Development of an Energy-Based Sound Power Measurement Device

by

Joel Andrus

Submitted to the Department of Physics and Astronomy in partial fulfillment

of graduation requirements for the degree of

Bachelor of Science

Brigham Young University

April 2004

Advisor: Scott D. Sommerfeldt Capstone Coordinator: Branton S. Campbell

Department Chair: Scott D. Summerfeldt

CONTENTS

Development of an Energy-Based Sound Power Measurement Device

CHAPTER I. Introduction

1.1 Overview

Sound Power is a widely used quantity used to describe how much sound an acoustic source is radiating. It is more commonly used than Sound Pressure Level because it is independent of the measurement location. Sound Power, which is measured in watts (W), is often expressed on a logarithmic scale. This gives the Sound Power Level, which has the units of decibels (dB). Sound Power Level can be determined by finding the average Sound Pressure Level at several points in the far field surrounding the source, which can be found from several different ways. Since the square of the pressure in the far field is proportional to the intensity, it can be integrated to get sound power. Sound Power can also be determined by scanning over a surface with an intensity probe, and then integrating the intensity over the area scanned. Another possible way to measure sound power is with an energy-density probe. This is the method being investigated in this paper.

1.2 Three Current Methods

Currently, there are three basic methods for measuring Sound Power. The first one, which necessitates free field conditions (i.e. those found in an anechoic room), involves taking sound pressure measurements in the far field along a hemispherical surface surrounding the source. The second method assumes highly reflective boundary conditions, which can be found in a reverberation room. The Sound Power can then be measured directly as long as the total sound absorption of the room is known. Since the reflected sound is now considered diffuse, it is equally probable that the sound will come into the microphone from all directions, which allows the determination of the power using the pressure measured at a number of locations in the room. The third method measures the acoustic intensity directly with an intensity probe. By scanning over a closed surface enclosing the source, it is possible to estimate the total radiated power. While this method is more expensive, it can be applied to more general field conditions including the outdoors, which is important if what you are trying to measure is big. Determining which method is best takes into account several considerations. First, the size of the noise source should be considered. The reverberation room works best when the source is less than one percent of the volume of the room. The free field method places a 15 meter limit on the largest dimension of the source. The second issue is the character of the noise, in terms of it's frequency content. The reverberation method requires a frequency above that of the Schroeder frequency, which is the frequency limit for diffusion of sound waves in a room. Sources with a relatively low frequency content will require a large chamber to lower the Schroeder frequency. Third, the accuracy required is also an important consideration. Generally, the free field and reverberation methods will give the highest accuracy while the intensity measurements will give the least [1]. But greater accuracy requires greater measurement effort. The question also exists as to which test environment is suitable for measurements. If the source is small and movable, any method will work. However, if the source is installed outside, or has a significantly large volume (about two m3), then the reverberation and the free field methods may not be applicable.

1.3 Energy-based Quantities

Energy-based measurements could offer many advantages over current methods. For instance, they have the potential to provide simplified (possibly single-point) measurements in a laboratory reverberation room because they are much more global in

nature than pressure. Because spatial variations in the energy quantities should be much smaller than variations in pressure waves, they promise greater accuracy and less complexity in measuring the Sound Power. This leads to the capacity to decrease the cost involved in finding out the power a source is emanating. Another significant advantage of the proposed method is that energy-based methods could reduce the restrictions on reverberation room measurements by allowing sub-Schroeder frequencies (objects can radiate at a lower frequency in a smaller test room), and by potentially
allowing testing at the site where the source will be located. That would have a huge impact on current techniques and what researchers are able to afford in the testing of their equipment.

2. Interpolation of the Near-Field Pressure

The first step taken in this project was to compare energy-based measurements with regular pressure measurements to see if the number of measurements could be reduced at all in determining the pressure in the near-field.

To do this, a Matlab m file was created to produce exact pressure values at 1000 points equidistant from the source. Several ideal sources were modeled, including a dipole, a tesseral quadrupole, and an axial quadrupole. Least squares methods were then devised to produce the pressure in between measurements.

