PHYS 512 - PS3

Damien Pinto (260687121)

October 2019

1)

Code for this question: python3 relaxation.py params.txt

I let the method of relaxation solver run either until the maximum number of iterations were undergone, or until the sum total of the magnitudes of change in values across all pixels from one step to the next dipped below a certain threshold. Working in a 256 pixel cube, setting the potential inside the cylinder of radius 26 to be held constant at a value of 1.0 and the minimum threshold of the total change in value across pixels to 1.0, the solver didn't terminate after 6000 steps and approximately 100 minutes. This produces the output seen in Figure 1.

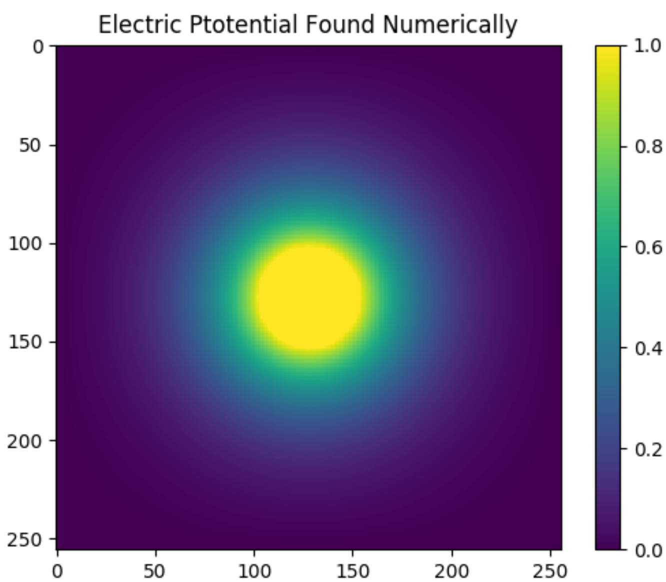

Figure 1: *Electric potential generated with the boundary conditions of an "infinite" cylinder of finite radius with a potential maintained at* 1.0 *inside of it, and then acted upon with the method of relaxation over 6000 iterations.*

The charge density is shown in Figure 2, and we can see that the charge is distributed along the surface of the cylinder. It is not uniform in its distribution, but this is almost certainly due to the jaggedness involved in tracing out a circle in in a pixelated grid. The average surface density is $0.019\frac{C}{u^2}$ where u is some unit that would represent the length our data spans.

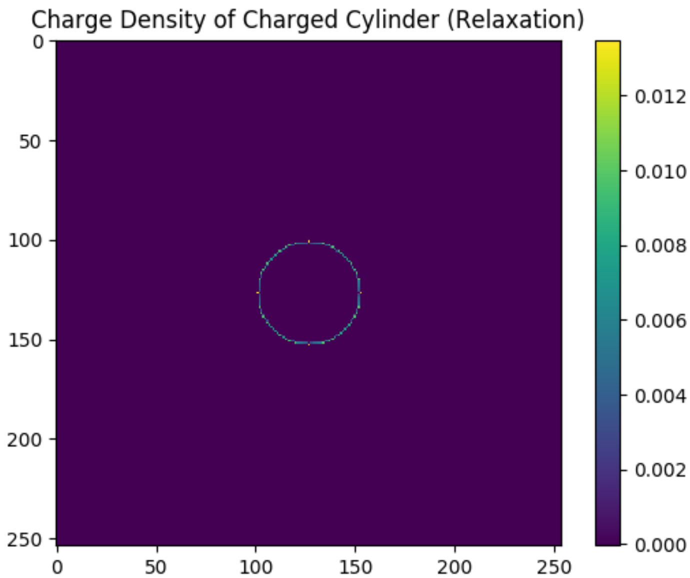

Figure 2: *Charge distribution obtained with the same process as used to produce Figure 1.*

Comparing to the an analytic solution got problematic. I don't know if I just can't remember E&M, but here was my thought process. We're holding the electric potential constant inside the cylinder, which means the the electric field is null, which implies a null charge inside the cylinder. This all makes sense with our results so far.

