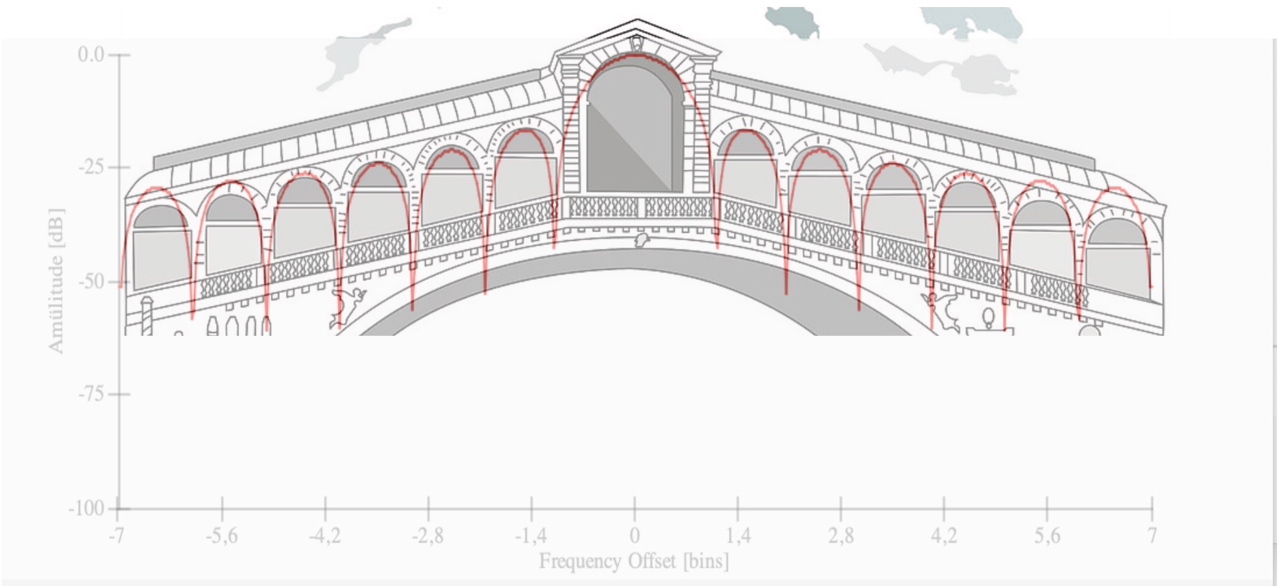
Fourier Round Trip Program

Alan Shepherd

Like a circle in a spiral, like a wheel within a wheel
Never ending or beginning on an ever spinning reel
As the images unwind, like the circles that you find
In the windmills of your mind!

From the lyrics of the song "The Windmills of your Mind"
by Alan Bergman, Marilyn Bergman



Frontispiece 1 Hann Window Superimposed on Rialto Bridge, Venice, Italy

Table of Contents

1 Introduction

The Fourier Round Trip program is intended for exploring wave forms produced by harmonics or overtones, for example the sound waves produced by musical instruments, or any other type of wave.

It has two main functions with the following features:

- Synthesis to build up a wave from its harmonics:
  - Visualisation of the component waves, frequency spectrum, resultant wave
  - Visualisation of the constructor and generator figures.
  - Traverse the angles showing the vectors of the reference figure and the projection of the constructor figure to the wave.
  - Preset the component waves for sawtooth, square or triangular waveforms. Sawtooth and square waves can have linear or exponential amplitude progressions.
  - Add random noise.
  - Edit the frequency, amplitude and phase of each harmonic.
  - Export the harmonics to a CSV-file and import them.
  - Play the resultant wave at any frequency and any duration.
  - Save the resultant wave to a WAV sound file.
  - Send the resultant wave directly to the Analysis function.
- Analysis to perform Fourier analysis of a sound file:
  - Read a WAV sound file and display the waveform of left or right channel.
  - Display the meta-data from the file.
  - Zoom and pan the wave plot over time. (x-axis)
  - Zoom the wave plot amplitude (y-axis).
  - Select a window (or frame) for analysis.
  - Apply a variety of window-functions to the frame.
  - Show the window-function, its transform and main parameters.
  - Pad the frame with zeros to increase frequency resolution.
  - Play any section of the sound as a loop.
  - Perform a Discrete Fourier Transform (DFT).
  - Perform a Fast Fourier Transform (FFT).
  - Perform a Chirp Z-Transform (CZT) of part of the frequency spectrum.
  - Perform a Short Time Fourier Transform STFT over multiple frames of DFT, FFT or CZT.
  - Average the transform of multiple frames of DFT, FFT or CZT.
  - Apply a DC filter before transform.
  - Visualise analysis results as frequency and phase spectra.
  - Visualise the STFT result in an animated spectrum.
  - Zoom and pan the frequency axis of the resulting spectrum.
  - Use linear or logarithmic (dB) scales for amplitudes.
  - Export the analysis results as a CSV-file.
  - Send analysis results to the Synthesis function.
- Analysis of Decay Rate of a sound file
  - Averaging of the wave with variable span.
  - Linear interpolation of decay.
  - Linear or Logarithmic (decibel) scales.

In combination, the synthesis and Fourier analysis enable a "round trip" of synthesising a wave, sending it to the analysis, sending the harmonics back to the synthesis.

What it is not
We follow the philosophy that a program should perform one or more specific functions and not try to perform other, possibly related, functions that other programs already do well. Instead, it should be open to exchanging data with these other programs. Therefore, this program:

- Is not a synthesiser with attack, decay, filters etc., but the generated waves could probably be used as samples for such programs.
- Does not have a MIDI interface to play music, but again, the results could be used as samples for such programs.
- Does not edit sound files but reads and writes WAV-files which can be edited and converted with other programs.
- Does not perform real-time Fourier analysis from a microphone, there are other programs for that, and we are interested in a single cycle or a few cycles that we wish to select from a recorded or synthesised wave. Recordings made with other programs and saved as a WAV file can, of course, be used.

In keeping with this philosophy, the synthesised harmonics, and the analysed harmonics can be exported to CSV files for further analysis, e.g. in Excel, and the synthesised wave can be output to a WAV file.

Some important terms are defined in chapter 2. Chapter 3 gives the theoretical background of the program and chapter 4 describes some possible uses. The installation of the program is described in chapter 5. An overview of the main window and menus is given in chapter 6. Chapter 7 describes the synthesis function and chapter 8 the analysis. Chapter 9 introduces some experiments to demonstrate the purpose of some of the functions. The appendices give further mathematical background and a list of the windowing functions provided.

2 Definitions

Frame	1) In a sound file, the bytes that make up a sample. 2) In analysis, a section of the wave being analysed, equivalent to a Window (see below). The term “frame” is used when the window is shifted in time for repeated analysis.
Fundamental	The natural frequency with which a string or pipe etc. vibrates.
Harmonic	A multiple N of the fundamental frequency. For N=1 the first harmonic is the fundamental.
Overtone	A multiple M of the fundamental frequency. For M=1 the first overtone is twice the fundamental frequency and is the second harmonic.
Spectrum	The component sine (or cosine) waves that make up a wave, each given by frequency, amplitude and phase.
Reference Figure	The figure traced out by the end of the vector resulting from the addition of the vectors of the component waves.
Resultant	The wave resulting from the addition of its component harmonics.
Constructor Figure	The figure that can be used to construct a waveform by laying off the angles and plotting the vertical displacement.
Generator	An electronic device or computer program that produces a certain waveform, e.g., sine wave, square wave. Also known as Function Generator.
Window	A short section of sound, e.g., selected from a music file and possibly modified with a window-function. (The word is also used for a part of a computer display.)
Window -function	A function used to smooth the ends of a window to avoid discontinuities when it is putatively extended in time.

3 Theoretical Background

3.1 Introduction

Musical notes are sound waves of certain frequencies caused by vibrating strings or tubes of air. The timbres of musical instruments are caused by the presence of harmonics or overtones in addition to the basic frequency. Apart from acoustics, there are many other areas where waves occur.
The simplest waveform is the sine wave, and all waveforms can be decomposed into a set of sinewaves at different frequencies and amplitudes by Fourier analysis. The result is a frequency spectrum.
A sinewave can be constructed from a circle: the x-axis is the angle going anti-clockwise from 0° to 360° or from 0 to 2π radians. The y-axis is the sine of the angle. The circle is traversed once for each cycle of the wave. A rotation rate of 2π radians per second equals a frequency of the wave of 1Hz (Hertz or cycles per second). The circle is a representation of a sine wave in polar coordinates, with the angle θ and the length *r*.

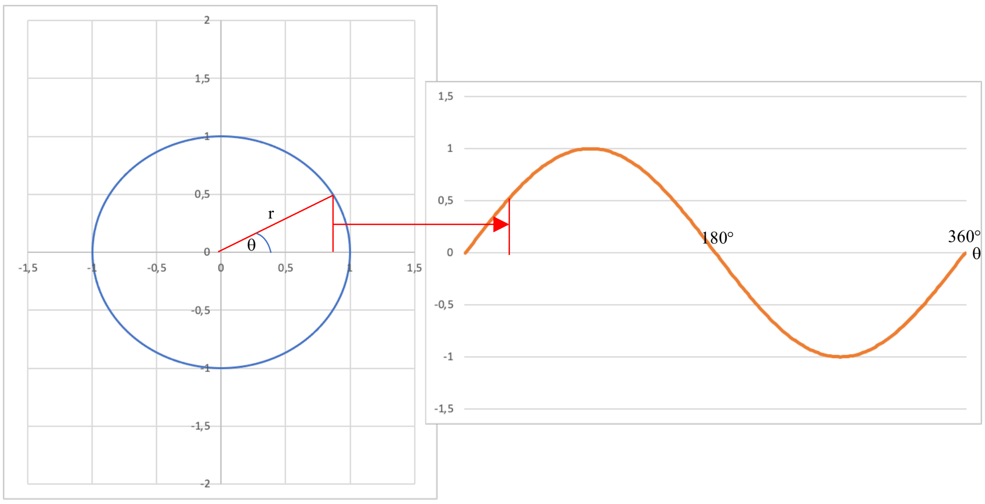


Figure 3-1Circle and Derived Sine Wave

3.2 Occurrences of Waves

The sine wave also results from harmonic oscillation such as a mass on the end of a spiral spring, a pendulum, or some electronic circuits. They occur in large bodies of water such as lakes and oceans. Electromagnetic waves range through radio waves, microwaves and heat, light, ultra-violet, x-ray and gamma-ray radiation. They occur as gravitational waves in cosmology and seismic waves resulting from earthquakes.
Wave functions are used in physics to mathematically describe the behaviour of sub-atomic particles and quantum effects.

3.3 Harmonics

A string, for example on a guitar, is fixed at both ends and so can only vibrate with its full amplitude in the middle. The modes of vibration are therefore restricted as shown in Figure 3-2. The string can vibrate at 1, 2, 3, … times the fundamental frequency (we see ½, 1, 1½, … wavelengths).

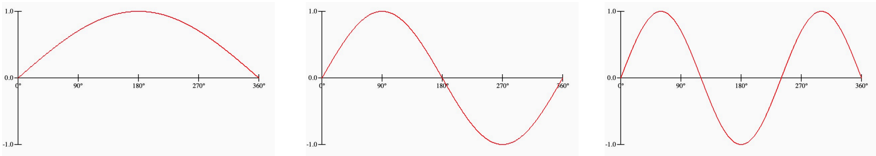


Figure 3-2 First Three Modes of Vibration of a String

The air in a pipe, as in a church organ, closed at one end and open at the other, can only vibrate as shown in Figure 3-3, where the pipe is horizontal with the open end at the right. The y-axis represents the amplitude of the compressions of the air at the point along the x axis. This can vibrate at 1, 3, 5, … times the fundamental frequency (we see ¼, ¾, 1¼, … wavelengths).

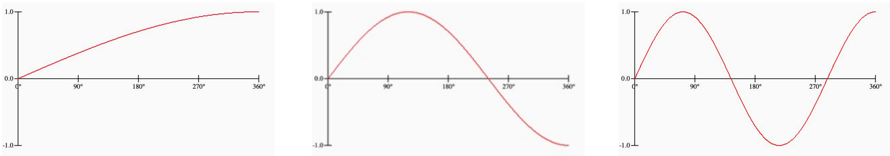


Figure 3-3 Vibration Modes of a Pipe Open at One End

3.4 Waveforms

Two common waveforms are the sawtooth and the square wave. The sawtooth sounds like a violin and has all the harmonics, and the square wave is the basic sound used in electronic organs and has all the odd harmonics.

To generate a sawtooth wave the octave harmonics are added with successively reduced amplitudes:

$$y = \sin(\theta) + \frac{1}{2}\sin(2\theta) + \frac{1}{3}\sin(3\theta) + \cdots$$

The reference figures of the harmonics are a circle of half the radius being traversed at twice the frequency, a circle of one third of the radius being traversed at three times the frequency, etc.

For a square wave, only the odd harmonics are used:

$$y = \sin(\theta) + \frac{1}{3}\sin(3\theta) + \frac{1}{5}\sin(5\theta) + \cdots$$

It is used in electronic organs because a square wave is easy to generate electronically by switching a circuit on and off.

The triangular wave has odd harmonics of alternating signs with amplitudes $\frac{1}{N^2}$:

$$y = \sin(\theta) - \frac{1}{3^2}\sin(3\theta) + \frac{1}{5^2}\sin(5\theta) - \cdots$$

Another variant is an approximation to a harpsichord sound given in [Sethares] p.233[1] as

$$y = \sin(\theta) + a^2\sin(2\theta) + a^3\sin(3\theta) + \cdots$$

[1] "The amplitude of the partials is assumed to die away at a rate of $.75^n$, where *n* is the partial number. Surviving historical harpsichords vary considerably in these parameters. The low strings of some have more than 80 discernible partials, decreasing with an exponent as high as 0.9, whereas the strings of others display as few as 8 partials with a more rapid decay."

With the amplitude factor for harmonic n being $a^{(n-1)}$ where a is between 0.75 and 0.9. This progression can be applied to sawtooth and square waves, but not to triangular.

3.5 Questions

My original question when I started this research was:

- If a sine wave is constructed from a circle, from what figures are other waves constructed?

And arising from this:

- Can they tell us anything useful?
- Can they be related to other waves such as earthquakes, quantum wave equations, etc.?
- In quantum physics, where particles have a wave representation, what does the wave represent? What does the corresponding polar figure represent?

3.6 Vector Representation

The reference figure can be represented in polar coordinates. Adding harmonics is equivalent to adding the vectors of the component sine waves.

Figure 3-4 below on the left shows the vectors to the reference circles for the angle of 10° for the fundamental sin(10°) in blue, the second harmonic $\frac{1}{2}\sin(20°)$ in red, the third harmonic $\frac{1}{3}\sin(30°)$ in purple and the fourth harmonic $\frac{1}{4}\sin(40°)$ in yellow. The addition of the vectors is shown on the right.

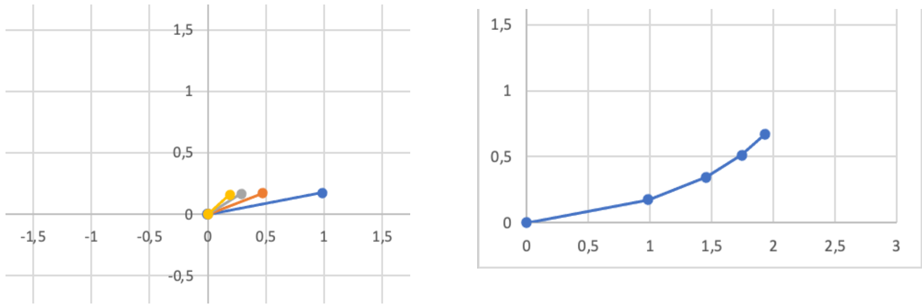


Figure 3-4 Vectors of 4 Harmonics

Note that the sum of the vectors does not arrive at the figure at the point indicated by the fundamental angle (10°), but at a point further round (see Figure 3-5).

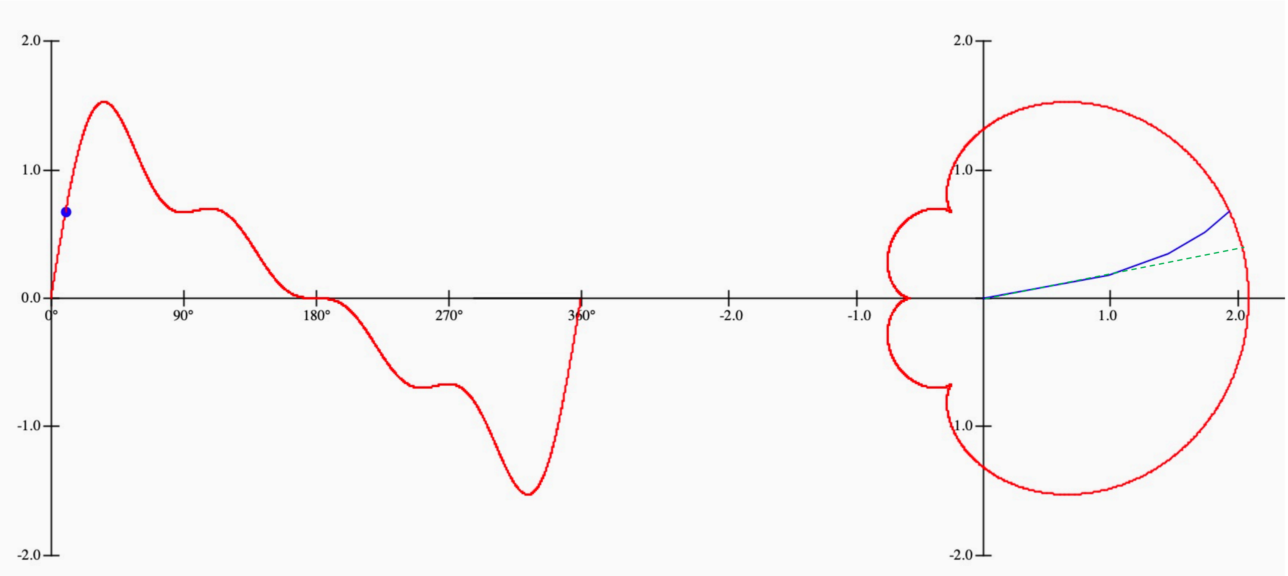


Figure 3-5 Vector of 4 Harmonics with Resultant Wave and Reference Figure

3.7 Constructor Figure

To obtain the constructor figure, i.e., the figure from which one can derive the resultant wave in the same way as the sine wave is derived from the circle, we must produce (extend) the first vector, sin(x), until it has the same y-value as the resultant vector.
For a wave function of the form $f_s = \sin(\theta) + \frac{1}{2}\sin(2\theta) + \frac{1}{3}\sin(3\theta) + \cdots$ or similar,

$$y = f_s(\theta)$$

$$r = \frac{y}{\sin(\theta)} = \frac{f_s(\theta)}{\sin(\theta)}$$

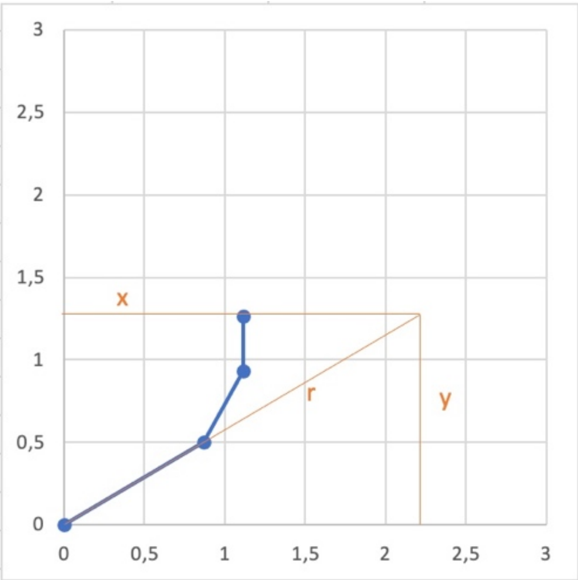
$$x = \sqrt{r^2 - y^2}$$

When using this, the sign of x must be changed to negative for $90° < \theta < 270°$.
Alternatively

$$x = \frac{y}{\tan(\theta)}$$

In which x is not defined for θ = 0° and θ = 180°, where tan(θ) = 0. There can also be precision problems when tan(θ) becomes very small or very large.

In the Figure 3-6 the red shape is the result from the vectors with the point for 30° shown at the end of the blue vectors. The orange shape is the constructor figure which could be used to construct the wave by laying off the angles – the orange dot is the point for 30° and has the same y-value as the end of the blue vectors. It can be seen that the two figures are different!



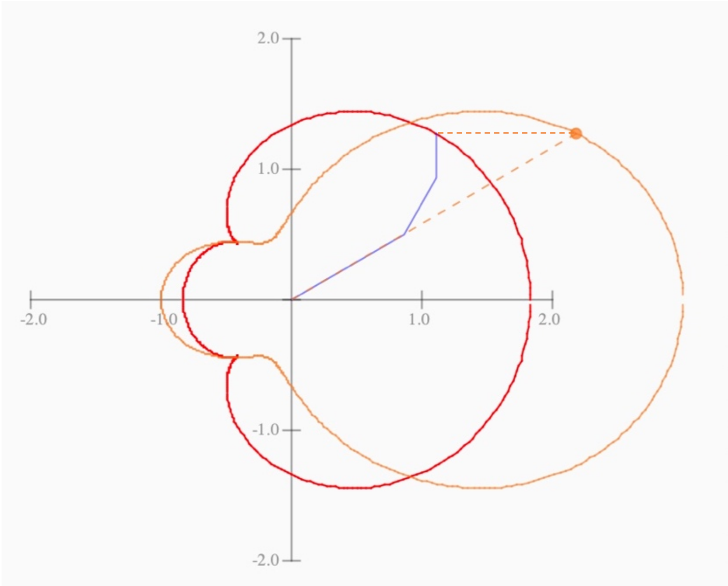


Figure 3-6 Derivation of Constructor Figure for 3 Harmonics

3.8 Incomplete Constructor Figures

It is not always possible to obtain a complete constructor figure for a given wave. If the amplitude of the wave is negative at any angle between 0° and 180°, or if the amplitude is positive any angle between 180° and 360°, the line at the given angle will never reach the y-value in the Cartesian two-dimensional space. Figure 3-7 shows this for a wave synthesised from frequences of 2 and 3 of equal amplitude the line of the constructor figure at 110° must be followed upwards until it reaches the height of the wave, but the wave is negative at that point so can never be reached. Also, as the y-value approaches zero, the length of the line becomes infinite or undefined (hence the gap at the right in Figure 3-6[2] and Figure 3-7).

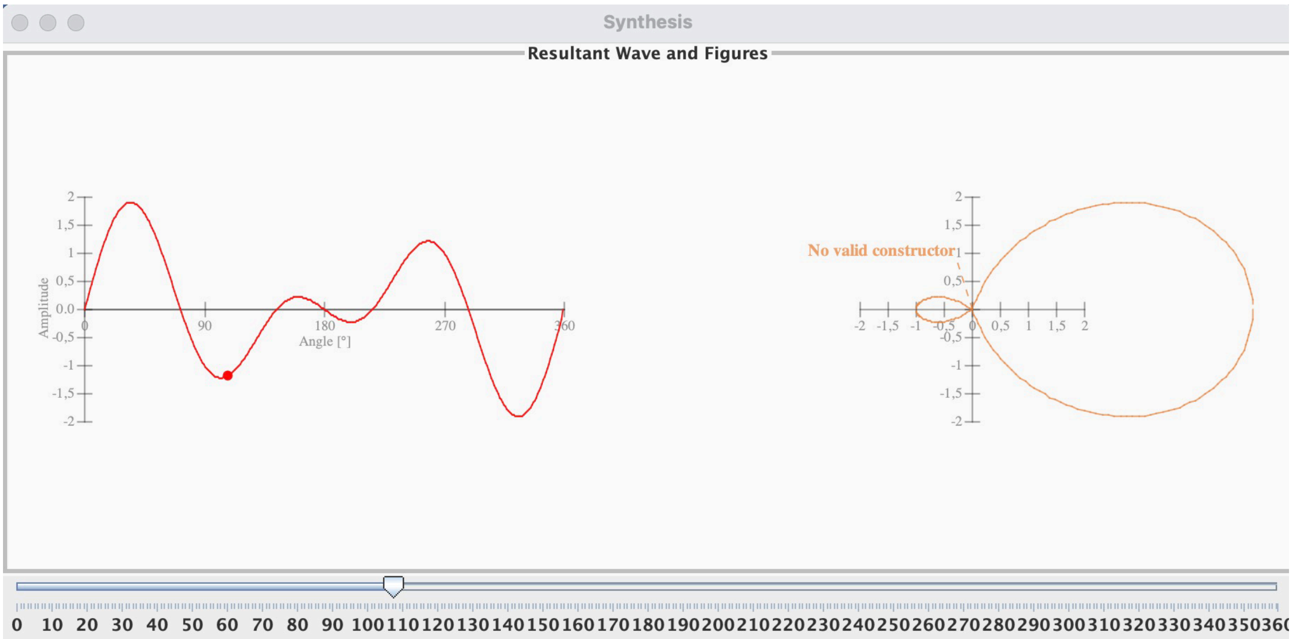


Figure 3-7 Wave with no Valid Constructor

Question: Is there a different coordinate space that would make this work, e.g. a cylinder or sphere or torus?

