Physics 124 – Problem Set # 8

(due Wednesday, November 27)

(Problems 1 and 3 are cribbed from Heald and Marion.)

1. An aperture is in the form of an annulus with inner radius b and outer radius a. Find the intensity of the Fraunhofer diffraction pattern in terms of Bessel functions. Now plot carefully the intensity as a function of radius on a distant screen for the cases $b = 0$ (circular aperture) and $b = a/2$. Find the position of the first dark ring and the position of the second maximum in each case. Show that the resolution is increased by adding the central disk, but that the contrast is reduced because the relative intensity of the second maximum is increased.

2. It is not so easy to compute the diffraction pattern of a half-circle. However, we can get a first approximation to this by writing the integral as

$$\int d^2y = \int dr r \int d\phi \Theta(\pi/2 - |\phi|) \tag{1}$$

and replacing the theta function by its Fourier series

$$\Theta(\pi/2 - |\phi|) \approx \frac{1}{2} + \frac{2}{\pi} \cos\phi + \cdots \tag{2}$$

Well, maybe this also is not so easy to solve.

(a) Consider the simpler problem in which we replace

$$\Theta(\pi/2 - |\phi|) \to \frac{1}{2} + \epsilon(r/a)\cos\phi \tag{3}$$

where ϵ is small. Evaluate the diffraction pattern. How are the positions of the zeros shifted by the perturbation.

(b) If the pattern on the screen is rotated by an angle α, how does the diffraction pattern change?

(c) Compute the diffraction pattern for the case

$$\Theta(\pi/2 - |\phi|) \to (r/a)\cos\phi \tag{4}$$

with no '1' term. Where are the zeroes? What symmetry does the pattern have?

3. Here are some additional properties of the Cornu spiral.

(a) Show that the slope of the Cornu spiral at any point is $\tan(\pi u^2/2)$.

(b) Show that the local radius of curvature at any point on the Cornu spiral is

$$\left[\frac{d}{du}\frac{\pi}{2}u^2\right]^{-1} = \frac{1}{\pi u} \tag{5}$$

(c) Suppose a railroad track is laid out in the shape of a Cornu spiral. If a train moves along the track at constant speed, show that the sideways centrifugal force observed within the train changes *linearly* with time (this is called a *railroad curve*).

4. This problem involves a Java applet that computes diffraction patterns. The setup is as follows: You draw an aperture in the box on the left. When the applet is working, you can click 'Compute' and the corresponding diffraction pattern will appear in the box on the right.

(a) Consider first a discretization in which each pixel on the left is treated as a point source. Show that the diffraction pattern a distant screen is then computed as

$$\left|\sum_{i_x,i_y} \exp\left[-ik(\frac{d_x}{Z}i_x a + \frac{d_y}{Z}i_y a)\right]\right|^2 \tag{6}$$

where Z is the distance to the screen, (d_x, d_y) is a position on the screen, a is the spacing of pixels in the aperture, and (i_x, i_y) integers specifying a given pixel, and the sum is taken only over illuminated (white) points. It is convenient to discretize (d_x, d_y) by writing

$$k\frac{d_x}{R}a = \frac{2\pi}{N_{res}}m_x \ , \qquad k\frac{d_y}{R}a = \frac{2\pi}{N_{res}}m_y \ , \tag{7}$$

where (m_x, m_y) are integers. Write the formula for the diffraction pattern at each point (m_x, m_y).

(b) Actually, it is not so much harder to obtain the exact diffraction pattern of the figure on the left. Instead of treating each illuminated pixel as a point, treat it as an illuminated $a \times a$ square. Write the formula for the diffraction pattern at each point (m_x, m_y).

(c) Download from the class website the files:

`Diffraction.html, Diffraction.java, DiffractionGUI.java,`
`PhysicsApplet.java` (8)

Edit `Diffraction.html` to remove the phrase: `archive="Diffraction.jar"`. Modify the file `Diffraction.java` to implement the algorithm of part (a) or part (b). Note that (i_x, i_y) appear in the file as `(i,j)` and (m_x, m_y) appear in the file as `(m,n)`. Compile the edited file, and you should have a working applet that computes diffraction patterns.

(d) Your first try at getting this applet to work probably runs very slowly. This is because `Nx = Ny = 100`, so you are summing over 10,000 index items. You do not want to compute sines and cosines separately for each item. Put the sines and cosines you need into an array, in the manner indicated in the file `Diffraction.java`, and compute them once and for all in the applet constructor, *i.e.*, at initialization. Then the applet will run faster, though the speed will still not be blazing.

(e) Compute the diffraction pattern of a rectangular aperture. Make the rectangle pixel-perfect and show that the positions of the zeros of the diffraction pattern are exactly in the right pattern.

(f) Show that the diffraction pattern of an aperture of fixed shape is independent of its position on the screen. Does the applet reproduce this?

(g) Compute the diffraction pattern of a circle. Fill in a circle in the center of this one. Can you see the effect present in problem 1?

(h) Now draw a circle and make it asymmetric. Can you see the effect presented in problem 2?

(i) Draw a small circle in the left half of the left screen. Then draw another circle of the same radius to the right of it. Compute the diffraction pattern using the applet, and then explain it.

(j) Experiment with other aperture shapes: gratings, filled and unfilled boxes, figures with various symmetries.

Hand in your code for the `Diffraction` class, your solution to (a), (b), (f) and illustrative plots from (e),(g),(h),(i),(j).

```
import java.awt.*;
import java.awt.event.*;
import java.applet.Applet;

public class Diffraction extends DiffractionGUI {

  int Nres = 150;
  double [] allsines;

  public Diffraction(){
      /* example of pre-computation of an array: */
      double dNres = Nres;
      allsines = new double[2*Nres];
      for (int j = 0; j< 2*Nres ; j++){
allsines[j] = Math.sin(2.0 * Math.PI * j/dNres);
      }
  }

  void solve(){
    resetArrays();   /* zeros the array phi */
    for (int m = 1; m < Nx; m++){
      for (int n =1; n < Ny ; n++){
double realamp = 0.0;
        double imamp = 0.0;
   /* so that, effectively, (0,0) appear in the center of the square */
        int mm = m - (Nx/2);
        int nn = n - (Nx/2);
        for (int i = 1; i < Nx; i++){
          for (int j =1; j < Ny ; j++){
    if (State[i][j] == blackState) continue;
                /* add contributions from white squares ;
          put something more sensible here:  */
    realamp += 0.0;
    imamp +=   0.0;
          }
        }
        phi[m][n] = (realamp*realamp + imamp*imamp);
      }
    }
    Legend.write(" ");
    refreshPicture();
  }

}
```


Figure 1: The source file `Diffraction.java`.