Name: Andrew Brown	Lab Time: T 12:00

Names of people you worked with:

- Hannah Gamache
- Cailin Moore
- Youta Kashiwa
- Michael Pious

Websites you used:

- https://www.mathworks.com/help/matlab/creating_plots/plotting-in-polar-coordinates.html

Approximately how many hours did it take you to complete this assignment (to nearest whole number)?	8

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:

You should accomplish the following:

- Understand how to create arrays, calculate with them, and extract data from them.
- Plot the results of calculations.
- Be able to split up plots into multiple windows
- Plot multiple things in the same window with different colors and attributes.

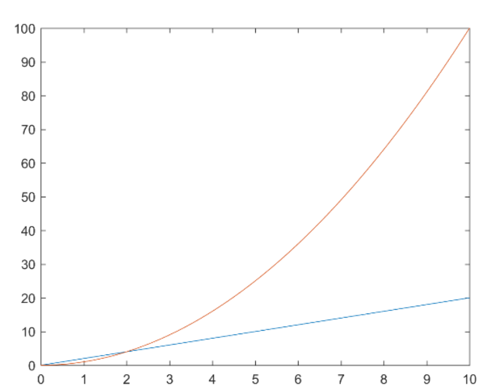
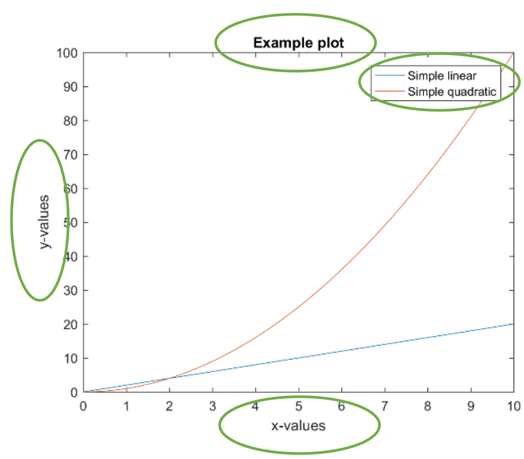
Homework Guidelines:

1) Use example scripts from online for help plotting and vectors.
2) Save scripts as "HW2Script1.m" for problem 1, "HW2Script2.m" for problem 2, etc.

Specific Coding Notes:

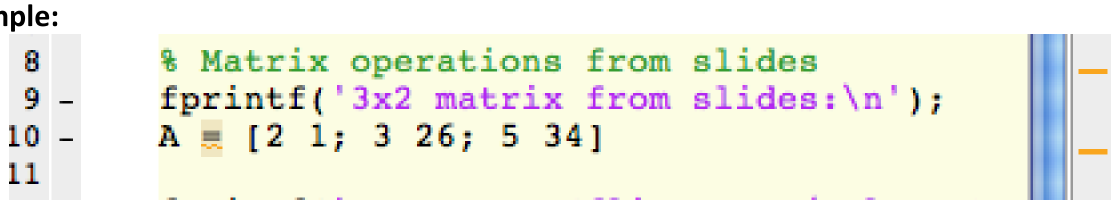
Plotting – Label axes (with units when known), use titles, and use legend when plotting more than one thing on a graph.

Example:

☹ NO ☹	☺ YES ☺
	

Warnings and Errors – You may see orange and red bars on the right-hand side of the editor. Orange bars are warnings and red bars are errors. Hovering with the cursor over a par will show you the problem.

Example:



Grading Checkpoints

Criteria	Component	No	Yes
[30%] Comments and Pseudocode	Declared units on all variables?		
	English description of problem at top?		
	Comments outlining your steps?		
[20%] Output formatting	Used fprintf() to make complete sentences (when required)?		
	Correct units on answers?		
	Correct number of decimal places?		
	Proper labels on graphs/plots?		
[50%] Functionality	Script computes/plots correct value(s)?		
	Did you use the correct **ranges** for input variables?		

Problem 1

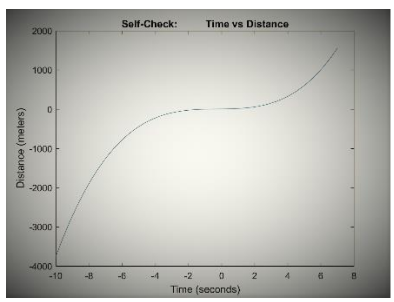
a) Plot the following mathematical function in the domain -10 ≤ t ≤ 7 in increments of 0.004. Assume that the x-axis is time measured in seconds, and the y-axis is distance measured in meters. Print the minimum and maximum values of y for that range of ts, and the t-values that produce the minimum and maximum values. Label the axes with time in seconds (t) and distance in meters (y).
   a. *Hint: Look up the different versions of the function min/max.*

b) Check the minimum value against the analytic solution (hint: minimum and maximum values occur when the derivative is zero).
   a. If you don't want to do the derivative by hand, try this web-site (http://www.derivative-calculator.net/). This web-site (http://www.numberempire.com/equationsolver.php) can solve the resulting equation for you.

