

```

%
%     template for BME/EECS 516 tomography project
%
%     replace all '?'s
%

%     parameters for the 3 disk phantom
%     x,y center, radius, 'amplitude' (e.g. attenuation coefficient)
clear all
clc
circ = [0 0 75 0.1; 30 30 18 0.2; -53 0 9 0.3];
nobj = size(circ,1);

%
%     image parameters
%

nx = 192; ny = 192;
dx = 1;                % 1 mm / pixel

%
%     geometry parameters
%

nr = 192;      dr = 1;      % # of radial samples, ray spacing

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%
% Question 1 - determine number of angular samples
%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% calculate the number of angular samples needed
na = 304;
r = dr*[-nr/2:nr/2-1]';
ang = [0:(na-1)]'/na * pi;      % angular sample positions
disp(sprintf('number of angles = %g', na))

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%
% Question 2 - compute sinogram for disk phantom (NO CHANGES NEEDED)
%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

rr = r(:,ones(1,na));

sinogram1 = zeros(nr, na);
for ii=1:nobj
    cx = circ(ii,1);      % center of object in x
    cy = circ(ii,2);      % center of object in y
    rad = circ(ii,3);      % radius of object
    amp = circ(ii,4);      % amplitude of object (in atten

    % correct amplitude for overlying objects
    if ii > 1, amp = amp - circ(1,4);, end

    % center location of object for each projection
    tau = cx * cos(ang) + cy * sin(ang);
    tau = tau(:,ones(1,nr))';

    % find all locations where "rr" is within "rad" of "tau"
    t = find( (rr-tau).^2 <= rad.^2 );

    % update the sinogram with length of segment (a bit of geometry...)
    sinogram1(t) = sinogram1(t)+amp*2*sqrt(rad^2-(rr(t)-tau(t)).^2);

end

%
% Output Image of Singram
%
```

```

figure(1)
close(ffigure(1))
figure(2)
imagesc(r,ang,sinogram1'); colormap('gray')
title('Sinogram of Disk Phantom')
xlabel('R(mm)')
ylabel('angular(radius)')

sinogram = sinogram1;

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%
% Question 3 - Implement plain backprojection
%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

bpimage = zeros(nx,ny);
for ia = 1:na
    %disp(sprintf('angle %g of %g', ia, na))

    % first backproject at theta = 0
    tmpim = repmat(sinogram(:,ia),1,nr);
    % now rotate the projection
    rotim = imrot3(tmpim, ang(ia,:), 'bilinear');
    bpimage = bpimage + rotim;
end

%
% Display Image
%\
dy = 1;
x = dx*[-nx / 2 : nx / 2-1];
y = dy*[-ny / 2 : ny / 2-1];
figure(3)
imagesc(x,y,bpimage'); colormap('gray'); axis('image');axis('xy');
title('Simple Backprojection Image')
xlabel('x(mm)')
ylabel('y(mm)')

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%
% Question 4 - Filter Projections
%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

% zero pad sinogram (if using Fourier methods)

% filter the sinogram
% !!!Warning - fftshift works differently for vectors and matrices!!

%
% Plot Filtered Sinogram
%
projection = zeros([nr na]);
F_projection_filtered = zeros([nr na]);
projection_filtered = zeros([nr na]);
ramp_filter = abs(r);
for ia = 1:na
    projection(:, ia) = sinogram(:, ia);
    F_projection(:, ia) = fftshift(fft(projection(:, ia))); % Move to frequency domain and shift zero frequency to center
    for ir = 1 : nr
        F_projection_filtered(ir,ia) = F_projection(ir,ia) * ramp_filter(ir,:); % Apply the filter
    end
    projection_filtered(:,ia) = ifft(ifftshift(F_projection_filtered(:,ia))); % Move back to spatial domain
end

sinogrampad = real(sinogram);
sinogramfilt = real(projection_filtered);

figure(4)

```

```

plot(r, sinogrampad(:,1)./max(sinogram(:,1)), '-',...
