Name: Andrew Brown	**Lab Time** *(ECampus write "ECampus")*: T 12:00

Names of people you worked with:
• Jake • Kevin Daellenbach

Websites you used:
• n/a

Approximately how many hours did it take you to complete this assignment (to nearest whole number)?	6

By writing or typing your name below you affirm that all of the work contained herein is your own, and was not copied or copied and altered.

Andrew Brown

Note: Failure to sign this page will result in a 50-point penalty. Failure to list people you worked with may result in no grade for this homework. Failure to fill out hours approximation will result in a 10-point penalty.

Turn .zip files to Canvas or your assignment will not be graded

Learning Objectives:

- More practice with for loops and if statements
- More practice with using functions as inputs to other functions
- More practice writing functions

Homework Guidelines:

1) Whenever you create a function, you must decide what goes into the function and what goes into the calling script. For example, do you pass gravity in as a variable or do you just set it in the function? **Document your choices.**

2) There is more than one right way to answer these homework problems. If you come up with a different answer than your fellow classmates, discuss the relative merits of each answer (is it clear? Is it more/less generalizable?)

Specific Coding Notes:

Debugging: Set a break point at the first line of your function file. This way, you can see what happens in the function file. First, check that the parameters that got passed in have the values you expect. Second, make sure the output variable(s) are set correctly.

Grading Checkpoints

Criteria	Component	No	Yes
[20%] Comments and Pseudocode	Declared units on all variables?		
	English description of problem at top?		
	Comments outlining your steps?		
[10%] Output formatting	Used fprintf() to make complete sentences (when required)?		
	Correct units on answers?		
	Correct number of decimal places?		
[70%] Functionality	Script computes correct value(s)?		
	Correctly converted units in script when needed?		

Problem 1

From elementary physics and Homework #1, we know that the distance a projectile travels when fired from a cannon depends on both the initial velocity v and the launch angle θ:

$$x(t) = v\,t\cos(\theta)$$
$$y(t) = v\,t\sin(\theta) - \frac{1}{2}gt^2$$

where x(t) and y(t) are the distances traveled in x and y, respectively, after time t has passed (g is gravity, g = 9.8 meters/second^2). Assume an initial velocity of 13.5 meters per second and an angle of 89 degrees above the ground. **Use fzero** to find the time and x location where the projectile hits the ground**. Print the time and location to 4 decimal places.**

Self-check: It lands really, really, close to the cannon after x.7xx7 seconds

Comments for grader/additional information (if any)

I used anonymous function for x(t) and y(t)

Script File

```
%Andrew Brown Homework 5 Problem 1

clc
clear

%More practice with functions and fzero.
%Projectile motion simulated

%Define constants
g=9.8; %acceleration due to gravity in m/s^2
theta=89; %launch angle in degrees
v=13.5; %initial velocity in m/s

%Declare given position functions
x = @(t) [v.*t.*cosd(theta)]; %x distance traveled
y = @(t) [v.*t.*sind(theta)-(1/2).*g.*t.^(2)]; %y distance traveled

%Plot and calculate time and location when the ball hits the ground
yZero=fzero(y,9.3); %find the time, t when the cannon ball hits the ground
xPos=x(yZero); %find the distance from the cannon the ball hits the ground

%Print out calculated values
fprintf('It lands reall, really close to the cannon (%0.4f m) after %0.4f
seconds\n',xPos,yZero) %Print out calculated values
```

Function Files

```
% Copy and paste your functions here (if any were used). Must be size 10,
same as MATLAB font and color.
```

Command Window Output

```
It lands really, really close to the cannon (0.6490 m) after 2.7547 seconds
>>
```

Problem 2

Find all of the intersections of two functions of x: $F_1(x) = 100e^{-x} - 1$ and $F_2(x) = \sin(\pi x)$ in the domain $2 \le x \le 10$.

a) **Plot both functions on the same graph. Draw X's where the functions intersect.**
b) **Print the x values for all of the intersections found.**

Use a **for** or **while** loop to automatically find all of the intersections in the given domain (as opposed to estimating them by hand). Before printing your answers, **filter out** all solutions that are outside of the domain given in the question.

