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

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

Websites you used:
•

Approximately how many hours did it take you to complete this assignment (to nearest whole number)?	12

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:

- Practice with loops and if statements.
- How to create a function file.
- How to create an anonymous function.
- How to use functions as inputs to other functions

Homework Guidelines:

1) Functions are a way to encapsulate functionality, and to reduce errors caused by copying code. From this homework on you should be using functions (either anonymous or function files) for writing equations, and **for** loops to avoid duplicating code.
2) 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.
3) **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?		
	Accomplished all functionality in the problem		

For 1) and 2), please write which homework problem it is in the comments box. You must re-do a **different** problem for 1) and 2).

Bonus [up to +10 each]: Add more complex functionality to the original problem by, for example, using a for loop, plotting it in a different way, searching for a minimum value, etc. Make sure you clearly document what changes you made.

Problem 1

Pick any one of the previous homework problems and re-do it with an anonymous function.

Comments for grader/additional information (if any)
Homework 2, Problem 3 Extra changes you made: I put the 4 calculations made with the anonymous function into a for loop, so I do not have to call the function 4 times. Only once. I also had to change from having 4 different arrays to having one 4-row matrix.

Script File

```
%Andrew Brown Homework 4 Problem 1

clc
clear
clf

%Practice creating and calling anonymous functions.

%Part A: What is the energy that is generated by a 10-meter fall for a 120
%pounds heavy object on the surface of Earth?

%Define given constants
massPounds=120; %mass in pounds
massKg=massPounds/2.246; %mass in kilograms
g=9.8; %acceleration due to gravity on earth
h=10; %height off the ground in meters

%Given equation
E=massKg*g*h; %energy generated from the fall in joules

%Print out the energy given off
fprintf('Energy from a 10m fall on Earth: %0.2f Joules\n', E)

%Part B: Choose three different planets of your choice, plot the energy
%generated by the fall of a 120 pounds object over the surface of each
%of Earth and the other planets. Your plot should show the energy
%generated between values between 0 to 50 meters.
%and
```

```
%Part C: Identify on your graph and command window the harmful heights
%that will generate the same energy as a 10-meter fall on earth.

%Define given constants
massPounds=120; %mass in pounds
massKg=massPounds/2.246; %mass in kilograms
g=[9.8, 26.0, 3.61, 3.75]; %accelerations due to gravity in m/s^2
h=linspace(0,50,100); %height off the ground in meters

%Given equation in anonymous function
EAnonymous=@(G) massKg.*(G).*h; %energy generated from the fall in joules

%Calculate the energy of falls on different planets
EStore=zeros(length(g),length(h)); %store values of the E calculation in a
matrix with each row being a different gravity
for i=1:length(g)
Ecalc=EAnonymous(g(i)); %call the anonymous function w/ gravity as the g
array
EStore(i, :)=Ecalc; %store the values of G each time through the loop
end

%Plotting the graph for energy generated from a fall on Earth
subplot(2,2,1)
plot(h,EStore(1, :)) %plot height vs energy
title('Self-Check: Energy Generated from a fall on Earth')
xlabel('Height (m)')
ylabel('Energy (J)')
legend('Energy')
axis([0,50,0,3*10^4])



%Plotting the graph for energy generated from a fall on Jupiter
subplot(2,2,2)
plot(h,EStore(2, :)) %plot height vs energy
title('Self-Check: Energy Generated from a fall on Jupiter')
xlabel('Height (m)')
ylabel('Energy (J)')
axis([0,50,0,7*10^4])

%Calculate and plot the point of a drop equivalent to a 10m drop on earth
hold on
JupiterHeight=E/(g(2)*massKg);
plot(JupiterHeight,E,'r*')
legend('Energy', 'Equivalent of 10m drop on Earth')



%Plotting the graph for energy generated from a fall on Mercury
subplot(2,2,3)
plot(h,EStore(3, :)) %plot height vs energy
title('Self-Check: Energy Generated from a fall on Mercury')
xlabel('Height (m)')
ylabel('Energy (J)')
axis([0,50,0,12*10^3])
```

```
%Calculate and plot the point of a drop equivalent to a 10m drop on earth
hold on
MercuryHeight=E/(g(3)*massKg);
plot(MercuryHeight,E,'r*')
legend('Energy', 'Equivalent of 10m drop on Earth')



%Plotting the graph for energy generated from a fall on Mars
subplot(2,2,4)
plot(h,EStore(4, :)) %plot height vs energy
title('Self-Check: Energy Generated from a fall on Mars')
xlabel('Height (m)')
ylabel('Energy (J)')
axis([0,50,0,12*10^3])

%Calculate and plot the point of a drop equivalent to a 10m drop on earth
hold on
MarsHeight=E/(g(4)*massKg);
plot(MarsHeight,E,'r*')
legend('Energy', 'Equivalent of 10m drop on Earth')

%Print out the equivalent falls.
fprintf('The energy-equivalent fall to a 10m drop on Earth:\n
Jupiter: %0.2fm\n Mercury: %0.2fm\n
Mars: %0.2fm\n',JupiterHeight,MercuryHeight,MarsHeight)
```

