Question 2

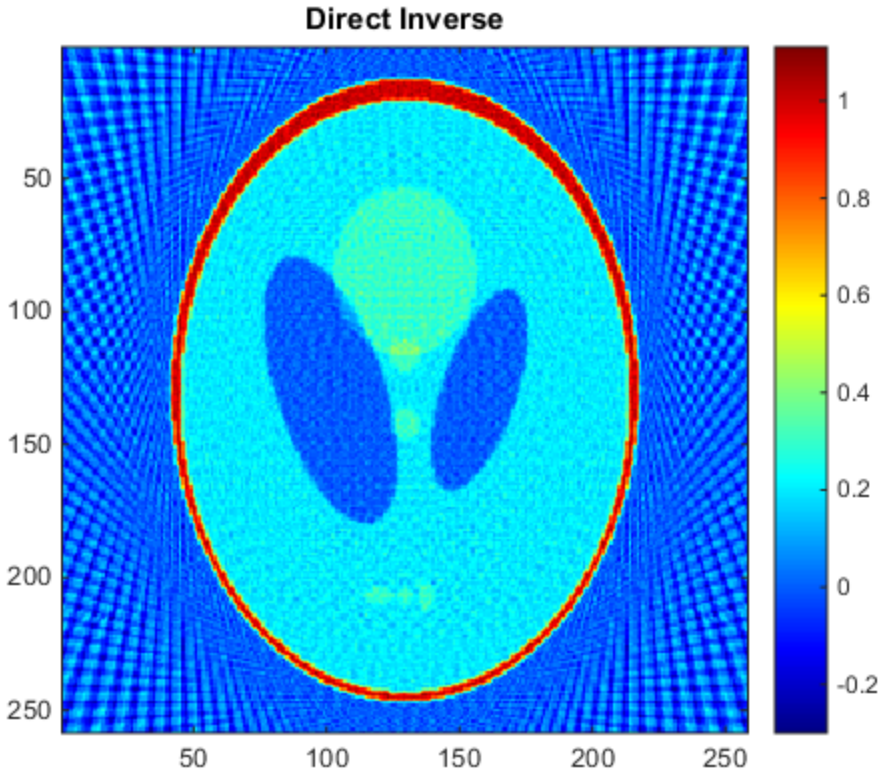
Table of Contents

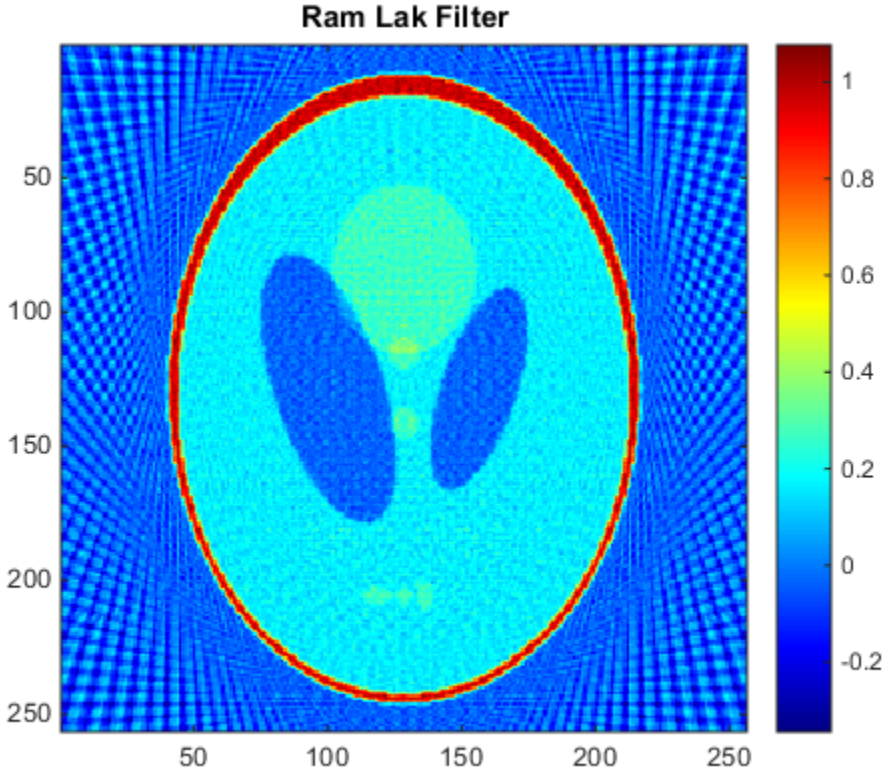
Part (a)