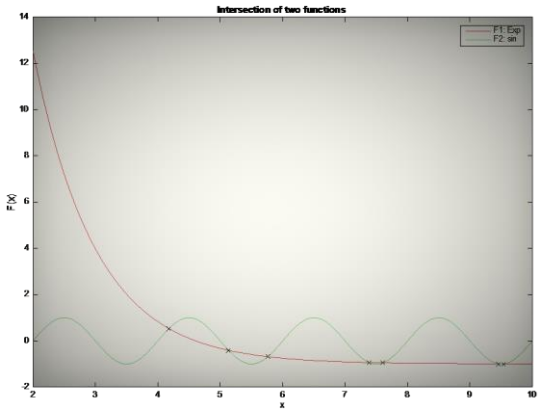
The numerical approximation that you're using in this problem (fzero) can cause headaches if you're not careful, so be sure to read the hints below.

Hint 1: How do you make an anonymous function that takes one variable (x) and returns zero when the two functions intersect?

Hint 2: You need to call fzero a bunch of times with a reasonable set of guesses, enough to make sure that you actually get all of the intersections. Each time you calculate a new intersection, compare it to ALL of the intersections that you have already calculated. If the difference between the new intersection and any of the old ones is very small ***(<0.00006)****, do not add it to your list and just move on to calculating the next one. Don't forget to check that the intersection is in the given domain.*

a) [+10 Extra credit] In hint #2, why should you use (currentRoot – oldRoots) < *0.00006* instead of (currentRoot – oldRoots) == 0? Write your answer in the Word document.
b) [+10 Extra credit] Even if you do everything correctly, fzero can return solutions far away from your guess. Why do you think this happens? Write your answer in the Word document.

Self-check: `First root is at 4.X7X, last is at 9.X3X`
(Note: Print ALL roots. This self-check only gives you the first and last.)



Comments for grader/additional information (if any)

Script File

```
%Andrew Brown Homework 5 Problem 2

clc
clear

%Find all the intersections of two functions in a given domain

%Anonymous functions of the given equations
F1=@(x) [100.*exp(-x)-1]; %function of the first given equation
F2=@(x) [sin(pi.*x)]; %function of the second given equation

%Plot the two functions
hold on
fplot(F1) %plot exp
fplot(F2) %plot sin
axis([2,10,-2,14]) %define axes

%Anonymous function that will equal zero where F1 and F2 intersect
FInt=@(x) [F1(x)-F2(x)]; %when == 0, F1=F2

%Calculate intersections
interval=2:0.1:10; %define common interval
FStore=ones(length(interval),1); %preallocate fStore
for j=1:length(interval) %count up by one for the length of the times I
calculate fzero
    i=interval(j); %to calculate fzero many times
    FZero=fzero(FInt,i); %calculate fzero
    FStore(j)=FZero; %store values of fzero
    xInt=uniquetol(FStore,0.00001); %calculate x values of unique
intrsections
end

%Plot the points of intersection on the graph
yInt=ones(length(xInt)); %preallocate yVals
for k=1:length(xInt) %loop through the number of unique values found
    yInt(k)=F2(xInt(k)); %calculate Y values of unique intersections
    plot(xInt(k),yInt(k),'xk') %plot the x vs y values of the unique
intersections
end
%Plotting stuff
title('Intersection of Two Functions') %title the plot
xlabel('x') %x label of the plot
ylabel('y') %y label of the plot
legend('F1: Exp', 'F2: sin') %legend of the plot

%Print out the x values of the unique intersections
fprintf('The equations intersect at the following x values:\n')
disp(xInt)
```

Function Files

```
% Copy and paste your functions here (if any were used). Must be size 10,
same as MATLAB font and color.
```

Command Window Output

```
The equations intersect at the following x values:
    4.1786
    5.1349
    5.7599
    7.3874
    7.6011
    9.4602
    9.5382

>>
```