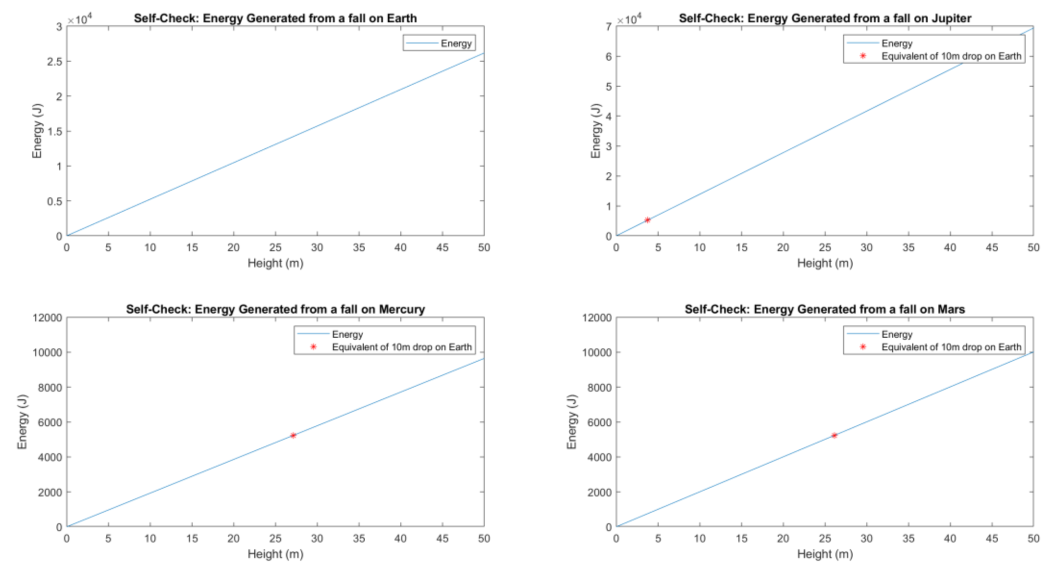
Function Files

Command Window Output

```
Energy from a 10m fall on Earth: 5235.98 Joules
The energy-equivalent fall to a 10m drop on Earth:
 Jupiter: 3.77m
 Mercury: 27.15m
 Mars: 26.13m
```

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

Image Output


Self-Check: Energy Generated from a fall on Earth
Energy
Energy (J)
Height (m)
Self-Check: Energy Generated from a fall on Jupiter
Energy
Equivalent of 10m drop on Earth
Energy (J)
Height (m)
Self-Check: Energy Generated from a fall on Mercury
Energy
Equivalent of 10m drop on Earth
Energy (J)
Height (m)
Self-Check: Energy Generated from a fall on Mars
Energy
Equivalent of 10m drop on Earth
Energy (J)
Height (m)

Problem 2

Pick any one of the previous homework problems (except Euler, teacup or Epidemic) and re-do it with a function declared in a function file.

If you have an equation or problem from another class that you'd like to plot/calculate instead, please feel free. Make sure you explain what that equation is and what the parameters are.

Must include:

- One script file
- At least one function file
- Fully defined problem statement
- Pseudo code
- Comments

What to turn in:

- Fully Explained Problem Statement (describe exactly what problem you're solving)
- Pseudo code
- Script and Function(s)
- All Outputs (and Inputs, if any)

Comments for grader/additional information (if any)
Describe where problem is from: HW 2 Problem 2 - Script was far too long and did not make use of any if statements or for loops for the calculations or plotting. I fixed this by doing the calculations and plotting in for loops using matrices and putting the calculations into a function called "Roses" to calculate 6 different roses given inputs that define the number of petals.

