ATMO5332 MATLAB Primer (To at least get through HW#1)

To run MATLAB through the HPCC on the nocona cluster, do the following:

1) Make sure you have MobaXterm running (if using a PC)
2) Type “interactive -p nocona” to get a core on nocona
3) Load the matlab module by typing “module load matlab”
3) Type “matlab” to start matlab

Vectors and Matrices

Create a vector A: A = [1 2 3]

→This creates a 1 X 3 row vector (1, 2, 3)

Create a matrix B: B = [1 2 3; 4 5 6; 7 8 9]

→This creates a 3 X 3 matrix $\begin{pmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \end{pmatrix}$

MATLAB indices work row by column, so for matrix B the entry B(3,1) = 7 and for vector A the entry A(1,2)=2.

Initialize a vector like this: C = 1:1:10

→This creates a row vector with values increasing by 1 from 1 to 10.

You can also reference portions of a vector or matrix with a colon like this:

```
B(2,:) = 99
```

→This would set all values in the 2nd row of matrix B above to 99. Note the colon as a variable means “all entries”

If Statements

MATLAB allows things to happen conditionally, for example:

```
n=2
if (n == 2)
  Disp(‘MATLAB is soooo fun’)
end
```

In addition to the example above using the ‘equal to’ comparison, the following relational operators can also be used:

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

To test more than one thing, such as if n was both greater than 1 AND less than 5, do the following:

```
if  (n > 1)  &&  (n < 5)
  disp(‘MATLAB is soooo fun’)
end
```

The “or” logical operator (| |) can also be used in this context.

For Loops

Sometimes you may want to execute an operation on every item in a vector in sequence. To do this you would use a for loop:

```
a=1:1:10
for i=1:10
  a(i)=99
end
```

→This would modify the 10-entry vector a entry by entry so that all values were equal to 99.

Plotting

For row vectors of the same length, you can plot one against the other using the command:

```
a=1:1:10
b=1:1:10
plot(a,b,’-r’)
```

→This would plot a red line including the points (1,1), (2,2),…..,(10,10). To change the color of the line to blue, black, or green, change ‘-r’ to ‘-b’, ‘-k’, or ‘-g’ respectively. To plot two lines on the same plot in different colors, do this:

```
a=1:1:10
b=1:1:10
c=1:2:19
plot(a,b,’-r’,a,c,’-b’)
```

You can put a title, axis labels, a legend, and axis bounds, and axis tick marks on plots like this:

```
title('Pressure Errors','FontSize',[14]);
xlabel('Forecast Hour','FontSize',[14]);
ylabel('Mean Absolute Pressure Errors hPa','FontSize',[14]);
legend('plot1','plot2');
axis([0 12 0 20]);
set(gca,'XTick',[0:2:12]);
set(gca,'YTick',[0:4:20]);
```

→ This would give a title, axis labels, and legend to the plot, and would set the x-axis to go from 0 to 12 with labeled tick marks every 2, and the y-axis to go from 0 to 20 with labeled tick marks every 4.

<u>Movies</u>

MATLAB has a movie feature. To play movies, use the following code:

```
runs=3;              %% Sets # of times movie will run
fps=100;               %% Sets the frames/second speed of movie
Frames=moviein(vectorlength);    %% Initializes movie frames
h.Visible=’off’;
for i=1:vectorlength            %% Loops over all times
plot(x,y,’-r’);              %% Makes plot for single time
title(‘test movie’);           %% Makes title for plot
xlabel('x-axis', 'FontSize',[14]);  %% makes x-axis label
ylabel('y-axis', 'FontSize',[14]);  %% makes y-axis label
axis([ xbeg  xend  ybeg  yend]);          %% Sets axis bounds
Frames(:,i)=getframe;             %% Assigns plot to set of frames
end              %% ends loop
h.Visible=’on’;
movie(Frames,runs,fps);          %% Plays movie
```

Note: The semicolon at the end of the lines suppresses output to the screen.

<u>MATLAB programs</u>

Although you can execute MATAB commands on the command line, generally you want to execute many lines quickly. To do this you would use a text editor to bring up a file (named with an extension of “.m”), and write many lines of code to it. Then to execute all that code, just type “testcode” on the MATLAB command line (to execute code in MATLAB program “testcode.m”). It’s always a good idea to put the following two lines at the top of your code:

```
warning off all;
clear all;
```

Lastly, you can put percent signs in your code that will cause MATLAB to ignore that specific line, or the rest of the line. You would do this, for example, if you want comments in your code to help you remember what's going on. This would appear as follows:

```
%% This next line initializes a vector
A = [  2  4  5  ];
```