

Problem #1 – Revise Formulation of 1D FDTD

Revise your 1D-FDTD formulation document to include the following additions:

- Dirichlet boundary conditions
- Calculating the grid resolution
- Courant stability condition
- 1D perfectly absorbing boundary condition
- Gaussian source (simple soft source)
- Block diagram or outline summarizing the calculations and the order they should be implemented

Derive the necessary equations for the above items and formulate the equations necessary to be implemented in MATLAB. Show and explain each step in the formulation in as much detail as possible.

Hint: This work is essentially just adding detail and explanations to the material discussed in class. Pretend you are creating a document for yourself that you will use ten years from now when you have forgotten most of what we discussed in class, but you have to implement and understand the FDTD method.

Problem #2 – Revise MATLAB Implementation of 1D FDTD

Revise your 1D-FDTD MATLAB code for the E_y/H_x mode to include the following additions:

- Include perfectly absorbing boundaries on both the z -low and z -high sides of the grid
- Incorporate a simple Gaussian soft-source that emerges from the center of the grid. Modify just the E field at the center of the grid to do this.
- Plot the E and H field at each iteration using your `draw1d()` function so that you can watch the fields during the simulation.
- Follow your block diagram exactly.

Follow the exact algorithm you outlined in Problem #1 and use the exact equations you derived. Capture several intermediate steps using `draw1d()` to show that your code is working correctly. You may continue to use the following parameters or revise any of these as you see fit.

$$\begin{array}{lll} N_z = 180 & \Delta z = 1.4286 \times 10^{-8} \text{ m} & \epsilon_r(z) = 1.0 \\ \text{STEPS} = 1000 & \tau = 1.2 \times 10^{-15} \text{ sec} & \mu_r(z) = 1.0 \end{array}$$

Hint: Inside the main FDTD loop, follow your plot commands with `drawnow`. This will make sure the graphics are updated at each iteration.