Machine Learning

Prof. Jose L. Mendoza-Cortes

Scientific Computing Department, Dirac Science Building
Materials Science and Engineering, High Performance Materials Institute
Florida State University
jmendozacortes@fsu.edu

Condensed Matter Theory, National High Magnetic Field Laboratory
Florida State University
mendoza@magnet.fsu.edu

Chemical and Biomedical Engineering
Florida State University — Florida A&M University — College of Engineering
mendoza@eng.famu.fsu.edu

Web: http://mendoza.eng.fsu.edu/

Review on vectorization: Example

Code 1: Form 1

```
i = 0;
for t = 0:0.02:10
    i = i+1;
    y(i) = cos (t);
end
t = 0:0.02:10;
plot(t,y)
```

Code 2: Form 2

```
close; clc;
t=0:0.02:10;
y = cos(t);

plot(t,y)
```

Review on Anonymous Functions

```
% Formal definition for ano... function
% fhandle = @(arglist) expression
clear; clc;
var1 = @(x,y) x^2 + y^2
%Create
%function [out] = namefunc(input)
var1(3,2);
a=4; b=2;
var2 = @(x) a*x^b
```

Notice how we can define two variables in the anonymous function and compare this to the regular way we declare explicit function (where another file is created)

Review on 'For' Loops: Example 1

```
%Definition
% for index = start:step:finish
%       statements
% end
for i = 1:0.1:2
    disp(i);
end

for j = 10:-1:1
    disp(j)
end
```

The loop can 'increase' in negative increments or positive increments. The second example of the code above explore the negative increment.

Review on 'For' Loops: Example 2

```
function [ factor ] = Myfactorial( N )
%Summary of this function goes here
% Myfactorial (N)
factor = 1;
    for k=1:N
        factor = factor*k
    end
% x = 1;
%     for k=1:N
%         x = x*k;
%     end
% factor = x;
end
```


Review on 'While' Loops: Typical While

```
% while condition
%    statements
% end
w = 8
while w > 0
    w = w - 5; disp(w)
end
```

Notice the way this is written `while condition`. This is the simplest way to do a while loop. Remember that this type of loop will repeat indefinitely if not used with care.

Review on 'While' Loops: Alternative While

```
% while (1)
%    statements
%    if condition, break, end
%    statements
% end
w = 8
while (1)
    %if w < 0, break, end
    w = w - 5;
    if w < 0, break, end
    disp(w)
end
```

Warning

This type of `while(1)` loop will not stop until there is a condition to break it. Thus there must be a `break` command inside the loop.

Using 'Pause' command

```
for n = 3:7
    mesh(magic(n))
    axis([0 7 0 7 0 50])
    tic
    pause(3)
    toc
end
```

In class we tried different variation of `pause` versus `pause(3)`. Remember that if no number is used in parenthesis, the code will stop indefinitely while if a number is place there, it will wait that number of seconds.

Using 'Animation'

```
% create an animation w/standard plot
% for j=1:n
%   plot command
%   M(j) = getframe;
% end
% movie(M)
for n = 3:7
    mesh(magic(n))
    axis([0 7 0 7 0 50])
    %tic
    pause(2)
    %toc
    M(n-2) = getframe;
end
```


Using 'Animation'(continuation)

```
%movie (m,n,fps)
% var m = the vector holding the
%         sequence of frames
% var n = an opt, tells you how many
% times the movie is repeated dftl=1
% var fps = frame rate (dftl= 12 fps)
movie(M)
```

Exercise

In class we tried different variation for animation, can you reproduce the ones that we tried in class?

See you next class

"Just as there is not royal road to geometry, there is no royal road to programming".- Euclid and J. V. Guttag

The computer will do what you TELL them to do NOT what you WANT them to do.- Someone in the Internet (Perhaps)

Think twice, code once.- Anonymous

The sooner you start to code, the longer the program will take.- R. Carlson

Any fool can write code that a computer can understand. Good programmers write code that humans can understand.- M. Fowler

Simplicity is the soul of efficiency.- A. Freeman

If you cannot grok the overall structure of a program while taking a shower, you are not ready to code it.- R. Pattis

Appendix: Scripts included

Code 3: SumIniFin.m

```
function [theArray theSum] = SumIniFin(start,final)
count = 1; theSum = 0;
for i=start:final
theSum = theSum + i;
theArray(count) = i;
count = count + 1;
end
end
```

Try these commands in your own workstation, i.e. have the lectures on one half side of your screen and Matlab/Octave-GUI on the other half.

Check the scripts/functions under the directory for this note number (X): /NX_Notes_directory