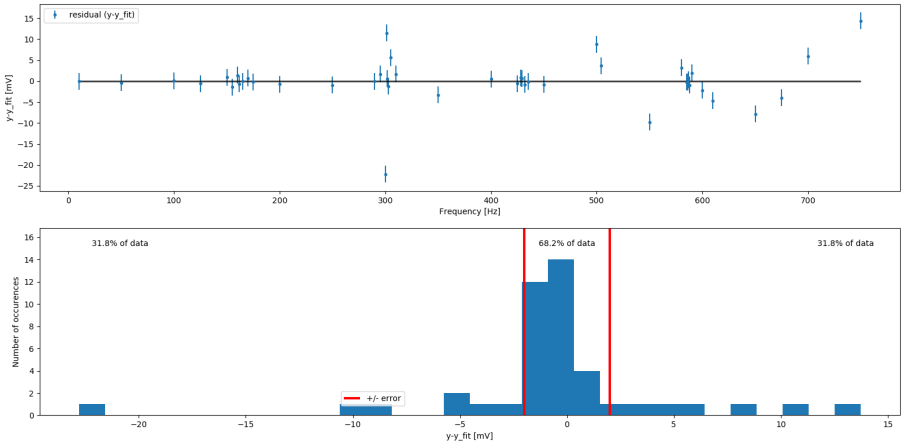

FIGURE C.2: Residuals of Lorentz Distribution Fit to Resonant Frequency Data

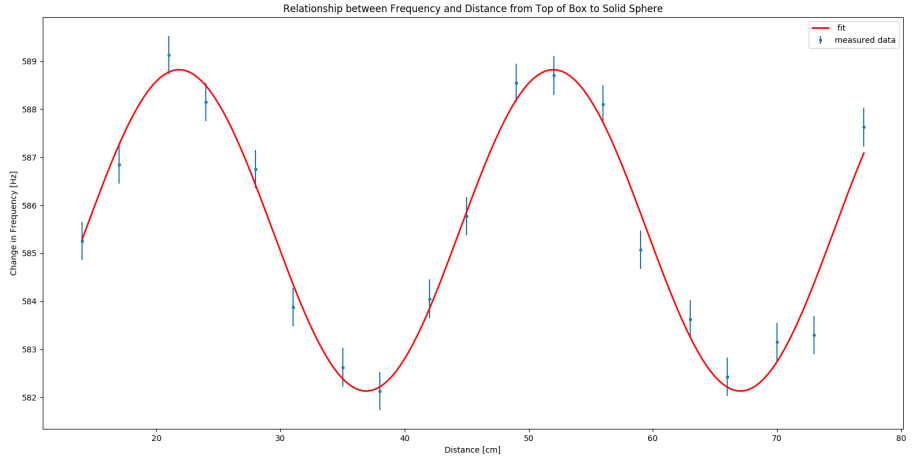

FIGURE C.3: Fitted Sine Function to Glass Sphere Perturbation Data

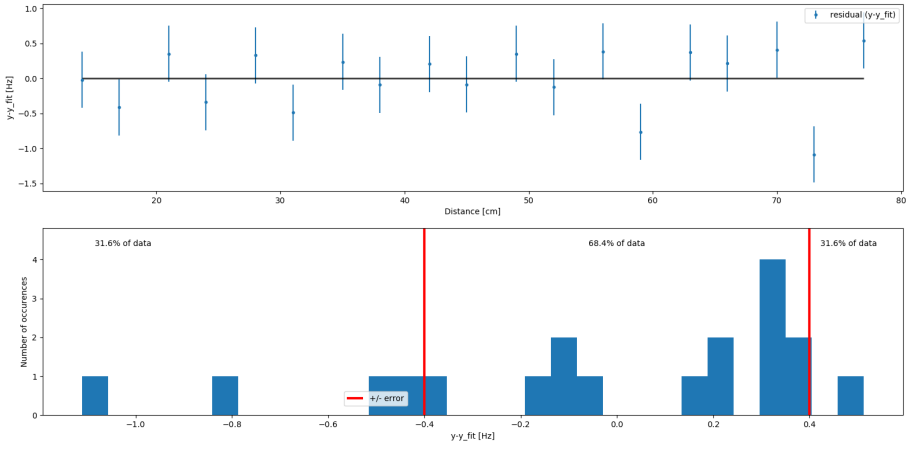

FIGURE C.4: Residuals of Sine Function Fit to Glass Sphere Perturbation Data

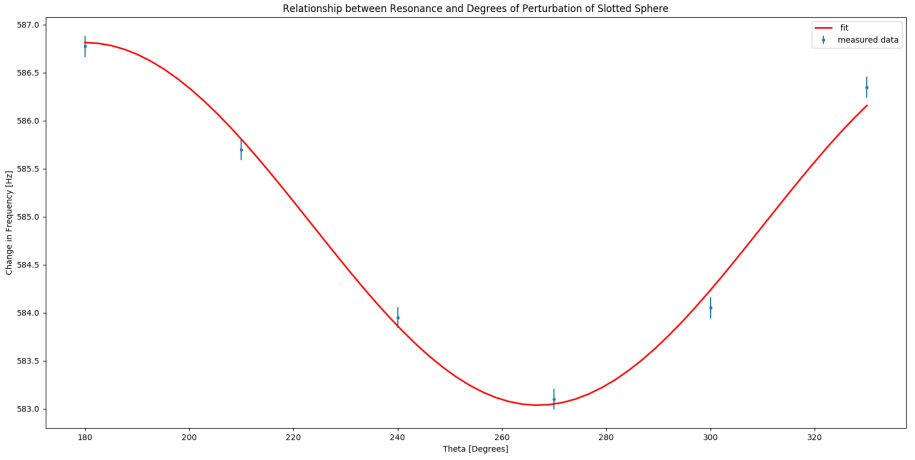

FIGURE C.5: Fitted Sine Function to Slotted Sphere Perturbation Data

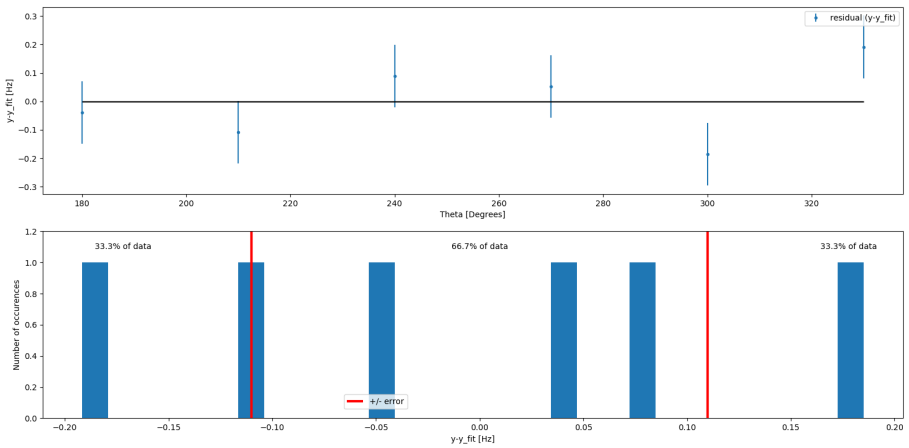

FIGURE C.6: Residuals of Sine Function Fit to Slotted Sphere Perturbation Data

Appendix D

Code

```
import numpy as np
import matplotlib.pyplot as plt
from scipy.optimize import curve_fit as cf

def Lorentzian(freq, V0, V1, V2, V3, delta1, delta2, delta3, delta4,
               resonance1, resonance2, resonance3, resonance4, Voff):
    LorentzOne = V0/((delta1**2+(freq - resonance1)**2)**(1/2))
    LorentzTwo = V1/((delta2**2+(freq - resonance2)**2)**(1/2))
    LorentzThree = V2/((delta3**2+(freq - resonance3)**2)**(1/2))
    LorentzFour = V3/((delta4**2+(freq - resonance4)**2)**(1/2))
    return LorentzOne+LorentzTwo+LorentzThree+LorentzFour+Voff

def chi_square (params, x, y, sigma):
    if sigma is None:
        sigma = 1.
    return np.sum((y-Lorentzian(x, *params))**2/sigma**2)

filename = 'C:/Users/ryank/Desktop/Work/Classes/Python/PHYS229/Acoustics/Data'
filename += '/AcousticsDataP1.csv'
x_label = 'Frequency'
x_units = 'Hz'
y_label = 'Voltage'
y_units = 'mV'
Data = np.loadtxt(filename, delimiter=',', comments='#', usecols=(0,1))

guess = [47, 60, 214, 650, -5.5, -1, -8.5, -3, 164, 303, 429, 587, 3.713]
sigma = 2

################################################################################

x_guess = np.linspace(min(Data[:, 0]),max(Data[:, 0]),500)
