

# Linear Algebra and Music

Derrick Smith<sup>1</sup>

## 1. Introduction

In this project you will see how to use linear algebra to understand music and other types of sound. Specifically, you will see that a given sound can be viewed as elements of a linear space and its coordinates relative to a carefully chosen orthonormal basis will explain many different properties of the sound. After completing this project, you will be able to answer the following questions.

- What is a good basis for the space of all sounds?
- What are musical notes?
- How do notes make up a song?
- Why do some notes sound pleasing when played together, while others do not?
- Why do pianos and flutes sound different even when playing the same note?

## 2. Why Sines and Cosines?

A basis for a linear space is a fundamental set of building blocks that can be used to make any element in the space. To analyze sound and music, we need to find a set of basic sounds that can be combined to make all other sounds. To do this, we first look at how sounds travel through the air and what your ear does when it receives that sound.

All sounds are produced by vibrations which cause variations in air pressure to propagate. If you hold your fingers against your throat when you speak, you can feel your larynx vibrate. When a bow is pulled across a flute string, the string vibrates. You have also probably felt these vibrations at a concert or when standing next to a loud speaker.

These variations in air pressure travel from the source of the sound to your ear where they are processed and then sent to your brain. How the ear processes sound is not completely understood, but we do know the basic story. The variations in air pressure cause your eardrums to vibrate which causes some liquid in your inner ear to slosh around. This liquid surrounds a hair-lined membrane and is enclosed in a tapered chamber. Different variations in air pressure cause differently shaped waves to propagate through the liquid. Because the chamber containing the membrane is tapered, some waves travel further than others along the membrane and stimulate different hairs. These hairs are connected to neurons that transmit the information to your brain.

A crude model of what's happening to a point on the membrane is given by the

differential equation  $\frac{d^2 y}{dt^2} = -k y$  where  $t$  is time and  $y$  is the distance of that point on the

---

<sup>1</sup> Question? Comment? Suggestion? Please send me an email at [dexsmith@gmail.com](mailto:dexsmith@gmail.com)

membrane from its equilibrium solution. The solutions to this differential equation give us the basic building blocks to understand all sounds.

**Problem:**

**2a.** Verify that  $\cos(\sqrt{k}t)$  and  $\sin(\sqrt{k}t)$  are solutions to the differential

equation  $\frac{d^2y}{dt^2} = -ky$ .

In your Differential Equations course, you will see that every solution to the differential equation above is a linear combination of  $\cos(\sqrt{k}t)$  and  $\sin(\sqrt{k}t)$ . A proof is also sketched in 4.2 #58 in your text. In the language of linear algebra, they form a basis for the space of solutions to the differential equation.

Because the solutions to the differential equation are sine and cosine, you will use sine waves and cosine waves will to analyze sounds and music in the rest of this lab.

### 3. How the shape of the graph affects what you hear.

To start, you will graph and then listen to various sine waves. You will see the differences in their graphs and then hear the differences when you listen to them in MATLAB.

First, you are going to use MATLAB to graph and then to play two different sounds. You will listen to two seconds of each of the functions  $\sin(2\pi \cdot 440t)$  and  $\sin(2\pi \cdot 880t)$ . The first function represents a vibration at a rate of 440 cycles per second and the second at 880 cycles per second.<sup>2</sup>

Here is how to use MATLAB to plot the two graphs in the same window:

```
>> % First we set the domain. Here it is [0,2] with 16000
>> % sample points.
>> t = linspace(0,2 ,16000)'; %Note the ' at the end
>> %
>> % Now define the two functions.
>> sound1 = sin(2*pi*440*t); %define the first function
>> sound2 = sin(2*pi*880*t); %define the second function
>> %
>> % Plot the two functions. Note we only plot the first
>> % 1/100th of a second.
>> subplot(2,1,1); plot(t,sound1); axis([0,.01,-1, 1])
>> subplot(2,1,2); plot(t,sound2); axis([0,.01,-1, 1])
```

---

<sup>2</sup> The unit cycles per second occurs often enough to warrant its own name: Hertz

Here is how to use MATLAB to listen to  $\sin(2\pi \cdot 440t)$ :

```
>> soundsc(sound1,8000) %The 8000 is needed to tell the
>> %soundsc command the sampling frequency. Here there are
>> % 8000samples per second.
```

### Problems

**3a.** Plot  $\sin(2\pi \cdot 440t)$  and  $\sin(2\pi \cdot 880t)$  on the interval  $[0, .01]$  on two separate graphs in the same window. What differences do you see between the two graphs? Include the two graphs in your write up.

