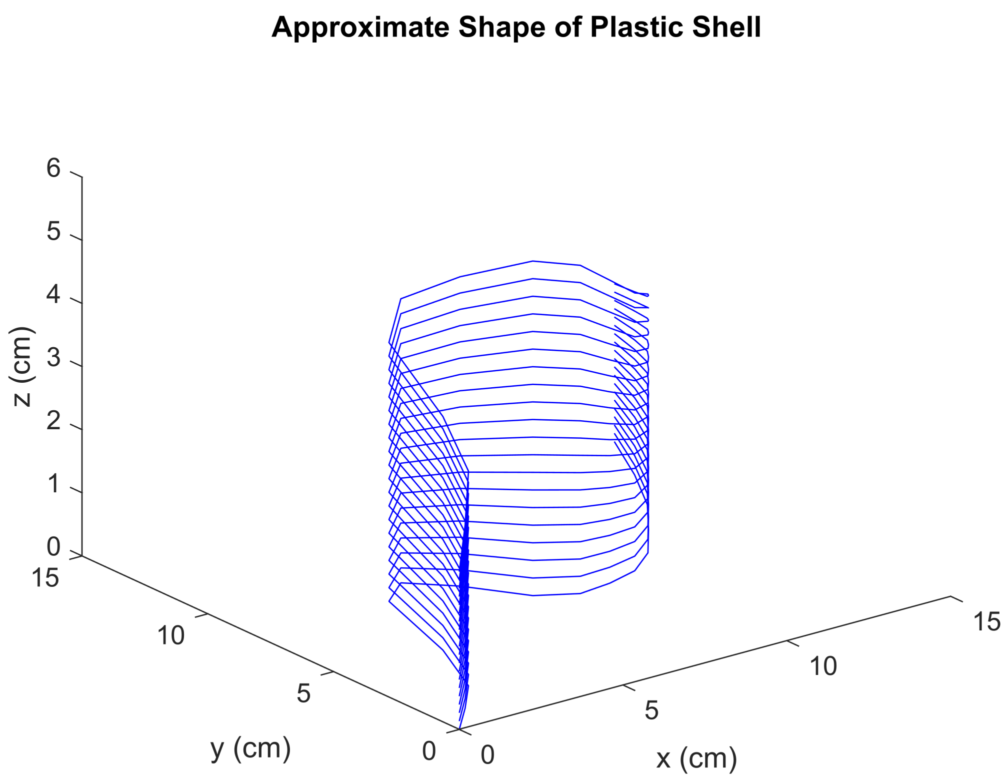
MATLAB Project 5: 3D Printing

Due Monday, Dec 2

```
% Load in measurements and plot the shell
%load measurements.mat
x = measurements(1,:);
y = measurements(2,:);
z = measurements(3,:);
plotShell(x,y,z)
```



