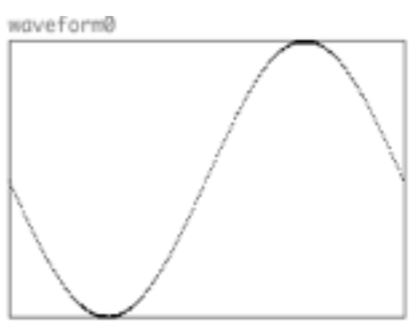
Sound and Interactive Music - FFT

Joseph Fourier demonstrated that any periodic wave can be expressed as the sum of harmonically related sinusoids, each with its own amplitude and phase

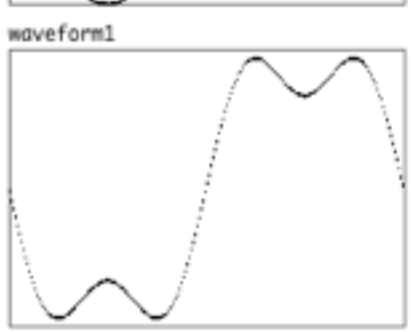
Every signal can be represented by a sum sine waves which have a **frequency, amplitude and phase**

Simple examples:

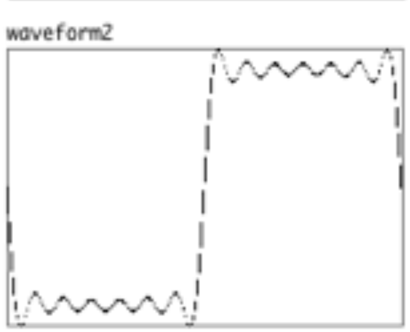
Square Wave



Produced by adding all odd harmonics of a sine wave at decreasing amplitude 1/h - where h = harmonic

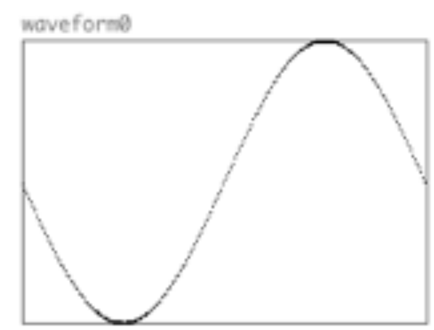


4 harmonics

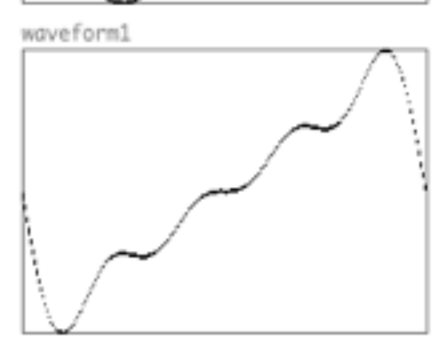


12 harmonics

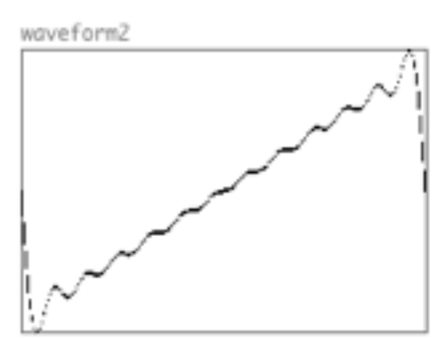
Sawtooth Wave



Produced by adding all harmonics of a sine wave at decreasing amplitude 1/h - where h = harmonic



4 harmonics



12 harmonics

Fourier Transform

the fft identifies all the **frequency components** within particular wave form

the general idea of **analysing a sound** by **breaking it into its frequency** components, and conversely, using a **bunch of frequency components** to **synthesise** a new sound.

This process performs what is called Fourier transformation. It divides the entire **frequency spectrum into parts of equal size** and determines the **amplitude and phase** for each part. One could in turn reconstruct the original signal from these values.

The derivation of the component parts is called analysis;
the reconstruction is called resynthesis.
You can realize this using the objects "rfft~" and "irfft~"

Fourier Transform

The output of the FFT is a **complex vector** containing information about the frequency content of the signal. The **amplitude** tells you the strength of the frequency components relative to other components. The **phase** tells you how all the frequency components align in time.

Since a Fourier Transform assumes periodicity, any **discontinuity** between the last sample and the repeated first sample will cause artefacts in the spectrum (e.g. "smearing" of the peaks).

