Exercise 1: Matrices and Linear Algebra

Drew Silcock
Physics Department, University of Bristol
(Dated: March 5, 2014)

I. RELAXATION METHODS

A common method of solving partial differential equations, both linear and nonlinear, is by iterative relaxation methods. Here this means taking advantage of the finite difference discretisation of the differential equation. The essential method has six steps:

Step 1: Define a regular spatial grid covering the region of interest including nodes at the boundaries.

Step 2: Set the boundary conditions of the problem by fixing the boundary nodes to the boundary values.

Step 3: Guess an initial state for the interior of the grid.

Step 4: Calculate the finite difference equations.

Step 5: Choose a convergence criterion.

Step 6: Iterate the equations at each boundary node until the convergence condition is satisfied.

As applied to the Poisson equation

$$\nabla^2\varphi = -\rho, \tag{1}$$

this means that, given φ is a smooth function $\varphi : \mathbb{R} \to \mathbb{R}$ and thus has Taylor expansion about point x_i of

$$\varphi(x) = \varphi(x_i) + (x - x_i)d_x\varphi(x_i) + \frac{(x - x_i)^2}{2}d_x^2\varphi(x_i) + \mathcal{O}((x - x_i)^3),$$

the second derivative can be approximated by

$$d_x^2\varphi(x) = \frac{\varphi(x - h) + \varphi(x + h) - 2\varphi(x)}{h^2} + \mathcal{O}(h^2) \tag{2}$$

where h is the grid spacing.

Equation 2 gives the finite difference for the second differential.

Generalising this to two dimensions and applying to Equation 1 gives the following iterative formula at each node:

$$\varphi(x_i, y_i) = \frac{1}{4}\Big[\varphi(x_{i-1}, y_j) + \varphi(x_{i+1}, y_j) + \varphi(x_i, y_{i-1}) + \varphi(x_i, y_{i+1}) + \rho(x_i, y_j)h^2\Big] \tag{3}$$

where x_i, y_j are points on the grid. If $\rho = 0$, i.e. there are no sources and Poisson's equation becomes Laplace's equation, then this iteration simply becomes the average value of the nearest neighbour points on the grid.

There are two methods of solving this equation. The first method, called the Jacobi method, involves making a new copy of the grid for each iteration so that there are two copies of the grid at all times. Each individual node is calculated based on the values of the neighbouring nodes in the previous grid, meaning every node is updated based on the prior grid. Another method, called the Gauss-Seidel method, involves continually updating each node based on the neighbouring node so that the update is partially calculated from old values and partially new values.

II. SOLVING LAPLACE'S EQUATION IN TWO DIMENSIONS

Here Laplace's equation in two dimensions,

$$\nabla^2\varphi = 0, \tag{4}$$

is solved using the finite difference equation given in Equation **??** with $\rho = 0$. The Jacobi method is used to iterate the grid. The chosen convergence condition is that the maximum change in any node is less than an absolute error tolerance, called ϵ. By default ϵ is set to 10^{-5}, but choosing a different scale for the charge density requires altering this to be appropriate, e.g. if the charges involved are $\mathcal{O}(10^{-6})$, the absolute error tolerance should be $\sim 10^{-11}$.All non-boundary nodes were set to random guesses between 0 and 1. The number of iterations was capped at `max_it`= 10000.

I might chang this.

III. CALCULATING POTENTIAL AND ELECTRIC FIELD OF PARALLEL PLATE CAPACITOR

IV. SOLVING THE DIFFUSION EQUATION FOR IRON POKER