Fields & Waves II

Project III - An Introduction to the Finite Difference Time Domain

Aaron Rosen

Team Partners: Jade Gullic & Sydney Northcutt

Oklahoma State University

Fields & Waves II - ECEN-3623

22 February 2022

A20198898

Contents

1 Introduction

This assignment will stretch your ability to program in MATLAB, and to develop a basic simulation tool that demonstrates plane wave interactions. Develop a brief report with figures and captions to demonstrate your success. Provide a brief description of your responses to questions in the problem sets. Organize your report by section numbers corresponding to the problem set numbers.
All plots must have figure numbers and captions. In a paragraph below each plot provide a discussion of the plot highlighting important characteristics. You need to find three things to say about each plot you provide. Include as an appendix, your MATLAB code used to obtain your results. Remember - although I am encouraging you to work together to get the code working this is an individual assignment and you are responsible to provide a written discussion of the problem sets above. More importantly, you need to understand what the code is telling you about the interactions and explain why the results make sense.

2 Problem Set 1.1

2.1 1.1.1

Get the program fdld_1.1.c running.What happens when the pulse hits the end of the array? Why?

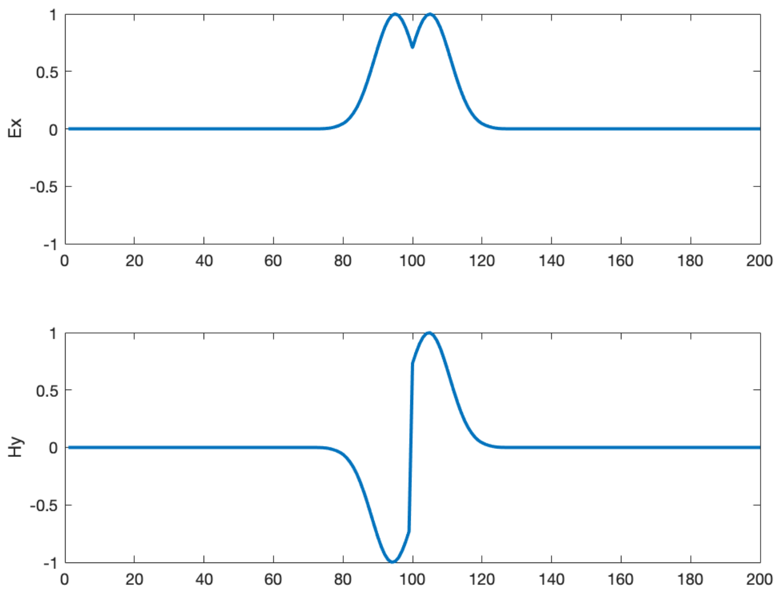

Figure 1: Gaussian Pulse Initial

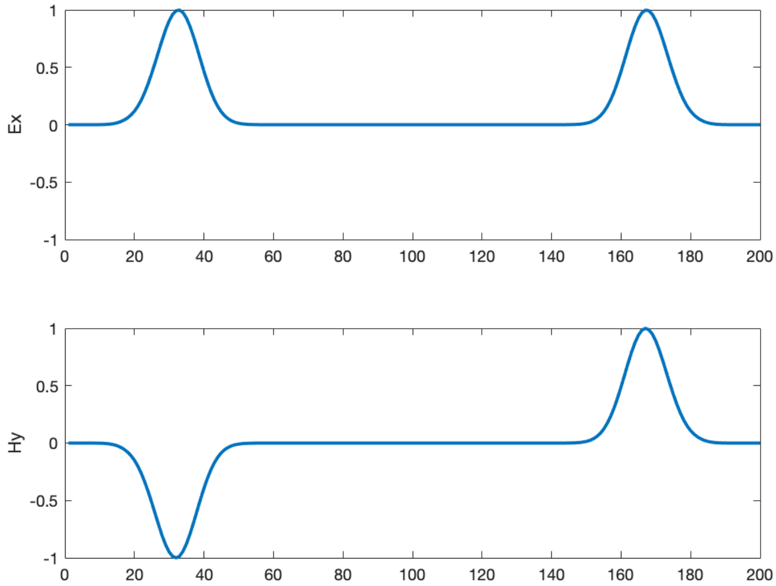

Figure 2: Gaussian Pulse After Some Time

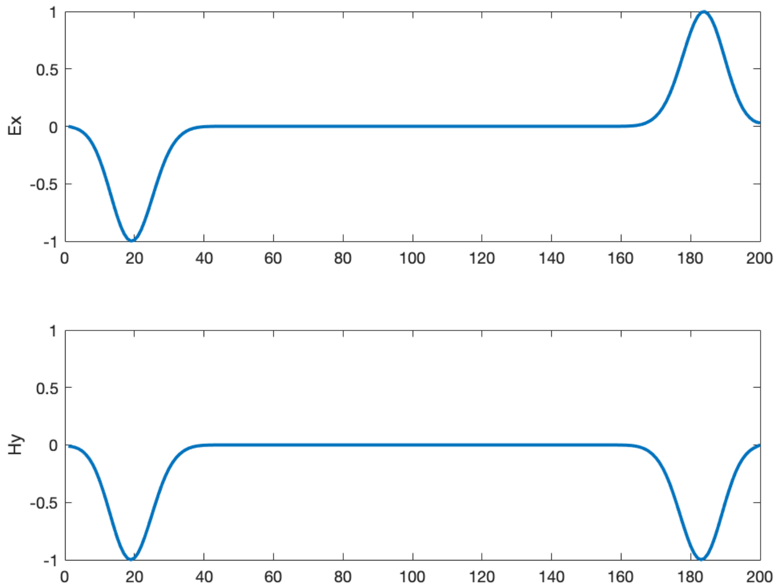

Figure 3: Gaussian Pulse After Hitting Array Boundaries

Following getting the program running, the wave of the Gaussian pulse for $NSTEPS = 500$ demonstrates the wave beginning propagation at $kc = 100$, traveling towards (0 & 200) array bounds, and reflecting. The wave acts as expected. The phase change after the array boundary/ reflection is caused by the equal and opposite force exerted by the medium on the wave.

2.2 1.1.2

Modify the program so it has two sources, one at $kc-20$ and one at $kc+20$ (Notice that kc is the center of the problem space). What happens when the pulses meet? Explain this from basic EM theory.