The equation:

$$y = 4t^3 + 3t^2 + 5t + 5$$

Self-check:




```
Minimum occurs at y = -3xx5.xxx, maximum at edge of graph
```

Comments for grader/additional information (if any)

Script File

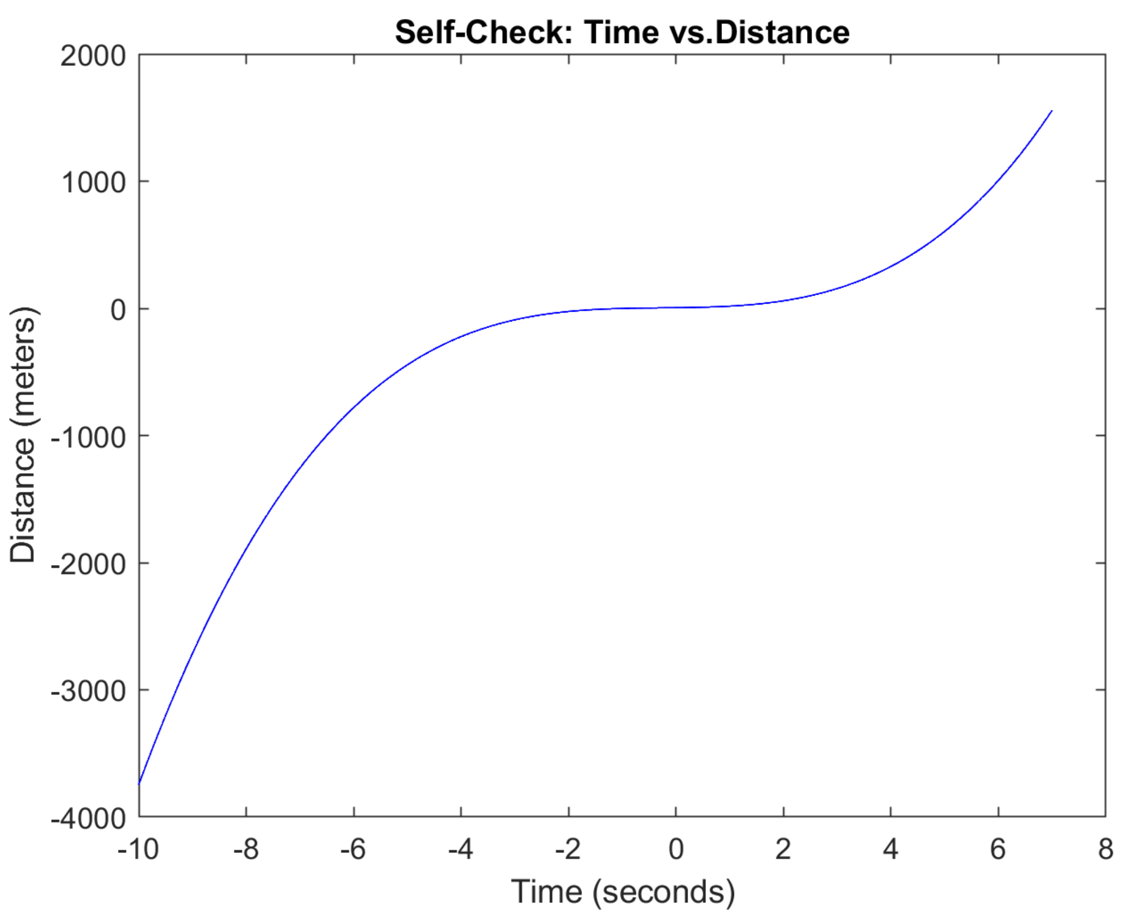
```
%Andrew Brown Homework 2 Problem 1

clc
clear
clf

%Practice plotting arrays with function.

t=-10:0.004:7;
y=(4*t.^3)+(3*t.^2)+(5*t)+5;
```

```
plot(t,y,'b')
xlabel('Time (seconds)')
ylabel('Distance (meters)')
title('Self-Check: Time vs. Distance')

fprintf('Minimum occurs at y = %0.3f, maximum occurs at y = %0.3f\n',
min(y), max(y))
```