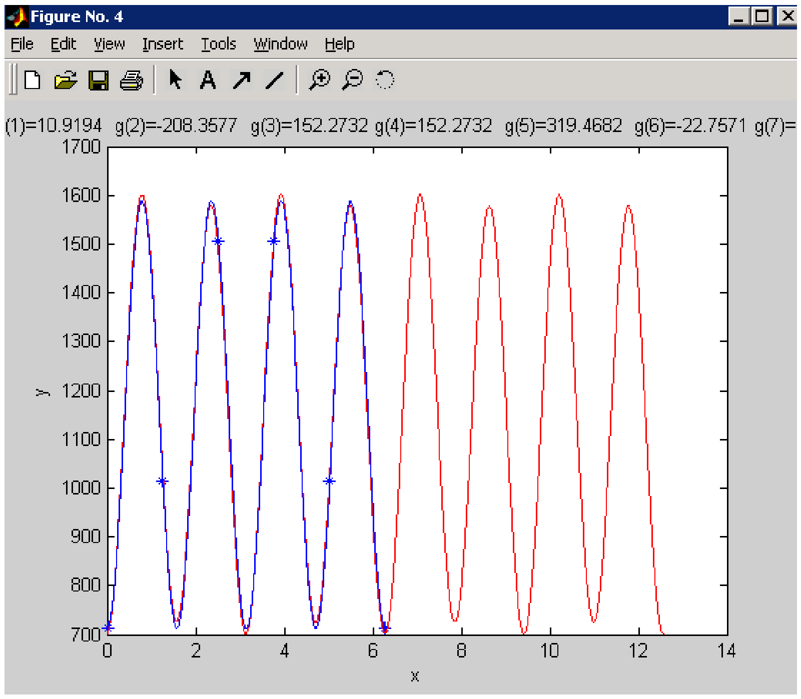
2.1 Ideal Source Derivations

Some concerns were expressed regarding the assumptions made in the derivation of the ideal sources. In the tesseral quadrupole, it is assumed that the distance between the four sources is small, so you can use a Taylor expansion for the Green's function describing the medium. This allows several higher order terms to drop out. However, the error for this approximation was generally less than ten dB at 2.5 centimeters from the source. Since the actual distance would be closer to half a meter, it was considered negligible. So the expressions containing this assumption were used.

2.2 Least Squared Pressure

At this point, another m file was created that would take part of those exact pressure measurements, and then use a least squares method to interpolate the pressure between them. Using this technique, it was determined that using a combination of pressure and angular velocity would produce the same fit as pressure measurements alone, with half the measurements. The only drawback is that velocity measurements require more microphones. So, in reality, the same amount of information was being collected, there was just fewer measurements.

In analyzing these methods, ideal sources were used to form a basis for error. These are easily programmed into Matlab. The fits that were obtained for Sound Pressure, Angular Velocity, Radial Velocity and Energy Density are extremely good, even for an ideal Tesseral source. Figure 1 shows one of these fits for Energy Density emanating from a Tesseral source. The theoretical result was plotted from 0-4π to show the wave pattern better. These fits are obtained



```
(1)=10.9194  g(2)=-208.3577  g(3)=152.2732  g(4)=152.2732  g(5)=319.4682  g(6)=-22.7571  g(7)=
```

Figure 1. The far field extension of Energy Density (blue) with theoretical values (red).theoretical profile (red), with the fit obtained from the Near field extension (blue)

from a combination of 4 Matlab scripts which can be found in Appendix A. (datafit.m, funcfit.m, leastsq.m, Tessercheckl.m)

Excel was also used to fit the curves by another student working on the project. Eventually, the best fit in Excel was given with a splines type function. Results from the two different programs are comparable. Using these methods allowed us to cut the number of measurements in half, which is what we were looking for.

3. Extension to the Far Field with Huygens Principle

Once a proper fit was obtained, it was attempted to extend the pressure field outward. Two possible approaches have been suggested. The first is based on the well known Huygens's principle. The second is called Near-Field Acoustic Holography (NAH), which was developed Dr. Earl Williams. For my involvement with the research, it was decided to focus on Huygens's principle to determine the pressure field.

Matlab was used to model the waves being propagated outward to the far-field. This proved to be a daunting task because the waves propagate outward in 3-D. The code has to be able to distinguish the direction of each wave front and then the direction of subsequent wave fronts of sound off of those. It also needs to be able to determine if the sphere will block the wave so that it does not inappropriately add pressure to the end result. So the model must choose only those parts that are contributing to the point they are measuring. The code at this point assumes a dipole source for simplicity so that it is straight forward to see if the propagation to the far field is correct. Currently, work on this part of the project has stopped because it was determined that the NAH approach would be more suitable for the far field reconstruction.

4. Sensor Development

NASA recently awarded a grant to BYU, in conjunction with Larson-Davis, a local company, to develop a new energy density measurement probe. Among other applications, this probe could be used to measure Sound Power. Larson–Davis is developing the new probe based on the results from our research. They will then work with us to validate the performance of the probes.

The standard energy-based probe that is used in acoustic measurements today is a six microphone arrangement as can be seen in Fig. 2. There are two microphones on each axis, one in the negative direction and one in the positive direction. The current design for this probe is extremely fragile and expensive, with prices ranging in the thousands of dollars.

