Chapter 1

1. Getting Started With MATLAB

To run MATLAB, double-click on the MATLAB icon on the desktop. Three windows should open up, including the command window which will display the prompt “>>”. This is where you will be typing all of your commands.

There are several ways to get help with MATLAB.

- `>> demo`
  will open a new window that will allow you to access MATLAB tutorials.

- `>> help size`
  will help you with MATLAB’s size command. You can use `help` with any MATLAB command.

- Help desk. The help desk can be accessed from the help menu. If you have no prior experience with MATLAB, you should work through the help desk’s “Getting Started”. You can access that by going to the top of the MATLAB screen, and navigating to `help>MATLAB Help.` Click on the “Getting Stated” link in the right window.

2. Some Basic MATLAB Commands:

To assign the matrix $\begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \end{bmatrix}$ to the variable *A*, enter

```
>> A = [1 2 3; 4 5 6; 7 8 9]
```

(Make sure you put spaces between the numbers)

MATLAB should display

```
A =

     1     2     3
     4     5     6
     7     8     9
```

If you want to suppress the display after the assignment, add a semicolon to the end of your statement.

```
>> A = [1 2 3; 4 5 6; 7 8 9];
```

MATLAB will ignore everything typed after a '%'. This allows you to add comments to your work.
Here are some MATLAB commands followed by comments. Type these in to verify they do what I claim they are doing.

```
>> A(2,3) % displays the element of A that is in the second row, third column.

>> A(1,3) = 30 % Sets the element in the first row, third column to 30.

>> A(3,:)      % Displays the entire third row of A.

>> A(:,2)      % Displays the entire second column of A.

>> A(1,:) = [3 1 4] % Sets the first row of A to [3 1 4]

>> A(:,2) = 10*A(:,1) % Replaces the second column of A by 10 times the first column of A.
```

3. The `diary` command

MATLAB allows you to record everything you did in a session. Here's how it works

```
>> diary my_notes.txt on
>>                        %start typing here
>> diary off              %stops recording
>>                        %some things I don't want saved
>> diary on               %starts recording again
>> diary off              %I'm done
```

On my computer, the file is saved in C:\MATLAB6p5\work.

4. Elementary Row Operations [1.2][1]

Remember that there are three elementary row operations. After working through this section, you will know how to implement them all in MATLAB.

A) Multiply a row by a scalar. Here we multiply the first row by 10:

```
>> A(1,:) = 10*A(1,:)
```

[1] [1.2] means this section follows 1.2 in the text. You should read the text before you start this section.

B) Add a multiple of one row to another. Here we multiply the first row by 10 and add it to the second row:

```
>> A(2,:) = 10*A(1,:) + A(2,:)
```

C) Exchange two rows. Here we exchange the second and third rows.

```
>> temp = A(2,:);    %we need a temporary variable
>> A(2,:) = A(3,:); %set row2 = row3
>> A(3,:) = temp    %set row3 = old row2
```

Here is an example where we use Gauss-Jordan elimination to solve the system

$$x + 2y = 4$$
$$2x + y = 5$$

```
>>A = [1 2 4; 2 1 5]; %put the problem in matrix form
>>A(2,:) = -2*A(1,:) + A(2,:) %row2 = -2*row1 + row2

A =

     1     2     4
     0    -3    -3

>>A(2,:) = (-1/3)*A(2,:) %multiply the second row of A by -1/3

A =

     1     2     4
     0     1     1

>>A(1,:) = -2*A(2,:) + A(1,:) %row1 = -2*row2 + row1

A =

     1     0     2
     0     1     1
```

The matrix is now in reduced row echelon form and we see that the solution is $\begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} 2 \\ 1 \end{bmatrix}$

Problems

4a. 1.2 #11 (Do not use `>>rref`)
4b. 1.2 #35 (Do not use `>>rref`)

5. Symbolic Algebra [1.2]

MATLAB can also work with variables. Here is 1.3 #61 from the textbook:

61. Solve the linear system

$$\begin{vmatrix} & & y & + & z & = & a \\ x & & & + & z & = & b \\ x & + & y & & & = & c \end{vmatrix}$$

```
>> syms a b c;       %designate a, b, and c as symbolic variables
>> A = [ 0 1 1 a; 1 0 1 b; 1 1 0 c]; %set up the matrix
>> temp = A(1,:);    %get ready to switch row1 and row2
>> A(1,:) = A(2, :);%row1 is correct now
>> A(2,:) = temp     %row2 is correct

A =

[1, 0, 1, b]
[0, 1, 1, a]
[1, 1, 0, c]

>> A(3,:) = -1*A(1,:) + A(3,:); %row3 = -1*row1 + row3
>> A(3,:) = -1*A(2,:) + A(3,:)  %row3 = -1*row2 + row3

A =

[1,  0,   1,     b]
[0,  1,   1,     a]
[0,  0,  -2, c-b-a]

>> A(3,:) = (-1/2)*A(3,:);          %row3 = (-1/2)*row3
>> A(2,:) = -1*A(3,:) + A(2,:);     %row2 = -1*row3 + row2
>> A(1,:) = -1*A(3,:) + A(1,:);     %row1 = -1*row3 + row1
>> A

A =

[1,  0,   0,  1/2*c+1/2*b-1/2*a]
[0,  1,   0,  1/2*c-1/2*b+1/2*a]
[0,  0,   1, -1/2*c+1/2*b+1/2*a]
```

So the solution is $\begin{bmatrix} x \\ y \\ z \end{bmatrix} = \begin{bmatrix} -\frac{1}{2}a+\frac{1}{2}b+\frac{1}{2}c \\ \frac{1}{2}a-\frac{1}{2}b+\frac{1}{2}c \\ \frac{1}{2}a+\frac{1}{2}b-\frac{1}{2}c \end{bmatrix}$

MATLAB can also treat matrices symbolically, which can be used to force MATLAB to do rational arithmetic.