To reduce the effect of this we apply a tapered window function such as a Hann window which smooths out any such discontinuity and thereby reduces artefacts in the spectrum.

Usually a normalisation process is conducted after a FFT process, because the amplitude values become fairly high.

FFT - Analysing Signals

- Each bin corresponds to a certain frequency within the spectrum. With block size of 64 and bin size of 689Hz analysing a signal containing only frequency of 689Hz (or a multiple of it) is fine and can be represented by a amplitude value in a single bin.

- Signals that are not of frequency 689Hz are divided among several multiple bins.

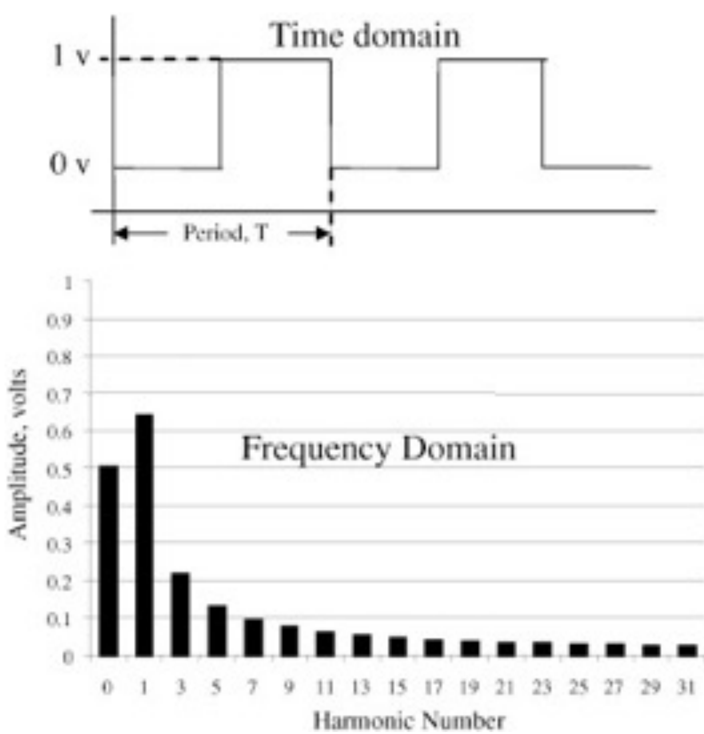
- Each time the analysis is carried out the phase of the corresponding frequency values will change. - THIS IS A PROBLEM!

- To solve this problem use overlapping windows.

Fourier Transform

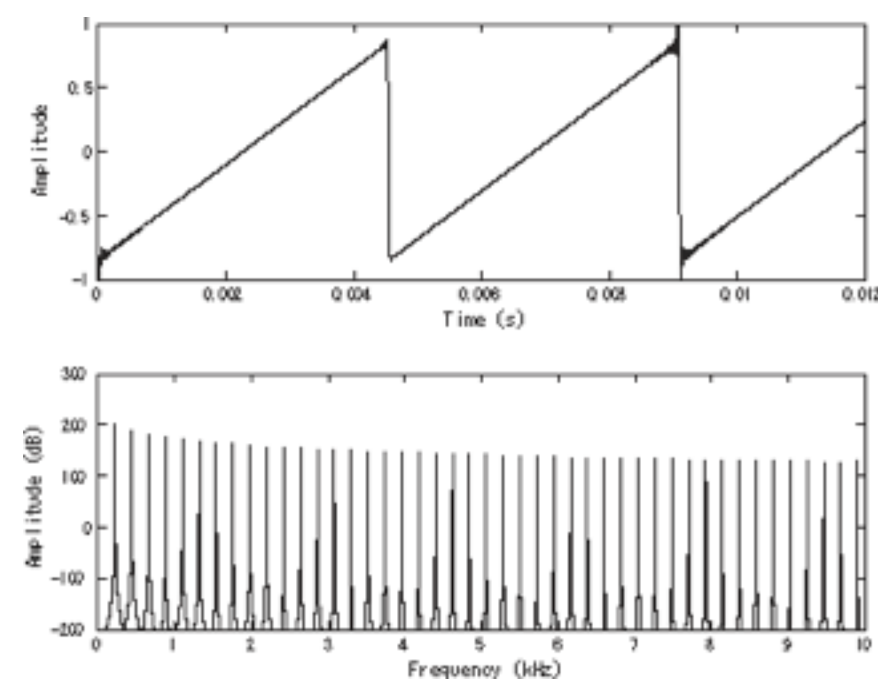
• **<u>The Fourier Transform</u> -** Transforms a **Signal** from the **Time Domain** representation into its **Frequency Domain** representation. This is also know as **Analysis** frequency components of the signal are worked out.