**3b.** Listen to two seconds of  $\sin(2\pi \cdot 440t)$  and to two seconds of  $\sin(2\pi \cdot 880t)$ . What is the difference between the two sounds you hear?

**3c.** Plot  $\sin(2\pi \cdot 440t)$  and  $.25\sin(2\pi \cdot 440t)$  on the interval  $[0, .01]$  on two separate graphs in the same window. What differences do you see between the two graphs? Include the two graphs in your write up.

**3d.** Listen to two seconds of  $\sin(2\pi \cdot 440t)$  and to two seconds of  $.25\sin(2\pi \cdot 440t)$ . What is the difference between the two sounds you hear? Use the command `>>sound` instead of the command `>>soundsc` for this problem.

**3e.** Plot  $\sin(2\pi \cdot 440t) + \cos(2\pi \cdot 660t)$  and  $\sin(2\pi \cdot 440t) + \sin(2\pi \cdot 660t)$  on the interval  $[0, .01]$ . What differences do you see between the two graphs? Include the two graphs in your write up.

**3f.** Listen to two seconds of  $\sin(2\pi \cdot 440t) + \cos(2\pi \cdot 660t)$  and to two seconds of  $\sin(2\pi \cdot 440t) + \sin(2\pi \cdot 660t)$ . What is the difference (if any) between the two sounds you hear? Your result here will be important later in this lab. Be careful when you answer this.

**3g.** A more realistic function to model musical notes is  $e^{-5t} \sin(2\pi \cdot 440t)$ . (This is the solution to another differential equation that models the ear better than the one above.)

Plot  $\sin(2\pi \cdot 440t)$  and  $e^{-5t} \sin(2\pi \cdot 440t)$  on the interval  $[0, .5]$  on two separate graphs in the same window. What differences do you see between the two graphs? Include the two graphs in your write up.

(Hint: `>> note1 = exp(-5*t).*sin(2*pi*440*t);`)

**3h.** Listen to one-half second of  $\sin(2\pi \cdot 440t)$  and to one-half second of  $e^{-5t} \sin(2\pi \cdot 440t)$ . What is the difference between the two sounds you hear? Explain why the second function sounds more realistic. What role does  $e^{-5t}$  play? You will need to redefine  $t$  for this problem.

## 4. Songs

A song is a sequence of notes. In this section, you will use MATLAB to play a song. Use the following table of frequencies to play the famous children's song given by the sequence of notes B A G A B B B.

Note	Frequency
G	392
A	440
B	494

MATLAB trick: There is an easy way to make your sequence of notes. Suppose you have already defined the notes A and B. To play the song A B B A, you would use

```
>> soundsc([A B B A], 8000);
```

### Problem

**4a.** Show the MATLAB commands you used to play the song. Do you recognize it? Be sure to use the more realistic notes you generated in **3g** above.

## 5. Consonance and Dissonance

Most music you hear does not consist of just one frequency played at a time. Instead, multiple frequencies enter your ears simultaneously. We can use sine waves to explain why some combinations of frequencies are more pleasing than others.

Translating music to math, two notes are said to be an octave apart if the frequency of the higher pitched note is twice the frequency of the lower pitched note. For example, the two notes represented by  $e^{-2t} \sin(2\pi \cdot 440t)$  and  $e^{-2t} \sin(2\pi \cdot 880t)$  are an octave apart. More generally, anytime the ratio of the frequencies is a ratio of two small integers, the notes combine to make a pleasant sound. In the language of music, they are said to be consonant. The ratio of the frequencies of the notes represented by  $e^{-2t} \sin(2\pi \cdot 440t)$  and

$e^{-2t} \sin(2\pi \cdot 660t)$  is 3:2, so these notes are consonant. The notes  $e^{-2t} \sin(2\pi \cdot 440t)$  and  $e^{-2t} \sin(2\pi \cdot 450t)$  are dissonant (not pleasing to the ear) because the ratio of the frequencies is 45:44. In the problems below, you will graph and listen to various combinations of these notes. Make each note last for two seconds.

### Problems

**5a.** Plot  $e^{-2t} \sin(2\pi \cdot 440t)$  and  $e^{-2t} \sin(2\pi \cdot 880t)$  on the interval  $[0, .004]$  on two separate graphs in the same window. What differences do you see between the two graphs? Include the two graphs in your write up.

