Name: Andrew Brown	**Lab Time:** T 12:00
People Worked With: Cailin Moore	**Websites Used: n/a**
Time spent on zyBooks (hrs): 3	**Time spent on lab (hrs):2**
Submission Instructions Turn all work in to Lab 2 on Gradescope (PDF) and Canvas (.zip file), even if it is not complete yet. If you are not finished, complete the assignment outside of lab and re-submit to Lab 2 on Gradescope and Canvas. All labs are typically due at the same time on Monday every week, but check Canvas if in doubt.	

Learning objectives:
How do you make, edit, and use an array? How do you write an equation using variables that are arrays? How do you plot those variables? More practice on turning word problems into MATLAB code.

New MATLAB commands
These are highlighted in the instructions below.

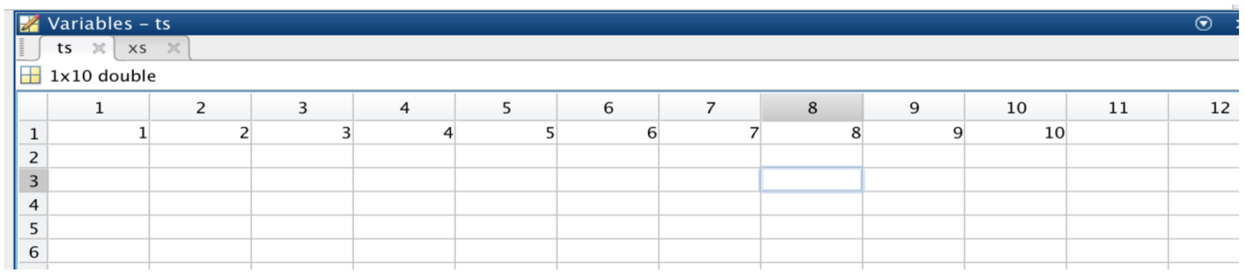
- `plot(x,y)` – plot x versus y
- `xlabel('string'), ylabel('string'), title('string')`
- `linspace( firstValue, lastValue, numElements )` – make an array
- `zeros( 1, n )` – create an array of n elements, all zeros
- `ones( 1, n )` – create an array of n elements, all ones
- `length(array)` – number of elements in an array
- `start:space:end or start:end` - colon operator
- `fprintf('formatted %3.2f printing\n', var);` - formatted printing to command window

Lab Problems

Getting Started

Try the following in the command window.

- Type `ts = 1:10`
  - What happens?
- Type `xs = ts * 0.1` – what do you get?
- Find `ts` in the variable window and double click it to open up the values window. Double click `xs` to get it to show up in the same window. Click back and forth between `ts` and `xs`. Notice:
  - There are as many values in `xs` as there were in `ts`
  - Each value in `xs` is 0.1 of the value in `ts`.
  - The numbers along the top – in this picture, 8 is highlighted. This is the *index*
  - Because we made it this way, `ts(index)` will return `index`. Try it with different values of `index`, e.g. `ts(8)`.
    - index starts at 1 and goes to 10. What happens if you do `ts(0)`? What about `ts(0.3)`? Are these valid indices?
  - Now do `xs(8)`. What is the relationship between `xs(8)` and `ts(8)`?
- Type `k = 1`. Now type `ts(k)` and `xs(k)`.
  - You can use variables as indices.
  - Change `k` to 3 and re-type `ts(k)` and `xs(k)`.
- Type `2:4`. What do you get?
- Now type `ts(2:4)` and `xs(2:4)`
  - What do you get?
- Try `ts(4:-1:2)` and `ts(3:end)` and `ts(:)` and `ts(3:2:9)` and `ts([7 3 0])`
  - What's happening: The colon operator creates an array, as does []. You can index an array with an array. end is a special keyword for the colon operator that only works when the colon operator is used to index an array.



1x10 double

	1	2	3	4	5	6	7	8	9	10	11	12
1	1	2	3	4	5	6	7	8	9	10		
2												
3												
4												
5												
6												

A few other things to try in the command window. Check the value of ts after each command and make sure you understood the answer. If you don't, ask a neighbor or a TA.

- Type `ts = [ -0.2 3 7 0.8]`

- Type `ts = zeros(1,10);`
- Type `ts = ones(1,10);`
- Type `ts = zeros(1,10) + 3;`
- Type `ts = linspace(0, 2 * pi, 8)`
- Type `min(ts), max(ts), mean(ts)`
- Type `length(ts)`
- Type `ts( 1:length(ts))`
    - This is exactly the same as `ts(1:end)` – why?
- Type `ts .* ts`
- Type `ts * ts`
    - This will generate an error. You need the `.*` to tell MATLAB to multiply each pair of elements.
        - Look-ahead: When you type `ts * ts` MATLAB tries to do a matrix multiplication. That's why you get the error that you do.

