Fourier Analysis and Application in Harmonising

May 2023

Aim & Rationale

Aim

Through this investigation, the main aim is to identify why humans prefer certain harmonic relations over others through a fourier analysis and also to unpack the process behind digital harmonising. This topic includes mathematical concepts such as trigonometry, complex numbers and integration.

Rationale

As a musician, the connection between maths and music has always been an interesting topic for me. When first getting introduced to the concept of trigonometric functions, I have tried to connect it to musical frequencies. However, over time I have learnt that any sound signal can be expressed as frequencies through fourier transforms. This has answered one of my questions in music which is the working behind sound editing and autotune. Autotune is a common sound editing tool utilised by artists as it can hide any imperfections in the human voice and can also be used as a stylistic choice for a particular piece of music. Thus, I have decided to investigate further into the Fourier transforms and understand how autotune works on a basic level. To accomplish this, I have also decided to take a sample of sound and try to harmonise it according to my musical knowledge. As Fourier transform is a topic beyond the scope of higher level of Mathematics: Analysis and Approaches, I have referred to textbooks and online sources such as the book, “The Fourier Transform: A Tutorial Introduction.” [1]

[1] James V. Stone, *The Fourier Transform: A Tutorial Introduction* (Sebtel Press, 2021).

Introduction

Fourier transforms are a vital concept in mathematics pairing trigonometric functions with complex numbers and integration. They have a lot of real life applications including harmonising and sound editing. Essentially, Fourier Transforms can be used to deconstruct sound into base frequencies. This is used in a musical tool known as 'autotune' which can harmonise any recording into musical notes. Basically, by converting a piece of sound into its fundamental frequencies, the software is able to transform the individual frequencies according to harmonic relations and recombine all the sounds back together. This is known as inverse fourier transform.

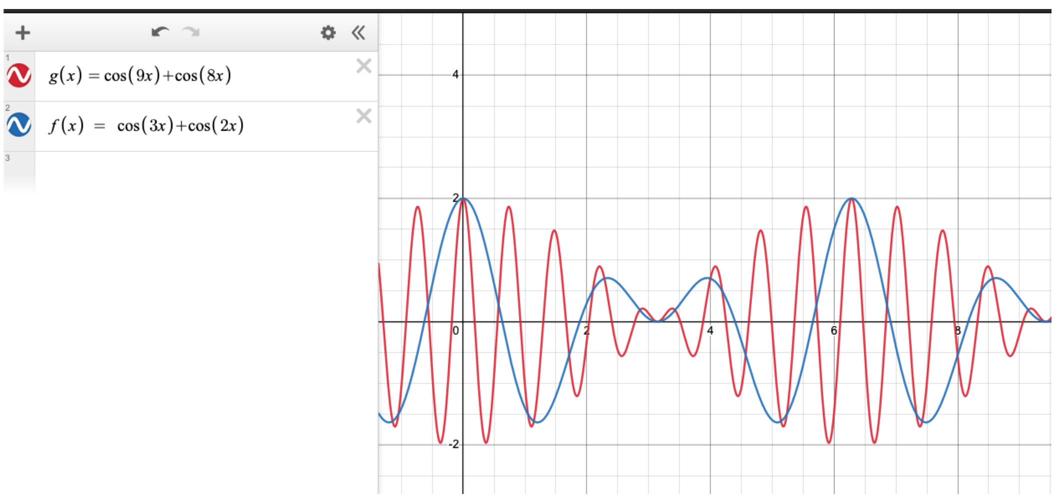
However, to successfully harmonise a recording by hand, it is first important to note what is considered as harmonic for the human ear. Harmony is a broad musical term, however for the purposes of fourier transforms, it can be boiled down to chords (multiple notes played at once). Humans consider certain chords to be more harmonic than others as the ratios between the frequencies are simple leading to periodic vibrations.[2]

[2] Elizabeth Norton, "Human Brain Is Wired for Harmony," Wired (Conde Nast, November 13, 2012), https://www.wired.com/2012/11/human-brain-harmony/.

Note Relations and their Consonance[3]

Examples of just musical intervals:	2:1	octave
	3:2	fifth
	4:3	fourth
	5:4	major third
	6:5	minor third

This table showcases the various note intervals and ratios. If the table is continued, the ratios would slowly reach 1. When the frequencies are extremely close, it sounds extremely dissonant as the sound wave is much more complex for our brains to analyse.

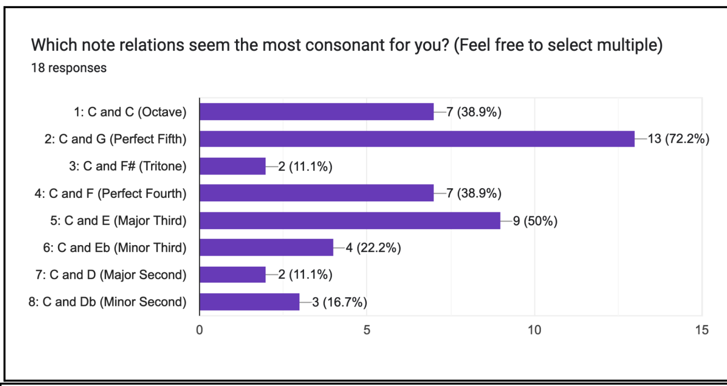


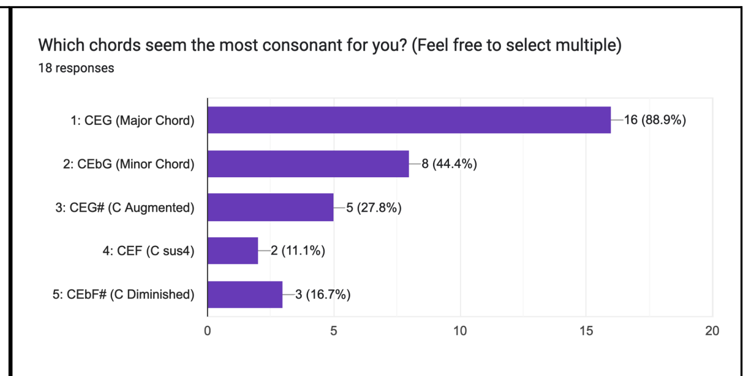
In this simple graph, it is seen how when the frequencies have ratios approaching 1 have a more complicated shape. This leads to less pleasant sounds and also harder for our ears to distinguish the sound properly.