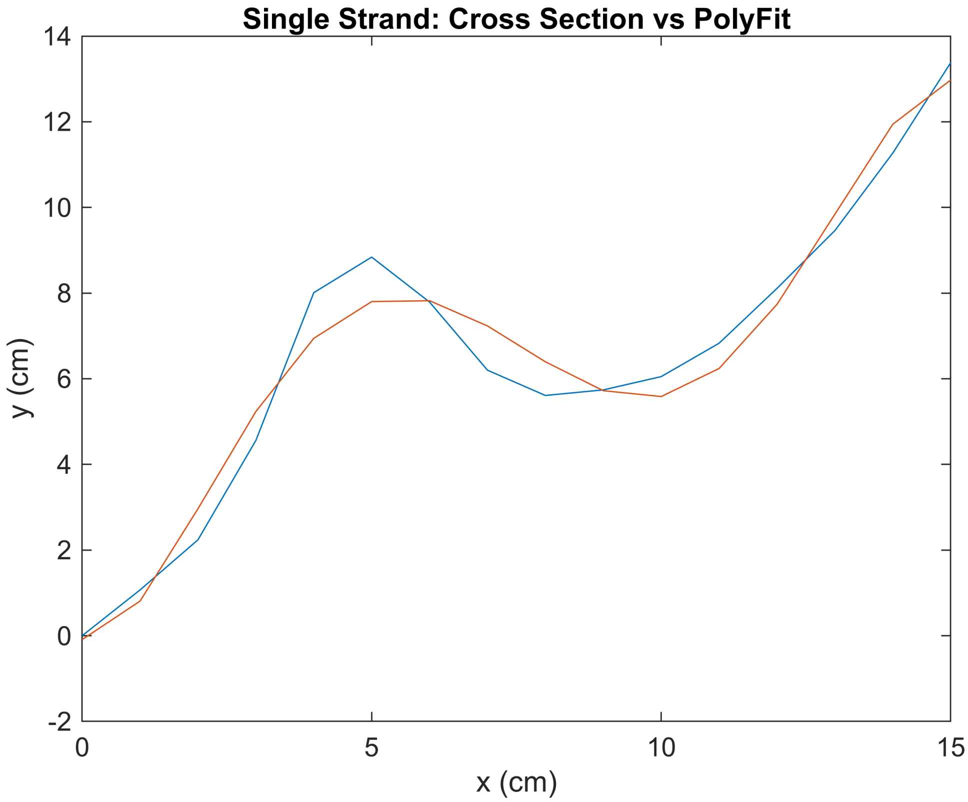

```
% Code for #1
density=1.25;
area=0.05^2*pi;
rowLength=size(measurements, 2);
totalArcLength=0;
for i=1:rowLength-1
    totalArcLength=totalArcLength+((x(i)-x(i+1))^2 + (y(i)-y(i+1))^2)^0.5;
end
totalArcLength
```

```
totalArcLength =
25.9418
```

```
mass=totalArcLength*density*area
```

```
mass =
0.2547
```

```
% Code for #2
figure
p=polyfit(x, y, 5);
q=polyder(p);
Y=polyval(p, x);
plot(x, y)
hold on
plot(x, Y)
hold off
xlim([0, 15])
xlabel('x (cm)')
ylabel('y (cm)')
title('Single Strand: Cross Section vs PolyFit')
```




```
% Code for #3
figure
r = @(t) [t, -0.0007*t^5 + 0.0266*t^4 - 0.3325*t^3 + 1.4505*t^2 -0.2444*t -0.0901];
dr= @(t) [1, q(1)*t^4 + q(2)*t^3 + q(3)*t^2 + q(4)*t + q(5)];
drMag = @(t) (1 + (q(1)*t.^4 + q(2)*t.^3 + q(3)*t.^2 + q(4)*t + q(5)).^2).^0.5;
newLength=integral(drMag, 0, 15)
```

```
newLength =
24.4968
```

```
newMass=newLength*density*area
```

```
newMass =
0.2405
```

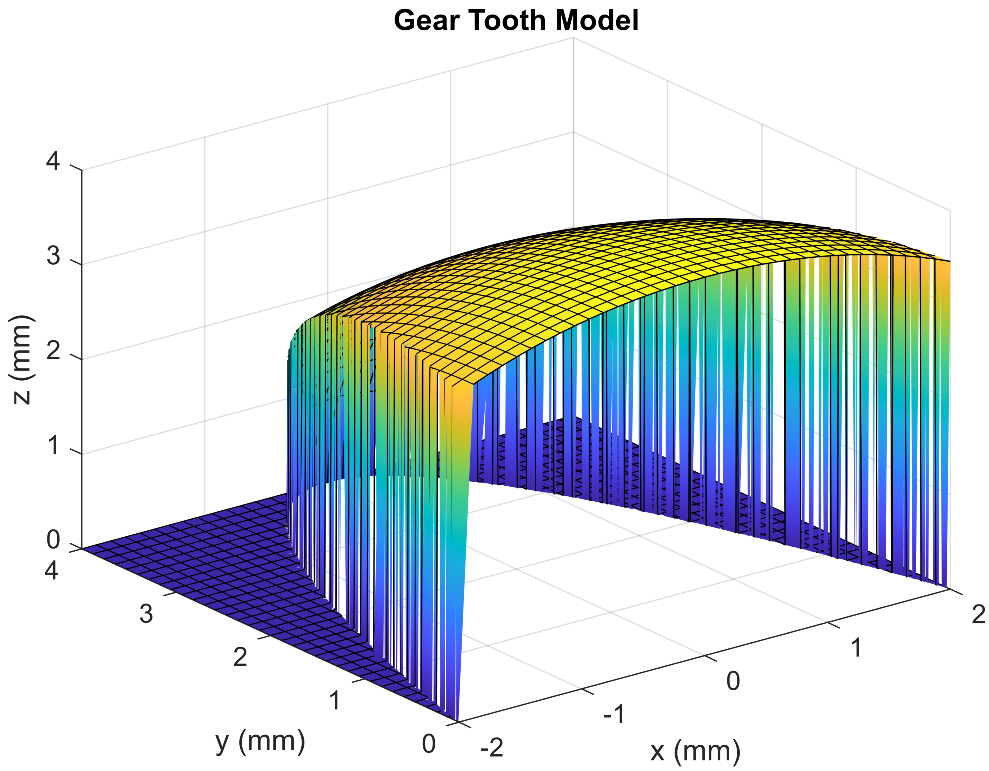
3. Fitting a polynomial to the data should give a better estimate of the true mass of the strand. When segmenting the strand, each piece of arc length (ds) is very large, meaning the approximation for the total arc length will not be as accurate. However, using a high degree polynomial reduces the size of each segment and will therefore provide a more accurate measure of total arc length. Because the arc length is more accurate when using a polynomial fit, the polynomial fit will also give a more accurate estimate of the mass of each strand.

```
% Code for #4
shellThickness=0.1;
rowLength=size(measurements, 2);
areaOfShell=0;
for i=1:rowLength-1
    areaOfShell=areaOfShell+((x(i)-x(i+1))^2 + (y(i)-y(i+1))^2)^0.5 * (z(i)
+z(i+1))/2;
end
newMass=density*(pi/4)*shellThickness*areaOfShell
```

```
newMass =
9.8210
```