<u>Square wave</u>



taken from http://www.informit.com/articles/article.aspx?p=1374896&seqNum=7

<u>Sawtooth wave</u>

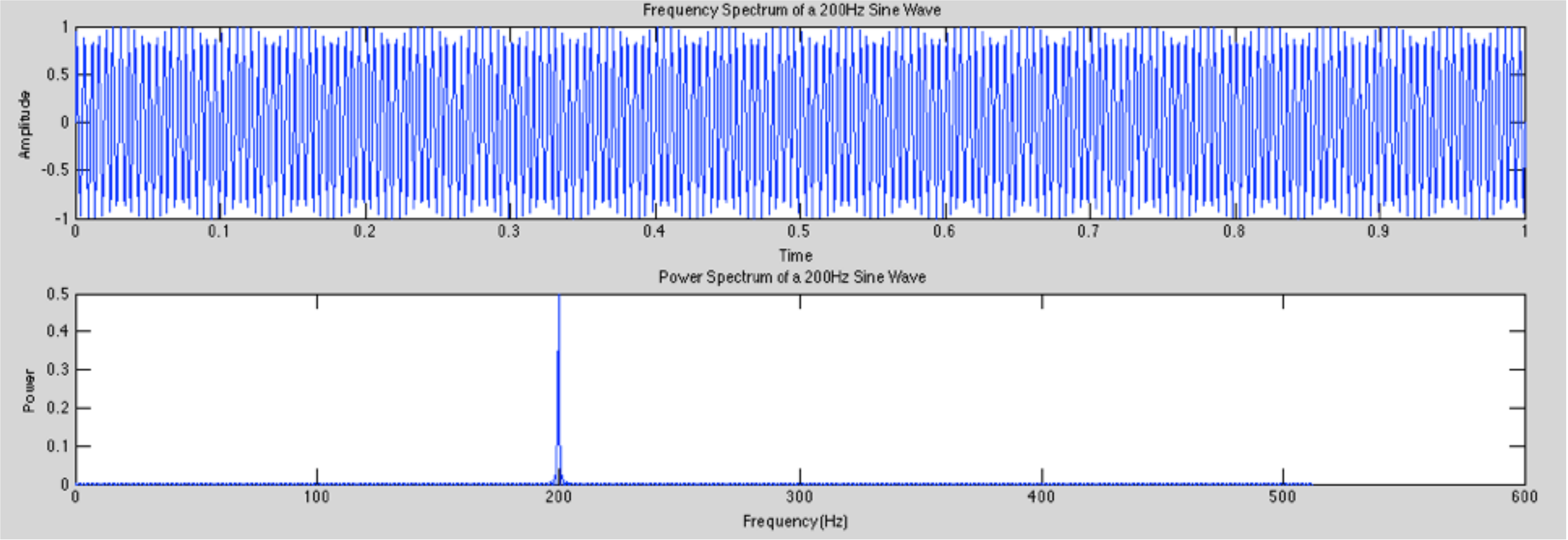


taken from http://en.wikipedia.org/wiki/File:Sawtooth-td_and_fd.png

•**<u>Inverse Fourier Transform</u> -** transforms a **Signal** from the **Frequency Domain** representation **back** into the **Time Domain** representation. Also know as re-synthesis as the signal is re-constructed or re-synthesised.