Script File

```
% Andrew Brown Homework 4 Problem 2

clc
clear
clf

%Practice using the subplot and polar coordinate operators to make polar
%roses.

%Define given variables that change number of petals in the roses
n=5:7;
d=[1,7,5];

%Call the roses function that creates 3 polar roses and 3 cartesian roses
[x,y,r,interval,theta,t] = Roses(n,d);
```

```
%Plot all of the polar and cartesian roses using their output variables
for j=1:length(n)
    subplot(2,length(n),j) %plot the polar plots in the top row
    polarplot(theta(j,:),r(j,:),'b') %plot all 3 polar plots
    t1=strcat('Polar Rose for n=', num2str(n(j)), 'and
d=',num2str(d(j))); %Concatanate strings for the title and turn numbers into
strings
    title(t1) %title all 3 polar plots
    subplot(2,length(n),j+length(n)) %plot the cartesian plots in the bottom
row
    plot(x(j,:),y(j,:),'b') %plots all 3 cartesian plots
    t2=strcat('Polar Rose on Cartesian Coordinates for
n=',num2str(n(j)),'and d=',num2str(d(j)));%Concatanate strings for the title
and turn numbers into strings
    title(t2) %title all 3 cartesian plots
end
```

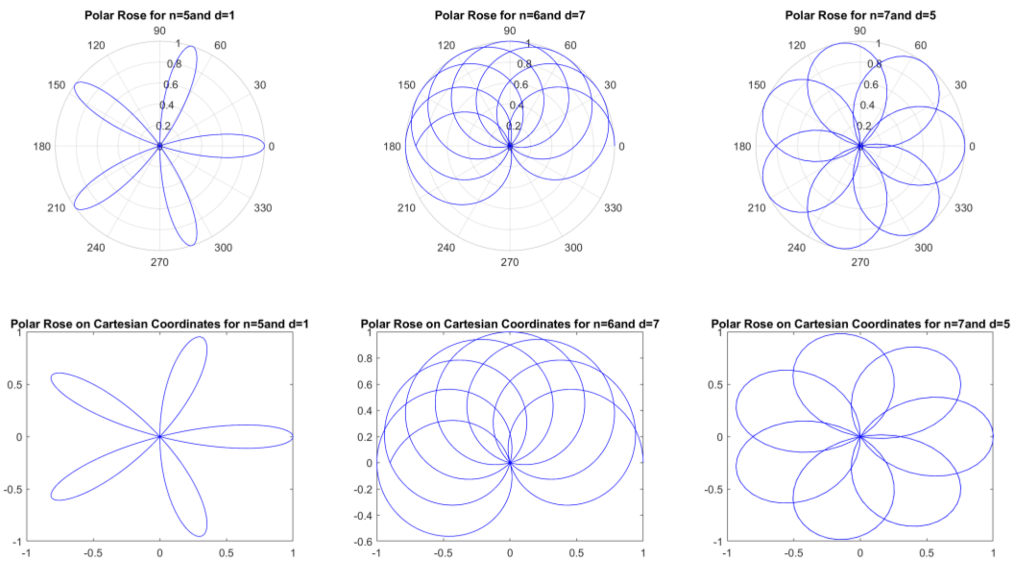
Function Files

```
function [x,y,r,interval,theta,t] = Roses(n,d)
%ROSES:
%   Change the number of petals on 6 different polar roses

%Constants to create 6 different kinds of polar roses
interval=500; %plots 500 points so we have a smooth plot
theta=[linspace(0,pi,interval);linspace(0,8*pi,interval);linspace(0,12*pi,int
erval)]; %define theta values in a matrix
t=[linspace(0,pi,interval);linspace(0,8*pi,interval);linspace(0,12*pi,interva
l)]; %define t values in a matrix

%Express given equations.
x=zeros(length(t),interval); %Preallocate x matrix
y=zeros(length(t),interval); %Preallocate y matrix
r=zeros(length(theta),interval); %Preallocate z matrix
for i=1:length(n)
    x0=cos((n(i)/d(i)).*t(i,:)).*cos(t(i,:)); %Cartesian coodinates
    y0=cos((n(i)/d(i)).*t(i,:)).*sin(t(i,:)); %Cartesian coodinates
    r0=cos((n(i)/d(i)).*theta(i,:)); %Polar coodinates
    x(i,:)=x0; %Store all 3 x arrays in a matrix
    y(i,:)=y0; %Store all 3 y arrays in a matrix
    r(i,:)=r0; %Store all 3 r arrays in a matrix
end
end
```