     r, sinogramfilt(:,1)./max(sinogramfilt(:,1)), ':');
xlabel('R(mm)');
ylabel('signal');
legend('before filtering','after filtering')

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%
% Question 5 - Backproject the filtered sinogram
%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

imagefbp = zeros(nx,ny);
for ia = 1:na
    tmpimfbp = repmat(sinogramfilt(:,ia),1,nr);
    rotimfbp = imrot3(tmpimfbp, ang(ia,:), 'bilinear');
    imagefbp = imagefbp + rotimfbp;
end

for ix = 1 : nx
    for iy = 1 : ny
        if imagefbp(ix,iy) < 0
            imagefbp(ix,iy) = 0;
        end
    end
end

%
% Display Reconstructed Image with Negatives Set to Zero
%

figure(5)
imagesc(x,y,max(imagefbp',0)); colormap('gray'); axis('image');axis('xy');
title('FBP Reconstruction Image')
xlabel('x(mm)')
ylabel('y(mm)')

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%
% Question 6 - Generate Fourier Interpolation Image
%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

% Note that one can use either interp2 or griddata
% template is for interp2, but griddata is fine
% Fourier locations in of gridded data
fov = dr*nr;
kx = 1/fov.*[-nx/2:nx/2-1];
ky = 1/fov.*[-ny/2:ny/2-1];
kr = 1/fov.*[-nr/2:nr/2-1];
[kaa,krr] = meshgrid(ang,kr);
[kxx,kyy] = meshgrid(kx,ky);
kxxin = -krr.*sin(kaa);
kyyin = krr.*cos(kaa);

sinogramft = ft(sinogram);
fdata = griddata(kxxin,kyyin,sinogramft,kxx,kyy, 'linear');
t = find(isnan(fdata));
fdata(t) = zeros(size(t));
imagefi = ift2(fdata);
imagefi = imrot3(imagefi, pi / 2, 'bilinear');

figure(6)

imagesc(x,y,abs(imagefi)); colormap('gray'); axis('image'); axis('xy')
title('FI Reconstruction Image')
xlabel('x(mm)')
ylabel('y(mm)')

```

```

% =====
% %
% % Question 7 - Generate Fourier Interpolation Image using zeropadding
% %
% =====
fov_zp = 2 * dr * nr;
kx_zp = 1/fov_zp.*[-nx:nx-1];
ky_zp = 1/fov_zp.*[-ny:ny-1];
kr_zp = 1/fov_zp.*[-nr:nr-1];
[kaa_zp,krr_zp] = meshgrid(ang,kr_zp);
[kxx_zp,kyy_zp] = meshgrid(kx_zp,ky_zp);
kxxin_zp = -krr_zp.*sin(kaa_zp);
kyyin_zp = krr_zp.*cos(kaa_zp);

sinogram_zp = padarray(sinogram, [(384-nr)/2 0], 'post');
sinogram_zp = ft(padarray(sinogram_zp, [(384-nr)/2 0], 'pre'));

fdata_zp = griddata(kxxin_zp,kyyin_zp,sinogram_zp,kxx_zp,kyy_zp,'linear');
t = find(isnan(fdata_zp));
fdata_zp(t) = zeros(size(t));
imagefizp = ift2(fdata_zp);
imagefizp = imrot3(imagefizp, pi / 2, 'bilinear');
imagefizp = imagefizp(nx / 2 : (nx / 2 + nx - 1), ny / 2 : (ny / 2 + ny - 1));

figure(7)
subplot(1,2,1)
imagesc(x,y,abs(imagefi)); colormap('gray'); axis('image'); axis('xy')
title('FI Reconstruction Image')
xlabel('x(mm)')
ylabel('y(mm)')
subplot(1,2,2)
imagesc(x,y,abs(imagefizp)); colormap('gray'); axis('image'); axis('xy')
title('FI Reconstruction Image with ZP')
xlabel('x(mm)')
ylabel('y(mm)')

% =====
% %
% % Question 8 - Plot profiles through reconstructed images
% %
% =====
%
if 1==exist('linefi')
    y = dx * ([1:ny]'- (ny+1)/2);
    linefbp = abs(imagefbp(ny/2, :));
    linefi = abs(imagefi(ny/2, :));
    linefizp = abs(imagefizp(ny/2, :));
    figure(8)
    plot(y, linefbp/max(linefbp), '-', y, linefi/max(linefi), '--', ...