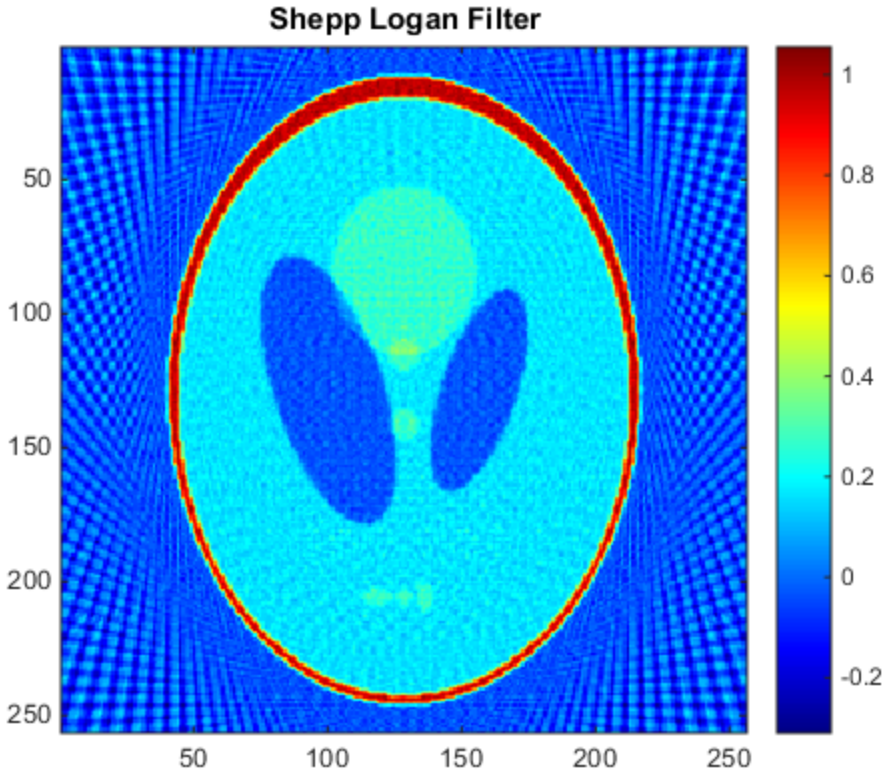
```
image_size = 256;
phantom_image = phantom(image_size);
thetas = 0:3:177;

phantom_radon = radon(phantom_image, thetas);
phantom_fft = fft(phantom_radon, [], 1);
phantom_fft = fftshift(phantom_fft, 1);

w_max = floor(size(phantom_fft, 1) / 2);
L1 = w_max;
L2 = w_max / 2;
```

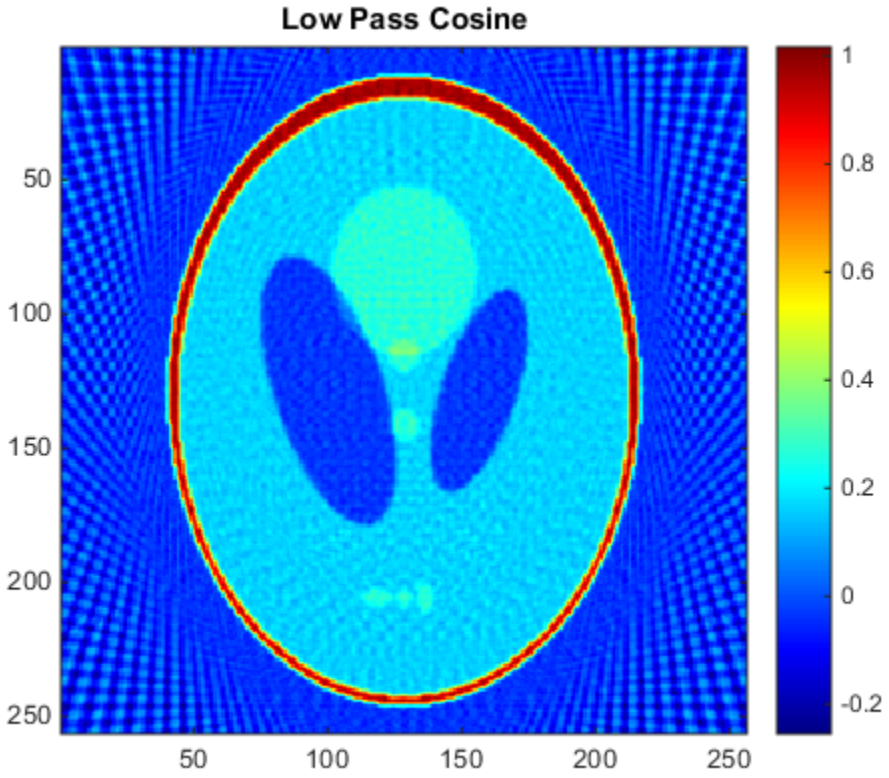
L1 = w_max

```
% Direct Inverse
direct_inverse = iradon(phantom_radon, thetas);
showImage(direct_inverse, 'Direct Inverse');

% Ram Lak Filter
phantom_ram_lak_fft = applyRamLak(phantom_fft, L1);
ram_lak_inverse = getImageFromFiltered(phantom_ram_lak_fft, ...
    thetas, image_size);
showImage(ram_lak_inverse, 'Ram Lak Filter');

% Shepp Logan Filter
phantom_shepp_logan_fft = applySheppLogan(phantom_fft, L1);
shepp_logan_inverse = getImageFromFiltered(phantom_shepp_logan_fft, ...
    thetas, image_size);
showImage(shepp_logan_inverse, 'Shepp Logan Filter');

% Low Pass Cosine Filter
phantom_low_pass_cosine_fft = applyLowPassCosine(phantom_fft, L1);
low_pass_cosine_inverse = getImageFromFiltered( ...
    phantom_low_pass_cosine_fft, thetas, image_size);
showImage(low_pass_cosine_inverse, 'Low Pass Cosine');
```


