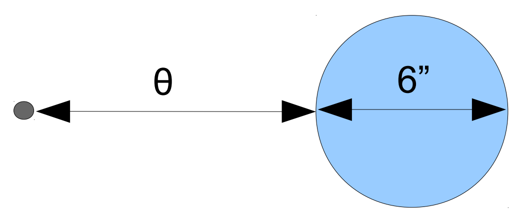
Physics 3340/8300-005- "Computational Physics" - Fall 2013
Homework 8
Assigned: Monday, 28 October
Due: Friday, 1 November 11:59pm by Blackboard upload

Reading:

Cheney and Kincaid sections 4.1,4.2
http://www.physics.smu.edu/fattarus/InterpolationLab.html

Problems:
For the following problems submit your solutions in your email report, with enough text or gnuplot graphics to trace the process of your work.

1) When viewed through a telescope, the diameter of a planet spans 6 seconds of arc. The position of its moon relative to the edge of the planet is recorded exactly at midnight each night for several nights. The moon is initially to the left of the planet, as shown in the figure:



On four successive nights the moon is observed at separation angles of 0.4083885“, 2.6100975“, 2.8256509” and 0.9787359“ of arc to the left of the planet. On the fifth and sixth nights the moon passes behind the planet, and then on nights seven, eight and nine, again at midnight, the moon is observed to the right of the planet at 2.3127362“, 2.9562532“ and 1.4910645“ of arc relative to the right planet edge. The orbital path is in the plane of the planet's equator. Use interpolation to estimate the orbital period of the moon.

2) Expand the class lab interpolate.c source code file with three new functions: lagrange_inverse(), find_newton_inverse_coeffs() and newton_inverse(). Make copies of the functions lagrange(), find_newton_coeffs() and newton() already in the source code file and edit them to create your new functions. The new functions should perform inverse polynomial interpolation, where the role of x and y coordinates are reversed.

Then make use of the main program interpolate_solve.c that is contained in the class lab exercise tar file for polynomial interpolation to test out your new expanded version of interpolate.c. List the contents of the file data_cosine9.dat that is also supplied by the interpolation.tar file. Note that this file evaluates the function $\cos(2\pi x)$ at nine evenly spaced points between 0 and 0.25, with an x spacing every 1/32. Use this file as a data file to test the interpolate_solve program that calls your new functions, solving for a target value of 0.5. Run the program twice, after compiling for the Newton formulation and the

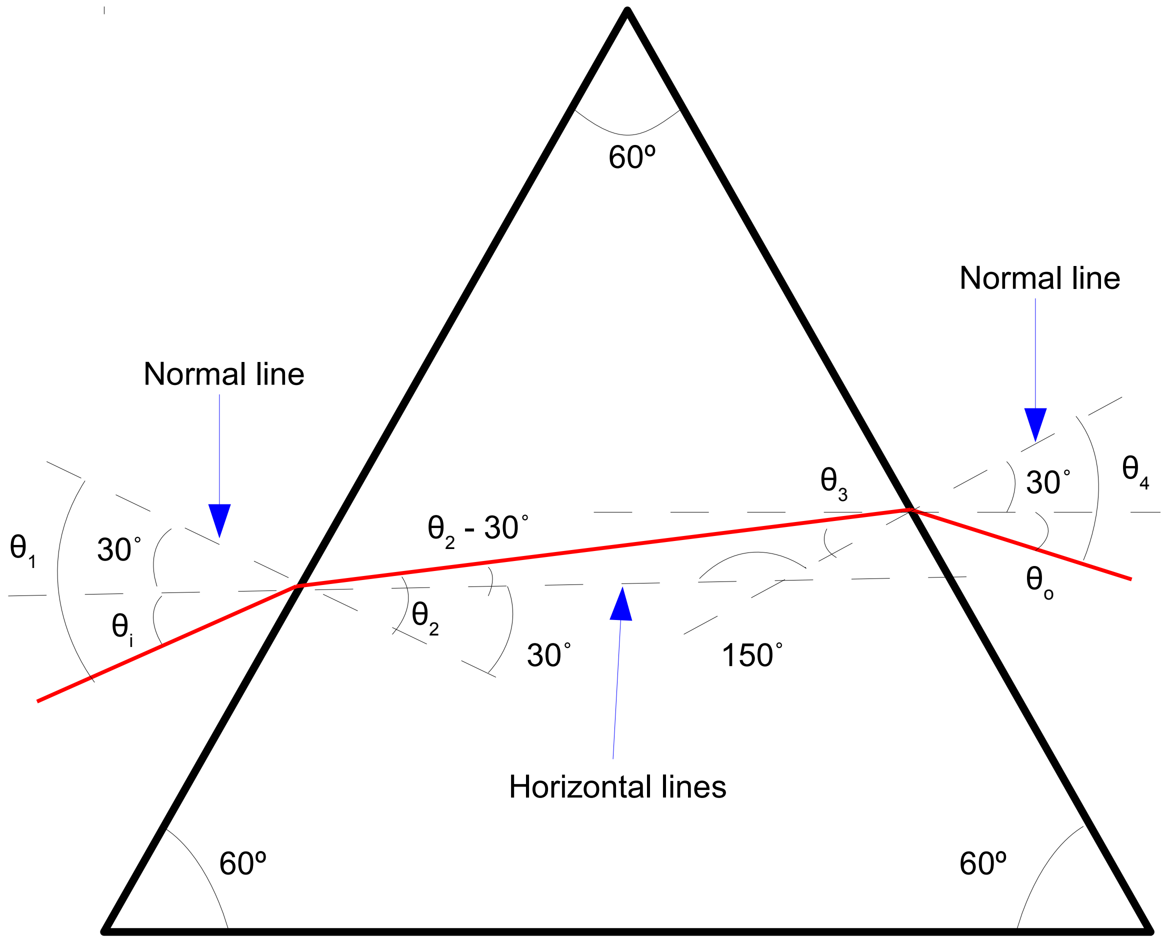
Lagrange formulation respectively. Compare the values found by the two runs of the program with the value you expect. Submit your new interpolate.c source code file.