If we assume that the method of relaxation achieved a suitably appropriate charge distribution for the cylinder to correspond to a constant electric potential inside the cylinder, then using the sum of the charge distribution output by my code should give us a reasonable estimate of the charge q in the cylinder. Combining this with Gauss's law when considering an imaginary cylinder surface around our charged cylinder, then we get:

$$E(2\pi r)l = \frac{q}{\epsilon_0} \tag{1}$$

$$E = \frac{q}{(2\pi\epsilon_0 l)r}, \tag{2}$$

$$\tag{3}$$

when at a distance r from the charged cylinder of length l. Then, using the fact that $E = -\Delta V$, we calculate the electric potential V:

$$V = -\frac{q}{(2\pi\epsilon_0 l)} \int_{\frac{L}{2}}^{r} \frac{1}{r'} dr' \tag{4}$$

$$= -\frac{q}{(2\pi\epsilon_0 l)} [\log{(r)} - \log{(\frac{L}{2})}] \tag{5}$$

$$= \frac{q}{(2\pi\epsilon_0 l)} \log{(\frac{L}{2r})} \tag{6}$$

Where L is the length of the space we are working in (and here we use $\frac{L}{2}$ because the cylinder is positioned in the center of the space). This makes sense to me, as it ensures the potential dies down to 0 at the edges of our space (which occurs naturally in the method of relaxation simulation without me enforcing it), and decreases the further away from the cylinder we get. The problem arises when I try to implement this, I get the results shown in Figure 3.

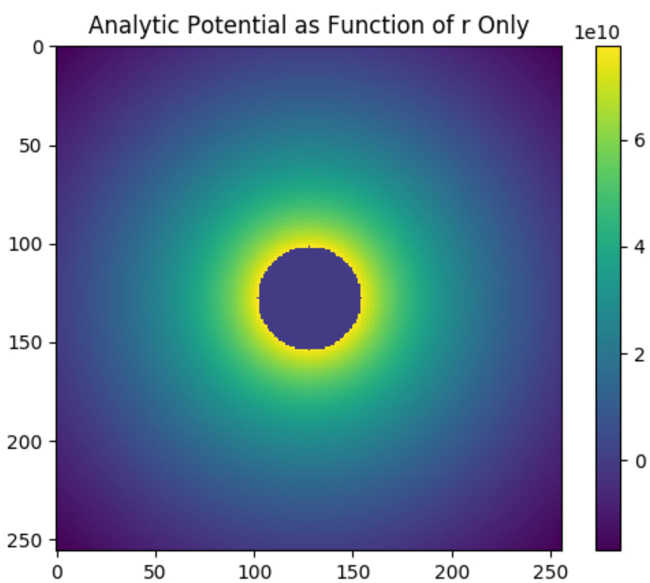

Figure 3: *Analytic potential obtained when applying the E&M analysis done above and sourcing the charge distribution generated by the method of relaxation applied previously.*

As we can see, the scaling is all off. I'm not sure if this is because my assumption/method of sourcing the charge distribution is wrong, or if my derivation is incorrect, but things aren't lining up, as evidenced by Figure 4.

2)

Code for this question: python3 conj_grad.py params.txt

If we keep the standard that we enforced for the method of relaxation, that is that the sum of the magnitude of the changes in values across all pixels being at least of size 1.0 between two subsequent steps, the

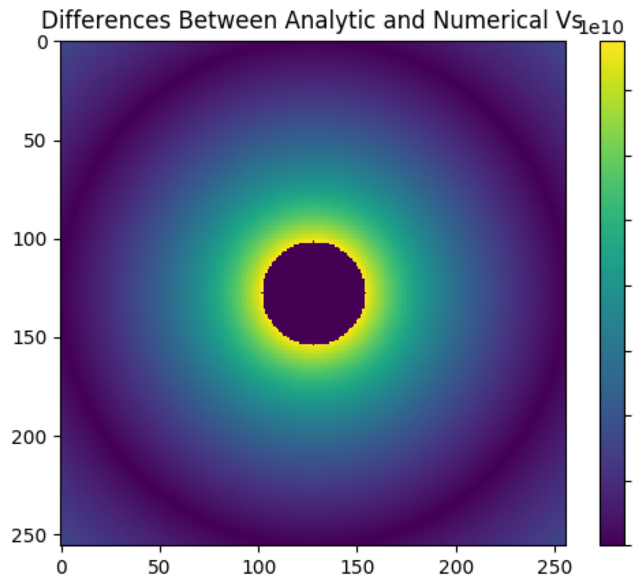

Figure 4: *Differences between analytic and numerically obtained results for the electric potential of a charged cylinder.*

conjugate gradient method achieves such a state after 263 steps and about 810 seconds. (Note: The times cited here all include time the script took to plot the current state of the system in real time) The results can be seen in Figure 5.

