

Assignment VII, PHYS 409 (Electromagnetism I)
Fall 2019 Due Thursday, October 10th at start of class

This is an unusual assignment; please read the following instructions carefully.

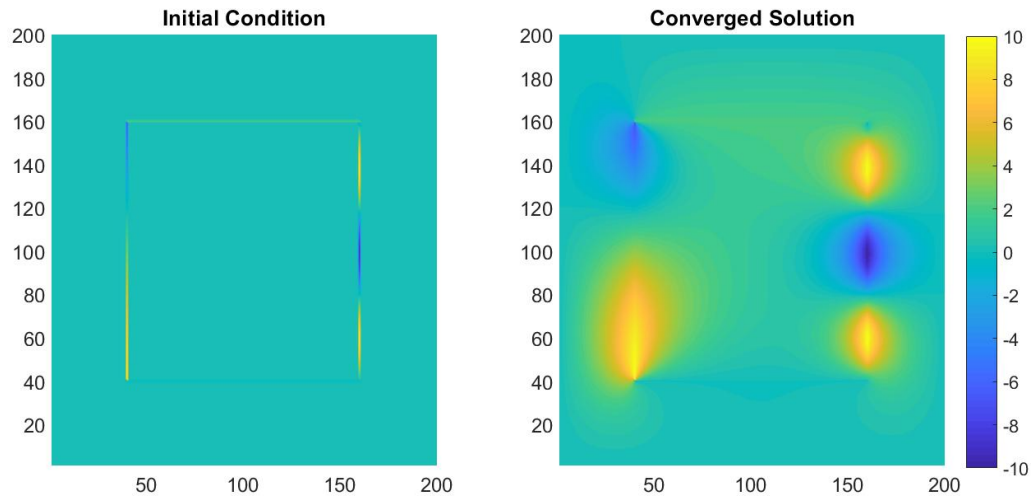
You may or may not be aware of this, but one of our stated assessment goals for both our BA and BS programs in Physics is to demonstrate that our students develop computational problem solving skills. To help us assess this, PHYS 409 has been chosen by the powers that be (not me) as a place we determine if you have developed these skills. The way we will do this is through the assignment below. This assignment counts as a normal homework assignment, but you will be given a bit more time than usual to work on it. That being said, you probably won't get this back on the normal schedule (which shouldn't matter too much – the results from this homework assignment will have little impact on your understanding of the other parts of the course content).

You will have other homework to deal with between now and when this homework is due, but I *highly* recommend you don't wait until the last minute to attack this homework. TL;DR version – START THIS NOW! There won't be any additional lecture content that will make this any easier for you.

Recall that a general property of solutions of Laplace's equation $\nabla^2 V = 0$ is that an extreme value occurs on the boundary of the domain. Another important property is that for any point in the domain of the solution, the value of $V(\vec{r})$ is equal to the average of the neighboring points. This latter property allows an interesting method of numerically finding a solution to Laplace's equation, generally called the "method of relaxation". The idea is that the value of $V(\vec{r})$ on the boundary of some region is fixed, and the values everywhere else in the domain are initialized to 0.

The solution occurs when this field is successively averaged. The basic algorithm to solve such a problem in 2-D (to find $V(x, y)$) is outlined in the attached algorithm description.

There are other algorithms that can do this more efficiently, but this can work as a proof-of-principle. For example, my simple 20-line MATLAB code can solve a 200 x 200 system with tolerance of 1×10^{-5} V in about 4.5 seconds and 9000 iterations; below I show the initial field on the left and the final field on the right. Just from something simple like this we can get a pretty good sense of how the field looks. (The figure shows a top-down view of the boundary conditions on the left (with x and y in pixel numbers) and the converged solution on the right (with x and y in pixel numbers). In both cases, the z axis (indicated by color) represents the electrical potential at the specified point in the 2-dimensional domain.



To see how the algorithm works, I have also created a video (https://youtu.be/VcpIQ10p_zg) that shows you what the first 200 iteration steps of my algorithm look like in solving this particular example.

I could make this a big ordeal for y'all and ask you to make a movie, try a bunch of different algorithms to see what converges fastest, etc. – but my goal here is to just satisfy the minimum departmental assessment criteria. (I'm sorry you have to do this, but this is the literal least I'm allowed to give you). There is a fair amount to be learned here, but I don't want this to be a significant fraction of our course efforts for the semester since we only get one semester of Electromagnetism. As such, I'm only asking you to solve one geometry each – and I'm just going to ask for a few things from your solution.

For each of you I have generated a 200x200 matrix. I don't care if you use MATLAB, Python, IDL, or any other computer language of your choice. To prevent you from being tied to any particular programming environment I have left your data files as raw text files. Each row has 200 numbers separated by commas. There are 200 total lines in each data-set. Anything that's a zero is part of the domain to find the potential in, and anything that's not zero is part of the specified boundary condition.

What are you turning in?

- A printout of your source code to run this relaxation algorithm (in whatever language or environment you did your analysis in).
- A hard-copy of a figure showing the initial condition and converged solution, similar to what I have shown above.
- The value of your converged solution at a few particular points in the domain. The class before this is due, I will hand you an index card with 5 row/column pairs – you are to give me the value of the electrical potential at those points.

To help you test your code, you can download the initial condition for the system I solved above at http://larsenml.people.cofc.edu/Larsen_files/example.csv. If you run a working code on this initial condition with a tolerance of $10 \mu V$, you should get the following values:

row	column	potential (V)
34	26	2.3235
44	36	6.8543
100	100	1.2109
135	135	2.4177
19	114	0.0090

If you need help/advice, my assistance will be in MATLAB – that’s the language I’m most comfortable with. MATLAB is available in the department’s large computer lab, and you can get it on your own computers for free. Check out <http://larsenml.people.cofc.edu/lab-internals.html> for some resources that might help you.

Important additional note! I know some of you have a lot more programming/simulation experience than others. If you have done this sort of thing before (or are trying to code it for the first time), you might wonder what I did for the domain boundary – e.g. the first row, last row, first column, and last column don’t HAVE 8 neighbors. Although normally you might want to institute “periodic boundary conditions”, I didn’t want this to be that fancy – that would have made my solution go from 20 lines to about 30. Rather, I only applied my algorithm to the 498×498 domain in the center of my system (essentially treating the zeros on the outside edge like another fixed boundary condition). Please do the same in your solutions.

Your personal case can be found at http://larsenml.people.cofc.edu/Larsen_files/Yourlastname.csv. Replace Yourlastname with your actual (capitalized) last name. If you have problems downloading the file, let me know.