y_guess = Lorentzian(x_guess, *guess)

plt.errorbar(Data[:, 0], Data[:, 1], yerr=sigma, marker='.', linestyle='',
             label="measured data")
plt.plot(x_guess, y_guess,marker="", linestyle="-", linewidth=1, color="g",
         label="initial guess")
plt.xlabel('{} [{}]'.format(x_label, x_units))
plt.ylabel('{} [{}]'.format(y_label, y_units))
plt.title(r'Comparison between the data and the intial guess')
plt.legend(loc='best', numpoints=1)
```

```
print ('\nDisplaying plot 1')
plt.show()

###############################################################################

params, cov = cf(Lorentzian, Data[:, 0], Data[:, 1],
                 sigma=sigma*np.ones(len(Data[:, 0])), p0=guess, maxfev=10**5)

chi2 = chi_square(params, Data[:, 0], Data[:, 1], sigma)
dof = len(Data[:, 0]) - len(params)
print ("\nGoodness of fit - chi square measure:")
print ("Chi2 = {}, Chi2/dof = {}\n".format(chi2, chi2/dof))

cov = cov*dof/chi2
paramserr = np.sqrt(np.diag(cov))

param_names = ['V0', 'V1', 'V2', 'V3', 'delta1', 'delta2', 'delta3', 'delta4',
               'resonance1', 'resonance2', 'resonance3', 'resonance4',
               'Voffset']

print ("Fit parameters:")
for i in range(len(params)):
    print ('{} = {:.3e} +/- {:.3e}'.format(param_names[i], params[i],
                                           paramserr[i]))

###############################################################################

x_fit = np.linspace(min(Data[:, 0]),max(Data[:, 0]),len(Data[:, 0])*10)
y_fit = Lorentzian(x_fit,*params)

plt.errorbar(Data[:, 0], Data[:, 1], yerr=sigma, marker='.', linestyle='',
             label="measured data")
plt.plot(x_fit, y_fit, marker="", linestyle="-", linewidth=2, color="r",
         label=" fit")
plt.xlabel('{} [{}]'.format(x_label, x_units))