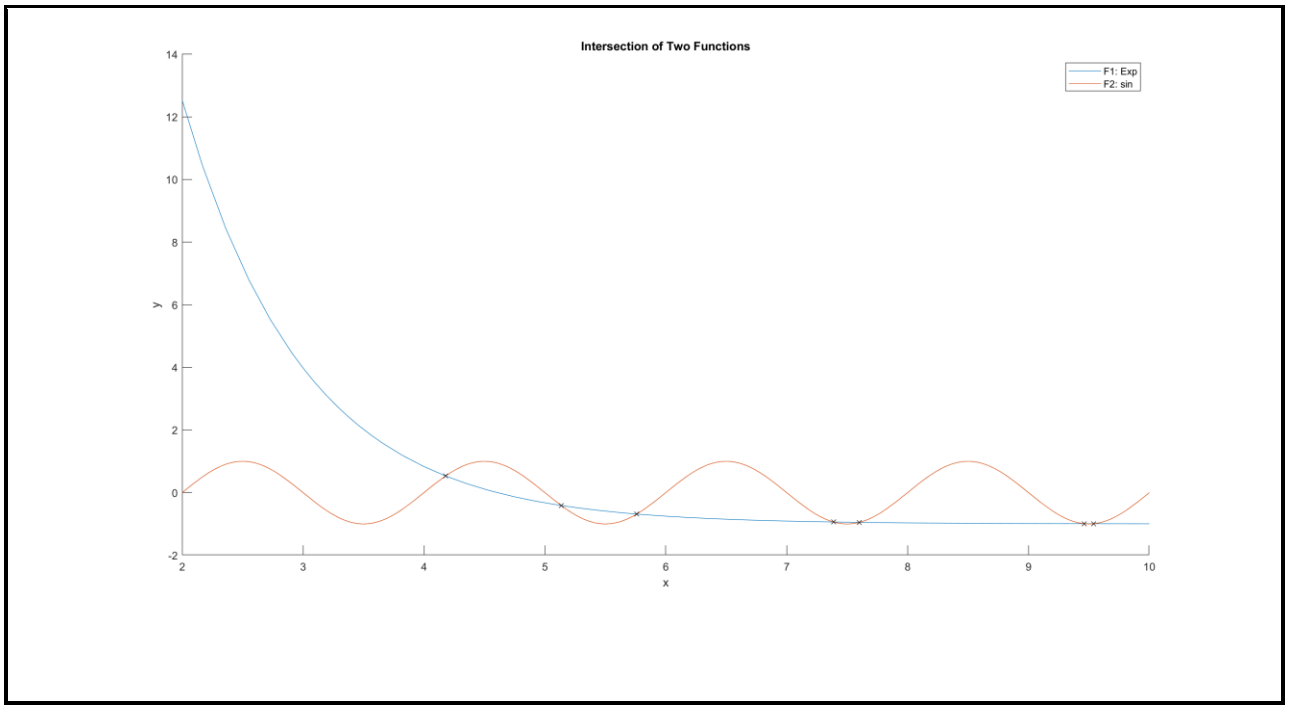
Answers to question(s) asked in the homework (if any)

Extra credit responses:

a) If some of the fzero terms were rounded by matlab, they may be ever so slightly off. Using the >0.0006 as opposed to ==0 allows for proper tolerance.

b) You made a bad guess

Image Output


Intersection of Two Functions
F1: Exp
F2: sin
y
x
14
12
10
8
6
4
2
0
-2
2
3
4
5
6
7
8
9
10

Problem 3

Epidemic Part 5

Comments for grader/additional information (if any)