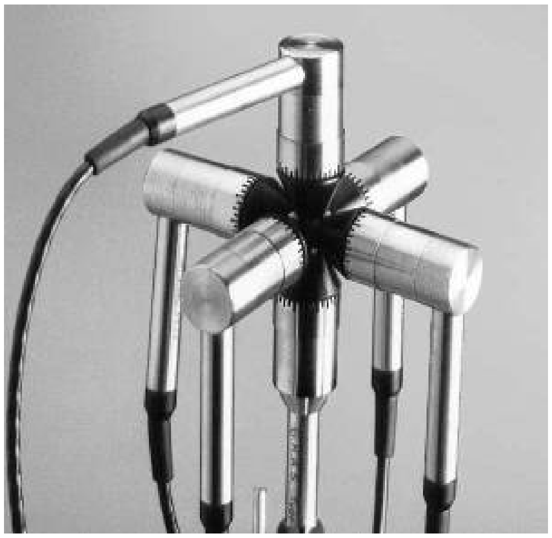


Figure 2--Sound Energy Probe

Currently, two new probe configurations have been proposed by the BYU acoustics research group. Both use an array of four microphones circumscribed in a solid sphere. The radius of these has not been definitely determined yet, but the prototypes have a radius of 1 and 0.313 inches. A larger probe is desired for the lower frequency range, and the smaller probe is needed for higher frequencies (up to about 6kHz). One advantage of the spherical probe is that it improves some of the bias errors.

One configuration investigated is an orthogonal arrangement with all four microphones in one half of the sphere, which will be referred to hereafter as the ortho probe and can be seen in Fig 3. The other is a tetrahedron arrangement, which will hereafter be referred to as the tetra probe and can be seen in Fig 4.

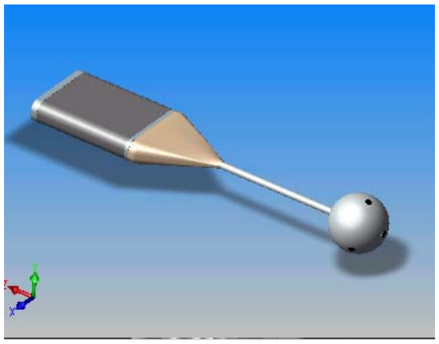


Figure 3--Orthogonal Probe

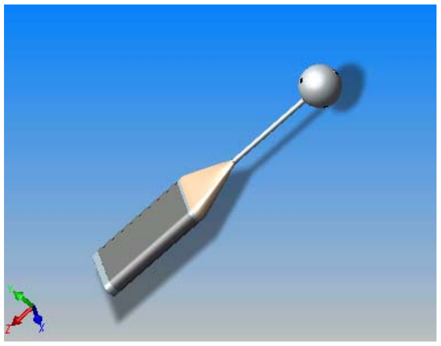


Figure 4—Tetrahedron Probe

4.1 Orthogonal Array

The time over the summer of 2003 was used to determine the scattering of the sound from the spherical surface of the sphere. The equation for scattering from a sphere involves several spherical Bessel functions, as well as a Legendre polynomial. The equation below defines a plane wave traveling to the right along the polar axis:

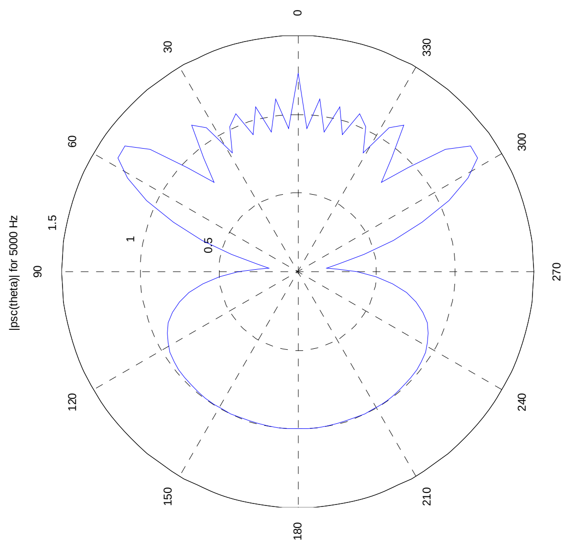
$$p_p = Ae^{ik(r\cos\theta - ct)} = A\sum_{m=0}^{\infty}(2m+1)i^m P_m(\cos\theta)\, j_m(kr)e^{-2\pi i\nu t}$$

where $A=\sqrt{\rho_o cI}$, $\rho_o = 123$, c=343 m/s, and $I = \frac{|p|^2}{\rho_o c}$. Plugging I into A results in the realization that $A=|p|$, meaning that the amplitude of the wave is equal to the magnitude of the pressure for the sound wave. P_m is a Legendre Polynomial of order m, and j_m is a spherical Bessel function of order m [2]. The expression for the wave scattered from a sphere of radius a with its center at the origin is:

$$p_s = -A\sum_{m=0}^{\infty}(2m+1)i^{m+1}e^{-i\delta_m}\sin\delta_m P_m(\cos\theta)[j_m(kr)+in_m(kr)]e^{-2\pi i v t}$$

$$\text{where } \delta_m = \tan^{-1}\left(\frac{(m+1)j_m-(m)j_m}{(m)y_m-(m+1)y_m}\right)$$

