Full of Sound and Fourier

A collection of apps for demonstrating how different instruments produce different sounds by generating sound waves that are different combinations of overtones.

What is sound? Amplitude and frequency

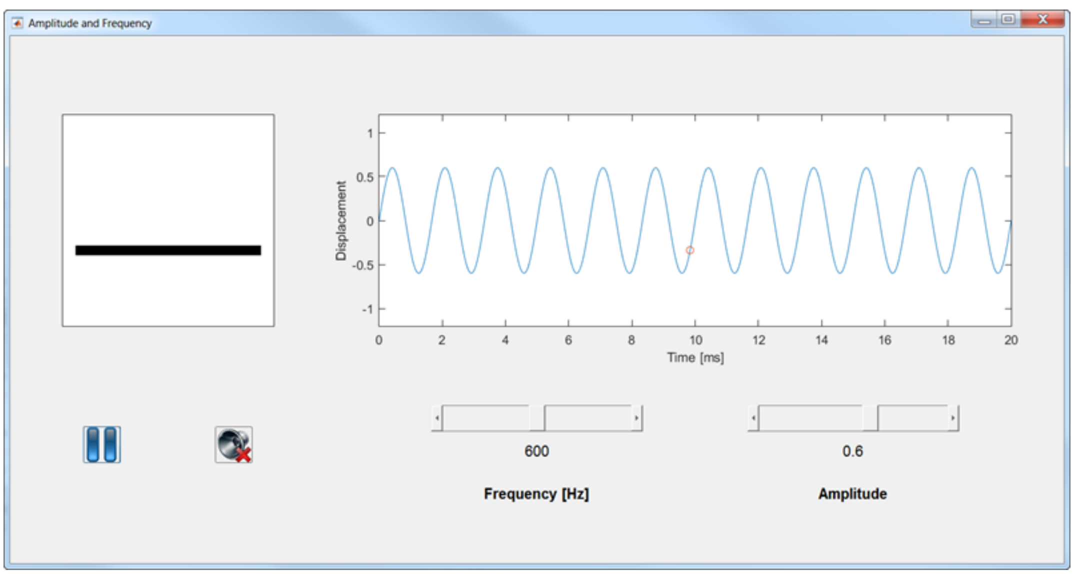
Sound is a pressure wave created by vibration. The vibration pattern (waveform) is transmitted through a dense medium such as air or water. A simple sound wave can be created by the oscillation of an object -- the membrane of a drum, the side of a glass, the tine of a tuning fork.

The larger the oscillation, the louder the sound produced. That is, the *amplitude* of the sound wave is experienced as the volume of the sound.

The faster the oscillation, the higher the pitch of the sound produced. That is, the *frequency* of the sound wave is experienced as the pitch of the sound.

Waves are represented mathematically as periodic functions of displacement as a function of time. A simple oscillation can be represented as a sinusoid: $\mathrm{displacement} = A\sin(\omega t)$, where A is the amplitude of the wave, and ω is the frequency.

Run `ampFreqDemo` to demonstrate the properties of amplitude and frequency, and how these affect the mathematical representation of a simple sinusoidal wave.