Direct Inverse
50
100
150
200
250
50
100
150
200
250
1
0.8
0.6
0.4
0.2
0
-0.2

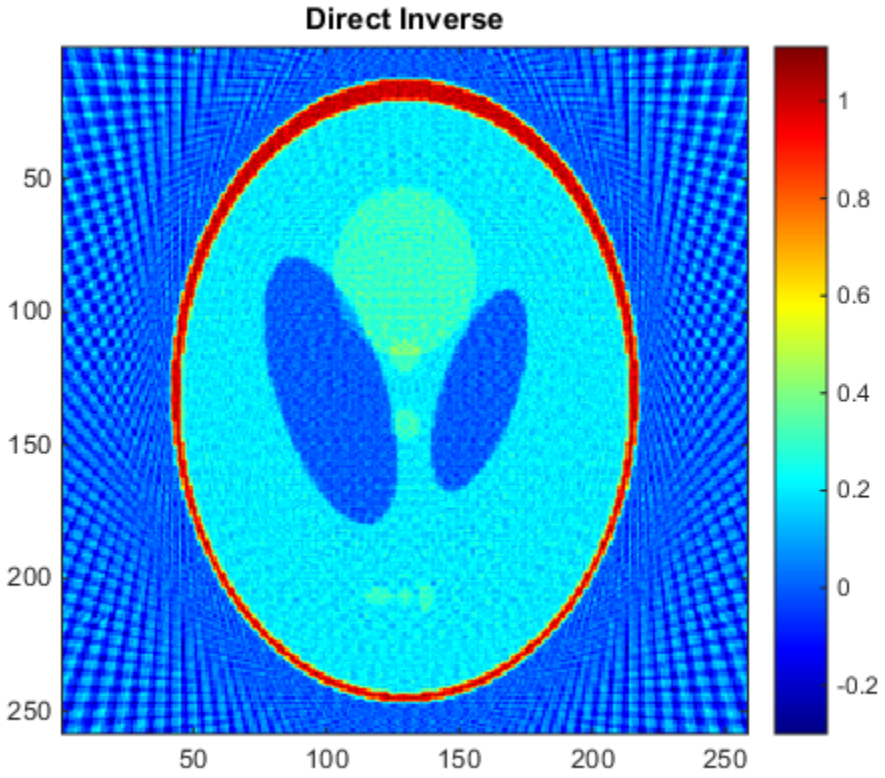

Ram Lak Filter
50
100
150
200
250
50
100
150
200
250
1
0.8
0.6
0.4
0.2
0
-0.2

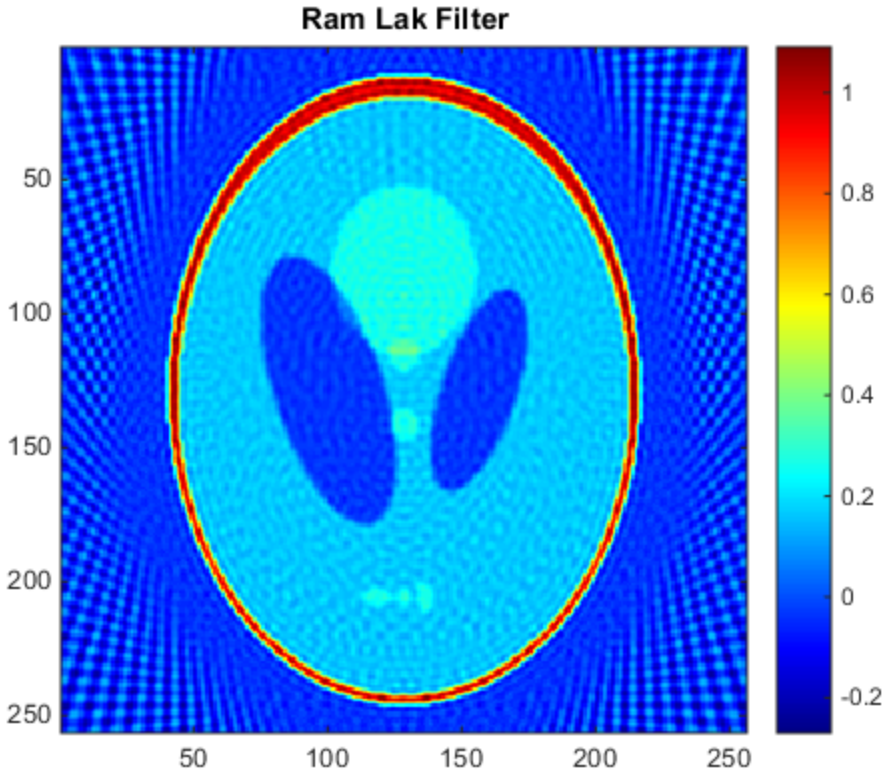

Shepp Logan Filter
50
100
150
200
250
50
100
150
200
250
1
0.8
0.6
0.4
0.2
0
-0.2