        y, linefizp/max(linefizp), '-');
    xlabel('radial position (mm)');
    ylabel('projection value');
    legend('FBP image', 'FI image', 'FI ZP image')
end
%
% =====
% %
% % Question 9 - Do again for subsampled image
% %
% =====
na = na / 4;
x = dx*[-nx / 2 : nx / 2-1];
y = dy*[-ny / 2 : ny / 2-1];
sinosubsamp = sinogram(:,1:4:end);
ang_9 = ang(1:4:end,:);
projection = zeros([nr na]);
F_projection_filtered = zeros([nr na]);
projection_filtered = zeros([nr na]);
ramp_filter = abs(r);
for ia = 1:na
    projection(:, ia) = sinosubsamp(:, ia);

```

```

F_projection(:, ia) = fftshift(fft(projection(:, ia)));
for ir = 1 : nr
    F_projection_filtered(ir,ia) = F_projection(ir,ia) * ramp_filter(ir,:);
end
projection_filtered(:,ia) = ifft(ifftshift(F_projection_filtered(:,ia)));
end

sinogramfilt = real(projection_filtered);

imagefbp = zeros(nx,ny);
for ia = 1:na
    tmpimfbp = repmat(sinogramfilt(:,ia),1,nr);
    rotimfbp = imrot3(tmpimfbp, ang_9(ia,:), 'bilinear');
    imagefbp = imagefbp + rotimfbp;
end

for ix = 1 : nx
    for iy = 1 : ny
        if imagefbp(ix,iy) < 0
            imagefbp(ix,iy) = 0;
        end
    end
end

[kaa_zp,krr_zp] = meshgrid(ang_9,kr_zp);
[kxx_zp,kyy_zp] = meshgrid(kx_zp,ky_zp);