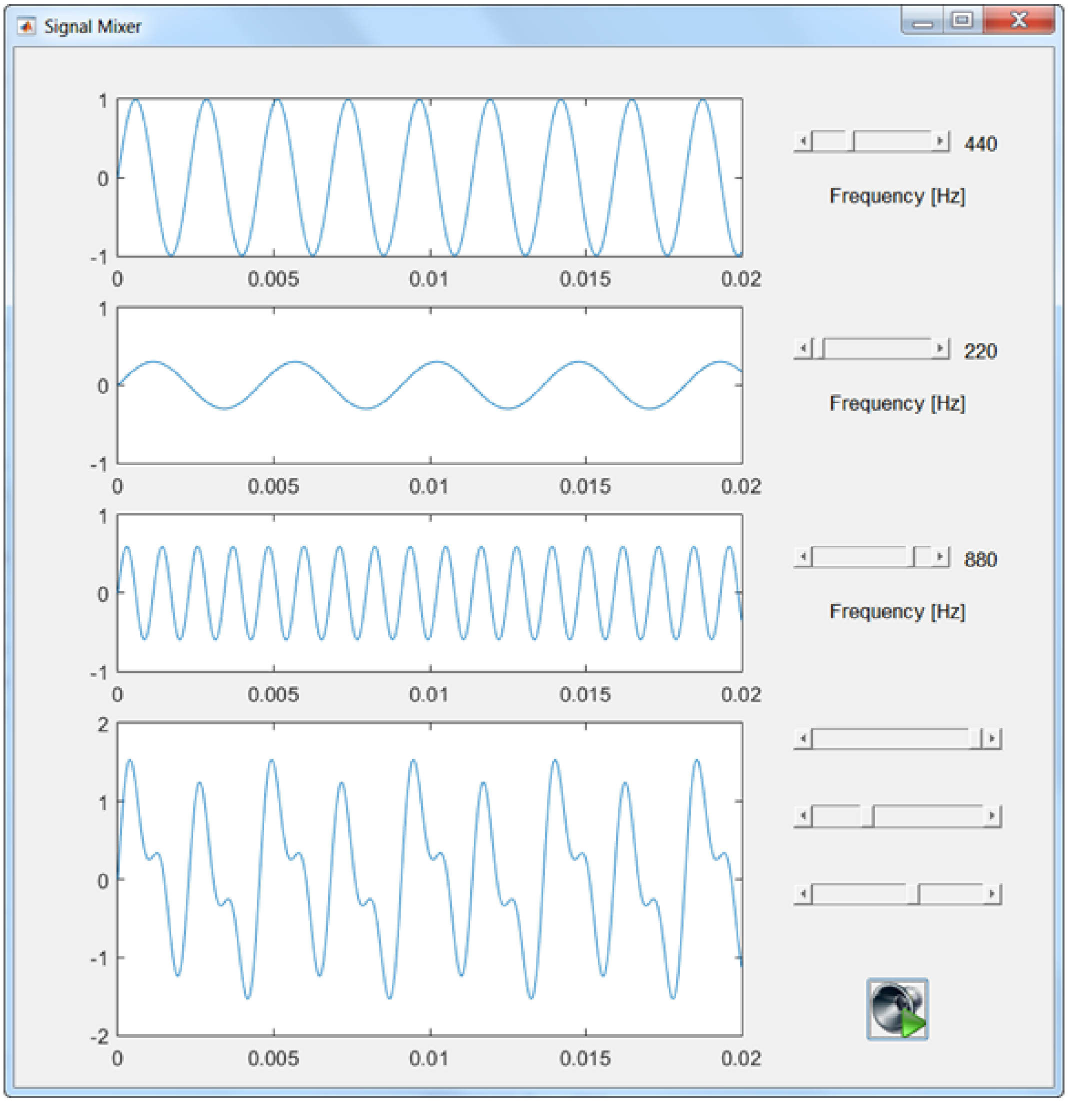


Change the amplitude and frequency sliders to see how this affects the movement of the physical oscillator (left) and the mathematical representation (right). Use the buttons to pause the animation or play the sound.

Greater than the sum of their parts

The richness of the sounds we encounter in life comes from the fact that they are not just simple sinusoids. But complicated waveforms can be broken down into a combination of simple sinusoids. This is equivalent to combining pure notes of different frequencies (pitches) in different amounts (amplitudes).

Run `mixingSignalsDemo` to see how combining just three different notes can produce a variety of sounds.



The three top graphs show single pure notes of various frequencies, which are controlled by the sliders to the right. These three waves are joined together in various amounts to produce the waveform at the bottom. The sliders to the right control the amplitudes of the three components making up the composite waveform. Initially only one component is present, so the result is a pure note. Change the amplitude sliders to see the effect on the resulting waveform. Use the play button to hear the sound.