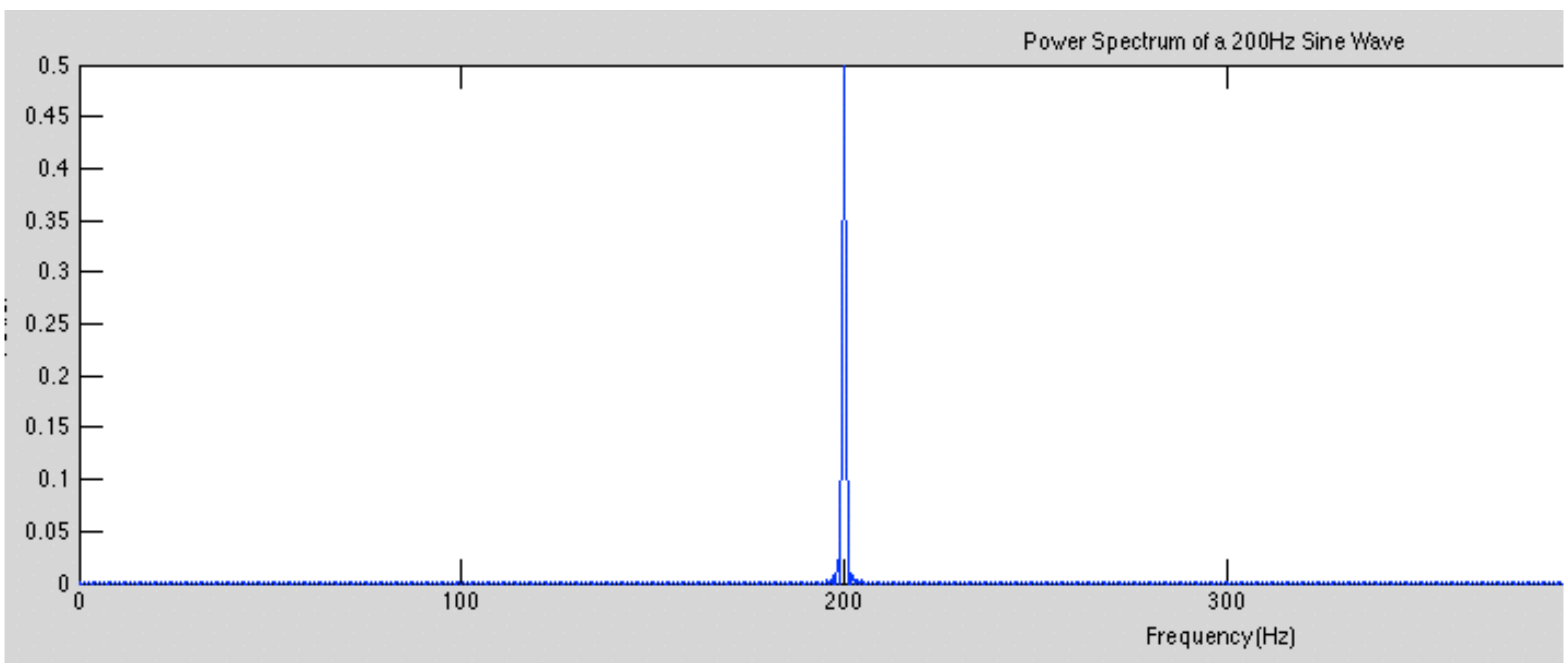
Sine Wave Example

- Time Domain Plot of a 200Hz Sine Wave (top) - note signal is present all times - snapshot of 1 second is shown.
- Frequency Domain Plot (bottom) - note only frequency is present at 200Hz.



In the Real World

- Because the real signals are limited in both domains the Fourier analysis cannot be perfect.
- Example is here with the sine wave frequency plot - ideally this should be a single line but due to errors introduced because the signal is not of infinite time frequencies around 200Hz are plotted with small amplitudes.

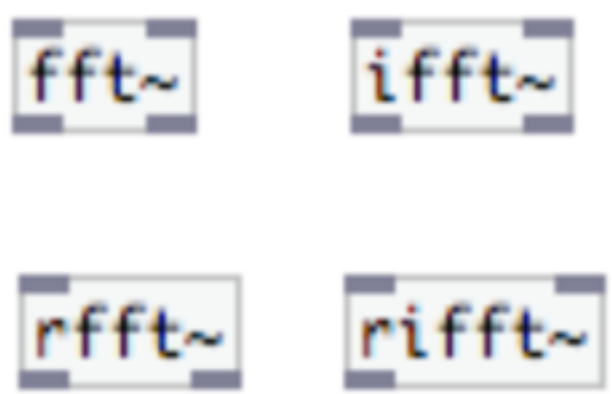


Fast Fourier Transform - FFT

• Computationally efficient way of carrying out the Fourier Transform in the Digital Domain (aka Discrete Fourier Transform - DFT)

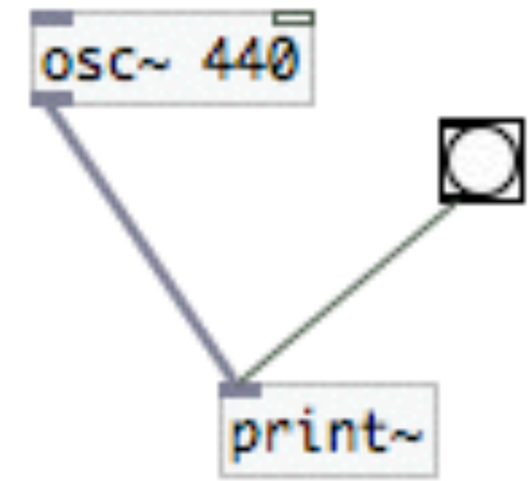
• Understanding of the maths that specifies what the FFT is doing and how it is doing it is not needed in order to use the FFT and preform Fourier analysis and re-synthesis.