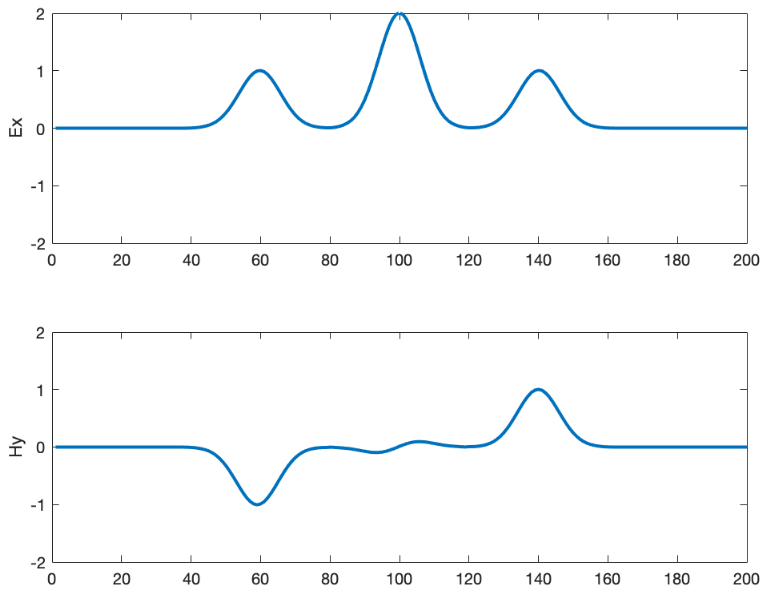

Figure 4: Gaussian Pulse Slightly After Initial

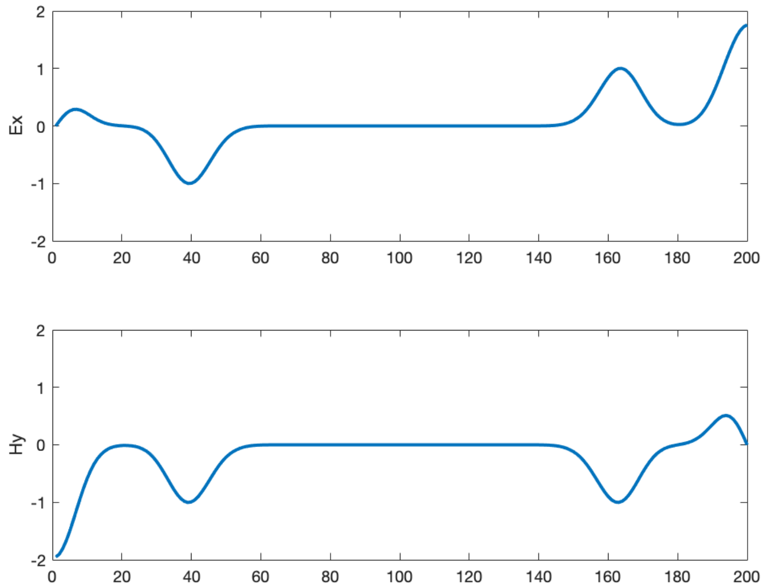

Figure 5: Gaussian Pulse After Some Time

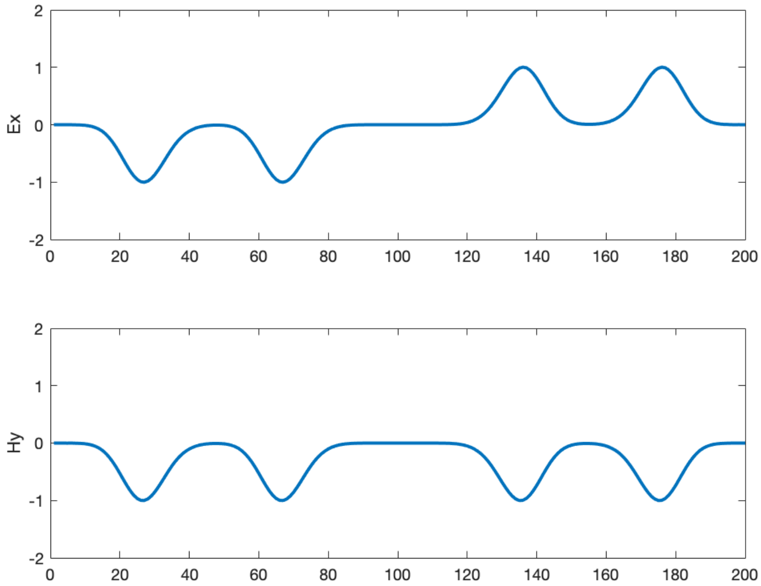

Figure 6: Gaussian Pulse After Hitting Array Boundaries

This demonstration showed two Gaussian Pulses with a width of 40 propagating. The increase in amplitude is expected because of constructive interference. Following the wave hitting the