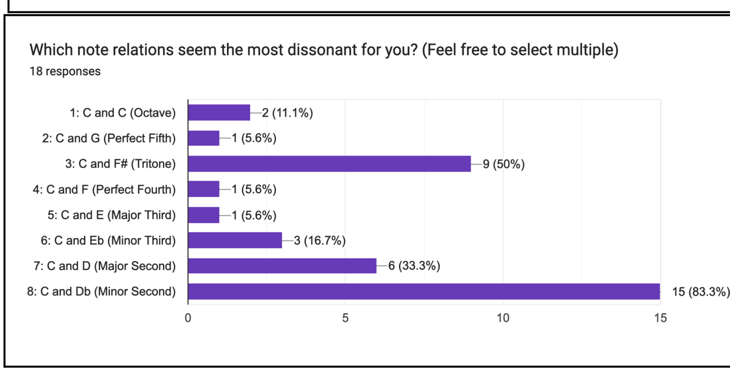
[3] Nave R, "Musical Scales," Musical Scales and Intervals, accessed October 2022, http://hyperphysics.phy-astr.gsu.edu/hbase/Music/mussca.html.

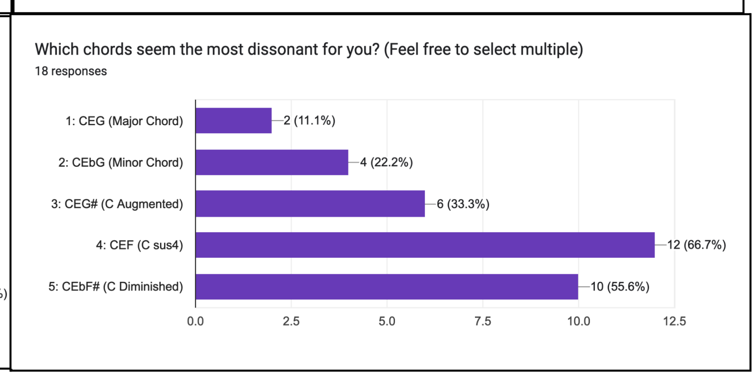
To recheck this theory of harmonic relation and how the human ear works, a primary investigation in the form of a survey was taken where the subjects were asked which note relations they find more harmonic. This survey helps rank the various note relations according to their perceived level of consonance and dissonance, allowing for further investigation in fourier analysis where computers are able to turn dissonant sounds to consonant notes.

The results of the survey:









These results show that the Perfect Fifth is considered the most consonant sound (along with the major third and perfect fourth) while the Minor Second was the most dissonant (along with the Tritone and Major Second). The subjects also found a Major Chord to be the most consonant and sus4 chord/diminished chord to be the most dissonant. Therefore, this survey supports the music theory discussed above and conclusively proves how human ears perceive harmony.

Additionally, this data can be used in the investigation as well, where a dissonant set of notes such as the diminished chord is converted to a consonant set of notes such as the major chord. This primary investigation has helped in providing clear results and increased the scope of this discussion of applications of fourier analysis in music editing and producing.

Understanding these harmonic relations are essential in sound editing as the software converts frequencies to match these simpler ratios. To understand how to deconstruct and reconstruct a specific frequency, it is important to first understand how the fourier transform works. Then to conduct the investigation, a digital recording of a musical chord will be used and using fourier analysis, it will be reconstructed to be more appealing to human ears.

Understanding Fourier Transforms[4]

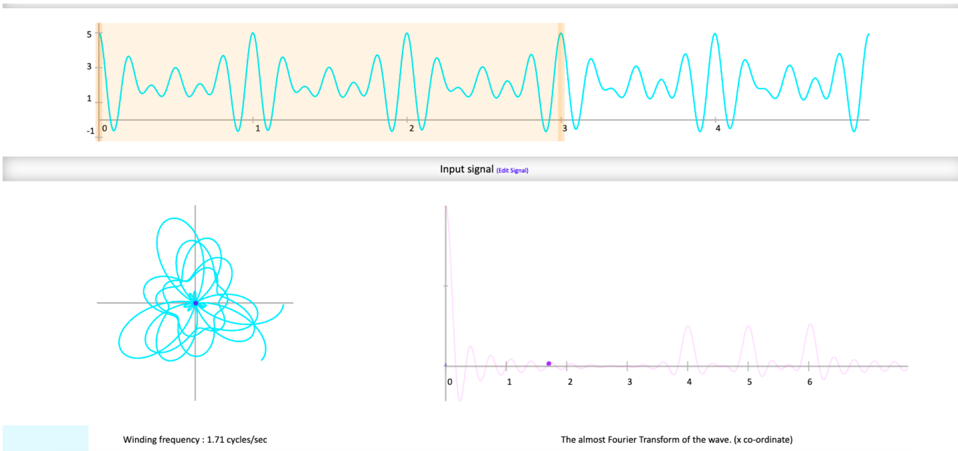
Essentially, Fourier transforms can be summarised in one single formula:

$$F(\omega) = \int_{-\infty}^{\infty} f(t) e^{-i\omega t} dt$$

This formula converts a function of amplitude and time to amplitude and frequency. However, understanding this formula is necessary to properly understand sound editing. Essentially, this formula traces the original function on a complex plane and finds the centre of mass of this new function (the average coordinates).

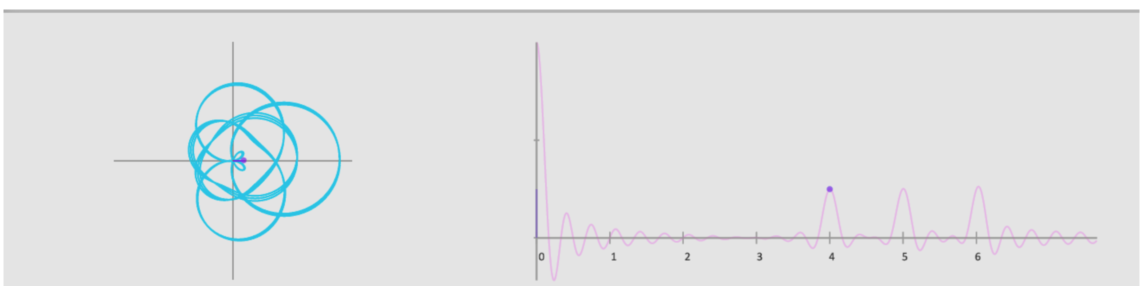
[4] Grant Sanderson, “But What Is the Fourier Transform? A Visual Introduction.,” 3 Blue 1 Brown (YouTube, January 26, 2018), https://www.youtube.com/watch?v=spUNpyF58BY.