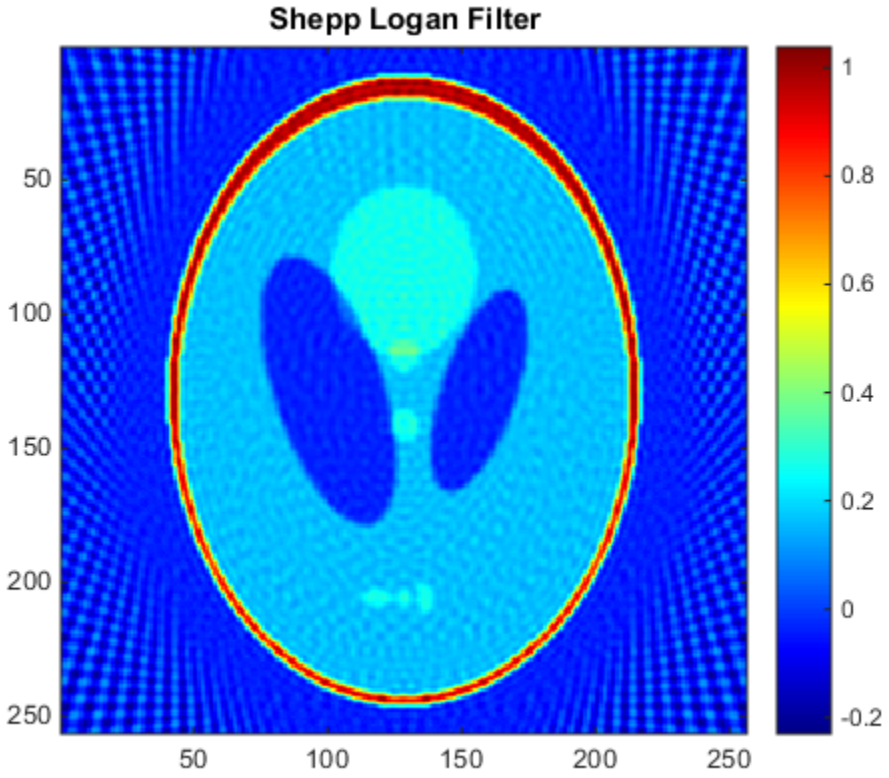

Low Pass Cosine
50
100
150
200
250
50
100
150
200
250
1
0.8
0.6
0.4
0.2
0
-0.2

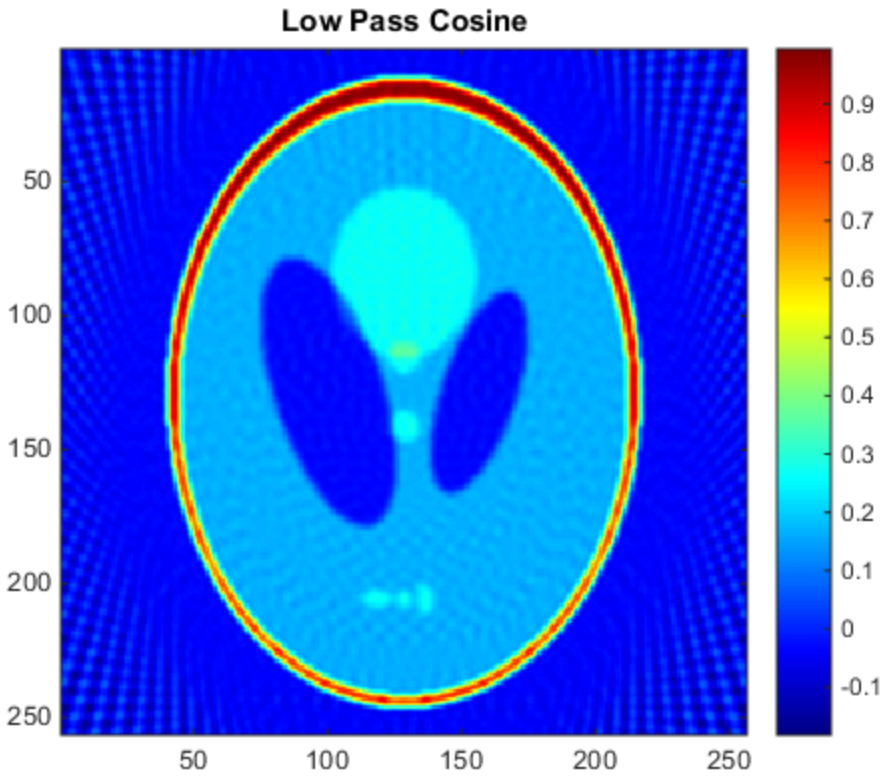
L2 = w_max / 2

```
% Direct Inverse
direct_inverse = iradon(phantom_radon, thetas);
showImage(direct_inverse, 'Direct Inverse');

% Ram Lak Filter
phantom_ram_lak_fft = applyRamLak(phantom_fft, L2);
ram_lak_inverse = getImageFromFiltered(phantom_ram_lak_fft, ...
    thetas, image_size);
showImage(ram_lak_inverse, 'Ram Lak Filter');

% Shepp Logan Filter
phantom_shepp_logan_fft = applySheppLogan(phantom_fft, L2);
shepp_logan_inverse = getImageFromFiltered(phantom_shepp_logan_fft, ...
    thetas, image_size);
showImage(shepp_logan_inverse, 'Shepp Logan Filter');

% Low Pass Cosine Filter
phantom_low_pass_cosine_fft = applyLowPassCosine(phantom_fft, L2);
low_pass_cosine_inverse = getImageFromFiltered( ...
    phantom_low_pass_cosine_fft, thetas, image_size);
showImage(low_pass_cosine_inverse, 'Low Pass Cosine');
```