3.9 DC in Constructor Figures

A special case of this problem occurs when a wave that would have a constructor figure contains DC (direct current) components, i.e., values that do not change with time and therefore have a frequency of 0. To deal with this, we can initially omit the DC components from the calculation of the constructor figure and add the total DC at the end to shift the whole figure up or down the y-axis accordingly. This also shifts the origin of the constructor figure. Figure 3-8 shows a wave with a DC component of 1.0 and two harmonics:

$$y = 1.0 + \sin(x) + ½\sin(2x).$$

[2] There is no gap on the left at 180° because the line has been drawn from the previous valid point to the next valid point. The occurrence at 0° was not treated in this way because this is the starting point.

For the red reference figure, the first vertical vector from the origin up the y-axis to 1.0 is the DC component and the two harmonics go from there. For the orange constructor figure, the whole figure is shifted up 1.0 by the DC component. (Compare this to Figure 3-11.)

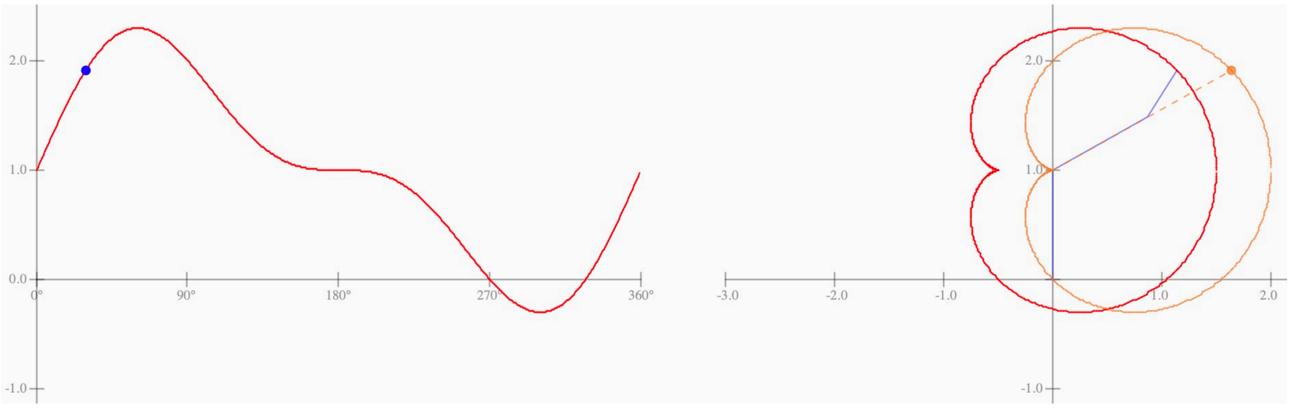


Figure 3-8 A DC Component of 1.0 and Two Harmonics

For angles where there is no valid constructor, this is indicated on the diagram as shown in Figure 3-9, where at the angle shown of 130° cannot meet the negative value of the phase-shifted sine wave.

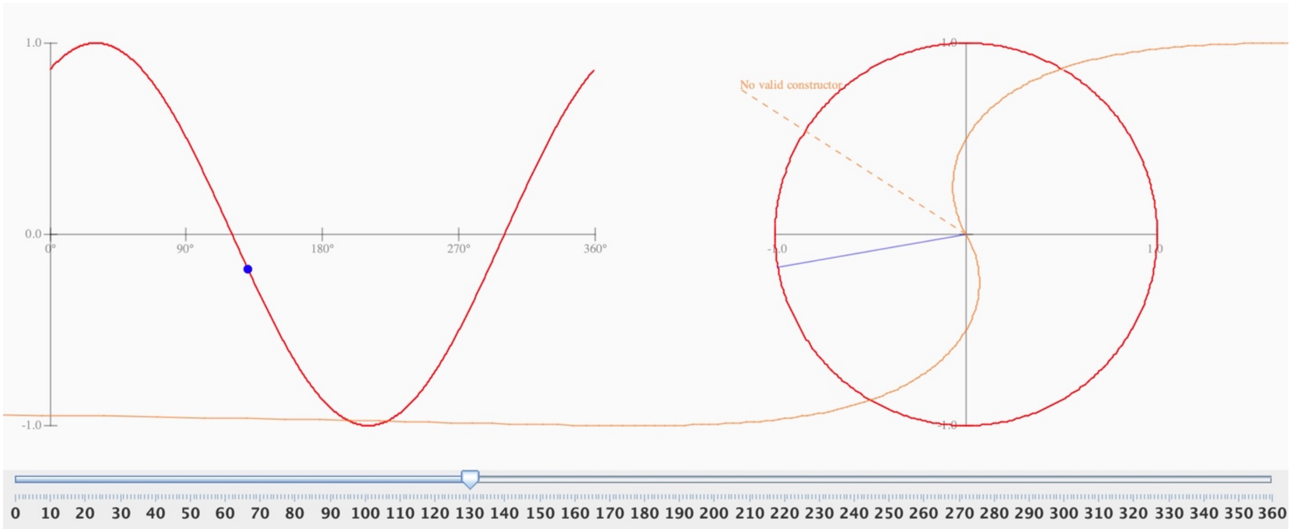


Figure 3-9 Example of Angle with no Valid Constructor

3.10 Difference between Reference Figures and Constructor Figures

For the fundamental alone the reference and constructor figures are the same. See Figure 3-10.

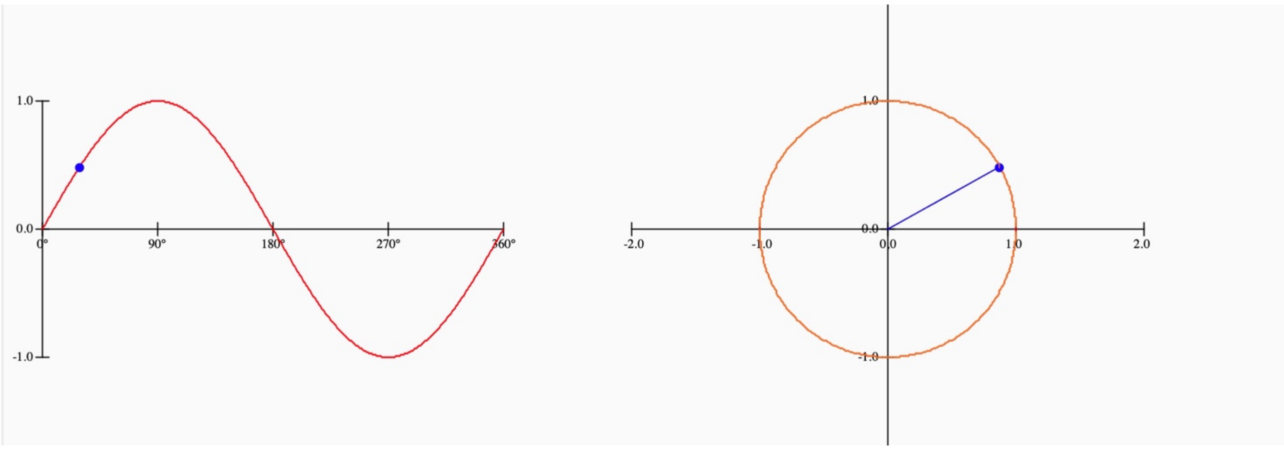


Figure 3-10 Reference and Constructor Figures for Fundamental

With the second harmonic (or first overtone) the shape is the same, but the constructor figure is translated 0.5 units to the right. See Figure 3-11. The proof is in appendix 11.

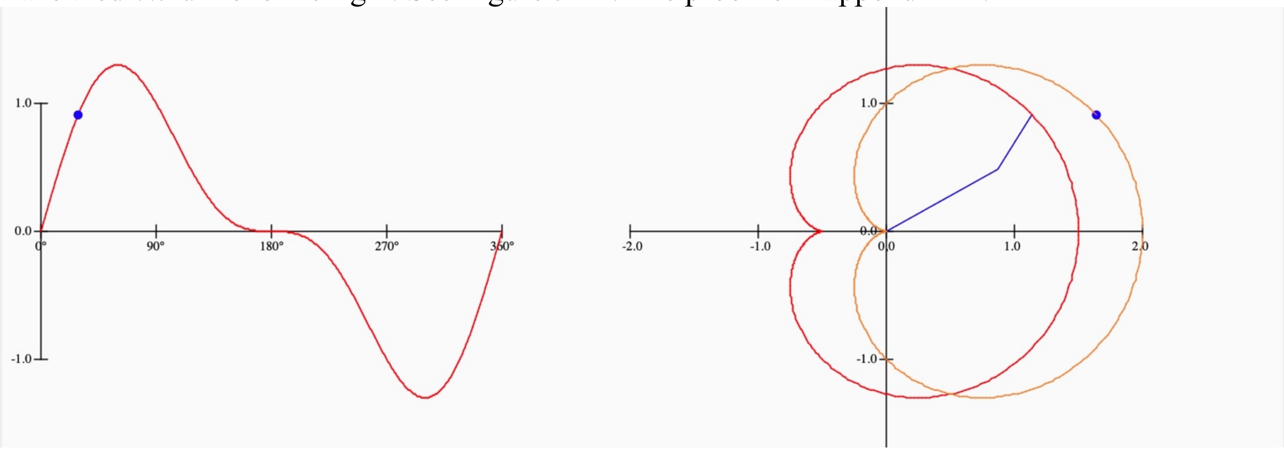


Figure 3-11 Fundamental and Second Harmonic with Reference (red) and Constructor (orange) Figures

From the third harmonic on, the constructor figure looks slightly different to the reference figure – see Figure 3-12.

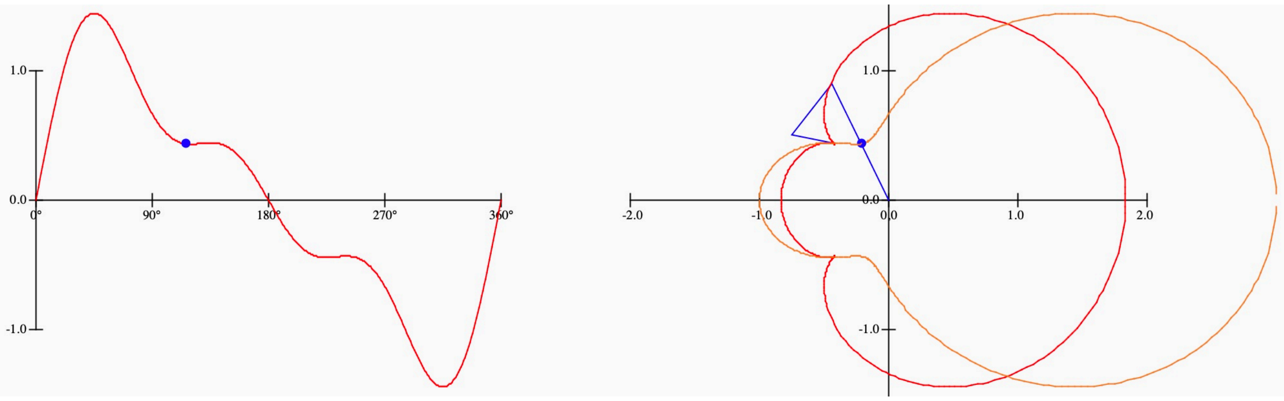


Figure 3-12 Fundamental, Second and Third Harmonic with Reference (red) and Constructor (orange) Figures

Using the visualisation program, we observe that as we pass through the angles of the cusps where the resultant wave is almost flat, the end of the vector lingers at the cusp, and the constructor must be nearly flat to correspond to the resultant wave – see Figure 3-13.

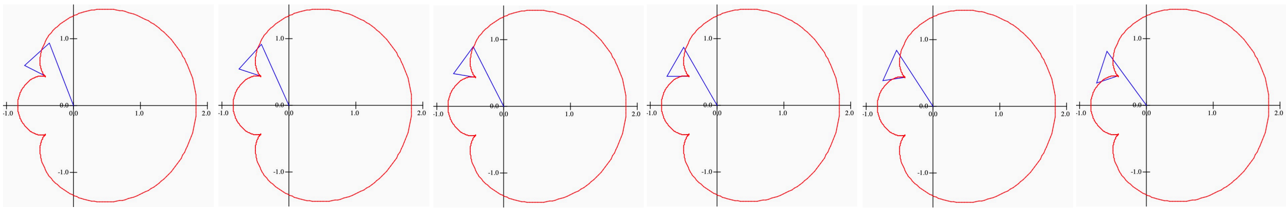


Figure 3-13 Vector passing through cusp around 120°

3.11 The Sound

The waves are synthesised with the overtones having relative frequencies to the fundamental. To play the sound, a wave must be produced with a particular frequency. A wave has a wavelength, which is the distance from 0° to 360°. The frequency is related to the wavelength by the speed of sound in the medium through which it is travelling (usually air):

$$speed = frequency * wavelength$$

For a vibrating string or pipe, the wavelength is fixed and determined by the length of the pipe or string[3] (see 3.3), so:

$$frequency = \frac{speed}{wavelength}$$

(This explains why inhaling Helium and speaking gives the voice a higher frequency – the speed of sound in Helium is about three times higher than in air.)
The program generates a sound at a selected frequency. (In modern concert pitch A is 440Hz, middle C is 263.18139Hz.)
The program allows the frequency to be set, and this could be used for waves other than sound waves to make them audible.

3.12 The WAV File

The sound can be saved as a WAV file (see 7.3.2). The file consists of digital samples of the calculated sound wave at the selected sampling frequency on one channel (mono, not stereo). The length of the file depends on the duration specified by the user, the default being 1 second. A frequency of 1Hz for 1 second at a sampling rate of 44000 samples/second will therefore give a WAV file with 44000 samples of one cycle. Doubling the frequency halves the number of samples per cycle. This in turn affects the frequency resolution of the Fourier transform that can be performed on the WAV file – see 3.13.

3.13 Analysis – the Fourier Transforms

There is plenty of literature about the Fourier transform, which transforms the representation of the wave as amplitude over time into a representation of amplitude against frequency, and the basics are not repeated here.
The analysis function in the program uses either a simple Discrete Fourier Transform (DFT) or a Fast Fourier Transform (FFT) or a Chirp Z-Transform (CZT). DFT is slower than FFT but more flexible in the choice of sample window – the FFT needs a window whose length is a power of 2. The CZT can effectively zoom in to a specific range of frequencies within the DFT. The time taken for the DFT is proportional to n^2, the square of the number of samples in the window. The time taken for the FFT is proportional to nlog(n).

[3] This is simplified – the frequency of a string also depends on its mass per unit of length and its tension.

The following optimisations are implemented in the DFT to give the best results:

- The algorithm is based on an input of only real values (rather than complex).
- Only the left half of the symmetric double-sided result is calculated and shown (symmetry about the Nyquist frequency). The amplitude values are therefore doubled as the total energy is usually divided between the two symmetric frequencies.
- The arctan function is very sensitive to small values, so phases may not be accurate.
- Averaging and STFT calculations are distributed over the available processors of the computer hardware.

The FFT is a faster algorithm but requires that the number of samples in the window is a power of 2 (…, 64, 128, 256, 512, …). This also only uses real values; the imaginary values of the complex numbers all being set to 0.

The Chirp Z-Transform, implemented here in its simple form (not actually using chirp signals[4]), calculates the Fourier Transform over a given range of frequencies with a given step (delta) between the frequencies. Note that this does not increase the effective resolution, but zooms in on a part of the spectrum, and will distribute the amplitude of a DFT frequency bin over the neighbouring bins.
It is not possible to increase the frequency resolution without losing resolution in time. Taking a longer window will give better frequency resolution, but if the frequency is not constant, more frequencies will be present in the time window and less will be known about when the frequencies occurred. This is a kind of uncertainty principle: the better the frequency resolution, the worse the time resolution and vice-versa.
A further method is to use wavelets which use variable windows to optimise the time vs. frequency trade-off. This has not (yet) been implemented in the program.

The frequency spectrum x-axis starts at 0Hz which is the DC (direct current) component of the wave if the amplitude is not symmetrical about 0 amplitude.

The following are worth bearing in mind:

- The DC component is the mean over the period.
- The integral over the period p is the same for any starting point t_0, so integrating over *0 to p* is the same as from t_0 *to* $p+t_0$. In other words, it is not necessary to set the marks at the zero crossings of the wave, but this will naturally give different phase results.
- The human ear is phase deaf and cannot perceive differences in the phases of the harmonics (Ohm's law of acoustics). For audible acoustic waves the phases can be ignored by setting all the phases to zero in the synthesis.

Various windowing functions are provided. These are useful when the window being analysed is not a whole number of cycles or the sampling rate is not divisible by the frequency (there are not two samples exactly 360° apart).

[4] Computers have advanced in processing power since the 1990's such that it is not necessary to use the fast algorithm of Bluestein or Rabiner with of convolutions with chirps by forward and inverse FFTs. The complexity of the programming is not justified by the gain in speed.

A Short Time Fourier Transform (STFT) is used to analyse the spectrum over time rather than at a single point in time. The result is usually displayed as a spectrogram (an example is in Figure 3-14). These have the frequency on the y-axis and time on the x-axis. The colour represents the amplitude of each frequency at each point in time. However, these are often difficult to interpret. This program displays the spectrum as an animation over time: when the animation is played the spectrum changes, the speed can be controlled or the animation stepped from one time unit to the next.

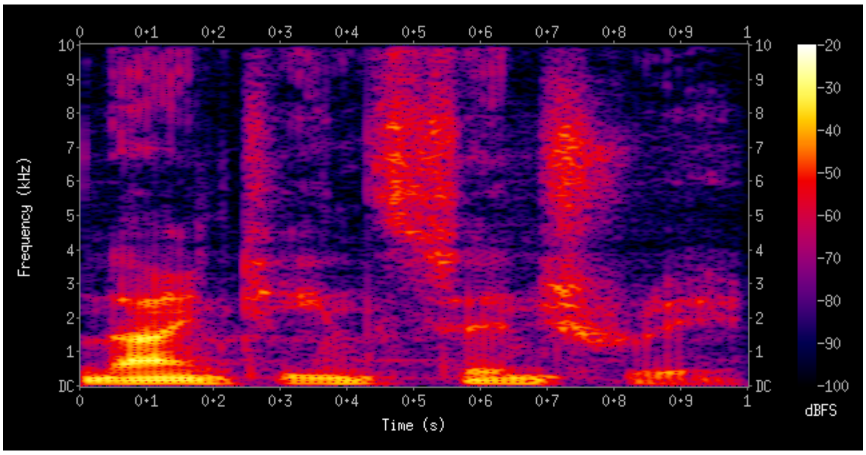


Figure 3-14 Spectrogram By Aquegg - Own work, Public Domain, https://commons.wikimedia.org/w/index.php?curid=5544473

4 Possible Uses

4.1 Sounds

The most obvious and most frequent use is to analyse sound data to determine its spectrum.

4.2 Planetary Orbits

To show the orbit of another planet as seen from Earth, the orbit of that planet must be combined with that of Earth. Approximating the orbits with circles rather than ellipses can do this. The resulting reference figure shows the required orbit. The program is designed to show one cycle, but by multiplying the frequencies by a common factor, multiple cycles can be shown, limited in resolution by the number of points calculated.

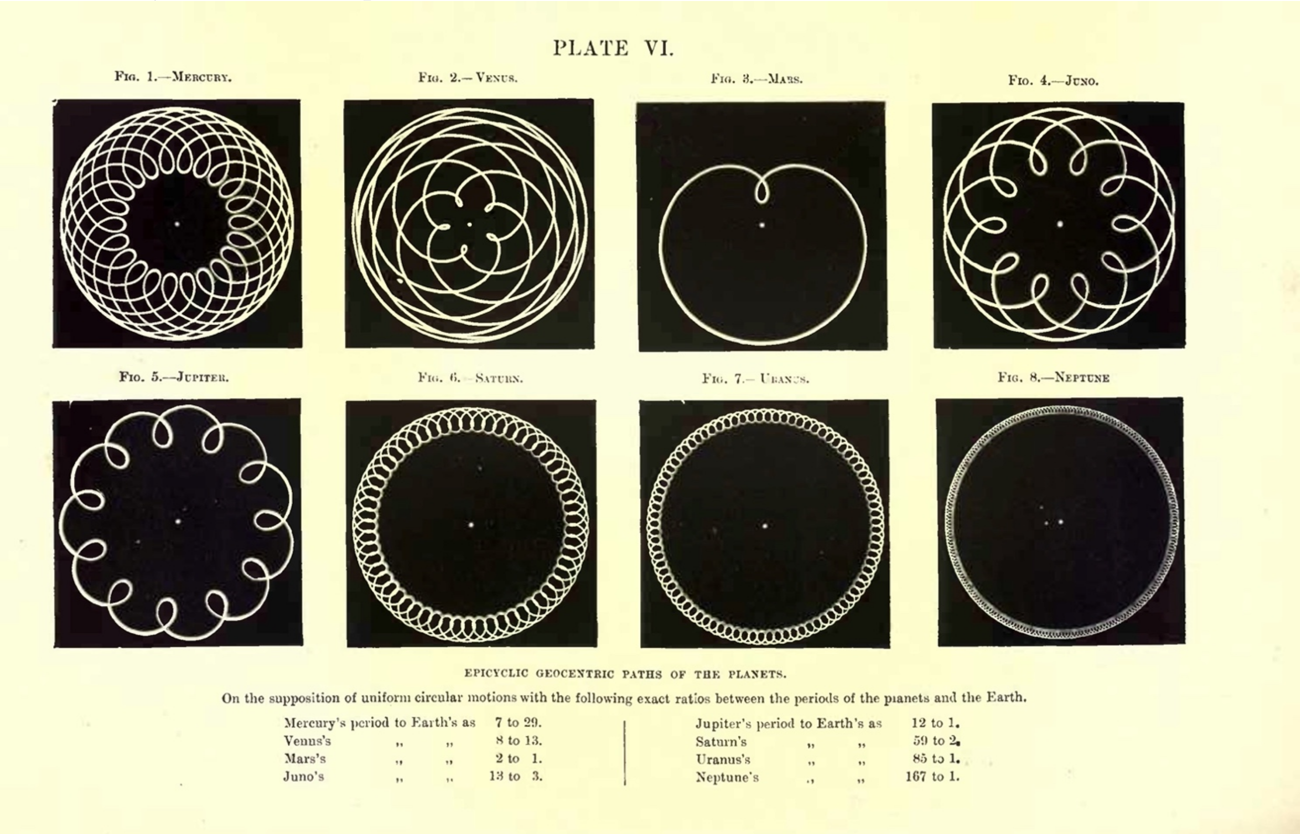


Figure 4-1 Some Planetary Orbits as Seen from Earth. Source: Old and New Astronomy. Proctor, Richard Antony. 1892. Longmans London. Between pp. 164&165. https://archive.org/details/oldnewastronomy00procuoft/page/n185/mode/2up

The commonly shown diagrams in Figure 4-1 can be reproduced by converting the orbit radii as amplitudes relative to Earth's orbital radius and the frequencies with suitable factors in relation to Earth's orbit frequency as shown in Table 4-1 (decimal is comma) – this is with more modern data from http://gerdbreitenbach.de/planet/planet.html.

Table 4-1 Conversion of Planetary Orbits as Waves

	R rel Earth	Period T	Frequency 1/T	1/R	Factor	Best Freq. 000	Best Ampl. 000	Best Freq. 001	Best Ampl. 001
Earth	1	1	1,00000	1,00000					
Mercury	0,387	0,24084	4,15213	2,58398	8	8,00000	1,00000	33,21707	0,38700
Venus	0,723	0,61519	1,62551	1,38313	10	10,00000	1,00000	16,25514	0,72300
Mars	1,524	1,88082	0,53168	0,65617	2	2,00000	1,00000	1,06337	1,52400

	R rel Earth	Period T	Frequency 1/T	1/R	Factor	Best Freq. 000	Best Ampl. 000	Best Freq. 001	Best Ampl. 001
Jupiter	5,203	11,862	0,08430	0,19220	15	15,00000	1,00000	1,26454	5,20300
Saturn	9,539	29,458	0,03395	0,10483	30	30,00000	1,00000	1,01840	9,53900
Uranus	19,201	84,014	0,01190	0,05208	50	50,00000	1,00000	0,59514	19,20100
Neptune	30,047	164,793	0,00607	0,03328	50	50,00000	1,00000	0,30341	30,04700
Pluto	39,482	247,94	0,00403	0,02533	50	50,00000	1,00000	0,20166	39,48200

For example, Mars is shown in Figure 4-3 with the above data, which does not quite join up after one cycle.

