Learning to Program through Digital Audio and MIDI Applications

Jennifer Burg
Department of Computer Science
Wake Forest University
Winston-Salem, NC

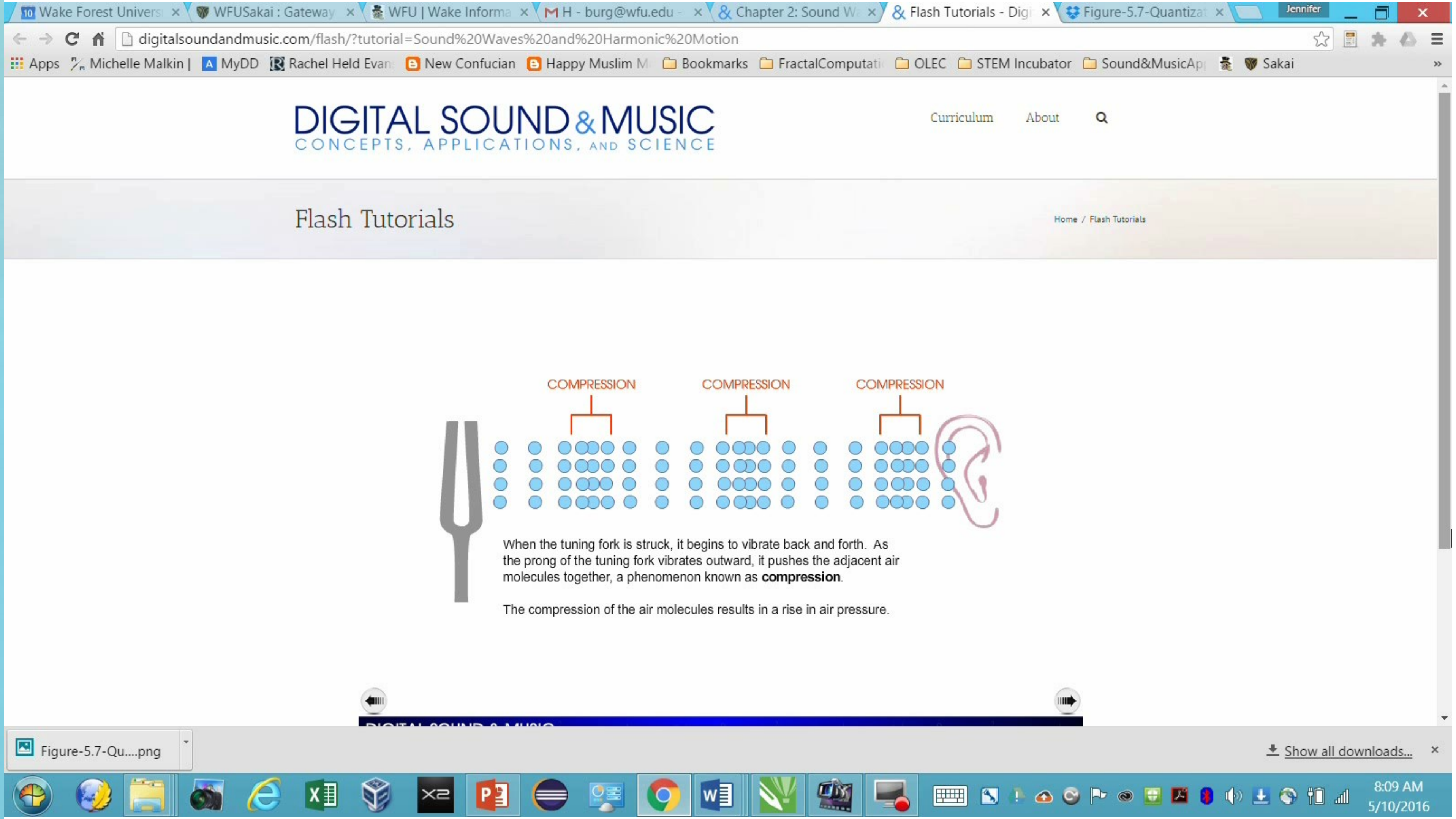
Funded by NSF grant DUE1323593, entitled "Computing in the Arts—A Community Building Initiative," a collaborative grant with College of Charleston, Wake Forest University, and UNC Asheville

- If you’re not familiar with digital audio, MIDI, or computer programming, imagine this course is for you.
- If you *are* familiar with digital audio, MIDI, and computer programming, imagine teaching this course.
- Hang on to your hats! We’re going to move fast!

Lesson One: The Physics and Mathematics of Sound

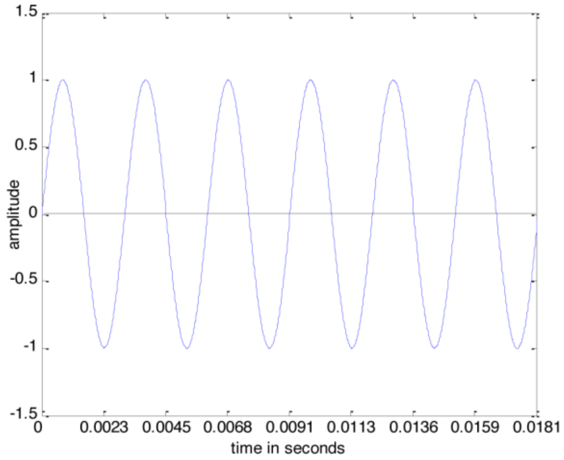
- **Main concept:** Sound is the result of vibrations of air molecules, causing air pressure to rise and fall in a pattern that propagates from the source of the sound to the hearer of the sound.