**5b.** Do the following:

- 1) Listen to two seconds of  $e^{-2t} \sin(2\pi \cdot 440t)$ ,
- 2) Listen to two seconds of  $e^{-2t} \sin(2\pi \cdot 880t)$
- 3) Listen to two seconds of  $e^{-2t} \sin(2\pi \cdot 440t) + e^{-2t} \sin(2\pi \cdot 880t)$

Show the code you used to generate these three sounds.

**5c.** The note B has a frequency of 494 Hertz. Find a higher pitched note where the ratio of frequencies between your note and B is 3:2. Graph the two functions on a suitably small interval. Listen to two seconds of the sum of your note and B. Show the code you used to generate the sound.

**5d.** Play the notes with frequencies 494 Hertz and 504 Hertz simultaneously. Is the sound consonant or dissonant? Explain why. Show the code you used to generate the sound.

## 6. A Brief Hacking Interlude<sup>3</sup>

Ultimately, you can build on the ideas developed above to answer the following question: Why does a piano sound different than a flute even when they are playing the same note? Before you can answer this question, you first need to be able to take a given sound and retrieve its coordinates relative to a carefully chosen basis. To learn how to do this, imagine you are in the following situation: You cannot remember the three digit code to your answering machine, but you have a recording of a phone call where you successfully accessed it. How do you obtain the code from the sounds?

Before you can decode your recording, you first need to understand how the digits on a telephone are coded as sounds. The diagram<sup>4</sup> below explains how it works:

---

<sup>3</sup> As I am sure you will figure out, the material in this section could be put to nefarious uses. As you continue your math education, you will find how powerful math really is. Be sure to use your power for good, and not evil.

1	2	3	: 697 Hz
4	5	6	: 770 Hz
7	8	9	: 852 Hz
*	0	#	: 941 Hz
----	----	----	
1209	1336	1477	Hz

Every button is the combination of two pure tones; one corresponding to its row, and one to its column. For example, when you press 4, a tone of 1209 Hz and a tone of 770 Hz are generated simultaneously. Here is how to get MATLAB generate the same sound for one-half of a second.

```
>> t = linspace(0, .5, 4000)';
>> four = sin(2*pi*1209*t) + sin(2*pi*770*t);
>> soundsc(four, 8000)
```

In the language of linear algebra, every number's sound can be thought of as an element of the linear space with basis that has

$\left( \begin{array}{l} \sin(2\pi 697t), \sin(2\pi 770t), \sin(2\pi 852t), \sin(2\pi 941t), \\ \sin(2\pi 1209t), \sin(2\pi 1336t), \sin(2\pi 1477t) \end{array} \right)$ . To recover your

password, you need a way to take the sound and generate its coordinates relative to this basis. If you could define an inner product on the space so that the basis was orthonormal with respect to this inner product, then you could use the following fact:

Consider an orthonormal basis  $f_1, \dots, f_n$  of  $V$ . Then  $\langle g, f_1 \rangle = \langle g, f_1 \rangle + \dots + \langle g, f_n \rangle = \langle g, f_n \rangle$  for any  $g$  in  $V$ .

In other words, given the sound  $g$  in  $V$  where  $g = c_1 f_1 + c_2 f_2 + \dots + c_n f_n$  we know that  $c_1 = \langle g, f_1 \rangle, c_2 = \langle g, f_2 \rangle, \dots$ . To define the inner product, we use the continuous version of the idea of multiplying components and adding the products. Unfortunately, the obvious integral makes the basis orthogonal, but not orthonormal. To make sure each vector has length one, we need to scale the integral. So, we define

$$\langle g, f \rangle = 4 \int_0^{\frac{1}{2}} g(t) f(t) dt .$$

---

<sup>4</sup> Original diagram from <http://margo.student.utwente.nl/el/phone/dtmf.htm>

MATLAB trick: Use the `trapz` command to integrate. Here is how to calculate

$$4 \int_0^{\frac{1}{2}} \sin(2\pi \cdot 440t) \cos(2\pi \cdot 880t) dt$$

```
>> t = linspace(0,1/2,4000)';  
>> f = sin(2*pi*440*t);  
>> g = cos(2*pi*880*t);  
>> 4*trapz(t,f.*g)
```

### Problems

**6a.** Pick two distinct elements  $f_1$  and  $f_2$  from the basis above and show that  $\langle f_1, f_2 \rangle = 0$ . (Hint: Due to rounding errors, MATLAB might give you an answer that is almost, but not quite, zero. Assume any number less than  $10^{-12}$  is really 0 for this problem.)