Visual Outline of Fourier Transform



[5]

The above image is a demonstration of this process. In the formula, *f(t)* is the original signal. This signal is then multiplied by $e^{-i\omega t}$ which wraps the function in a circle as seen above. The number of times at which the function is wrapped around a circle is known as the wrapping frequency represented by ω in the function.



Using this wrapped graph, the centre of mass of the graph is found, which essentially means the average of all the coordinates. This is represented by the purple dot in the simulation. To calculate the centre of mass, the integral of $f(t)e^{-i\omega t}$ is taken.

[5] Prajwal DSouza, Fourier transform visualization, 2018, https://prajwalsouza.github.io/Experiments/Fourier-Transform-Visualization.html.

When the wrapping frequency matches one of the frequencies of the signal the centre of mass is away from the origin. This is represented by the peaks in the transform graph. The above simulation only shows the real coordinate of the graph as it's easier to calculate and visualise than the imaginary coordinate.

However, this formula cannot be used in investigating sound editing. This is because the formula only applies for continuous functions but sound signals can only be inputted as discrete data points due to limitations of recording. Therefore, for discontinuous signals, the fourier transform formula is modified to:

Discontinuous Fourier Transform

$$X(k) = \sum_{n=0}^{N-1} x(n) e^{\frac{-i2\pi kn}{N}}$$ [6]

Where,

n is the nth data point

N is the total no.of data points

j is the imaginary number ‘i’

k is the wrapping frequency

x(n) is the input signal and

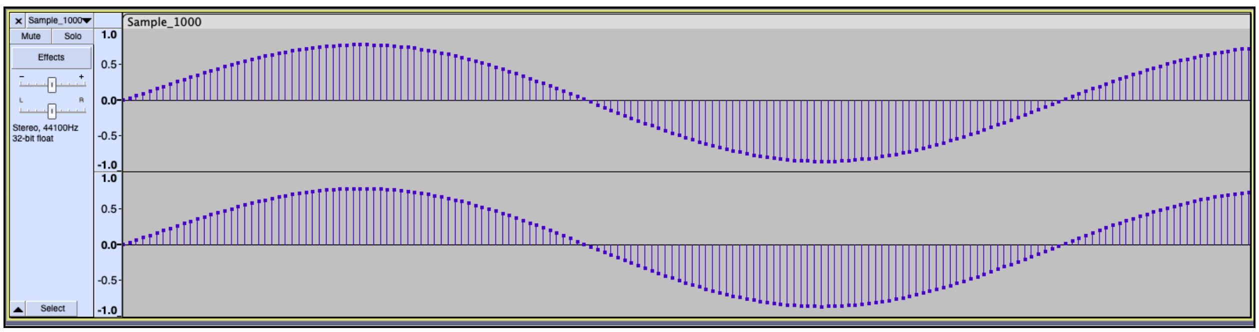
X(k) is the fourier transform output

This formula uses Sigma notation instead of Integral as it takes the sum of k values for that nth data point.

[6] Shaheen Gandhi, “An Interactive Guide to the Fourier Transform,” BetterExplained, 2016, https://betterexplained.com/articles/an-interactive-guide-to-the-fourier-transform/.

Deconstructing a Diminished Chord

A diminished chord is considered to be the most dissonant set of notes in music. Deconstructing this and harmonising it can provide good context and idea about how fourier transform can help in autotune technology. In this audio sample, the notes C, D# and F# were used, so ideally the fourier transform process should highlight these three frequencies or their overtones. Overtones are essentially related notes which simultaneously play when the root note is played. For example, if the note C is played, G is simultaneously played as its an overtone of C but its amplitude is lower.



The above wave diagram is a small extract of a sample of a diminished chord. Using the amplitude values at each sample, a table of 1059 amplitude samples at a sampling frequency of 44100 Hz were collected:

0	0.002136295663
1	0.03436384167
2	0.06689657277
3	0.09933774834
4	0.1321756645
5	0.1645863216
6	0.1972411267
7	0.2294991913
8	0.2616046632
9	0.2933439131
10	0.3247169408
11	0.3555711539
12	0.3858760338
13	0.4155705435
14	0.444532609
15	0.4727927488
16	0.5001068148
17	0.5266273995
18	0.5520187994
19	0.5765251625
20	0.5998718223
21	0.6219672231
22	0.6430249947
23	0.6626178777
24	0.6810510575
25	[illegible]

The Fourier transform process on each of these data points was conducted, where the frequency (k) is changed based on the notes of the musical scale. To simplify the process, the frequencies from one octave are taken, that is, the frequency range is between 110 Hz to 220 Hz.

$$X(k) = \sum_{n=0}^{N-1} x(n) e^{\frac{-i2\pi kn}{N}}$$

This equation can be decomposed using polar form of a complex number[7]:

$$X(k) = \sum_{n=0}^{N-1} x(n) \cos \frac{-2\pi kn}{N} + x(n) i \sin \frac{-2\pi kn}{N}$$

Example Calculation for Frequency k = 110:

$$k = 110Hz$$

$$N = 1059$$

$$X(110) = \sum_{n=0}^{1058} x(n) \cos \frac{-220\pi n}{1059} + x(n) i \sin \frac{-220\pi n}{1059}$$

Expanding the Summation:

$$X(110) = x(0) \cos \frac{-220\pi(0)}{1059} + x(0) i \sin \frac{-220\pi(0)}{1059} +$$

$$x(1) \cos \frac{-220\pi(1)}{1059} + x(1) i \sin \frac{-220\pi(1)}{1059} + x(2) \cos \frac{-220\pi(2)}{1059} + x(2) i \sin \frac{-220\pi(2)}{1059} +$$

$$+ x(1058) \cos \frac{-220\pi(1058)}{1059} + x(1058) i \sin \frac{-220\pi(1058)}{1059}$$

$$... + x(1057) \cos \frac{-220\pi(1057)}{1059} + x(1057) i \sin \frac{-220\pi(1057)}{1059}$$

[7] Simon Xu, "Discrete Fourier Transform - Simple Step by Step," Simon Xu (YouTube, August 3, 2015), https://www.youtube.com/watch?v=mkGsMWi_j4Q.