Sound Wave


digitalsoundandmusic.com/flash/?tutorial=Sound%20Waves%20and%20Harmonic%20Motion
DIGITAL SOUND & MUSIC
CONCEPTS, APPLICATIONS, AND SCIENCE
Curriculum
About
Flash Tutorials
Home / Flash Tutorials
COMPRESSION
COMPRESSION
COMPRESSION
When the tuning fork is struck, it begins to vibrate back and forth. As the prong of the tuning fork vibrates outward, it pushes the adjacent air molecules together, a phenomenon known as compression.
The compression of the air molecules results in a rise in air pressure.

Single Frequency Sounds (One Pitch)

- The note D. 
- The rising and falling air pressure that creates a single frequency sound can be modeled by a sine function.
- $y = A sin 2\pi f$, where A is the amplitude and f is the frequency



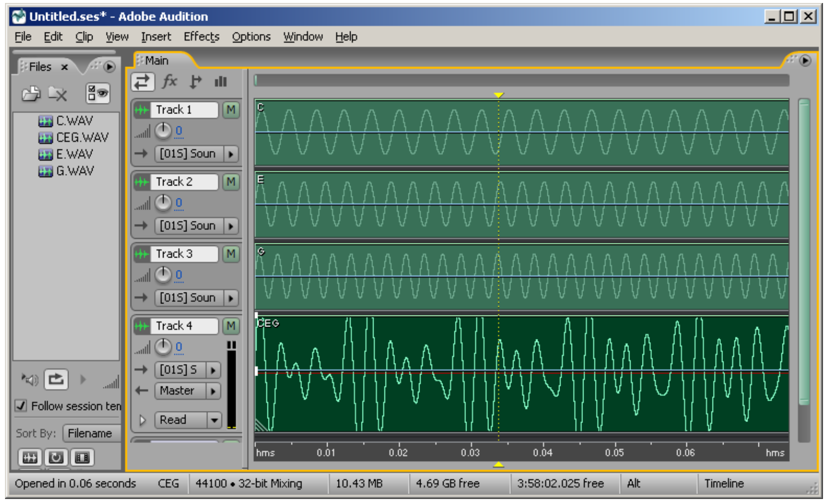
Frequency, Perceived as Pitch

- The frequency of a single-frequency sound – that is, the rate, per second, at which the air pressure rises and falls – is measured in Hertz (abbreviated Hz).
- The frequency is perceived as pitch by the human hearing system.

The notes C, E, and G

Complex Sounds

- Any sound, no matter how complex, can be represented mathematically as the sum of frequency components.
- For example, this chord in the key of C is the sum of three frequency components: the notes C, E, and G.

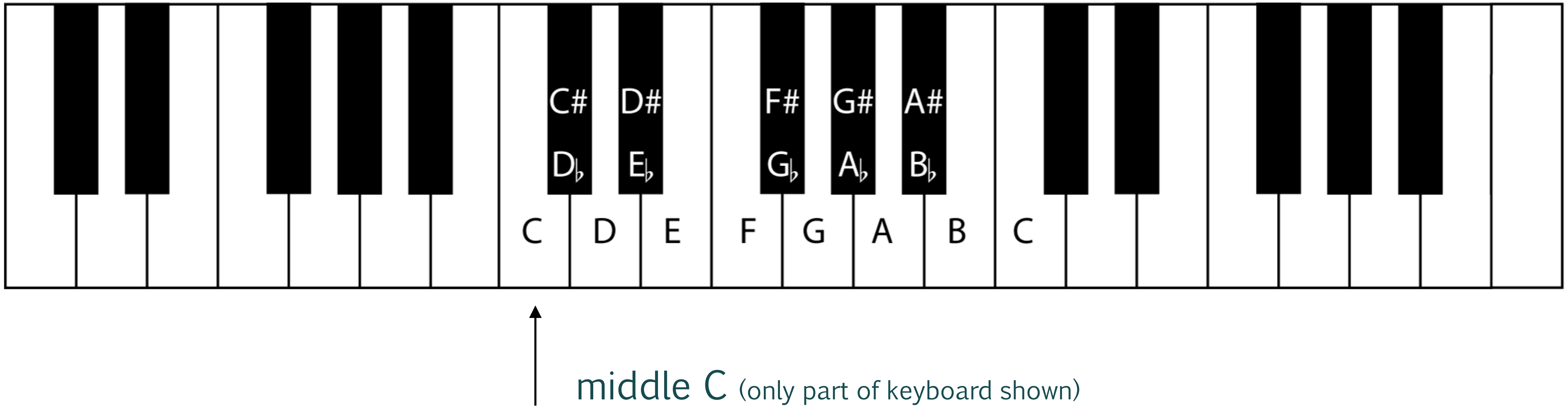


Lesson Two: The Mathematics of Musical Sound

- **Main concept:** In the tradition of Western music, certain sound frequencies are used and combined in sequences of notes (melodies) and combinations of notes (chords and harmonies).