Ram Lak Filter
50
100
150
200
250
50
100
150
200
250
1
0.8
0.6
0.4
0.2
0
-0.2


Shepp Logan Filter
50
100
150
200
250
50
100
150
200
250
1
0.8
0.6
0.4
0.2
0
-0.2



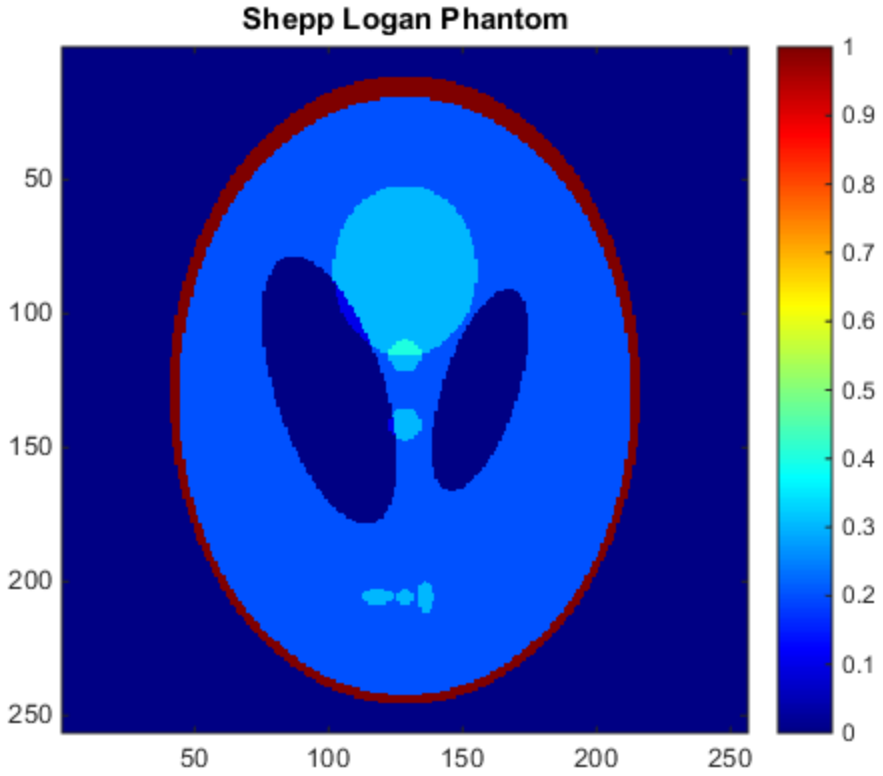
Part (b)

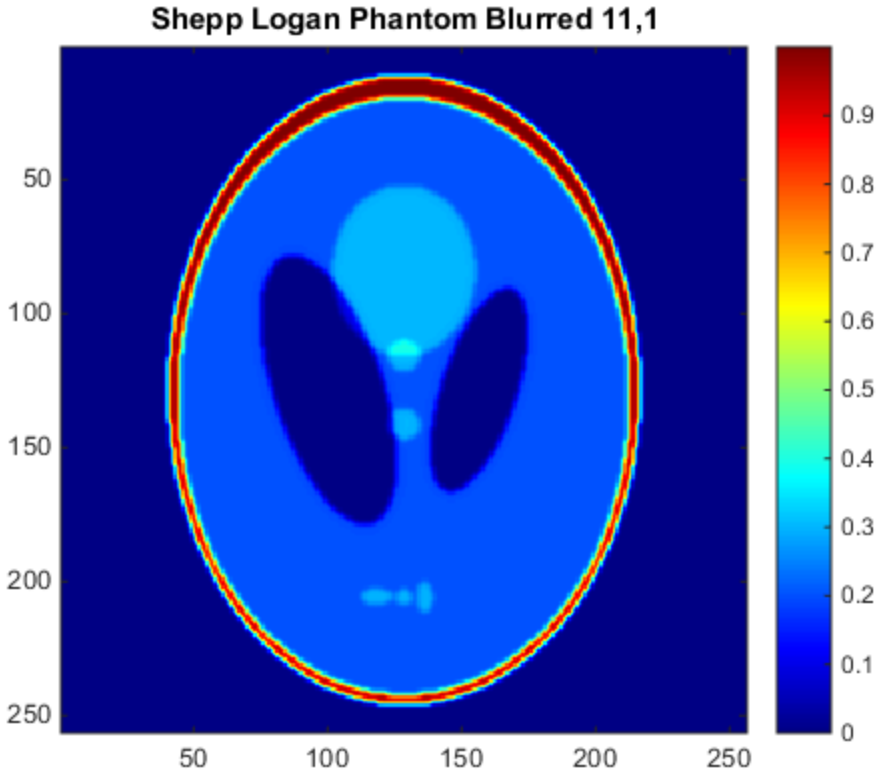
The RRMSE for the most blurred is the least because the image is blurred the noise is minimised. However the reconstruction is not substantially better as the RamLak Filter create sharpening which the blurry image is not wanting.