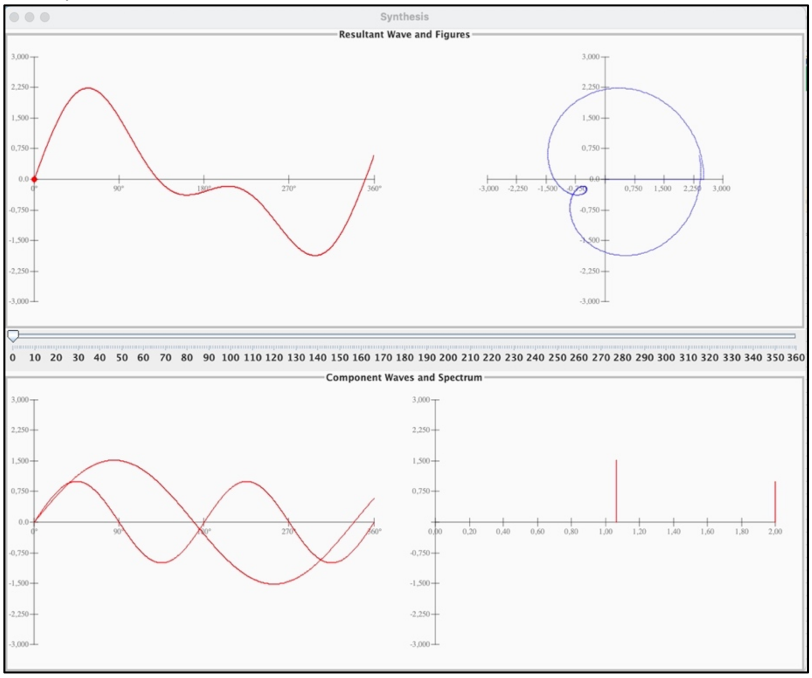


Figure 4-3 Relative Orbit of Mars as Seen from Earth – Modern Data

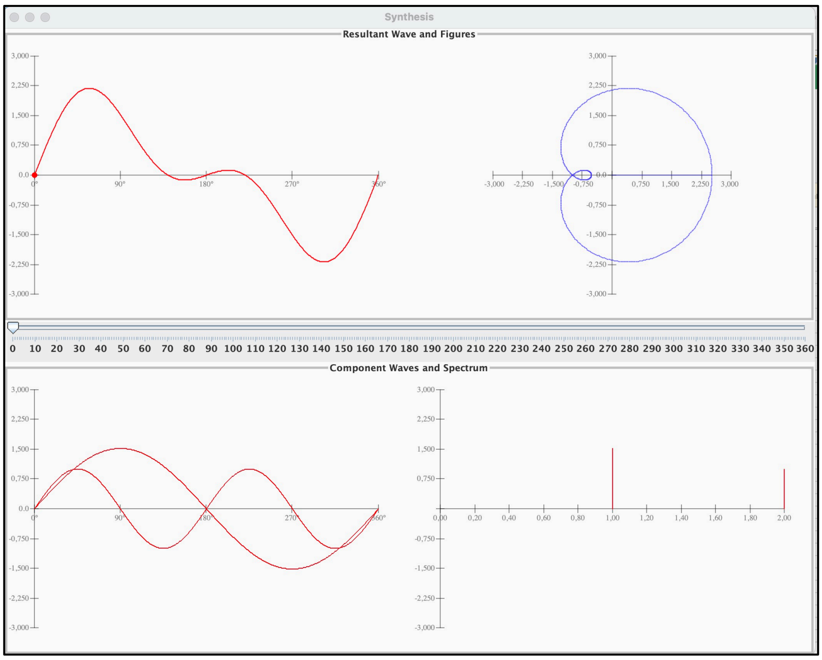


Figure 4-2 Relative Oribits of Mars Seen from Earth - Proctor's Data

Using the frequency ratio given by Proctor in Figure 4-1 we can reproduce his image – see Figure 4-2.

4.3 Musical Temperament

Musical intervals can be shown by combining frequencies in the appropriate ratio.
Figure 4-4 shows the unison 1:1, that is two equal waves giving a sine wave with amplitude 2.

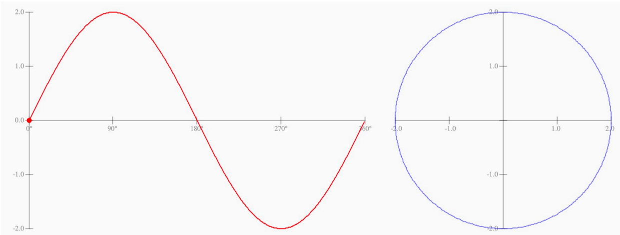


Figure 4-4 Unison

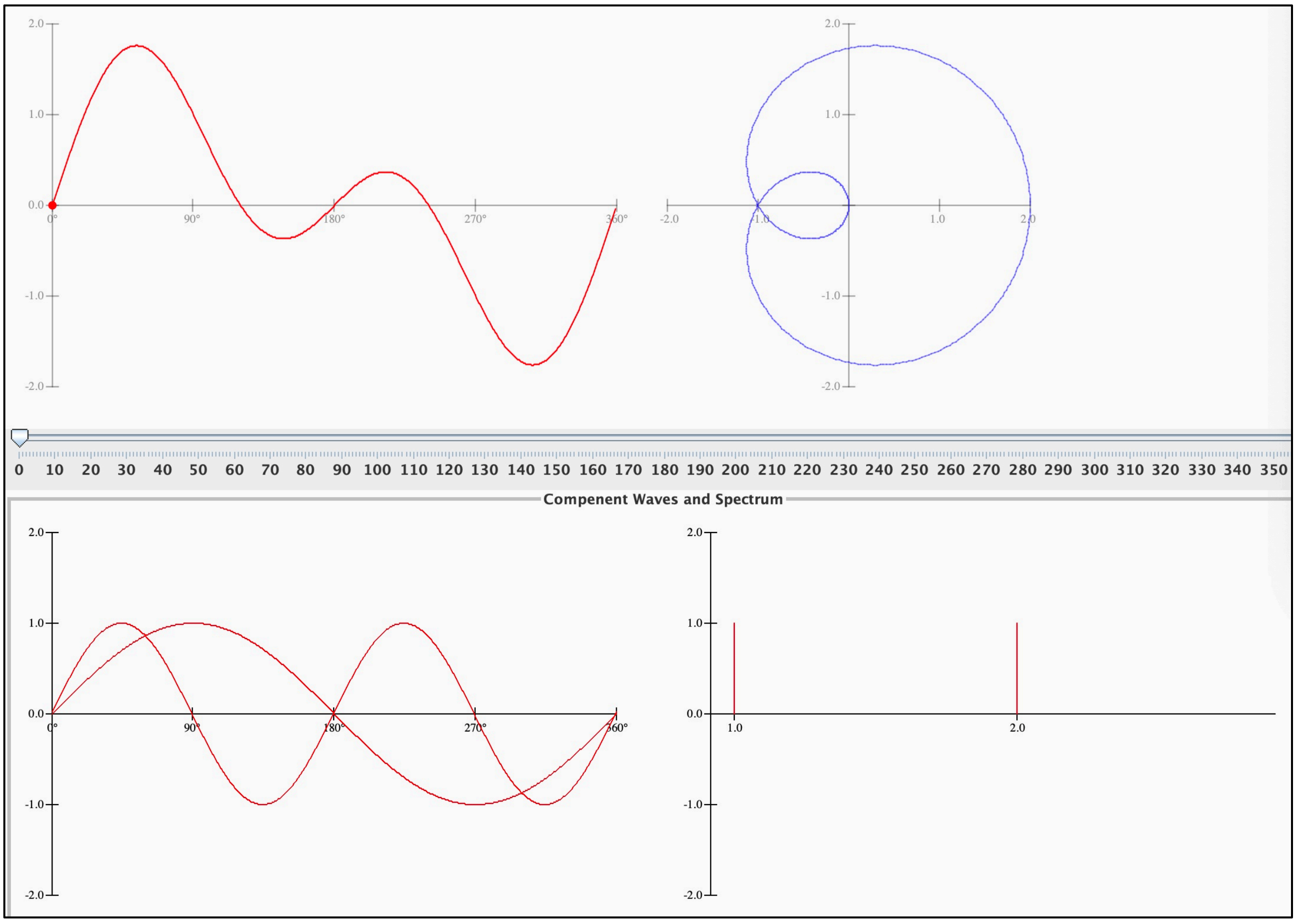


Figure 4-5 Octave

Figure 4-5 shows the octave 1:2 with the second wave twice the frequency of the first and the same amplitude.

Figure 4-6 shows a perfect fifth or 2:3 as 1: 1.5.

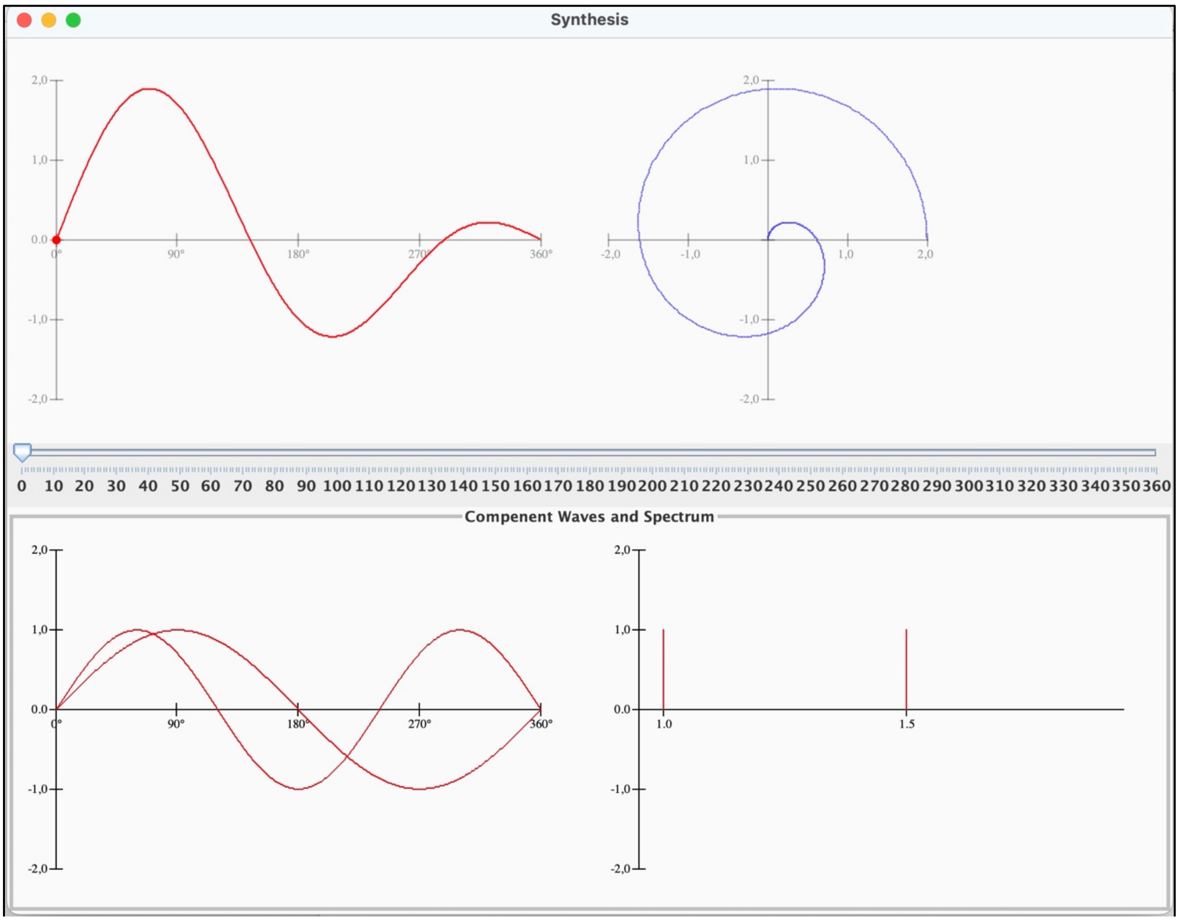


Figure 4-6 Perfect Fifth

Figure 4-7 shows a "wolf" fifth that occurs with meantone temperaments, a ratio of 1:1.46 instead of 1:1.5. Comparing this with Figure 4-6, the second wave, and thus the resultant wave, does not return to 0 at 360° and the reference figure does not reach the origin.

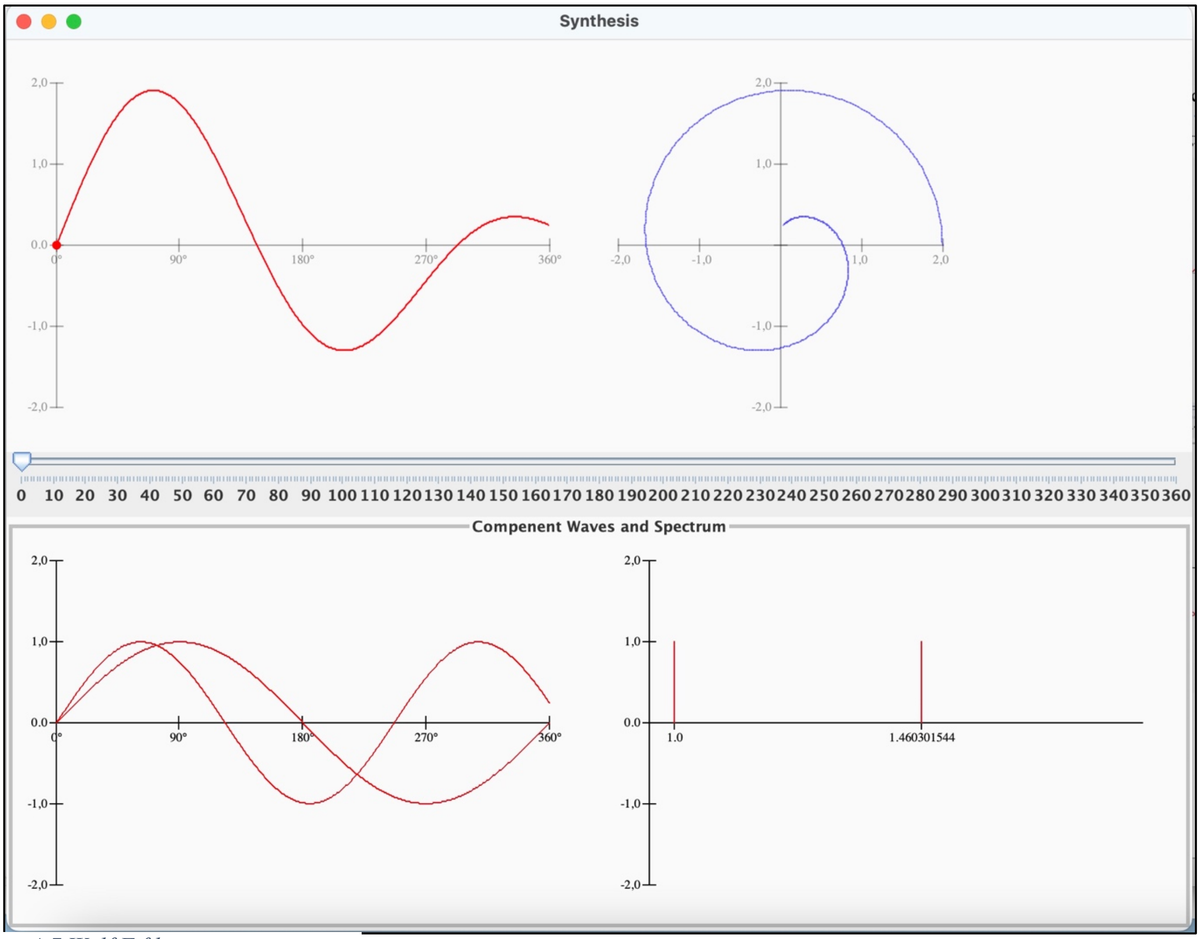


Figure 4-7 Wolf Fifth

4.4 Drawing

The reference figure can be used to construct almost any line drawing from the epicycles that form the reference figure. There are examples on the Internet, e.g.
https://contra.medium.com/drawing-anything-with-fourier-series-using-blender-and-python-c0881e1b738c
https://www.jezzamon.com/fourier/index.html
https://blog.wolfram.com/2013/05/17/making-formulas-for-everything-from-pi-to-the-pink-panther-to-sir-isaac-newton/
https://www.wolframalpha.com/input/?i=butterfly+curve
https://www.youtube.com/watch?v=qS4H6PEcCCA

This one (Figure 4-8) was constructed by modifying https://www.jezzamon.com/fourier/index.html to extract the component wave parameters.

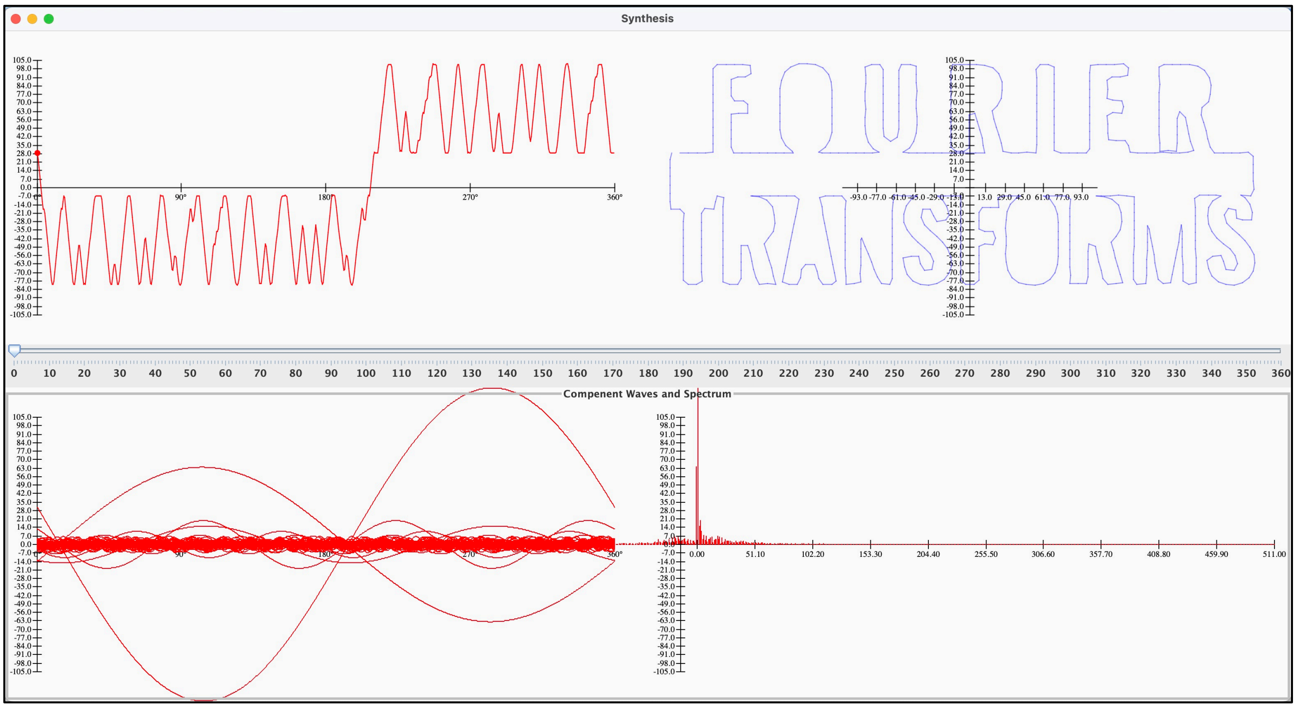


Figure 4-8 Drawing from waves

4.5 Seismology

4.6 Oceans and Lakes

4.6.1 Tides

4.6.2 Waves

4.6.3 Hydrophones

4.7 Gravitational Waves

See [GEO600], [LIGO], [Heinzel].

5 Installing the Program

5.1 Download

The program itself can be downloaded from:
https://github.com/Goldberg53/harmonics.git
with no guarantees that it will work as expected.

5.2 Prerequisite

The program is written in the Java programming language. The Java runtime environment (JRE) must be installed before the program can run.
Obtain the official free JRE from Oracle at www.java.com and install it as instructed.

5.3 Caveats

The program has been tested, but as with any software, there can be no guarantee that all results are correct, and no responsibility is taken for any harm caused by using the program.

The following differences between the systems are known:

- Apple MacOS has tighter security restrictions on downloading and running unsigned files. The program file is not signed, so the user must confirm download and execution.

5.4 Installing the Program

The program does not need to be installed as an application. Download the .jar file to a convenient place on your computer. You can create a shortcut to the jar file and copy the shortcut to the desktop.
This user manual is included in the .jar file and can be extracted and opened with the “Extract Manual” function in the Help menu. Further information is in the Help function in the Help menu.

5.5 Removing the Program

To remove the program from your computer, simply delete the .jar file.
If you extracted the user manual, you can delete the extracted file in your home folder, called Harmonics.pdf or similar at any time.
If desired, uninstall the Java runtime environment in the usual way e.g. on Windows with Programs and Features in the Control Panel, on Mac by deleting the files.

5.6 Error Handling

Error messages will appear in dialogue boxes on the screen. Some programming errors may crash the program with a stack dump to the Java console.

6 Main Window

6.1 Menu

The Program Menu has

Close

Terminates the program.

Help

The help menu contains the items Release Notes, About, Help and Extract Manual.

Release Notes shows the version history of the program with the changes made to each version.

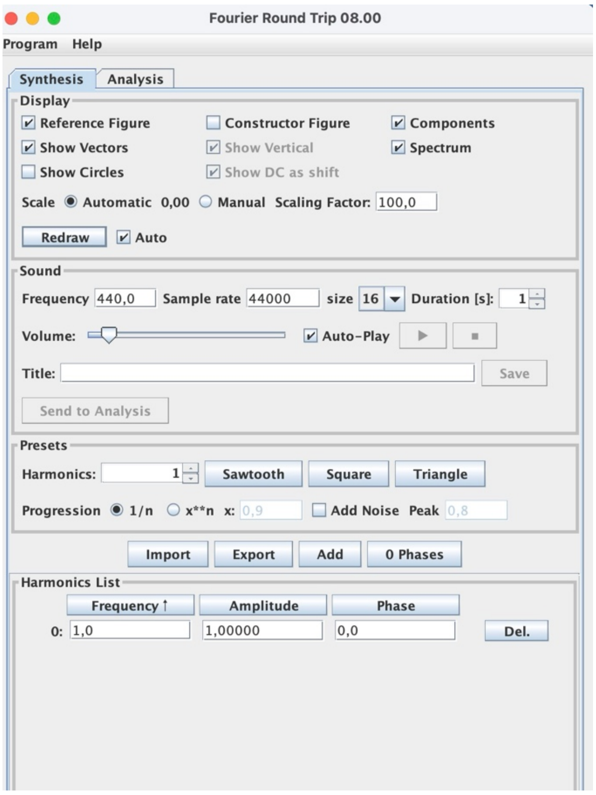
About gives the origin and development information of the program.

Help gives assistance with extracting the manual.

Extract Manual will extract this user manual document from the .jar file and open it. It first tries to extract it to the directory in which the .jar file is located, and if that fails, tries the user's home directory. The location is shown in a confirmation message. The manual will then be opened in the PDF reader. If any of these operations fails, an appropriate message is given – see Help for alternatives.

6.2 Tabs

The main window has two tabs:



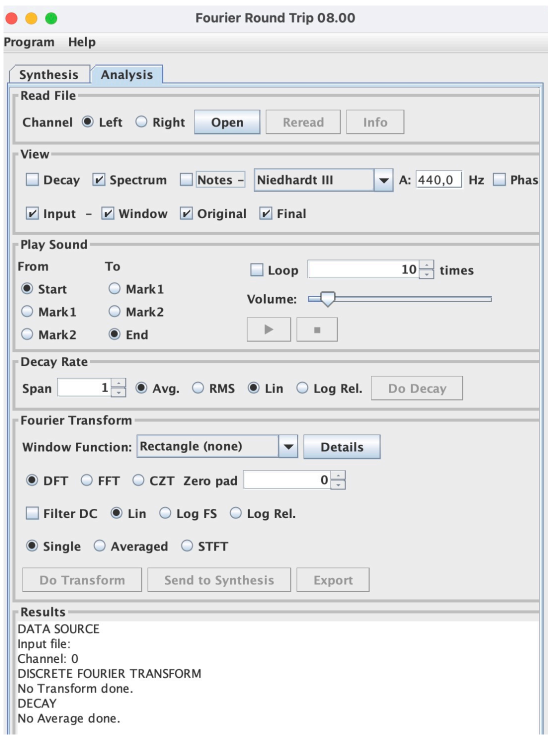


Figure 6-1 Tabs of Main Window

The left tab contains the settings for the synthesis, the right tab those for analysis.