Notes on a Piano Keyboard

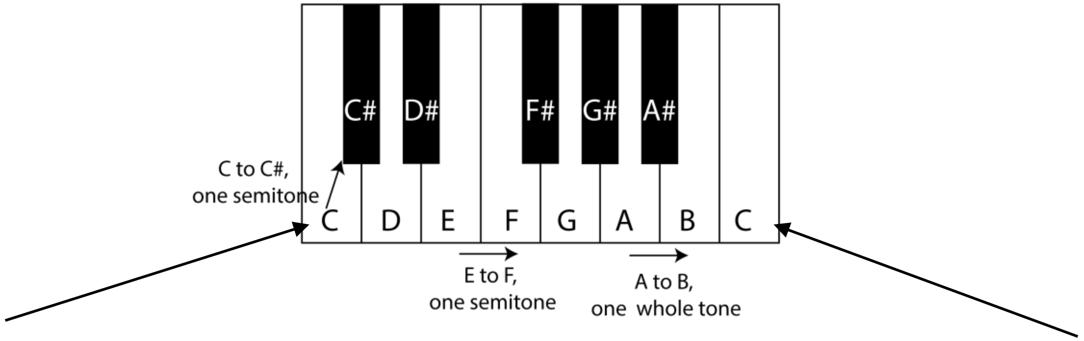
- Notes on a piano keyboard are labeled with letters.
- The notes correspond with specific frequencies. For example, “middle C” on the keyboard has a frequency of 262 Hz.
- In the tradition of Western music, only certain fixed frequencies are generally used.



(only part of keyboard shown)

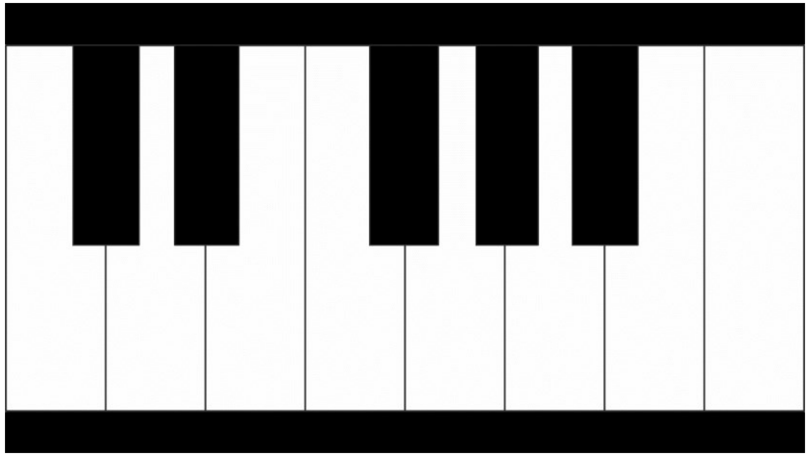
Notes in a Chromatic Scale

- A scale is a sequence of notes that our ears have been trained to recognize as a pattern in Western music.
- A chromatic scale uses 13 frequencies in a row such that each frequency is multiplied by $\sqrt[12]{2}$ (the twelfth root of 2) to get the next one. In this way, at the note at the end of the scale has 2 times the frequency of the first note. The notes are separated by **an octave**.
- The notes in the chromatic scale are considered to be **a semitone** apart.



Chromatic Scale
one semitone between successive notes

1 1 1 1 1 1 1 1 1 1 1 1

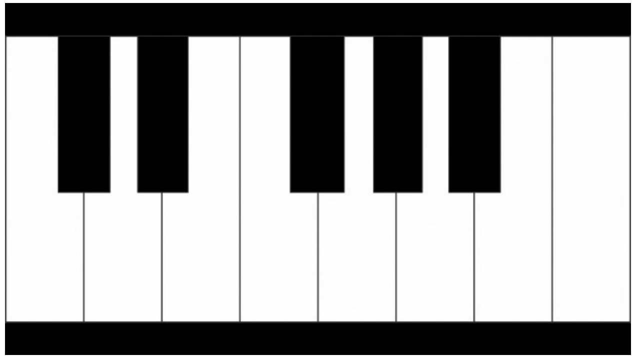


Notes in a Diatonic Scale

- The diatonic scale is a sequence of frequencies that our ears have been trained to recognize, the familiar

do, re, mi fa, sol, la, ti, do

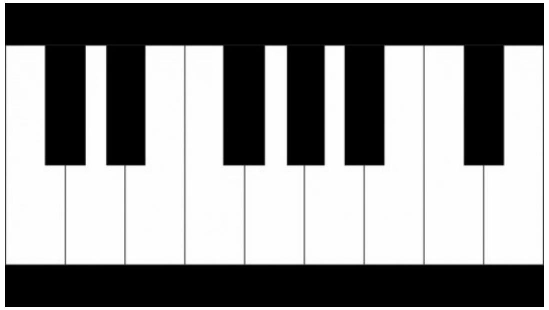
- The successive notes in the diatonic scale are separated by the following number of semitones: 2 2 1 2 2 2 1



- A diatonic scale can be started on any initial note, as long as it follows the pattern 2 2 1 2 2 2 1.