This equation was programmed into Matlab to study the effect of radius and frequency on scattering. Knowing the scattering is necessary to determine the actual sound pressure at each microphone of the probe. The m files for this are: spherbesselj.m, spherbessely.m, and scatteredpressure.m, which can be found in the appendix. Figures 5 and 6 show the results of the Matlab model for the scattering of sound on a probe with a radius of 1 inch. The sound wave of 5000 Hz in Fig. 6 approaches from the bottom to the top.



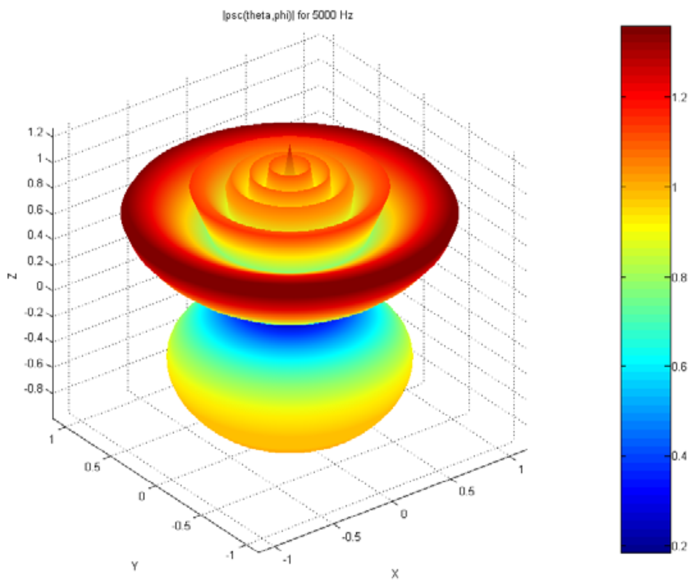


Figure 5. Polar plot of the scattered pressure from a plane wave approaching from the left.

Figure 6. Surface plot of the scattered pressure from a sphere, with the incident plane wave approaching from the bottom.

The Matlab model that determines the pressure measured by the ortho probe, currently gives an error of less that 2 dB at 5000 Hz. The ortho probe is being developed first in this project. It's currently being tested for accuracy by Lance Locey.

4.2 Tetrahedron Array

The Tetra probe is not being developed yet. There is a prototype for it, but no calibration has been attempted due to the work being focused on the ortho probe at this point.   All of the work done for the scattering of the waves off of the probe still applies just the same. The only difference is how the probe determines what the velocity at each microphone is.

5. Conclusion

In this project I was exposed to several different aspects of research. First, I had to identify the problem. This occurred with each step of the research. If you don't know where you need to go, you won't get there. I also had to learn everything that I could about that subject, so that I could be productive with my research. I also had to learn how to use numerical models to verify the theoretical models. This was the hardest step for me, as things often made sense in thinking about them, but it was difficult to make the computer reproduce it. Working and talking things through was the most beneficial part of research. Many times I found answers to my questions in trying to explain the problem to someone else.

The new probe will be a great asset to those who need to make a sound power measurement quickly and without having to scan over the surface to get the intensity. This probe reduces the number of measurements by half with very little error. It is also a great improvement in design and functionality over current sound-energy probes.

6. References

[1] C. Harris, *Acoustical Measurements & Noise Control*, 2nd Edition, (McGraw Hill) pp.13.1-6

[2] P. Morse and K. U. Ingard, *Theoretical Acoustics*, 3rd Edition, (Princeton University Press) pp.419

