Introduction

We have provided copies of the Introduction to MATLAB located on the EWS website (www.ews.uiuc.edu). This document will be a good reference and it contains an extensive collection of the commands that you will need to implement in your programs. However, we understand that many of you are unfamiliar with MATLAB. Therefore, we have given a brief introduction to the programming language including some of the commands that might be more useful to you.

Working in UNIX

Most of the work you complete for this portion of CEE 361 will be done using the UNIX workstations in the EWS lab. To help you become familiar with UNIX, we have provided some basic commands that will be useful to you.

<table>
<thead>
<tr>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>pwd</td>
<td>gives your current location in the directory tree</td>
</tr>
<tr>
<td>ls</td>
<td>lists all the files in the directory you are currently located in</td>
</tr>
<tr>
<td>mkdir &lt;dir name&gt;</td>
<td>creates a directory named &lt;dir name&gt;</td>
</tr>
<tr>
<td>cd &lt;dir name&gt;</td>
<td>changes to the directory named &lt;dir name&gt;</td>
</tr>
<tr>
<td>cd ..</td>
<td>moves up one level in the directory tree</td>
</tr>
<tr>
<td>rm &lt;dir name&gt;</td>
<td>removes a directory named &lt;dir name&gt;</td>
</tr>
<tr>
<td>rm &lt;file name&gt;</td>
<td>removes a file named &lt;file name&gt;</td>
</tr>
<tr>
<td>mv &lt;file1&gt; &lt;file2&gt;</td>
<td>moves &lt;file1&gt; to &lt;file2&gt;</td>
</tr>
<tr>
<td>cp &lt;file1&gt; &lt;file2&gt;</td>
<td>creates a copy of &lt;file1&gt; named &lt;file2&gt;</td>
</tr>
<tr>
<td>cd</td>
<td>returns to your home directory</td>
</tr>
</tbody>
</table>

To call MATLAB, type `matlab` at the UNIX prompt. To run Adobe Acrobat in UNIX, first change your directory in your xterm to the directory where you want to run Adobe Acrobat. Once in that directory, type at the UNIX prompt `acroread &` to open the Adobe Acrobat reader. Once the reader starts, just go to "file" and select "open". After selecting "open," a list of pdf files in the directory will pop up. Select the correct file and hit the "open" button.

Besides MATLAB and Adobe Acrobat, you will also be using a text editor to complete the lab assignments. There are several editors that you can use on the UNIX platforms. There is pico which is simple to use and similar to pine. Also, there is emacs which is more like a regular word processor. To call an editor, type the editor’s name at the UNIX prompt. To open a specific file using a specific editor, type the name of the editor at the UNIX prompt followed by the name of the file including any file extensions.

The Basics

The MATLAB commands / operators: +, -, *, /, ^ correspond to addition, subtraction, multiplication, division and exponentiation. Some common built-in functions like sine, cosine, natural log, absolute value and exponential are `sin()`, `cos()`, `log()`, `abs()`, `exp()` respectively. Each time an operator is used, a floating point operation or flop occurs.
Matlab counts how many flops that you have used in the current session. To access the number of flops at a given instant, type **flops** at the Matlab prompt.

**Matrices**

\[
A = \begin{bmatrix}
1 & 5 \\
4 & 2
\end{bmatrix} \quad \Rightarrow A = [1 \ 5; \ 4 \ 2]
\]
\[
a = \begin{bmatrix}
17 & -7
\end{bmatrix} \quad \Rightarrow a = [17 \ -7]
\]
\[
b = \begin{bmatrix}
3 \\
-3
\end{bmatrix} \quad \Rightarrow b = [3; \ -3]
\]
\[
B = \begin{bmatrix}
1 & 4 \\
0 & 2
\end{bmatrix} \quad \Rightarrow B = [1 \ 4; \ 0 \ 2]
\]

Once a matrix exists, the matrix elements are accessed by specifying the corresponding row and column indices. Thus, \(A(i,j)\) in MATLAB refers to the element \(a_{ij}\) of matrix \(A\), i.e. the element in the \(i^{th}\) row and \(j^{th}\) column. \(A(2,1)\) is equal to 4 in the above example. You can also change individual elements of a matrix after a matrix has been defined. For example, entering \(A(2,1)=9\) will change \(A\) to

\[
A = \begin{bmatrix}
1 & 5 \\
9 & 2
\end{bmatrix}
\]

**Matrix and Array Operators**

1. \(C = A + B\) or \(C = A - B\) is valid if \(A\) and \(B\) are the same size
2. \(C = A \times B\) is valid if \(A\)'s # of columns equal \(B\)'s # of rows
3. \(x = A\backslash b\) solves the system \([A]\{x\} = \{b\}\) (generally use this)