While trying to just see what form of the log function actually fit my data I found that the data returned by the conjugate gradient method corresponded closely to Eq. 7, as illustrated in Figure 6.

$$V(r) = \begin{cases} 1.0 & \text{if } 0 \leq r < R \\ 0.6 \log\left(\frac{L}{2r}\right) & \text{if } R \leq r \end{cases} \tag{7}$$

3)

Code for this question: python3 res_conj_grad.py params.txt

In this section, I reduced the resolution by a factor of 2 a total of three times. This means the program operated on resolutions 32^3, 64^3, 128^3, and 256^3. The program then applied the conjugate gradient method until the sum of the magnitudes of all the changes in the pixels of the image was less than 10^{-3} from one step to another.

We can see immediately, during the simulation, that the progression of the field outwards from the cylinder towards the edge of the space much quicker than the simulations of the previous numbers. This in terms of fraction of the distance between the edge of the cylinder to the edge of the space covered per iteration. This is without a doubt due to the drastic reduction in number of elements in the matrix operations.

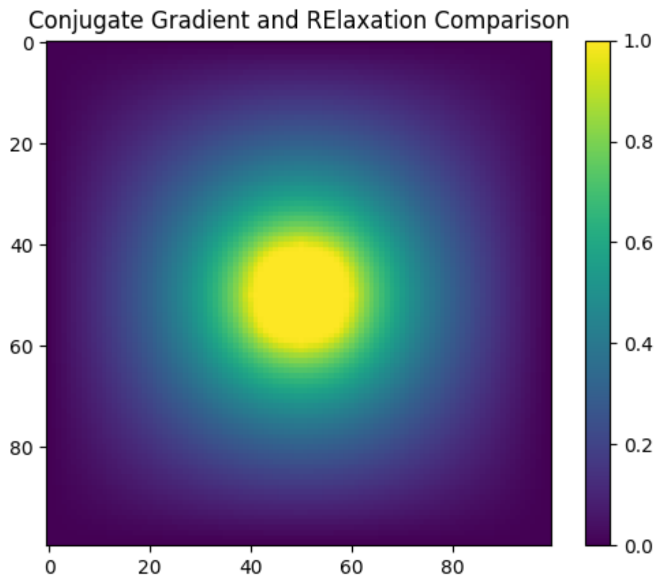

Figure 5: *Electric potential with same boundary conditions as in 1), but iterated over with the conjugate gradient method for iterations.*

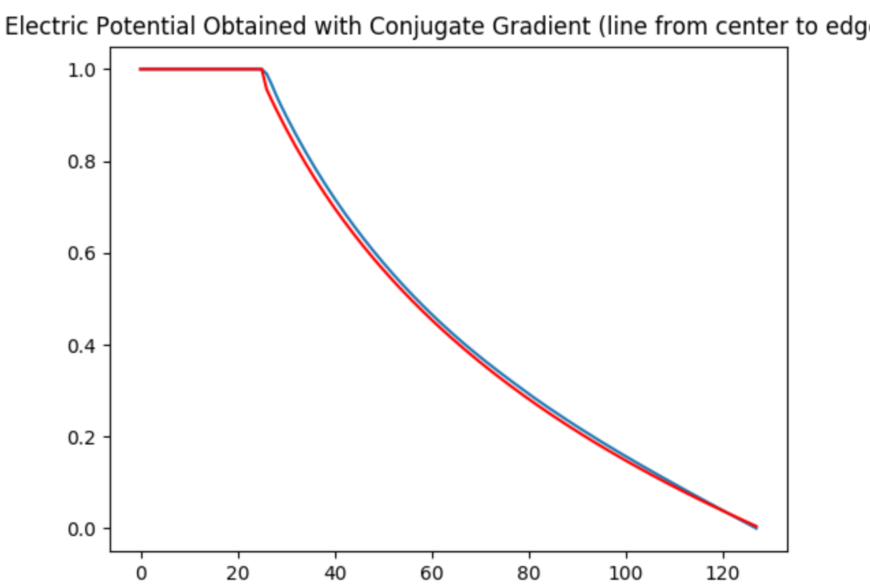

Figure 6: *Modeling of the data obtained through the conjugate gradient method (blue) with a hand-tailored* log *function (red).*

This method of progressively increasing the resolution of the system being acted on resulted in attaining the pre-determined degree of precision in 153 steps and about 210 seconds (again, this includes the time taken to plot the progression of the simulation in real-time). The results of this can be seen in Figure 7