Command Window Output

```
Minimum occurs at y = -3745.000, maximum occurs at y = 1559.000
```

Image Output



Problem 2

A polar rose is a sinusoidal function that looks like a rose in polar coordinates. It can also be represented in Cartesian coordinates. The equations for a polar rose are given below:

<u>Polar Coordinates</u>

$$r = \cos\left(\frac{n}{d}\theta\right)$$

<u>Cartesian (Regular) Coordinates</u>

$$x = \cos\left(\frac{n}{d}t\right)\cos(t)$$

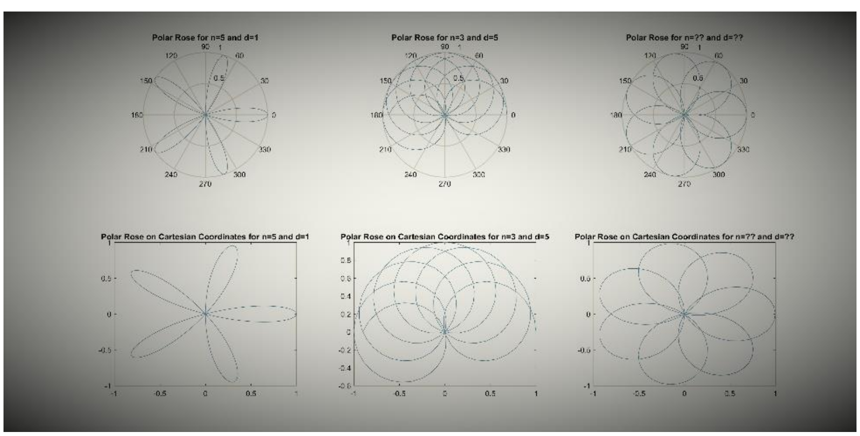
$$y = \cos\left(\frac{n}{d}t\right)\sin(\mathrm{t})$$

Create a 2x3 grid of six polar rose plots in the same window (using the subplot function). Include a descriptive title for each subplot, but axis labels are not required. Assume units of radian for θ and t. Look up the `polar()` command for the plots using the polar coordinate system.

Use the following blueprint to create and organize your plots in the figure window:

<u>Polar Coordinates</u> n = 5 (1,1) d = 1 0 ≤ θ ≤ π (use 200 values total) 0 ≤	<u>Polar Coordinates</u> n = 6 (1,2) d = 7 0 ≤ θ ≤ 8π (use 200 values total)	<u>Polar Coordinates</u> n = Your Choice (1,3) d = Your Choice Choose an appropriate θ domain
<u>Cartesian Coordinates</u> n = 5 (2,1) d = 1 0 ≤ t ≤ π (use 200 values total)	<u>Cartesian Coordinates</u> n = 6 (2,2) d = 7 0 ≤ t ≤ 8π (use 200 values total)	<u>Cartesian Coordinates</u> n = Your Choice (2,3) d = Your Choice Choose an appropriate t domain

Self check: *Note – the ones in the 3rd column – your choice – don't have to look like the ones shown*



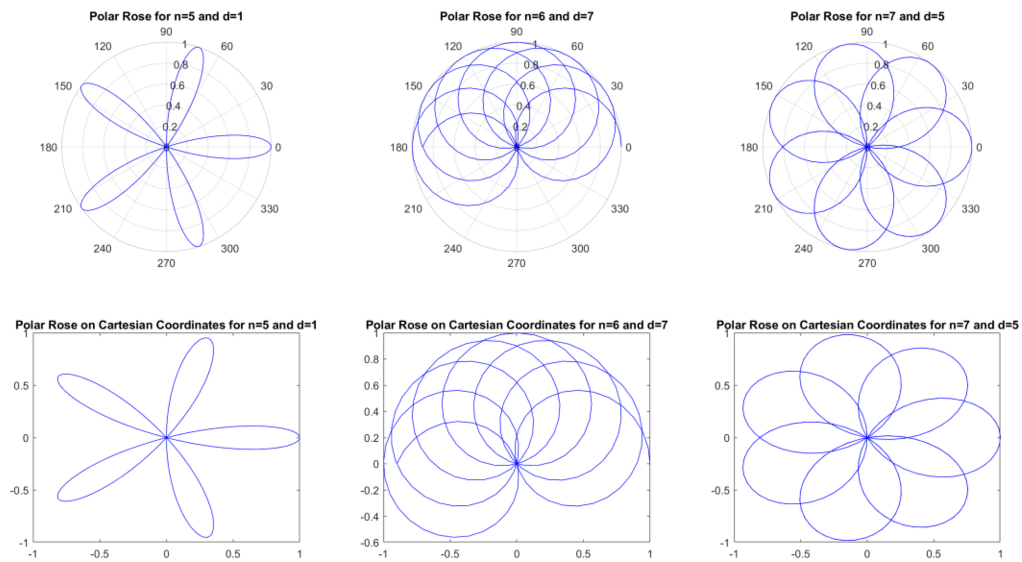
Comments for grader/additional information (if any)

