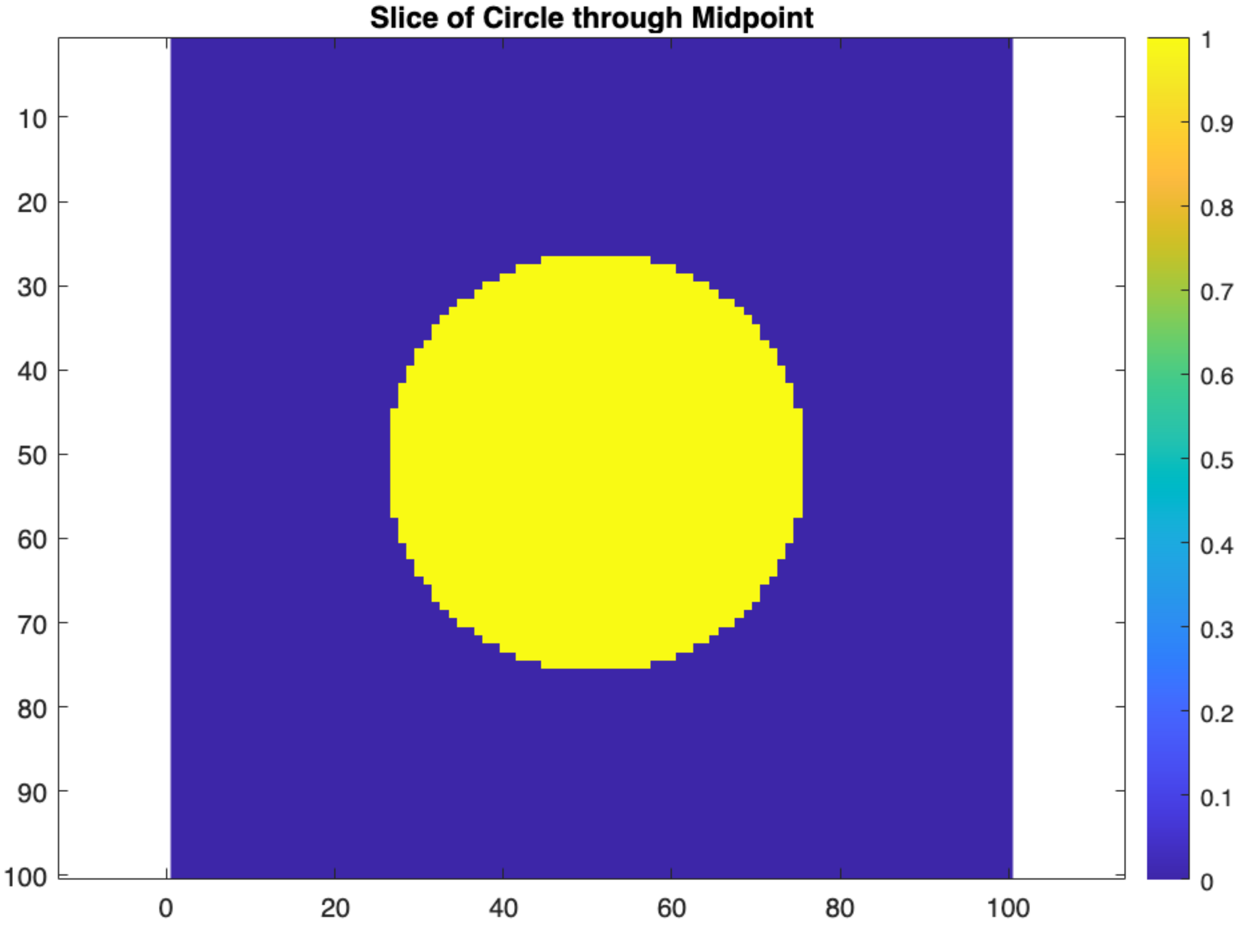
Part 1

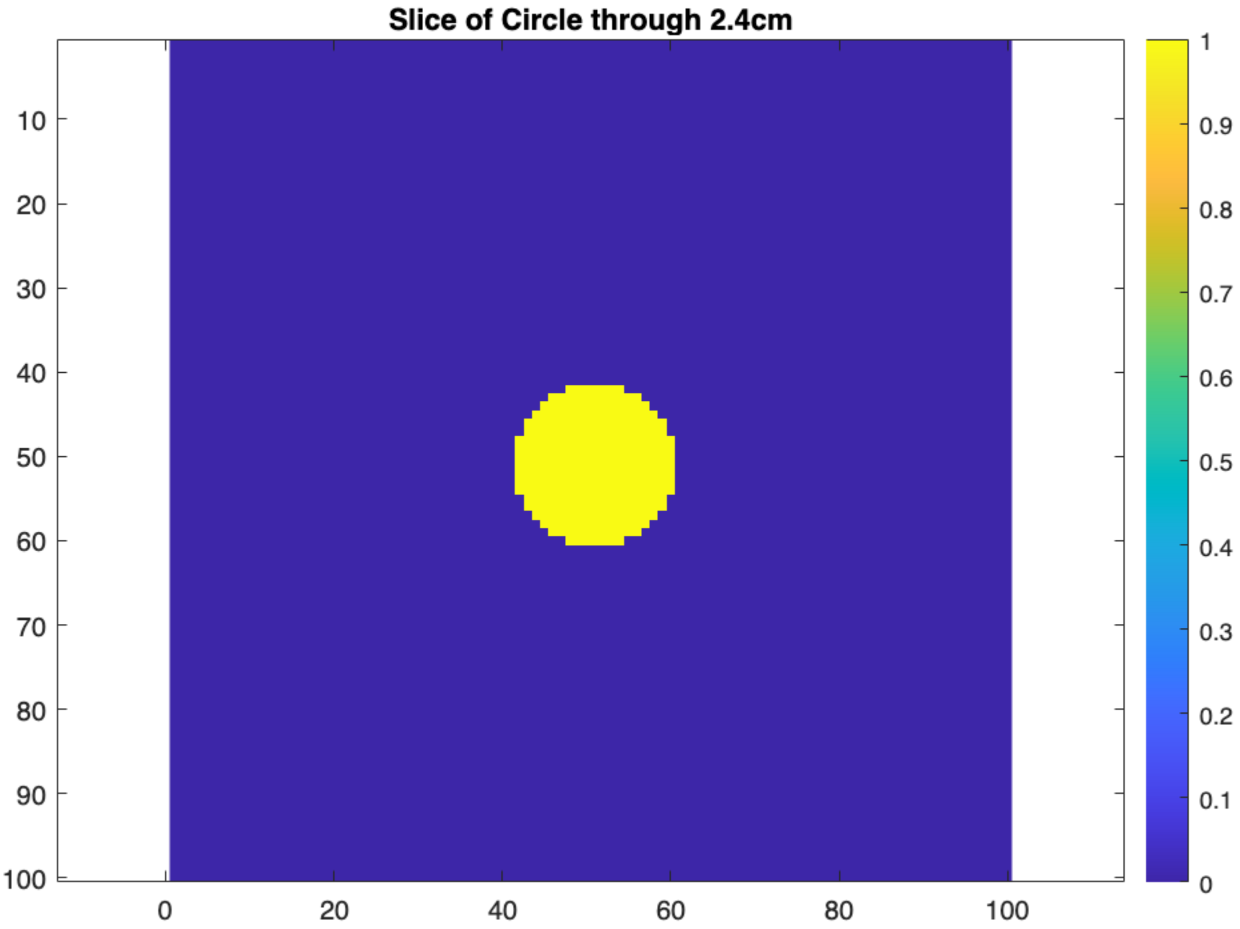
```
n = 100; %1mm^3 pixel sizes I believe if cube size is 10cm.