**6b.** Pick another element  $f$  from the basis and show that  $\langle f, f \rangle = 1$

Next you are going to generate a sound, and then recover the coefficients. Then you will download a mystery tone and figure out which number it represents.

Generate the sound for 4 as above. Given the sound file, this is how you would determine that it is a 4:

```
>> 4*trapz(t, sin(2*pi*697*t) .* four)
```

```
ans =
```

```
-2.7457e-015
```

```
>> % Its not in the first row because the inner product is  
0.
```

```
>> 4*trapz(t, sin(2*pi*770*t) .* four)
```

```
ans =
```

```
1.0000
```

```
>> % It must be in the second row because the inner product  
is 1.
```

```
>> 4*trapz(t, sin(2*pi*1209*t).*four)
```

```
ans =
```

```
1.0000
```

```
>> %Bingo! It's in the first column. The number is in the  
second row, first column. The number is 4.
```

There is one subtlety that you need to understand before you can decode sounds. In problem 3f above, you saw that the same sound could also be generated with cosines or a combination of sines and cosines. In fact, instead of generating the number 4 with  $\sin(2\pi 770t) + \sin(2\pi 1209t)$ , you could have also generated it with something like  $\cos(2\pi 770t) + \cos(2\pi 1209t)$ , or even

$\frac{3}{5}\sin(2\pi 770t) + \frac{4}{5}\cos(2\pi 770t) + \frac{5}{13}\sin(2\pi 1209t) + \frac{12}{13}\cos(2\pi 1209t)$ . So, even though you can generate any sound with just sines, the sound you are trying to analyze may have cosines in it as well.

Surprisingly, if you analyzed different pieces of the same sound by using two different time intervals, you could end up with different coefficients for the sines and cosines even if the duration of the two intervals is the same.<sup>5</sup> The following theorem can be used to determine how much of a particular frequency is present in the sound:

Theorem: Let  $g(t) = f(t - k)$ , and suppose  $a = \langle f, \cos(\omega t) \rangle$ ,  $b = \langle f, \sin(\omega t) \rangle$ ,  $a' = \langle g, \cos(\omega t) \rangle$ ,  $b' = \langle g, \sin(\omega t) \rangle$  where  $\langle f, g \rangle$  is the inner product used above. Then  $\sqrt{a^2 + b^2} = \sqrt{(a')^2 + (b')^2}$

This tells us that wherever we start to analyze the signal, the quantity we want to look at for a particular frequency component is  $\sqrt{a^2 + b^2}$ .

Make sure you take this into account when solving the next two problems.

---

<sup>5</sup> This will happen if you start analyzing your sample at two different places, since the graph of cosine is just a shifted graph of sine.

**6c.** Download the sound `mystery_number` from <http://www.dersmith.com/classes/Laney/math3e/lab.html> and determine the number.

Hint: To download the sound and load it into MATLAB, do the following<sup>6</sup>:

1. Go to the website <http://www.dersmith.com/classes/Laney/math3e/lab.html>
2. Right click on the link `mystery_number`
3. Select “Save Target As...”
4. Click on My Computer
5. Click on D:
6. If there is not a directory called “temp”, click on the create new folder icon.
7. Double click on the “temp” folder.
8. Click the “Save” button.
9. To tell MATLAB where to look, type `>> addpath 'D:\temp'`
10. To load the sound, type `>> [number, Fs] = wavread('mystery_number');`
11. To listen to the sound, type `>> soundsc(number, Fs);`

The variable `number` holds the sound you want to analyze. The variable `Fs` is the sampling rate. That is, `Fs` is the number of times per second the sound was sampled. In this example, `Fs` is 8,000, which means that every  $\frac{1}{8000}$ th of a second, the sound was recorded and saved. (Hint: For technical reasons, your coefficients will not be just 0 and 1. When trying to determine whether a frequency is present or not, use the fact that a coefficient close to 0 means the frequency is absent, and a coefficient around  $\frac{1}{2}$  indicates that the frequency is present.)