Command Window Output

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

Image Output



Problem 3

There are a variety of methods for calculating pi. These all involve computing an infinite series, so you can't calculate the exact value of pi, but the more terms you compute, the better the approximation. Below are two different ways of calculating pi:

$$\pi = \frac{4}{1} - \frac{4}{3} + \frac{4}{5} - \frac{4}{7} + \cdots$$

$$\pi = 3 + \frac{4}{2*3*4} - \frac{4}{4*5*6} + \frac{4}{6*7*8} - \frac{4}{8*9*10} + \cdots$$

a) **Calculate pi using the first equation.** Stop when your answer is within 1e-5 of MATLAB's value of pi. **Print out your value of pi and MATLAB's value of pi using 12 digits**. Also **print out the number of iterations (loops) it took**.
   a. *Hint1: Use **mod(k,2)** to determine if you should be adding or subtracting each loop iteration. **mod(k,2)** returns 0 if k is even, 1 if k is odd.*
   b. *Hint2: It's helpful to have two counter variables, one that keeps track of even or odd, one that keeps track of where the denominator starts.*

b) Repeat a), but using the second equation.

c) Which converges faster/took fewer iterations?

d) Can you represent the actual value of pi in MATLAB? Why or why not? Put your answer in the Word document

e) Extra credit, 10%: For each equation count how many *operations* you needed. An operation is any one of: adding two numbers, multiplying two numbers, assigning to a variable, checking a conditional statement.

Self-check: `First method: xxx,xxx iterations, second method: xx iterations`

Comments for grader/additional information (if any)

Script File

```
%Homework 4 Problem 3

clc
clear

%Calculate pi with first method
loopNum=0; %set the initial number of loops to zero
valuePi=0; %Set the initial value of value pi to zero
i=1; %set i equal to 1
while valuePi<3.14159-0.0000001 || valuePi>3.14159+0.0000001 %Only run loop
until sufficiently accurate (10^-5)
    loopNum=loopNum+1; %count the number of loops
```

```
    myPi0=4/i; %given equation to find pi
    myPi(loopNum)=myPi0; %store the values of myPi in an array
    if mod(loopNum,2)~=0 %run if number of loops is odd
        realPi(loopNum)=myPi(loopNum); %realPi index is positive version of
myPi index
    else %Run if number of loops is not odd (even)
        realPi(loopNum)=-myPi(loopNum); %realPi index is negative version of
myPi index
    end
    valuePi=valuePi+realPi(loopNum); %calculate the value of pie using the
array realPi
    i=i+2; %count odd numbers
end

fprintf('First Method: %0.0f iterations, \n',loopNum) %Print out number of
iterations
fprintf('Pi=%0.12f\n',valuePi) %Print out calculated value of pi
clear

%Calculate pi with second method
loopNum=0; %set the initial number of loops to zero
valuePi=0; %Set the initial value of value pi to zero
i=2; %Set i equal to 2
while valuePi<0.14159-0.00001 || valuePi>0.14159+0.00001 %Only run loop
until sufficiently accurate
    loopNum=loopNum+1; %count the number of loops
    myPi0=4/(i*(i+1)*(i+2)); %given equation to find pi
    myPi(loopNum)=myPi0; %store the values of myPi in an array
    if mod(loopNum,2)~=0 %run if number of loops is odd
        realPi(loopNum)=myPi(loopNum); %realPi index is positive version of
myPi index
    else %Run if number of loops is not odd (even)
        realPi(loopNum)=-myPi(loopNum); %realPi index is negative version of
myPi index
    end
    valuePi=valuePi+realPi(loopNum); %calculate the value of pie using the
array realPi
    i=i+2; %make i count even numbers
end
valuePi=valuePi+3; %add the 3 to value pi as specified by the given equation

fprintf('Second Method: %0.0f iterations, \n',loopNum) %Print out number of
iterations
fprintf('Pi=%0.12f\n',valuePi) %print out final calculated value of pi
```

Function Files

Command Window Output

```
First Method: 363164 iterations,
Pi=3.141589900013
```

```
Second Method: 28 iterations,
Pi=3.141582418248
```

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

c) Which converges faster/took fewer iterations?