5. The previous example was an example of a scalar line integral, where we were integrating along a strand, but at each piece of strand (ds), there was some associated height value. By multiplying the height by each piece of strand and summing the result, we effectively took a line integral where ds was each segment of strand. If we wanted to do polynomial fitting, we would first fit the x and y positions, then relate the z position to the x and y positions. We can then parameterize the function of x and y into a vector valued function r(t). We would then be able to use the formula A = the integral from a to b of F(r(t)).ds where f is the function relating z to x and y, and ds is the magnitude of r'(t) dt.

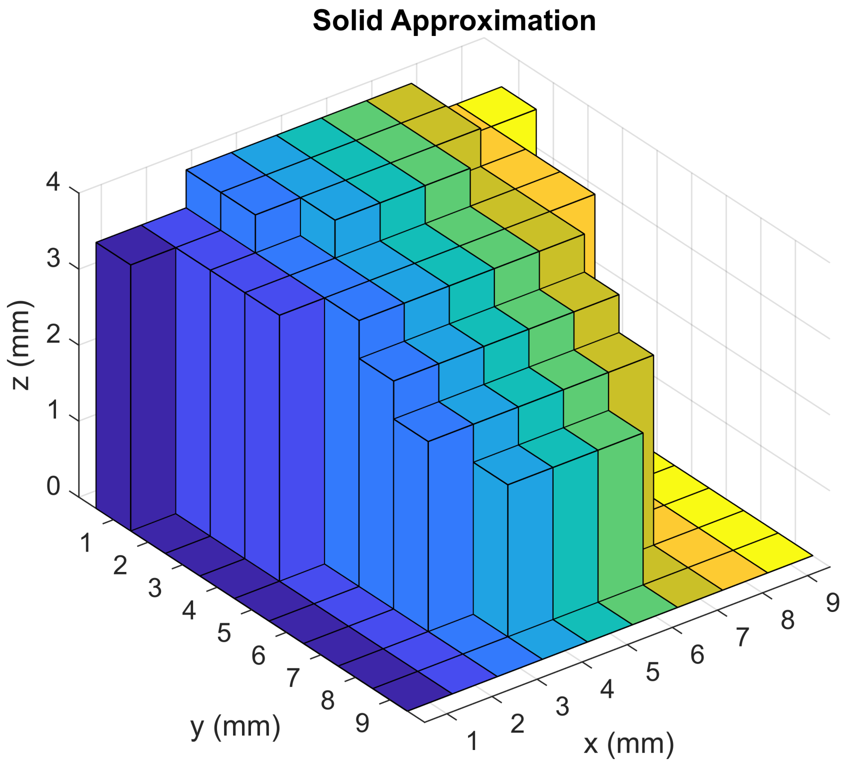
```
% Code for #6
f = @(x,y) ((16-x.^2-y.^2).^0.5).*(y>=0).*(y<=(4-x.^2));
fsurf(f, [-2, 2, 0, 4])
xlabel('x (mm)')
ylabel('y (mm)')
zlabel('z (mm)')
title('Gear Tooth Model')
```