**Matrix Properties**

1. eigenvalues of a matrix \(A\) \(\Rightarrow\) \text{eig}(A)
2. determinate of \(A\) \(\Rightarrow\) \text{det}(A)
3. transpose of \(A\) \(\Rightarrow\) \(A'\)
4. inverse of \(A\) \(\Rightarrow\) inv(A)

**Input-Output**

MATLAB supports interactive computation, i.e. the taking of input from the screen and then sending the output to the screen. In addition, MATLAB can read input files and write output files. Some key notes:

1. **Data Type:** all data in MATLAB are treated as complex matrices. Real numbers and integers are just special cases. When a single real number is used, MATLAB sets the imaginary part to zero and then treats the number as an entry in a one by one matrix.
2. Dimensioning: automatically completed by MATLAB. The dimensions of an existing matrix C can be determined using the `size(C)` command at the command prompt. This returns a row vector where the first component is the number of rows and the second is the number of columns in C.

3. Case sensitivity: MATLAB is case-sensitive. Therefore a variable named Temp is different from a variable named temp.

4. Output: the output of every command is displayed on the screen unless MATLAB is directed to do otherwise. A semicolon (;) placed at the end of a line of code will suppress the output.

5. History: MATLAB saves all previously typed commands in a buffer. These commands can be recalled with the up-arrow key. Also, all variables defined at the command prompt are saved until the variable is redefined or the buffer is cleared. To see what variables are currently stored in the buffer, type `who`. To clear the buffer, type `clear` at the MATLAB prompt.

6. Diary: it is possible to create a text file that contains all the output displayed on the screen. At the MATLAB prompt, type `diary <file name>` where `<file name>` is the name of the text file. Anything displayed to screen after this command will be recorded in the diary file. To stop recording, type `diary off` at the MATLAB prompt.

### Functions

**Form of a function**

```matlab
function [output variables] = <file name>(input variables)
%
% ... description of the function ...
%
% dictionary of variables
% 1 ) input variables = ...
%
% 2 ) internal variables = ...
%
% 3 ) output variables = ....
%
% <MATLAB commands>
```

**Example**

```matlab
function [A,p] = rect_props(b,h)
%
% finds area and perimeter of a rect.
%
% dictionary of variables
% input variables
% b = base length of rectangle
% h = height of rectangle
% internal variables:
% not applicable
% output variables
% A = area of rect.
% p = perimeter of a rect.
%
A = b*h;
p = 2*(b+h);
```

Comment line: any code placed on a line after the "%" will be ignored by MATLAB

To call a function:

```matlab
>> b = 2; h = 3
>> [area,perimeter]=rect_props(b,h)  
or
>> [A,p]=rect_props(2,h)
```

**Loops**

A "for loop" is used to repeat a statement or a group of statements for a fixed number of times. By employing "for loops" into your code, the programming can be simplified.
immensely. A "for loop" has a counter "i" that starts with an initial value "m". The counter changes by an increment "k" after the completion of the MATLAB commands in the loop until the counter equals the final value "n". The last iteration is completed with the counter equal to "n" and then the loop ends.

<table>
<thead>
<tr>
<th>Format</th>
<th>Example</th>
</tr>
</thead>
<tbody>
<tr>
<td>for i=m:k:n</td>
<td>A=zeros(1,3)</td>
</tr>
<tr>
<td>&lt;MATLAB commands&gt;</td>
<td>for i=1:3</td>
</tr>
<tr>
<td>end</td>
<td>A(i)=2*i;</td>
</tr>
<tr>
<td></td>
<td>end</td>
</tr>
<tr>
<td>Creates a matrix of zeros with one row and three columns</td>
<td>This semicolon prevents the output from this calculation appearing on the screen</td>
</tr>
</tbody>
</table>

**Logic Statements**

There are six relational operations that you might need to complete the assignments in this course. They include:

< less than

<= less than or equal to

> greater than

>= greater than or equal to

== equal

~= not equal

**Printing**

This class will require an extensive amount of printing and therefore we recommend being careful to avoid exceeding your EWS print quota. You have only 300 pages per semester. To help you conserve paper; we recommend using mpage to print several pages of text on a single piece of paper. The format to use mpage is

    mpage -x -Pprinter <file name>

Replace x with the number of pages of text to print on a single page. Four is a good number. Replace printer with the printer name (celas1 if in the EWS lab in Newmark). Replace <file name> with the complete name of the file you want to print.

**Help**

For help on a MATLAB built-in functions and programming language constructs, use the commands help and lookfor. Typing help <MATLAB command> will provide any available help on that specific MATLAB command. Also, always check the Frequently Asked Questions portion of the class website (cee.ce.uiuc.edu/paulino/cee361).