Problem 1

Repeat problem 2 from lab 1, only this time, use an array for x and calculate all three values at the same time. You will need three print statements in order to print out the three values.

If you are feeling fancy you can do a look ahead to a `for` loop and use that to do the print as well (extra credit). You should still do the original 3 print statements.

Step by Step Instructions:

- Copy your script from Lab 1 to your directory for lab 2 (rename it problem 1).
- Create x as an array with the three values from lab 1
    - x = [#1 #2 #3];
- You will need to make some of your operators be element-wise (e.g., ./ instead of /)
    - You do not need to for 3 * x, but you will need to for x^3
- If you get it wrong, MATLAB will spit out a red error message. Just click on it and look for a missing .
- To do the `fprintf`, remember that x(1) will return the first value of x, y(1) the first value of y, etc.
    - What happens if you just leave the fprintf as it was? How many times does it print out? Does it print x and y out in the right place?
    - This is a "feature" of fprintf – it will let you use arrays as values. There is a way to make this work correctly; in the meantime, when you see lots of fprintfs and you thought you should only have one, you probably passed an array instead of a single value.

Self-check:
```
For x= 10.200000, y is XX8.XXX8XX, z is XX.045578
For x= 12.800000, y is X7X.XXX970, z is XXX.702069
For x= 0.100000, y is X.XXX260, z is X.845962
```

Grading Criteria:
[30 pts] *Code is commented*
[10 pts] *x is created as an array*
[30 pts] y and z are correctly calculated from x using element-wise operators
[30 pts] Three print statements, one for each x
[10 pts extra credit] Do the print with a FOR loop

Answer script here:
```
%Andrew Brown Lab 2 Problem 1

clc
clear
```

```
%I am practicing using arrays with mathmatical calculations.

x=[10.2,12.8,0.1]; %Units are radians
y=((sqrt(2*x).*(4*x.^3)))./((4*x)+(7.^(x/10)));
z=log10((2*x)+5)+(((4*x)+exp(x))./((2/3)+(4*x.^2)));

%Practice using the "fprintf" function.
fprintf('For x = %0.6f radians, y is %0.6f, and z is %0.6f.\n', x(1), y(1),
z(1))
fprintf('For x = %0.6f radians, y is %0.6f, and z is %0.6f.\n', x(2), y(2),
z(2))
fprintf('For x = %0.6f radians, y is %0.6f, and z is %0.6f.\n', x(3), y(3),
z(3))


%Practice using a for loop to print different array values.
for i=1:3
    fprintf('For x = %0.6f radians, y is %0.6f, and z is %0.6f.\n', x(i),
y(i), z(i))
end
```

Command window output

```
For x = 0.100000 radians, y is 0.001260, and z is 2.845962.
For x = 10.200000 radians, y is 398.777844, and z is 66.045578.
For x = 12.800000 radians, y is 670.823970, and z is 553.702069.

For x = 0.100000 radians, y is 0.001260, and z is 2.845962.
For x = 10.200000 radians, y is 398.777844, and z is 66.045578.
For x = 12.800000 radians, y is 670.823970, and z is 553.702069.
```

Problem 2

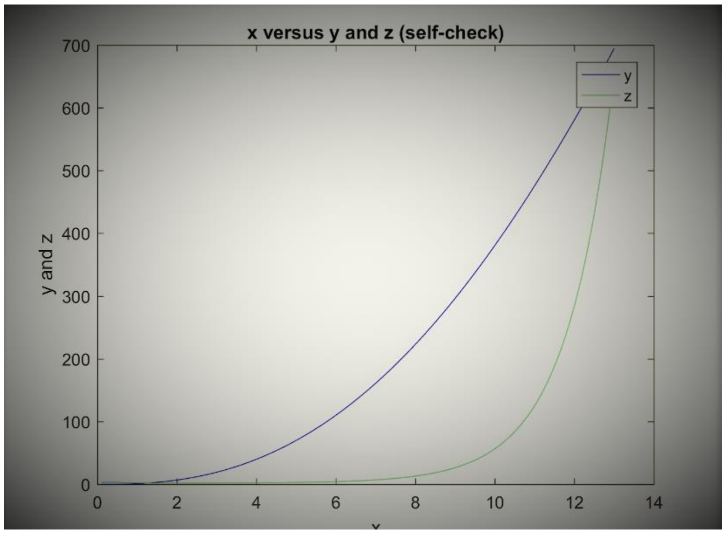
Plot x versus y (in blue) and x versus z (in green) on the same plot. x should go from 0.1 to 13, evenly spaced. Extra credit: Add a legend

---

Step by Step Instructions:

- *Copy your script from the first problem.*
- *Add a clf to the top of your script – this clears the figure as well*
- *You need evenly spaced values in x; which array creation function should you use to make x?*
- *Plot x versus y – remember that '-b' plots a blue line*
    - *This is my favorite link https://www.mathworks.com/help/matlab/ref/linespec.html?requestedDomain=true for looking up line styles*
- *Don't forget* `hold on`
- *Plot x versus z*
- *Add labels and title.*

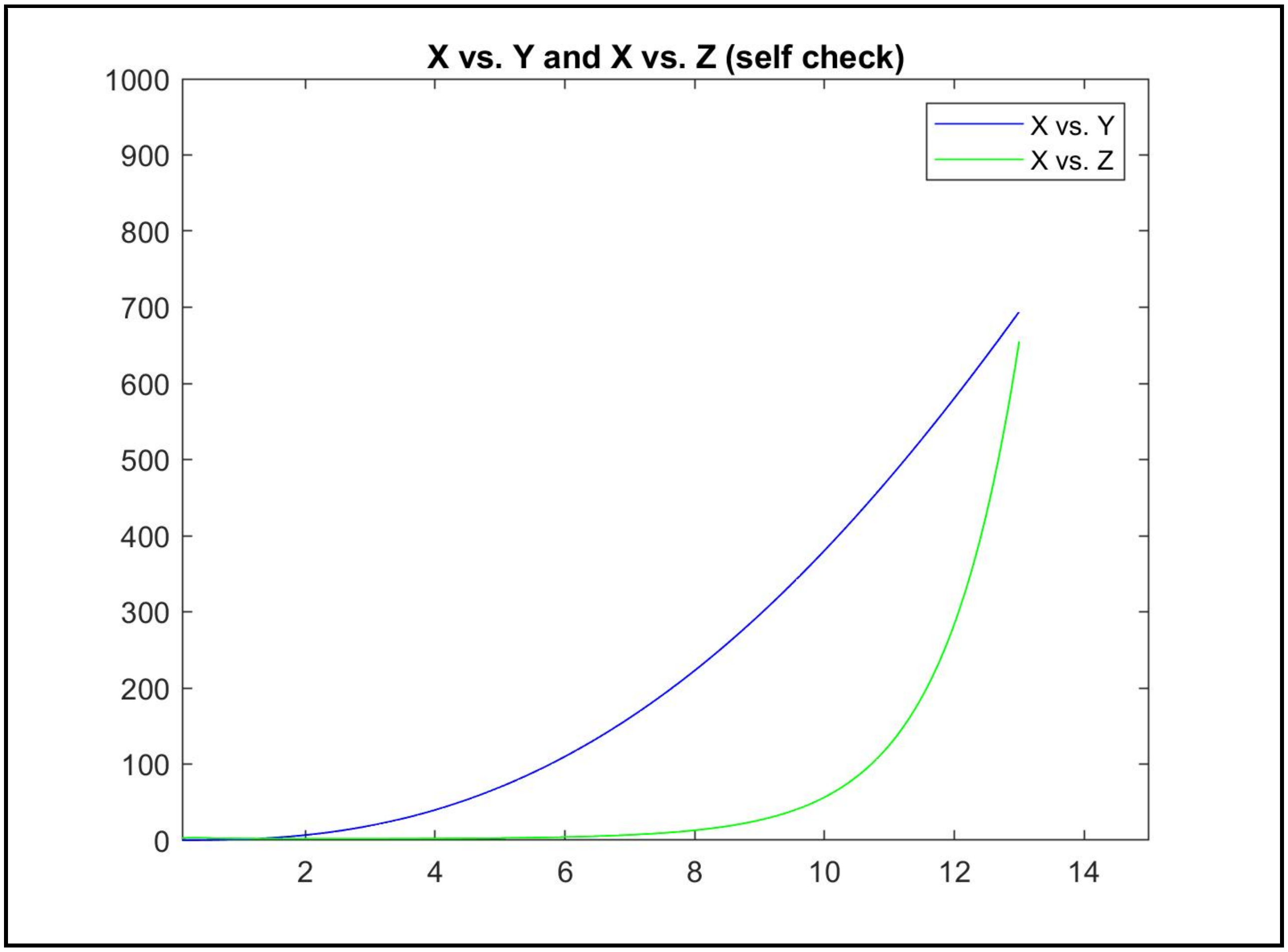
Self-check:



Grading Criteria: [30 pts] Commented code [20 pts] Creating x array [10 pts] Plotting x vs y with correct colors [20 pts] Plotting x vs z with correct colors *on same plot* [20 pts] Labeling and titling plot [10 pts Extra Credit] Adding a legend
Answer script here:

```
%Andrew Brown Lab 2 Problem 2

clc
clear
clf

%I am practicing using linspace with mathmatical calculations and plotting.

%Define variable x as many values between 0.1 and 13 to have a smooth graph
x=linspace(0.1,13,200); %Units are radians

%Practice with mathmatical equations.
y=((sqrt(2*x).*(4*x.^3)))./((4*x)+(7.^(x/10)));
z=log10((2*x)+5)+(((4*x)+exp(x))./((2/3)+(4*x.^2)));

%Practice using a for loop to print different array values.
for i=1:3 %Runs with integer values of i from 1 to 3.
    fprintf('For x = %0.6f radians, y is %0.6f, and z is %0.6f.\n', x(i),
y(i), z(i))
end

%Plottng X vs. Y and X vs. Z
plot(x,y,'b') %Plots x vs y and make the line blue
hold on %Keep the current figure visible
plot(x,z,'g')%plot x vs z and make the line green
legend('X vs. Y', 'X vs. Z') %add a legend
axis([0.1,15,0,1000]) %Define the axis limits
title('X vs. Y and X vs. Z (self check)') %titles the plot
```

Plot here:


X vs. Y and X vs. Z (self check)
1000
900
800
700
600
500
400
300
200
100
0
2
4
6
8
10
12
14
X vs. Y
X vs. Z

Problem 3

Make an array x = [10 13 12 14]. Now swap the 2nd and 3rd elements in the array so the numbers are in the right order; do not do this by making a new x! Edit the x array. Hint: You will need a temporary variable to accomplish this. Do not assume that you know the values in x; your code should if x was changed to [8 10 9 11] instead.

Step by Step Instructions:

- Do the following on the command line, until you figure out the answer, then copy it into a script
- Create x with four elements as given.
- Set the second element to be 35
  - Make sure you know how to set a specific element of an array
- Print out the third element to the command line
  - Make sure you know how to get the 3rd element
- Now set the second element to the third element
- Print out x to verify that the second element now has the value of the third
- Ok, that's half the problem… but… you now no longer have the second element around. Oops.
- Re-make x with the 4 elements.
- Save the second element in a variable (you can call it anything, tempSave maybe?)
- Set the second element to the third.
- Now – set the third element to the value that *use* to be in the second element
  - Where did you save it? Hint, see three lines above.
- Now do this in a script. Print out x at the start and the end (you can use `disp` to show x or just leave the semi-colon off)
- Repeat with x = [8 10 9 11 12]

Self-check:

```
Original

x =

    10    13    12    14

Swapped

x =

    10    12    13    14

>>
[Repeat for other x array]
```

Grading Criteria:
[20 pts] Comments
[50 pts] Values correctly swapped using a temporary variable
[30 pts] Showed that it worked for both arrays

Answer script here:

```
%Andrew Brown Lab 2 Problem 3

clc
clear

%I am practicing editing arrays.

%Defining x as a given array that is out of order.
x=[10, 13, 12, 14];

%Print and display the original array.
fprintf('Original\n\n')
disp(x)


i=x(2); %Use a temporary variable to hold the second value of the array.
x(2)=x(3);
x(3)=i; %Swap the second and third array values.

%Print and display the new array.
fprintf('Swapped\n\n')
disp(x)


%REPEAT!!


%Defining x as a given array that is out of order.
x=[8, 10, 9, 11, 12];

%Print and display the original array.
fprintf('Original\n\n')
disp(x)


i=x(2); %Use a temporary variable to hold the second value of the array.
x(2)=x(3);
x(3)=i; %Swap the second and third array values.

%Print and display the new array.
fprintf('Swapped\n\n')
disp(x)
```

Command window output

```
Original

   10   13   12   14

Swapped

   10   12   13   14

Original

    8   10    9   11   12

Swapped

    8    9   10   11   12
```

zyBooks Challenge Exercises

Do the challenge activities for the following in Week 2

1. Intro to arrays
   a. Construct an array
2. Row arrays
   a. Arithmetic array operations
   b. Indexing an array element
   c. [optional] Indexing the array: Moving values
3. Constructing row arrays
   a. Double colon operator: Counting up
   b. [optional] Double colon operator: Increment by x
   c. [optional] Double colon operator: Counting down
4. Functions to create row arrays
   a. Linear-spaced points array
5. More indexing
   a. Indexing the last element: Print queue
   b. [optional] Variable sized row arrays
   c. [optional] Indexing the array: Shift right with variable sized arrays
6. 1D element-wise arithmetic operators
   a. Element-wise division: Convert pounds to kilograms
   b. Multiple element-wise operations: Percentage change
7. Functions and 1D arrays
   a. [optional] Nth root
   b. Function definition: BMI calculator
8. Constructing strings
   a. Customized greeting
   b. [optional] Markup mania