6.3 Axes and Scaling

Synthesis Window

The resultant wave (top left) and the component wave (bottom left) diagrams are scaled vertically to accommodate the maximum amplitude. They are calibrated such that the end of the axis is the next round number equal to or above the maximum needed. They are then divided into a fixed number of divisions. Any numbers that overlap others are omitted.

The wave diagrams are scaled horizontally to take up half of the available window size and calibrated 0° to 360°.

The reference and constructor figures (top right) are scaled to match the amplitude and axis calibration of the waves vertically and with the same scaling and calibration horizontally. The diagram is positioned to be centred at ¾ of the window width and does not rescale with the window size in order to preserve the correct aspect ratio and to give some leeway in accommodating wider diagrams.

The frequency spectrum (bottom right) has a vertical axis corresponding to the waves as above. The horizontal frequency axis is given ten ticks up to the maximum frequency. As above, numbers are omitted if they would overlap. The y-axis is placed over the zero frequency, also when there are negative frequencies.

Analysis Window

The wave from the input file is calibrated vertically according to the amplitude similarly to the waves in the synthesis window. The y-axis calibration is in the sampling units, e.g. -127 to +127.

The horizontal axis is the time in seconds (to three decimal places giving milliseconds).

The Frequency Spectrum panel gives the relative amplitudes on the y-axis with 1 as the largest or relative amplitudes in decibels. The frequencies on the x-axis are calibrated both in bins and in Hz.

The Phase Spectrum panel gives the phase in radians on the y-axis (π is 3.14) and the same frequencies as the frequency spectrum on the x-axis.

7 Synthesis

7.1 Notes

DC

To make the visualisation correspond to the use of sine waves starting with an amplitude of 0 at 0 degrees, we use sine wave throughout. The more usual convention for Fourier analysis is to use cosine waves because this allows DC components to be included – cos(0) = 1.
We therefore adopt the convention of treating a zero frequency as DC for the sine waves.
See also 3.9.

7.2 Synthesis Display

A new synthesis window is created with the **New** button and shows the results depending on the parameters set in the main window. It is updated with the **Redraw** button.

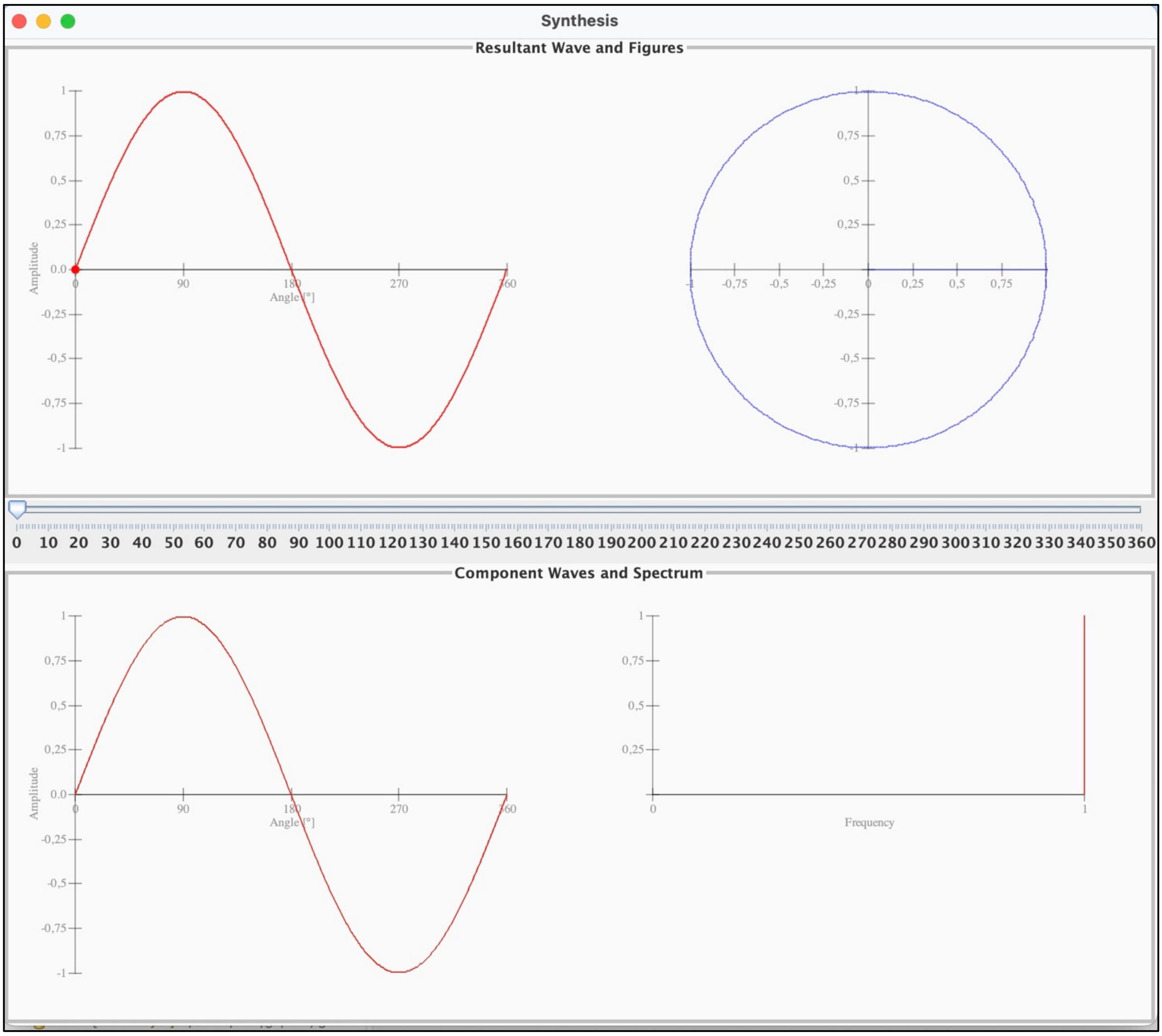


Figure 7-1 Synthesis Window

In the top half, the resultant wave on the left and optionally the reference and constructor figures superimposed on each other on the right.
The slider moves a blue dot along the resultant wave, shows a vector representation of the components on the reference figure in blue and an orange dot on the constructor figure with a dotted

orange line at the corresponding angle. All these are for the angle selected with the slider. For angles where no constructor is possible, the segments of the constructor figure are omitted.
The bottom half shows the component waves superimposed on each other on the left and the frequency spectrum on the right.

Axes and Scaling
The resultant wave (top left) and the component wave (bottom left) diagrams are scaled vertically to accommodate the maximum amplitude. They are calibrated such that the end of the axis is the next round number equal to or above the maximum needed. They are then divided into a fixed number of divisions. Any numbers that overlap others are omitted.
The wave diagrams are scaled horizontally to take up half of the available window size and calibrated 0° to 360°.
The reference and constructor figures (top right) are scaled to match the amplitude and axis calibration of the waves vertically and with the same scaling and calibration horizontally. The diagram is positioned to be centred at ¾ of the window width and does not rescale with the window size in order to preserve the correct aspect ratio and to give some leeway in accommodating wider diagrams.
The frequency spectrum (bottom right) has a vertical axis corresponding to the waves as above. The horizontal frequency axis is given ten ticks up to the maximum frequency. As above, numbers are omitted if they would overlap. The y-axis is placed over the zero frequency, also when there are negative frequencies.

7.3 Synthesis Parameters

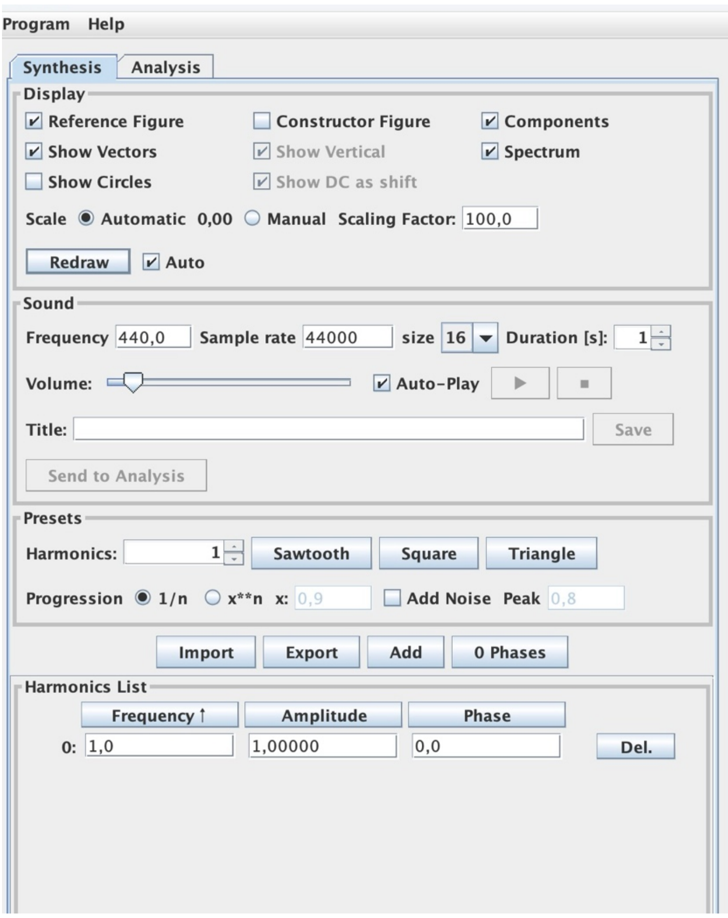


Figure 7-2 Synthesis Parameters Window

These are in the left tab of the main window – see Figure 7-2

7.3.1 Display Section

This section determines whether the reference and constructor figures are displayed.

Reference Figure

Check the box to display the reference figure.

Show Vectors

Check the box to display the vectors of the harmonics in the reference figure.

Show Circles

Check the box to display a circle for each harmonic. The circle traces the path of the end of the vector.

Constructor Figure

Check the box to display the constructor figure.

Components

Check the box to display the component waves.

Spectrum

Check the box to display the spectrum.

Show Vertical

Check the box to add the vertical line for the y-value and a horizontal line from there to the wave.

Show DC as shift

Check the box to move the origin of the constructor vertically for any DC (0 frequency) component. This will help in some cases where the constructor figure is incomplete. See section 3.8 and 3.9.

Scaling Factor

This is the scaling factor to make the graphics fit on the screen. Note that the automatic scaling is optimised for the height of the wave and the reference figure and may need to be scaled manually in some cases, in particular for the sideways direction of the reference and constructor figures, which can become very large.

When set to **Automatic**, the calculated scaling factor is shown on the right.

If **Manual** is selected, the value entered in the field will be used.

Redraw

Click this button to redraw the graphics. Depending on the number of harmonics, the calculation of the waves and the sound may take some time, so in that case a progress window is shown. The calculation can be cancelled.

Auto

If the box is checked the graphic will be redrawn automatically whenever a parameter is changed. The sound will also be played if "Play on Redraw" is selected in the Sound section (a sound already being played will be interrupted first).

7.3.2 Sound Section

These control the sounding of the resultant wave played via the computer's audio output.

Frequency

The frequency of the note to be played.

Sample Rate

The sample rate in samples per second to be used to write the WAV file. It is best to choose a rate that is divisible by the most important frequency to be measured, and for Fast Fourier Transform FFT), a power of 2.

Sample Size

The number of bits in each sound sample for playing the sound and sending it to the analysis function.

Duration

How many seconds the note will play when triggered, and how long the sound sent to analysis will be

Volume

Controls the volume of the note played.

Note: the volume is also affected by the volume setting on the computer itself and the connected amplifier and speakers.

NOTE: the volume control does not work at sample sizes above 16 bits, and these play at full volume. Reduce the computer's volume control before playing!

Auto-Play

The sound will be generated and the note played whenever the graphic is redrawn or any of the frequency, sample rate or duration are changed.

Title

Enter a one-line title (of effectively unlimited length). This is included as meta-data for the Track Title when the sound file is written with Save and in the CSV export.

Save

Click this button to save the sound as a .wav file. A file chooser will appear for selection of the file location and name. The file is saved as a single-channel (monophonic) sound sampled at the set sample rate, sample size 16 bits, with the duration given above. This can be read by the analysis function – see chapter 8. It includes meta-data for the track title (see above) and comments with the harmonics list (see 0 below).

Play►

Click this button to play the sound at any time.

Stop■

Click this button to stop the sound playing (useful if you have selected a long duration).

Sent to Analysis

Send the generated sound to the analysis function.

7.3.3 Presets Section

The buttons will create an approximation to various wave forms with the selected number of harmonics. The constructed values of frequency, amplitude and phase are populated into the Harmonics List below.

Harmonics

Select the number of harmonics to be populated by the other buttons.

Sawtooth

Creates the harmonics for a sawtooth wave.

Square

Creates the harmonics for a square wave

Triangle

Creates the harmonics for a triangular wave.

Progression

1/n or **x**n**

The progression of the amplitudes of the harmonics can be selected either as the usual 1/N or with exponentials x^n – see section 3.4. The latter is not applicable to Triangle.

x

The value of x can be entered in the field.

Add Noise

Tick this checkbox to add random noise to the generated wave.

Peak

Set the peak value of the added noise. The value is relative to the amplitudes of the harmonics below, e.g. 1.0 is the same amplitude as the default first harmonic.

7.3.4 Buttons

Import

Click this button to import the parameters from a CSV file (see 7.3.6).

Export

Click this button to export the parameters to a CSV file (see 7.3.6).

Add

Click this button to add another harmonic to the list. It will preset the frequency to the next integer above the last frequency in the list (which depends on the sorting of the list), preset the amplitude and the phase to 0. Note that it does not attempt to set the amplitude to the next in the progression, as it is assumed that this is for a manual setup. To add harmonics to a preset series, increase the number of harmonics and regenerate the waveform.

0 Phases

Click this button to set all the phases to zero. This is useful when the harmonics have been sent from an Analysis.

7.3.5 Harmonics List Section

This shows the harmonics being used. A list created from the Presets above can be modified. The list can be exported as a CSV file and imported from a previously exported file.

Sorting

Click on the column header buttons to determine how the harmonics list will be sorted. The sorting occurs immediately when the button is clicked. The current sorting is shown with an arrow, up for ascending and down for descending on the sorted column.

This can be used to choose how the vectors in the reference figure are shown: sorting by ascending amplitude will plot the longest vectors first.

Frequency fields

Set the relative frequency of the harmonic as a multiple of the fundamental frequency. Frequency 0 is the DC (direct current) component and Frequency 1.00 is the fundamental frequency. The frequency of the sound is set in the sound section.

Amplitude fields

Set the relative amplitude of the harmonic.

Phase fields

Set the phase of the harmonic in degrees. A phase shift of 90° makes the sine into a cosine.

Delete

Click this button to remove the harmonic from the list.

7.3.6 Import and Export Format

The CSV file must start with the first line giving the CSV separator (comma or semicolon) – this can be read by Excel:

```
sep=,
```

or

```
sep=;
```

Line 2: Title
Line 3: Frequency and Sample Rate
Line 4: Noise – true/false for on/off and the peak
Line 5: column titles for the harmonics:

```
Frequency,Amplitude,Phase
```

followed by the data values.
The data consists of rows with the frequency, amplitude and phase as decimals.

Other parameters are not saved as they are likely to be set for a session independently of the individual data files.
For export the CSV separator is chosen as follows:

> If the system language has comma as the decimal separator, then the CSV separator is semicolon.
> Otherwise, the CSV separator is comma.

The thousands and decimal separators are according to the language set on the system. Conversion may be necessary if files written in English are to be read in German, French, etc. or vice-versa.
Example:

```
sep=;
Description;100 Hz with 3 sawtooth harmonics for 1 second.
Frequency;100.0;44000
Noise;false;0.5
Frequency;Amplitude;Phase
1,000000;1,000000;0,000000
2,000000;0,500000;0,000000
3,000000;0,333333;0,000000
```

8 Analysis

8.1 Overview

8.1.1 WAV files

The following data codings are currently supported for reading from WAV files:

1 or 2 channels
8-, 12-, 16-, 24-, 32-, 64-bits per sample
Any sample rate
PCM unsigned 8-bit is always unsigned)
PCM signed (12-, 16-, 24-, 32- and 64[5]-bits per sample (>8 bits are always signed)
PCM floating point 32-bit and 64-bit
A-Law, μ-Law 8-bit (these are dynamic range compression methods for telephony)
Little endian (No big-endian files have been found so far.)

There are other programs which can convert other file formats to a usable WAV format. Floating point input file formats are converted to 24-bit signed for further processing.

8.1.2 Round Trip

As stated in the synthesis section 7.1, we use sine waves rather than cosine waves throughout. This means that the phase spectrum has an alteration of +90° to give sine waves in the result. When passed back to the synthesis in a round trip they then correspond to the original wave.

8.1.3 Window-Functions

Some formulae for the window-functions give results with varying peaks. All these window-functions have been normalised to have a peak of 1.0.

8.1.4 Analysis Window

The analysis window appears when a wav file is opened or is sent from the synthesis and has four sections,

- The **Average and Decay** panel shows the input wave and performs the average calculation between the marks and the interpolated decay rate.
- The **Input Wave** showing the wave read from a file or passed from the synthesis, and is used to set up the Fourier Transform.
- The **Fourier Transform Frequency Spectrum** showing the result of the Fourier Transform. The y-axis is calibrated with the amplitude values and can be set to linear or logarithmic. The x-axis is the relative frequency of the spectral components (bins), with a second x-axis for the corresponding frequencies in Hz (derived from the sampling rate of the input file). The red vertical lines show the component frequencies with their amplitudes.
- The **Fourier Transform Phase Spectrum** showing the corresponding phases on the same relative frequency x-axis as the frequency spectrum. The y-axis is calibrated in radians +/- π.

Below the frequency and phase diagrams are:

- a scroll bar and magnification slider. These operate on the frequency axis of both the frequency and the phase spectra.
- A set of buttons to set the magnification and scroll to commonly used settings.

If a Short Time Fourier Transform is selected, a panel of buttons for controlling the animated display of frames is below the spectrum.

[5] 64-bit PCM files are reduced to 32-bit to keep the values manageable.

Each section can be hidden by unchecking the corresponding "Show" button in the Analysis Parameters window to give more vertical space to the other sections.
The file name of the file read in is shown in the title bar.
The status lines of the average and input panels show the current magnification of the time axis in samples per pixel, the size of the window selected for analysis in samples and in milliseconds, and (for the input wave) the effective fundamental frequency given by the selected window assuming the window covers one cycle; the absolute frequency can be seen from the middle Hz x-axis of the spectrum.

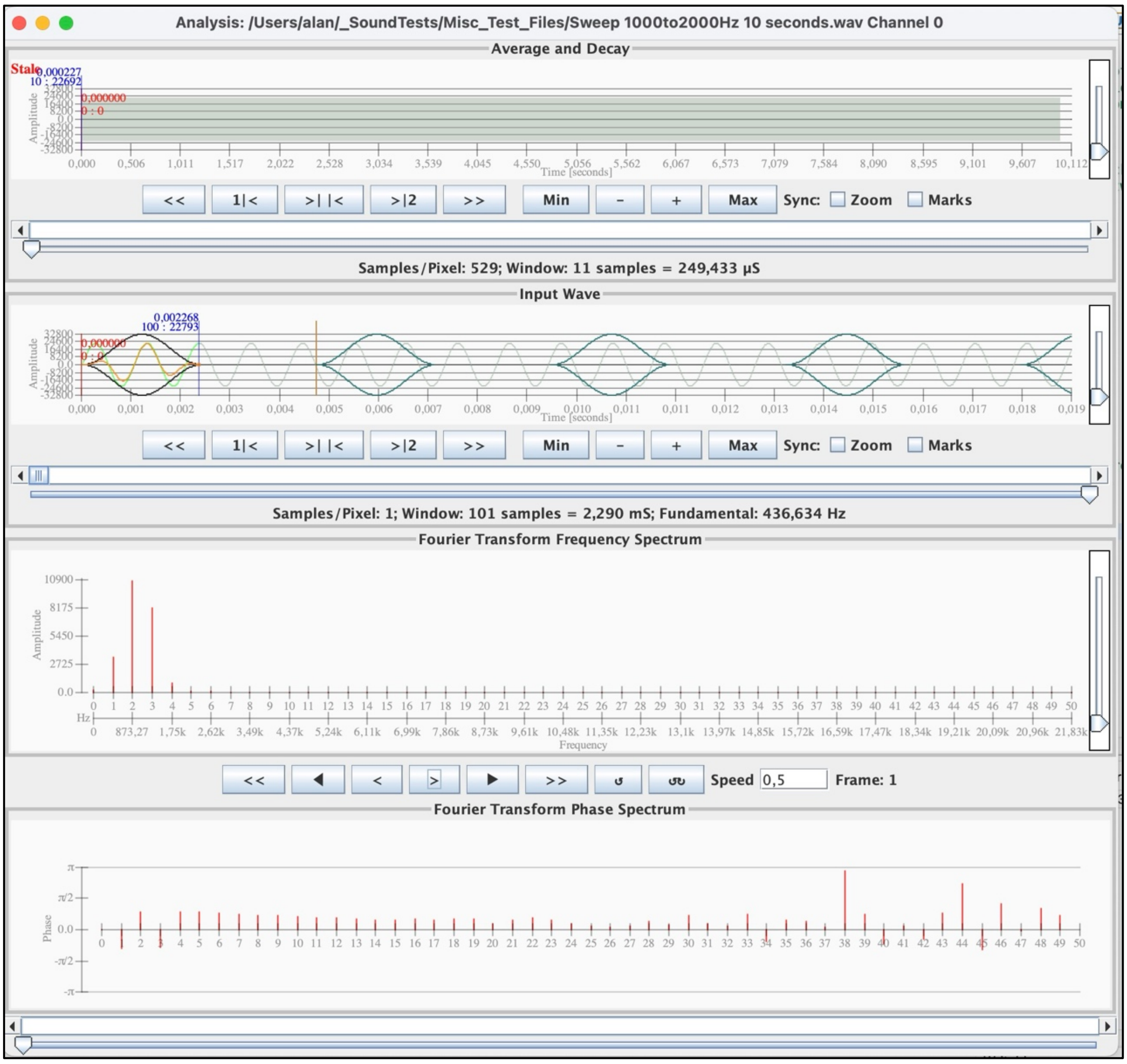


Figure 8-1 Overview of Analysis Window

8.2 Input Wave Display

To start the analysis of a wave file, click New and select a file. The file is read and displayed in a new window. The Input Panel shows the complete wave file that has been read in. The window title gives the path of the file, or "from Synthesis" if the data was sent from the synthesis function (see 7.3.2 "Send to Analysis").

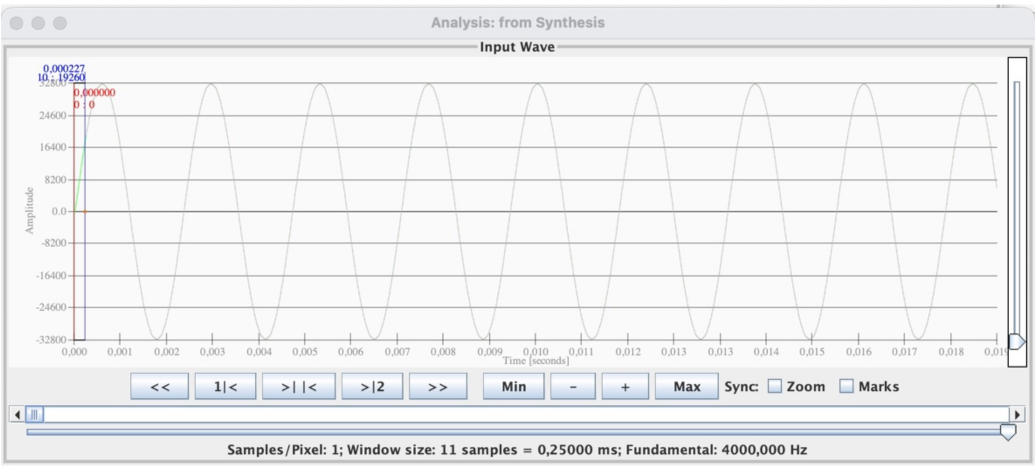


Figure 8-2 Analysis Input Diagram

The Y-axis is calibrated with the absolute values of the samples.
The X-axis shows the time in seconds.