Substituting the values of x(n):

$$X(110) = 0.002136\cos(0) + 0.002136i\sin(0) + 0.034364\cos(\frac{-220\pi}{1059}) +$$
$$0.034364i\sin(\frac{-220\pi}{1059}) + ... + 0.020295\cos(\frac{-232540\pi}{1059}) + 0.020295i\sin(\frac{-232540\pi}{1059})$$
$$+0.004852\cos(\frac{-232760\pi}{1059}) + 0.004852i\sin(\frac{-232760\pi}{1059})$$
$$X(110) = 0.0021 + 0.027 - 0.021i + ... + 0.0053 + 0.020i + 0.0039 + 0.0029i$$
$$X(110) = -0.1088 - 0.0184i$$

Magnitude/Amplitude of the Frequency:

$$Magnitude = \sqrt{Real^2 + Img^2}$$
$$Magnitude = \sqrt{(-0.1088)^2 + (-0.0184)^2}$$
$$Magnitude = 0.1104$$

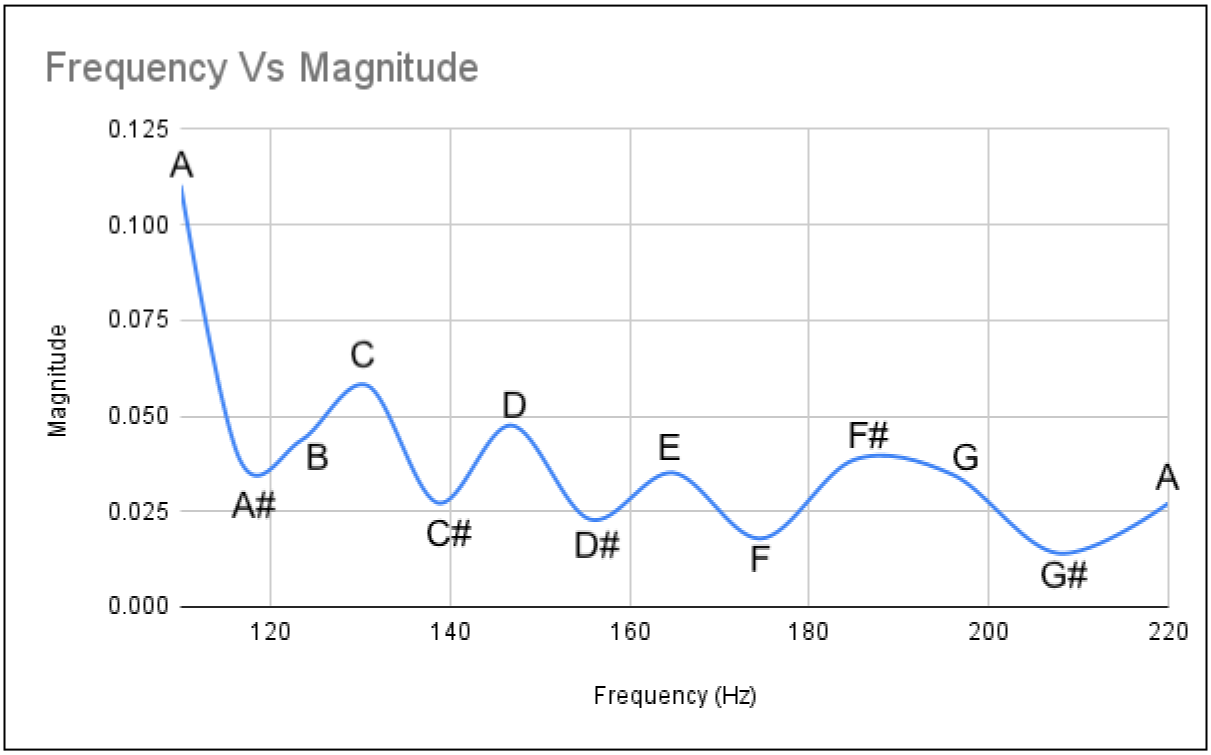
Repeating the same process with all other values of k (the frequency) provides a list of amplitude values for each note of the musical scale. This gives an idea about the base frequencies composing the original recording and allows for manipulation of the audio to improve its harmony (according to human perception).

Frequency Vs Magnitude

Frequency (Hz)	Equivalent Note	Magnitude
110	A	0.11038
116.5409	A#	0.03885
123.4708	B	0.04370
130.8128	C	0.05802
138.5913	C#	0.02728
146.8324	D	0.04757
155.5635	D#	0.02294

