Music 270a: Matlab Tutorial 2

Tamara Smyth, trsmyth@ucsd.edu
Department of Music,
University of California, San Diego (UCSD)

January 8, 2013

Creating functions

In addition to making scripts, we can also create functions. If they are stored in a location that is in the Matlab path, they can be used as a toolkit to help limit repeated code used for common tasks.

Matlab functions are also created in an m-file, however the file is initiated with the following

```
function [output_parameters] = <function_name>(input_parameters}
```

The $< functionname >$ with an ".m" extension must be the name of the m-file. To create a function that adds two numbers for example, your function may be called "myadd", and your m-file would be called "myadd.m" and would contain the following

```
function y = myadd(a, b)

% MYADD -- adds two numbers a and b
%
% See also SUM.

y = a+b;
```

Loops and Conditional Statements

As in all programming languages, Matlab can be used to repeat statements for a specific number of times—or until a condition has been reached—by using the `for` or `while` loops.

```
N = 10;
```

```
for n = 1:N
  for m = 1:N
    a(n,m) = 1/(n+m+1);
  end
end
```

Conditional operators like `if` and `switch` are also available. Call a `help` on these operators for more information.

Creating a sinusoid

Let's say we want to create a sinusoid of the form

$$x(t) = A\cos(\omega t + \phi); \tag{1}$$

Our first task is to discretize the continuous time variable t. To do this, we must first determine a sampling rate. Recall this sampling rate has to be twice the frequency of our signal frequency. So let's first set $f_0 = 2$. Therefore, our sampling rate must be at least 4, but let's make it a little higher, say $f_s = 8$. So this means that our sampling period T_s is the inverse and the time between samples is $T_s = 1/f_s$. Once we set a duration, say `dur=1`, we have everything we need to create a sinusoid.

```
fs = 44100;
dur = 1;
f0 = 2;

nT = 0:1/fs:dur-1/fs;
x = cos(2*pi*f0*nT);
```

Plotting

If we want to plot our resulting vector `x`, we can execute the command `plot(x)`. This isn't very meaningful however, because it doesn't give us any time information—it just plots the samples from `0:length(x)`. An alternative command would be to plot the sinusoid with respect to time using the command `plot(nTs, x)` where `nTs` is the discretized time variable, with a sampling period of `Ts`.

It is also possible to plot two things on the same figure. For example, if we want to see our sinusoid as a continuous line, we would plot as above (and Matlab will interpolate between the points). We may, in addition, want to see where the samples lie, in which case we can plot in the following way:

```
plot(nTs, x, nTs, x, 'o', 'MarkerFaceColor', black, 'MarkerEdgeColor', black);
```

Therefore you can use the same plot function to plot multiple things, including additional functions (for example, if you want to see two different sinusoids at the same time).

Creating and reading `.wav` files

Refer to Matlab's documentation for the functions `wavwrite` and `waveread` by typing

```
>> help wavread
```

and

```
>> help wavwrite
```

at the Matlab command line.

For .wav files, specify 16 bits and a sampling rate of 44100 Hz.