Physics 121 – Problem Set # 6

(due Friday, May 24)

1. Griffiths, problem 9.23.
2. Griffiths, problem 9.25.
3. The vibrations of a solid medium (rock) propagate as sound waves. Let $\vec{\chi}(t, \vec{x})$ be the displacement of the rock from its equilibrium position. Then, for small displacements, $\vec{\chi}$ will solve a wave equation. The velocity of the wave can depend on the polarization. If we write a simple wave solution as

$$\vec{\chi} = \text{Re}\ \left[\chi_0 \vec{\epsilon} e^{i\vec{k}\cdot\vec{x} - i\omega t}\right] \tag{1}$$

then we might have

$$\begin{aligned} \vec{\epsilon} \parallel \hat{k} \quad &: \quad \omega = c_c k \\ \vec{\epsilon} \perp \hat{k} \quad &: \quad \omega = c_s k \end{aligned} \tag{2}$$

where c_c is the velocity of compression waves and c_s is the velocity of shear waves. Typically, $c_c > c_s$, because the restoring force is larger.

Consider now an interface between two types of rock, one of low density, one of high density. In the low density rock, $c_c = c_s = c$. In the high density rock, $c_c > c_s$, but both are less than c. Analyze the reflection and transmission of sound waves coming from the low-density medium into the high-density medium at angle of incidence θ_i.

   (a) Let the interface between the media be a plane normal to $\hat{z}$, and let the initial wavevector of the sound wave be in the $\hat{x}$-$\hat{z}$ plane. Impose the boundary condition that $\vec{\chi}$ and $\partial\vec{\chi}/\partial z$ are continuous at the interface. The three possible polarizations of the initial sound wave can be divided in to two cases, $\vec{\epsilon} \parallel \hat{y}$ and $\vec{\epsilon}$ in the $\hat{x}$-$\hat{z}$ plane. Analyze the case $\vec{\epsilon} \parallel \hat{y}$. Show that the reflected and transmitted waves are also polarized parallel to $\hat{y}$. Compute the reflection coefficient and the fraction of the initial energy that is reflected.

   (b) Now consider the case of $\vec{\epsilon}$ in the $\hat{x}$-$\hat{z}$ plane. Consider first the case in which the transmitted wave is pure compression: $\vec{\epsilon}_T = \hat{k}_T$. Find incident and reflected waves that match onto this and give a complete solution to the wave equation. Notice that the initial and reflected waves have a shear component, that is their polarizations do not satisfy $\vec{\epsilon} = \hat{k}$.

   (c) Consider the opposite case, in which the transmitted wave has a polarization in the $\hat{x}$-$\hat{z}$ plane but normal to $\hat{k}_T$. Again, find the incident and reflected waves that fit together with this into a complete solution to the wave equation.

(d) By taking linear combinations of the solutions in (b) and (c), compute the reflection coefficient and the reflected energy for initial compression waves, that is, waves in which the incident waveform has $\vec{\epsilon} = \hat{k}$. Notice that there are two transmitted waves that go in different directions.

(e) Finally, find the reflection coefficient and the reflected energy for initial waves that are pure shear with initial polarization in the $\hat{x}$-$\hat{z}$ plane.

4. Griffiths, problem 9.37.

5. This problem involves a Java applet that solves the wave equation in a dispersive medium.

(a) Since the wave equation is linear, it is easy to solve in Fourier space. For each Fourier coefficient, we simply replace

$$e^{ikx} \to e^{ikx} e^{-i\omega(k)t} \tag{3}$$

This idea can be turned into a fairly efficient numerical method. For a function on a discrete set of points $f(n)$, write a Fourier transform representation:

$$f(n) = \frac{1}{N} \sum_{n=0}^{n=N-1} \tilde{f}(m) e^{2\pi i m n/N} \tag{4}$$

If the lattice spacing is a, define $k = 2\pi m/aN$, so that the exponential is $\exp(ikna)$. Then the time evolution of the waveform is computed by replacing

$$e^{ikna} \to e^{ikna} e^{-i\omega(k)t} \tag{5}$$

Take $a = 1$ for the numerical calculations. In problem set 2, you wrote a numerical Fourier Transform program. This program represented $\tilde{f}(m)$ by its real and imaginary parts `fcos[m]` and `fsin[m]`. Look back at your solution to problem 4 of problem set 2 (which you hopefully have not discarded) and remember how you did this. Then, write the equations for `fcos` and `fsin` at nonzero t that give the time evolution of the wave for a given dispersion relation $\omega(k)$.

(b) Implement this method in java. Download the files `Dispese.html`, `Disperse.java`, and `DisperseGUI.java` and `PhysicsApplet` from the Physics 121 web site. You will also find there the original form of `FourierTransform.java`. Hopefully, you have kept the finished version of this class from problem set 2, which you can now just copy into the directory for this program. The code of `Disperse.java` is given at the end of the problem set. Replace the marked lines of code with the algorithm from part (a). Compile the applet and watch the waves move.

(c) Notice that (if you have programmed correctly), the waves move to the right. Why doesn't the initial condition split, with waves going in both directions? What happens when the wave reaches the boundary, and why?

(d) The function `omega(m)` in `Disperse.java` returns $\omega = k$. This is put in *ad hoc*. Contrast with this the actual solution of the wave equation on a lattice:

$$\frac{\partial^2 f_n}{\partial t^2} = \frac{1}{a^2}(f_{n+1} + f_{n-1} - 2f_n) \qquad (6)$$

By introducing a Fourier representation of f_n, solve this equation and find the dispersion relation. Show that

$$\omega(k) = [(2/a^2)(1 - \cos ka)]^{1/2} \; . \qquad (7)$$

Show that this expression has the limiting form $\omega = k$ as $a \to 0$.

(e) By modifying the function `omega(m)`, implement the dispersion relation of part (d), with $a = 1$, in the java code. Can you see the difference from $\omega = k$? Which is 'better'?

(f) Now try out different dispersion relations. Consider (i) $\omega = k + 2k^2$, (ii) $\omega = [1 + k^2]^{1/2}$, (iii) $\omega = 3 - k$, and others that you are curious about. Describe how an initial wavepacket distorts in each case.

(g) For each of the three listed cases in part (e), compute the group velocity v_g. What does the behaviour of v_g have to do with the numerical results that you are seeing?

Hand in your code for the `Disperse` class, your solution to (a), (c), (d), (e), and (g), and illustrative plots from (f).

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

public class Disperse extends DisperseGUI{

  void solve(){
    FT.Transform();
    for (int i = 0; i <= N2 ; i++){
       ficos[i] = FT.fcos[i];
       fisin[i] = FT.fsin[i];
    }
    t = 0;
    while(true) {
      t += tstep;
      for (int i = 1; i <= N2 ; i++){
        double wt = omega(i) * t;
        FT.fcos[i] = 0.0;
        FT.fsin[i] = 0.0;
    /* replace this with code that solves the wave equation */
      }
      FT.InverseTransform();
      refreshPicture();
      if (timetostop) break;
    }
  }

  double omega(int n){
      double w;
      double k =  2.0 * Math.PI * n/Nx;
      w = k;
          /* replace this with your favorite dispersion relation */
      return w;
  }

  void inputWave(){
    for (int n = 0 ; n < Nx; n++){
       double t = (n- 0.2*Nx)/(0.2*Nx);
       fi[n] = Math.exp(- 10.0 * t * t) * Math.sin(24.0* Math.PI * t);
    }
  }
}
```