164.8138	E	0.03514
174.6141	F	0.01789
184.9972	F#	0.03854
195.9977	G	0.03463
207.6523	G#	0.01407
220	A	0.02714

Graphical Representation



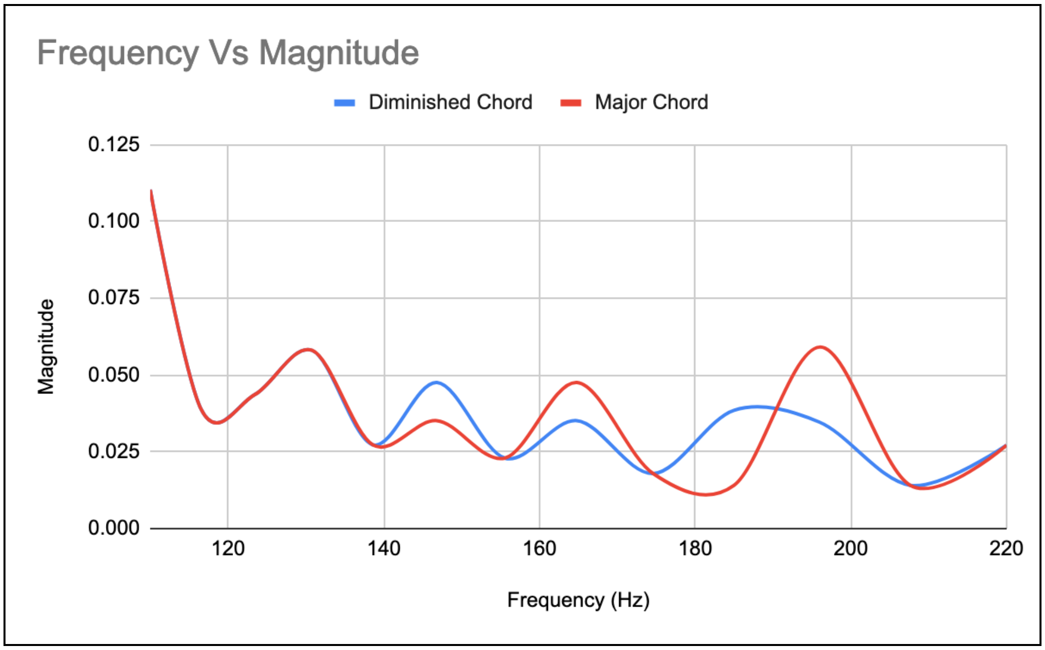
Though the original sound recording was composed of only C, D# and F#, it is clear that many other frequencies were also part of the recording in the form of overtones. As explained previously, each note has certain overtone notes which simultaneously play along. Here, the presence of the E and G peak can be explained as the overtones of C while D can be explained as the overtone of F#. The note A also has a high peak as its the start of the octave, thus being the base overtone of all the notes.

Reconstructing to a Major Chord

To reconstruct this audio into a major chord, the magnitude of the frequencies must be transposed according to the new combination of frequencies required. A C major chord consists of C, E and G, thus these must be the main peaks visible in the graph. The overtones of each of these notes include only D and B. Therefore, the main transposition required is between F# and G and D and E. This provides us with a new table and a new graph.

Frequency Vs Magnitude for Major Chord

Frequency (Hz)	Equivalent Note	Magnitude
110	A	0.11038
116.5409	A#	0.03885
123.4708	B	0.04370
130.8128	C	0.05802
138.5913	C#	0.02728
146.8324	D	0.03514
155.5635	D#	0.02294
164.8138	E	0.04757
174.6141	F	0.01789
184.9972	F#	0.01407
195.9977	G	0.05910
207.6523	G#	0.01407
220	A	0.02714

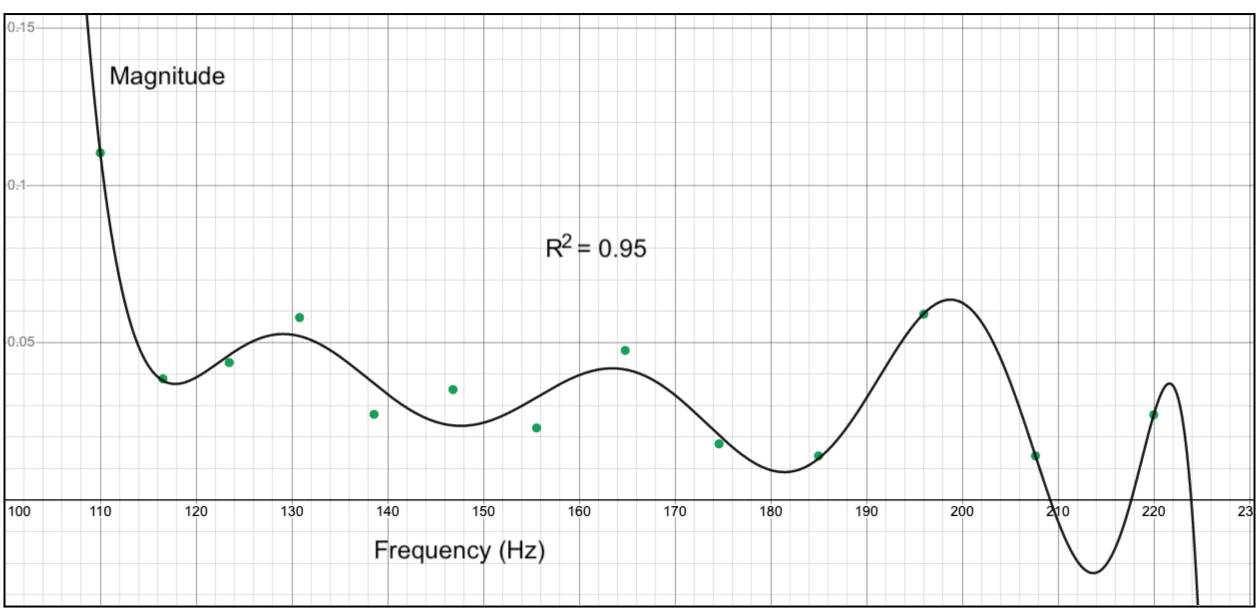


To find the equation of this graph polynomial regression can be used. As this is labour intensive to do manually, a software was used for this purpose:

$$F(\omega) = -2.19 \cdot 10^{-19}\omega^{11} + 3.83 \cdot 10^{-16}\omega^{10} - 3.02 \cdot 10^{-13}\omega^{9} + 1.42 \cdot 10^{-10}\omega^{8} - 4.42 \cdot 10^{-8}\omega^{7} + 9.56 \cdot 10^{-6}\omega^{6} - 1.47 \cdot 10^{-3}\omega^{5} + 0.16\omega^{4} - 12.12\omega^{3} + 609\omega^{2} - 18300\omega + 248000$$

Where ω is the frequency

$F(\omega)$ represented as a graph:



Though the function does not match the data one to one, it is a helpful representation of the data and covers the major peaks required (C, E and G frequencies). Thus this function can be used to reconstruct the diminished chord to a major chord using continuous reverse fourier transform.

Forward Fourier transform:

$$F(\omega) = \int_{-\infty}^{\infty} f(t)e^{-2\pi i\omega t}dt$$

Reverse Fourier transform:

$$f(t) = \int_{-\infty}^{\infty} F(\omega)e^{2\pi i\omega t}d\omega$$ [8]

Essentially, the reverse fourier transform is the result of switching the two main functions. To derive this formula, the application of Fourier series is required as well, which is beyond the scope of this investigation.