Script File

```
% Andrew Brown Homework 2 Problem 2

clc
clear
clf

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

%Define given variables.
n=5:7;
d=[1, 7, 5];
theta1=linspace(0,pi,200);
theta2=linspace(0,8*pi,200);
theta3=linspace(0,12*pi,500);
t1=linspace(0,pi,200);
t2=linspace(0,8*pi,200);
t3=linspace(0,12*pi,500);

   %Express given equations.
   x1=cos((n(1)/d(1)).*t1).*cos(t1); %Cartesian coodinates
   y1=cos((n(1)/d(1)).*t1).*sin(t1); %Cartesian coodinates
   r1=cos((n(1)/d(1)).*theta1); %Polar coodinates

       %Create a 2x3 subplot and plot in polar coodinates in spot 1.
       subplot(2,3,1)
       polarplot(theta1,r1,'b')
       title('Polar Rose for n=5 and d=1')

       %Plot in cartesian coodinates in subplot spot 4.
       subplot(2,3,4)
       plot(x1,y1,'b')
       title('Polar Rose on Cartesian Coordinates for n=5 and d=1')

   %Express given equations again for second t and theta values.
   x2=cos((n(2)/d(2)).*t2).*cos(t2); %Cartesian coodinates
   y2=cos((n(2)/d(2)).*t2).*sin(t2); %Cartesian coodinates
   r2=cos((n(2)/d(2)).*theta2); %Polar coodinates

       %Plot in polar coodinates in subplot spot 2.
       subplot(2,3,2)
       polarplot(theta2,r2,'b')
       title('Polar Rose for n=6 and d=7')

       %Plot in cartesian coodinates in subplot spot 5.
       subplot(2,3,5)
```

```
    plot(x2,y2,'b')
    title('Polar Rose on Cartesian Coordinates for n=6 and d=7')

%Express given equations again for third t and theta values.
x3=cos((n(3)/d(3)).*t3).*cos(t3); %Cartesian coodinates
y3=cos((n(3)/d(3)).*t3).*sin(t3); %Cartesian coodinates
r3=cos((n(3)/d(3)).*theta3); %Polar coodinates

    %Plot in polar coodinates in subplot spot 3.
    subplot(2,3,3)
    polarplot(theta3,r3,'b')
    title('Polar Rose for n=7 and d=5')

    %Plot in cartesian coodinates in subplot spot 6.
    subplot(2,3,6)
    plot(x3,y3,'b')
    title('Polar Rose on Cartesian Coordinates for n=7 and d=5')
```

Image Output



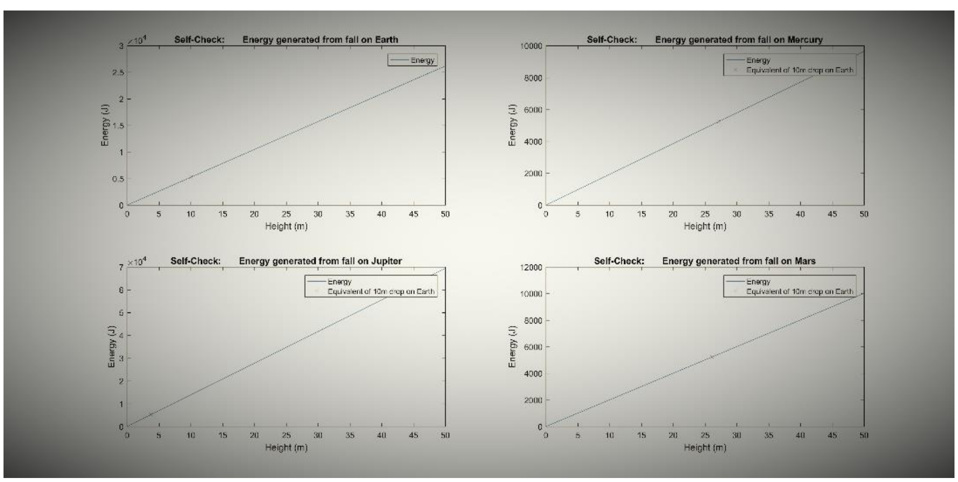
Problem 3

Sami weighs about 120 pounds. After reading an article about falling from heights (https://www.theguardian.com/science/2004/may/20/thisweekssciencequestions2) he wonders, what could be an equivalent energy generated by a 10 meter high fall if it happened in other planets? In order to simplify this problem, we will ignore the existence of air resistance. Find the following:

a) What is the energy that is generated by a 10-meter fall for a 120 pounds heavy object on the surface of earth? Print your answer.
   a. 1 kg = 2.246 pounds.
   b. Energy of a fall: $Energy\ [J] = mass\ [kg] * \ g \left[\frac{m}{s^2}\right] * h\ [m]$
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.
   a. Gravitational constants: http://www.physicsclassroom.com/class/circles/Lesson-3/The-Value-of-g
   b. If the range of 0 to 50 meters is too large or too small, change it to an appropriate range.
c) Identify on your graph and command window the harmful heights that will generate the same energy as a 10-meter fall on earth.

Self check:

```
Energy from a fall on Earth: 5XXX.9X Joules
```



Comments for grader/additional information (if any)

Script File

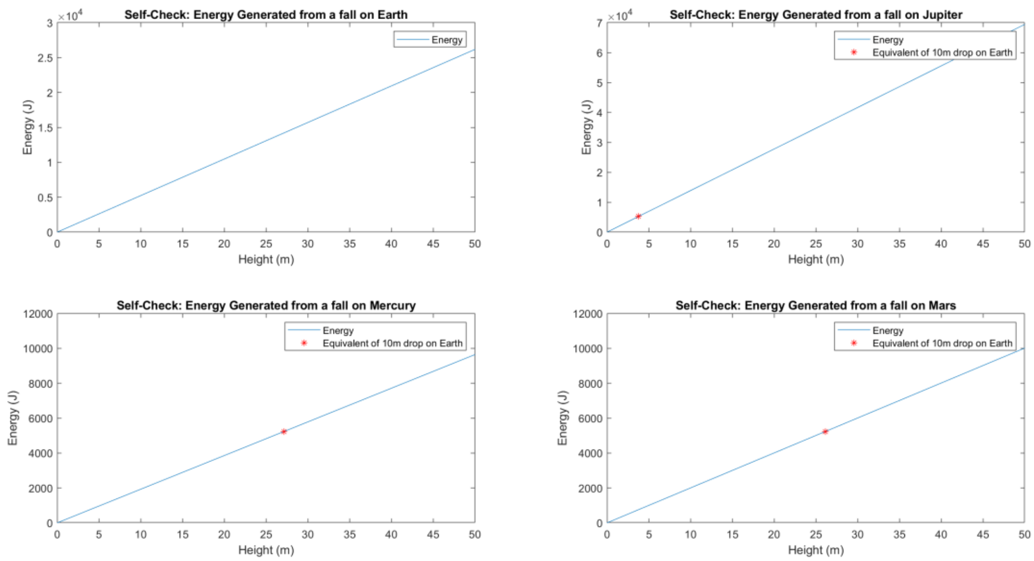
```
%Andrew Brown Homework 2 Problem 3

clc
clear
clf

%Phyisics of a harmful fall on different planets.

%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 equations separated by planet
EEarth=massKg*g(1)*h; %energy generated from the fall in joules on earth
EJupiter=massKg*g(2)*h; %energy generated from the fall in joules on jupiter
EMercury=massKg*g(3)*h; %energy generated from the fall in joules on mercury
EMars=massKg*g(4)*h; %energy generated from the fall in joules on mars

%Plotting the graph for energy generated from a fall on Earth
subplot(2,2,1)
plot(h,EEarth)
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,EJupiter)
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,EMercury)
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,EMars)
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)
```

