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

Names of people you worked with:
• Kevin Daellenbach • Hannah Gamache • Cailin Moore

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.

Learning Objectives:

- Making (and using) matrices to move things around
- Plotting curves in 3D using plot3
- Making and plotting surfaces using meshgrid and surf

Problem 1

Compute the total mass (g) of the rocket components shown in the table below using a dot product. ***Print your answer in the command window.***

Component	Density (g/cm^3)	Volume (cm^3)
Propellant	0.7	700
Tank supports	6.8	150
Tank material	2.9	300
Tile tank lining	4.5	50

Self-check: `X6XX.00`

Comments for grader/additional information (if any)

Script File

```
%Andrew Brown Homework 9 Problem 1

clc
clear

%Compute the total mass (g) of the rocket compononets shown in the given
%table using a dot product.

%Given vlues from the table
density=[0.7,6.8,2.9,4.5]; %g/cm^3
volume=[700,150,300,50]; %cm^3

%Take the dot product of the 2 matrices made and print the result
dotProduct=dot(density,volume);
fprintf('Total Mass of Rocket Parts: %0.2f g\n',dotProduct)
```

Command Window Output

```
Total Mass of Rocket Parts: 2605.00 g
>>
```

Problem 2

An anti-symmetric cross-ply composite laminate has two layers in which the fibers are aligned perpendicular to one another. A laminate of this type will deform into a saddle shape due to residual thermal stresses as described by the equation

$$z = k(x^2 - y^2)$$

where *x* and *y* are the in-plane coordinates and *k* is the curvature. ***Make a surface plot*** showing the deflection of a one-meter by one-meter square plate using $-0.5 \le x \le 0.5$ m and $-0.5 \le y \le 0.5$ m. Assume k=0.77. Choose a step size that captures the shape but doesn't have too many points (use the same step size in x and y).

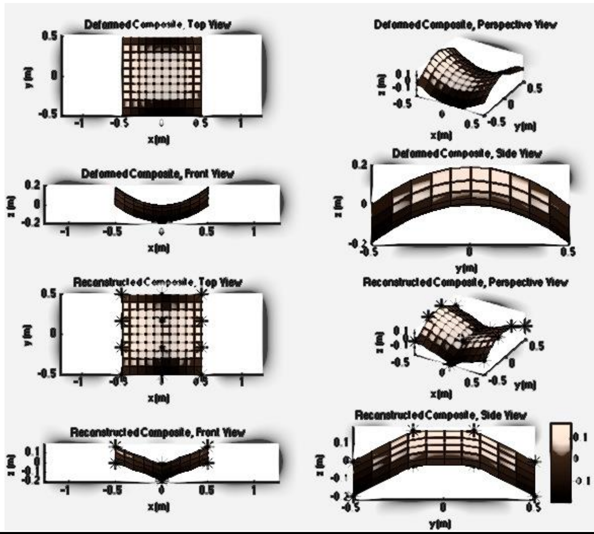
a) Use the `view` command to print four subplots as shown below, then add a `colorbar` to the graph. Use `help graph3d` in the command window to find some different color maps, then use `colormap` to choose your favorite. Use `axis equal` to show the true deformation.

b) Now create a set of x and y sample points of the function using 3 sample directions in x and 4 in *y*. Use `interp2` to interpolate these sample points with the same mesh you used in part a). Plot the sample points and the resulting surface. How does this compare to the original?

For fun: Click on the rotate tool in the figure window then click and drag on one of your figures

Self-check:

TOP VIEW (original)	PERSPECTIVE (30 degrees, 30 degrees) (original)
FRONT VIEW (original)	SIDE VIEW (original)
TOP VIEW (reconstructed)	PERSPECTIVE (reconstructed)
FRONT VIEW (reconstructed)	SIDE VIEW (reconstructed)



Comments for grader/additional information (if any)