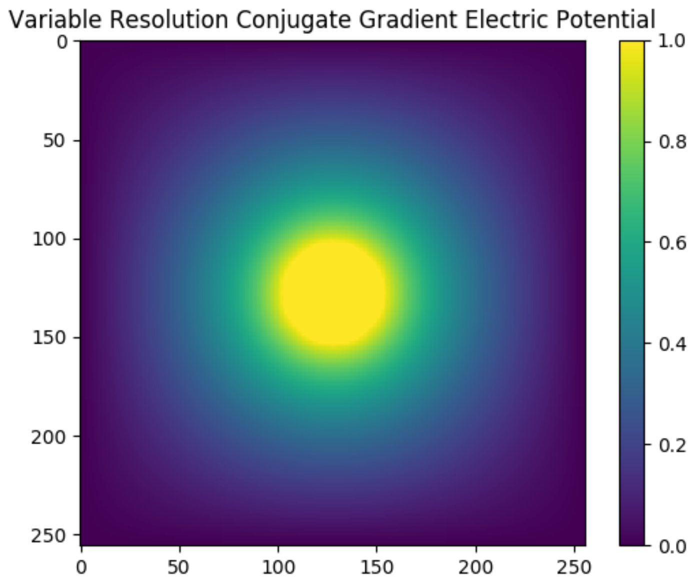

Figure 7: *Electric Potential obtained by applying the conjugate gradient method on our system at reduced resolutions and progressively increasing the resolution once a threshold of precision is met.*

One caveat that one can see by overlapping the output electric field in the previous section and that output by this number's simulation is that the coarseness of the lowest resolution has some effect on the highest resolution's shape. Somewhat predicatably, the integrity of curves is sacrificed in return for the increase in speed. By looking at "cg_V.png" from the previous number and "var_res_conj_grad_V.png" and switching quickly between the two, I saw that the latter has less of a circular shape to its field, and has faint remnants of "corners" which I think are from the lower-resolution stages.

4)

Directory: 4/Radial Bump/
Code for this question: python3 bump_conj_grad.py params.txt

Because of the structure of the code I use to generate my cylinder and masks, I let the cylinder have two bumps, but diametrically opposed. The angular width of these bumps is 20 degrees, and their width is a tenth of the radial radius of the cylinder. The simulation ran for about 265 seconds and underwent 156 iterations. The results can be seen in Figure 8.

In Figure 9, we can observe the electric field generated by this electric potential. We can see that the value of the field at about 2 pixels from the surface of the cylinder is about $0.0243u$. If we then move to Figure 10 and examine the field,the largest value it that is generated at a distance of two pixels is a value of $0.0326u$. This represents an increase in generated electric field of $\approx 30\%$.

5)

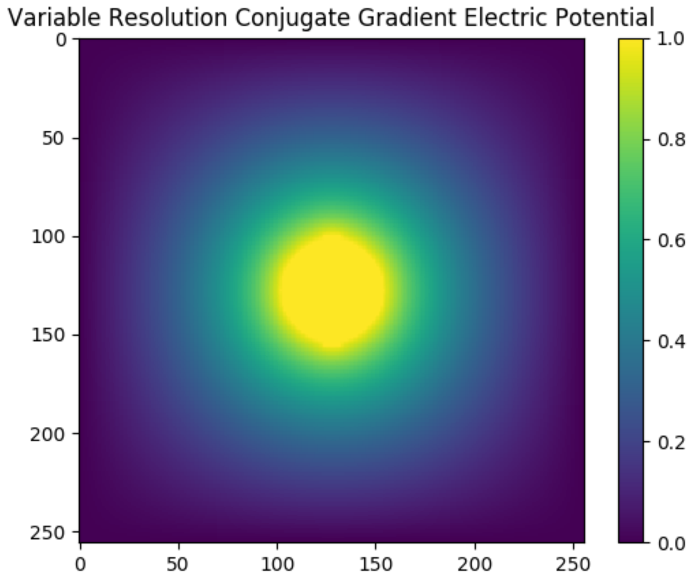

Figure 8: *Electric Potential produced by letting a conjugate gradient simulator (utilising variable resolutions) iterate over the initial condition of a cylinder with two diametrically opposed "bumps" and being held at a constant potential.*