Reconstruction

To integrate the current function, the complex number must be divided into real and imaginary parts. This will ease the process of integration as the complex number 'i' can be excluded from the integration:

$$f(t) = \int_{-\infty}^{\infty} F(\omega)(cos(2\pi\omega t) + isin(2\pi\omega t))d\omega$$

$$f(t) = \int_{-\infty}^{\infty} F(\omega)cos(2\pi\omega t)d\omega + i\int_{-\infty}^{\infty} F(\omega)sin(2\pi\omega t)d\omega$$

[8] Erik Cheever, "Introduction to the Fourier Transform," Linear Physical Systems (Swarthmore College, 2005), https://lpsa.swarthmore.edu/Fourier/Xforms/FXformIntro.html#de.

The above integration can be solved using the method ‘integration by parts’. However, as $F(\omega)$ is a polynomial of degree 11, this method must be done 11 times to simplify. Instead of doing this manually, a pattern can be used which is observed after the first two times:

$$
\begin{array}{lll}
u = F(\omega) & v' = cos(2\pi\omega t) & w' = sin(2\pi\omega t) \\
u' = F'(\omega) & v = \frac{1}{2\pi t} sin(2\pi\omega t) & w = -\frac{1}{2\pi t} cos(2\pi\omega t)
\end{array}
$$

$$\int uv'dx = uv - \int u'vdx$$

$$f(t) = \left[\tfrac{1}{2\pi t} F(\omega) sin(2\pi\omega t)\right]_{-\infty}^{\infty} - \tfrac{1}{2\pi t}\int_{-\infty}^{\infty} F'(\omega) sin(2\pi\omega t) d\omega) +$$
$$+ i\left[\tfrac{-1}{2\pi t} F(\omega) cos(2\pi\omega t)\right]_{-\infty}^{\infty} + \tfrac{1}{2\pi t}\int_{-\infty}^{\infty} F'(\omega) cos(2\pi\omega t) d\omega)$$

$$f(t) = \tfrac{1}{2\pi t}([F(\omega) sin(2\pi\omega t)]_{-\infty}^{\infty} - \int_{-\infty}^{\infty} F'(\omega) sin(2\pi\omega t) d\omega$$
$$+ i\,[-F(\omega) cos(2\pi\omega t)]_{-\infty}^{\infty} + \int_{-\infty}^{\infty} F'(\omega) cos(2\pi\omega t) d\omega)$$

To simplify the calculation, some parts of the question can be contracted to make the integration clearer:

$$A = [F(\omega) sin(2\pi\omega t)]_{-\infty}^{\infty}$$

$$B = [-F(\omega) cos(2\pi\omega t)]_{-\infty}^{\infty}$$

$$A' = [F'(\omega) sin(2\pi\omega t)]_{-\infty}^{\infty}$$

$$B' = [-F'(\omega) cos(2\pi\omega t)]_{-\infty}^{\infty}$$

$$f(t) = \frac{1}{2\pi t}(A - \int_{-\infty}^{\infty} F'(\omega) sin(2\pi\omega t) d\omega +$$
$$i(B + \int_{-\infty}^{\infty} F'(\omega) cos(2\pi\omega t) d\omega))$$

Integrating by parts:

$$u = F'(\omega) \qquad v' = sin(2\pi\omega t) \qquad w' = cos(2\pi\omega t)$$
$$u' = F''(\omega) \qquad v = -\frac{1}{2\pi t} cos(2\pi\omega t) \qquad w = \frac{1}{2\pi t} sin(2\pi\omega t)$$

$$f(t) = \frac{1}{2\pi t}(A - ([-\frac{1}{2\pi t}F'(\omega) cos(2\pi\omega t)]_{-\infty}^{\infty} + \frac{1}{2\pi t}\int_{-\infty}^{\infty} F''(\omega) cos(2\pi\omega t) d\omega) +$$
$$i(B + ([\frac{1}{2\pi t}F'(\omega) sin(2\pi\omega t)]_{-\infty}^{\infty} - \frac{1}{2\pi t}\int_{-\infty}^{\infty} F''(\omega) sin(2\pi\omega t) d\omega)))$$

$$f(t) = \frac{1}{2\pi t}(A - \frac{1}{2\pi t}(B' + \int_{-\infty}^{\infty} F''(\omega) cos(2\pi\omega t) d\omega) +$$
$$i(B + \frac{1}{2\pi t}(A' - \int_{-\infty}^{\infty} F''(\omega) sin(2\pi\omega t) d\omega)))$$

From this, a pattern can be observed. The integration alternates between sine and cosine while the degree of $F(\omega)$ is decreasing with each step. The term which is out of the integration too follows the same pattern as observed in the alternating A and B. This pattern can be continued further until $F^{(n)}(\omega)$ becomes a constant:

$$f(t) = \frac{1}{2\pi t}(A - \frac{1}{2\pi t}(B' + \frac{1}{2\pi t}(A'' - \frac{1}{2\pi t}(B''' + \frac{1}{2\pi t}(A^{(4)} -$$
$$\frac{1}{2\pi t}(B^{(5)} + \frac{1}{2\pi t}(A^{(6)} - \frac{1}{2\pi t}(B^{(7)} + \frac{1}{2\pi t}(A^{(8)} - \frac{1}{2\pi t}(B^{(9)} +$$
$$\frac{1}{2\pi t}(A^{(10)} - \int_{-\infty}^{\infty} F^{(11)} sin(2\pi\omega t) d\omega)))))))))) +$$

$$i(A - \frac{1}{2\pi t}(B' + \frac{1}{2\pi t}(A'' - \frac{1}{2\pi t}(B''' + \frac{1}{2\pi t}(A^{(4)} - \frac{1}{2\pi t}(B^{(5)} +$$
$$\frac{1}{2\pi t}(A^{(6)} - \frac{1}{2\pi t}(B^{(7)} + \frac{1}{2\pi t}(A^{(8)} - \frac{1}{2\pi t}(B^{(9)} + \frac{1}{2\pi t}(A^{(10)} -$$
$$\int_{-\infty}^{\infty} F^{(11)} sin(2\pi\omega t) d\omega)))))))))))$$