I = zeros(n,n,n); %creating 0 matrix of large sampling size

dx = 0.1;
x = -5:dx:5-dx; %5-dx is needed to make vector size fit 100x100x100
y = x;
z = x;

[X, Y, Z] = meshgrid(x,y,z); %combines xyz vectors to make 3D matrix


distance = sqrt(X.^2 + Y.^2 +Z.^2); %assuming this creates a matrix describing the distance of each coordinate from the center
mask = distance<2.5; %mask takes every distance and gives a voxel value of 1 only to those within a 2.5cm radius of the center
I(mask) = 1; %applies the mask to our zeros matrix
I(50,50,50) %Proof that the voxels are labeled 1, correctly.
```

```
ans = 1
```

```
figure, imagesc(squeeze(I(:,:,50))), axis equal , colorbar,...
    title("Slice of Circle through Midpoint")%visualizes the circle with z = 0cm.
```



```
imwrite(squeeze(I(:,:,50)),"Slice 1.png");
figure, imagesc(squeeze(I(:,:,(24+50)))), axis equal, colorbar,...
    title("Slice of Circle through 2.4cm")
```

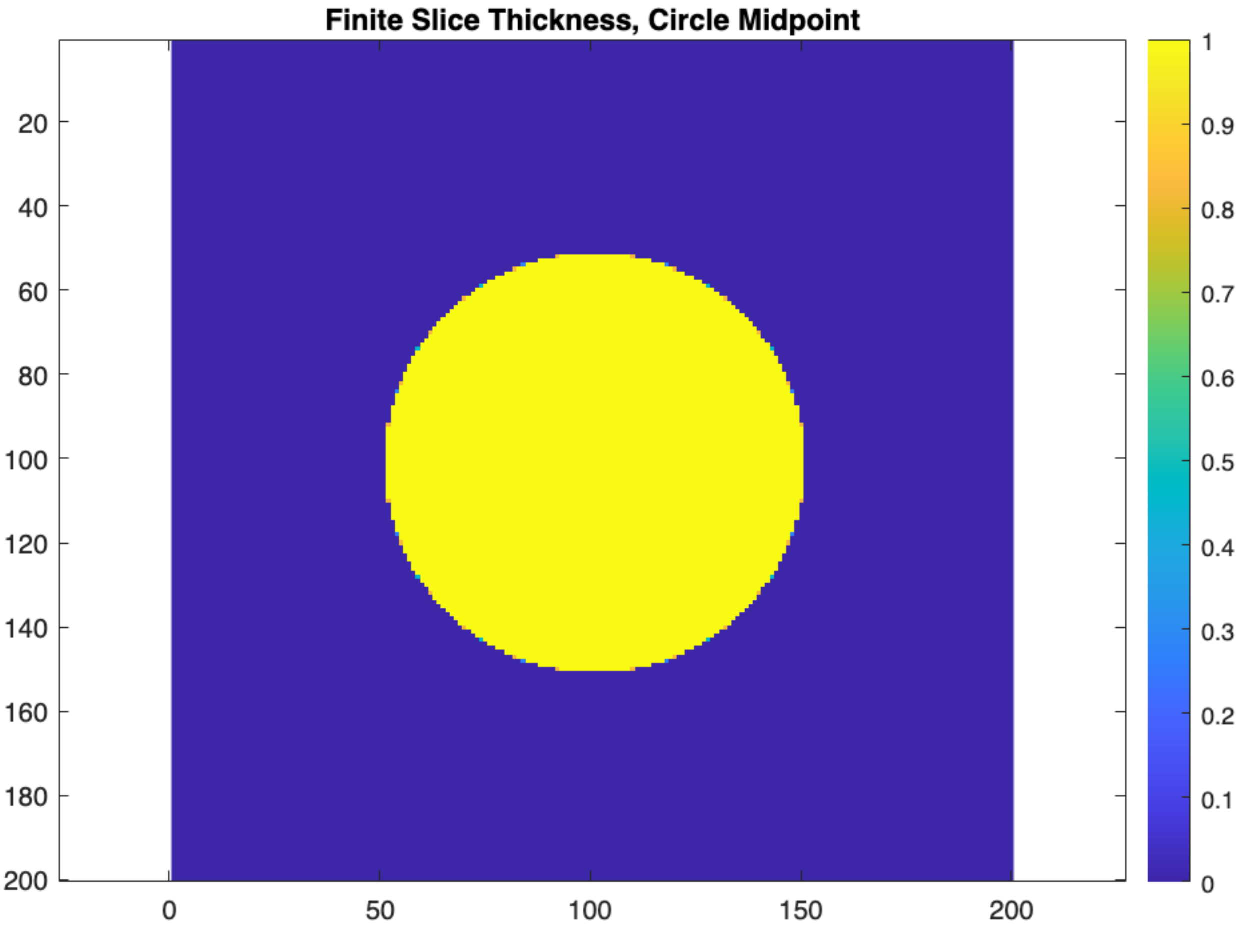


```
%visuazlizes the circle with z = 2.4cm; if 50 is the center adding 24 is adding 2.4
imwrite(squeeze(I(:,:,74)),"Slice 2.png");
```

Part 2.1 Simulating Finite Slice Thickness (Sphere Midpoint)

```
%Instead of last time, I'm going to use the zeros function with the size X
%after I use the meshgrid function. Which should make a compatable Image
%matrix instead of overthinking it (which I definitely am at this point).