kxxin_zp = -krr_zp.*sin(kaa_zp);
kyyin_zp = krr_zp.*cos(kaa_zp);

sinogram_zp = padarray(sinosubsamp, [nr/2 0], 'post');
sinogram_zp = ft(padarray(sinogram_zp, [nr/2 0], 'pre'));

fdata_zp = griddata(kxxin_zp,kyyin_zp,sinogram_zp,kxx_zp,kyy_zp,'linear');
t = find(isnan(fdata_zp));
fdata_zp(t) = zeros(size(t));
imagefizp = ift2(fdata_zp);
imagefizp = imrot3(imagefizp, pi / 2, 'bilinear');
imagefizp = imagefizp(nx / 2 : (nx / 2 + nx -1), ny / 2 : (ny / 2 + ny -1));

figure(9)
subplot(1,2,1)
imagesc(x,y,max(imagefbp',0)); colormap('gray'); axis('image');axis('xy');
title('N_{theta} / 4 FBP Reconstruction Image')
xlabel('x(mm)')
ylabel('y(mm)')
subplot(1,2,2)
imagesc(x,y,abs(imagefizp)); colormap('gray'); axis('image'); axis('xy')
title('N_{theta} / 4 FI Reconstruction Image with ZP')
xlabel('x(mm)')
ylabel('y(mm)')

% %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% %
% % Question 10 - Do again for limited view angles
% %
% %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
sino4 = sinogram(:,1:na);
ang_10 = ang(1:na,:);
projection = zeros([nr na]);
F_projection_filtered = zeros([nr na]);
projection_filtered = zeros([nr na]);
ramp_filter = abs(r);
for ia = 1:na
    projection(:, ia) = sino4(:, ia);
