Solutions For Homework #7

Problem 1:[10 pts]

Let

$$f(r) = \frac{1}{r} = \frac{1}{\sqrt{x^2 + y^2}} \quad (1)$$

We compute the Hankel Transform of $f(r)$ by first computing its Abel Transform and then calculating the 1D Fourier Transform of the result.

$$H(q) = \int_{-\infty}^{\infty} \left[\int_{-\infty}^{\infty} \frac{1}{\sqrt{x^2 + y^2}} \, dy \right] e^{-i2\pi qx} \, dx \quad (2)$$

In the above, the inner integral can be seen as the Abel transform of $f(r)$, while the outer integral is the 1D Fourier transform evaluated at frequency q.

We make the following variable substitutions

$$\begin{aligned} x &= r\cos(\theta) \\ y &= r\sin(\theta) \end{aligned}$$

In other words, we change coordinate systems from Cartesian to Polar. Notice that

$$dxdy = rdrd\theta \quad (3)$$

Thus, the integral for the Hankel Transform above becomes

$$H(q) = \int_0^{2\pi} \int_0^{\infty} \frac{1}{r} \, e^{-i2\pi r\cos(\theta)} \, rdrd\theta \quad (4)$$

Recalling the definition of the Bessel function

$$2\pi J_0(2\pi rq) = \int_0^{2\pi} e^{-i2\pi rq\cos(\theta)} \quad (5)$$

we see that,

$$\begin{aligned} H(q) &= \int_0^{\infty} 2\pi \, J_0(2\pi rq) \, dr \\ &= \frac{2\pi}{2\pi q} \int_0^{\infty} J_0(u) \, du \\ &= \frac{1}{q} \end{aligned}$$

where we have made a change of variables in the second line above, $u = 2\pi qr$. Thus, we see that the Hankel Transform of $f(r) = \frac{1}{r}$ is $\frac{1}{q}$

Problem 2:[10 pts]

(a) We model the aperture illumination function as a circular pillbox as follows

$$f(r) = \text{rect}\left(\frac{r}{D}\right) \tag{6}$$

where $D = 2.5$ meters. We know that the Hankel Transform of $f(r)$ is a jinc function

$$F(q) = D^2\text{jinc}(Dq) = D^2\frac{J_1(\pi Dq)}{2Dq} \tag{7}$$

The Fraunhofer approximation gives the power-pattern of the antenna in the far-field

$$\begin{aligned} P(\theta) &= |F\left(\frac{\sin(\theta)}{\lambda}\right)|^2 \\ &\approx |D^2\text{jinc}\left(D\frac{\theta}{\lambda}\right)|^2 \end{aligned}$$

where we have applied the small-angle approximation $\frac{\sin(\theta)}{\lambda} \approx \frac{\theta}{\lambda}$. Figure 1 shows the 2D power-pattern, while a cut through the power-pattern is shown in Figure 2. The cut through the power-pattern is denoted by the dotted line in Figure 2. As we can see, the null-to-null width is about 0.1 seconds of arc. Note: The horizontal axis of Figure 2 is plotted in terms of seconds of arc by the following relation

$$q\frac{\lambda}{\theta} \times \frac{180^o}{\pi} \times 3600 \tag{8}$$

where $q = \frac{1}{r}$ is spatial frequency. We anticipate this value of 0.1 arc seconds for the null-to-null width of the untapered power-pattern through recognizing that the first null of the jinc function given above occurs at the value of 1.22, corresponding to a value of θ, in arc seconds, of

$$1.22 \times \frac{\lambda}{D} \times \frac{180}{\pi} \times 3600 = 0.05 \tag{9}$$

thereby yielding a null-to-null width of about 0.1 arc seconds.

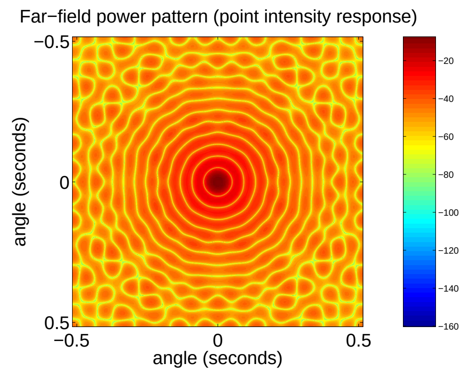

Figure 1:

(b) We also note from Figure 2 that the peak-to-sidelobe ratio, in dB, is about 18.