Script File

```
% Andrew Brown Homework 5 Problem 3

clc
clear
close all

% Epidemic Part 5: Practicing plotting with fzero and functions while
% finding the maximum number of people ifected in an epidemic

%Initial Values
a=10; %the contact rate: the average # of people a person comes in contact
with.
b=1.25; %the amount of time in days that a person is infectious
S0=0:500:2000; %Susceptibles, those who have never had the illness and can
catch it.
I0=100; %Infectives, those who are infected and are contagious.
R0=0; % Recovered, those who already had the illness and are immune.
h=0.05; %timestep in days
nSteps=1:140; %timesteps

[S,I,R,N] = DisSimulate(nSteps);

[bIsEpidemic,peakIValue,peakTime]=IsEpidemic(S0,I0,a,b,nSteps,h);

%Print out ending values after 140 timesteps
for j=1:length(S0)
    fprintf('Start S=%0.0f, Ending: S = %0.0f, I = %0.0f, R = %0.0f,
total %0.0f\n',
S(1,j),S(length(nSteps),j),I(length(nSteps),j),R(length(nSteps),j),N(1,j))
    if bIsEpidemic(j)==1
        fprintf(' Is Epidemic, time %0.2f days number of
people %0.0f\n',peakTime(j),peakIValue(j))
    end
end

%Plot S, I, R, and Totals vs time
titleName = {'Susceptibles', 'Infected', 'Recovered', 'Total'};
plotColor = {'r', 'g', 'b', 'k', 'c'};
legendVar = {'Start S = 0','Start S = 500','Start S = 1000','Start S =
1500','Start S = 2000'}; %add a legend
for k=1:length(S0)
    %Plot S vs Time
    hold on %keep all on same plot
```

```
    subplot(1,4,1) %plot in first position
    plot(nSteps.*h,S(:, k),plotColor{k},'Linewidth',3) %plot s vs time
    title(strcat(titleName{1}, ' versus time') )%title the plot
    xlabel('Time (Days)') %X label for the plot
    ylabel('Number of People') %Y label for the plot
    axis([0,7,0,2200]) %Set proper axes
    if k==5
        legend('Start S = 0','Start S = 500','Start S = 1000','Start S =
1500','Start S = 2000') %add a legend
    end

    %Plot I vs Time
    hold on %keep all on same plot
    subplot(1,4,2) %plot in second position
    plot(nSteps.*h,I(:, k),plotColor{k},'Linewidth',3) %plot I vs time
    if bIsEpidemic(k)==1 %if there is an epidemic occuring
        plot(peakTime(k),peakIValue(k),'xk','Markersize',15) %plot the
maximum number of infected and the time
    end
    title(strcat(titleName{2}, ' versus time') )%title the plot
    xlabel('Time (Days)') %X label for the plot
    ylabel('Number of People') %Y label for the plot
    axis([0,7,0,2200]) %Set proper axes

    %Plot R vs Time
    hold on %keep all on same plot
    subplot(1,4,3) %plot in third position
    plot(nSteps.*h,R(:, k),plotColor{k},'Linewidth',3) %plot R vs time
    title(strcat(titleName{3}, ' versus time') )%title the plot
    xlabel('Time (Days)') %X label for the plot
    ylabel('Number of People') %Y label for the plot
    axis([0,7,0,2200]) %Set proper axes

    %Plot Total vs Time
    hold on %keep all on same plot
    subplot(1,4,4) %plot in fourth position
    plot((1:length(N))*h,N(:,k),plotColor{k},'Linewidth',3) %plot total vs
time
    title(strcat(titleName{4}, ' versus time') )%title the plot
    xlabel('Time (Days)') %X label for the plot
    ylabel('Number of People') %Y label for the plot
    axis([0,7,0,2200]) %Set proper axes
end
```

Function Files

```
function [bIsEpidemic,peakIValue,peakTime]=IsEpidemic(S0,I0,a,b,nSteps,h)
%ISEPIDEMIC: analyzes if there is an epidemic. If there is not do nothing,
%if there is, find the maximum amount of people infected and the time at
%which that happens

[S,I,R,N] = DisSimulate(nSteps); %Call the simmulated diseases

%Calculate maximum # of people sick and time if it is an epidemic
```

```
bIsEpidemic=zeros(1,length(S0)); %preallocate
peakTime=zeros(1,length(S0)); %preallocate
peakIValue=zeros(1,length(S0)); %preallocate
for k=1:length(S0) %run for each simulated epidemic
    if I(2,k)>I0 %if there is an epidemic (increasing number of sick people
over time)
        bIsEpidemic(k)=1; %make array of true/false values to show if there
is an epidemic
        peakIValue(k)=max(I(:,k)); %Find the maximum number of people
infected during each simulated epidemic
        peakTime(k)=find(I(:,k)==peakIValue(k))-1; %Find the I index at its
max value for each simulated epidemic
        peakTime(k)=nSteps(peakTime(k))*h; %Set the indices of I column and
nStep equal to find the time at the max value
    else
        bIsEpidemic(k)=0; %make array of true/false values to show if there
is an epidemic
    end
end
```