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
```

Image Output



Problem 4

Total Solar Eclipse Part 1.

Comments for grader/additional information (if any)

Script File

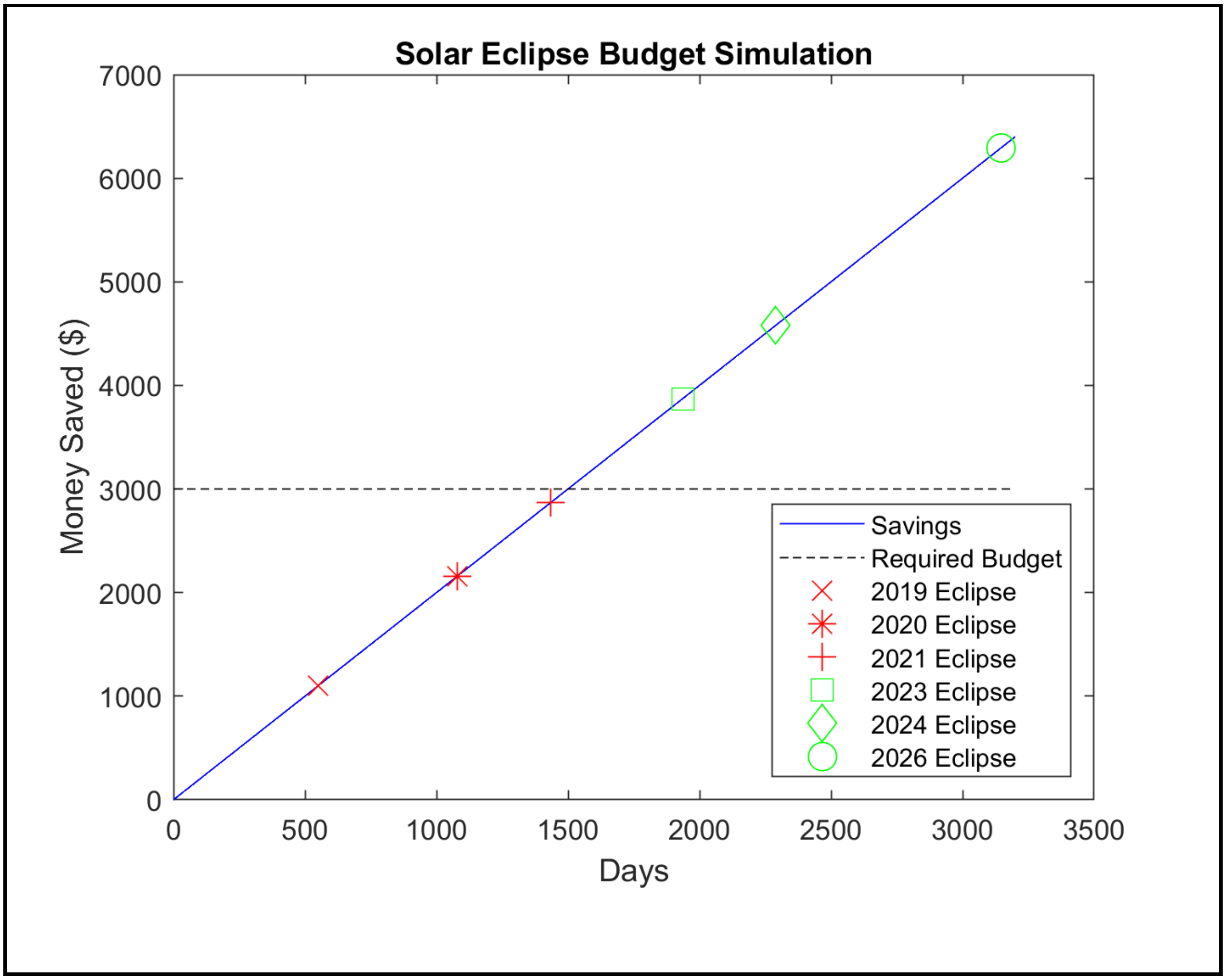
```
%Andrew Brown Homework 2 Problem 4

clc
clear
clf

%Total Solar Eclipse Part 1

%Part A: Create three variables: Year, Month, and Day. These three
%variables should contain a vector of the corresponding values of the
%TSE dates.

%Defining the three time-based variables with the given data.
day=[2,14,4,20,8,12]; %Days of the future TSE
month=[7,12,12,4,4,8]; %months of the future TSE
year=[2019,2020,2021,2023,2024,2026]; %years of the future TSE

%Part B: Calculate how many days between January 1st, 2018 and each of
%the upcoming TSE dates. Use the assumption that the Gregorian (western)
%solar calendar month, on average has 30.44 days. And every year has on
%average 365.2425 days.