plt.ylabel('{} [{}]'.format(y_label, y_units))
plt.title(r'Relationship between Voltage and Frequency')
plt.legend(loc='best', numpoints=1)
print ('\nDisplaying plot 2')
plt.show()

###############################################################################

y_fit=Lorentzian(Data[:, 0],*params)
residual = Data[:, 1]-y_fit
hist,bins = np.histogram(residual,bins=30)

fig = plt.figure(figsize=(7,10))
ax1 = fig.add_subplot(211)
ax1.errorbar(Data[:, 0], residual, yerr=sigma, marker='.', linestyle='',
             label="residual (y-y_fit)")
ax1.hlines(0,np.min(Data[:, 0]),np.max(Data[:, 0]),lw=2,alpha=0.8)
ax1.set_xlabel('{} [{}]'.format(x_label,x_units))
ax1.set_ylabel('y-y_fit [{}]'.format(y_units))
ax1.legend(loc='best',numpoints=1)
```

```
ax2 = fig.add_subplot(212)
ax2.bar(bins[:-1],hist,width=bins[1]-bins[0])

ax2.set_ylim(0,1.2*np.max(hist))
ax2.set_xlabel('y-y_fit [{}]'.format(y_units))
ax2.set_ylabel('Number of occurences')

if sigma != None:
    within_err=100.*np.sum((residual<=sigma)&(residual>=-sigma))/len(residual)
    print ("\nResidual information:")
    print ('{:.1f}% of data points agree with fit'.format(within_err))
    ax2.vlines(-sigma,0,np.max(hist)*1.3,lw=3,color='r')
    ax2.vlines(+sigma,0,np.max(hist)*1.3,lw=3,color='r',label='+/- error')
    ax2.text(0.0,np.max(hist)*1.1,'{:.1f}% of data'.format(within_err),
             horizontalalignment='center',verticalalignment='center')
    ax2.text(np.min(bins),np.max(hist)*1.1,'{:.1f}% of data'.format(100-within_err),
             horizontalalignment='left',verticalalignment='center')
    ax2.text(np.max(bins),np.max(hist)*1.1,'{:.1f}% of data'.format(100-within_err),
             horizontalalignment='right',verticalalignment='center')
    ax2.legend(loc=(0.35,0.05))

print ('\nDisplaying plot 3')
plt.show()
```