Script File

```
%Andrew Brown Homework 9 Problem 2

clc
clear
close all

%Practice with graphing in 3D and adding color to the graph

%Define given variables
x=linspace(-0.5,0.5,11); %in-plane coordinate
y=linspace(-0.5,0.5,11); %in-plane coordinate
[x,y]=meshgrid(x,y); %Create the grid vectors of x and y
k=0.77; %curvature

%Define gven equation with the given variables
z=k*(x.^(2)-y.^(2));

%Top View
subplot(4,2,1)
surf(x,y,z)
colormap('summer')
axis equal
view([90,90])
title('Deformed Composite Top View')

%Perspective View
subplot(4,2,2)
surf(x,y,z)
colormap('summer')
axis equal
view([30,30])
```

```
title('Deformed Composite Perspective View')

%Front View
subplot(4,2,3)
surf(x,y,z)
colormap('summer')
axis equal
view([0,0])
title('Deformed Composite Front View')

%Side View
subplot(4,2,4)
surf(x,y,z)
colormap('summer')
axis equal
view([90,0])
title('Deformed Composite Side View')

%Reconstruct the composite
minx=min(x,[],'all'); %minimum of x matrix
maxx=max(x,[],'all'); %maximum of x matrix
rangex=linspace(minx,maxx,3); %range b/w the min and max of x
miny=min(y,[],'all'); %minimum of y matrix
maxy=max(y,[],'all'); %maximum of y matrix
rangey=linspace(miny,maxy,4); %range b/w the min and max of y
[rangex,rangey]=meshgrid(rangex,rangey);%run the ranges through meshgrid
z2=k*(rangex.^(2)-rangey.^(2));%find another z with the new x and y points
newZ=interp2(rangex,rangey,z2,x,y); %find interpolated z values

%Top View
subplot(4,2,5)
surf(x,y,newZ)
colormap('summer')
axis equal
view([90,90])
title('Reconstructed Composite Top View')

%Perspective View
subplot(4,2,6)
surf(x,y,newZ)
colormap('summer')
axis equal
view([30,30])
title('Reconstructed Composite Perspective View')

%Front View
subplot(4,2,7)
surf(x,y,newZ)
colormap('summer')
axis equal
view([0,0])
title('Reconstructed Composite Perspective View')

%Side View
subplot(4,2,8)
surf(x,y,newZ)
```

```
colormap('summer')
axis equal
view([90,0])
title('Reconstructed Composite Perspective View')
colorbar
```

Function Files (if any)

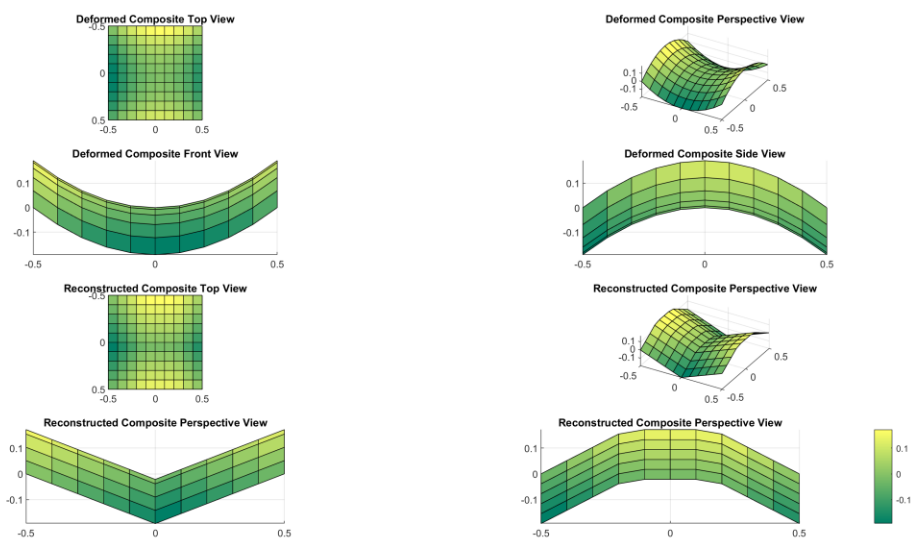
```
% Copy and paste your functions here. Must be size 10, same as MATLAB font and color.
```

Answers to question asked in the homework

b) How does this (interpolation) compare to the original?

The interpolation makes a more jagged graph because it can't tell with so few points that the graph is as curvy as it is. This makes the graph look more linear between each point.