%Do the needed year and month conversions into days
daysInAMonth=30.44; %Average number of days in a month
daysInAYear=365.2425; %Average number of days in a year

%Set up the arrays into only days for the end calculation
daysSubtracted=736695; %Days before Jan. 1, 2018
monthsToDays=(month-1).*daysInAMonth; %convert months to days
yearsToDays=(year-1).*daysInAYear; %convert years to days

%Calculation of days between Jan 1, 2018 and the given TSEs
daysUntilTSE=yearsToDays+monthsToDays+day-daysSubtracted;
roundedDaysUntilTSE=floor(daysUntilTSE);
%Part C: Calculate the daily saving balance that you will accumulate in
%your account starting from day one until and ending on day 3,200.
%(Assume that you are adding $2/day into your account. No interest is
%added to your saving account). Assign the daily savings of all 3,200
%values into one array.

dailySavingsBalance=2:2:6400; %Units in $

%Part D: Generate a plot that shows all of the following:
%1. The value of your daily savings.
```

```
%2. Plot a marker on the plot for each TSE and your savings that day.
%3. Plot a dashed horizontal line for savings = $3000
%4. Print out how many days are left until the first TSE you can afford and
%their savings balance that day.


countDays=1:3200; %Array to count the days in order

%Plot the Solar Eclipse Budget Simulation
plot(countDays,dailySavingsBalance,'b')
title('Solar Eclipse Budget Simulation')
xlabel('Days')
ylabel('Money Saved ($)')
hold on

%Plot a horizontal line at y=3000 using the ines function
horizontalLine=ones(1,3200);
plot(countDays,3000.*horizontalLine,'--k')

%Plot each solar eclipse date and savings.
plot(daysUntilTSE(1),dailySavingsBalance(roundedDaysUntilTSE(1)),'xr','Marker
Size',10) %2019 eclipse
plot(daysUntilTSE(2),dailySavingsBalance(roundedDaysUntilTSE(2)),'*r','Marker
Size',10) %2020 eclipse
plot(daysUntilTSE(3),dailySavingsBalance(roundedDaysUntilTSE(3)),'+r','Marker
Size',10) %2021 eclipse
plot(daysUntilTSE(4),dailySavingsBalance(roundedDaysUntilTSE(4)),'sg','Marker
Size',10) %2023 eclipse
plot(daysUntilTSE(5),dailySavingsBalance(roundedDaysUntilTSE(5)),'dg','Marker
Size',10) %2024 eclipse
plot(daysUntilTSE(6),dailySavingsBalance(roundedDaysUntilTSE(6)),'og','Marker
Size',10) %2026 eclipse

%Create the legend in the SW corner
legend('Savings','Required Budget','2019 Eclipse','2020 Eclipse','2021
Eclipse','2023 Eclipse','2024 Eclipse','2026 Eclipse','location','SE')

%Print out the first affordable TSE trip and your savings on that day
fprintf('The first affordable TSE trip will be on day %0.0f\nat a budget of
$%0.0f.\n', daysUntilTSE(4), dailySavingsBalance(roundedDaysUntilTSE(4)))
```

Answers to question(s) asked in the homework or command window output (if any)

Command Window Output:
The first affordable TSE trip will be on day 1937
at a budget of $3872.

Image Output


Solar Eclipse Budget Simulation
Money Saved ($)
Days
0
1000
2000
3000
4000
5000
6000
7000
0
500
1000
1500
2000
2500
3000
3500
Savings
Required Budget
2019 Eclipse
2020 Eclipse
2021 Eclipse
2023 Eclipse
2024 Eclipse
2026 Eclipse

Problem 5

Teacup Part 2 [Make a teacup and plot it in two positions]

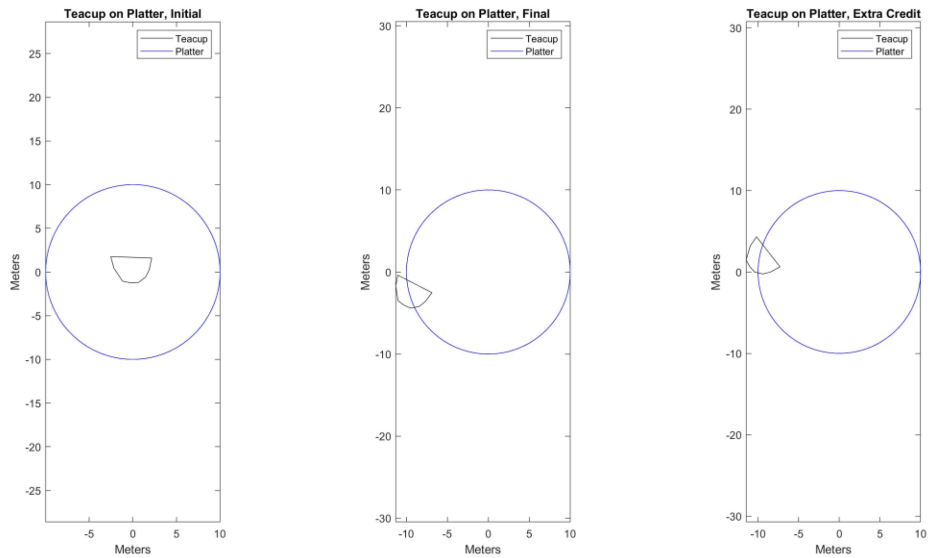
Comments for grader/additional information (if any)

Script File

