ECE5340/6340: Homework 6
ANSWER KEY

Department of Electrical and Computer Engineering
University of Utah, Salt Lake City, Utah
February 22, 2012

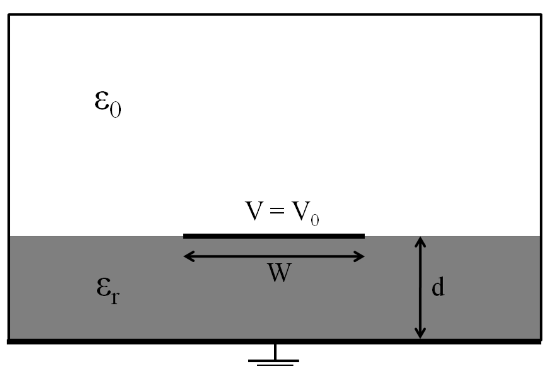
Write your section (ECE5340 or ECE6340) by your name. Turn in a printed copy containing the problem solutions, plots, and the code used to generate them. Remember to comment and format the code so it is legible to the graders. Label the plots appropriately, including units for each axis and for the values plotted. Assume all units to be SI units unless stated differently. Due Wednesday 2/22 BEFORE class begins.

ASSIGNMENT

1. Starting from the generalized Poisson equation, derive Equation (42) from the notes.

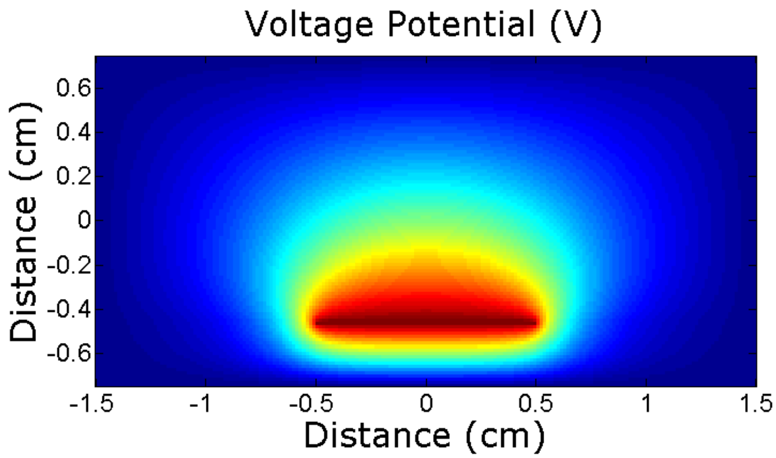
   This is just a copy/paste from the notes again. Show that you understand what is going on by explaining each step along the way.

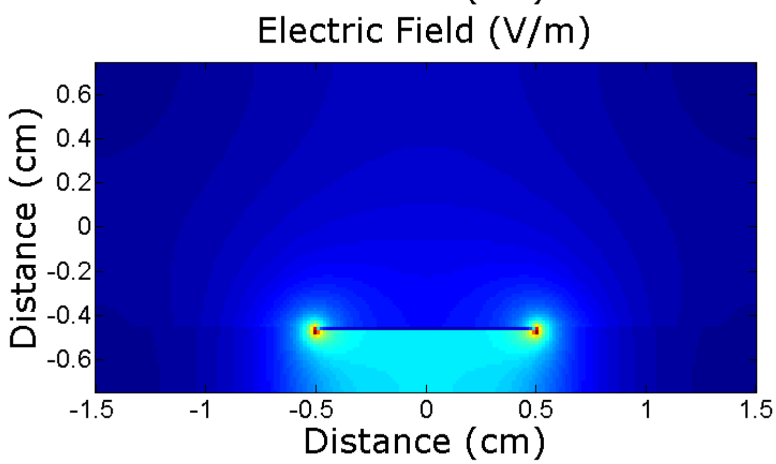
2. Write a Matlab function that simulates the voltage distribution of a microstrip line. Allow the user to input the dimensions (W and d) of the microstrip as well as the dielectric constant ϵ_r of the substrate and the voltage difference V_0 between the top conductor and the ground plane. Use the system geometry depicted below:



   - Test your simulation under the conditions $W = 1.0$ cm, $d = 3.0$ mm, and $\epsilon_r = 3$. Use a value of $V = 1.0$ V for the potential of the top conductor. Use only Dirichlet boundaries of 0.0 V along the borders of the simulation domain. Choose a simulation boundary size and grid spacing that you think is appropriate in order to obtain accurate results. Plot the voltage and electric field distributions before calculating characteristic impedance.

See images below. I deliberately left things a little amiguous so that students will have to think about what parameters should give a reasonably accurate simulation result. I chose to use a simulation length of $3W$ and a simulation height of $5d$. The grid spacing was set so that $h = d/20$.





The analytically calculated impedance was 41.6 Ω while the simulated value came in at 38.0 Ω. Slightly better agreement can be achieved by using smaller grid spaces or by moving the boundaries further away. Of course, the trade-off is longer simulation time. The analytical computations also have buried assumptions that we are not necessarily aware of, making perfect agreement unlikely.

- Resimulate your microstrip line by applying a Neumann boundary condition along the center of the domain. Recalculate the characteristic impedance. Did you get the same value? How much time and memory did you save by exploiting symmetry?

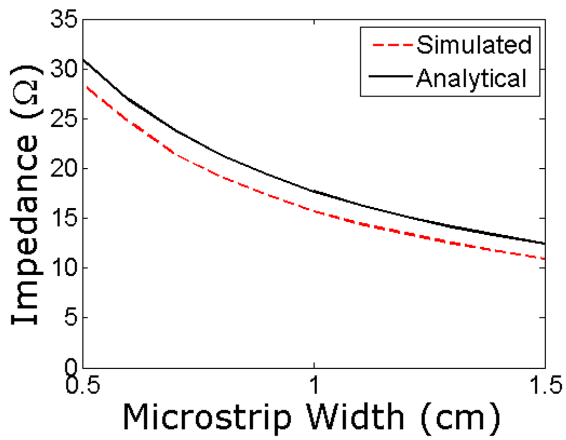
  If done correctly, the output from this should be almost idential to the previous example. There are multiple ways to make it work, though. One variation is to simulate only the half-sized domain and then copy/paste a mirror image into a full-sized result. Doing this creates an overlap along the Neumann boundary since it gets copied into the mirror image. An accurate result therefore requires that this overlapped boundary get removed from one of the halves of the mirror.

  Another variation is to simply perform half of the Gaussian integral on the half-sized domain and then multiply the stored charge by 2. This should give accurate results if done properly. An image of the half-sized simulation domain is therefore acceptable as well.

  The key point here is that symmetry makes things run a whole lot faster because we cut the size of the voltage potential matrix down by half.

- Generate a graph of Z_0 vs W for $d = 1.0$ mm and $\epsilon_r = 3$ over the interval $W \in [0.5, 1.5]$ cm. Choose 11 evenly-spaced values (0.5 : 0.1 : 1.5) along the interval and simulate them with your program. Compare these values against the analytical expressions by superimposing them on the same graph (Be sure to cite your source for the analytical expressions). How well does your simulation agree with the closed-form expressions?

  This can take a while if your simulations are slow, but the whole thing can just be automated if you are clever. Typical simulation results are shown below:

  Notice that they both behave pretty similarly, but with a slight error between them. This is to be expected from simulations when compared against analytical solutions. Both the analytical value and the simulated value have some intrinsic errors buried within them, but they should both exhibit the same general trends. My average error here is about 10 %, so anything less than 15 % should be acceptable.