• FFT objects in Pure Data:



• These are [fft~] to preform analysis and [ifft~] to preform re-synthesis these produce real and imaginary components of the frequency spectrum.
• [rfft~] and [rifft~] do the same thing but only use real values of the frequency spectrum.

FFT - in Pure Data



- The FFT divides the frequency spectrum up into parts of equal size.
    - The amplitude and phase for each of these parts is then determined

- The parts of the frequency spectrum are called **bins**
- The **size** of each frequency **bin** is determined by the **block size**

- Within Pure Data **all tasks** a processed in blocks. Blocks are **groups of samples**
- Default block size in Pure Data is 64 - (can be changed in a subpatch)
- Can use [print~] to see snapshot of amplitudes within one DSP block.

- With default block size a spectrum is divided by 64. Example:

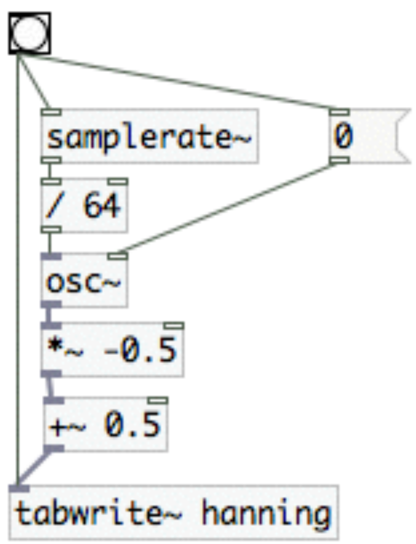
44100 / 64 = 689Hz

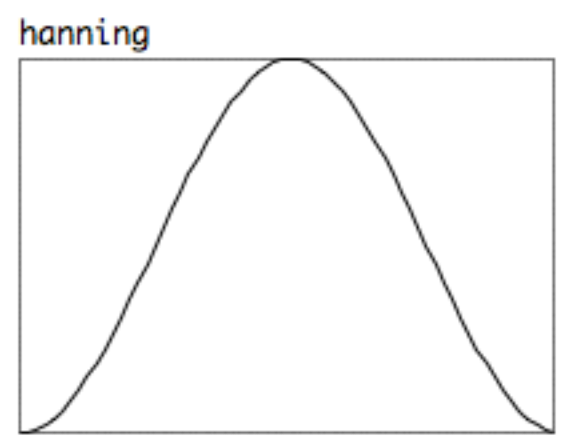
- Each bin therefore has a size of 689Hz
- The **larger the block size** the **better** the **frequency resolution -** but this also makes the process more computationally demanding.

FFT - Hanning Window

- The common window used is called a Hanning window

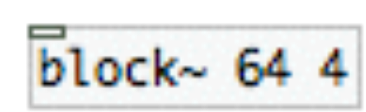
- Made in Pure Data in the following way:





- This stops the amplitudes being so spread out.

- Windows need to overlap. Easily done by using [block~ 64 4] - but this means the FFT analysis needs to be in a subpatch



FFT - Re-synthesis

• Re-synthesis of the signal is done by passing the outputted values from the [fft~] or [rfft~] into the corresponding inverse abstraction ([ifft~] or [rifft~])

• By altering the values obtained from the analysis before re-synthesis will alter how the original signal sounds.

• Things that you can do with FFT include:

- Filter effects
- Convolution
- Compression
- Spectral Delay

• As well as using the analysis to create control information and create things like:

- Tuners
- Octave Doublers
- Pitch Followers

Further sources of Information

- http://www.pd-tutorial.com/english/ch03s08.html
- http://footils.org/2007/02/20/beginners-guide-fft-objects-pd/
- http://academicearth.org/courses/the-fourier-transform-and-its-applications
- http://www.dsprelated.com/dspbooks/mdft/