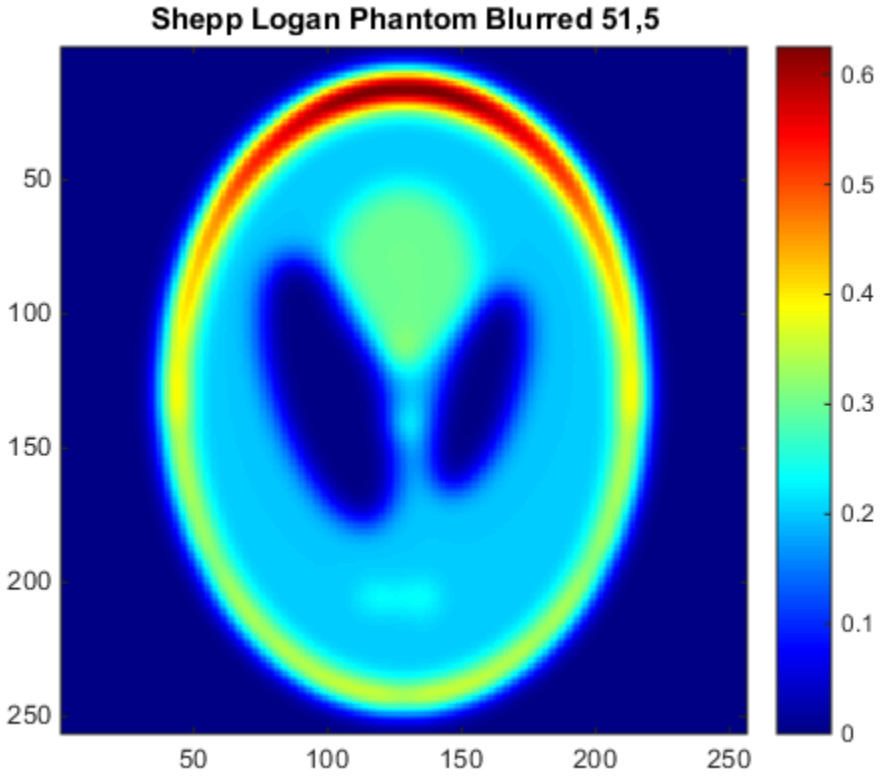
```
phantom_image = phantom(image_size);
mask = fspecial ('gaussian', 11, 1);
blurred_image_a = conv2 (phantom_image, mask, 'same');

mask = fspecial ('gaussian', 51, 5);
blurred_image_b = conv2 (phantom_image, mask, 'same');

showImage(phantom_image, 'Shepp Logan Phantom');
showImage(blurred_image_a, 'Shepp Logan Phantom Blurred 11,1');
showImage(blurred_image_b, 'Shepp Logan Phantom Blurred 51,5');

thetas = 0:3:177;

rrmse_phantom = rrmseForPartB(phantom_image, thetas, ...
    'Shepp Logan Phantom Reconstructed', image_size);
rrmse_phantom_blurred_a = rrmseForPartB(blurred_image_a, ...
    thetas, 'Shepp Logan Phantom Blurred 11,1 Reconstructed', image_size);
rrmse_phantom_blurred_b = rrmseForPartB(blurred_image_b, ...
    thetas, 'Shepp Logan Phantom Blurred 51,5 Reconstructed', image_size);
```