(c) Figure 3 shows a 1D profile of the Bessel tapering function. To compute this tapering function, we first note that the first minimum of the $J_0(x)$ function occurs at $x_0 = 3.82$, with a corresponding value of -0.4028. Now, our tapering function, which multiplies the original aperture illumination function, is described as follows

$$h(r) = J_0(ar) - (-0.4028) \tag{10}$$

where the scale factor a is determined so that the minimum of the Bessel function coincides with the edges of the aperture. Since the diameter D of the aperture is 2.5 meters, we know that $a = \frac{x_0}{D/2}$. Thus, when $r = D/2$ (i.e. at the aperture edges), the tapering function $h(r) = h(x_0) = 0$ as desired. The power pattern, in dB, of the tapered aperture illumination function is shown in Figure 4. A cut through this power pattern in shown as the solid line in Figure 2. As can be seen, the peak-to-sidelobe ratio is about 35 dB, implying a significant reduction of sidelobe level in the weighted-aperture power-pattern. The reduction of sidelobe level in the weighted aperture case is important for the detection of a weak start. This is because interfering radiation from a strong star that is located off the main-lobe direction would be supressed. However, this reduction in sidelobe level comes at the expense of a broadening of the main lobe. The null-to-null width of the main lobe of the tapered-aperture power pattern is about 0.18 seconds of arc, compared with 0.1 seconds for the untapered case.

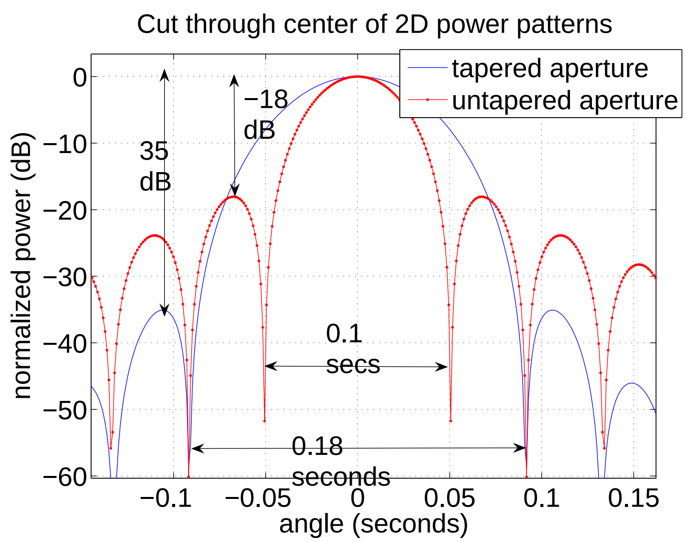

Figure 2:

(d) We inspect the plots of the power-patterns corresponding to the tapered and untapered aperture, looking for the 3dB point (corresponding to half power) down from the peak. We take the half-power width to be twice the width from 0 to the location on the θ axis where the power-pattern is 3dB down from the peak. We find that **for the untapered aperture, the half-power width is about 0.04 seconds of arc, while for the tapered aperture the half-power width is about 0.06 seconds of arc**.

Problem 3:[10 pts]

(a) We wish to numerically evaluate the Hankel Transform of the following function

$$f(r) = \frac{1}{\pi(a/2)^2}\text{rect}\left(\frac{r}{a}\right) \tag{11}$$

where, for this part, $a = 32$ and the spatial variable $r = 0, 1, 2, ..., n-1$ and n is the first power of 2 greater than $4a$. Thus, $n = 256$ and the spatial variable ranges from 0 to 255. We recognize the function above as a circular pillbox. The integral definition of the Hankel Transform is

$$F(q) = \int_0^\infty f(r)J_0(2\pi qr)\, rdr \tag{12}$$

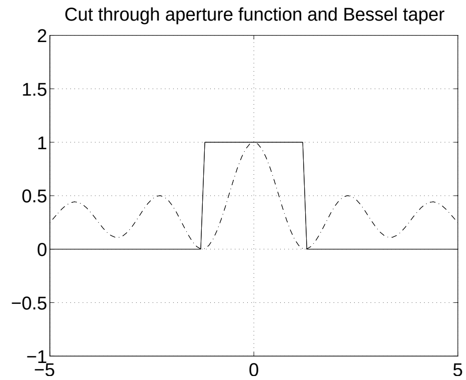

Figure 3:

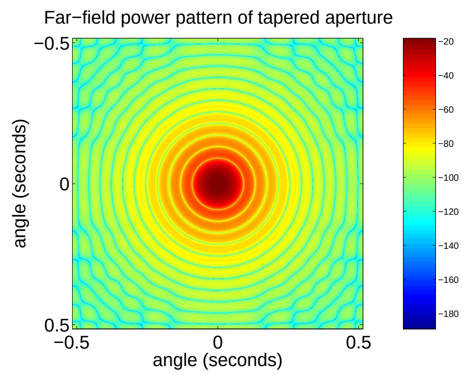

Figure 4:

We approximate this integral as a sum, as follows

$$F(q) = \sum_{r=0}^{n-1} f(r) J_0(2\pi q r) r \Delta r \tag{13}$$

where, of course, the step size $\Delta r = 1$. In addition, the frequency variable q assumes the values $0, \frac{1}{2n}, ..., \frac{n-1}{2n}$. To implement the Bessel function, we make use of the MATLAB command

```
besselj(0,r)
```

.

(b) We implement the same function $f(r)$ given in Part (a) in two-dimensions as follows

$$f(r) = \frac{1}{\pi(a/2)^2}\text{rect}\left(\frac{\sqrt{x^2+y^2}}{a}\right) \tag{14}$$

where $x = -n, n+1, ...0, ...n-1$ and $y = -n, n+1, ...0, ...n-1$. The Hankel Transform of this 2D circularly symmetric function can be easily computed using the 2D FFT (remember that for a circularly-symmetric function, the Hankel Transform is identical to the two-dimensional Fourier Transform). The Hankel Transform of the 2D function is shown in Figure 5. We are required to take cuts through the 2D function shown in Figure 5 in

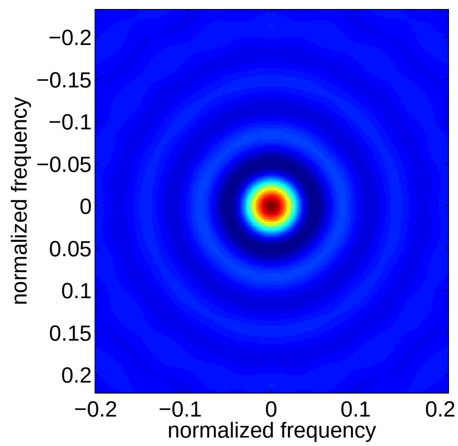

Figure 5:

the directions defined by the lines at angles 0^o and 45^o from the horizontal axis. The cut at 0^o is easily obtained by extracting values along a horizontal line passing through the center of the function. The code that extracts the cut at 45^o is given at the back. Note: when displaying the cut at 45^o, the 1D frequency axis needs to be stretched by a factor of $\sqrt{2}$, reflecting the fact that discrete steps (in the $u-v$ plane) at 45^o are $\sqrt{2}$ times longer than corresponding discrete steps along the horizontal axis.

(c) We know that the analytical form of the Hankel transform of the circularly symmetric function $f(r) == \frac{1}{\pi(a/2)^2}\text{rect}\left(\frac{r}{a}\right)$ is

$$F(q) = \frac{a^2}{\pi(a/2)^2}\text{jinc}(aq) \tag{15}$$

Recall that the jinc function is given by $\mathrm{jinc}(x) = \frac{J_1(\pi x)}{2x}$, and so we again make use of MATLAB's *besselj* routine to compute the analytical answer. Figure 6 shows the results the numerical evaluation of the Hankel Integral (solid line), cuts through the 2D Fourier Transform along lines at angles 0^o and 45^o from the horizontal axis (dotted lines and dots respectively) as well as the analytical solution (crosses). We see that the three methods of eval-