    F_projection(:, ia) = fftshift(fft(projection(:, ia)));
    for ir = 1 : nr
        F_projection_filtered(ir,ia) = F_projection(ir,ia) * ramp_filter(ir,:);
    end
    projection_filtered(:,ia) = ifft(ifftshift(F_projection_filtered(:,ia)));
end

sinogramfilt = real(projection_filtered);

```

```

imagefbp = zeros(nx,ny);
for ia = 1:na
    tmpimfbp = repmat(sinogramfilt(:,ia),1,nr);
    rotimfbp = imrot3(tmpimfbp, ang_l0(ia,:), 'bilinear');
    imagefbp = imagefbp + rotimfbp;
end

for ix = 1 : nx
    for iy = 1 : ny
        if imagefbp(ix,iy) < 0
            imagefbp(ix,iy) = 0;
        end
    end
end

figure(10)
imagesc(x,y,max(imagefbp',0)); colormap('gray'); axis('image');axis('xy');
title('First 1/4 projection Image')
xlabel('x(mm)')
ylabel('y(mm)')

% %%
% %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% %
% % Question 11 - load and reconstruct mystery object
% %
% %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
load mys23;
na = 360;
projection = zeros([nr na]);
F_projection_filtered = zeros([nr na]);
projection_filtered = zeros([nr na]);
ramp_filter = abs(r);
for ia = 1:na
    projection(:, ia) = sinogram(:, ia);
    F_projection(:, ia) = fftshift(fft(projection(:, ia)));
    for ir = 1 : nr