Appendix (Matlab code)

datafit.m

```
%this file will do the least squares fit to find the coefficients for Pressure,
% Radial Velocity, Angular Velocity, and Energy Density.  It then saves those
% coefficients to a distinct file for later use.
clear; close all;
global R w rho k
npts=6;

%load data points from Pressure
load presspoints.xls;
x=presspoints(:,1);y=presspoints(:,2);
xmin=min(x);
xmax=max(x);
dx=(xmax-xmin)/(npts-1);
xplot=xmin:dx:xmax;
% ifit=0;
% while ifit==0
%     disp('Enter an initial guess for the function parameters')
%
a=input('[a(1).*sin(4.*x)+a(2).*cos(4.*x)+a(3).*sin(2.*x)+a(4).*cos(2.*x)+a(5).*(sin(x)).
^2+a(6).*(cos(x)).^2+a(7).*cos(x).*sin(x)+a(8)] in vector form [...]-\n\n')
%     yplot=funcfit(a,xplot);
%     %plots the data and the intial function guess
%     plot(x,y,'b*',xplot,yplot,'r-')
%     xlabel('x')
%     ylabel('y')
%     title('Pressure Points Least Squares Fit')
%     ifit=input('Enter 0 to try again, 1 to try to fit -')
% end
%with the option TolX set, fminsearch will adjust (a) until each of its elements is
determined to within TolX.
a=[1 1 1 1 1 1 1 1 1 1];
option=optimset('TolX',1e-7);
a=fminsearch(@leastsq,a,option,x,y);
B=0:2*pi/999:4*pi;
yplot=funcfit(a,B);
%Now we plot the data and the final function fit
plot(x,y,'b*',B,yplot,'r-');
xlabel('x');
ylabel('y');
q=a;
Y=['N=' num2str(length(y)) ', Tess Pressure where kR=' num2str(k*R) '--a(1)= '
num2str(q(1)) '  a(2)=' num2str(q(2)) '  a(3)=' num2str(q(3)) ' a(4)=' num2str(q(3)) '
a(5)=' num2str(q(5)) '  a(6)=' num2str(q(6)) ' a(7)=' num2str(q(7)) '  a(8)='
num2str(q(8))];
```

```
title(Y);
save pressurecoefficients a

hold on
load realpresspoints.xls;
x=realpresspoints(:,1);y=realpresspoints(:,2);
xmin=min(x);
xmax=max(x);
nnpts=1000;
dx=(xmax-xmin)/(nnpts-1);
xplot=xmin:dx:xmax;
plot(x,y,'b');
hold off;
figure(1);
figure;

%Radial Velocity

load rvpoints.xls;
x=rvpoints(:,1);y=rvpoints(:,2);
xmin=min(x);
xmax=max(x);
dx=(xmax-xmin)/(npts-1);
xplot=xmin:dx:xmax;
b=input('[b(1).*sin(4.*x)+b(2).*cos(4.*x)+b(3).*sin(2.*x)+b(4).*cos(2.*x)+b(5).*(sin(x))
.^2+b(6).*(cos(x)).^2+b(7).*cos(x).*sin(x)+b(8)] in vector form [...]-\n\n')
b=[1 1 1 1 1 1 1 1 1 1];
%with the doption TolX set, fminsearch will adjust (a) until each of its elements is
determined to within TolX.
option=optimset('TolX',1e-7);
b=fminsearch(@leastsq,b,option,x,y);
B=0:2*pi/999:4*pi;
yplot=funcfit(b,B);
%Now we plot the data and the final function fit
plot(x,y,'b*',B,yplot,'r-');
xlabel('x');
ylabel('y');
q=b;
Y=['Radial velocity-kR=' num2str(k*R) '-b(1)=' num2str(q(1)) '  b(2)=' num2str(q(2)) '