Image Output



Problem 3

An elliptical staircase of height *h* and with *n* turns can be modeled by the parametric equations *x, y, z*

$$r(t) = \frac{b}{\sqrt{1 - \epsilon^2 cos^2(t)}}$$
$$x(t) = r(t)\cos(t)$$
$$y(t) = r(t)\sin(t)$$
$$z(t) = \frac{ht}{2\pi n}$$

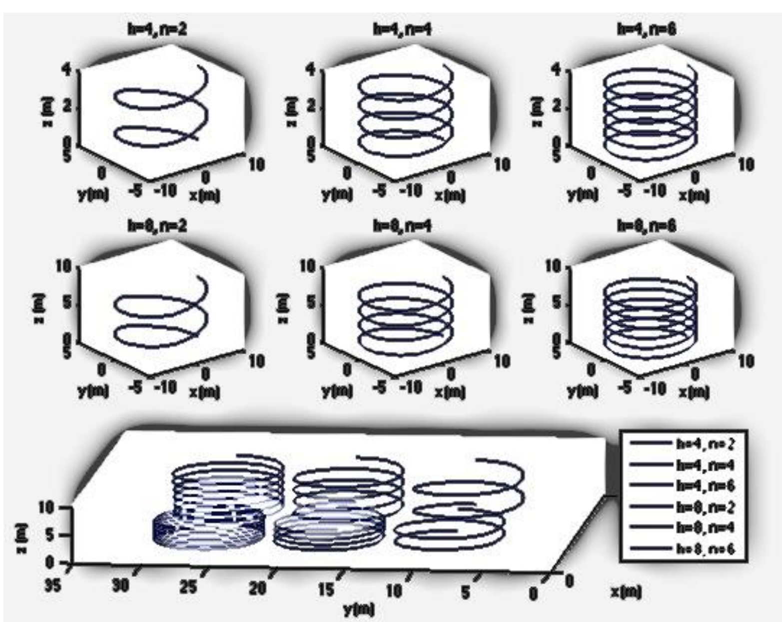
Where *r* is the radius and $\epsilon = \sqrt{1 - {b^2}/{a^2}}$ defines the eccentricity of the ellipse (*a* and *b* are the semimajor and minor axes of the ellipse).

a) ***Write a staircase function*** that takes in *a, b, h*, and *n* and returns the corresponding *x, y*, and *z* values. It should set the t range so that the staircase has sufficient points.

b) Use your staircase function to ***plot a grid*** of 6 copies of the staircase with *a* = 8m, *b* = 4m, *h* = 4m and 8m, and *n* = 2, 4, and 6. Use `subplot`.

c) Extra credit: [30pts] Use translation to plot all 6 copies in the same window.

Self-check: Extra credit shown on the bottom of the plot



Comments for grader/additional information (if any)