The second method

d) Can you represent the actual value of pi in MATLAB? Why or why not? Put your answer in the Word document

No, you cannot represent the actual value of pi in Matlab, because pi is irrational. If you were to represent Pi in matlab, it would run infinitely trying to calculate it.

Problem 4

Epidemic Part 4 [Using functions]

Comments for grader/additional information (if any)

Script File

```
% Andrew Brown Homework 4 Problem 4

clc
clear
close all

% Epidemic Part 4: more practicing with if and for loops. In particular, I
% practiced with matrixes in this code.

%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] = DiseaseSimulate();

%Print out ending values after 140 timesteps
for j=1:length(S0)
    fprintf('Start S=%0.0f, Ending after %0.2f Days: S = %0.0f, I = %0.0f, R
= %0.0f, total %0.0f\n',
S(j),h*length(nSteps),S(length(nSteps),j),I(length(nSteps),j),R(length(nSteps
),j),N(j))
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,S(:, k),plotColor{k},'Linewidth',3) %plot s vs time
    title(strcat(titleName{1}, ' versus time') )%title the plot
    xlabel('Time') %X label for the plot
    ylabel('Number of People') %Y label for the plot
```

```
    axis([0,140,0,2200]) %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,I(:, k),plotColor{k},'Linewidth',3) %plot I vs time
    title(strcat(titleName{2}, ' versus time') )%title the plot
    xlabel('Time') %X label for the plot
    ylabel('Number of People') %Y label for the plot
    axis([0,140,0,2200]) %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 R vs Time
    hold on %keep all on same plot
    subplot(1,4,3) %plot in third position
    plot(nSteps,R(:, k),plotColor{k},'Linewidth',3) %plot R vs time
    title(strcat(titleName{3}, ' versus time') )%title the plot
    xlabel('Time') %X label for the plot
    ylabel('Number of People') %Y label for the plot
    axis([0,140,0,2200]) %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 Total vs Time
    hold on %keep all on same plot
    subplot(1,4,4) %plot in fourth position
    plot(1:length(N),N(:,k),plotColor{k},'Linewidth',3) %plot total vs time
    title(strcat(titleName{4}, ' versus time') )%title the plot
    xlabel('Time') %X label for the plot
    ylabel('Number of People') %Y label for the plot
    axis([0,140,0,2200]) %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
end
```

Function Files

```
function [S,I,R,N] = DiseaseStep(S,I,R,S0,I0,R0,a,b,h,nSteps)
%DISEASE STEP:
%   Runs one step of he disease and spits out S, I, R, and N

%Set first rows to the given values S0, I0, and R0
if nSteps==1
    S(1, :)=S0; %Set first row of S to the S0 array
```