```
% Andrew Brown Homework 2 Problem 5

clc
clear
clf

%Teacup Part 2

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Read them back in again - put this in the top of your script
xsys = dlmread('TeacupPts.txt');

% Get x and y points back out
xs = xsys(1,:);
ys = xsys(2,:);

% Plot the points from the original sketch at the origin
subplot(1,3,1)
plot(xs,ys, '-k');
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%Plot the plate as a circle of radius 10
hold on
radius=10;
angle=linspace(0,2*pi,720);
x=cos(angle)*radius;
y=sin(angle)*radius;
plot(x,y,'b')
axis equal
title('Teacup on Platter, Initial')
xlabel('Meters')
ylabel('Meters')
legend('Teacup','Platter')


%Part1.2a


%Define constants
theta=-25; %degrees
dx=-9.52; %translation distance in meters
dy=-3.05;%translation distance in meters
```

```
%Rotation then translation of the initial points.
%Rotation of the teacup points
x0RotateTeacup=(xs*cosd(theta))-(ys*sind(theta));
y0RotateTeacup=(xs*sind(theta))+(ys*cosd(theta));

%Translation of the teacup with rotation positions
xTranslationTeacup=x0RotateTeacup+dx;
yTranslationTeacup=y0RotateTeacup+dy;

%Plot the rotated, then translated teacup with the plate
subplot(1,3,2)
plot(xTranslationTeacup,yTranslationTeacup,'-k')
hold on
plot(x,y,'b')
axis equal
title('Teacup on Platter, Final')
xlabel('Meters')
ylabel('Meters')
legend('Teacup','Platter')

%Rotate again using translation positions
x1RotateTeacup=(xTranslationTeacup*cosd(theta))-
(yTranslationTeacup*sind(theta));
y1RotateTeacup=(xTranslationTeacup*sind(theta))+(yTranslationTeacup*cosd(thet
a));

%Plot position after second rotation with plate
subplot(1,3,3)
plot(x1RotateTeacup,y1RotateTeacup,'-k')
hold on
plot(x,y,'b')
axis equal
title('Teacup on Platter, Extra Credit')
xlabel('Meters')
ylabel('Meters')
legend('Teacup','Platter')
```

Image Output


Teacup on Platter, Initial
Teacup on Platter, Final
Teacup on Platter, Extra Credit
Teacup
Platter
Meters
Meters

Problem 6

Epidemic Part 2 [Equations with arrays]

Comments for grader/additional information (if any)

Script File

```
% Andrew Brown Homework 2 Problem 6

clc
clear

% Epidemic Part 1

%Part 1.1

%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

%Print out parameters.
fprintf('Parameters: avg contacted, %0.0f, number of days
infectious, %0.2f\n', a, b)

%Print out starting S values.
fprintf('Starting S
values:\n   %0.0f   %0.0f   %0.0f   %0.0f   %0.0f\n\n',S0(1),S0(2),S0(3),S0(4
),S0(5))

%Part 1.1

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

%Calculations for values over time using the first timestep.
S1=S0+(h.*dS); %# of S after time h.
I1=I0+(h.*dI); %# of I after time h.
R1=R0+(h.*dR); %# of R after time h.

%Equations used for calculations of the second timestep.
dS=(-a.*S1.*I1)./N; %How S changes
```

```
dI=((a.*S1.*I1)./N)-(I1./b); %How I changes
dR=I1./b; %How R changes

%Calculations for values over time using the second timestep.
S2=S1+(h.*dS); %# of S after time h.
I2=I1+(h.*dI); %# of I after time h.
R2=R1+(h.*dR); %# of R after time h.

%Print out ending values after 2 timesteps
fprintf('Ending S values:\n')
disp(S2)
fprintf('Ending I values:\n')
disp(I2)
fprintf('Ending R values:\n')
disp(R2)
fprintf('Totals (S + I + R):\n')
disp(N)
```

Command Window Output

```
Parameters: avg contacted, 10, number of days infectious, 1.25
Starting S values:
   0   500   1000   1500   2000

Ending S values:
   1.0e+03 *

         0    0.4058    0.8932    1.3882    1.8856

Ending I values:
   92.1600  184.7410  197.1713  202.0398  204.6360

Ending R values:
    7.8400    9.5067    9.6582    9.7150    9.7448

Totals (S + I + R):
         100          600         1100         1600         2100
```