8.2.1 Markers

There are two coloured markers which define the window to be analysed, Mark1 in red and Mark2 in blue. The wave within the marks is coloured green, outside the marks it is grey.
They can be moved as follows:

- Right click with the mouse at a point in the wave and select Mark1 or Mark2 to place it at that point.
- Drag one of the marks left or right with the mouse.
- Dragging with the shift-key pressed moves both marks together.
- A mark can be selected, either by click on the line or clicking on the label, and moved with the left or right arrow keys on the keyboard. Each key-press moves by one sample, so at magnifications below the maximum (1 sample per pixel), the mark might not visibly move. The shift-key acts as above, moving both marks together.

The positions of the markers are restricted such that:

– Mark1 must always be before (to the left of) Mark2.
– Neither marker can be moved beyond the start or end of the wave file. Note that when moving both marks with the shift-key, the movement will be blocked if one mark, which may not be visible, reaches the start or end of the file.
– The marks cannot be moved while an analysis is running.

The label above each marker shows the time in seconds to six decimal places (micro-seconds) and the sample number starting from 0 below that, followed by a colon and the amplitude value of the wave at that point.
Note: selecting the mark by clicking on the line can cause it to move inadvertently. For precise positioning, click on the label to select it and then use the left and right arrow keys.
Note that an excessive number of samples between the marks can cause the program to run out of memory. It will indicate this with an error message before exiting.

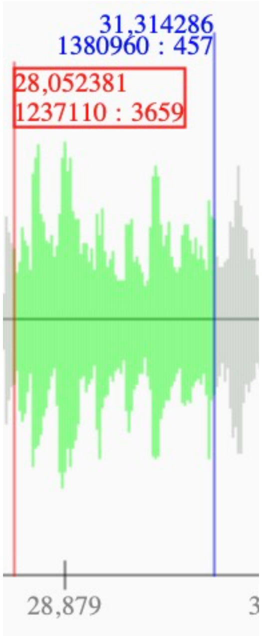


Figure 8-3 Marks for Analysis Window

For the best analysis results both markers should be placed at points with the same amplitude value for one cycle of the wave. This is often not possible as there may not be an actual sample at that point. This is best done at maximum zoom.
Note: the segment of the wave analysed (the analysis window) excludes the sample at Mark2. If Mark1 is at sample 0 and Mark2 is at sample 100, the window is 100 samples long from sample 0 to sample 99. The window size is shown in the status bar at the bottom of the analysis window and in the results summary in the Analysis tab of the main window.

8.2.2 Zero-Padding Marker

If zero-padding is added to the window. A third marker above the x-axis and a thick line extending the final windowed wave in orange shows its position. This may be beyond the end of the file, in which case it is not shown.

8.2.3 Cursor

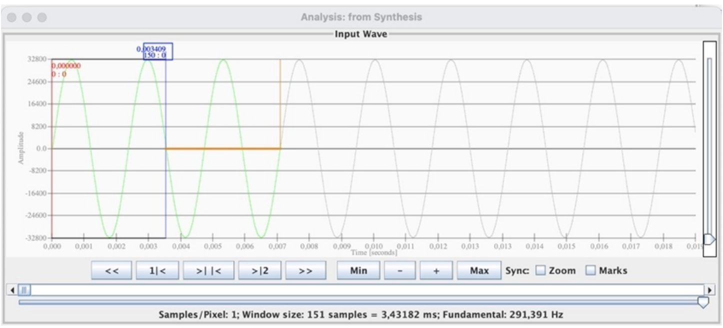


Figure 8-4 Cursor in Analysis Window

When playing the sound, a yellow cursor shows the current position in the input wave. When showing the frames of a STFT, the cursor is at the start of the frame currently displayed.

8.2.4 Analysis Window and Windowed Wave

The shape of the analysis window is shown in black, and the final windowed wave is shown in brown. These can be turned on or off with the checkboxes (section 8.6.2).

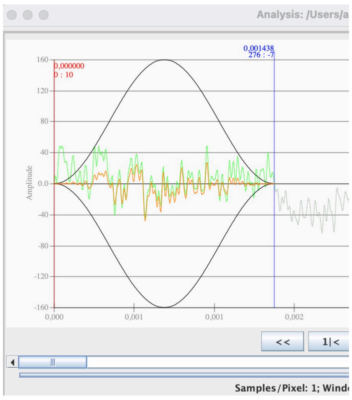


Figure 8-5 Analysis Window and Windowed Wave

8.2.5 Multiple Windows

Multiple windows can be analysed, either to obtain the averaged spectrum of multiple frames or for a Short Time Fourier Transform. Averaging can be used for noise reduction and the frames are normally chosen to overlap. The percentage overlap is given in the results pane (see section 8.2.9) and is negative if the frames have a gap between them.

The number of frames and the number of samples between frames (step) can be set by the user. The additional windows are shown with feinter lines in the graphic.

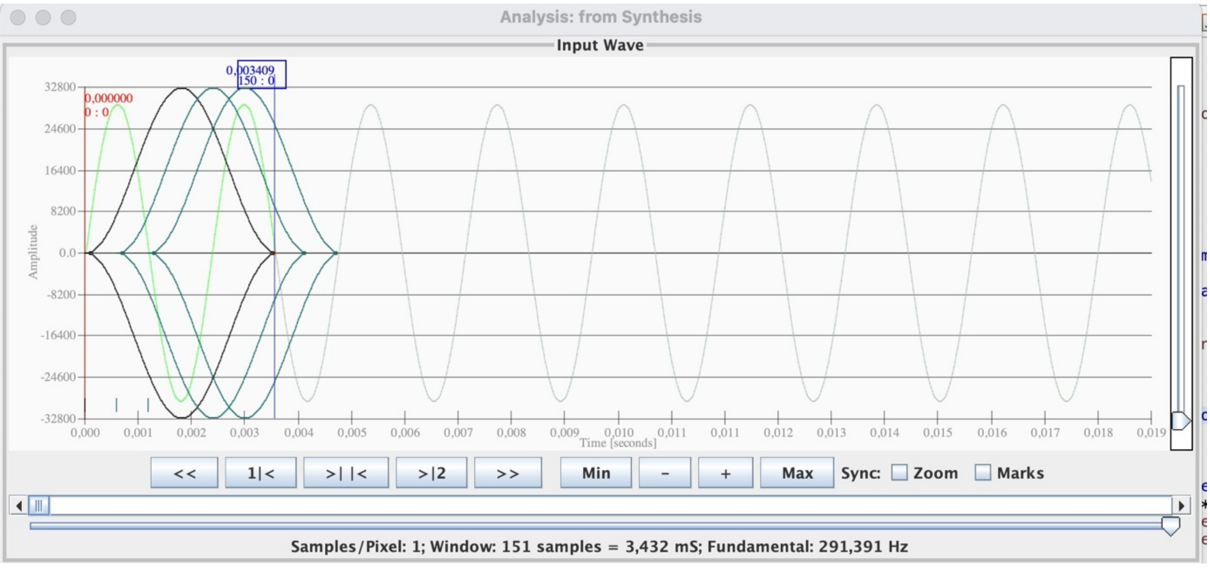


Figure 8-6 Overlapping Windows

Additional small tick marks are shown above the time axis for the start of each window. This is particularly useful for the rectangular window, which is otherwise not easily visible.

8.2.6 Time Position and Zoom Buttons

Below the wave there are buttons for rapid positioning and zooming of the wave plot:

Figure 8-7 Position and Zoom Buttons

<< Go to start of the wave
1|< Go to Mark 1
>| |< Set the zoom and pan to show the entire window. This will put Mark1 on the left of the plot and set the magnification such that Mark2 is on the graph. Since the magnification is in integer values of samples per pixel, it will use less than the whole width if the next magnification level would put the end of the window outside the plot. Note that resizing the window does not maintain the entire window in the view.
>|**2** Go to Mark 2
>> Go to end of the wave
Min Set minimum zoom to show the entire file in the width of the plot
- Zoom out by one step of the slider
\+ Zoom in by one step of the slider
Max Set maximum zoom, one sample per pixel

The Sync checkboxes are used to synchronise the zoom/pan and the marks between the Average and Decay panel and the Input Wave panel – see section 8.4.

8.2.7 Time Axis Zoom and Scroll

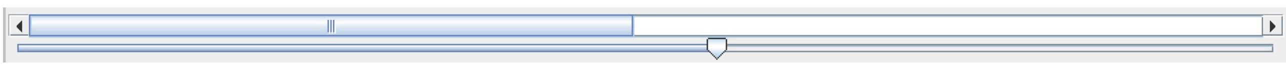


Figure 8-8 Zoom and Scroll Sliders

At the bottom of this section there is a slider for the magnification, and above that, a scroll bar.

The zoom slider has maximum zoom at the right at one sample per pixel and minimum zoom on the left which shows the entire file in the width of the screen. The slider moves from the maximum in steps of 1 from 1-5 and thereafter linearly to the value for minimum zoom. The current setting is shown in the status bar. Note that since the samples per pixel will not usually divide the total number of samples exactly, the last pixel may contain more or less samples than the given samples per pixel.

8.2.8 Amplitude Axes and Scaling

The wave from the input file (top) is calibrated vertically according to the amplitude similarly to the waves in the synthesis window. The y-axis calibration is in the sampling units, e.g. -127 to +127. This can be magnified with the vertical slider on the right. The horizontal axis is the time in seconds (to three decimal places giving milliseconds).
The frequency spectrum (middle) gives the amplitudes on the y-axis, scaled linearly or logarithmically as selected by the radio buttons – see 8.6.5. This can be magnified with the vertical slider on the right. For logarithmic scales the slider raises or lowers the floor of the plot.
The phase spectrum (bottom) gives the phase in radians on the y-axis (π is 3.14) and the same frequencies as the frequency spectrum on the x-axis.

8.2.9 Results Display

The results of the analysis are shown in the analysis display window – see 8.6.6. Any changes to the parameters, e.g., the window -function or position of the marks, are shown on the display. Any change means that the spectrum results are stale (out of date), and this is indicated in the spectrum plot – the plot remains in place for viewing until a new analysis is run.

8.3 Average and Decay Display

8.3.1 Overview

This is similar to the input wave display and is used to calculate the decay rate between the marks; this is based on a running average amplitude of the waveform to avoid false results that would occur if a peak or trough was used as the reference points.
Since most sounds decay in a logarithmic fashion, it is normally more useful to show the decay rate on a logarithmic scale in decibels. However, the logarithm of the wave values at the zero crossings cannot be calculated, as log(0) is minus infinity. The wave is therefore averaged from the absolute amplitudes of the samples. The number of samples either side of the current sample that are taken into the average of that sample is selected by the user and should normally be at least the number of samples in one cycle of the wave. This is called the span. To assist in selecting this the span is shown as horizontal lines near the top of the mark lines (it may be invisible at lower horizontal magnifications).
The decay marks, as in the input wave display, show the time within the file and the sample number and sample value at the top of the mark. Additionally, the sample number and value of the average are shown at the bottom of the marks. Mark 2, the end of the averaging, also shows the difference from the average at mark 1 and the resulting decay rate, in decibels in dB/second for logarithmic scales. Note that this may be different from the result shown in the analysis window, which is derived from the linear interpolation. See Figure 8-10.

The average itself is shown on the wave in dark grey and a linear interpolation is shown in purple. The vertical zoom works differently in logarithmic mode – it effectively moves the floor upwards so that more detail is shown in the smaller dB values.

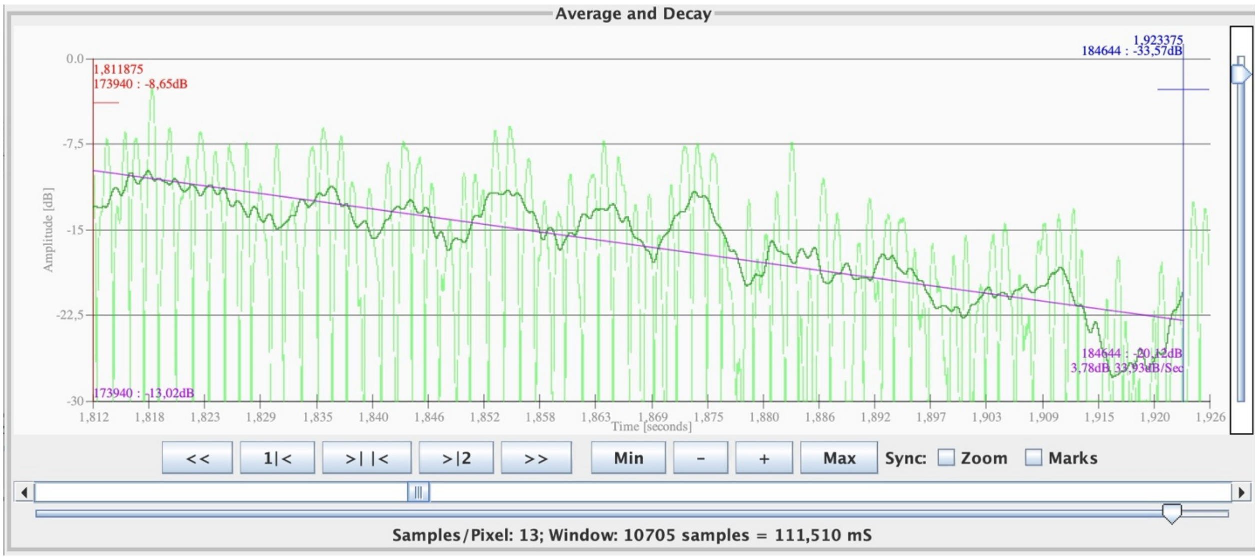


Figure 8-9 Average and Decay Display

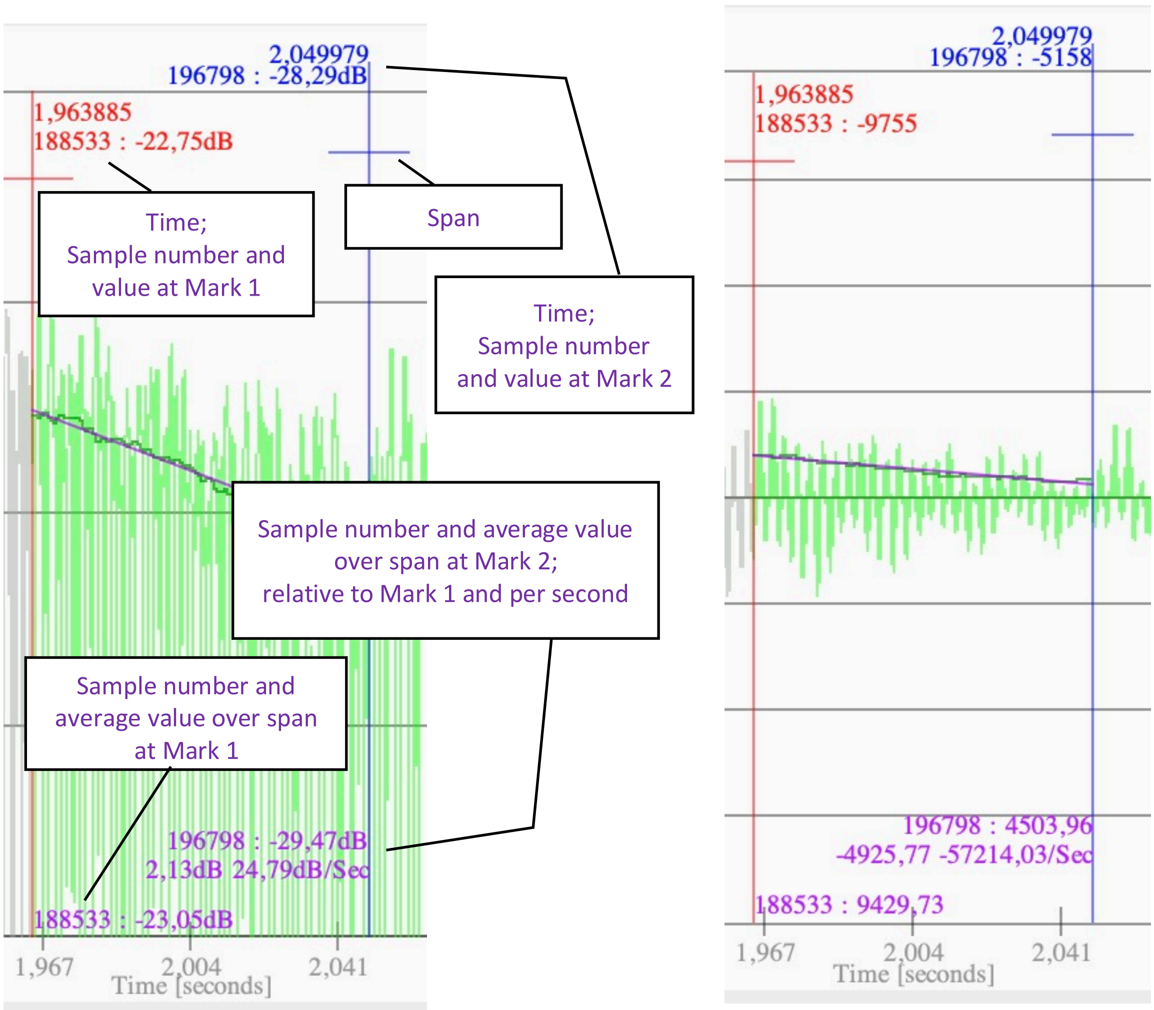


Figure 8-10 Details of Decay Values at Marks - logarithmic (left) and absolute (right)

8.4 Synchronisation

The Average/Decay panel and the Input Wave panel can be synchronised.
Clicking the Sync Zoom button on one panel sets the zoom and pan of the other panel to match.
Clicking the Sync Marks button on one panel sets the marks of the other panel to match.
When Sync Zoom is ticked, moving the zoom or pan of one panel moves the other panel as well.
When Sync Marks is ticked, moving the marks in one panel moves them in the other panel as well.
The buttons act in the same way.
Some use cases are:

Analyse a Fourier transform and decay over the same sample window – check both Zoom and Marks.
Display the input wave on linear and log scales – check Zoom.
Analyse the Fourier transform of part of the decay curve – check Zoom but not Marks.

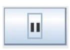
8.5 Spectrum Display

8.5.1 Overview

The spectrum shows a vertical line for each frequency bin. The x-axis is calibrated in bins, and the corresponding frequency in Hz on a second axis below this. The graph can be zoomed and panned together with the controls, which, if the phase plot is show, appear below that (they both zoom and pan together).
The y-axis calibration depends on the chosen scale, linear, logarithmic full scale or logarithmic relative to the maximum. The log scales are in decibels.
The y-axis magnification is controlled by the vertical slider on the right of the plot.

8.5.2 STFT Animation Controls

When a Short Time Fourier Transform has been done, the animation controls appear below the spectrum plot. These control how the frames are viewed and are as follows:

<< Go to first frame (number 0)
◀ Play backwards. When clicked the frames are played from the previous position. The button shows a pause sign to pause the animation.
< Step backwards by one frame
> Step forwards by one frame
▶ Play forwards
>> Got to the last frame
↺ Loop: play repeatedly from beginning to end
↺↻ Bounce: play forwards then backwards repeatedly
Speed Sets the playback speed. 1.0 is real-time, i.e. if the frames cover one second it will play in one second. 0.5 will play half as fast, 2.0 will play twice as fast, etc.
Frame: indicates the frame being currently shown. Note: this starts at 0 and ends at 1 less than the number of frames.
 The buttons that play the animation show a pause sign while running, and clicking this will pause the animation. The Go to and Step buttons will stop any running animation.

The best way to save the animation is to make a video of the screen while it is running (e.g. on Mac OS with QuickTime Player).

8.6 Analysis Parameters

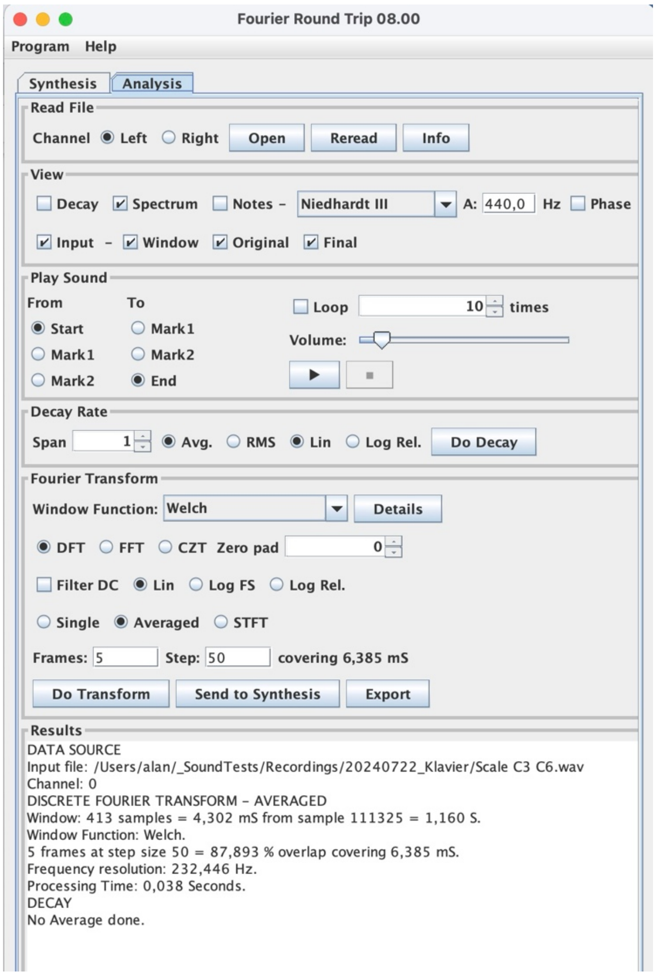


Figure 8-11 Analysis Parameters

8.6.1 Read File Section

This section controls the reading of wav files.
The channel to read can be selected before reading a new file or rereading the same file with a different channel.

Left
: Read the left channel (channel 0).

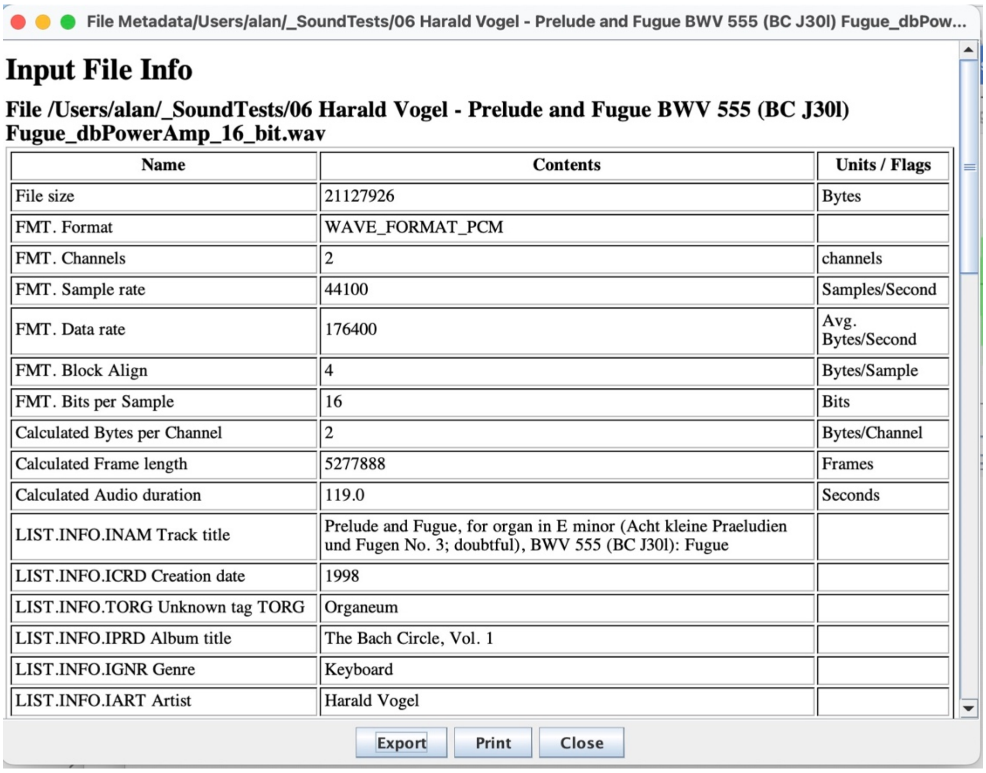
Right
: Read the right channel (channel 1).

Open
: This opens a file chooser to select a new input file. For large files a progress monitor will display with the option to cancel the read.
If the read is cancelled or fails, the previous file remains open. After a successful read the analysis window is closed and re-created with the new file.

Reread
: This reads the open file again, usually used after changing the channel.

Info

Brings up a window showing the information about the wave file being analysed. This shows the data derived from the formatting of the wav file and other meta-data contained in the file. Entries are preceded by the internal tag name from which they are taken.
It attempts to show all embedded information but only shows the contents of those that are relevant; others have “skipped” in the contents.
There are buttons to export the content to an html file, print it and close the window.
Note that graphics will not be shown, as the built-in html interpreter does not support this, but exporting and displaying in the standard browser should show them correctly.
See [ID3] for details of the ID3 tags.