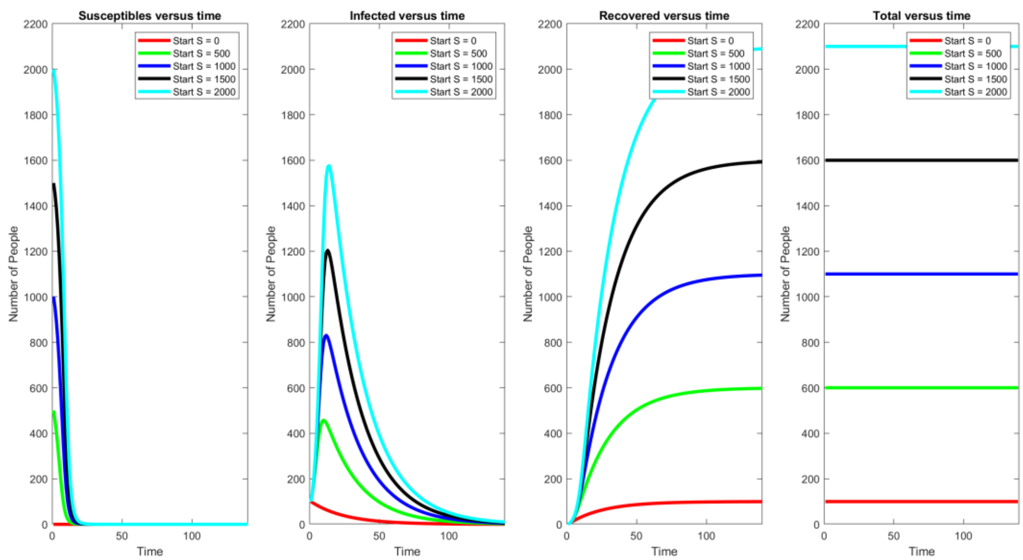
```
    I(1, :)=I0; %Set the first row of I to all equal the constant I0
    R(1, :)=R0; %Set the first row of R to all equal the constant R0
    S=S.*ones(1,length(S0)); %Preallocate S
    I=I.*ones(1,length(S0)); %Preallocate I
    R=R.*ones(1,length(S0)); %Preallocate R
end

%Equations used for calculations of the first timestep.
N=S0+I0+R0; %Total population
dS(nSteps, :)=(-a.*S(nSteps, :).*I(nSteps, :))./N(nSteps, :); %How S changes
dI(nSteps, :)=((a.*S(nSteps, :).*I(nSteps, :))./N(nSteps, :))-
(I(nSteps, :)./b); %How I changes
dR(nSteps, :)=I(nSteps, :)./b; %How R changes

%Calculations for values over time using the previous timestep.
S(nSteps+1, :)=S(nSteps, :)+(h.*dS(nSteps, :)); %# of Sick people after time
h.
I(nSteps+1, :)=I(nSteps, :)+(h.*dI(nSteps, :)); %# of Infected people after
time h.
R(nSteps+1, :)=R(nSteps, :)+(h.*dR(nSteps, :)); %# of Recovered people after
time h.
end

----------------------------------------------------------------------------

function [S,I,R,N] = DiseaseSimulate()
%DISEASE SIMULTE:
%   Calls DiseaseStep 140 times with 140 being the length of nSteps
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(1, :)=S0; %Set first row of S to the S0 array
I(1, :)=I0; %Set the first row of I to all equal the constant I0
R(1, :)=R0; %Set the first row of R to all equal the constant R0

for i=1:(length(nSteps)-1) %run until S, I, and R have 140 values
    [S,I,R,N] = DiseaseStep(S,I,R,S0,I0,R0,a,b,h,nSteps(i)); %call
DiseaseStep and loop through the nSteps array
    S0=S; %Update with new S0
    I0=I; %Update with new I0
    R0=R; %Update with new R0
end

end
```

Command Window Output
`Start S=0, Ending after 7.00 Days: S = 0, I = 0, R = 100, total 100`

```
Start S=0, Ending after 7.00 Days: S = 0, I = 2, R = 598, total 100
Start S=0, Ending after 7.00 Days: S = 0, I = 5, R = 1095, total 100
Start S=0, Ending after 7.00 Days: S = 0, I = 7, R = 1593, total 100
Start S=0, Ending after 7.00 Days: S = 0, I = 10, R = 2090, total 100
```

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

Image Output



Problem 5 [Extra credit]

[Extra credit]: Teacup Part 3 [Multiple Cups].

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)

Image Output
Copy and paste images here

Problem 6 [Extra credit]

[Extra credit]: There are three ways to calculate the number *e*, also known as Euler's number:

$\left(1 + \frac{1}{n}\right)^n$ - the bigger the value n, the better the approximation. n does not have to be an integer.

OR

$(1 + n)^{\frac{1}{n}}$ – the smaller the value n, the better the approximation. In this case make n be

OR

$$\frac{1}{0!} + \frac{1}{1!} + \frac{1}{2!} + \frac{1}{3!} + \frac{1}{4!} + \frac{1}{5!} + \frac{1}{6!} + \frac{1}{7!} + \ldots$$

Note: The ! is factorial – there is a function in MATLAB called factorial that will calculate it for you. But for your own reference, k! is k * (k-1) * (k-2)… all the way down to 1
Note: `exp(1)` is e to the 1st power which is e. MATLAB doesn't have an e variable like it does pi.

a) For the first method, find the first integer n where the approximation is within 0.01 of MATLAB's version of e
b) For the second method, find the smallest value of n=1/m where the approximation is within 0.01 of MATLAB's version of e. Make m be an integer.
c) Keep summing 1/k! until the approximation is within 0.01 of MATLAB's version of e.

Self-check:

```
My e version (1+1/n)^n: 2.718181832191, MATLAB e: 2.718281828459,
 n is 1XX91
My e version (1+n)^(1/n): 2.718181832191, MATLAB e: 2.718281828459,
 n is 0.000073XX8103, m is 1XX91
My e version factorial: 2.71XXXX968254, MATLAB e: 2.718281828459,
 8 iterations, k is 7
>>
```

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).

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