Where,

$$A^{(n)} = \left[F^{(n)}(\omega)sin(2\pi\omega t)\right]_{-\infty}^{\infty}$$

$$B^{(n)} = \left[-F^{(n)}(\omega)cos(2\pi\omega t)\right]_{-\infty}^{\infty}$$

To simplify this equation, the limits must be changed according to the equation derived above. As the data only has a range of 110 Hz to 220 Hz, these must be the limits due to the errors which arise from extrapolation. This way the expression can be simplified in terms of t as the independent variable. When simplified, the equation would be in the form of:

$$f(t) = a_1 sin(220\pi t) + b_1 sin(440\pi t) + c_1 cos(220\pi t) + d_1 cos(440\pi t) +$$
$$i(a_2 sin(220\pi t) + b_2 sin(440\pi t) + c_2 cos(220\pi t) + d_2 cos(440\pi t))$$

The coefficients could be found by simplifying the expression, however, this is not possible due to the number of terms present in $F(\omega)$ making it almost impossible to process manually. Yet, this equation clearly shows how a computer can recombine the frequencies to reconstruct a diminished chord into a major chord. The magnitude of this equation provides the amplitude at each t value:

$$Real = a_1 sin(220\pi t) + b_1 sin(440\pi t) + c_1 cos(220\pi t) + d_1 cos(440\pi t)$$

$$Img = a_2 sin(220\pi t) + b_2 sin(440\pi t) + c_2 cos(220\pi t) + d_2 cos(440\pi t)$$

$$Magnitude = \sqrt{Real^2 + Img^2}$$

Essentially, this is how an autotune software functions. By deconstructing a sound sample into its frequencies using fourier transform, it can translate these points and reconstruct them once again using reverse fourier transform. A caveat in this investigation was that a limited range was taken for the fourier transform. In actuality, the software takes the entire range of music frequencies to ensure accurate reconstruction of the sound.

12 Tone Equal Temperament and Pythagorean Tuning

In this investigation, the frequencies used for the notes were all separated by a common ratio which is $\sqrt[12]{2}$. However, this is one way of tuning known as the 12 tone equal temperament[9]. 12 Tone Equal temperament does not match the previously shown frequencies as $\sqrt[12]{2}$ is an irrational number. This leads to irregular patterns between the note frequencies. Sometimes to avoid this effect, many musicians try to tune their vocals to simple ratios to increase the harmony in their music.

Example of 12 tone Temperament against Pythagorean Tuning:

Perfect Fifth:

$$Pythagorean = 3/2$$

$$Pythagorean = 1.5$$

$$12tone = (\sqrt[12]{2})^7$$

$$12tone = 1.498$$

[9] Henry Reich, "Why It's Impossible to Tune a Piano," Minute Physics (YouTube, September 17, 2015), https://www.youtube.com/watch?v=1Hqm0dYKUx4.

The only note relation where these two tunings agree is for an octave. This small difference is almost imperceivable for humans but some musicians insist on using the simple ratios over the 12 tone tuning. This requires changes in the fourier transform process. Instead of translating the amplitudes of frequencies to traditionally used tuning, the softwares must instead tune them to pythagorean notes. This is a relatively small change but changes the function for the reconstruction.

Conclusion

Fourier Transform is a function with wide applications in physics, music and seismology. In this exploration, its procedure and application in music was investigated and the mathematics behind autotune was outlined successfully. By understanding human harmony and basic music theory, a diminished chord, or what was found out to be one of the most dissonant sounds, has been turned to a major chord, or what was found out to be the most consonant set of sounds. However, even after successfully exploring the process behind autotune, there are still few caveats in the procedure of the investigation. Firstly, only 1059 samples were taken from the recording which is less than 0.1 seconds. This leads to less accurate data, as the Fourier Transform works best with large sets of data as indicated by the formula (negative infinity to infinity). Secondly, the chord was reconstructed using a best fit curve for the data points, therefore, there is an inaccuracy in the reconstruction. These errors and uncertainties make the exploration less reliable, however, the process is still applicable in a real life scenario.

Resources

Stone, James V. *The Fourier Transform: A Tutorial Introduction*. Sebtel Press, 2021.

Norton, Elizabeth. “Human Brain Is Wired for Harmony.” Wired. Conde Nast, November 13, 2012. https://www.wired.com/2012/11/human-brain-harmony/.

R, Nave. “Musical Scales.” Musical Scales and Intervals. Accessed October 2022. http://hyperphysics.phy-astr.gsu.edu/hbase/Music/mussca.html.

Sanderson, Grant. “But What Is the Fourier Transform? A Visual Introduction.” 3 Blue 1 Brown. YouTube, January 26, 2018. https://www.youtube.com/watch?v=spUNpyF58BY.

DSouza, Prajwal. Fourier transform visualization, 2018. https://prajwalsouza.github.io/Experiments/Fourier-Transform-Visualization.html.

Gandhi, Shaheen. “An Interactive Guide to the Fourier Transform.” BetterExplained, 2016. https://betterexplained.com/articles/an-interactive-guide-to-the-fourier-transform/.