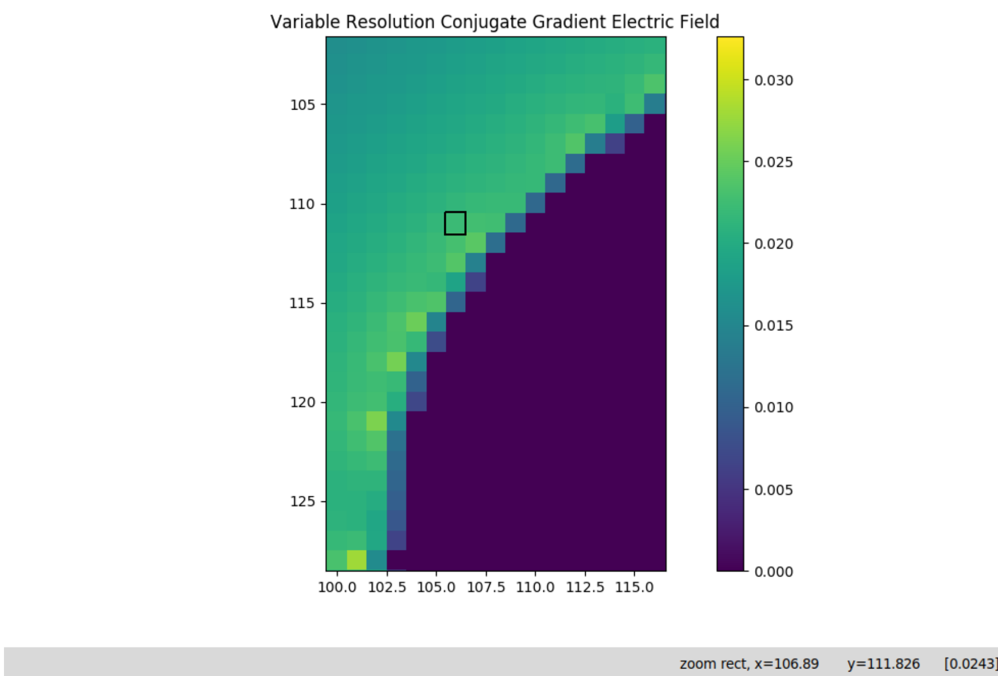

Figure 9: *Electric Field generated at a two pixel distance away from a cylinder of constant potential. On the bottom-right corner of the figure we can see that the value of the field at the point delineated by a black box is 0.0243 units.*

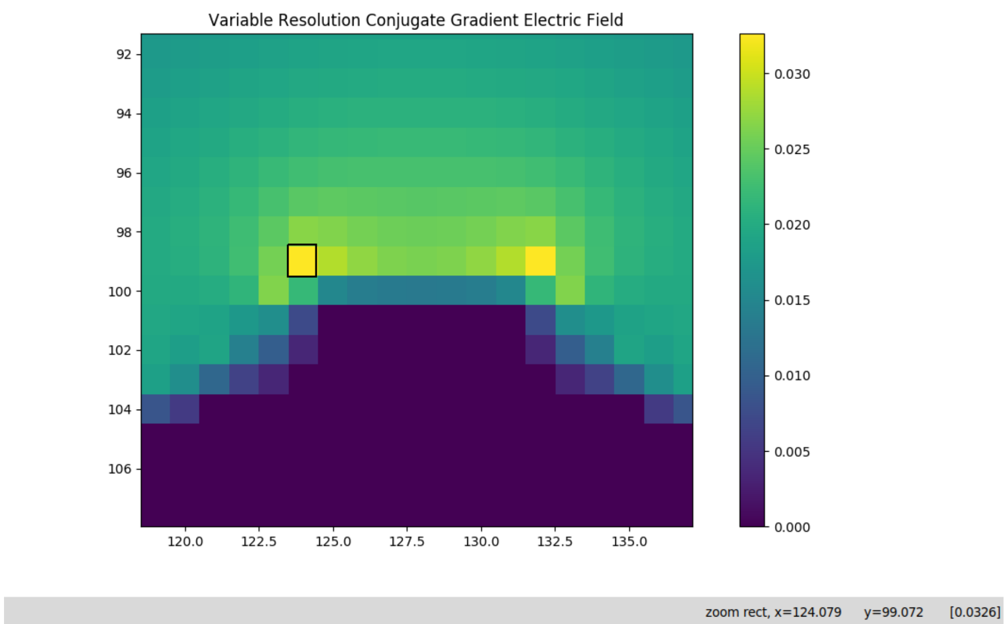

Figure 10: *Largest instance of the Electric Field generated at a two pixel distance away from a bump on cylinder of constant potential. On the bottom-right corner of the figure we can see that the value of the field at the point delineated by a black box is 0.0326 units.*