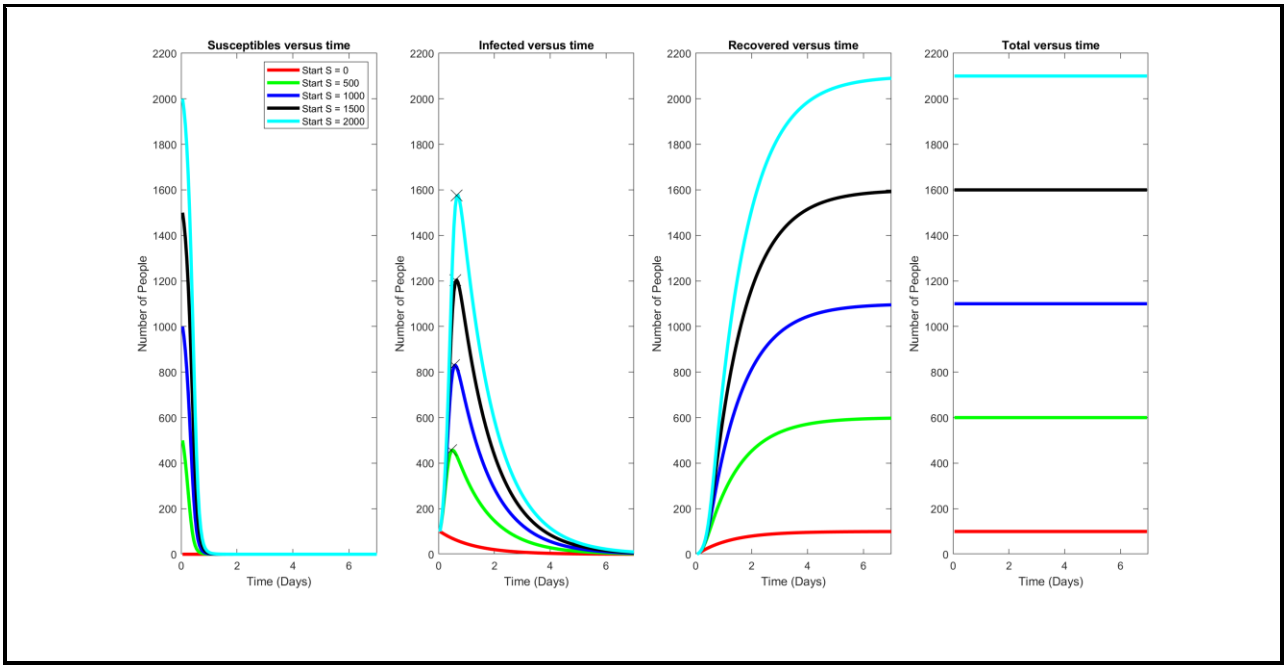
Command Window Output

```
Start S=0, Ending: S = 0, I = 0, R = 100, total 100
Start S=500, Ending: S = 0, I = 2, R = 598, total 600
 Is Epidemic, time 0.45 days number of people 457
Start S=1000, Ending: S = 0, I = 5, R = 1095, total 1100
 Is Epidemic, time 0.55 days number of people 830
Start S=1500, Ending: S = 0, I = 7, R = 1593, total 1600
 Is Epidemic, time 0.60 days number of people 1204
Start S=2000, Ending: S = 0, I = 10, R = 2090, total 2100
 Is Epidemic, time 0.65 days number of people 1576
>>
```

Answers to question(s) asked in the homework (if any)

3a) [200]

Image Output


Susceptibles versus time
Infected versus time
Recovered versus time
Total versus time
Start S = 0
Start S = 500
Start S = 1000
Start S = 1500
Start S = 2000
Number of People
Time (Days)

Problem 4

Extra Credit: Teacup Part 4 [Function File].

Comments for grader/additional information (if any)

Script File

```
% Copy and paste your script here. Must be size 10, same as MATLAB font and
color.
```

Function Files

```
% Copy and paste your functions here. Must be size 10, same as MATLAB font
and color.
```

Command Window Output

```
Copy and paste the command window output here (same font, size 10).
```

Answers to question(s) asked in the homework (if any)

(If any)

Image Output

Copy and paste images here

Problem 5

Extra Credit: Watch the video at this link: https://curiosity.com/videos/6174-numberphile-numberphile/. Then download the Matlab file Kaprekars_operation.m from Homework 5. Test this file with a few random four digit numbers. Note that this script won't work for values less than 1000.

a. Change this file to be a function file that would receive one input of four digits number and would output two values - Output 1: the number the operation reaches (should be Kaprekar's constant); Output 2: the number of iterations it took to reach that number. **You do not need to submit the function file, but you will need it for your own reference in part b.**
b. Write a script that tests the Kaprekar's theory for initial values between **1000** and **9998** except for the values 1111, 2222, 3333, …., 8888. Your script should print the following:
    i. Count of how many initial values were tested.
    ii. Count of how many values returned Kaprekar's number 6174.
    iii. How many of these values took 1 iteration to reach Kaprekar's number, along with how many took 2 iterations, 3 iterations, 4 iterations, 5 iterations, 6 iterations, 7 iterations, and more than 8 iterations.

Self-check: for (ii), all values should return 6174. For (iii), all values should take less than 8 iterations.

Comments for grader/additional information (if any)

Script File
`% Copy and paste your script here. Must be size 10, same as MATLAB font and color.`

Command Window Output
`Copy and paste the command window output here (same font, size 10).`

Image Output
Copy and paste images here