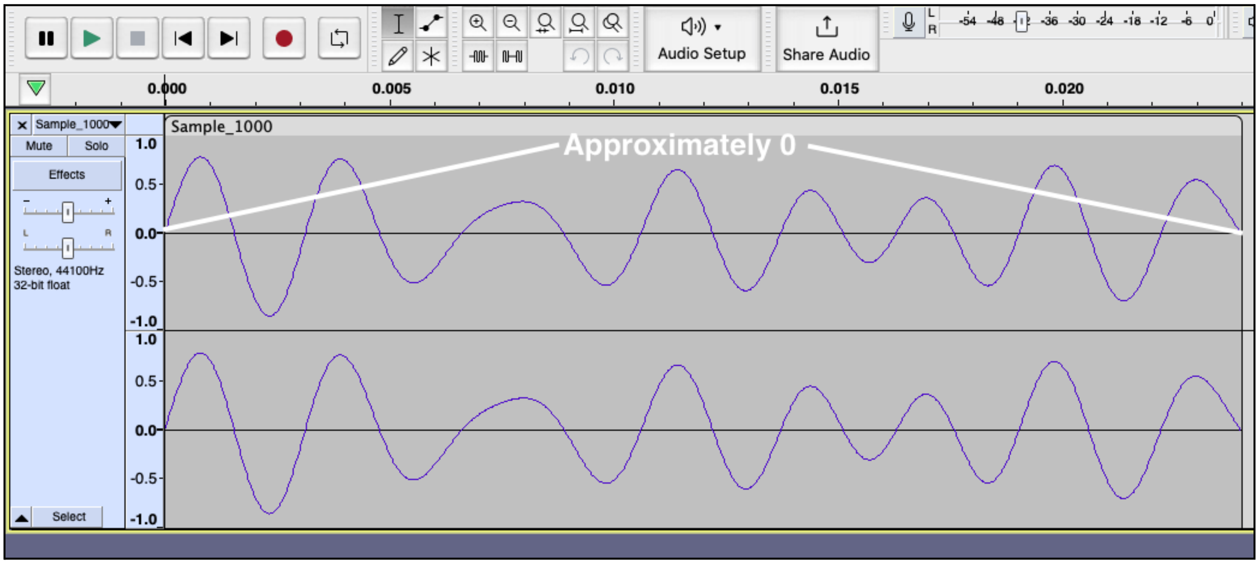
Xu, Simon. “Discrete Fourier Transform - Simple Step by Step.” Simon Xu. YouTube, August 3, 2015. https://www.youtube.com/watch?v=mkGsMWi_j4Q.

Cheever, Erik. “Introduction to the Fourier Transform.” Linear Physical Systems. Swarthmore College, 2005. https://lpsa.swarthmore.edu/Fourier/Xforms/FXformIntro.html#de.

Reich, Henry. “Why It's Impossible to Tune a Piano.” Minute Physics. YouTube, September 17, 2015. https://www.youtube.com/watch?v=1Hqm0dYKUx4.

Appendix

Creating Audio Sample:



Extracting the Amplitude Values:

```
from scipy.io import wavfile
import numpy as np

samplerate, data = wavfile.read('Sample_1000.wav')

print(f"samples = {data.shape[0]}")

for i in data:
  a = i[0]*(1/32767)
  print(a)
```

```
samples = 1059
0.0021362956633198035
0.034363841669972 84
0.06689657277138585
0.09933774834437085
0.13217566454054383
0.16458632160405287
0.19724112674336985
0.22949919125949889
0.26160466322824794
0.293343913083285
0.3247169408246101
0.3555711539048433
0.3858760338145085
0.4155705435346538
0.4445326090273751
0.4727927488021485
0.500106814783166
0.5266273995178076
0.5520187994018372
0.576525162511063
0.5998718222602009
0.6219672231208228
```

Data Analysis of the Sample Audio:

https://docs.google.com/spreadsheets/d/17Toxy3bivrPdtziFdT9_eJZX0j1E7V18mcrY6ToScVo/edit?usp=sharing

		k = 110		k = 116.5409		k = 123.4708	
Sample Number	Amplitude	Real Part	Imaginary Part	Real Part	Imaginary Part	Real Part	Imaginary Part
0	0.002136295663	0.002136295663	0	0.002136295663	0	0.002136295663	0
1	0.03436384167	0.02730140839	-0.02086879763	0.02647117884	-0.02191233226	0.02554811322	-0.02298189559
2	0.06689657277	0.01755360115	-0.06455247892	0.01249547611	-0.06571920971	0.007055118371	-0.06652350527
3	0.09933774834	-0.03750383787	-0.09198614238	-0.04793517224	-0.08700693943	-0.05827591269	-0.08044815875
4	0.1321756645	-0.113974205	-0.06693494514	-0.1229525123	-0.04850861797	-0.1292354273	-0.02772382768
5	0.1645863216	-0.1633704049	0.01996917759	-0.1564539176	0.05109627114	-0.1427288914	0.08195804298
6	0.1972411267	-0.1410134328	0.1379103834	-0.1053851475	0.1667274205	-0.06147950572	0.1874148672
7	0.2294991913	-0.03290617379	0.227127855	0.02924514094	0.2276282068	0.09265559324	0.2099638536
8	0.2616046632	0.127427596	0.2284714591	0.1911336891	0.1786194634	0.238586066	0.1073018589
9	0.2933439131	0.2691035537	0.1167644156	0.2928140473	0.01762342382	0.2793675077	-0.08946757491
10	0.3247169408	0.3151567222	-0.07821337543	0.2621239246	-0.1916563064	0.1636781046	-0.280447089
11	0.3555711539	0.2221659241	-0.2776205102	0.0872823806	-0.3446920822	-0.07212865618	-0.3481785497
12	0.3858760338	0.00858474007	-0.3857805279	-0.1655622509	-0.348553374	-0.3108963531	-0.2285689635
13	0.4155705435	-0.2449638686	-0.3356956653	-0.3767112337	-0.1754637372	-0.4135516493	0.04091344522
14	0.444532609	-0.4262547184	-0.1261592469	-0.4300955498	0.1123701853	-0.2996171946	0.3283881502
15	0.4727927488	-0.4416659184	0.1687133659	-0.2761650676	0.3837523143	-0.00333248006	0.4727810042
16	0.5001068148	-0.2627898499	0.4254977332	0.03381330927	0.4989624097	0.3318330943	0.374157218
17	0.5266273995	0.052250092	0.5240289551	0.362467395	0.3820390104	0.5232860332	0.05922959906
18	0.5520187994	0.3770942669	0.4031434841	0.548033965	0.06620821739	0.4493210247	-0.3206795466
19	0.5765251625	0.5685878569	0.09533683429	0.484995743	-0.311705618	0.1248963133	-0.5628340554
20	0.5998718223	0.5302667927	-0.280469841	0.1819213764	-0.5716212172	-0.2950403956	-0.5223000748
21	0.6219672231	0.2602041126	-0.5649221596	-0.2326242508	-0.5768268237	-0.5896006	-0.1980261576
22	0.6430249947	-0.140960277	-0.6273845265	-0.5655327977	-0.3060290812	-0.5901049149	0.255455149
[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]