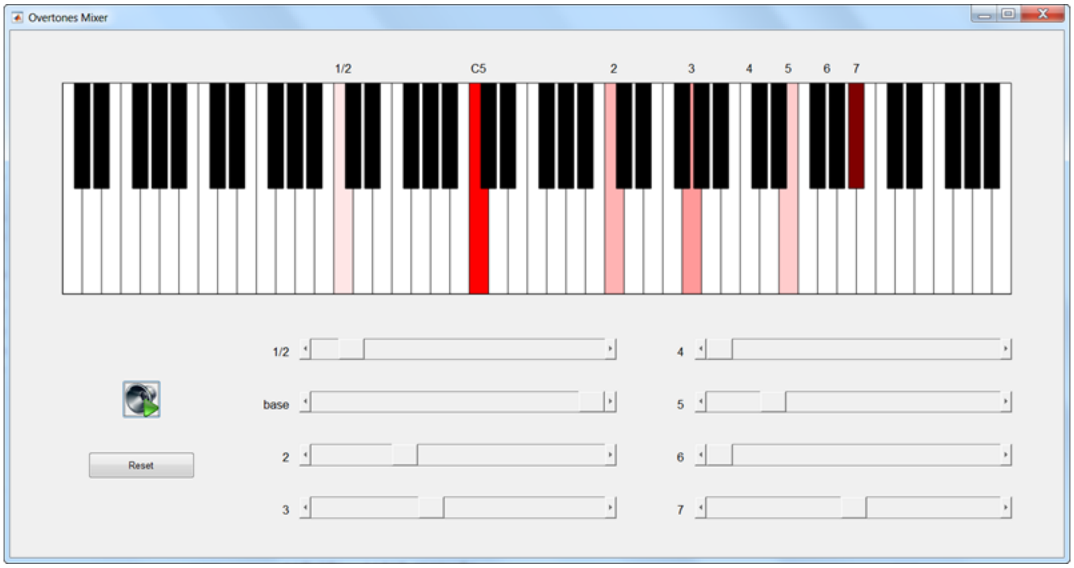
Variety is the spice of music

Why do different musical instruments produce different sounds (*timbres*) even when playing the same note? The sound created by an instrument is not a pure note, but a combination of notes, similar to what was just previously demonstrated. Typically, an instrument playing a note will produce a sound wave that includes a significant contribution from the frequency associated with that note, but also contributions (in varying amounts) from multiples of that frequency (called *overtones*). For example, orchestras often tune to the note A4 which has a nominal frequency of 440 Hz. The sound wave produced by an instrument playing an A4 will consist predominantly of 440 Hz (the *fundamental frequency*), but often with significant contributions of 220 Hz, 880 Hz, 1320 Hz, 1760 Hz, etc. (the overtones).

When the frequency of a note is doubled, we perceive it as the same note, an octave higher. Thus, 220 Hz is A3, 880 Hz is A5, and 1760 Hz (2*880 = 4*440) is A6. Note that this means that the overtone 1320 Hz (3*440) is not an A but (approximately) an E6. Similarly, the overtone 2200 Hz is a C#7.

Hence, the presence of overtones in an instrument's sound means that each note is actually a chord. The way an instrument is constructed (and played) determines how much of each note in the chord contributes to the overall sound.

Run `overtonesDemo` to see how overtones contribute to different kinds of sounds.