Code D.1: Lorentz Fitting Code

```
import numpy as np
import matplotlib.pyplot as plt
from scipy.optimize import curve_fit as cf

def Periodic(distance, f0, omega, phase, resonance):
    return f0*np.sin(omega*distance+phase)+resonance

def chi_square (params, x, y, sigma):
    if sigma is None:
        sigma = 1.
    return np.sum((y-Periodic(x, *params))**2/sigma**2)

filename = 'C:/Users/ryank/Desktop/Work/Classes/Python/PHYS229/Acoustics/Data'
filename += "/AcousticSolidSphereDataP4.csv"
x_label = 'Distance'
x_units = 'cm'
y_label = 'Change in Frequency'
y_units = 'Hz'
Data = np.loadtxt(filename, delimiter=',', comments='#', usecols=(0,1))

guess = [-3.3, 0.2, 0.06, 585]
sigma = 0.4

################################################################################

x_guess = np.linspace(min(Data[:, 0]),max(Data[:, 0]),500)
y_guess = Periodic(x_guess, *guess)
```

```
plt.errorbar(Data[:, 0], Data[:, 1], yerr=sigma, marker='.', linestyle='',
             label="measured data")
plt.plot(x_guess, y_guess,marker="", linestyle="-", linewidth=1, color="g",
         label="initial guess")
plt.xlabel('{} [{}]'.format(x_label, x_units))
plt.ylabel('{} [{}]'.format(y_label, y_units))
plt.title(r'Comparison between the data and the intial guess')
plt.legend(loc='best', numpoints=1)
print ('\nDisplaying plot 1')
plt.show()

###############################################################################

params, cov = cf(Periodic, Data[:, 0], Data[:, 1],
                 sigma=sigma*np.ones(len(Data[:, 0])), p0=guess, maxfev=10**5)

chi2 = chi_square(params, Data[:, 0], Data[:, 1], sigma)
dof = len(Data[:, 0]) - len(params)
print ("\nGoodness of fit - chi square measure:")
print ("Chi2 = {}, Chi2/dof = {}\n".format(chi2, chi2/dof))

cov = cov*dof/chi2
paramserr = np.sqrt(np.diag(cov))

param_names = ['f0','omega', 'phase','resonance']

print ("Fit parameters:")
for i in range(len(params)):
    print ('{} = {:.3e} +/- {:.3e}'.format(param_names[i], params[i],
                                           paramserr[i]))

###############################################################################

x_fit = np.linspace(min(Data[:, 0]),max(Data[:, 0]),len(Data[:, 0])*10)
y_fit = Periodic(x_fit,*params)

plt.errorbar(Data[:, 0], Data[:, 1], yerr=sigma, marker='.', linestyle='',
             label="measured data")
plt.plot(x_fit, y_fit, marker="", linestyle="-", linewidth=2, color="r",
         label=" fit")
plt.xlabel('{} [{}]'.format(x_label, x_units))
plt.ylabel('{} [{}]'.format(y_label, y_units))
plt.title(r'Relationship between Frequency and Distance from Top of Box to Solid Sphere')
plt.legend(loc='best', numpoints=1)
print ('\nDisplaying plot 2')
plt.show()

###############################################################################

y_fit=Periodic(Data[:, 0],*params)
residual = Data[:, 1]-y_fit
hist,bins = np.histogram(residual,bins=30)

fig = plt.figure(figsize=(7,10))
```

```
ax1 = fig.add_subplot(211)
ax1.errorbar(Data[:, 0], residual, yerr=sigma, marker='.', linestyle='',
             label="residual (y-y_fit)")
ax1.hlines(0,np.min(Data[:, 0]),np.max(Data[:, 0]),lw=2,alpha=0.8)
ax1.set_xlabel('{} [{}]'.format(x_label,x_units))
ax1.set_ylabel('y-y_fit [{}]'.format(y_units))
ax1.legend(loc='best',numpoints=1)
ax2 = fig.add_subplot(212)
ax2.bar(bins[:-1],hist,width=bins[1]-bins[0])

ax2.set_ylim(0,1.2*np.max(hist))
ax2.set_xlabel('y-y_fit [{}]'.format(y_units))
ax2.set_ylabel('Number of occurences')

if sigma != None:
    within_err=100.*np.sum((residual<=sigma)&(residual>=-sigma))/len(residual)
    print ("\nResidual information:")
    print ('{:.1f}% of data points agree with fit'.format(within_err))
    ax2.vlines(-sigma,0,np.max(hist)*1.3,lw=3,color='r')
    ax2.vlines(+sigma,0,np.max(hist)*1.3,lw=3,color='r',label='+/- error')
    ax2.text(0.0,np.max(hist)*1.1,'{:.1f}% of data'.format(within_err),
             horizontalalignment='center',verticalalignment='center')
    ax2.text(np.min(bins),np.max(hist)*1.1,'{:.1f}% of data'.format(100-within_err),
             horizontalalignment='left',verticalalignment='center')
    ax2.text(np.max(bins),np.max(hist)*1.1,'{:.1f}% of data'.format(100-within_err),
             horizontalalignment='right',verticalalignment='center')
    ax2.legend(loc=(0.35,0.05))

print ('\nDisplaying plot 3')
plt.show()
```