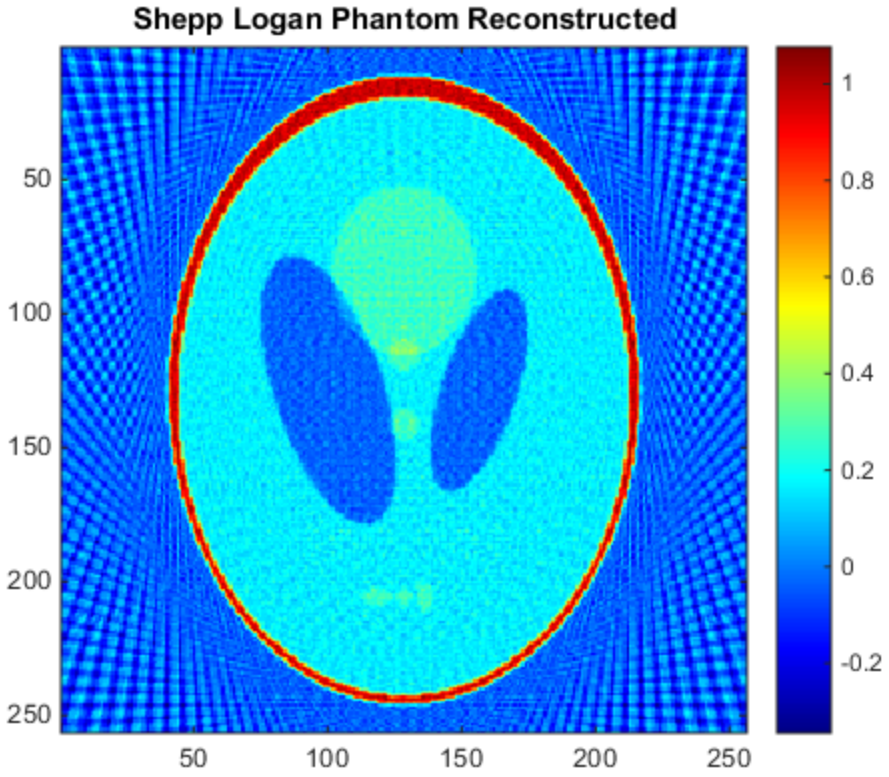
```
display(sprintf('RRMSE for default phantom %f', ...
    rrmse_phantom));
display(sprintf('RRMSE for blurred (11,1) phantom %f', ...
    rrmse_phantom_blurred_a));
display(sprintf('RRMSE for even blurred (51,5) phantom %f', ...
    rrmse_phantom_blurred_b));
```

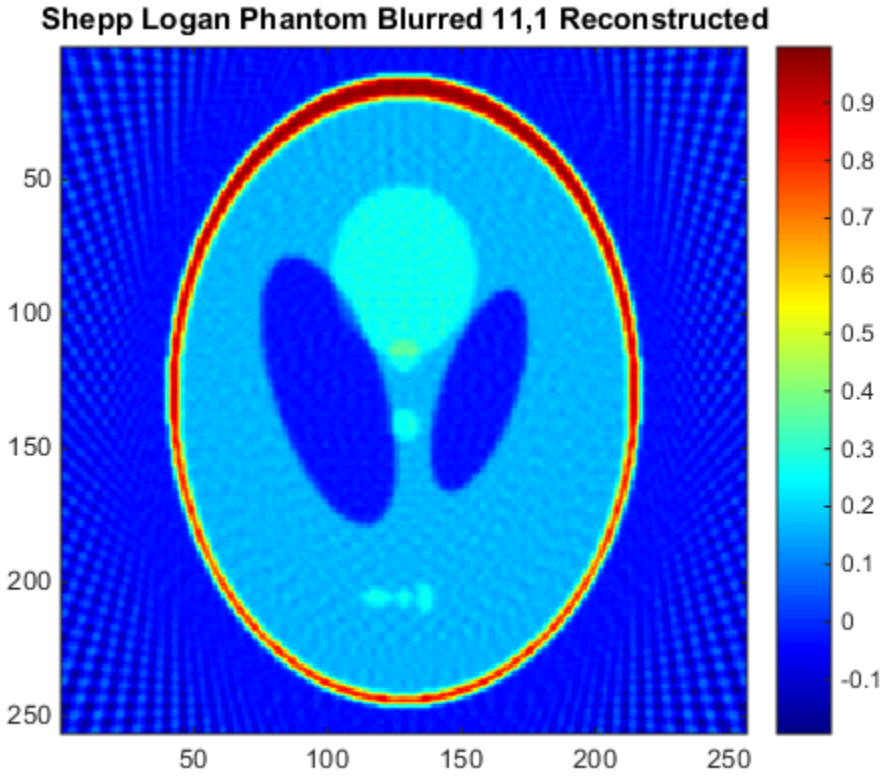
```
RRMSE for default phantom 0.326902
RRMSE for blurred (11,1) phantom 0.206541
RRMSE for even blurred (51,5) phantom 0.203948
```