        F_projection_filtered(ir,ia) = F_projection(ir,ia) * ramp_filter(ir,:);
    end
    projection_filtered(:,ia) = ifft(ifftshift(F_projection_filtered(:,ia)));
end
sinogrampad = real(sinogram);
sinogramfilt = real(projection_filtered);

imagefbp = zeros(nx,ny);
for ia = 1:na
    tmpimfbp = repmat(sinogramfilt(:,ia),1,nr);
    rotimfbp = imrot3(tmpimfbp, ang(:,ia), 'bilinear');
    imagefbp = imagefbp + rotimfbp;
end

for ix = 1 : nx
    for iy = 1 : ny
        if imagefbp(ix,iy) < 0
            imagefbp(ix,iy) = 0;
        end
    end
end

figure(11)
imagesc(x,y,max(imagefbp',0)); colormap('gray'); axis('image');axis('xy');
title('FBP Reconstruction Image')
xlabel('x(mm)')
ylabel('y(mm)')

figure(12)
imagesc(r,ang,sinogram); colormap('gray')
title('Sinogram')
xlabel('R(mm)')
ylabel('angular(radius)')

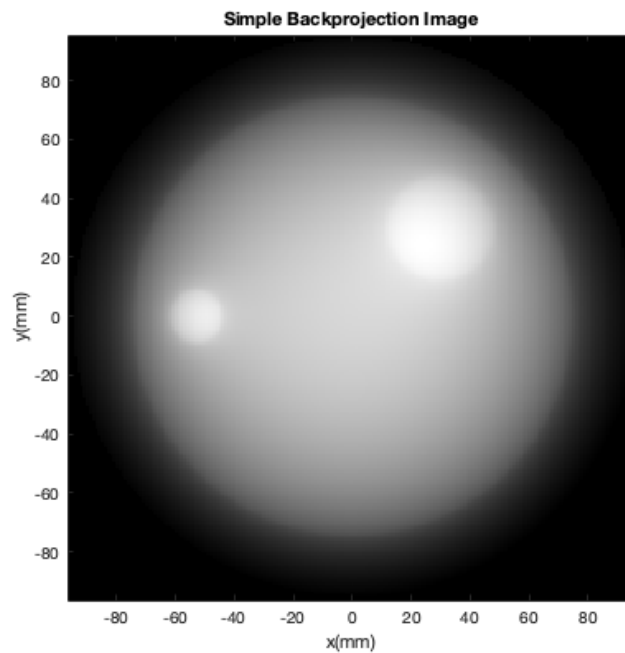
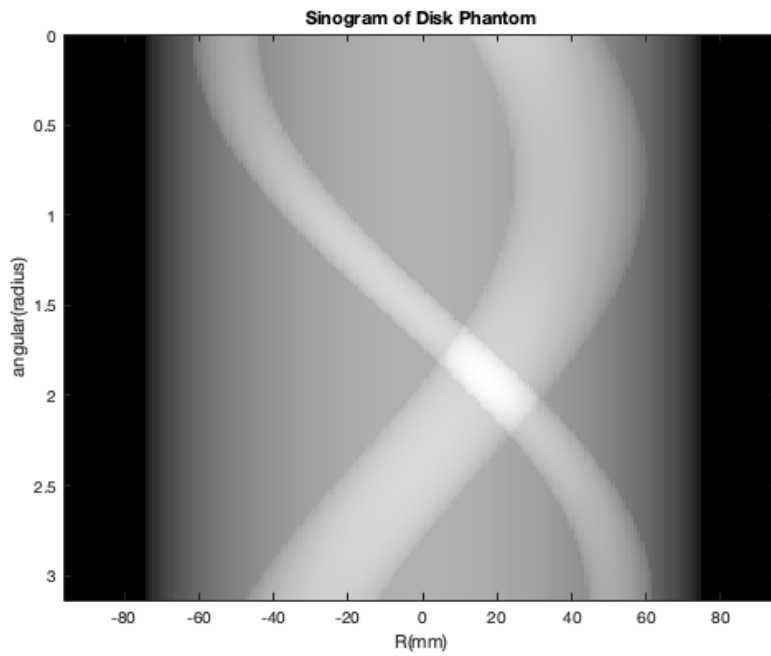
```

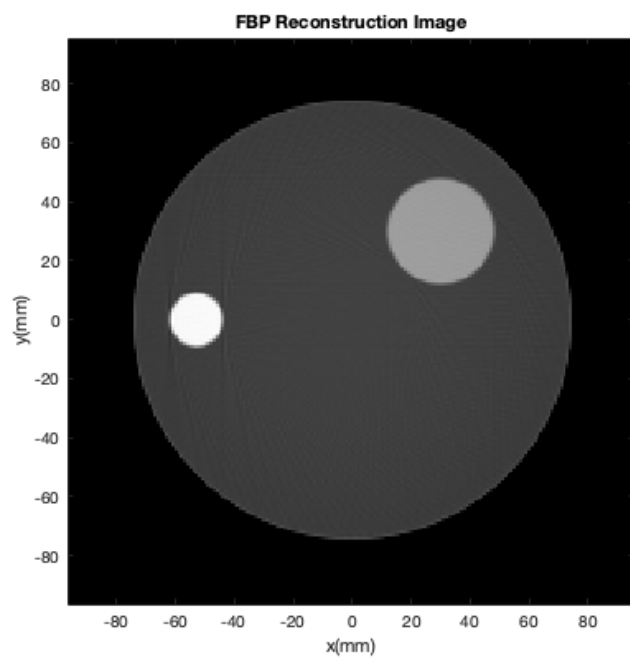
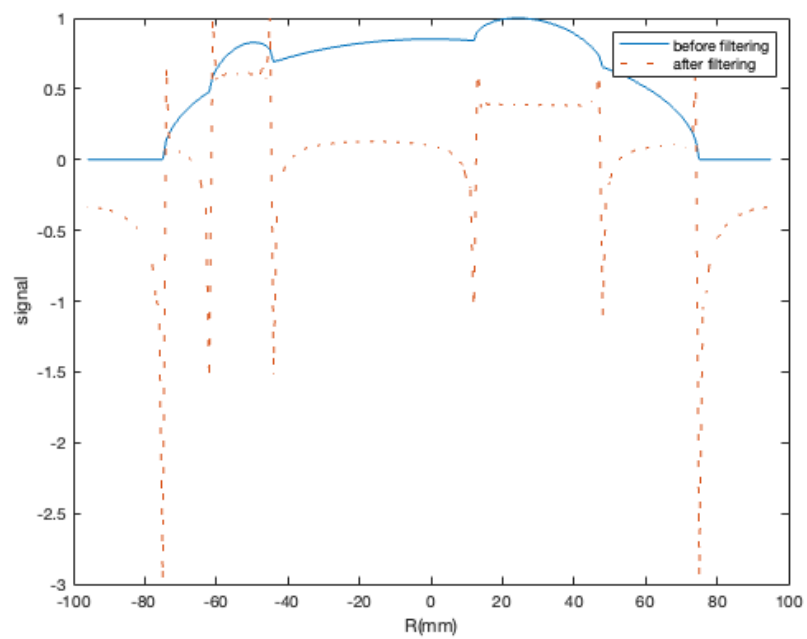
number of angles = 304

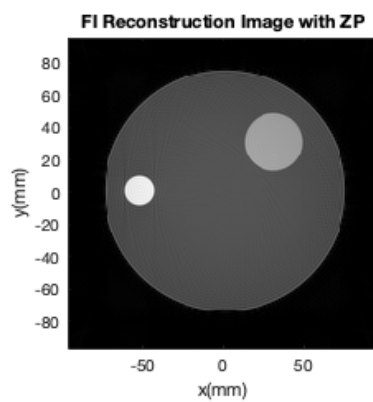
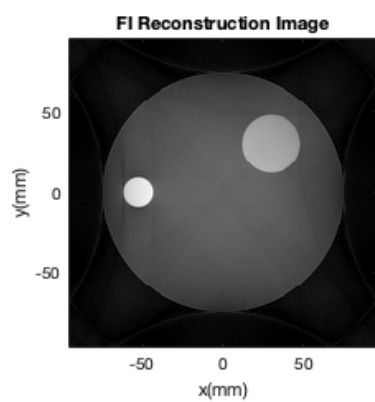
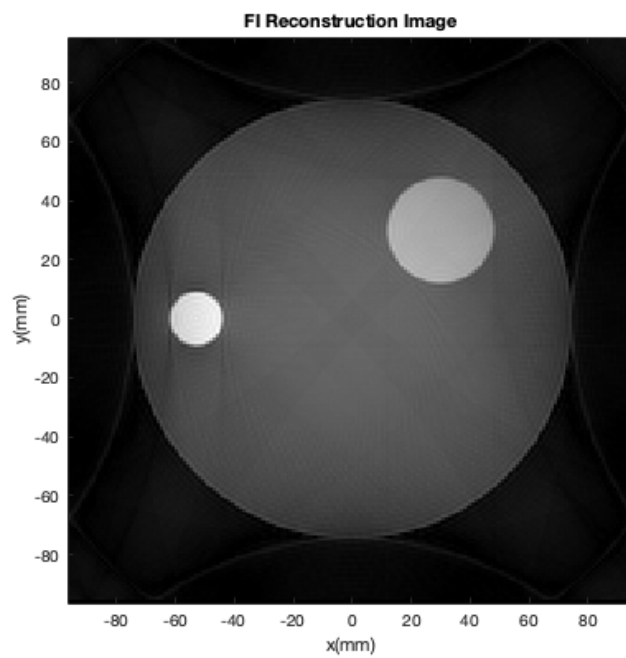
Warning: Duplicate data points have been detected and removed - corresponding values have been averaged.