dx = 0.05; %0.5mm resolution we think
x = -5:dx:5-dx;
y = x;
z = x;
[X, Y, Z] = meshgrid(x,y,z);

I = zeros(size(X));

distance = sqrt(X.^2 + Y.^2 +Z.^2);
mask = distance<2.5; %circle creation again
I(mask) = 1;


%Simulating finite slice thickness:
%Updated plan: Previously I was integrating across the entire image,
%instead of within one .5cm slice (essentially ignoring the entire
%exercise). Now, I will integrate across one slice, with 11 indexes at the
%midpoint and at the edge.

slice_number = -0.25:.05:0.25; %gives me equal slice positions
slice_int = zeros(size(I)); %creates matrix for the averaged slices

for i = 1:numel(slice_number) %loop to add average each slice at the desired thickness
```

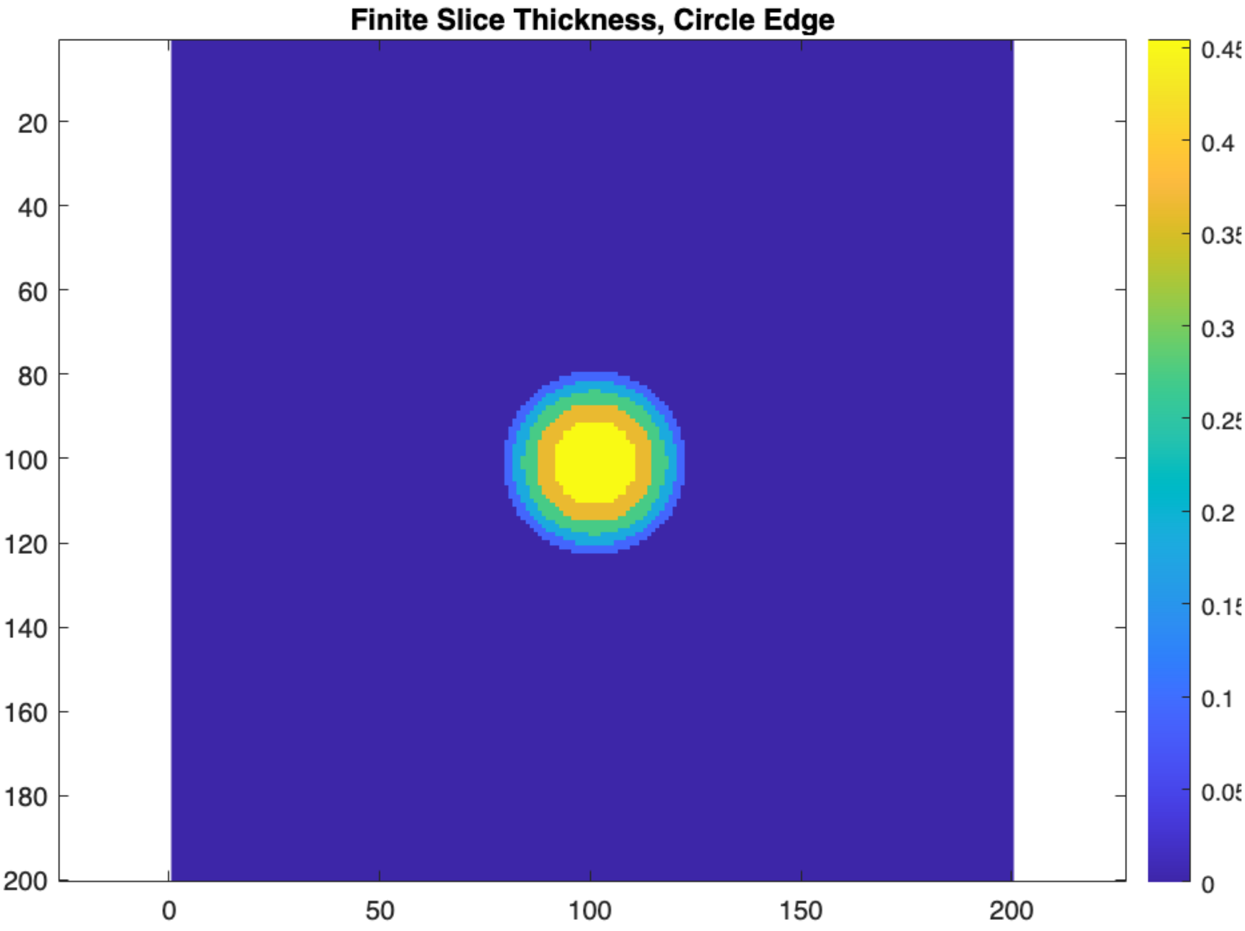
```
[~,J] = min(abs(z-slice_number(i))); %creates z-index for the slice we want to add from I

slice_z = I(:,:,J);
slice_int = slice_int + slice_z;
end
slice_int = slice_int / numel(slice_number); %normalizes image?

figure, imagesc(squeeze(slice_int(:,:,100))), axis equal, colorbar, ...
    title("Finite Slice Thickness, Circle Midpoint");
```



```
imwrite(squeeze(slice_int(:,:,100)),"Slice 3.png");
```

Part 2.2 Simulating Finite Slice Thickness (Sphere Edge)

```
%recreating same experiment but at the sphere edge.
slice_number = 2.25:.05:2.75; %divides .5cm slice into 11 indexes
slice_int = zeros(size(I));

for i = 1:numel(slice_number)
[~,J] = min(abs(z-slice_number(i)));
slice_z = I(:,:,J);
slice_int = slice_int + slice_z;
end
slice_int = slice_int / numel(slice_number);

figure, imagesc(squeeze(slice_int(:,:,100))), axis equal, colorbar, ...
