Accelerator Physics - Prof. Dr. Yuri Litvinov - Winter Term 2022/23

Exercise 3: Electrostatic Potential*

Discussion on: Wednesday, November 02, 16h15

Introduction:

Charged particles can be accelerated by electric fields. The simplest case is an electrostatic field. If the charge distribution is known, the field is given by Coulomb's law. However, in many cases there is no known charge distribution but voltage sources are used to create potentials from metallic structures. The free electrons in the metal rearrange and the field can be difficult to compute. What is known is that the electrostatic potential ϕ outside of the metal obeys the Laplace equation:

$$\Delta\phi(x, y, z) = 0$$

The electric field is given by the gradient of the potential. Nevertheless it is not possible to get the potential or the electric field directly. An algorithm must be found that returns a potential that obeys the Laplace equation. Although this may sound complicated, the numerical solution to this is rather straight forward. Numerically, the left hand side of the Laplace equation can be approximated by:

$$\frac{1}{d}\left(\frac{\phi(x+d, y, z) - \phi(x, y, z)}{d} - \frac{\phi(x, y, z) - \phi(x-d, y, z)}{d}\right) + \cdots$$

plus equivalent expressions for y and z. d is the distance between equidistant grid points. The approximated form of the Laplace equation is:

$$\phi(x+d, y, z) - 2\phi(x, y, z) + \phi(x-d, y, z) + \phi(x, y+d, z) - 2\phi(x, y, z) +$$

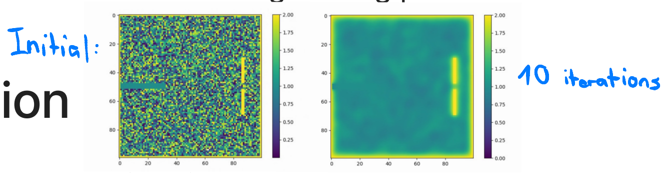
For 2D:

Solved for $\phi[x, y, z]$: $0 = \phi(x+d,y) - 2\phi(x,y) + \phi(x-d,y) + \phi(x,y+d) - 2\phi(x,y) + \phi(x,y-d)$

$$\phi(x, y, z) = \frac{1}{6}(\phi(x+d, y, z) + \phi(x-d, y, z) + \phi(x, y+d, z) + \phi(x, y-d$$

For 2D: $\phi(x,y) = \frac{1}{4}\Big(\phi(x+d,y) + \phi(x-d,y) + \phi(x,y+d) + \phi(x,y-d)$

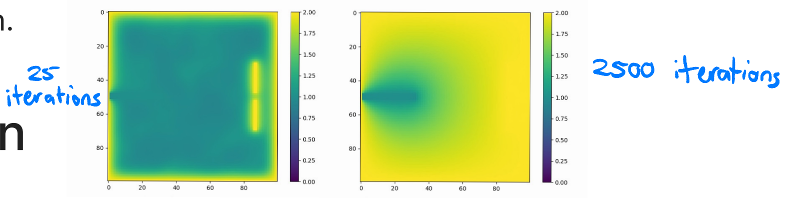
One could say the Laplace equation states that the potential at every point in space is given by the arithmetic mean of the neighboring points.



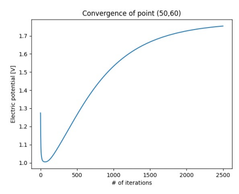
3.1 Iterative Solution

Limit the problem to two dimensions (x, y) and use a grid of $m \times n$ grid points (typically 100 in each direction). We want to accelerate ions from an anode (a rectangular shaped metallic structure) at a potential of $+100$ kV. They fly through an aperture plate (two rectangular shaped metallic structure spaced at the size of the opening) on ground potential 0 V. The anode and the aperture plate should be several grid points large. The remaining grid points should be initialized with random values. Best is to choose values between the anode and ground potential. You can use the sheet3_layout.dat file for the setup.

The actual algorithm is the following. Go through all grid points one by one and set the potential equal to the potential of the neighboring points. Use the value of the current iteration for the neighbors (like in a Gauss-Seidel algorithm). If the grid point is on one of the metallic structures, it keeps its fixed value! Think about how to deal with the borders and corners. Repeat this procedure until the potential does not change too much anymore. Display the result graphically - you can use the `plt.imshow( array2d )` function - and add a plot to your solution.



3.2 Over-relaxation

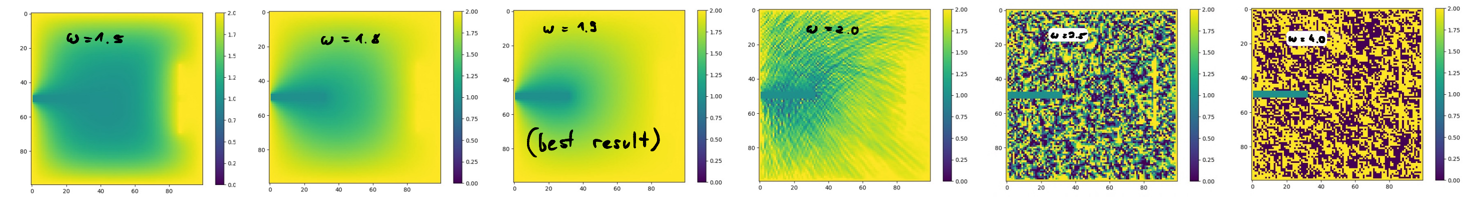


Pick a grid point and plot its potential value as a function of the iteration step in order to investigate the speed of convergence. ~2500 To improve the speed, apply over-relaxation. Keep track of the values of the last iteration and multiply the difference to the current iteration by some factor, typically called ω, before you assign it. If, e.g., the mean of neighbors is 54 V and the old value was 50 V, set the new value to 50 V $+4\omega$ instead. A good empirical value for ω should be $\omega = 1.5$. Play with the values of ω in order to see how large it can be without causing divergence. Add a plot to your solution.

*based on the jupyter notebook by Arthur Boltz

In order to prevent divergence, i added a clamp to the values that the potential can take (being a value between 0 and 2) This yielded some interesting results:

In [ ]: 100 iterations each


ω = 1.5
ω = 1.8
ω = 1.9
(best result)
ω = 2.0
ω = 2.5
ω = 4.0