Script File

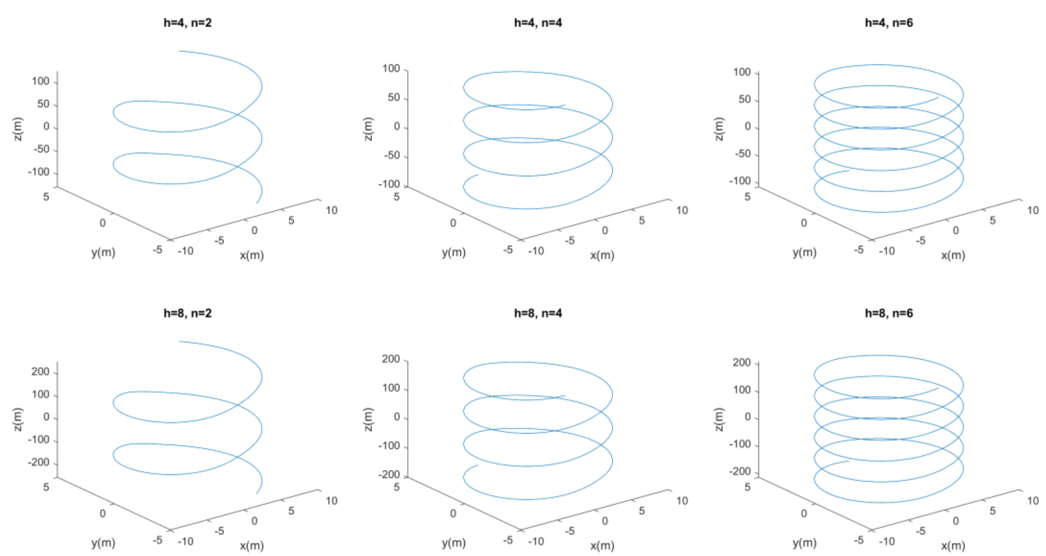
```
%Andrew Brown Homework 9 Problem 3

clc
clear
close all

%Model an elliptical staircase in different ways

%h=4 n=2 elipse
%define variables
t=linspace(-400,400,10000); %independent variable
a=8; %m
b=4; %m
h=4; %m
n=2; %m

[x,y,z] = Staircase(a,b,h,n,t); %Make the staircase
subplot(2,3,1) %place in subplot
plot3(x,y,z) %plot in 3 dimensions
xlabel('x(m)') %x label
ylabel('y(m)') %y label
zlabel('z(m)') %z label
title('h=4, n=2') %title

%h=4 n=4 elipse
%define variables
t=linspace(-600,600,10000); %independent variable
a=8; %m
b=4; %m
h=4; %m
n=4; %m

[x,y,z] = Staircase(a,b,h,n,t); %Make the staircase
subplot(2,3,2) %place in subplot
plot3(x,y,z) %plot in 3 dimensions
xlabel('x(m)') %x label
ylabel('y(m)') %y label
zlabel('z(m)') %z label
title('h=4, n=4') %title

%h=4 n=6 elipse
%define variables
t=linspace(-1000,1000,10000); %independent variable
a=8; %m
b=4; %m
h=4; %m
n=6; %m
```

```
[x,y,z] = Staircase(a,b,h,n,t); %Make the staircase
subplot(2,3,3) %place in subplot
plot3(x,y,z) %plot in 3 dimensions
xlabel('x(m)') %x label
ylabel('y(m)') %y label
zlabel('z(m)') %z label
title('h=4, n=6') %title

%h=8 n=2 elipse
%define variables
t=linspace(-400,400,10000); %independent variable
a=8; %m
b=4; %m
h=8; %m
n=2; %m

[x,y,z] = Staircase(a,b,h,n,t); %Make the staircase
subplot(2,3,4) %place in subplot
plot3(x,y,z) %plot in 3 dimensions
xlabel('x(m)') %x label
ylabel('y(m)') %y label
zlabel('z(m)') %z label
title('h=8, n=2') %title

%h=8 n=4 elipse
%define variables
t=linspace(-600,600,10000); %independent variable
a=8; %m
b=4; %m
h=8; %m
n=4; %m

[x,y,z] = Staircase(a,b,h,n,t); %Make the staircase
subplot(2,3,5) %place in subplot
plot3(x,y,z) %plot in 3 dimensions
xlabel('x(m)') %x label
ylabel('y(m)') %y label
zlabel('z(m)') %z label
title('h=8, n=4') %title

%h=8 n=6 elipse
%define variables
t=linspace(-1000,1000,10000); %independent variable
a=8; %m
b=4; %m
h=8; %m
n=6; %m

[x,y,z] = Staircase(a,b,h,n,t); %Make the staircase
subplot(2,3,6) %place in subplot
plot3(x,y,z) %plot in 3 dimensions
xlabel('x(m)') %x label
ylabel('y(m)') %y label
zlabel('z(m)') %z label
title('h=8, n=6') %title
```

Function Files

```
function [x,y,z] = Staircase(a,b,h,n,t)
%STAIRCASE Takes in 3D elipse parameters and returns the x,y, and z
%coordinates needed to plot the given elipse
E=sqrt(1-(b^(2)/(a^(2)))); %elipse eccentricity
r=b./sqrt(1-(E.*cosd(t)).^(2)); %elipse radius
x=r.*cosd(t); %x coordinates
y=r.*sind(t); %y coodinates
z=(h.*t)./(2*pi*n); %z coodinates
end
```

Image Output



Problem 4

Teacup Part 5 – [Matrices]

Important Deliverables: Function files (if you used them instead of anonymous functions); Plot of teacups. The matrices printed in the self-check are primarily for your reference; they do NOT need to be printed.