    title("Finite Slice Thickness, Circle Edge");
```

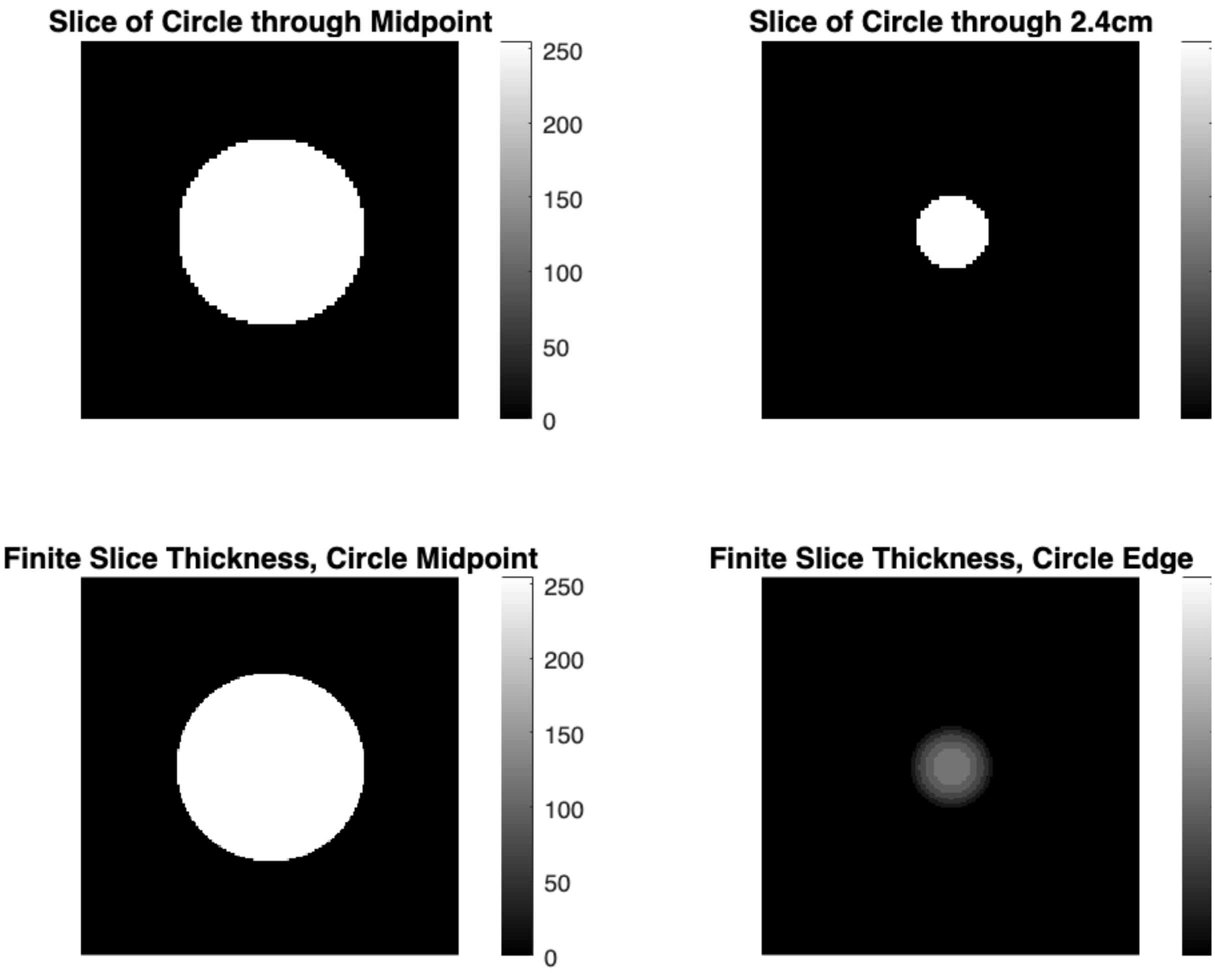


```
imwrite(squeeze(slice_int(:,:,100)),"Slice 4.png");
%Discussion: The result at the center makes essentially the same image.
%When summing and averaging the middle of the circle, most of the values
%are 1, so it makes very little change to the overall image. When
%integrating the slice at the edge, however, many 0 voxels are now included
%which will create this much more apparent blur effect.