b(3)=' num2str(q(3)) ' b(4)=' num2str(q(3)) '  b(5)=' num2str(q(5)) '  b(6)=' num2str(q(6))
' b(7)=' num2str(q(7)) '  b(8)=' num2str(q(8))];
title(Y);
save rvcoefficients b

hold on
load realrvpoints.xls;
x=realrvpoints(:,1);y=realrvpoints(:,2);
xmin=min(x);
```

```
xmax=max(x);
nnpts=1000;
dx=(xmax-xmin)/(nnpts-1);
xplot=xmin:dx:xmax;
plot(x,y,'b');
hold off;
figure(2);
figure

% Angular velocity
load tvpoints.xls;
x=tvpoints(:,1);y=tvpoints(:,2);
xmin=min(x);
xmax=max(x);
dx=(xmax-xmin)/(npts-1);
xplot=xmin:dx:xmax;
e=input('[e(1).*sin(4.*x)+e(2).*cos(4.*x)+e(3).*sin(2.*x)+e(4).*cos(2.*x)+e(5).*(sin(x)).
^2+e(6).*(cos(x)).^2+e(7).*cos(x).*sin(x)+e(8)] in vector form [...]-\n\n')
e=[1 1 1 1 1 1 1 1 1 1];
option=optimset('TolX',1e-7);
e=fminsearch(@leastsq,e,option,x,y);
B=0:2*pi/999:4*pi;
yplot=funcfit(e,B);
plot(x,y,'b*',B,yplot,'r-');
xlabel('x');
ylabel('y');
q=e;
Y=['Angular Velocity-kR=' num2str(k*R) '-e(1)=' num2str(q(1)) '  e(2)=' num2str(q(2)) '
e(3)=' num2str(q(3)) ' e(4)=' num2str(q(3)) '  e(5)=' num2str(q(5)) '  e(6)=' num2str(q(6)) '
e(7)=' num2str(q(7)) '  e(8)=' num2str(q(8))];
title(Y);
save tvcoefficients e

hold on
load realtvpoints.xls;
x=realtvpoints(:,1);y=realtvpoints(:,2);
xmin=min(x);
xmax=max(x);
nnpts=1000;
dx=(xmax-xmin)/(nnpts-1);
xplot=xmin:dx:xmax;
plot(x,y,'b');
hold off;
figure(3);
figure

% I had to scale energy density by a factor of 1 million in order to get it to work.
% Energy Density
```

```
load edpoints.xls;
x=edpoints(:,1);y=edpoints(:,2);
xmin=min(x);
xmax=max(x);
dx=(xmax-xmin)/(npts-1);
xplot=xmin:dx:xmax-dx;
g=input('[g(1).*sin(4.*x)+g(2).*cos(4.*x)+g(3).*sin(2.*x)+g(4).*cos(2.*x)+g(5).*(sin(x))
.^2+g(6).*(cos(x)).^2+g(7).*cos(x).*sin(x)+g(8)] in vector form [...]-\n\n')
% with the option TolX set, fminsearch will adjust (a) until each of its elements is
determined to within TolX.
g=[1 1 1 1 1 1 1 1 1 1];
option=optimset('TolX',1e-7);
g=fminsearch(@leastsq,g,option,x,y);
B=0:2*pi/999:4*pi;
yplot=funcfit(g,B);
% Now we plot the data and the final function fit
plot(x,y,'b*',B,yplot,'r-');
xlabel('x');
ylabel('y');
q=g;
Y=['Energy Density-kR=' num2str(k*R) '-g(1)=' num2str(q(1)) '  g(2)=' num2str(q(2)) '
g(3)=' num2str(q(3)) ' g(4)=' num2str(q(3)) '  g(5)=' num2str(q(5)) '  g(6)=' num2str(q(6))
' g(7)=' num2str(q(7)) '  g(8)=' num2str(q(8))];
title(Y);
save edcoefficients g

hold on
load realedpoints.xls;
x=realedpoints(:,1);y=realedpoints(:,2);
xmin=min(x);
xmax=max(x);
nnpts=1000;
dx=(xmax-xmin)/(nnpts-1);
xplot=xmin:dx:xmax;
plot(x,y,'b');
hold off;
figure(4);
```