Comments for grader/additional information (if any)

Script File

```
%Andrew Brown Homework 9 Problem 4

clc
clear
close all

%Teacup Part 5

% 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,:);
pts = [xs;ys;ones(1,length(xs))];

% Plot the points from the original sketch at the origin
subplot(1,2,1)
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')

%Define constants
theta1=-25; %degrees
theta2=60; %degrees
```

```
dx=radius; %translation distance in meters
dy=0;%translation distance in meters
sx=0.5; %x scale values
sy=0.5; %y scale values

%Define the Translation, Rotation, and Scaling matrices
[mTrans] = Translation(dx,dy); %translation matrix
[mRot] = Rotation(theta1); %rotation matrix
[mScale] = Scale(sx,sy); %scale matrix


%Calculations
rotTansScale=mTrans*mRot*mScale*pts; %Rotate then translate then scale
transRotScale=mRot*mTrans*mScale*pts; %Translate then rotate then scale
ptsMoved=mRot*mTrans*mRot*mScale*pts; %Rotate then translate then rotate
then scale

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

Function Files

```
function [mTrans] = Translation(dx,dy)
%TRANSLATION creates a translational matrix given the x and y translation
%values
%Define given variables and translation matrix
mTrans=eye(3); %idendtity matrix
mTrans(1,3)=dx; %translate x values
mTrans(2,3)=dy; %translate y values
end


function [mRot] = Rotation(theta)
%ROTATION creates a rotation matrix given an angle of ratation
%Define given variables and rotation matrix
mRot=eye(3); %identity matrix
mRot(1,1)=cosd(theta); %make rotation matrix
mRot(1,2)=sind(theta); %make rotation matrix
mRot(2,1)=-sind(theta); %make rotation matrix
mRot(2,2)=cosd(theta); %make rotation matrix
end


function [mScale] = Scale(sx,sy)
%SCALE creates a scale matrix given the x and y scale values
```

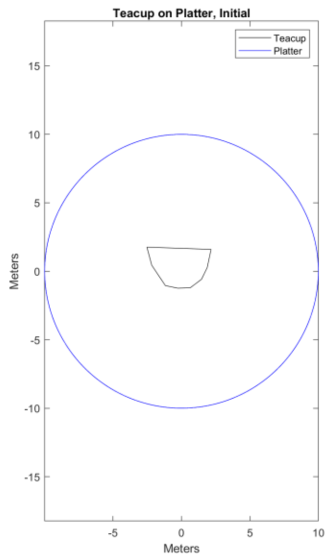
```
%Define given variables and scale matrix
mScale=eye(3); %identity matrix
mScale(1,1)=sx; %scale x values
mScale(2,2)=sy; %scale y values
end
```

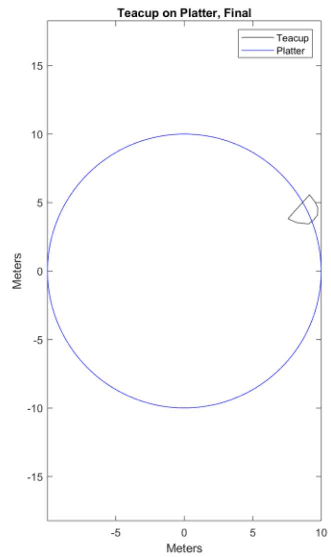
Command Window Output

```
>> mScale

mScale =

    0.5000         0         0
         0    0.5000         0
         0         0    1.0000

>> Rotation(theta1)

ans =

    0.9063   -0.4226         0
    0.4226    0.9063         0
         0         0    1.0000

>> mTrans

mTrans =

     1     0    10
     0     1     0
     0     0     1

>> Rotation(theta2)

ans =

    0.5000    0.8660         0
   -0.8660    0.5000         0
         0         0    1.0000

>> mRot*mTrans*mRot

ans =

    0.6428   -0.7660    9.0631
    0.7660    0.6428    4.2262
         0         0    1.0000

>>
```

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

Image Output





Problem 5

[Extra Credit] – Teacup Part 6 – [Pretty Pictures]

Important Deliverables: One snazzy plot.

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.

Image Output
Copy and paste images here