PHY 481 - Fall 2024
Homework 07

Due Saturday October 19, 2024

Preface

Homework 07 focuses on Laplace's equation and solving it using themethod of images ans infinite series solution in Cartesian coordinates. You should become comfortable with setting boundary conditions for PDE problems like this and develop a sense of the process for solving these problems analytically. Also, you will write a 2D method of relaxation solver for Laplace's equation.

1 The method of images

Griffiths Problem 3.10: A uniform line charge λ is placed on an infinite straight wire, a distance d above a grounded conducting plane. (Let's say the wire runs parallel to the x-axis and directly above it, and the conducting plane is the xy-plane.)

1. Find the potential in the region above the plane.
2. Find the charge density σ induced on the conducting plane.

2 Rectangular Pipe: Separation of Variables-Cartesian-2D

A square rectangular pipe (sides of length a) runs parallel to the z-axis (from $-\infty$ to ∞) The 4 sides are maintained with boundary conditions given in the figure. (Each of the 4 sides is insulated from the others at the corners).

1. Find the potential $V(x, y, z)$ at all points in this pipe.
2. Find the charge density $\sigma(x, y = 0, z)$ everywhere on the bottom conducting wall $(y = 0)$. Check the units for your charge density (show us!).

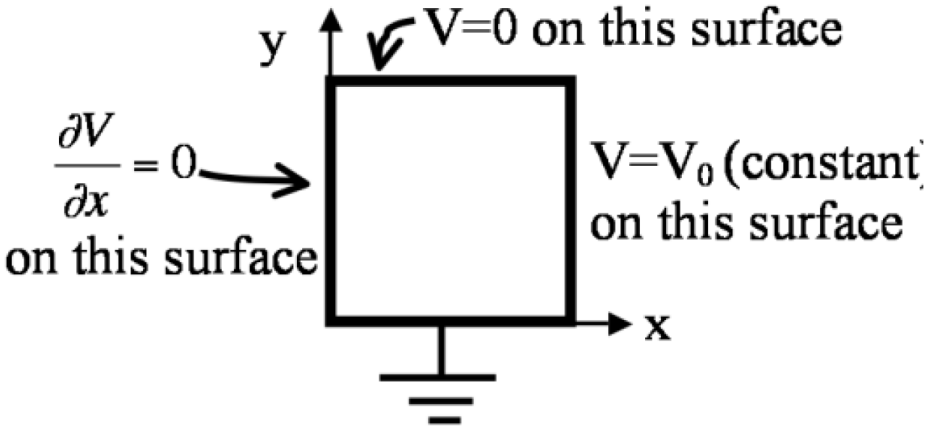

Figure 1: Tube

3 Python: Method of Relaxation in 2D

In this problem, you will write a 2D method of relaxation solver for Laplace's equation. One of the major properties of a solution to Laplace's equation is that the value of the potential at a point is equal the average of all the points surrounding it (i.e., a sphere in 3D or a circle in 2D). We can exploit this property to solve Laplace's equation numerically by successively computing the average value of the potential at a point on a mesh (a grid of 2D points in this case) based on the 4 other points that surround it (see the figure below).

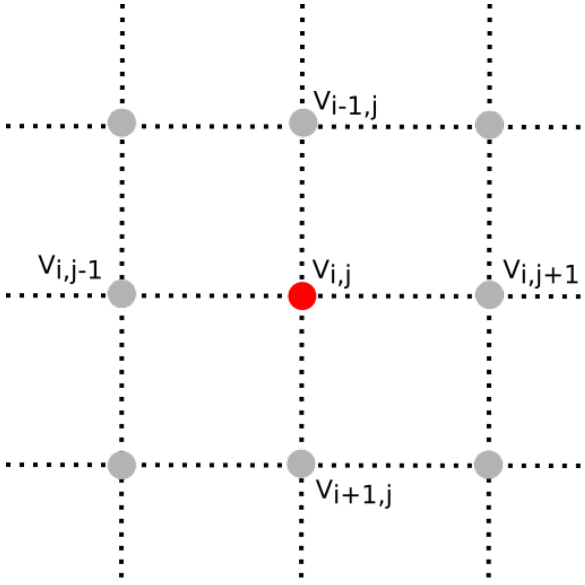

Figure 2: 2D Mesh

To be explicit, in the simplest relaxation codes, which can run for an inordinate amount of time given the size of the mesh and the error tolerance demanded, we replace the value

of the potential $V_{i,j}$ with the arithmetic average of its closest neighbors on the mesh:

$$V_{i,j} = \frac{1}{4}\left(V_{i-1,j} + V_{i,j-1} + V_{i,j+1} + V_{i+1,j}\right)$$

The procedure for solving Laplace's equation numerically involves the following steps:

Step 1: Slice up the space where $\nabla^2 V = 0$ (and the boundary) into a grid of points (called a "mesh") that are spaced an equal distance apart. *That mesh may have different spacing between points based on what the details of the problem being solved might be. For example, if the potential is expected to be change over short distances in some points and not others, it can make sense to change the spacing to optimize computational time (or memory).* In this problem, we will use a mesh of equally spaced points.

Step 2: Set the value of the boundary points given the specific problem you intend to solve. *This will typically be done in the initial parts of the program and can be changed easily to solve other kinds of problems.* In this problem, we will start with a non-zero constant value (10 V) on one edge and zero at the other 3 edges.

Step 3: Starting at some location away from the boundary, systematically loop through each point applying the averaging function given above. *It would be typical to start at one corner of the mesh and move systematically across (or down) and then down (or across) calculating the value at each new point as you go.*

Step 4: Compute the deviation between the starting values of the potential and the values after a full iteration. Compare this deviation to the tolerance that you decided on before starting the calculation, $\mathrm{Error}_{i,j} = V_{i,j}^{\mathrm{new}} - V_{i,j}^{\mathrm{old}}$. *Here, you could use the average error, the maximum error, or something else as your deviation (which must be a single number).* In this problem, you will can choose what deviation to use.

Step 5: Repeat steps 3 and 4 until the deviation is below the tolerance. *Note that you should build in a maximum number of steps to take in case the code doesn't converge on an answer quickly.*

Step 6: Plot the results as either a 3D plot.

Task: Using the code we had in class for 1D as a starting point (File name: Method-of-relaxation-1D.ipynb), expand it to solve the 2D problem. Set the boundaries for x from 0 to 10 and y also from 0 to 10. Set the voltage at one boundary (x=0) to 20 V and the other boundaries to zero. You will have to pick a reasonable step size for the mesh (make sure it can be adjusted!). Produce a 3D Plot of the potential.