Code D.2: Solid Sphere Fitting Code

```
import numpy as np
import matplotlib.pyplot as plt
from scipy.optimize import curve_fit as cf

def Periodic(distance, f0, omega, phase, resonance):
    return f0*np.sin(omega*distance+phase)+resonance

def chi_square (params, x, y, sigma):
    if sigma is None:
        sigma = 1.
    return np.sum((y-Periodic(x, *params))**2/sigma**2)

filename = 'C:/Users/ryank/Desktop/Work/Classes/Python/PHYS229/Acoustics/Data'
filename += "/AcousticSlottedSphereDataP5.csv"
x_label = 'Theta'
x_units = 'Degrees'
y_label = 'Change in Frequency'
y_units = 'Hz'
Data = np.loadtxt(filename, delimiter=',', comments='#', usecols=(0,1))
Data = Data[(int(len(Data)/2)):, :]
```

```
guess = [1.89, np.pi/180, -11.22, 585]
sigma = 0.11

###############################################################################

x_guess = np.linspace(min(Data[:, 0]),max(Data[:, 0]),500)
y_guess = Periodic(x_guess, *guess)

plt.errorbar(Data[:, 0], Data[:, 1], yerr=sigma, marker='.', linestyle='',
             label="measured data")
plt.plot(x_guess, y_guess,marker="", linestyle="-", linewidth=1, color="g",
         label="initial guess")
plt.xlabel('{} [{}]'.format(x_label, x_units))
plt.ylabel('{} [{}]'.format(y_label, y_units))
plt.title(r'Comparison between the data and the intial guess')
plt.legend(loc='best', numpoints=1)
print ('\nDisplaying plot 1')
plt.show()

###############################################################################

params, cov = cf(Periodic, Data[:, 0], Data[:, 1],
                 sigma=sigma*np.ones(len(Data[:, 0])), p0=guess, maxfev=10**5)

chi2 = chi_square(params, Data[:, 0], Data[:, 1], sigma)
dof = len(Data[:, 0]) - len(params)+1
print ("\nGoodness of fit - chi square measure:")
print ("Chi2 = {}, Chi2/dof = {}\n".format(chi2, chi2/dof))

cov = cov*dof/chi2
paramserr = np.sqrt(np.diag(cov))

param_names = ['f0','omega', 'phase','resonance']

print ("Fit parameters:")
for i in range(len(params)):
    print ('{} = {:.3e} +/- {:.3e}'.format(param_names[i], params[i],
                                           paramserr[i]))

###############################################################################

x_fit = np.linspace(min(Data[:, 0]),max(Data[:, 0]),len(Data[:, 0])*10)
y_fit = Periodic(x_fit,*params)

plt.errorbar(Data[:, 0], Data[:, 1], yerr=sigma, marker='.', linestyle='',