Keys

- A key in music has a home position (the starting note). It is represented by the number of sharps or flats (essentially, the black keys on a piano) that must be used to get the pattern of semitone steps 2 2 1 2 2 2 1 for the diachronic scale.



- You can play a song in one key and then in another, and it will sound "the same." The only difference in the way the melody and harmonics will sound is how "high" or "low" they are.

Lesson Three: Digitizing Sound for a Computer

- **Main concept:** Computers can't use lists of numbers that are infinitely long or numbers that have infinitely fine precision; thus, sine functions representing sound must be discretized – that is, we must represent the sine function of a single frequency sound as a finite list of digital numbers (base 2).

Binary – Base 2

- Computers operate on numbers that are represented in binary – that is, base 2.
- To convert from binary number *b* to decimal value *d* where *b[i]* represents the binary digit at position *i*:

$$d = \sum_{i=0}^{n-1} b[i] * 2^i$$

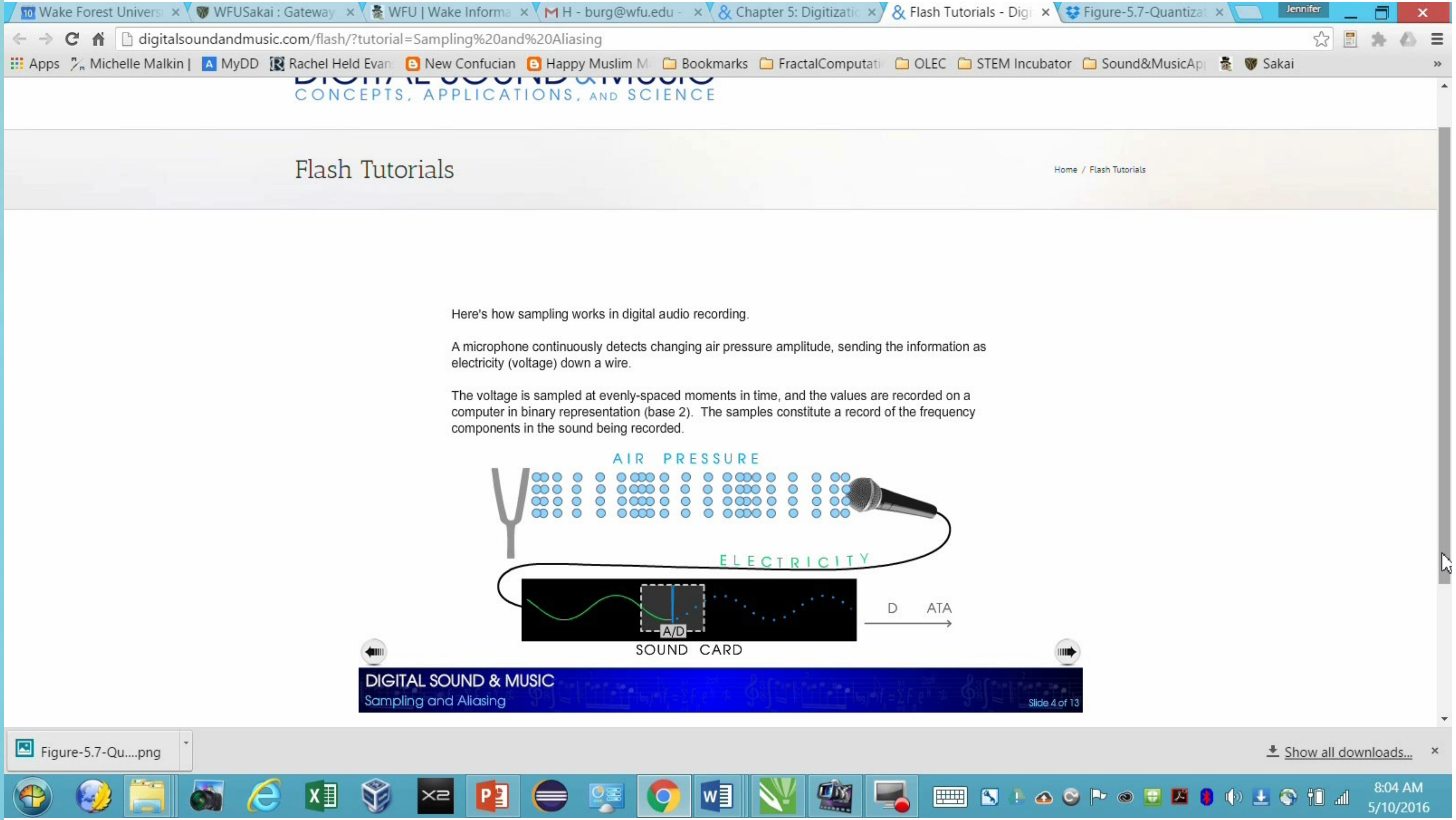
b[7]	b[6]	b[5]	b[4]	b[3]	b[2]	b[1]	b[0]
1	0	1	1	0	0	1	1
$2^7=128$	$2^6=64$	$2^5=32$	$2^4=16$	$2^3=8$	$2^2=4$	$2^1=2$	$2^0=1$