**6d.** You are now ready to recover the code to your answering machine. Download the file stored at the link [answering\\_machine\\_code](http://www.dersmith.com/classes/Laney/math3e/lab.html) at <http://www.dersmith.com/classes/Laney/math3e/lab.html> and store it in a MATLAB variable called `answering_machine`. By listening to the file, you can tell that your code is three digits. What is the code to your answering machine?

Hint: Each number is held for half a second and then there is a half-second pause before the next number. The sound was sampled `Fs` times per second. So, to recover the first and third half-second of sound, you would type something like this into MATLAB:

```
>> t = linspace(0, .5, 4000)';  
>> firstNumber = answering_machine(1:.5*Fs);  
>> secondNumber = answering_machine(Fs+1:1.5*Fs);
```

---

<sup>6</sup> This assumes that you are using one of the computers in the lab at school. If you are using a Macintosh, your step 3 will be different. If you are using a computer at home, save the file to `c:\temp` and modify step 9 appropriately.

## 7. Musical Instruments

You are now ready to answer the question “Why is it that even when playing the same note, flutes and pianos sound differently?” You can answer this question with the tools you developed above.

If a piano and a flute are both playing a note with frequency of 440 Hertz (A above middle C) then both are generating sound waves with that frequency. They are also generating waves with frequencies 880 Hertz, 1320 Hertz, 1760 Hertz, etc.<sup>7</sup> Notice that these frequencies are all multiples of the 440 Hertz. In other words, they are all the same note in different octaves and sound pleasant when played together, as you saw in section 5 above.<sup>8</sup> If the two instruments are both playing a note with frequency of 494 Hertz, then they are also producing sound waves with frequencies 988 Hertz, 1482 Hertz, 1976 Hertz, etc.

The difference between the two sounds is the relative strength of the different frequencies. The piano has a loud fundamental and relatively soft harmonics, while the flute’s first harmonic is louder than its fundamental.

In the language of linear algebra, if both of these instruments are playing the note with frequency 440 Hertz, the notes they play are both elements of the linear space with basis

$$(\sin(2\pi \cdot 440t), \cos(2\pi \cdot 440t), \sin(2\pi \cdot 880t), \sin(2\pi \cdot 1320t), \cos(2\pi \cdot 880t) \dots)^9.$$

In other words, you can write both the piano’s note and the flute’s note as  $a_1 \cos(2\pi \cdot 440t) + b_1 \sin(2\pi \cdot 440t) + a_2 \sin(2\pi \cdot 880t) + b_2 \sin(2\pi \cdot 880t) + \dots$

Because you don’t know how much of the sound is in the sines and how much is in the cosines, let  $c_1 = \sqrt{a_1^2 + b_1^2}$ ,  $c_2 = \sqrt{a_2^2 + b_2^2}$ ,  $\dots$  to see how much of each frequency is present. You should expect the piano to have a large  $c_1$  compared to  $c_2, c_3, \dots$  and the flute to have  $c_2$  larger than  $c_1$ .

In the problems below, you will download a flute and a piano playing the same note. By looking at the graph of the note, you will be able to determine the fundamental frequency, and then by building the appropriate basis, you will analyze the weights of the harmonics for both instruments.

---

<sup>7</sup> Actually, they are generating sounds at many other frequencies too, but primarily at the frequencies listed above.

<sup>8</sup> In the language of music, the 440 Hertz note is called the fundamental, the 880 Hertz note is called the first harmonic, the 1320 Hertz is called the second harmonic, etc.

<sup>9</sup> For practical purposes, we truncate the basis and orthogonally project the function onto the subspace. The ideas are still the same.

## Problems

**7a.** Go to the website <http://www.dersmith.com/classes/Laney/math3e/lab.html> and download the sound file piano. Load the sound into MATLAB using

```
>> [piano, FsPiano] = wavread('piano');
```

Listen to the sound and convince yourself that you are really listening to a piano. Show the code you used to listen to the sound.

**7b.** It is best to think of the piano sound as a function of time. In this exercise you will build the independent variable time and then graph the sound.

i) Determine the duration of the sound piano.

Hint: `FsPiano` is the number of samples/second.

`length(piano)` is the number of samples.

Assign your answer to the variable `pianoDuration`

ii) 

```
>> timePiano = linspace(0, pianoDuration, length(piano))';
```

iii) Use

```
>> subplot(2,1,1); plot(timePiano,piano);
```

```
>> legend('piano'); axis([0 pianoDuration -1 1]);
```

to graph the piano sound.

iv) At the same website, download flute listen to it, and produce a flute vs. time graph below the piano vs. time graph you did in iii). Use

```
>> subplot(2,1,2);
```

 to get MATLAB to generate your graph below the piano graph. Include the two graphs in your write up.