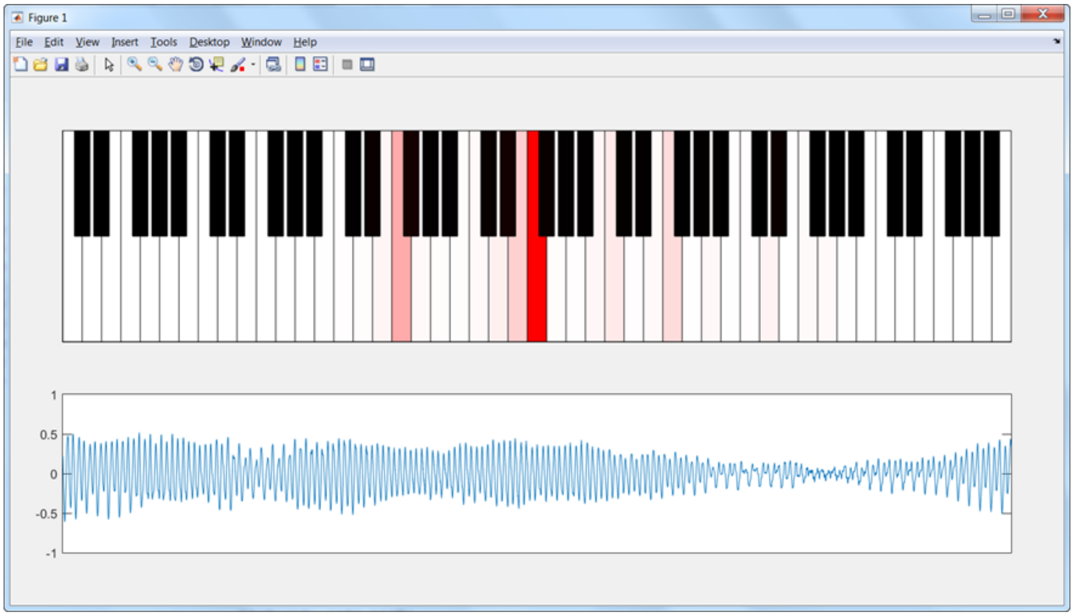


Click a key on the keyboard to set the fundamental note. The notes corresponding to the overtones (1/2, and 2 through 7) are labeled. Use the sliders below the keyboard to change the amount each overtone contributes to the overall sound; the keys are colored accordingly. Use the play button to hear the resulting sound. Note how different combinations of overtones produce different timbres for the same fundamental note.

Break it down

So far the examples have illustrated building complex waveforms by combining individual notes. The mathematical framework of *Fourier series* can be used to go the other way: take a sound wave and decompose it into a combination of individual notes (the wave's *spectrum*).

Run `soundAnalysisDemo` to see a waveform decomposed into its spectrum.

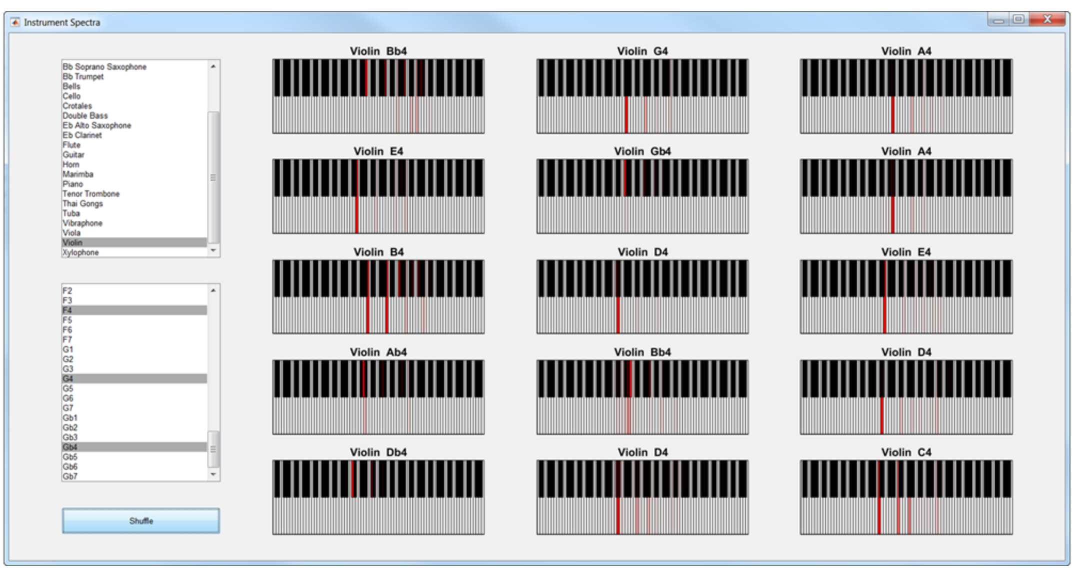


The waveform shown at the bottom is the last 0.25 seconds recorded by the computer's microphone. The contributions from each note are represented by the colors of the keys in the keyboard at the top.