$1\,0\,1\,1\,0\,0\,1\,1_2 = 128 + 0 + 32 + 16 + 0 + 0 + 2 + 1 = 179_{10}$

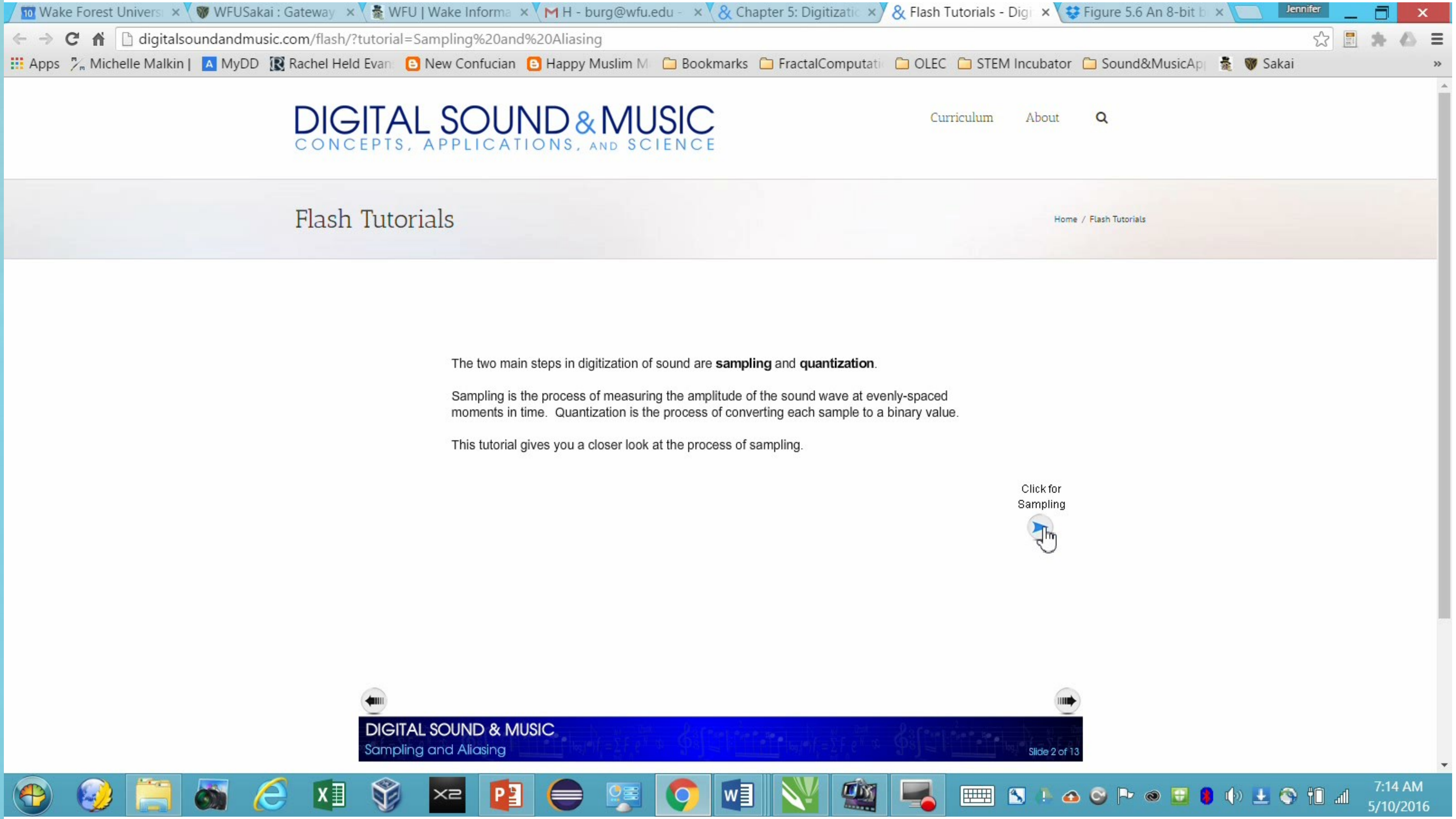
Digitization

- *Digitization* is the process of representing a continuous function as a discrete function of base 2 numbers.
- For single-frequency sounds represented as sine functions, this involves two steps:
  - sampling – that is taking the value of the sine function at evenly-spaced points in time
  - quantization – that is, representing each value in base two in a fixed number of bits