             label="measured data")
plt.plot(x_fit, y_fit, marker="", linestyle="-", linewidth=2, color="r",
         label=" fit")
plt.xlabel('{} [{}]'.format(x_label, x_units))
plt.ylabel('{} [{}]'.format(y_label, y_units))
plt.title(r'Relationship between Resonance and Degrees of Perturbation of Slotted Sphere')
plt.legend(loc='best', numpoints=1)
print ('\nDisplaying plot 2')
plt.show()
```

```
###############################################################################

y_fit=Periodic(Data[:, 0],*params)
residual = Data[:, 1]-y_fit
hist,bins = np.histogram(residual,bins=30)

fig = plt.figure(figsize=(7,10))
ax1 = fig.add_subplot(211)
ax1.errorbar(Data[:, 0], residual, yerr=sigma, marker='.', linestyle='',
             label="residual (y-y_fit)")
ax1.hlines(0,np.min(Data[:, 0]),np.max(Data[:, 0]),lw=2,alpha=0.8)
ax1.set_xlabel('{} [{}]'.format(x_label,x_units))
ax1.set_ylabel('y-y_fit [{}]'.format(y_units))
ax1.legend(loc='best',numpoints=1)
ax2 = fig.add_subplot(212)
ax2.bar(bins[:-1],hist,width=bins[1]-bins[0])

ax2.set_ylim(0,1.2*np.max(hist))
ax2.set_xlabel('y-y_fit [{}]'.format(y_units))
ax2.set_ylabel('Number of occurences')

if sigma != None:
    within_err=100.*np.sum((residual<=sigma)&(residual>=-sigma))/len(residual)
    print ("\nResidual information:")
    print ('{:.1f}% of data points agree with fit'.format(within_err))
    ax2.vlines(-sigma,0,np.max(hist)*1.3,lw=3,color='r')
    ax2.vlines(+sigma,0,np.max(hist)*1.3,lw=3,color='r',label='+/- error')
    ax2.text(0.0,np.max(hist)*1.1,'{:.1f}% of data'.format(within_err),
             horizontalalignment='center',verticalalignment='center')
    ax2.text(np.min(bins),np.max(hist)*1.1,'{:.1f}% of data'.format(100-within_err),
             horizontalalignment='left',verticalalignment='center')
    ax2.text(np.max(bins),np.max(hist)*1.1,'{:.1f}% of data'.format(100-within_err),
             horizontalalignment='right',verticalalignment='center')
    ax2.legend(loc=(0.35,0.05))

print ('\nDisplaying plot 3')
plt.show()
```

Code D.3: Slotted Sphere Fitting Code

Bibliography

[1] Lane, W. C. (2002). *The Wave Equation and Its Solutions*. Michigan State University.

[2] R. P. Feynman, R. B. Leighton, M. S. (1964). *The Feynman Lectures on Physics: Volume 1*, chapter Modes. Addison-Wesley.

[3] Staelin, D. (2009). *Electromagnetics and Applications*. Massachusetts Institute of Technology.