

- Previous class:
 - Color vectors – RGB
 - 1-dimensional array – vector

- Now:
 - Play with sound files
 - (more vectors!)

Computing with sound requires digitization

- Sound is continuous; capture its essence by sampling
- Digitized sound is a vector of numbers

The diagram illustrates the digitization process. The top part shows 'Analog input' as a smooth, continuous green curve over 'Time'. Vertical lines mark 'Sample times' where the curve is sampled. The bottom part shows 'Digital output' as a discrete staircase function, where the value of the signal is constant between sample times and changes at each sample time.

Sampling Rate

Given human perception, 20000 samples/second is pretty good (20000Hz or 20kHz)

8,000 Hz	required for speech over the telephone
44,100 Hz	required for audio CD
192,400 Hz	required for HD-DVD audio tracks

Resolution also affects the quality

Typically, each sampled value is encoded as an 8-bit integer in the .wav file.

Possible values: -128, -127, ..., -1, 0, 1, ..., 127

Loud: -120, 90, 122, etc.

Quiet: 3, 10, -5

16-bit used when very high quality is required.

Reading and playing .wav files

```
[y,rate,nBits] = wavread('austin.wav')
sound(y,rate)
```

A wav file is for the computer to process— software is required to play the sound.

Computing with sound in Matlab requires that we first convert the wav format data into simple numeric data—the job of **wavread**.

wavread converts the 8-bit values to floating point values between -1 and 1

```
[y,rate,nBits]= wavread('austin.wav')
```

```

0.4609
0.3516
0.2734
0.2891
0.2500
0.1484 ← y(50000:50012)
0.1094
0.1641
0.1484
0.0000
-0.1641
-0.2734
-0.3281
    
```

wavread

```
[y,rate,nBits]= wavread('austin.wav');
n = length(y);
```

```
n =
    54453
rate =
    11025
nBits =
     8
```

austin.wav encoded the sound with 54,453 8-bit numbers that were gathered over a span of about 54453/11025 secs

8

wavread

```
[data,rate,nBits]= wavread('austin.wav')
```

Name of the source file

The vector of sampled sound values is assigned to this variable

The sampling rate is assigned to this variable

The resolution is assigned to this variable

9

Hearing and "seeing" the sound

```
[y,rate]= wavread('austin');
sound(y, rate)
plot(1:length(y), y)
```

Usually playback at a rate equal to the sampling rate

11

Subvectors

- Can access just part of a vector, or subvector
- Suppose you have a vector v:
 - v(1) - value in 1st cell
 - v(k) - value in kth cell for valid k
 - v(2:5) - the 2nd thru 5th values in v, as a vector
 - length(v) - how many cells in vector v
 - v(1:length(v)) - all the values in v

11

Building a vector

Concatenate two vectors to make one...

```
v= ones(1,3); % a row of length 3
w= [4; 7]; % a column of length 2
x= [w; v']; % a column of length 5
```

Concatenate vectors repeatedly...

```
a= [];
for k= 1:4
    a= [a ones(1,2)];
end
% What is a?
```

12

Check out these two files that demonstrate sound and graphics commands

movies.m

showSound.m

13