Sampling and Quantization



Sampling and Quantization



Sampling

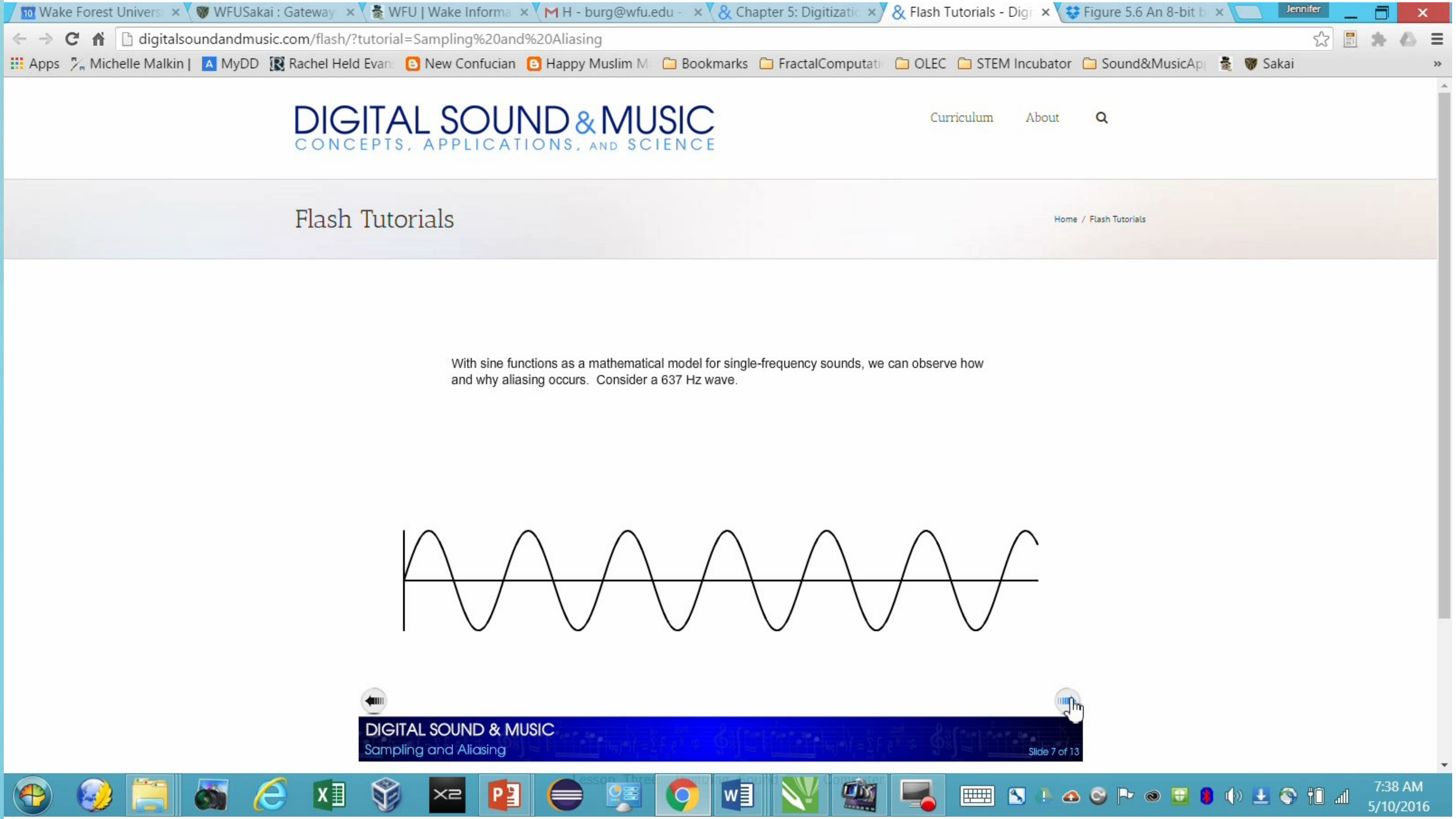
- The minimum sampling rate required in order to properly represent the sound is twice the frequency of a single-frequency sound (or twice the frequency of the highest frequency component for a complex sound). This is called the *Nyquist theorem*.

  minimum required sampling rate *r*:

$$r = 2 * f$$

- If the sampling rate is too low, aliasing occurs. That is, the recorded sound sounds lower than it really is.

