ENG 101: Inlab #8 -- Winter 2005

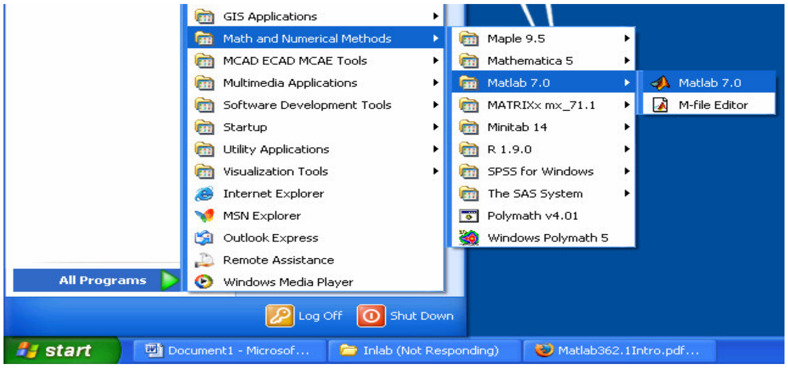
You are expected to turn this in to your lab instructor at the start of your next lab period. It is worth approximately 0.5% of your course grade. Recall that you will be allowed to drop two in-lab assignments. The **last page** of this write-up must be ***stapled*** to the front of the printout you hand in for this assignment. The assignment is graded out of 30 points. As always, start your answer sheet with your name and unique name.

The purpose of this in-lab is to become familiar with Matlab. You will be answering a series of questions and writing a short program.

I. Getting Started:

A. You may work in either Linux or Windows.

1. Windows:



2. Linux:

a) Open a terminal

b) At prompt type>> matlab &

B. Working Directory

1. Within Matlab, type>> pwd

It should be set to automatically to your afs space.
If not,

- For WINDOWS users: in Matlab type>> cd H:\eng101
- For LINUX users: in Matlab type>> cd ~\eng101

The MATLAB interface is a set of interacting windows. In some of these you "talk" to MATLAB, and in others MATLAB "talks" to you. Windows can be closed (click on the × button) or detached to become free-floating (curved arrow button).

To restore the original layout, use **View/Desktop Layout/Default** in the main menu bar.

The most important windows are **Workspace, Current Directory** and **Command History**.

II. Interactive calculations

The Command window is where you enter commands for MATLAB to execute *interactively*, meaning that the command is executed and the result is displayed as soon as you hit the Enter key.

For example, at the command prompt >>, type in 2+2 and hit Enter;
You will see **>>2+2**

```
           ans=
                4
```

Now type in**>> 2+2;**

Question 1: What has just happened? What is the difference between these two commands?

The variable "a" has been created, and assigned the value 4. By default, a variable in MATLAB is a matrix (a rectangular array of numbers); in this case "a" is a matrix of size 1×1 (one row, one column), which acts just like a number.

Variable names must begin with a letter, and followed by up to 30 letters, numbers, or underscore characters. **Matlab is case sensitive**:

`Abc` and `abc`

are not the same variable.

MATLAB uses +, -, *, /, and ^ for addition, subtraction, multiplication, division and exponentiation, respectively.

For example enter

>>x=5;y=2;z1=x*y,z2=x/y,z3=x^y

and you should see

```
z1 =

   10


z2 =

   2.5000


z3 =

   25
```

Even though x and y were not displayed, MATLAB "remembers" that values have been assigned to them. Type **`>>x,y`**

> **Question 2: What are the values displayed and how do you delete what is stored only in these variables? (try "help clear")**

Variables defined in a session are displayed in the Workspace window. Click on the tab to activate it and then double-click on x to launch a window summarizing z1's properties and entries. Since z1 is a 1×1 matrix, there's only one value.

MATLAB also has many **built-in functions** that operate on variables, such as:

```
abs(x)                      absolute value
cos(x),sin(x),tan(x)       cosine,sine,tangent of angle x in
radians
exp(x)                      exponential function
log(x)                      natural(base-e)logarithm
log10(x)                    common(base-10)logarithm
sqrt(x)                     squareroot
```

Question 3: Solve the following using Matlab.