You now need to look more deeply into the piano graph to determine the fundamental frequency. Before you start, close the window that has the graphs from the previous problem. For this part, you will look at the graph from time  $t = .6$  to  $t = .61$  seconds. To do this, you need to extract the appropriate samples from `piano` and from `timePiano`. The following table should make it clear how to do this.

$t$ (time)	$s$ (sample)
0	1
1	<code>FsPiano</code>

Since the relation is linear, we get  $s = (\text{FsPiano} - 1)t + 1$

For  $t = .6$ ,  $s = (\text{FsPiano} - 1) * .6 + 1 = 13230.4$

For  $t = .61$ ,  $s = (\text{FsPiano} - 1) * .61 + 1 = 13450.89$

Since neither of these is an integer, we need to round.

Here is how to see the graph for this time interval:

(Be sure to close any graphing window you have open before you type this.)

```
>> startingSample = round((FsPiano-1)*.6 + 1);
>> endingSample = round((FsPiano-1)*.61 + 1);
>> timeSample = timePiano(startingSample:endingSample);
>> pianoSample = piano(startingSample:endingSample);
>> plot(timeSample, pianoSample)
```

Note that the independent variable is the sample number here, not the time. Your graph will start with sample number 13230 (which corresponds to time .6 sec) and ends with sample number 13451 (which is .61 sec).

At first glance, the graph appears to be periodic, but if you look more closely, you will see that it is not quite. There is enough of a periodic structure in here that we can determine the frequency. Looking closely at the graph, it appears that the first minimum occurs somewhere between times  $t = .6$  and  $t = .603$  and the next minimum occurs somewhere between times  $t = .603$  and  $t = .605$  and both are less than  $-.175$ . We will use the MATLAB command `for` to look through the piano samples and print the values and sample numbers that are less than  $-.175$ . There are more elegant ways to find the length of the cycle, but this is the most straightforward.

MATLAB trick: To repeat something many times, use a `for` loop. For example, to display the numbers  $1^2, 2^2, \dots, 10^2$ , do the following:

```
>> for i = 1:10
disp(i^2);
end
```

```
>> for n = startingSample:endingSample
if piano(n) < -.175
disp(n); %This displays the value of n (the sample number)
disp(piano(n)); %This displays the value of piano at n
end %This closes the "if" statement above
end %This ends the for loop

13273

-0.1875

13274

-0.1953
```

```

13275
-0.1797

13323
-0.1797

13324
-0.1875

13325
-0.1797

```

There is one minimum value of  $-0.1953$  at sample number 13274 and another one with a minimum value of  $-0.1875$  at sample number 13324. Note that these samples are 50 units apart. To determine the frequency,

```

>> FsPiano/50

ans =

441

```

One of two things is happening here: Either the piano is out of tune or we did not find the true length of the cycle because the true minimums happened at places that were not sampled. Assume that the actual frequency is 440 Hz.

Let  $t$  be any time interval that lasts for  $\frac{1}{440}$ th of a second and consider the basis  $(\sin(2\pi \cdot 440t), \cos(2\pi \cdot 440t), \sin(2\pi \cdot 880t), \sin(2\pi \cdot 1320t), \cos(2\pi \cdot 880t) \dots)$  and inner product  $\langle f, g \rangle = 880 \int_a^b (f \cdot g) dt$  where  $b - a = \frac{1}{440}$ .

### Problems

**7c.** Convince yourself that the inner product makes the basis orthonormal by computing the following with  $t$  being the time interval lasting  $\frac{1}{440}$ th of a second starting at time .6 sec. Use 50 samples in your time interval.

```

>> t = linspace(.6, .6+1/440, 50)';

```

- i)  $\langle \sin(2\pi \cdot 440t), \sin(2\pi \cdot 440t) \rangle$
- ii)  $\langle \sin(2\pi \cdot 440t), \cos(2\pi \cdot 440t) \rangle$

iii)  $\langle \sin(2\pi \cdot 440t), \cos(2\pi \cdot 880t) \rangle$

Compute the above quantities again starting at another time interval that lasts  $\frac{1}{440}$ th of a second. You can pick any starting time you want.