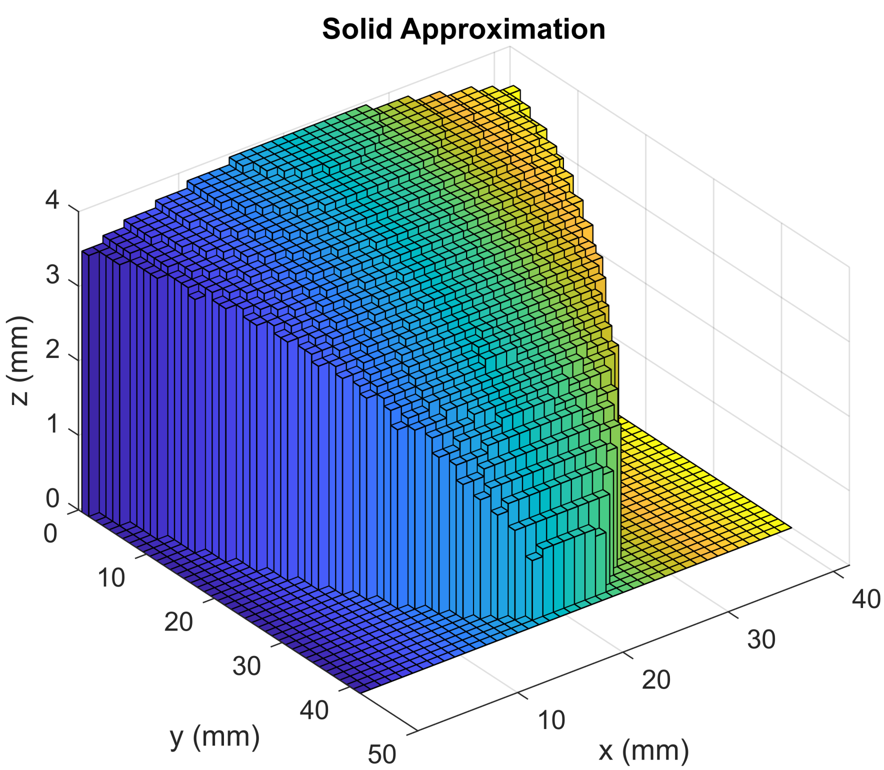

```
% Code for #7
g = @(x) 4-x.^2;
gearArea = integral2(f, -2, 2, 0, g)
```

```
gearArea =
35.6055
```

```
% Code for #8
plotSolid(f, 0.5)
```




```
plotSolid(f, 0.1)
```



```
% Code for #9
calcVolume(f, 0.5)
```

```
ans =
40.6250
```

```
calcVolume(f, 0.1)
```

```
ans =
36.4080
```

9. As the resolution approaches 0, the calculated volume of the rectangles should match the volume calculated by the integral in question 7. This is because the integral is calculated using infinitely small rectangles (resolution~0) which would match the actual volume of the gear.

Function Definitions

plotShell (already written for you)

```
function plotShell(x,y,z)
    figure
    plot3(x,y,z,'b')
    title('Approximate Shape of Plastic Shell')
    xlabel('x (cm)')
    ylabel('y (cm)')
    zlabel('z (cm)')
    hold on

    for t = linspace(0,1,20)
        plot3(x,y,t*z,'b');
    end
    hold off
end
```

plotSolid (for #8)

```
function plotSolid(f, resolution)
    figure
    [X, Y]=meshgrid(-2:resolution:2, 0:resolution:4);
    Z=round(f(X, Y)./resolution).*resolution;
    bar3(Z, 1)
    title('Solid Approximation')
    xlabel('x (mm)')
    ylabel('y (mm)')
    zlabel('z (mm)')
end
```

calcVolume (for #9)

```
function volume = calcVolume(f, resolution)
    volume=0;
```

```
    [X, Y]=meshgrid(-2:resolution:2, 0:resolution:4);
    Z=round(f(X, Y)./resolution).*resolution;
    for row = 1:size(Z,1)
        for col = 1:size(Z,2)
            volume = volume + Z(row, col)*resolution^2;
        end
    end
end
```