What makes a violin a violin?

As demonstrated earlier, different combinations of overtones produce different sound qualities (timbres). An instrument creates a certain distinctive signature of overtones, because of the mechanics of how it creates sound. Different instruments produce a different spectrum, even when playing the same note.

Run `instrumentSpectraDemo` to see spectra for actual recordings of instruments playing various notes.



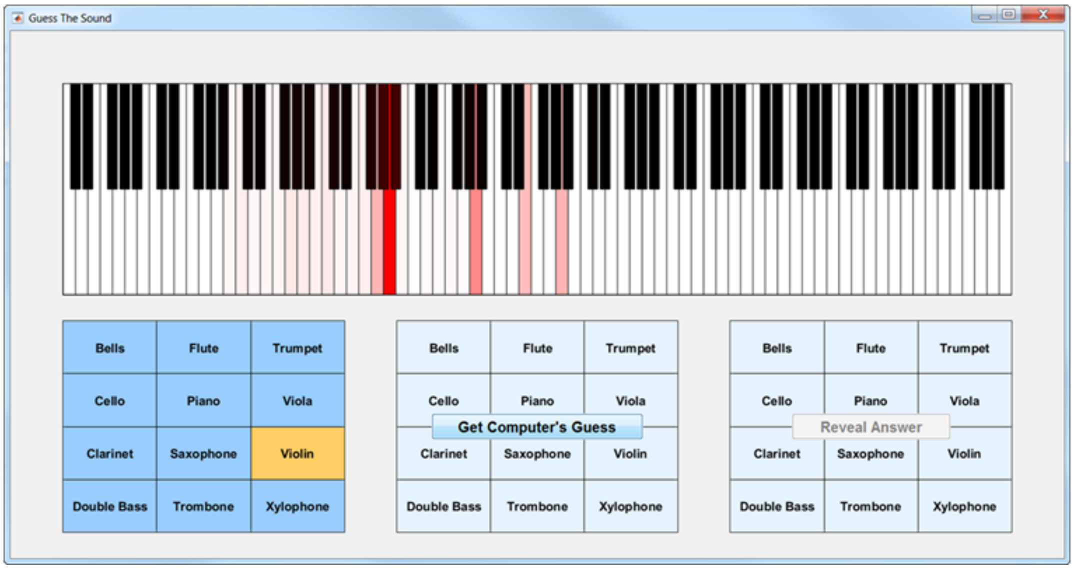
Use the selection boxes on the left to restrict the sound clips to specified instruments and/or notes. Click "Shuffle" to generate a new selection of sound clips with the given characteristics. The keyboards show the spectra (determined using Fourier series). Click a keyboard to play the sound.

So what?

Analyzing and manipulating spectra is a major part of audio processing (and signal processing in general, which can include other kinds of signals, such as earthquake signals). For example, touch-tone phones work by generating and decomposing tones consisting of two specific frequencies.

As an illustration of the power of spectral analysis, consider the problem of determining an instrument from its sound. This is something humans can do relatively well, but computers find difficult. One way to do it is to use *Machine Learning* classification algorithms, which are designed to learn from known data and make predictions on new (previously unseen) examples. But what does the classification algorithm use as its input data? The sound spectrum!

Run `guessTheSound` to play a game of guessing the instrument from the sound.



As before, the keyboard shows the spectrum visually. Click the keyboard to play the sound. Select one of the instruments in the left-most grid to make a guess. Click "Get Computer's Guess" to see the prediction made by a machine learning algorithm. Click "Reveal Answer" to see the actual instrument that generated the sound.

(For reference, the machine learning prediction used bootstrapped aggregation of classification trees, with the normalized discretized spectrum as input.)