1. $\frac{2^5}{2^5-1}$ and compare it with $\left(1-\frac{1}{2^5}\right)^{-1}$

2. $\sin(\pi/6), \cos^2(\pi/8)$

3. $\frac{2^5}{2^5-1}+4\sin(\pi/6)$

For **Fun** you may also want to try the following set of commands:

```
 >>syms x, syms y
>> x=exp(2*y)
>>diff(x,y)
>>int(x,y)
>>z=int(x,y)
>>pretty(1/(1+x))
```

III. Data Analysis Using Vectors

To analyze these data in MATLAB, first enter them into vectors:

```
>>R=[0 1.33  1.25 0.01 0.98 1.1 0.11 0.9 1.08 0.85] ;
>>D=[0.298 0.119 -0.24 -0.35 -0.82 -0.94 -0.116 -0.116 -0.161 -0.166];
```

You can either type these values in yourself or copy and paste them into Matlab.

A **vector** is a list of numbers. The commands above entered R and D as ***row vectors***. Double-click on R and D in the Workspace window and you'll see that they are stored in MATLAB as 1×10 matrices, i.e. a matrix with a single row. To see a histogram of the road densities

```
>>hist(R)
```

opens a graphics window and displays the histogram.

Question 4: There are some other built-in statistics functions, for example `mean(R)` gets you the mean, `std(R)` returns the standard deviation. What is the mean and standard deviations of R and D?

Now to see how R and D are related,

```
>>plot(R,D,'+')
```

Question 5: What does the '+' do in the plot comand? What is the default color of the markers? How can the color of the + be changed to be cyan? (Try "help plot")

IV. Vectors and Arrays

Jumping straight into arrays; Let's create a 3×2 matrix of 1's with the command **>>x=ones(3,2)**

Question 6: By typing **>>x(:,2)=0;** and then looking at what X is using the Workspace window, what manipulation have I done to x? Now type>> **flipud(x)**, Has anything changed? Why or why not? Now type>> **fliplr(x)**, Has anything changed? What do these commands do?

A set of regularly spaced values can be constructed by x=start:increment:end . Go to File-New-M file and within it type

```
x=10:2:24
```

Question 7: What is the value of x? What array does x=2:6 create?

Now consider indexing into an array. Try typing x(2) and x([1 3]).

Question 8: What is the value of x([1 3])? Explain how it got those values. What does the command, y=x(2:2:6) do? Why were those values gotten?

V. Iteration ("Looping")

A. For-loops

Loops make it easy to do the same operation over and over again, for example: making Salmon population forecasts 1 year ahead, then 2 years ahead, then 3, updating the state of every neuron in a model network based on the inputs it received in the last time interval.

There are two kinds of loops in Matlab: **for** loops, and **while** loops. A **for** loop runs for a specified number of steps. These are written as

```
for x = vector
commands
end
```

Here's an example (Copy/Paste into a new m file called **loop.m** and save the file):

```
%initial population size
initsize=4;
%create matrix to hold results sizes and store initial size
popsize=zeros(10,1);
popsize(1)=initsize;
%calculate population size at times 2 through 10,print to screen
for n=2:10
  popsize(n)=popsize(n-1)*2;
  x=log(popsize(n));
  q=[num2str(n),' ',num2str(x)];
  disp(q)
end
```

Now run the m file by typing "loop" in the Matlab command prompt.
Notice that you can use a for loop over any array. (In this case the array was 2 3 4 5 6 7 8 9 10 because of the 2:10 command).

Question 9: What does the code do? What type of variable is **q**? What does num2str do?

B. While-loops

A while loop lets an iteration continue until some condition is satisfied. For example, we can solve a model until some variable reaches a threshold. The format is:

```
while expression
commands
end
```

The conditions controlling a while loop are built up from operators that compare two variables:

```
==    equal to
~=    not equal to
<     less than
<=    less than or equal to
>     greater than
>=    greater than or equal to
```

These operators return a 1 for true statements, and a zero for false. For example:

```
>> a=1; b=3; c=a<b; d=(a>b);
```

Question 10: What are the values of c and d?

C. Branching

Logical conditions also allow the rules for "what happens next" in a model to be affected by the current values of state variables. The if statement lets us do this; the basic format is

```
if condition;
  commands
elseif condition;
  other commands
else;
  other commands
end;
```

Notice that their only one end per if statement regardless of the number of elseif's!!!

Question 11: Write a short program using if statements and a while loop to sum the first 20 odd integers. The pseudo-code would be:

```
i=0, sum=0
while(i<20)
  sum=sum+i*2+1
print sum
```

THE END FOLKS!!!

ENG 101: Inlab #8 -- Winter 2005

Name:____.

Section #:____.