```

Thresholding

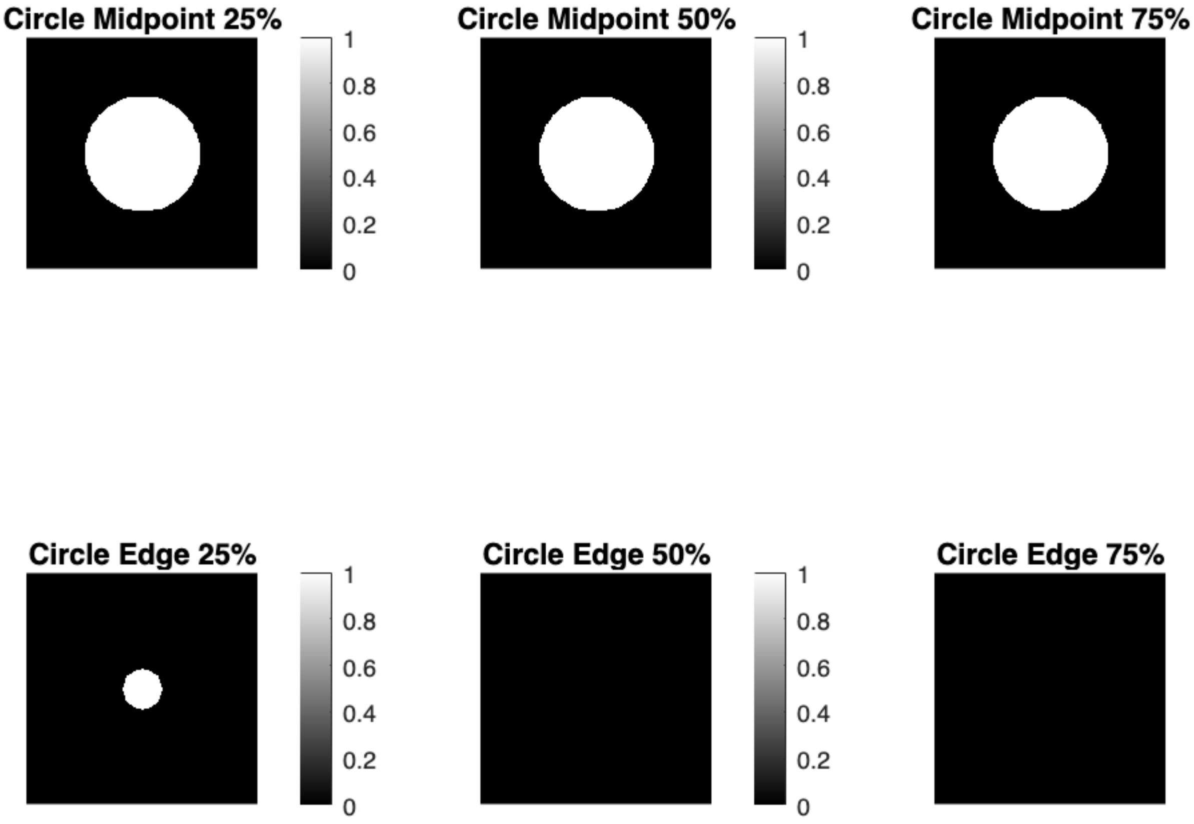
```
a = imread("Slice 1.png");
b = imread("Slice 2.png");
c = imread("Slice 3.png");
d = imread("Slice 4.png");

tiledlayout(2,2);
nexttile
imshow(a)
title("Slice of Circle through Midpoint")
colorbar

nexttile
imshow(b)
title("Slice of Circle through 2.4cm")
colorbar

nexttile
imshow(c)
colorbar
title("Finite Slice Thickness, Circle Midpoint")

nexttile
imshow(d)
```

```
colorbar
title("Finite Slice Thickness, Circle Edge")
```



```
thresh_cmp1 = im2bw(c, .25);
thresh_cmp2 = im2bw(c, .5);
thresh_cmp3 = im2bw(c, .75);
thresh_ce1 = im2bw(d, .25);
thresh_ce2 = im2bw(d, .5);
thresh_ce3 = im2bw(d, .75);

tiledlayout(2,3)

nexttile
imshow(thresh_cmp1)
title("Circle Midpoint 25%")
colorbar

nexttile
imshow(thresh_cmp2)
title("Circle Midpoint 50%")
colorbar

nexttile
imshow(thresh_cmp3)
title("Circle Midpoint 75%")
colorbar

nexttile
imshow(thresh_ce1)
```

```
title("Circle Edge 25%")
colorbar

nexttile
imshow(thresh_ce2)
title("Circle Edge 50%")
colorbar

nexttile
imshow(thresh_ce3)
title("Circle Edge 75%")
colorbar
```


Circle Midpoint 25%
Circle Midpoint 50%
Circle Midpoint 75%
Circle Edge 25%
Circle Edge 50%
Circle Edge 75%