File Metadata/Users/alan/_SoundTests/06 Harald Vogel - Prelude and Fugue BWV 555 (BC J30l) Fugue_dbPow...

Input File Info

File /Users/alan/_SoundTests/06 Harald Vogel - Prelude and Fugue BWV 555 (BC J30l) Fugue_dbPowerAmp_16_bit.wav

Name	Contents	Units / Flags
File size	21127926	Bytes
FMT. Format	WAVE_FORMAT_PCM	
FMT. Channels	2	channels
FMT. Sample rate	44100	Samples/Second
FMT. Data rate	176400	Avg. Bytes/Second
FMT. Block Align	4	Bytes/Sample
FMT. Bits per Sample	16	Bits
Calculated Bytes per Channel	2	Bytes/Channel
Calculated Frame length	5277888	Frames
Calculated Audio duration	119.0	Seconds
LIST.INFO.INAM Track title	Prelude and Fugue, for organ in E minor (Acht kleine Praeludien und Fugen No. 3; doubtful), BWV 555 (BC J30l): Fugue	
LIST.INFO.ICRD Creation date	1998	
LIST.INFO.TORG Unknown tag TORG	Organeum	
LIST.INFO.IPRD Album title	The Bach Circle, Vol. 1	
LIST.INFO.IGNR Genre	Keyboard	
LIST.INFO.IART Artist	Harald Vogel	

Figure 8-12 File Info Window

8.6.2 View Section

This section controls which panels are shown in the main window.
Decay: the average and decay panel
Spectrum: the spectrum plot can be shown or omitted.
Phase: the phase plot can be shown or omitted. Omitting it gives extra vertical space to the wave and spectrum plots.
Notes: the positions of musical notes can be shown as vertical lines in the spectrum. The temperament can be selected from the drop-down list and the frqeuency of A can be set in Hz.
Input: the Input Wave panel with details:

Window: the outline of the window -function is shown in black.
Original: the original wave is shown in green.
Final: the final wave after application of the window -function and the addition of zero padding is shown in orange.

8.6.3 Play Sound Section

This section makes it possible to listen to all or parts of the sound file on the computer audio system in various ways.

From and To

The radio buttons in the From column (**Start**, **Mark1**, **Mark2**) determine from where the playback will start.

The radio buttons in the To column (**Mark1**, **Mark2**, **End**) determine where the playback will end. The combinations are limited to those which are logically possible.

Loop times

If this box is checked the selected part of the sound will be played repeatedly for the number of times given in the field. The duration of the window size between the marks is shown in the status bar at the bottom of the screen window, and the actual playing time of the loop is given by multiplying this by the number of loops entered.

Volume

Set the desired playback volume.

Note: the volume is also affected by the volume setting on the computer itself and the connected amplifier and speakers.

Note: some audio formats do not support a volume control. In this case the slider is disabled and greyed out, and the wave plays with the equivalent of full volume on the slider – reduce the computer volume!

Play►

The play button starts playing all or part of the sound as selected above.

Stop■

The stop button stops the currently playing sound.

8.6.4 Decay Rate Section

A moving average of the input wave between the two marks is calculated and displayed.

The average is marked as “stale” when any of the following parameters are changed.

Span

Set the number of samples over which to average. Each averaged sample is derived from the samples on either side so that the averaged wave does not have an offset on the time axis. The span must therefore be odd. The marks must be at least half the span from the ends of the file.

Avg. or RMS

Select whether the average is taken with the arithmetic mean or the Root Mean Square.

Lin

Check to use a linear scale with the sample values.

Log Rel.

Check to use a logarithmic scale in decibels relative to the maximum absolute value of the input.

Do Decay Button

Click the button to perform the decay rate calculation. A progress monitor is shown. For log scaling it shows the progress of the averaging and then the progress of the log calculation.

8.6.5 Fourier Transform – Analysis Window Section

This section determines how the selected segment of the wave is to be modified before analysis.

Window-Function

The drop-down list selects the window-function applied to the sample window defined by the marks. Some functions have an extra parameter and a field for this will appear beside the function if required. See Appendix 12 for details.

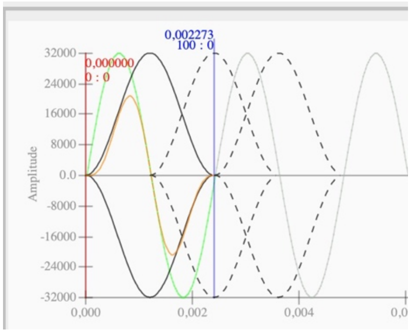


Figure 8-13 Example of a Hamming Window with 3 frames

Details

The Details button brings up a window with a plot of the window-function, its Fourier transform and some of its parameters – see section 8.7.

DFT, FFT and CZT

Select whether a Discrete Fourier Transform or a Fast Fourier Transform, or a Chirp Z-Transform is to be used (see section 3.13).

Note: for FFT the window size must be a power of 2. If it is not, the effective window is increased and filled with zeros at the start and end.

Zero pad

Enter the number of zero samples to add to the end of the sample window. This is shown on the wave plot by a third light orange mark (see 8.2.2).

CZT Parameters (only visible when CZT is selected)

f_0

Enter the starting frequency of the CZT.

f_n

Enter the end frequency of the CZT

Δf

Enter the frequency increment for the CZT

The resulting number of steps is shown.

Filter DC

If this is checked, the DC (direct current or constant bias) will be filtered out of the window before doing the Fourier Transform. The average of the wave over the window is subtracted from all the samples within the window.

Lin, Log FS, Log Rel.

These radio buttons determine how the resulting spectrum is scaled and how the vertical slider on the right of the spectrum acts.

Lin - Linear scaling. The slider magnifies the spectrum, and higher amplitudes will disappear off the top of the plot.

Log FS – Logarithmic Full Scale scaling in dB relative to the maximum possible value of the sample size. Moving the slider up increases the lower limit shown.

Log Rel. – Logarithmic scaling in dB relative to the highest amplitude in the spectrum.

The following radio buttons select the type of transform to be performed.

Single

When Single is selected a single Fourier Transform of the sample window is performed.

Averaged

When Averaged is selected Fourier Transforms are performed for the window multiple times for the given number of frames separated by the step (see below) and the results are averaged to give the spectrum and phase plots.

Note: the number of frames, the amount of overlap and the shape of the wave can produce unexpected results e.g., a single sine wave with an even number of overlapping rectangular windows with 50% overlap will give a zero result because they cancel each other when averaged.

STFT

When STFT (Short Time Fourier Transform) is selected, Fourier Transforms (DFT, FFT or CZT) are performed for the window multiple times for the given number of frames advanced by the step and the resulting spectra are displayed in an animation. The phase spectrum is shown for the first frame only.

STFT Parameters (only visible when Averaged or STFT is selected)

Frames

Select the number of frames to be transformed for Averaged or STFT. The frames are shown with their window-function in the Input Wave graphic. If Single is selected above only one frame is shown.

Step

This determines the distance in samples between the start of one frame and the start of the next. This should be chosen based on the window size (shown in the status bar below the Input Wave), e.g., 20% of the window size to give 80% overlap. The corresponding percent overlap will be shown in the results (see 8.6.6).

covering

The effective time covered by the combination of Frames and Step is shown so that it can be related to the Input Wave time scale.

Note: if the frames go beyond the end of the file an explanatory error message is shown, and the number of frames is set to the maximum value that will fit. Zero padding or expansion for the Fast Fourier Transform can extend beyond the end of the file.

Do Transform

This starts the Fourier transform analysis as set up with the marks in the Input Wave and the parameters above. The progress is shown in the results section.

A single transform is performed on one processor with one progress bar.

For Averaged and STFT the work is divided among the available hardware processors[6] with a progress bar for each. Each processor is started on a frame and when completed, it starts on the next available frame. Since they work asynchronously the frames will not be allocated to processors in any particular order.

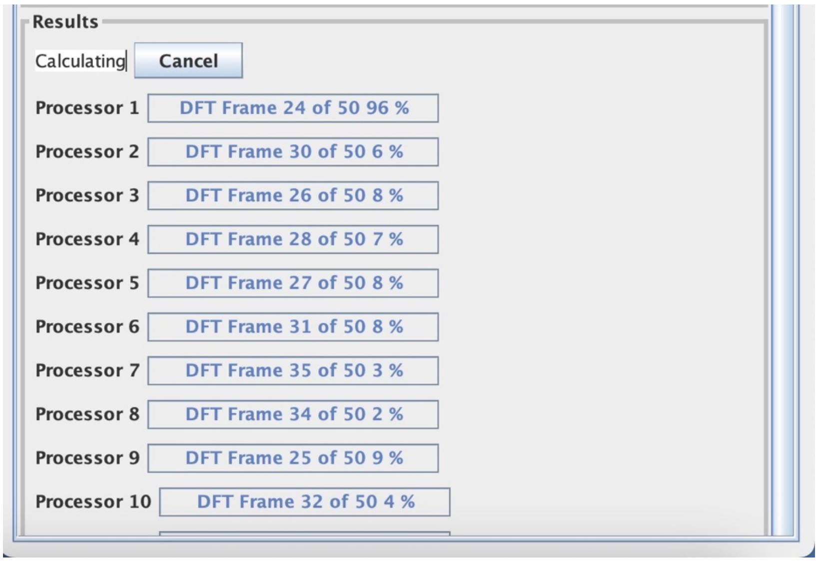


Figure 8-14 Analysis Progress

The transform is performed in up to three parts and the progress bars display the frame on which the processor is working "of" the total number of frames and the % progress of that frame, preceded by the part:
Part 1, the actual transform, shows either "DFT" or "FFT".
Part 2a, deriving the amplitudes and phases from the complex results of part 1, shows "Part 2a".
Part 2b, the logarithmic scaling if "Log FS" or "Log Rel." was selected, show "Part 2b".
Some calculations may happen so fast the no progress bar appears. When all calculations are complete, the result is displayed – see 8.6.6.

Cancel

The cancel button appears in the results window while the transform is running and stops the analysis.

Send to Synthesis

Send the resulting frequency spectrum to the synthesis window where it will be displayed as if it had been synthesised from the harmonics that resulted from the analysis. Note: the spectrum is the result after applying the window function and zero padding.
If no synthesis window is open a new one is created.
For STFT the first frame is sent.

Export

Write the components to a CSV file as a table of frequency, amplitude and phase.

[6] It leaves two processors for other work, so in this example12 of the available 14 processors are used.

8.6.6 Results Summary

At the bottom of the window a summary of the result of the analysis is shown as follows.

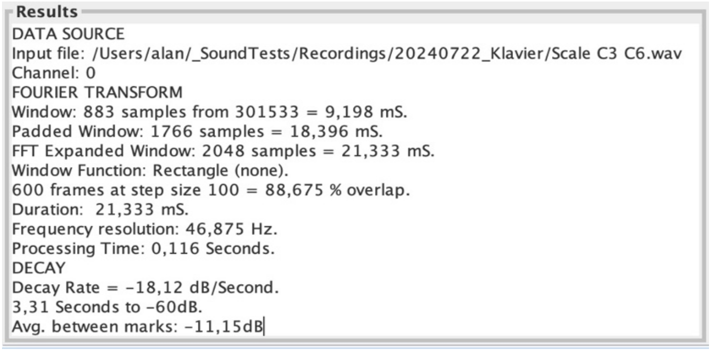


Figure 8-15 Analysis Result

DATA SOURCE:
- Input file path
- Channel (0 = left, 1 = right)

Type of Transform e.g. DISCRETE/FAST/CHIRP Z FOURIER TRANSFORM and SINGLE/AVERAGED/SHORT TERM FT ANIMATION
- The sample window size in samples and the position of the first sample (mark1) and time that the window covers.
- The size of the window including zero padding (if any)
- The size of the window after expansion to a power of 2 for Fast Fourier Transform (if applicable)
- The duration of the window including padding and FFT extension.
- The windowing function and its parameter (if any)
- The number of frames and the step size in samples and the % overlap (if applicable)
- The starting frequency, end frequency and delta of a CZT, and the duration covered
- The frequency resolution, i.e. the difference in frequency between the individual spectral lines.
- The processing time taken to process the analysis.
- This is replaced by “Cancelled” if the analysis was not complete.

DECAY
- The decay rate in decibels per second calculated from the linear interpolation of the averaged wave.
- The number of seconds to reduce the amplitude by 60dB (commonly used as a measure of the reverberation time of rooms such as concert halls). Note: this is omitted if the wave is not decaying.
- The overall average amplitude between the marks (only for log scale).

The results can be copied to the clipboard for use in other programs.

8.7 Fourier Analysis-Window Function Details

This window is brought up by the Details button for the analysis window – see section 8.6.5.
This shows the function name in the title bar and has fields for the display parameters as follows:

X max. bins: The number of frequency bins to show on each side of the x-axis. A high value of this can take a long time to run and may cause the program to run out of memory and crash.
Y max. -dB: The lower limit on the y-axis in dB below which values are cut off.
The window can be resized, but aliasing will occur as the samples of the transform are mapped to the pixels of the display. If the values of “X max. bins” or “Y max -dB” are changed, the display will be recalculated.

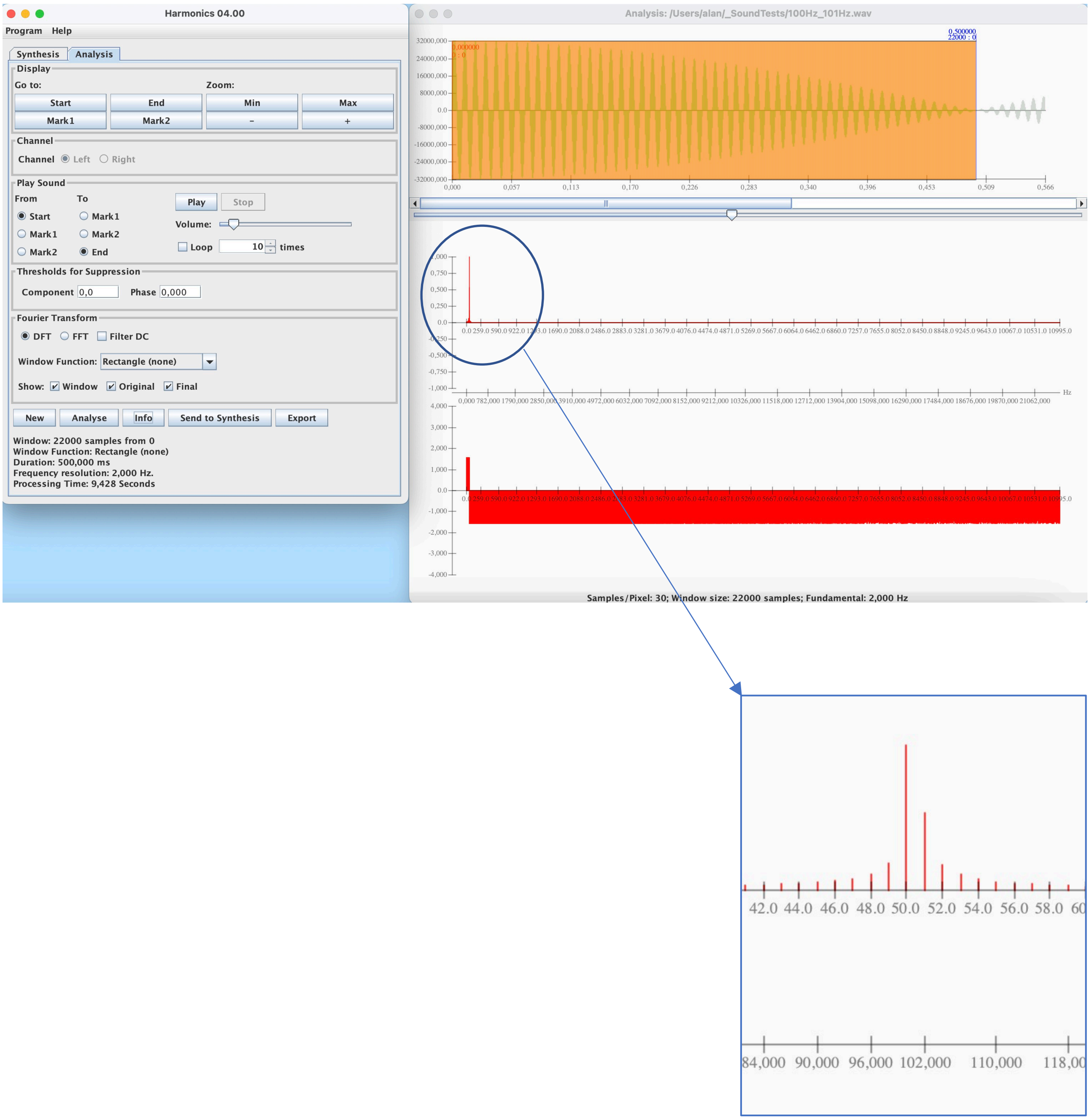
Display windows are not closed automatically so the different functions and settings can be compared.
The parameters shown are as follows:
Window width: The number of samples in the window -function, set at 4 times the plot width in pixels.
Transform width: The number of frequency bins in the transform. This is the next power of 2 above 10 times the window width.
Main Lobe Width [bins]: The width of the main lobe in frequency bins. This is half of the width seen on the graph, i.e., only one side.
NENBW, the Normalised Equivalent Noise Bandwidth: The width of a rectangular bin that would contain the equivalent amount of noise.
Correction Factor: The correction factor in dB resulting from the NENBW which should be applied to the spectrum values.
Max. **Side Lobe Freq.**: The normalised frequency of the highest side lobe (not the first side lobe in all windows).
Max. Side Lobe dB: The value in decibels of the highest side lobe.
Note that last two values are not always correct depending on the choice of "X max. bins" and the aliasing from the window width.

Incidentally, the Rialto Bridge (see Frontispiece 1) has a fall-off rate of about 11.5 dB/Octave, taking the distance between the arches to be an octave!

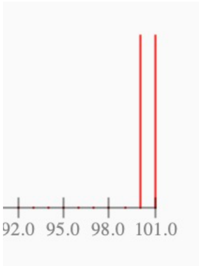
9 Experiments

9.1 Two Adjacent Frequencies

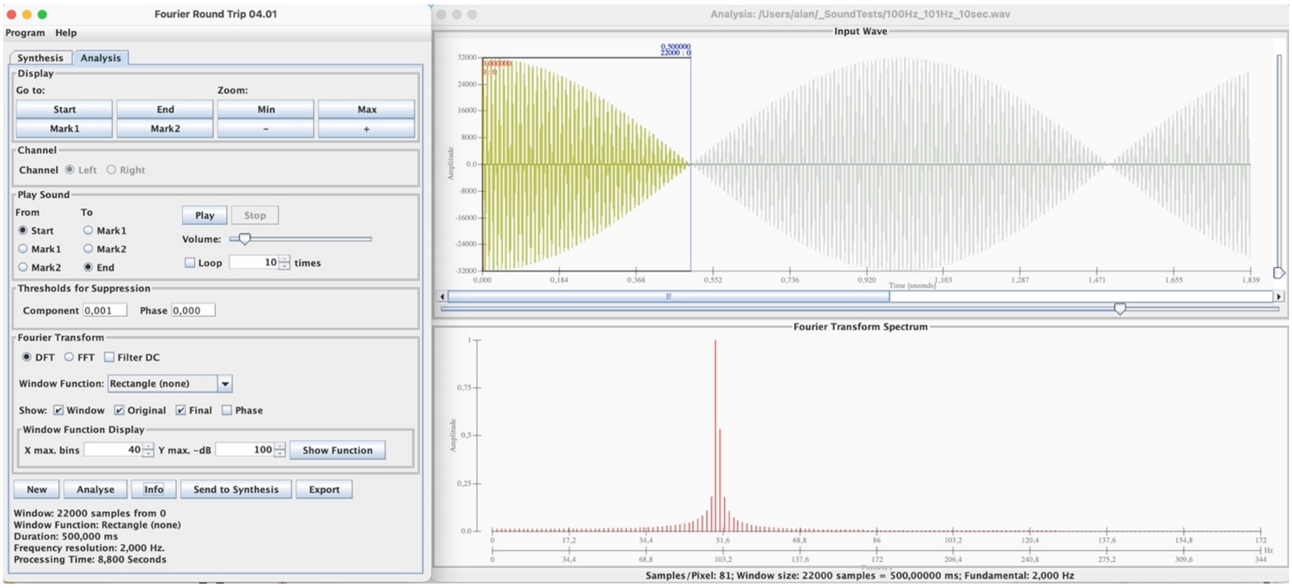
Take a signal with sine waves qt 100Hz and 101Hz.
With a sampling rate of 44000 samples/second, using half a second window (50 cycles of the 100Hz wave) the frequency resolution is 2Hz.
The two spectral lines 1Hz apart cannot be resolved.



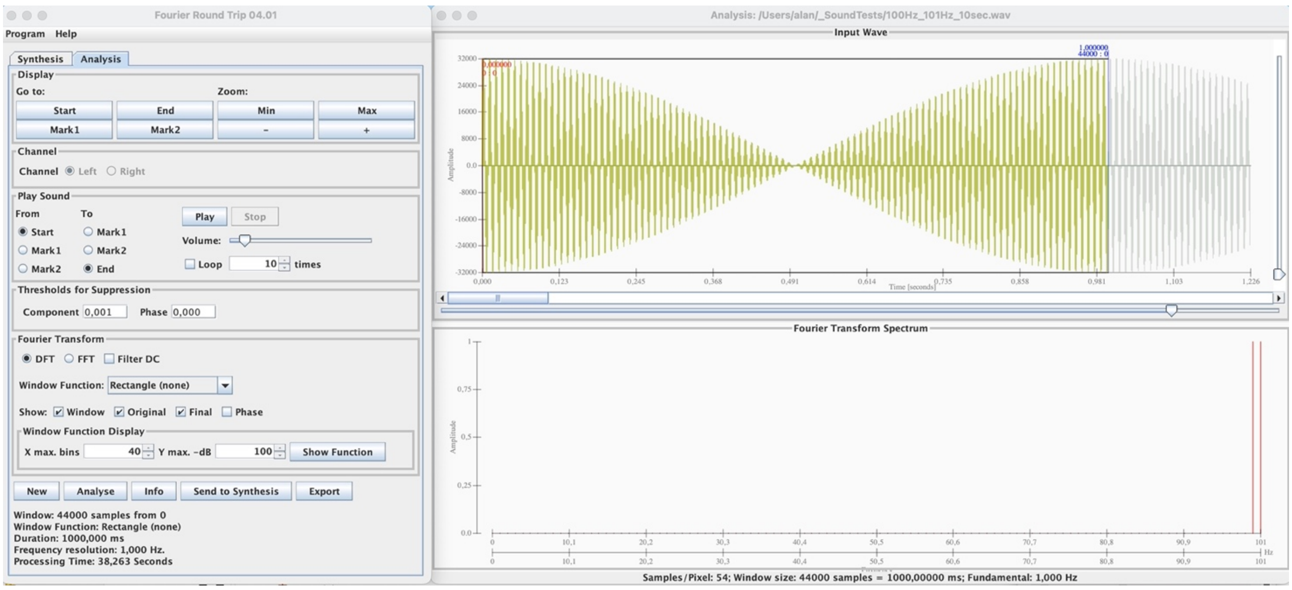
The 100 Hz signal is there because it falls on a frequency division, but the 101Hz signal is distributed over the whole spectrum, mainly in the adjacent frequencies and more above 100Hz than below.
If we now take a higher sample rate of 131072 samples/second (chosen to be a power of 2 for an optimal FFT), with the window over the entire 1 second, we obtain a resolution of 1Hz and can see both frequencies:



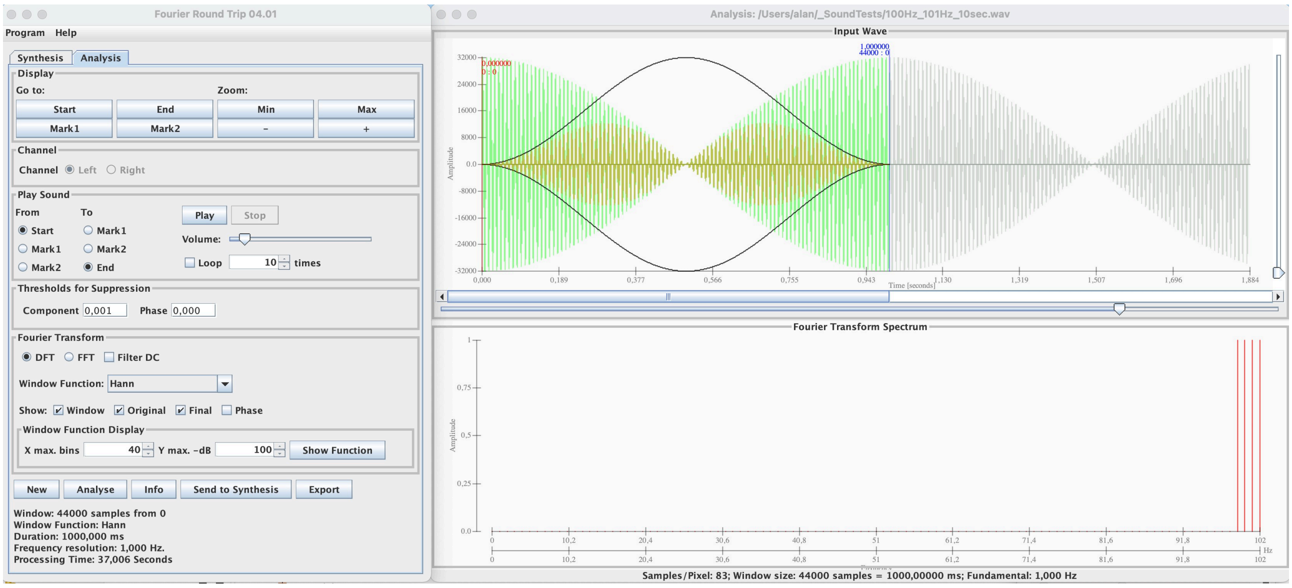
We can also see the effect of window size. Using a 10 second file at the lower sampling rate, taking a window over 0.5 seconds does not fully resolve both frequencies (see [Smith] p. 180ff.),



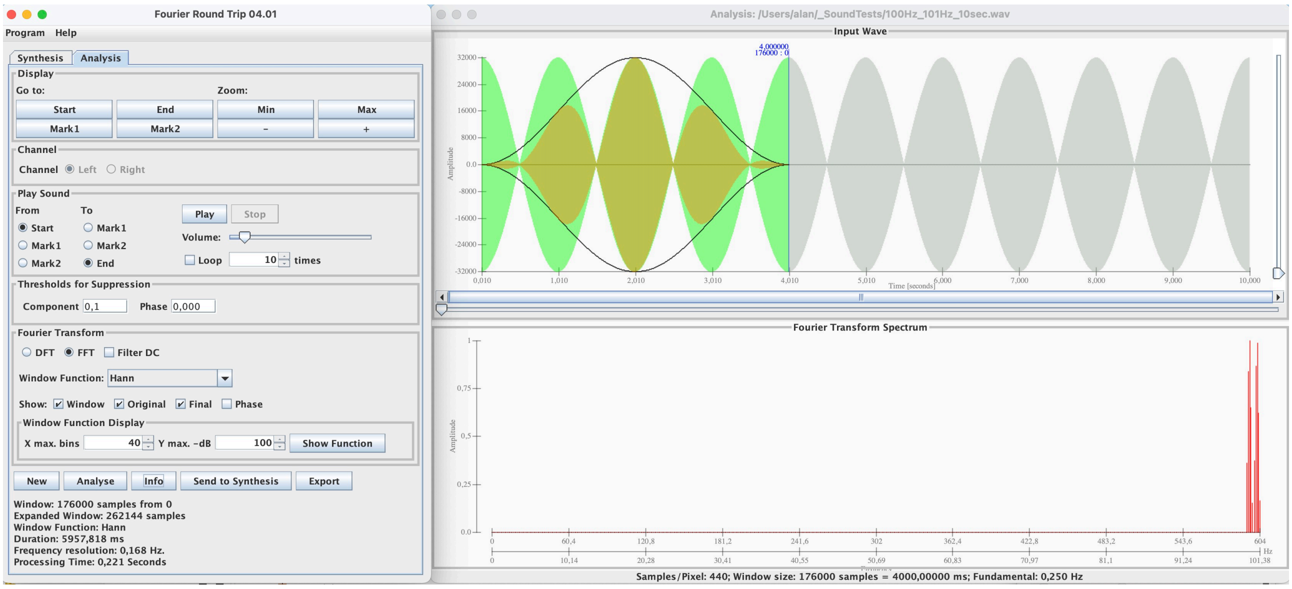
but using a window of over 1 second, to resolve the beat frequency of 1Hz, does resolve the two frequencies at the lower sampling.



A Hann window needs 4 seconds to resolve these frequencies. With 1 second:

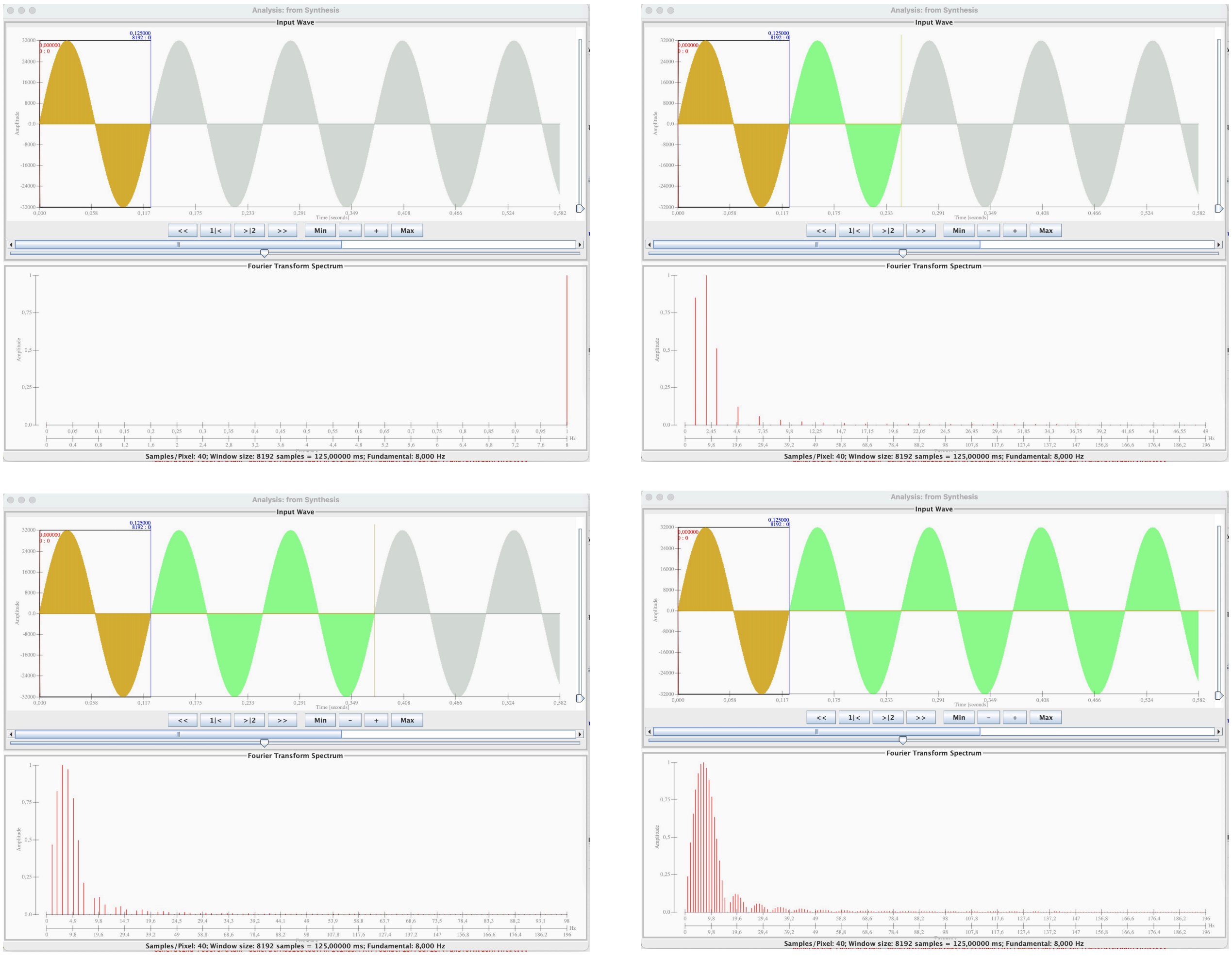


and 4 seconds:



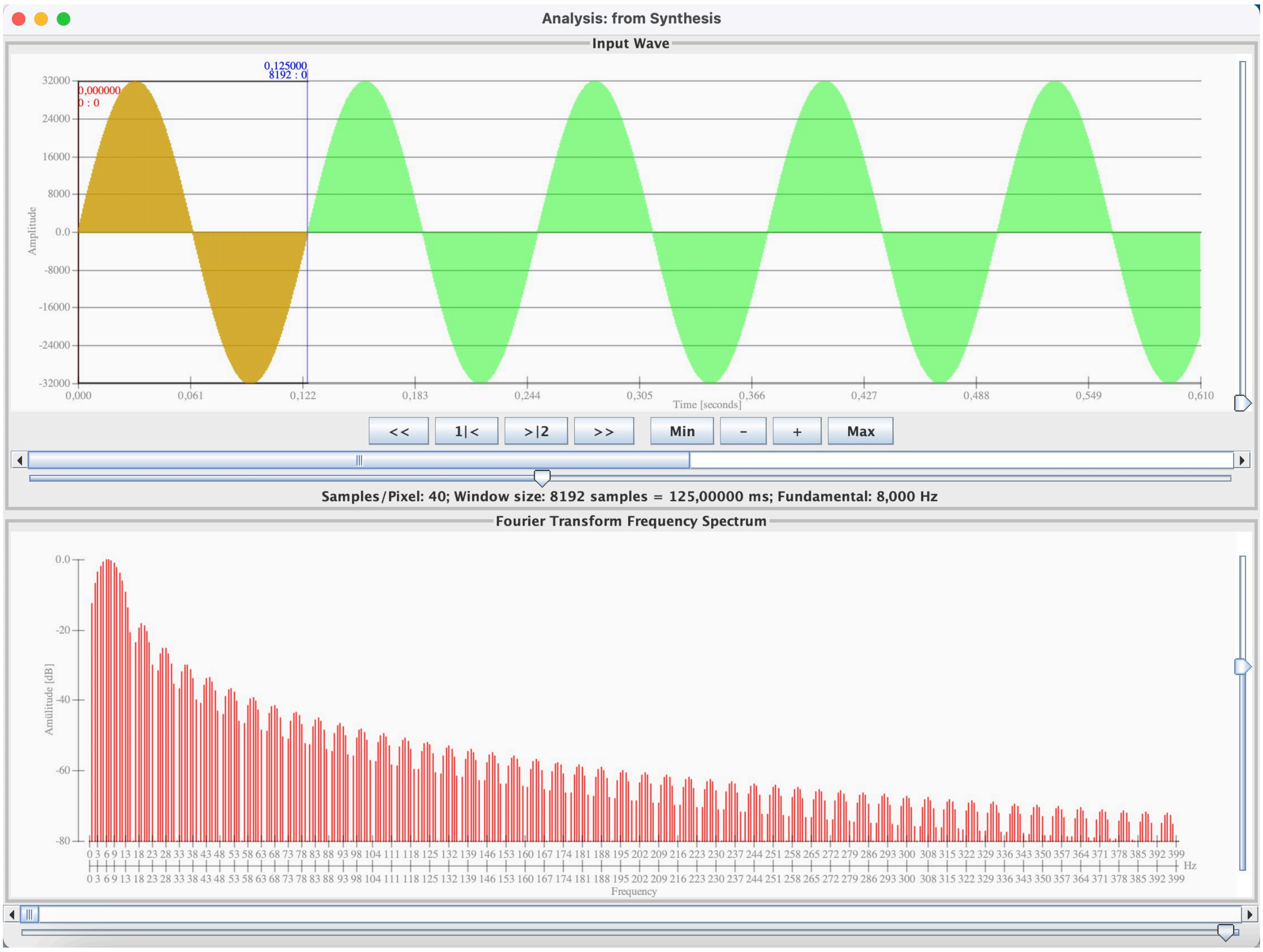
9.2 Zero Padding

Use a sine wave at 65536 samples/s and 8Hz to give optimal FFT.

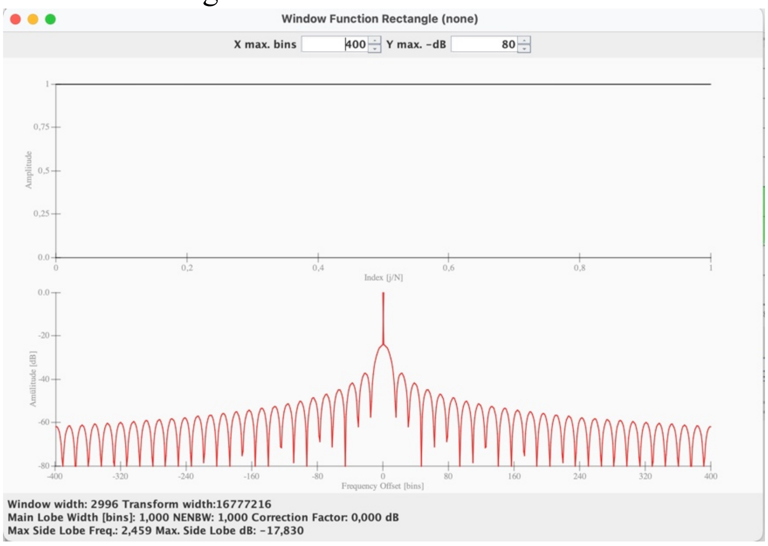


The first plot is with a window of 1 cycle = 8192 samples. Adding 8192, 16384, 32768 zeros shows better approximations to the continuous Fourier transform of the rectangular window

With logarithmic scale:



And the Rectangular window -function:



9.3 Sampling Rate

Using a sampling rate and a frequency that are powers of 2, e.g., 65536 samples/s and 8Hz.

9.4 Overlapping Analysis Windows

An even number of windows with 50% overlap averages to zero.

9.5 Chirp z-Transform

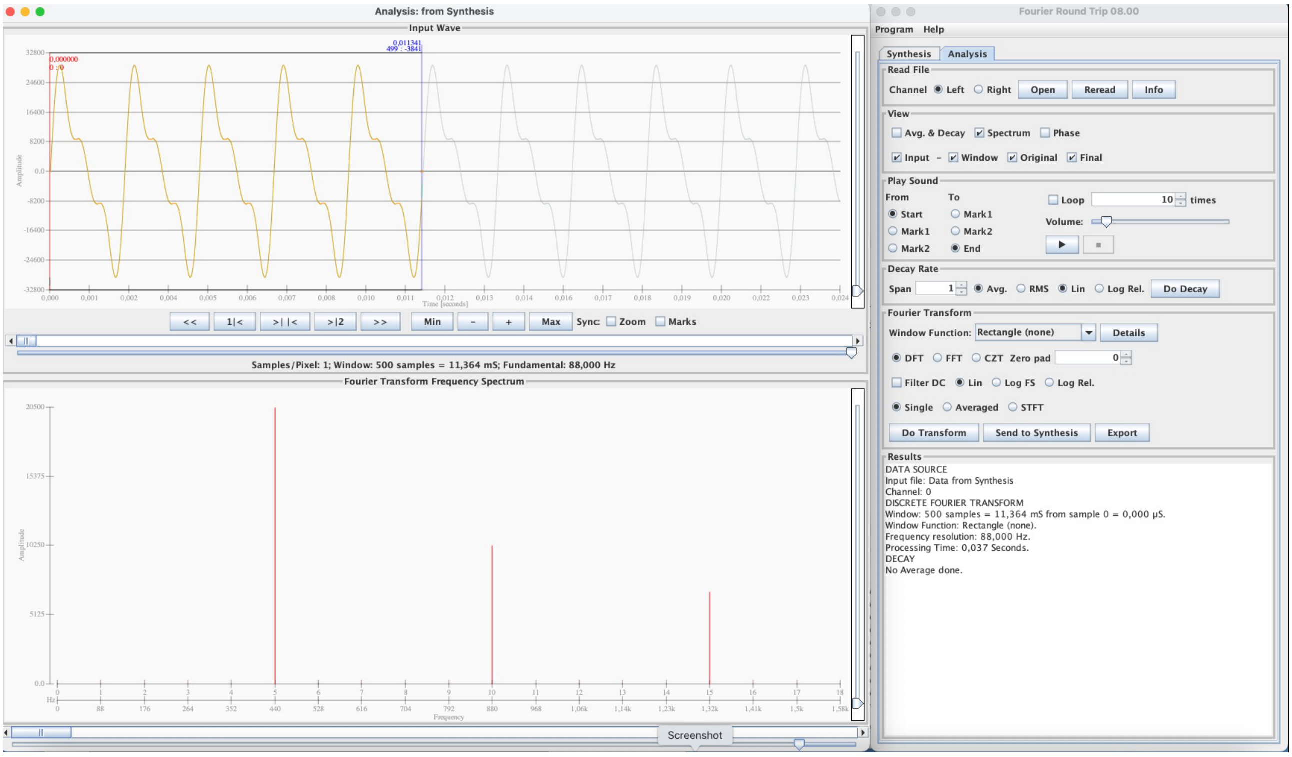


Figure 9-1 DFT of 3 Harmonics (zoomed)

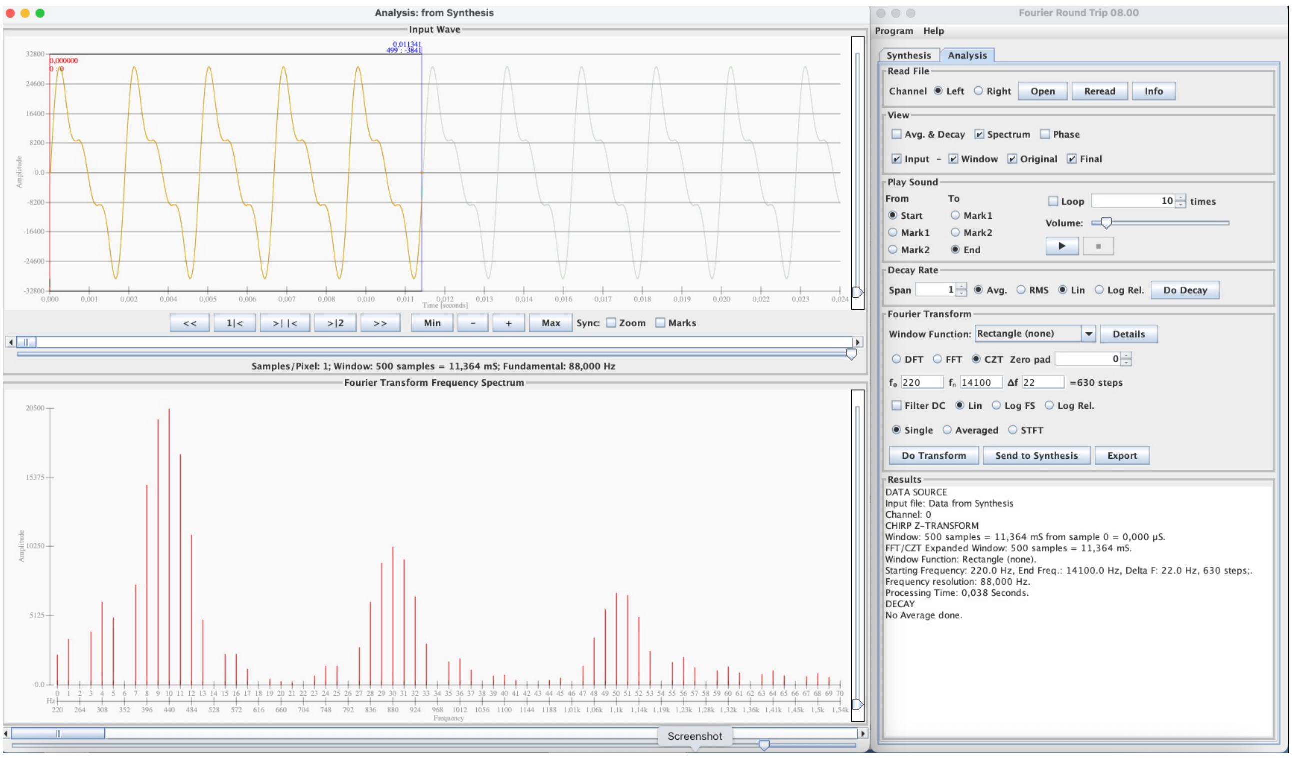


Figure 9-2 CZT of the same 3 Harmonics

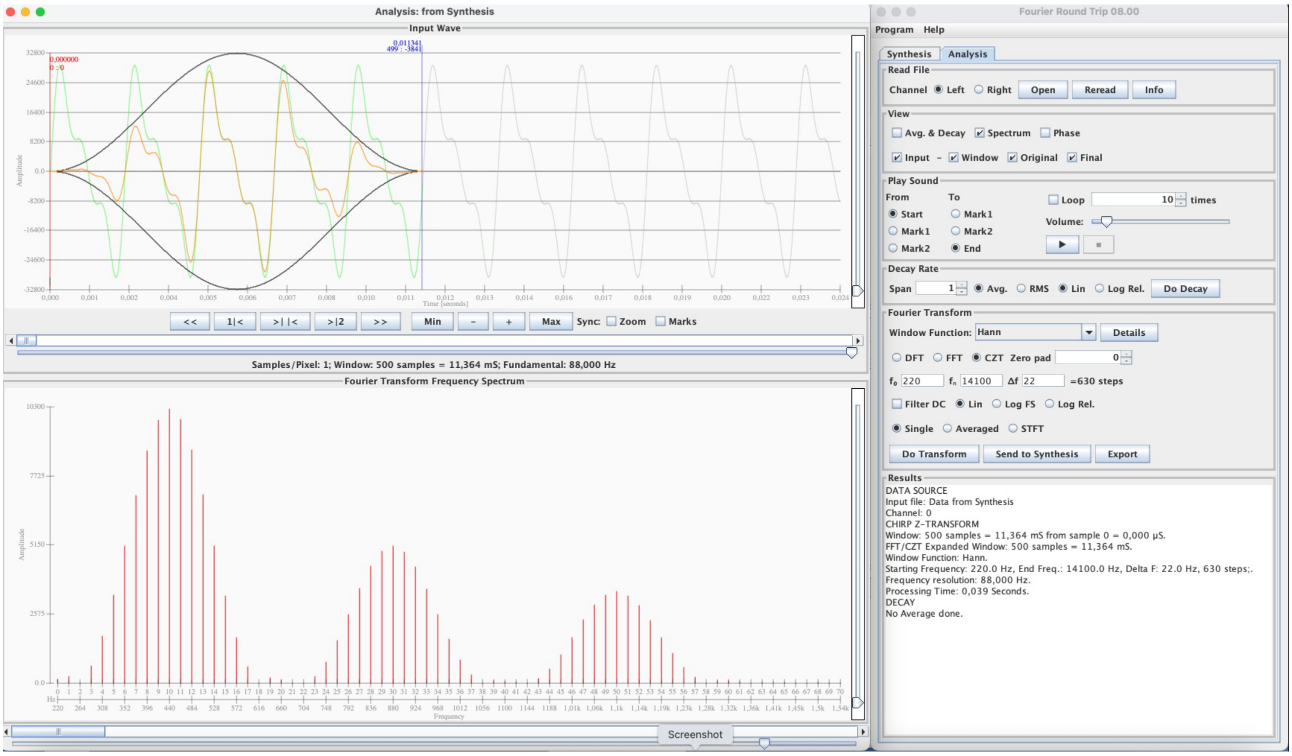


Figure 9-3 The same Harmonics, Chirp-Z Transform with Hann Window

10 Appendix Cardioids

There are several ways of generating a cardioid, the shape of the reference and constructor figures for a wave with one overtone.