funcfit.m

```
function f=funcfit(a,x)
%Evaluates the function that is to be fit to the data
f=a(1).*sin(4.*x)+a(2).*cos(4.*x)+a(3).*sin(2.*x)+a(4).*cos(2.*x)+a(5).*(sin(x)).^2+a(6
).*(cos(x)).^2+a(7).*cos(x).*sin(x)+a(8).*(cos(x)).^2.*(sin(x)).^2+a(9).*cos(2.*x).^2+a(1
0);
```

leastsq.m

```
function s=leastsq(d,x,y)
s=sum((y-funcfit(d,x)).^2);
```

Tessercheckl.m

```
close all;
clear;
global R w rho k
c=343;
rho=1.21;
% Q=input('Please type in the value of Q\n\n');
% d=input('Please type in the value of d (distance between sources)\n\n');
% f=input('Please type in the value of f\n\n');
% R=input('Please type in the value of R\n\n');
R=.163;
f=100;
Q=1;
d=.01;
k=2*pi*f/c;x0=.5*d;
Theta=0:2*pi/5 :2*pi;
theta=0:2*pi/999 :2*pi;

%Makes Pressure points from a Tesseral Quadrupole
M=-j.*k.^3.*d.^2.*rho.*c.*Q.*((cos(Theta)).*sin(Theta))./(4.*pi).*[1-3./(k.*R).^2-
(j.*3)./(k.*R)].*exp(j.*(-k.*R));
polar(Theta,M,'-r');
T=['Pressure of Quadrupole where kR=' num2str(k*R)];
title(T);
e=zeros(2,6);
e(1,:)=Theta;
e(2,:)=M;
e=e';
dlmwrite('presspoints.xls',e,'\t');
figure(1);
figure

%Makes Radial Velocity points from a Tesseral Quadrupole
W=-j.*k.^3.*d.^2.*Q./(4.*pi.*R).*cos(Theta).*sin(Theta).*(1-9/(k^2.*R.^2)-
j.*4./(k.*R)+j.*9./(k.^3.*R.^3))*exp(-j*k.*R);
polar(Theta,W,'-k');
T=['Radial Velocity of Quadrupole where kR=' num2str(k*R)];
title(T);
e=zeros(2,6);
e(1,:)=Theta;
e(2,:)=W;
e=e';
dlmwrite('rvpoints.xls',e,'\t');
figure
```

```
%Makes Angular Velocity points from a Tesseral Quadrupole
S=k^2*d.^2.*Q./(4*pi*R.^2).*(2.*(sin(Theta)).^2-1).*(1-3.*j./k./R-
3./k.^2./R.^2).*exp(j.*(-k.*R));
polar(Theta,S,'-g');
T=['Angular Velocity of Quadrupole where kR=' num2str(k*R)];
title(T);
e=zeros(2,6);
e(1,:)=Theta;
e(2,:)=S;
e=e';
dlmwrite('tvpoints.xls',e,'\t');
figure

% Notice the factor of 1 million in the energy density.  That was necessary to get a good
fit.
%Makes Energy Density points from a Tesseral Quadrupole
K=d.^4.*Q.^2.*rho.*k.^4./(32.*pi.^2.*(R).^2).*(k^2.*(cos(Theta)).^2.*(sin(Theta)).^2.*(
2+1./(k.*R).^2+18./(k.*R).^4+81./(k.*R).^6));
U=d.^4.*Q.^2.*rho.*k.^4./(32.*pi.^2.*(R).^2).*((cos(2*Theta)).^2./(R).^2).*(1+3./(k.*R)
.^2+9./(k.*R).^4);
r=1000000*(K+U);
polar(Theta,r,'-b')
T=['Energy Density of Quadrupole where kR=' num2str(k*R)];
title(T);
e=zeros(2,6);
e(1,:)=Theta;
e(2,:)=r;
e=e';
dlmwrite('edpoints.xls',e,'\t');

%Create a set of a thousand points to check it against

M=-j.*k.^3.*d.^2.*rho.*c.*Q.*((cos(theta)).*sin(theta))./(4.*pi).*[1-3./(k.*R).^2-
(j.*3)./(k.*R)].*exp(j.*(-k.*R));
e=zeros(2,1000);
e(1,:)=theta;
e(2,:)=M;
e=e';
dlmwrite('realpresspoints.xls',e,'\t');

W=-j.*k.^3.*d.^2.*Q./(4.*pi.*R).*cos(theta).*sin(theta).*(1-9/(k^2.*R.^2)-
j.*4./(k.*R)+j.*9./(k.^3.*R.^3))*exp(-j*k.*R);
e=zeros(2,1000);
e(1,:)=theta;
e(2,:)=W;
e=e';
dlmwrite('realrvpoints.xls',e,'\t');
```

```
S=k^2*d.^2.*Q./(4*pi*R.^2).*(2.*(sin(theta)).^2-1).*(1-3.*j./k./R-
3./k.^2./R.^2).*exp(j.*(-k.*R));
e=zeros(2,1000);
e(1,:)=theta;
e(2,:)=S;
e=e';
dlmwrite('realtvpoints.xls',e,'\t');

K=d.^4.*Q.^2.*rho.*k.^4./(32.*pi.^2.*(R).^2).*(k^2.*(cos(theta)).^2.*(sin(theta)).^2.*(2
+1./(k.*R).^2+18./(k.*R).^4+81./(k.*R).^6));
U=d.^4.*Q.^2.*rho.*k.^4./(32.*pi.^2.*(R).^2).*((cos(2*theta)).^2./(R).^2).*(1+3./(k.*R).
^2+9./(k.*R).^4);
E=1000000*(K+U);
e=zeros(2,1000);
e(1,:)=theta;
e(2,:)=E;
e=e';
dlmwrite('realedpoints.xls',e,'\t');
```

