

## Supplement to Chapter 1 of *The Science of Digital Media* – Digital Data Representation and Communication

### Worksheet – Sound as a Waveform<sup>1</sup>

---

#### Modeling environment: MATLAB and some sound editing program such as Audacity, Audition, Sound Forge, or Logic

##### Introduction:

Sound is produced by the vibrations of air molecules that cause air pressure to rise and fall. This phenomenon can be modeled by sinusoidal functions, whose values rise and fall in periodic cycles at fixed frequencies. Examples of sinusoidal functions are  $y = \sin(x)$ ,  $y = 3.5\sin(440x)$ ,  $y = \cos(2x)$ , and so forth. A pure tone can be modeled by a perfectly regular sinusoidal function – regular in that it has just one frequency component, and this frequency does not change. A more complex sound can be modeled by a sum of simple sinusoidal functions, each of which is a frequency component.

A digitized sound can be modeled by a discretized version of a sinusoidal function. By "discretized," we mean that the function is a finite array of discrete values representing points sampled at regular intervals along a continuous sinusoidal function. To help you visualize this, this exercise asks you to create a discretized sinusoidal function, listen to it, and graph it in MATLAB.

##### Exercise 1

In MATLAB, create a vector  $x$  of 44,100 values that are evenly-spaced between 0 and 1. These values will be the positions at which you sample a sine wave function over time. (A vector is just an array – a list of values.)

**Hint:** Use the *linspace* function.

##### Exercise 2

Still in MATLAB, create a sine function that has a frequency of 637 Hz (i.e., 637 cycles/s). Evaluate this function at the points in  $x$ , storing the results in vector  $y$ . Vector  $y$  represents a digitized sound – a pure tone at 637 Hz.

**Hint:** The  $y = \sin(x)$  function has one cycle every  $2\pi$  units. You want 637 cycles every 1 unit, a unit representing a second.

##### Exercise 3

Play  $y$  in MATLAB. (You should be able to play it as a sound file.)

---

<sup>1</sup>This material is based on work supported by the National Science Foundation under Grant No. DUE-0340969. This worksheet was written by Dr. Jennifer Burg (burg@wfu.edu).

**Hint:** Use *sound(y, s)* where *y* is the vector of sample values and *s* is the number of samples per second.

**Exercise 4**

Open a sound editing program like Audacity, Audition, Sound Forge, or Logic. Generate a pure tone that is one second long and has a frequency of 637 Hz.

Play the tone.

Does it sound the same as the tone you generated in MATLAB?

**Exercise 5**

In MATLAB, plot the sound wave that you created, displaying just a few cycles.

**Hints:**

Use *plot*.