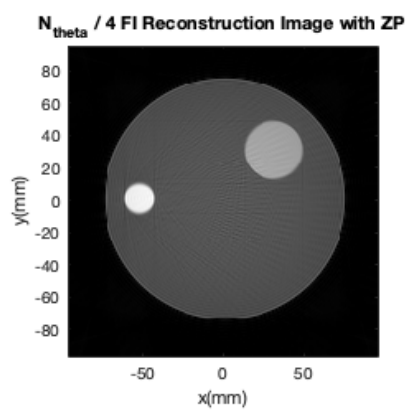
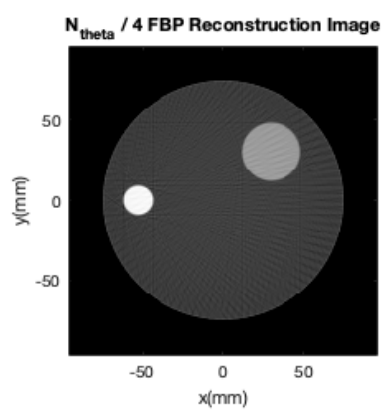
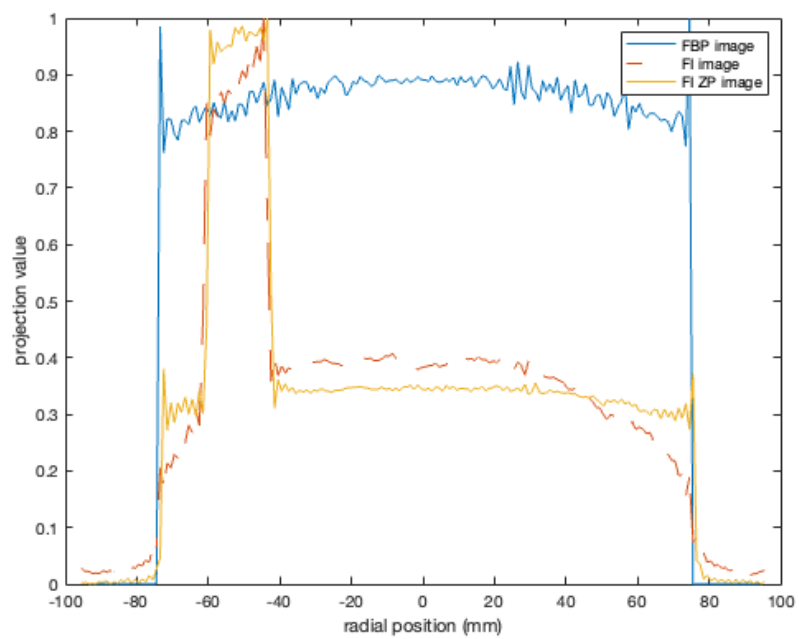
Warning: Duplicate data points have been detected and removed - corresponding values have been averaged.

Warning: Duplicate data points have been detected and removed - corresponding values have been averaged.

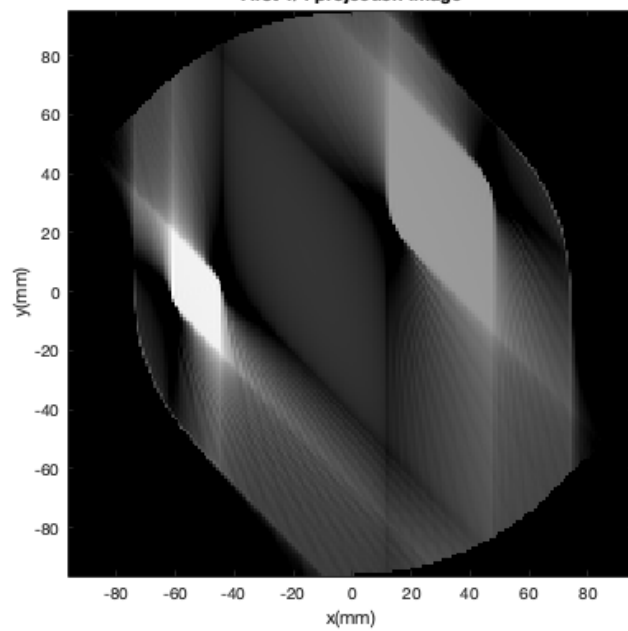








First 1/4 projection Image



FBP Reconstruction Image