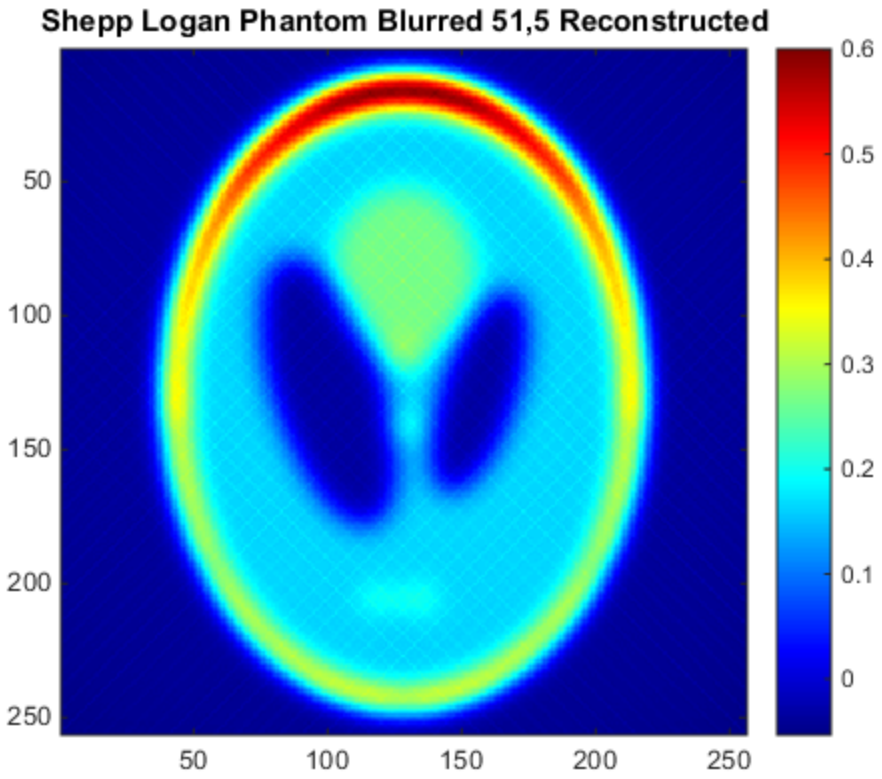



Shepp Logan Phantom Blurred 11,1
50
100
150
200
250
50
100
150
200
250
0.9
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0


Shepp Logan Phantom Blurred 51,5
50
100
150
200
250
50
100
150
200
250
0.6
0.5
0.4
0.3
0.2
0.1
0


Shepp Logan Phantom Reconstructed
50
100
150
200
250
50
100
150
200
250
1
0.8
0.6
0.4
0.2
0
-0.2

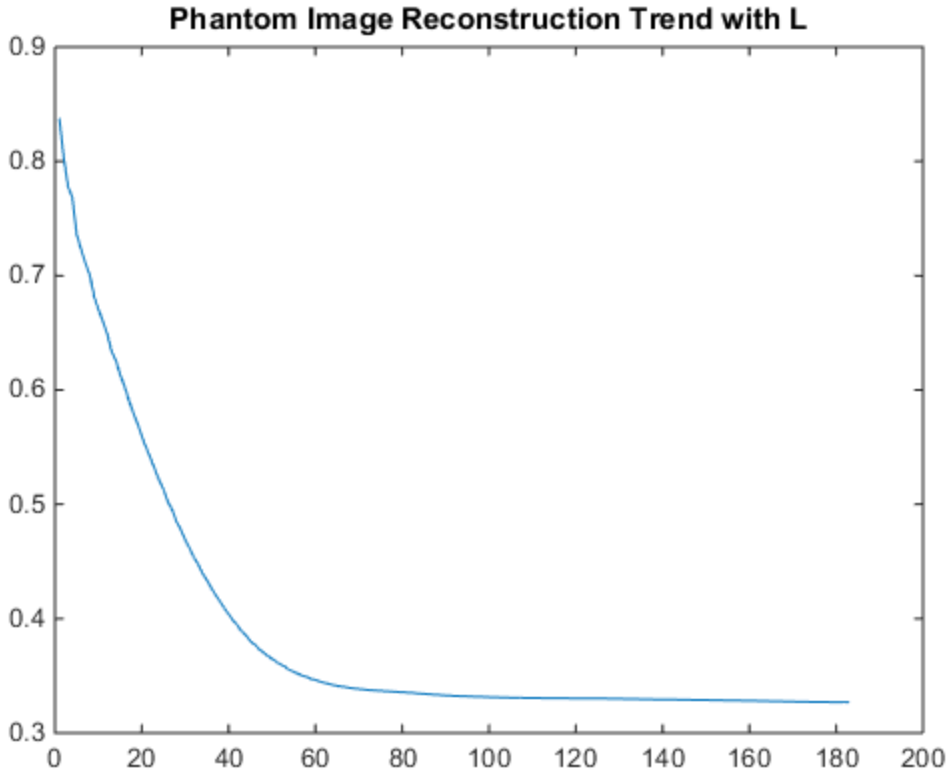

Shepp Logan Phantom Blurred 11,1 Reconstructed
50
100
150
200
250
50
100
150
200
250
0.9
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0
-0.1



Part (c)

The trend for all the three is such that the RMSE reduces as L increases till it becomes steady. This is because initially we are removing important frequency components from the source. As the image is mostly noise free, the increase after L increases is not visible.

```
phantom_image = phantom(image_size);

mask = fspecial ('gaussian', 11, 1);
blurred_image_a = conv2 (phantom_image, mask, 'same');

mask = fspecial ('gaussian', 51, 5);
blurred_image_b = conv2 (phantom_image, mask, 'same');

thetas = 0:3:177;

plotForPartC(phantom_image, thetas, ...
    'Phantom Image Reconstruction Trend with L', image_size);
plotForPartC(blurred_image_a, thetas, ...
    'Phantom Image Blurred 11,1 Reconstruction Trend with L', image_size);
plotForPartC(blurred_image_b, thetas, ...
    'Phantom Image Blurred 51,5 Reconstruction Trend with L', image_size);
```


Phantom Image Reconstruction Trend with L
0.9
0.8
0.7
0.6
0.5
0.4
0.3
0
20
40
60
80
100
120
140
160
180
200


Phantom Image Blurred 11,1 Reconstruction Trend with L
0.9
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0
20
40
60
80
100
120
140
160
180
200

Published with MATLAB® R2014b