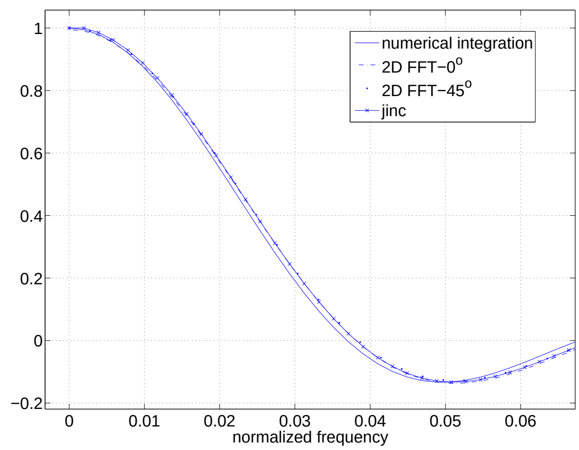

Figure 6:

uation produces answers that are very similar. The effect of approximating an integral by a discrete sum is evident as the numerical solution diverges slightly from the other solutions.

(d) Here, $a = 8$ and so $n = 64$. We perform the same summation as in Part (a) as well as form the 2D function as in Part (b). Now, of course, the 2D grid is smaller since we are only using 64 points. The 2D FFT of the circular pillbox function when $n = 64$ is shown in Figure 7. The comparison between the various solutions is shown in Figure 8. From Figure 8, we clearly see a divergence between the analytical solution and the numerical one. This is due to the fact that the approximation of the Hankel Integral by the sum given above is poor due to the limited number of points ($n = 64$). Furthermore, we also notice that the cuts through the 2D FFT of circular pillbox deviates from the analytical solution. As can be seen from Figure 7, the fewer number of grid points used in the definition of the 2D circular pillbox function $f(r)$ destroys some of the circular symmetry, causing a disparity

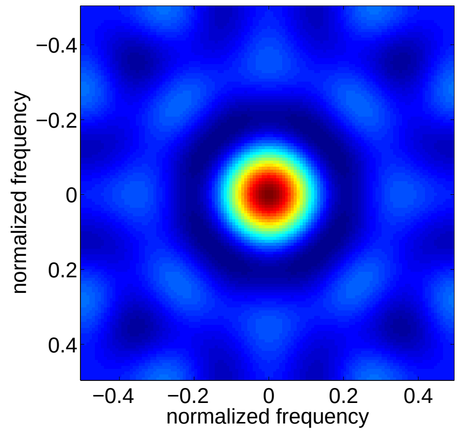

Figure 7:

in profiles along the horizontal (0^o) and 45^o lines.

Problem 4:[10 pts]

(a) In this problem, we seek to recover the 2D, circularly symmetric function from its Abel Transform. Recall the definition of the Abel transform:

$$a(x) = 2\int_x^\infty \frac{f(r)\, r dr}{\sqrt{r^2 - x^2}} \tag{16}$$

Here, we are given the data $a(x)$ evaluated at discrete values of x. Note that the spacing between points in the profile $a(x)$ is $\Delta x = 0.01$ and the number of samples in file is 600. Our task is to recover the 2D, circularly symmetric function $f(r)$ using the Abel-Fourier-Hankel relationship. Note: to load in the text file, we make use of MATLAB's

```
load
```

command. The first column gives the values a long the axis perpendicular to the projection direction where the projection is evaluated. This function is plotted in Figure 9

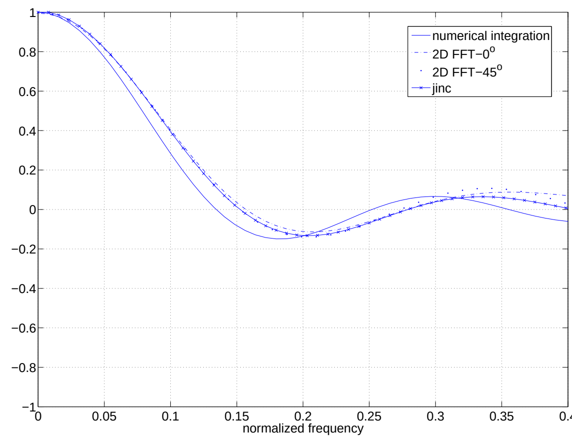

Figure 8:

(b) Recall the Abel-Fourier-Hankel relationship:**The Hankel Transform of the 2D circularly-symmetric function is equivalent to the 1D Fourier Transform of the function's Abel Transform**. Mathematically, letting $F(q)$ denote the Hankel Transform of the function $f(r)$,

$$F(q) = \int_{-\infty}^{\infty} A(x) e^{-i2\pi qx}\, dx \qquad (17)$$

where $A(x)$ is as was given in Part (a). To recover the 2D function, we would simply compute the Inverse Hankel Transform of $F(q)$. Note that the profile contained in the file *hw7p4data* is essentially $A(x)$. Notice, however, that the Abel Transform should return even functions. Thus, the data we are given, shown in Figure 9, is probably the right half of the full, even Abel Transform of the desired 2D function $f(r)$. We thus form a new, even function as follows

$$a'(x) = \{ \begin{array}{cc} a(x) & x > 0 \\ a(-x) & x < 0 \end{array} \qquad (18)$$

We then compute the 1D FFT of $A'(x)$ to get the 1D profile of the desired function's spectrum. Remember that the desired function is in two-dimensions, but circularly-symmetric, implying that all cuts through the origin of the function *or* its spectrum should be identical. The 1D FFT of $a'(x)$ is exactly one such radial cut of the 2D spectrum the desired 2D

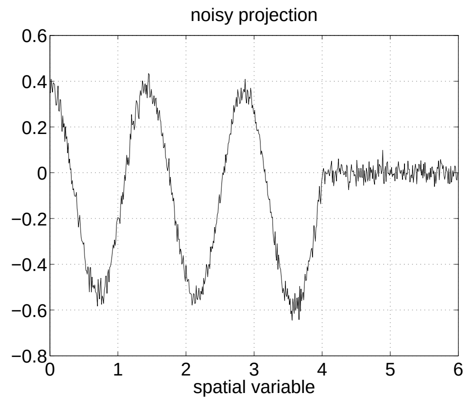

Figure 9:

function. Let this spectrum be denoted $A(q)$. To recover a radial cut of the desired 2D circularly-symmetric function $f(r)$, we compute the Inverse Hankel Transform, *in one dimension*, of the 1D spectrum $A(q)$. The Inverse Hankel Transform is defined as

$$f_{1D}(r) = \int_0^\infty A(q) J_0(2\pi q r)\, q dq \tag{19}$$

We approximate this integral by a sum, as we did in Problem 3.

$$f_{1D}(r) \approx \sum_{i=0}^{N-1} A(q_i) J_0(2\pi q_i r) q_i \, \Delta q \tag{20}$$

Here, Δq refers to the frequency spacing in the FFT domain. We know that the spatial separation of points in the data file *hw7p4data* is $\Delta x = 0.01$ and the number of samples $N = 2 \times 600 = 1200$ (remember, the length of $a'(x)$ is twice the length of $a(x)$). Therefore, the frequency spacing $\Delta q = \frac{1}{N \Delta x}$ and the FFT frequencies $q = 0, \frac{1}{N\Delta q}, ..., \frac{N-1}{N\Delta q}$. Plugging these values in the discrete sum above, we recover the 1D cut, shown in Figure 10, through the desired radially-symmetric 2D function $f(r)$.

(c) The MATLAB code that we use to form the "image" from the 1D radial cut $f_{1D}(r)$ shown in Figure 10 is given at the back. Essentially, we create a grid (x, y) and copy the i-th value in the 1D profile $f_{1D}(r_i)$ to all locations where $\sqrt{x^2 + y^2} \approx r_i$. The recovered 2D, circularly-symmetric function $f(r)$ is shown in Figure 11 $f(r)$.

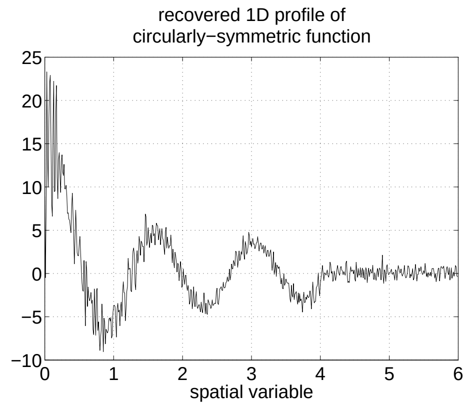

Figure 10:

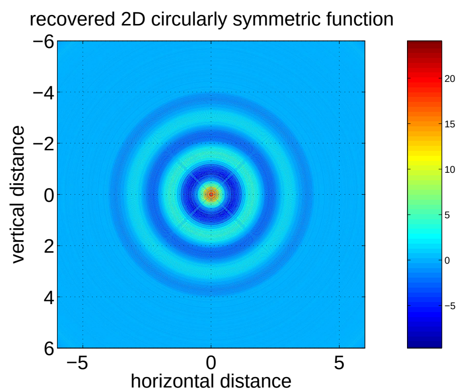

Figure 11:

Problem 5:[10 pts]

The integral definition of the Bessel function of First Kind, order 0, $J_0(2\pi qr)$ is given by

$$J_0(2\pi qr) = \frac{1}{2\pi}\int_0^{2\pi} \cos(2\pi qr\cos(\theta))\; d\theta \tag{21}$$

In the above, q denotes frequency. For this problem, we are asked to produce three Bessel functions of frequencies 4,7 and 10. This equation arises by adding up 2D cosine functions rotated from the x-axis at various angles θ. That is,

$$J_0(2\pi\, qr) \approx \frac{1}{2\pi} \sum_{\theta_i} \cos(2\pi q(x\cos(\theta_i) + y\sin(\theta_i)))\ \Delta\theta \tag{22}$$

Here, $q = \frac{1}{\text{wavelength}}$ is the spatial frequency, and we define wavelength as $\frac{N\Delta}{m}$, $m = 4, 7, 10$, where N is the number of points along one dimension in our grid, Δ is the spacing between points and m indicates number of cycles over the span of the grid. The MATLAB code implementing this is given at the back.

Figure 12 gives the three plots. The dots indicate the sum approximation, while the solid line is the theoretical curve $J_0(2\pi qr)$ where, of course, $r = \sqrt{x^2 + y^2}$.

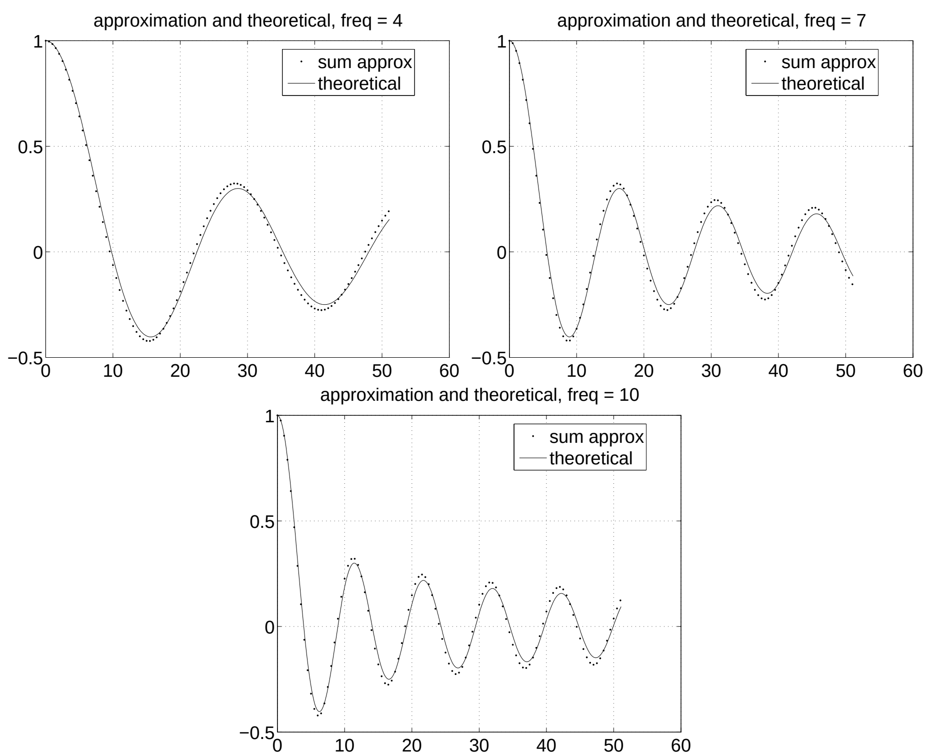

Figure 12:

MATLAB code for Problem 4

```
% load in data
im = load('hw7p4data');
data2 = im(:,2); x = im(:,1); data2 = data2';
Delta = x(2)-x(1);

% plot noisy projection
figure(1); plot(x,data2);grid on;
h=gca;set(h,'FontSize',20); xlabel('spatial variable');
h1=title('noisy projection');set(h1,'FontSize',20);

% make data an even function
data = [fliplr(data2(2:end-1)) data2];
N = length(data);

% Abel to Hankel - take FFT of data,
% but keep only positive Fourier coefficients since function
% real and even
F = real( fft(fftshift(data)) ); F = F(1:N/2);
q = [0:N/2-1]/N/(Delta);

% Inverse Hankel Transform, approximated by
% sum_{i=1}^N/2 F(q_i) q_i J0(2 pi q_i r) Delta q

r = [0:N/2-1]*(Delta*sqrt(2));
f = zeros(size(r));
Delta_q = q(2) - q(1);
for k=1:length(r)

  f(k) = sum(F(:).*Delta_q.*q(:).*besselj(0,2*pi*r(k).*q(:)));

end

% plot recovered 1D function
figure(2); plot(r, f); grid on;
h=gca;set(h,'FontSize',20); xlabel('spatial variable');
h1=title({'recovered 1D profile of','circularly-symmetric function'});
set(h1,'FontSize',20);
```

```
% create 2D function from 1D profile

rmax = max(r); NN = 4*ceil(rmax./Delta/sqrt(2));
horiz = [-NN/2:NN/2-1]*Delta; vert = [-NN/2:NN/2-1]*Delta;
theta = linspace(0,2*pi,2*length(f));

f2D = zeros(NN,NN); counts = zeros(NN,NN);

for k = 1:length(f)

  if(r(k) ~= 0)

    Rows = round( r(k)*sin(theta)./Delta/1 ); Rows = Rows + NN/2 + 1;
    Cols = round( r(k)*cos(theta)./Delta/1 ); Cols = Cols + NN/2 + 1;
    ind = find((Rows <= NN) & (Cols <= NN) );
    Rows = Rows(ind); Cols = Cols(ind);

    f2D(sub2ind(size(f2D),Rows,Cols)) =
    f2D(sub2ind(size(f2D),Rows,Cols)) + f(k);

    counts(sub2ind(size(f2D),Rows,Cols)) =
    counts(sub2ind(size(f2D),Rows,Cols)) + 1;

  else
    f2D(ceil(NN/2),ceil(NN/2)) = f(k);
  end

end
f2D(find(counts~=0)) = f2D(find(counts~=0))./counts(find(counts~=0));

figure(3); imagesc(horiz,vert,f2D); grid on;
h=gca;set(h,'FontSize',20); xlabel('horizontal distance');
ylabel('vertical distance'); axis image;
h1=title('recovered 2D circularly symmetric function');
set(h1,'FontSize',20); colorbar\\
```

MATLAB code for Problem 5

```
% parameters
N = 1024;
Delta = 0.1;
a = [N/4 N/7 N/10]*Delta;       % wavelength

% create spatial array
horiz = [-N/2:N/2-1]*Delta; vert = [-N/2:N/2-1]*Delta;
[Cols,Rows] = meshgrid(horiz,vert);
rr = sqrt(Rows.^2 + Cols.^2);
% create angles between 0 and 2 pi radians
% Angles are measured from x-axis;
theta = linspace(0,2*pi,20);

% create cosines of varying wavelengths

for k = 1:length(a)
   % create horizontal and vertical spatial frequencies
  % from theta definition and wavelengths
  u = a(k)*cos(theta); v = a(k)*sin(theta);
  q = 1./sqrt( u.^2 + v.^2 );

  % perform sum of 2pi radians as
  % (1/2/pi) sum_{i=1}^N cos( q r cos(theta) )Delta_theta
  map2 = zeros(size(Cols));
   for m = 1:length(theta)
     disp(m);
     map2 = map2 +
     cos( 2*pi*(q(m)*cos(theta(m))*Cols + q(m)*sin(theta(m))*Rows) );

  end
  map2 = map2./length(m);

  % extract horizontal cut and store

  num_bessel2(k,:) = map2(N/2+1,N/2+1:N)./max(map2(N/2+1,N/2+1:N));

  theo_bessel(k,:) = besselj(0,2*pi*q(1)*[0:N/2-1]*Delta);
end
```