10.1 Epicycloid with Equal-sized Circles

https://en.wikipedia.org/wiki/Cardioid (Viewed 25.02.2022)
$x(\theta) = 2a(1\text{-}\cos\theta).\cos\theta$
$y(\theta) = 2a(1\text{-}\cos\theta)\sin\theta$

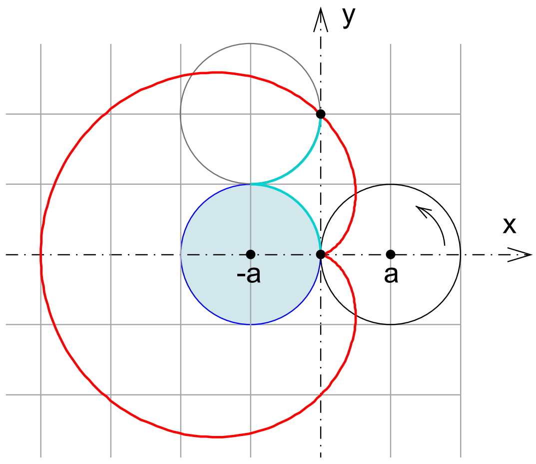
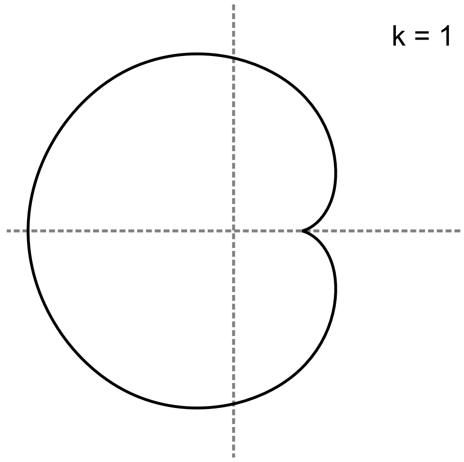


Figure 10-1 Cardioid constructed by rolling a circle round another (wikipedia)

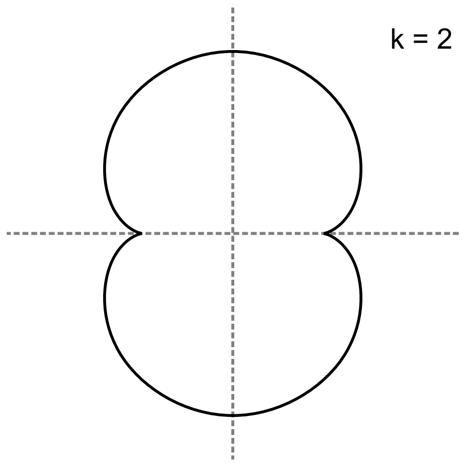
For a=1/2 this gives
$x = \cos(\theta) - \cos^2(\theta)$
$y = \sin(\theta) - \sin(\theta)\cos(\theta)$

10.2 Epicycloid with different-sized Circles

https://en.wikipedia.org/wiki/Epicycloid
Small circle radius r going round a larger circle radius R
$x(\theta) = (R+r)\cos\theta - r\cos(\frac{R+r}{r}\theta)$
$x(\theta) = (R+r)\sin\theta - r\sin(\frac{R+r}{r}\theta)$
Circles of radius R = kr
$x(\theta) = r(k+1)\cos\theta - r\cos((k+1)\theta)$
$x(\theta) = r(k+1)\sin\theta - r\sin((k+1)\theta)$



k = 2 rotated 90° is 2^{nd} overtone - nephroid



11 Appendix Proof that 2nd Harmonic Constructor is Reference Figure Translated

See Figure 3-10 – the orange constructor figure is 0.5 units to the right of the reference figure.

Reference Figure

$$y = \sin(\theta) + \frac{1}{2}\sin(2\theta)$$
$$x = \cos(\theta) + \frac{1}{2}\cos(2\theta)$$

Constructor Figure

$$y = \sin(\theta) + \frac{1}{2}\sin(2\theta)$$

$$x = \frac{y}{\tan(\theta)} = \frac{\sin(\theta) + \frac{1}{2}\sin(2\theta)}{\tan(\theta)}$$

Since $\tan(\theta) = \frac{\sin(\theta)}{\cos(\theta)}$

$$= \frac{\cos(\theta)\,[\sin(\theta) + \frac{1}{2}\sin(2\theta)]}{\sin(\theta)}$$

Since $\sin(x+y) = \sin(x)\cos(y) + \cos(x)\sin(y)$
and thus $\sin(2\theta) = 2\sin(\theta)\cos(\theta)$

$$= \frac{\cos(\theta)\,[\sin(\theta) + \sin(\theta)\cos(\theta)]}{\sin(\theta)}$$

$$= \cos(\theta) + cos^2(\theta)$$

Since $cos^2(x) = \frac{1}{2}(1 + \cos(2x)$

$$= \cos(\theta) + \frac{1}{2}[1 + \cos(2\theta)]$$

$$= \cos(\theta) + \frac{1}{2}\cos(2\theta) + \frac{1}{2}$$

This is the same pair of equations as the reference figure, but with x translated by $+\frac{1}{2}$ as we see it in Figure 3-10 lower diagram.

Note also that the original reference figure equations are easily transformed into the equations of the cardioid (section 10.1):

$$y = \sin(\theta) + \frac{1}{2}\sin(2\theta)$$
$$= \sin(\theta) + \sin(\theta)\cos(\theta)$$

And as shown above

$$x = \cos(\theta) + cos^2(\theta)$$

(The signs are inverted as we have the cardioid reversed.)

12 Appendix Window-Functions

A selection of window-functions is provided. Their basic characteristics and coefficients are summarised here. For further information see the literature below.

12.1 TBD Add

12.1.1 Exponential

For use with transients longer than the record length (ringing).

$w(t)=e^{-(t-t0)/\tau}$ for $t_0 \leq t < T$ and $0 \leq \tau < T$ 0=

w (t) = 0 elsewhere

τ is the time constant = length of window
t_0 is shift of window
Brüel&Kjaer TR3,1987

12.1.2 Force

https://www.uml.edu/docs/Windows-Leakage_tcm18-191248.pdf
i.e. Force-Exponential
https://pearl-hifi.com/06_Lit_Archive/15_Mfrs_Publications/10_Bruel_Kjaer/05_Technical_Reviews/Proper_Use_of_Weighting_Functions_for_Impact_Testing.pdf

12.1.3 Parzen

Defining $L \triangleq N+1,$ the Parzen window, also known as the **de la Vallée Poussin window**,[14] is the 4th order *B*-spline window given by:

$$w_0(n) \triangleq \begin{cases} 1-6\left(\frac{n}{L/2}\right)^2\left(1-\frac{|n|}{L/2}\right), & 0 \leq |n| \leq \frac{L}{4} \\ 2\left(1-\frac{|n|}{L/2}\right)^3 & \frac{L}{4} < |n| \leq \frac{L}{2} \end{cases}$$

$$w[n] = w_0\left(n-\frac{N}{2}\right),\ 0 \leq n \leq N$$

https://en.wikipedia.org/wiki/Window_function#Parzen_window

12.1.4 Poisson

See Harris 1978

12.1.5 Cauchy

See Harris 1978

12.1.6 Chebyshev

https://www.dsprelated.com/showarticle/42.php
https://en.wikipedia.org/wiki/Window_function#Dolph–Chebyshev_window

12.1.7 Others

See Evaluation_of_Various_Window_Functions_using_Multi-Instrument_D1003.pdf
See Window function - Wikipedia.pdf and cf. Wikipedia original.

12.1.8 Generalised Adaptive Polynomial

See Justo and Beccaro paper 2020.
Also Sun, Liu, Cai & Long “A Novel Method for Designing General Window Functions with Flexible Spectral Characteristics”.

12.2 Rectangle (none)

```
windowFunction[i] = 1.0;
```

The wave is unchanged.

12.3 Bartlett

As defined in [Heinzel], who also references [Harris]:

```
double z = 2.0 * i / windowWidth;
if (z <= 1.0) {
      windowFunction[i] = z;
} else {
      windowFunction[i] = 2.0 - z;
}
```

This is a type of triangular window.

12.4 Welch

From [Heinzel]

```
double z = 2.0 * i / windowWidth;
windowFunction[i] = 1 - Math.pow((z - 1), 2);
```

This is equivalent to https://en.wikipedia.org/wiki/Window_function#Welch_window which gives:

The Welch window consists of a single parabolic section:

$$w[n] = 1 - \left(\frac{n - \frac{N}{2}}{\frac{N}{2}}\right)^2, \quad 0 \le n \le N.$$ [20]

The defining quadratic polynomial reaches a value of zero at the samples just outside the span of the window.

12.5 Hamming

As defined in [Heinzel] and [Harris]:

```
windowFunction[i] = 0.54 - (0.46 * (Math.cos(Math.PI*2*i/windowWidth)));
```

12.6 Hann

```
windowFunction[i] = 0.5 * (1-(Math.cos(Math.PI*2*i/windowWidth)));
```

Taken from [Wikipedia] and [Heinzel] rather than [Harris] as the latter gives the function with the sign reversed.

12.7 Cosine Sum Windows

These are calculated by a sum of cosine terms with varying coefficients:

```
windowFunction[i] = a0 – a1cos(2πi/N) + a2cos(4πi/N) – a3cos(6πi/N) …
```

where N is the window width and a_0, a_1, a_2, etc. are coefficients which define the window.
Blackman from https://en.wikipedia.org/wiki/Window_function#Blackman_window
Exact Blackman from https://en.wikipedia.org/wiki/Window_function#Blackman_window
Blackman-Harris-4 from https://www.mathworks.com/help/signal/ref/blackmanharris.html

Blackman-Harris-7 from https://dsp.stackexchange.com/questions/51095/seven-term-blackman-harris-window

12.7.1 Blackman and Blackman-Harris Windows

Coefficient	Blackman	Exact Blackman	Blackman-Harris-4	Blackman-Harris-7
a_0	0.42	7938/18608	0.35875	0.27105140069342
a_1	-0.5	-9240/18608	-0.48829	-0.43329793923448
a_2	0.08	1430/18608	0.14128	0.21812299954311
a_3			-0.01168	-0.06592544638803
a_4				0.01081174209837
a_5				-0.00077658482522
a_6				0.00001388721735

12.7.2 Nuttall Windows

From [Nuttall] and [Heinzel]

Coefficient	Nuttal-3	Nuttal-3a	Nuttall-3b	Nuttall-4	Nuttall-4a	Nuttall-4c
a_0	0.375	0.40897	0.4243801	0.3125	0.338946	0.355768
a_1	-0.5	-0.5	-0.4973406	-0.46875	-0.481973	-0.487396
a_2	0.125	0.09103	0.0782793	0.1875	0.161054	0.144232
a_3				-0.03125	-0.018027	-0.012604

12.8 Windows with Parameters

12.8.1 Gaussian

From [Wikipedia] (which differs from [Harris])

```
windowFunction[i] = Math.exp(-0.5 * Math.pow((i-m) / (gaussianSigma*m),2));
```

$$w[n] = e^{-\frac{1}{2}\left(\frac{n-N/2}{\sigma N/2}\right)^2}, 0 \le n \le N$$

The variable σ GaussianSigma is entered by the user and should be <0.5.

[Harris]:

$$w[n] = e^{-\frac{1}{2}\left(\alpha\frac{n}{N/2}\right)^2}, -N/2 \le n \le N/2$$

Where $\alpha = 1/\sigma$

12.8.2 Kaiser-Bessel

From [Heinzel], [Harris], [Wikipedia].

The variable α is entered by the user and [Heinzel] gives characteristics for values between 2.0 and 7.0, and [Wikipedia] gives 3.0 as a typical value.

We find two forms of the formula:

[Heinzel] and [Wikipedia]:

$$w[i] = \frac{I_0\left(\pi\alpha\sqrt{1-\left(\frac{2i}{N}-1\right)^2}\right)}{I_0(\pi\alpha)}$$ [Heinzel] for i = 0…N-1; [Wikipedia] 0<=i<=N EQ1

[Harris]:

$$w[i] = \frac{I_0\left(\pi\alpha\sqrt{1-\left(\frac{2i}{N}\right)^2}\right)}{I_0(\pi\alpha)}$$ For 0<=|i|<=N/2 EQ2

Where I_0 is the zero-order Bessel function of the first kind:

$$I_0(x) = \sum_{k=0}^{\infty} \left[\frac{\left(x/_2\right)^k}{k!} \right]^2$$

We use EQ1 with i = 0 …N-1.

12.8.3 Tukey

From [Wikipedia], which gives these errors in [Harris]:

> *Harris 1978 (p 67, eq 38) appears to have two errors: (1) The subtraction operator in the numerator of the cosine function should be addition. (2) The denominator contains a spurious factor of 2. Also, Fig 30 corresponds to α=0.25 using the Wikipedia formula, but to 0.75 using the Harris formula. Fig 32 is similarly mislabeled.*

$$\left.\begin{array}{l} w[n] = \frac{1}{2}\left[1 - cos\left(\frac{2\pi n}{\alpha N}\right)\right], \qquad 0 \le n < \frac{\alpha N}{2} \\ w[n] = 1, \frac{\alpha N}{2} \le n \le \frac{N}{2} \\ \\ w[N-n] = w[n], 0 \le n \le \frac{N}{2} \end{array}\right\}$$

For α=0 it becomes rectangular and for α=1 it becomes a Hann window.

12.9 Flat Top Fast Decaying Windows

From [Heinzel]

Coefficient	Flat-Top-Fast-3	Flat-Top-Fast-4	Flat-Top-Fast-5
a_0	0.26526	0.21706	0.1881
a_1	-0.5	-0.42103	-0.36923
a_2	0.23474	0.28294	0.28702
a_3		-0.07897	-0.13077
a_4			0.02488

12.10 Flat Top Minimum Side-Lobe Windows

From [Heinzel]

Coefficient	Flat-Top-Min.Lobe-3	Flat-Top-Min.Lobe-4	Flat-Top-Min.Lobe-5
a_0	0.28235	0.241906	0.209671
a_1	-0.52105	-0.460841	-0.407331
a_2	0.19659	0.255381	0.281225
a_3		-0.041872	-0.092669
a_4			0.0091036

12.11 Windows from Commercial Spectrum Analysers

From [Heinzel]

Coefficient	National Instruments Flat-Top	Hewlett Packard	Stanford Research SR785
a_0	0.2810639	1.0	1.0
a_1	-0.5208972	-1.912510941	-1.93
a_2	0.1980399	1.079173272	1.29
a_3		-0.1832630879	-0.388
a_4			0.028

12.12 Flat-Top Windows Developed for GEO600 Gravitational Waves

From [Heinzel]

Coefficient	HFT70	HFT95	HFT90D	HFT116D	HFT144D
a_0	1.0	1.0	1.0	1.0	1.0
a_1	-1.90796	-1.9383379	-1.942604	-1.9575375	-1.96760033
a_2	1.07349	1.3045202	1.340318	1.4780705	1.57983607
a_3	-0.18199	-0.4028270	-0.440811	-0.6367431	-0.81123644
a_4		0.0350665	0.043097	0.228389	0.22583558
a_5				-0.0066288	-0.02773848
a_6					0.00090360

Coefficient	HFT169D	HFT196D	HFT223D	HFT248D
a_0	1.0	1.0	1.0	1.0
a_1	-1.97441842	-1.979280420	-1.98298997309	-1.985844164102
a_2	1.65409888	1.710288951	1.75556083063	1.791176438506
a_3	-0.95788186	-1.081629853	-1.19037717712	-1.282075284005
a_4	0.33673420	0.448734314	0.56155440797	0.667777530266
a_5	-0.06364621	-0.112376628	-0.17296769663	-0.240160796576
a_6	0.00521942	0.015122992	0.03233247087	0.056656381764
a_7	-0.00010599	-0.000871252	-0.00324954578	−0.008134974479
a_8		0.000011896	0.00013801040	0.000624544650
a_9			-0.00000132725	-0.000019808998
a_{10}				0.000000132974

13 Appendix Chirp z-Transform

Note: this has not been implemented as modern processors are fast enough not to need the FFT convolution method. The program implements the transform directly.

$$X[k] = \sum_{k=0}^{M} \sum_{n=0}^{N} x[n] e^{-2\pi(\delta t * n)(f_{0+(\delta f * k)})}$$

Where:
x[n] = input sample n
N = number of samples in input
M = number of resulting frequencies
δt = time between input samples
f_0 = first frequency
δf = desired frequency spacing
X[k] = frequency sample k

The following shows my understanding of the chirp z-transform based on [Oppenheim] (p. 656-661) and [Rabiner] for future reference.
The steps are described in [Rabiner], but as these are for the general case of a spiral in the z-plane and we wish to remain on the unit circle, we use A_0 = 1 and W_0 = 1, as in [Oppenheim].
Note the following equivalences in notation:

[Oppenheim]	[Rabiner]
ω	2 π Θ
ω_0	2 π Θ_0
Δω	2 π ϕ_0
x[n]	x_n
g[n]	y_n
$X(e^{j\omega n})$	X_k
$e^{j\omega_0}$	$A = A_0 e^{j2\pi\theta_0}$ (A_0=1)
$W = e^{-j\Delta\omega}$	$W = W_0 e^{2\pi\phi_0}$ (W_0=1)

Basics:

$$\omega = 2\pi f$$
$$e^{j\theta} = cos\theta + jsin\theta$$
$$a^b * a^c = a^{(b+c)}$$
$$(a^b)^c = a^{(b*c)}$$
$$(a/b)^c = a^c/b^c$$

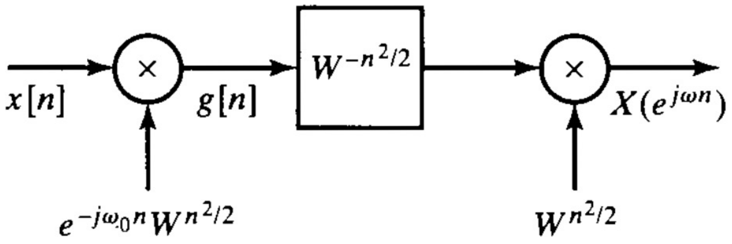


Figure 13-2 Oppenheim Fig. 9.26

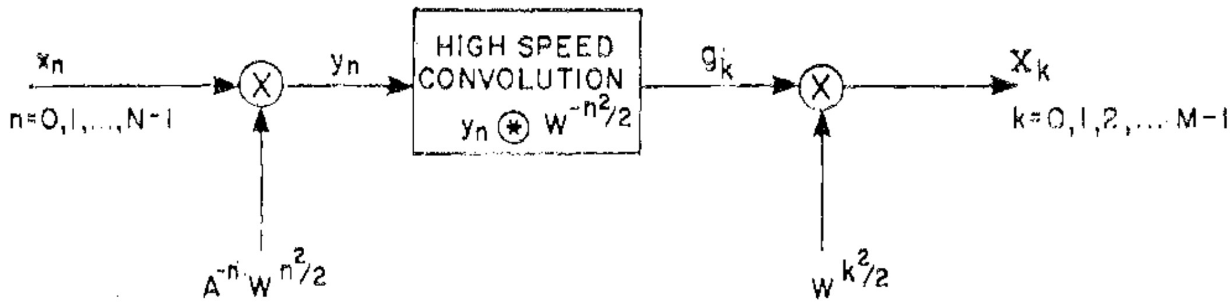


Figure 13-2Rabiner Fig. 3

For the algorithm we use frequency and convert ω = 2πf.
The chirp for the input "a" is:

$$e^{-j\omega_0}W^{\frac{n^2}{2}} = e^{-j2\pi f_0}\left(e^{-j2\pi\Delta f}\right)^{\frac{n^2}{2}} = e^{-j2\pi f_0}e^{-j2\pi\Delta f n^2/2} = e^{-j2\pi\left(f_0+\Delta f n^2/2\right)}$$

for 0<=n<=N-1, where N is the number of input samples.

The chirp for the output "b" is:

$$W^{\frac{k^2}{2}}=\left(e^{-j2\pi\Delta f}\right)^{\frac{k^2}{2}} = e^{-j2\pi\Delta f k^2/2}$$

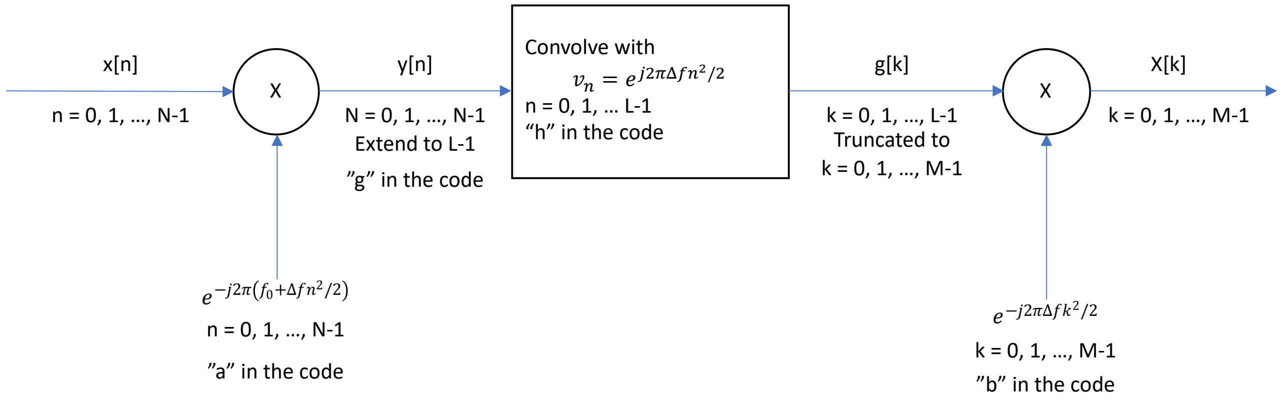
where 0<=k<=M-1, where M is the number of output frequencies.

The chirp for the convolution "h" is:

$$W^{\frac{-n^2}{2}}=\left(e^{-j2\pi\Delta f}\right)^{\frac{-n^2}{2}} = e^{j2\pi\Delta f n^2/2}$$

Where 0<=n<=L, where L is the next power of 2 greater than M+N+1 for the FFT.

To summarise:



14 Literature

[GEO600] Wikipedia page of the GEO600 gravitational wave detector .
https://en.wikipedia.org/wiki/GEO600 viewed on 20th December 2022.

[LIGO] Homepage of LIGO Laser Interferometer Gravitational-Wave Observatory
https://www.ligo.org viewed on 20th December 2022.

[Wikipedia] https://en.wikipedia.org/wiki/Window_function viewed on 17th December 2022.

[Heinzel] G. Heinzel et al., 2002. Spectrum and spectral density estimation by the Discrete Fourier transform (DFT), including a comprehensive list of window-functions and some new flat-top windows. E.g. at https://pure.mpg.de/rest/items/item_152164_2/component/file_152163/content viewed on 17th December 2022.

[Harris] Fredric J. Harris. On the Use of Windows for Harmonic Analysis with the Discrete Fourier Transform. Proceedings of the IEEE, Vol. 66 no.1 January 1978. E.g. at https://www.ak.tu-berlin.de/fileadmin/a0135/Unterrichtsmaterial/EDS_WS0910/Harris_1978_On_the_use_of_windows_for_harmonic_analysis_with_the_discrete_fourier_transform_IEEE.pdf viewed on 17th December 2022.
[Nuttall] Albert H. Nuttall. Some Windows with Very good Sidelobe Behavior. IEEE Transactions on Acoustics, Speech and Signal Processing, Vol. ASSP-29, No.1 February 1981.
https://ieeexplore.ieee.org/document/1163506

[ID3] The specifications of the ID3 tags can be found here https://github.com/id3/ID3v2.4 or here https://id3lib.sourceforge.net/id3/ . The original ID3 site www.id3.org appears to be defunct. Viewed on 28th March 2023.

[Brüel&Kjaer TR3,1987] Brüel & Kjaer, Technical Review No. 3 1987. Window to FFT Analysis (Part I)
Use of Weighting Function in DFT/FFT Analysis (Part I).
Signals and Units
https://www.bksv.com/media/doc/bv0031.pdf
Viewed on 6th April 2023.

[Brüel&Kjaer TR4,1987] Brüel & Kjaer, Technical Review No. 4 1987. Window to FFT Analysis (Part II)
Use of Weighting Function in DFT/FFT Analysis (Part II).
Acoustic Calibrator for Intensity Measurement Systems.
https://www.bksv.com/media/doc/bv0032.pdf
Viewed on 6th April 2023.

[Brüel&Kjaer TR1,1994] Brüel & Kjaer, Technical Review No. 1 1994. Digital Filter Techniques vs. FFT Techniques for Damping Measurements (Damping Part I).
https://www.bksv.com/media/doc/bv0044.pdf
Viewed on 6th April 2023.

[Brüel&Kjaer TR2,1994] Brüel & Kjaer, Technical Review No. 2 1994. Damping Measurements – From Impulse Response Functions – From Resonance and Non-resonance Excitation Techniques.
Damping Part II
The use of Impulse Response Function for Modal Parameter Estimation.
Complex Modulus and Damping Measurements using Resonant and Non-Resonant Methods.

https://www.bksv.com/media/doc/bv0045.pdf
Viewed on 6th April 2023.

[Oppenheim] Oppenheim, Allan V., Schafer, Ronald W., with Buck, John R. Discrete-Time Signal Processing. Second Edition, Prenctice Hall, 1999.

[Rabiner] Rabiner, L. R., Schafer, R. W., Rader, C. M. The Chirp z-Transform Algorithm. IEEE Transactions on Audio and Electroacoustics. Vol. Au-17, No. 2, June 1969.

[Sethares] Sethares, William A. Tuning, Timbre, Spectrum, Scale. London: Springer-Verlag London Limited, 1998, 2nd edition 2005.

[Smith] Smith, Julius O. III. Spectral Audio Signal Processing. W3k Publishing 2011.
Also at https://ccrma.stanford.edu/~jos/sasp/ and https://www.dsprelated.com/freebooks/sasp/