Science and Computers II: Project 3

One of the goals of this project is for you to gain familiarity with using a software library to solve a physical problem, rather than writing an entire code from scratch; consequently the first two exercises will likely be quite short. You are not explicitly told which GSL routines to use, as part of the exercise is for you to learn how to locate the appropriate routine to solve a problem.

All material in this project is assessed and your report should document the results of all three exercises in separate sections. I suggest writing up the sections as you complete them, rather than leaving all the writing until the end.

Before attempting Exercises 1 and 2, you should first read Section 2 of the GSL reference manual

`http://www.gnu.org/software/gsl/manual/html_node/Using-the-library.html`

which explains how to use GSL and link the library into your programs. Then work through the exercise. You should identify the relevant numerical problem, e.g. root finding, numerical integration, etc. and then choose an appropriate routine from GSL to solve the problem. If you are having trouble doing this, please see me.

Exercise 1

The amplitude of the Fraunhoffer diffraction pattern is given by the expression

$$A = A_0 \frac{\sin x}{x}$$

where $x = \frac{1}{2} ka \sin\theta$, k is the wavenumber of the light, a is the slit width and θ is the diffraction angle. Suppose that the wavelength of the light is 589.29 nm and the width of the slit is 2.8 μm. The purpose of this exercise is to find the angles of the maxima in the diffraction pattern. Clearly the values of x for which the zeros occur must satisfy $\sin x = 0$ and therefore are readily determined. What equation must be satisfied by the values of x that correspond to the maxima in the diffraction pattern? The maxima will generally lie between the minima—we therefore have upper and lower bounds for the maxima.

As a general rule: Whenever possible use root-finding routines that bracket the roots—that is safer than using a routine that only asks for an initial approximation.

How many maxima will there be? Call a suitable GSL routine to find the values of x that correspond to these maxima. Finally find the angles corresponding to the maxima. Check your results.

Exercise 2

Consider the following 3×3 matrix:

$$\begin{pmatrix} 1 & 2 & 3 \\ 2 & 2 & 3 \\ 3 & 3 & 3 \end{pmatrix}$$

Using suitable GSL functions find (a) its inverse, (b) its determinant, (c) its eigenvalues and eigenvectors.

Write a program that generates matrices of size 1000×1000, 2000×2000, ..., up to $10,000 \times 10,000$. How long does it take the computer to find the determinant of such a matrix? You must be careful not to print the matrices out to `stderr` or `stdout`, or this will slow the calculation. How does the timing scale with the problem? How long do you think it would take to find the determinant of a $10^6 \times 10^6$ matrix?

Exercise 3

In this exercise, you should write the code from scratch (no GSL routine). Write a code that solves the Laplace equation in two dimensions using the method of over-relaxation. Laplace's equation applies to many different physical systems, from electromagnetism in a region without charges, to steady-state heat flow, to the displacement of a membrane (see the Feynman Lectures, Vol II). (Material here taken from Numerical Solution of Laplace's Equation, Per Brinch Hansen) Laplace's equation in two-dimensions is

$$\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} = 0. \tag{1}$$

It is sometimes also written

$$\nabla^2 u = 0. \tag{2}$$

Consider a square region, with temperature 100 (in our arbitrary units) at the top of the square and zero elsewhere. We use the following notation for the values and coordinates of the temperature field surrounding a site,

$$u_n = u(x, y+h)$$

$$u_w = u(x-h, y) \qquad u_c = u(x, y) \qquad u_e = u(x+h, y)$$

$$u_s = u(x, y-h)$$

If the grid spacing h is sufficiently small, we can approximate the temperature using a Taylor expansion to the right (east) of the central point,

$$u_e \approx u_c + h\frac{\partial u}{\partial x} + \frac{1}{2}h^2\frac{\partial^2 u}{\partial x^2},$$

Similarly, we can expand about the left (west) point,

$$u_w \approx u_c - h\frac{\partial u}{\partial x} + \frac{1}{2}h^2\frac{\partial^2 u}{\partial x^2}.$$

Adding these two equations together, we get

$$u_e + u_w \approx 2u_c + h^2\frac{\partial^2 u}{\partial x^2}.$$

Similarly in the vertical direction, we have

$$u_n + u_s \approx 2u_c + h^2 \frac{\partial^2 u}{\partial y^2},$$

We can combine these two equations to obtain

$$\nabla^2 u \approx (u_n + u_s + u_e + u_w - 4u_c)/h^2.$$

Since the left-hand side is zero by Laplace's equation, we have

$$4u_c - u_n - u_s - u_e - u_w \approx 0,$$

or

$$u_c \approx (u_n + u_s + u_e + u_w)/4.$$

Thus, in thermal equilibrium, the temperature at each grid point is the average of its neighbors. This suggests the following algorithm, called Gauss-Seidel relaxation. First set the boundary conditions. Then, set all of the other values to the average over the boundary. Then, sweep through the lattice, setting each site to the average of its neighbors. This procedure will eventually converge to the correct result. It turns out that the convergence can be sped up using over-relaxation, where a new temperature is chosen according to

$$\text{next} = (1 - f) \times u_c + f \times (u_n + u_s + u_e + u_w)/4$$

for some value of f. For $f = 1$ we have Gauss-Seidel, and for over-relaxation, $f > 1$. The procedure only converges if $f < 2$. David Young [1954] developed a theory of overrelaxation, showing that for an $L \times L$ grid, the fastest convergence is obtained with the relaxation factor

$$f_{opt} = 2 - 2\pi/L.$$

Implement overrelaxation to solve the Laplace equations for the boundary conditions specified. Use a grid size of at least 100×100. Compare to the analytical solution for a 10 cm^2 metal plate with one side held at 100 degrees (in absolute units) and the other sides at zero:

$$T = \sum_{\text{odd } n} \frac{400}{n\pi \sinh n\pi} \sinh\left[\frac{n\pi}{10}(10 - y)\right] \sin\left(\frac{n\pi x}{10}\right).$$