You are now ready to start extracting the components for the piano.

```
>> startingSample = round((FsPiano-1)*.6+1);
>> endingSample = startingSample + 49;
>> timeSample = timePiano(startingSample:endingSample);
>> pianoSample = piano(startingSample:endingSample);
>> cosWeight =
880*trapz(timeSample,pianoSample.*cos(2*pi*440*timeSample));

>> sinWeight =
880*trapz(timeSample,pianoSample.*sin(2*pi*440*timeSample));

>> spec(1) = sqrt(cosWeight^2 + sinWeight^2);
```

spec(1) now has the value for the weight of the 440 Hertz component.

Use MATLAB to make the following assignments:

```
spec(2) = the 880 Hertz component
spec(3) = the 1320 Hertz component
spec(4) = the 1760 Hertz component
spec(5) = the 2200 Hertz component
spec(6) = the 2640 Hertz component
```

You now have all of the components for the piano. To see them graphed, first set up the horizontal axis: `>> for n = 1:6; freq(n) = 440*n; end`

Use

```
>> subplot(2,1,1);
>> stem(freq,spec)
>> axis([0 7*440 0 .15])
>> legend('piano')
```

to see your graph.

Fill in the bottom graph (`subplot(2,1,2);`) to see the graph of the relative weights of the frequencies for the flute sound.

Include your code and the completed graph in your write up.

## 8. Conclusion

In this lab, you have seen how to use Linear Algebra to understand sounds and music. The key idea was that by cleverly choosing a basis and defining the right inner product on it, you could easily break down the sounds in terms of their basis components.

### **Problems**

**8a.** Answer each of the questions listed in the introduction with a sentence or two.

## **9. An Annotated Bibliography**

There is much more to say about the relationship between math and music. If you want to learn more, here are some places to get started.

**Benson, Dave,** [Mathematics and Music](#)

This 500 page book can be found at <http://www.math.uga.edu/~djb/html/math-music.html>. This book covers many additional topics besides what is covered in this lab. I think the most interesting part is in the author's discussion of how math can be used to design synthesizers.

**Hall, Rachel W. and Josic, Kresimir,** [The Mathematics of Musical Instruments](#)

This article can be found in the April, 2001 "The American Mathematical Monthly". It gives a nice discussion of how different instruments are able to produce the sounds they do. A similar discussion is found in Dave Benson's book.

**Lee, Edward A. and Varaiya, Pravin,** [The Structure and Interpretation of Signals and Systems](#)

The analysis done in sections 6 and 7 above is called Fourier Analysis and can be applied to a surprising number of different phenomena. For example, Fourier himself first used this approach to understand heat flow in a metal rod. In an Electrical Engineering

curriculum, this material is taught in a Signals and Systems course. This book is an excellent introduction to these ideas and a whole lot of others. Because you have taken this Linear Algebra course, you already have the math background necessary to understand this book.

**Petersen, Mark, Musical Analysis and Synthesis in MATLAB**

This paper was published in College Mathematics Journal Vol. 35, No. 5, November 2004, p.396-401 and can also be downloaded from the first link at <http://amath.colorado.edu/pub/matlab/music/>. In addition to the ideas discussed in this project, the author also discusses the mathematics of plucked string instruments and how these ideas can be used to produce human-like sounds. The webpage itself has links to some interesting MATLAB programs and different sounds for you to analyze. This is where I got the sounds for the flute and the piano.

**Perdue, Katy, Digital Signal Processing for Music**

This webpage can be found at <http://www.cs.hmc.edu/~kperdue/MusicalDSP.html#top>. It uses the basic ideas above and goes a bit further with the analysis of the sounds. It also has a good discussion of the problems that arise in taking discrete samples of a continuous sound.

**Rusin, Dave, The Mathematics of Music**

This webpage can be found at <http://www.math.niu.edu/~rusin/papers/uses-math/music/> and contains many interesting links. The author discusses applications of different types of math to music. This is a great place to start out.

**Transnational College of Lex, Who is Fourier? A Mathematical Adventure**

This will be one of the strangest math books you will ever read. There is a large organization in Japan (The Hippo Family) whose members like to learn many languages at the same time. While learning these languages, the Hippo Family members noticed that they all had five different vowel sounds. After deciding that mathematics was a language itself, they set out to use math to understand why this was the case and wrote up their experiences. Each chapter is written by a different author and attempts to explain Fourier Analysis (what we did in sections 6 and 7) to a nonmathematical audience. This book is a lot of fun. You should check it out.