Type the following into MATLAB to see how this works.

```
>> A = [1/3  1/2  2/3;  1/8  3/4  2/3;  1/2  2  2/3]
>> As = sym(A)
>> 3*A
>> 3*As
```

Problem

5a. 1.1 #37 (Hint: Put the matrix into reduced row echelon form. What relationship must hold between *a, b* and *c*?)

6. Number of Solutions and Rank [1.3]

Recall from the text that a system of equations has no solutions, exactly one solution, or infinitely many solutions. Given a system of equations, the key to determine which of these three is the case is to look at the reduced row echelon form of the augmented matrix. Here is how to check:

- No solution: Look for a row that represents an inconsistent equation like $0=1$.
- Exactly one solution: Each column in the nonaugmented part of the matrix has a leading one.
- Infinitely many solutions: The system is consistent and some column in the nonaugmented part of the matrix does not have a leading 1.

> MATLAB trick: To find the number of leading ones in the row reduced echelon form of a matrix, use the `rank` command.

Problem

6a. One of the systems below is inconsistent, one has exactly one solution, and one has infinitely many solutions. Use MATLAB to determine which is which. Show your work.

$1x+2y+3z=0$	$1x+2y+3z=1$	$1x+2y+3z=2$
$4x+5y+6z=0$	$4x+5y+6z=3$	$4x+5y+6z=3$
$7x+8y+9z=0$	$7x+8y+8z=5$	$7x+8y+9z=5$

7. Matrix Algebra [1.3]

Rather than just being a convenient way of representing equations, matrices are also mathematical objects in their own right. As such, we can multiply a matrix by a scalar, and given two matrices of the right size, we can add, subtract and multiply them.

The MATLAB commands for doing this are exactly what you would expect. Here are two examples:

$$\begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \end{bmatrix} + \begin{bmatrix} 7 & 3 & 1 \\ 5 & 3 & -1 \end{bmatrix}$$

```
>> [1 2 3; 4 5 6] + [7 3 1; 5 3 -1]
```

$$3\begin{bmatrix} 2 & 1 \\ -1 & 3 \end{bmatrix}$$

```
>> 3*[2 1; -1 3]
```

You will learn more about the product of two matrices in the next chapter.

8. Linear Combinations [1.3]

In this class, we will often be using a given set of vectors as building blocks to build other vectors. We are allowed to do two things to build our new vectors: Multiply each of our original vectors by a scalar and add the scaled vectors.

Here is an example: We can build the vector $\begin{bmatrix} 7 \\ 3 \end{bmatrix}$ out of $\begin{bmatrix} 1 \\ 1 \end{bmatrix}$ and $\begin{bmatrix} 1 \\ -1 \end{bmatrix}$ as follows:

$$\begin{bmatrix} 7 \\ 3 \end{bmatrix} = 5\begin{bmatrix} 1 \\ 1 \end{bmatrix} + 2\begin{bmatrix} 1 \\ -1 \end{bmatrix}$$

We say that $\begin{bmatrix} 7 \\ 3 \end{bmatrix}$ is a *linear combination* of $\begin{bmatrix} 1 \\ 1 \end{bmatrix}$ and $\begin{bmatrix} 1 \\ -1 \end{bmatrix}$.

Problems:

8a. How would you use MATLAB to compute $3\begin{bmatrix}1\\2\end{bmatrix}-4\begin{bmatrix}3\\4\end{bmatrix}$?

8b. Express $\begin{bmatrix}7\\8\\9\end{bmatrix}$ as a linear combination of $\begin{bmatrix}1\\2\\3\end{bmatrix}$ and $\begin{bmatrix}4\\5\\6\end{bmatrix}$ (Hint: If you are stuck, you might want to read section 9 below.)

9. The Product $\vec{y} = A\vec{x}$ [1.3]

In your Elementary Algebra class you learned that solving a system of two equations in two variables could be interpreted as finding the point where two lines met. There is another useful picture for systems of equations using the idea of linear combinations.

Follow along with example 13 on page 33.

Consider the system of equations $\begin{matrix}3x + y = 7\\ x + 2y = 4\end{matrix}$

Another way to write this is $x\begin{bmatrix}3\\1\end{bmatrix}+y\begin{bmatrix}1\\2\end{bmatrix}=\begin{bmatrix}7\\4\end{bmatrix}$.

In other words, we are trying to write $\begin{bmatrix}7\\4\end{bmatrix}$ as a linear combination of $\begin{bmatrix}3\\1\end{bmatrix}$ and $\begin{bmatrix}1\\2\end{bmatrix}$. Our unknowns are the scalars we use in the linear combination.

Because the linear combinations come up so often in Linear Algebra, we define matrix multiplication so that $x\begin{bmatrix}3\\1\end{bmatrix}+y\begin{bmatrix}1\\2\end{bmatrix}=\begin{bmatrix}3 & 1\\1 & 2\end{bmatrix}\begin{bmatrix}x\\y\end{bmatrix}$.

This means that our system of equations $\begin{matrix}3x + y = 7\\ x + 2y = 4\end{matrix}$ can be rewritten as $\begin{bmatrix}3 & 1\\1 & 2\end{bmatrix}\begin{bmatrix}x\\y\end{bmatrix}=\begin{bmatrix}7\\4\end{bmatrix}$. This is called the *matrix form* of the system.

You will verify that MATLAB's definition of $A\,\bar{x}$ agrees with our definition in the next lab.