3) The following table contains measured data points of the atmospheric pressure as a function of altitude, measured at 5,000 feet intervals. The first column is the altitude in feet, and the second column the atmospheric pressure in pounds per square inch.

Assume that the interior of an aircraft has been pressurized to an equivalent altitude of 7,500 feet. It is known that the window in the cockpit will burst outward if there is greater than 7.5 pounds per square inch difference between the interior and exterior of the plane. What is the maximum safe altitude at which the airplane may fly? Hint: You may find the tools interpolate_sweep and interpolate_solve from the class lab exercise to be handy for this problem.

Altitude (feet)	**Atmospheric Pressure (psi)**
0	14.7
5000	12.2
10000	10.1
15000	8.29
20000	6.75
25000	5.45
30000	4.36
35000	3.46
40000	2.71
45000	2.10
50000	1.61

4) Below is a graphic diagram of a triangular prism with equal 60˚ angles at the corners of the triangular cross section:



Using the labeling of the various angles in the diagram, and assuming the angle that the incoming ray of monochromatic light forms with the horizontal base of the prism is θ_{in}, the algorithm for calculating the outgoing light ray angle θ_{out} is:

$$\theta_1 = \theta_{in} + 30^\circ$$

From Snell's law $\sin(\theta_1) = n\sin(\theta_2)$ so $\theta_2 = \arcsin(\frac{1}{n}\sin(\theta_1))$

From triangle law $\theta_3 = 180^\circ - (150^\circ + \theta_2 - 30^\circ) = 60^\circ - \theta_2$

From Snell's law $n\sin(\theta_3) = \sin(\theta_4)$ so $\theta_4 = \arcsin(n\sin(\theta_3))$

$$\theta_{out} = \theta_4 - 30^\circ$$

where n is the index of refraction of the glass in the prism at the wavelength of light in the ray. Assume the index of refraction in air is 1.0.

The following table lists measurements of the index of refraction of a particular glass to be used in the prism taken at four light wavelengths:

Wavelength (nm)	**Index of Refraction**
400	1.522

500	1.511
600	1.505
700	1.502

Use interpolation to generate a plot of the output angle versus light wavelength predicted for an input angle of 30 degrees below the horizontal. The plot should show output angle over the range of 400 to 700 nm, calculated for at least 100 wavelengths. As discussed in class, an outline of the program is:

```
struct point2d *data;
double theta_i;
int n_points;

double prism(double lambda) {
    ...
  n = newton(n_points,data,lambda);
    ...
  return(theta_o * RADTODEG);
}

int main(int argc,char *argv[]) {
    ...
  n_points = read_data_file(argv[1],&data);
    ...
  find_newton_coeffs(n_points,data);
  sweep(prism,lstart,lstop,lstep);
  exit(0);
}
```

Submit your program code and a PNG graphics file of the resulting plot. Include appropriate axis labels and title.

5) A colleague thinks the data points in the following table originate from a cubic dependence of the variable *y* on the independent variable *x*. How can this be tested?

x	*y*
-2	1
-1	4
0	11
1	16
2	13
3	-4