Aliasing



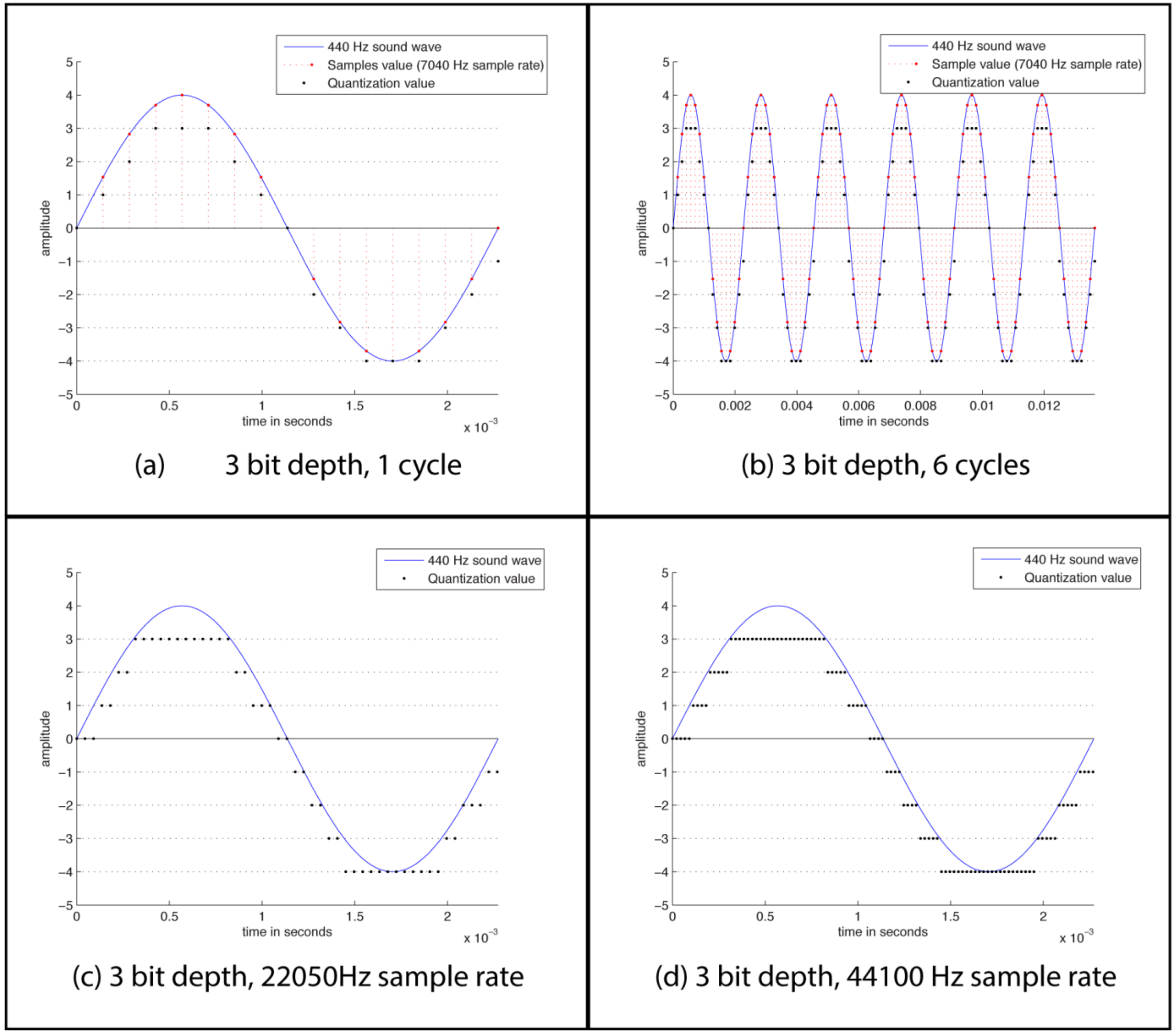
Link to full online tutorial: Sampling and Aliasing

Bit Depth

- The bit depth is the number of bits used for each sample.
- The bit depth determines the accuracy with which each sample is represented.
- If you have b bits, you have 2^b different sampling levels. You have to round each sample to one of those level. The rounding results in quantization error.
- 3 bits yields $2^3 = 8$ sampling levels