Scatteredpressure.m

```
close all;
clear;
global R w rho k
c=343;
rho=1.21;
% Q=input('Please type in the value of Q\n\n');
% d=input('Please type in the value of d (distance between sources)\n\n');
% f=input('Please type in the value of f\n\n');
% R=input('Please type in the value of R\n\n');
R=.163;
f=100;
Q=1; d=.01;
k=2*pi*f/c;x0=.5*d;
Theta=0:2*pi/5 :2*pi;
theta=0:2*pi/999 :2*pi;

%Makes Pressure points from a Tesseral Quadrupole
M=-j.*k.^3.*d.^2.*rho.*c.*Q.*((cos(Theta)).*sin(Theta))./(4.*pi).*[1-3./(k.*R).^2-
(j.*3)./(k.*R)].*exp(j.*(-k.*R));
polar(Theta,M,'-r');
T=['Pressure of Quadrupole where kR=' num2str(k*R)];
title(T);
e=zeros(2,6);
e(1,:)=Theta;
e(2,:)=M;
e=e';
```

```
dlmwrite('presspoints.xls',e,'\t');
figure(1);
figure

%Makes Radial Velocity points from a Tesseral Quadrupole
W=-j.*k.^3.*d.^2.*Q./(4.*pi.*R).*cos(Theta).*sin(Theta).*(1-9/(k^2.*R.^2)-
j.*4./(k.*R)+j.*9./(k.^3.*R.^3))*exp(-j*k.*R);
polar(Theta,W,'-k');
T=['Radial Velocity of Quadrupole where kR=' num2str(k*R)];
title(T);
e=zeros(2,6);
e(1,:)=Theta;
e(2,:)=W;
e=e';
dlmwrite('rvpoints.xls',e,'\t');
figure

%Makes Angular Velocity points from a Tesseral Quadrupole
S=k^2*d.^2.*Q./(4*pi*R.^2).*(2.*(sin(Theta)).^2-1).*(1-3.*j./k./R-
3./k.^2./R.^2).*exp(j.*(-k.*R));
polar(Theta,S,'-g');
T=['Angular Velocity of Quadrupole where kR=' num2str(k*R)];
title(T);
e=zeros(2,6);
e(1,:)=Theta;
e(2,:)=S;
e=e';
dlmwrite('tvpoints.xls',e,'\t');
figure

% Notice the factor of 1 million in the energy density.  That was necessary to get a good
fit.
%Makes Energy Density points from a Tesseral Quadrupole
K=d.^4.*Q.^2.*rho.*k.^4./(32.*pi.^2.*(R).^2).*(k^2.*(cos(Theta)).^2.*(sin(Theta)).^2.*(
2+1./(k.*R).^2+18./(k.*R).^4+81./(k.*R).^6));
U=d.^4.*Q.^2.*rho.*k.^4./(32.*pi.^2.*(R).^2).*((cos(2*Theta)).^2./(R).^2).*(1+3./(k.*R)
.^2+9./(k.*R).^4);
r=1000000*(K+U);
polar(Theta,r,'-b')
T=['Energy Density of Quadrupole where kR=' num2str(k*R)];
title(T);
e=zeros(2,6);
e(1,:)=Theta;
e(2,:)=r;
e=e';
dlmwrite('edpoints.xls',e,'\t');
```

```
%Create a set of a thousand points to check it against

M=-j.*k.^3.*d.^2.*rho.*c.*Q.*((cos(theta)).*sin(theta))./(4.*pi).*[1-3./(k.*R).^2-
(j.*3)./(k.*R)].*exp(j.*(-k.*R));
e=zeros(2,1000);
e(1,:)=theta;
e(2,:)=M;
e=e';
dlmwrite('realpresspoints.xls',e,'\t');

W=-j.*k.^3.*d.^2.*Q./(4.*pi.*R).*cos(theta).*sin(theta).*(1-9/(k^2.*R.^2)-
j.*4./(k.*R)+j.*9./(k.^3.*R.^3))*exp(-j*k.*R);
e=zeros(2,1000);
e(1,:)=theta;
e(2,:)=W;
e=e';
dlmwrite('realrvpoints.xls',e,'\t');

S=k^2*d.^2.*Q./(4*pi*R.^2).*(2.*(sin(theta)).^2-1).*(1-3.*j./k./R-
3./k.^2./R.^2).*exp(j.*(-k.*R));
e=zeros(2,1000);
e(1,:)=theta;
e(2,:)=S;
e=e';
dlmwrite('realtvpoints.xls',e,'\t');

K=d.^4.*Q.^2.*rho.*k.^4./(32.*pi.^2.*(R).^2).*(k^2.*(cos(theta)).^2.*(sin(theta)).^2.*(2
+1./(k.*R).^2+18./(k.*R).^4+81./(k.*R).^6));
U=d.^4.*Q.^2.*rho.*k.^4./(32.*pi.^2.*(R).^2).*((cos(2*theta)).^2./(R).^2).*(1+3./(k.*R).
^2+9./(k.*R).^4);
E=1000000*(K+U);
e=zeros(2,1000);
e(1,:)=theta;
e(2,:)=E;
e=e';
dlmwrite('realedpoints.xls',e,'\t');
```

Sphericalbesselj.m

```
function f=spherbesselj(n,x)
f=sqrt(pi./2./x).*besselj(n+1/2,x);
```

Sphericalbessely.m

```
function f=spherbessely(n,x)
f=sqrt(pi./2./x).*bessely(n+1/2,x);
```