Homework 1, Due: Thursday, 1/17

This assignment is due on **Thursday, January 17**, by 11:59 PM. Your assignment should be well-organized, typed (or neatly written and scanned) and saved as a .pdf for submission on Canvas. You must show all of your work to receive full credit. For problems requiring the use of MATLAB code, remember to include your output in the main body of your solution .pdf and to also submit your .m-files on Canvas as a part of your completed assignment. Your code should be appropriately commented to receive full credit.

Problems

1 (4 points) Consider the following commands and corresponding output from the MATLAB command window. For each part, describe what the command (next to the >> symbol) does and interpret its output.

(a)
```
>> x = [3 7 -1 14]

x =

     3     7    -1    14
```

(b)
```
>> size(x)

ans =

     1     4
```

(c)
```
>> x(2)

ans =

     7
```

(d)
```
>> y = 1:0.2:2

y =

    1.0000    1.2000    1.4000    1.6000    1.8000    2.0000
```

2 For this problem, you'll be entering commands into MATLAB. You'll want to create an .m-file script to run your commands, and you'll also want to save your command window output (you can do this with a diary file or by copying the output into a Word or LaTeX document). Please see "sample_hw1prob2_script.m" posted on Canvas for an example of how to do this. You can download and edit this sample file, or start from scratch by clicking "New Script" on the Home tab of the MATLAB toolbar.

(a) (6 points) Enter the following matrices into MATLAB (for help, see the example in the "Intro to MATLAB" .pdf on Canvas):

$$\mathsf{A} = \begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix} \quad \text{and} \quad \mathsf{B} = \begin{bmatrix} 1 & 0 & 2 \\ 0 & 3 & 1 \end{bmatrix}$$

Enter the following commands, then display and discuss the output that each command produces:

(i) `A*B`

(ii) `B'`

(iii) `A(2,:)`

(iv) `inv(A)`

(v) `inv(B)`

(b) (4 points) To test the formatting options, enter `y = [4/3 1.2345e-6];` then enter each of the following commands, followed by the variable `y` each time to display the changes to the output: `format short`, `format long`, `format short e`, `format rat`. Comment on how each command changes the formatting of `y`. Before moving on, enter `format short` to get back to the default setting.

(c) (4 points) Enter the vector `x = linspace(1,6,10)`. What does this command produce? Now type the command `length(x)`. What does the `length` command tell you? Next, enter the following code:

```
for i = 1:length(x)
    z(i) = log(x(i));
end
```

This is called a "`for` loop". Enter `help for` in the command window to learn more about `for` statements. What does this particular `for` loop compute? Display the output vector `z`. (Note: This particular `for` loop is for demonstration purposes only. One of the tricks of coding in MATLAB is to figure out ways to get around using `for` loops when possible to speed up computations. Explore `help log` and see if you can figure out a way to compute `z` without using a loop.)

(d) (4 points) `if` statements are used to execute certain commands / operations based on conditional statements. The commands `if`, `else`, and `elseif` are used in defining such statements, and the logical operators `==`, `<`, `>`, `<=`, `>=`, and `~=` can be used in the conditions. Use the `help` to explore these commands. Write an `if` statement that computes the natural log of a number a if $a > 0$ and displays the message "`log(a) is undefined`" if $a \leq 0$.

(Hint: You'll want to make use of the `disp` command – take a look at `help disp` to see the help documentation.) Test your `if` statement for a few choices of a, e.g. $a = 2$ and $a = -1$, and show that it is displaying the correct output.

3 This aim of this problem is to practice entering a function and making a plot in MATLAB.

(a) (6 points) There are two ways to write a function in MATLAB: hard-coded directly into a script file (an in-line function), or written as a separate file that may be called within a script file. Code the function

$$f(x) = x^2 - 3x + 5$$

as both an in-line function using the following command:

```
f = @(x) x.^2 - 3*x + 5;
```

and as a separate function file using the following code:

```
function f = func(x)

f = x.^2 - 3*x +5;

end
```

Discuss these two different implementations of the same function, and show that they give the same output when $x = 3$. What is the purpose of using the `.^` command instead of just `^` when implementing the x^2 term?

(b) (4 points) Let `x = 0:0.1:10` and make a plot in MATLAB showing $f(x)$ vs. x using the following code:

```
plot(x,f(x),'-k','LineWidth',2);
```

Explore the `plot` command in the help documentation to see how to change the color and line-style of the plot, how to change the line width, and other options (like giving the plot a title, axis labels, and a legend). Save your plot and include it in your assignment write-up.

4 (4 points) Consider the real number $p = \pi$. Write the normalized floating point decimal form of p using both 5-digit chopping and 5-digit rounding.

Note: For any of the above problems for which you use MATLAB to help you solve, you must submit your code/.m-files as part of your work. Your code must run in order to receive full credit. If you include any plots, make sure that each has a title, axis labels, and readable font size, and include the final version of your plots as well as the code used to generate them.