0	0	0
0	0	1
0	1	0
0	1	1
1	0	0
1	0	1
1	1	0
1	1	1

Quantization Error Resulting from Rounding



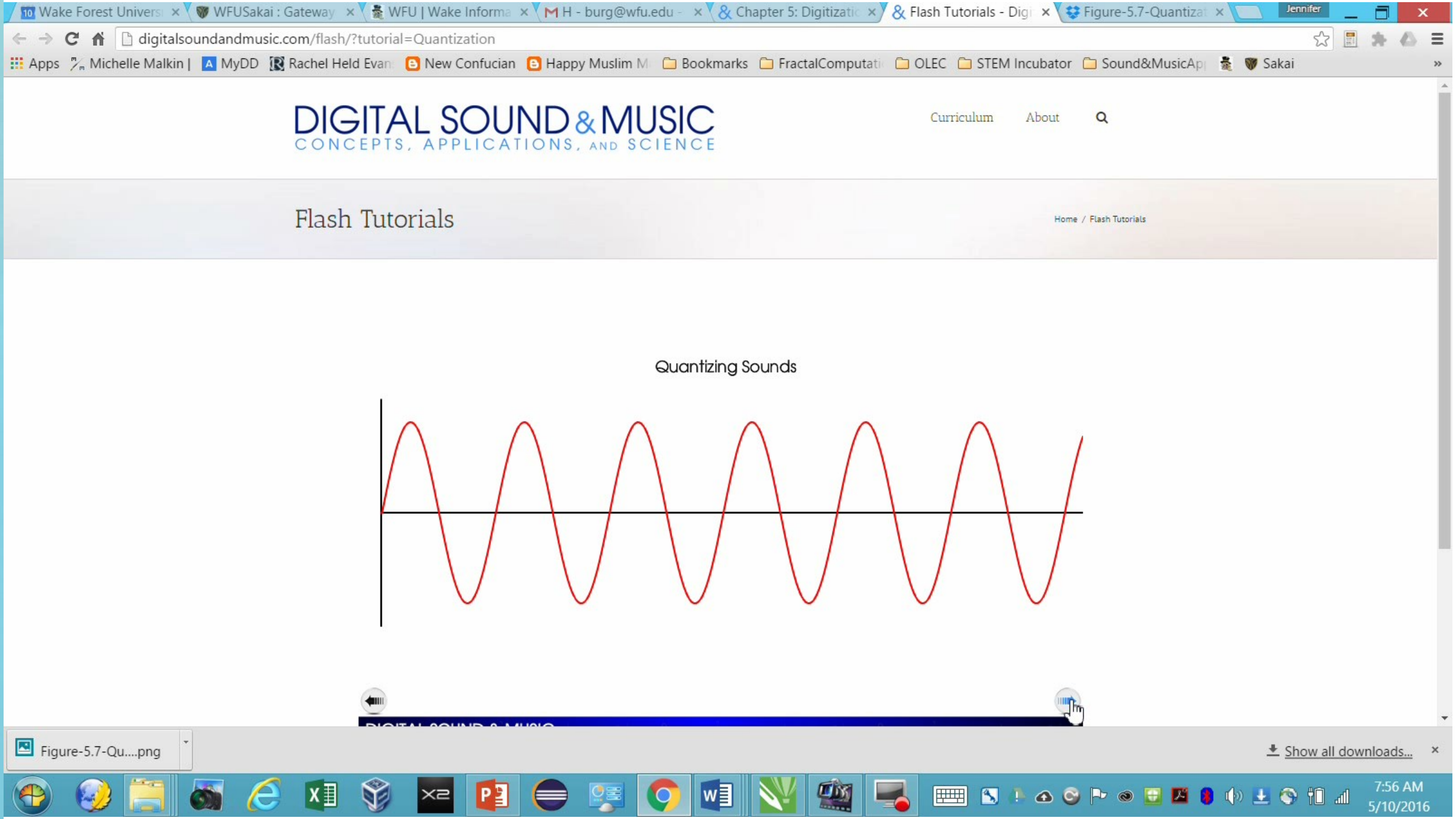
(a) 3 bit depth, 1 cycle

(b) 3 bit depth, 6 cycles

(c) 3 bit depth, 22050Hz sample rate

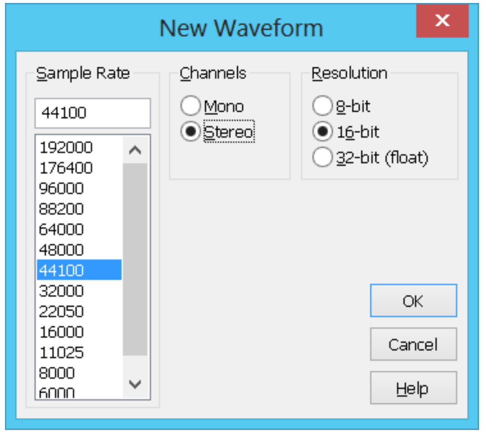
(d) 3 bit depth, 44100 Hz sample rate

Quantization Error Resulting from Rounding



"CD-Quality" Sampling Rate and Bit Depth

- When you record a song in audio processing software, you're asked to choose the sampling rate and bit depth.



- What is called "CD-quality" sampling rate of 44,100 samples per second, sometimes denoted as 44.1 kHz. CD-quality bit depth is 16 bits (i.e., 2 bytes) per sample, in stereo (two channels).

Signal-to-Noise Ratio and Dynamic Range

- The bit depth determines the signal-to-noise ratio and the dynamic range. In fact, these two things are mathematically the same.
- Put mathematical formula here.
- Show picture.
- The dynamic range can be understood intuitively as the difference between the loudest and softest parts you can represent in a piece of music you record.
- When you solve the formula above, you get a dynamic range of *** for a bit depth of b. For example,

Audio Files

- Given a sampling rate of 44.1 kHz and bit depth of 2 bytes per sample with two channels (stereo), the size of a 5 minute uncompressed sound file would be:
- Put mathematical formula here.
- You'll learn here how to create your own uncompressed sound files. You can save them as WAV files (for a PC, file names ending in .wav) or AIF files (for a Mac, files names ending in .aif or .aiff).
- Sound files are often compressed – for example, MP3 files.

Lesson Four: Levels of Abstraction in Sound Processing

- **Main concept:** You can work with sound a various levels of abstraction from
  - a *high level of abstraction* (where the details of what the computer is doing are hidden from you)
  - to a *low level of abstraction* (where you are creating a manipulating sounds "by hand" in programs that you write yourself).

Writing Programs in the Programming Language called C

- You can create sounds by evaluating sine functions and saving the resulting values in a file.
- You can read in already existing sound files and do things with them like filter out frequencies or change the amplitude.

Writing Programs in MATLAB or Octave

- Octave is a freeware version of MATLAB.
- You can do the same things that you can do in C, only the programming language is a little easier.
- Working with list of sound samples (called *arrays* or *vectors*)