boundaries/ array limits, the wave(s) reflect, similar to problem 1.1.1.

3 Problem Set 1.3

3.1 1.3.1

The program fdld_1.2.c has absorbing boundary conditions at both ends. Get this program running and test it to ensure the boundary conditions are completely absorbing the pulse.

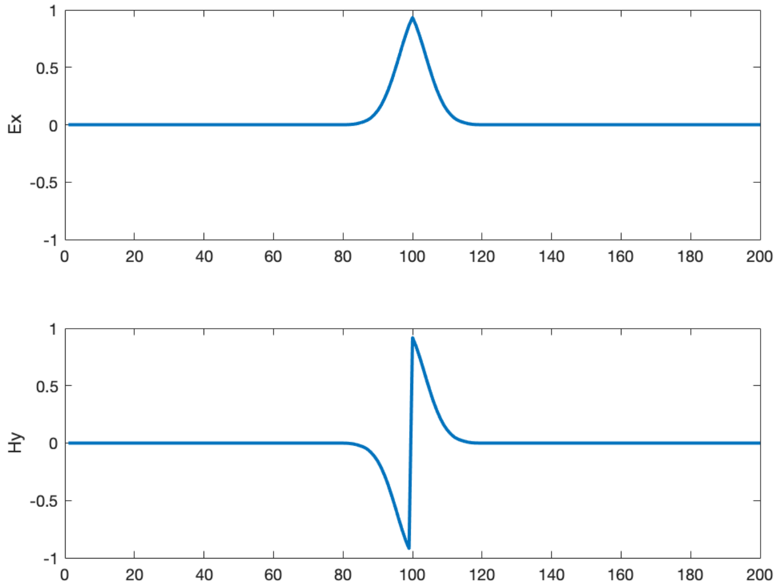

Figure 7: Gaussian Pulse Slightly After Initial

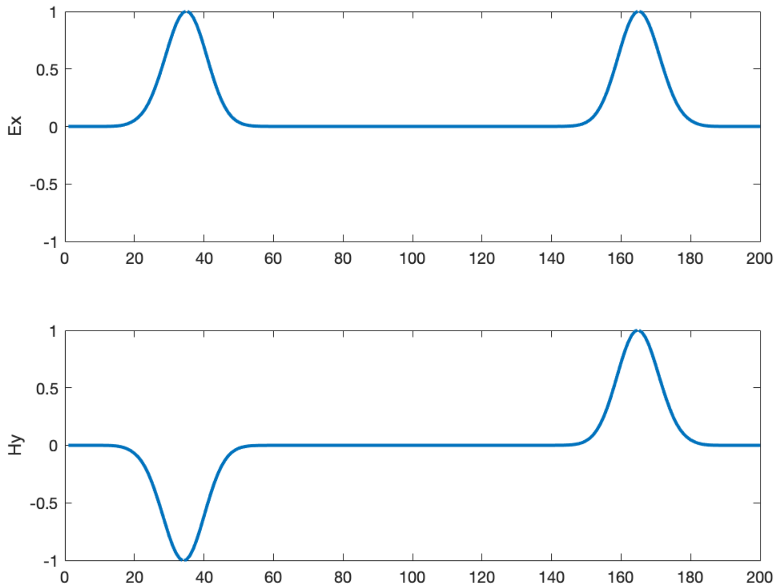

Figure 8: Gaussian Pulse After Some Time

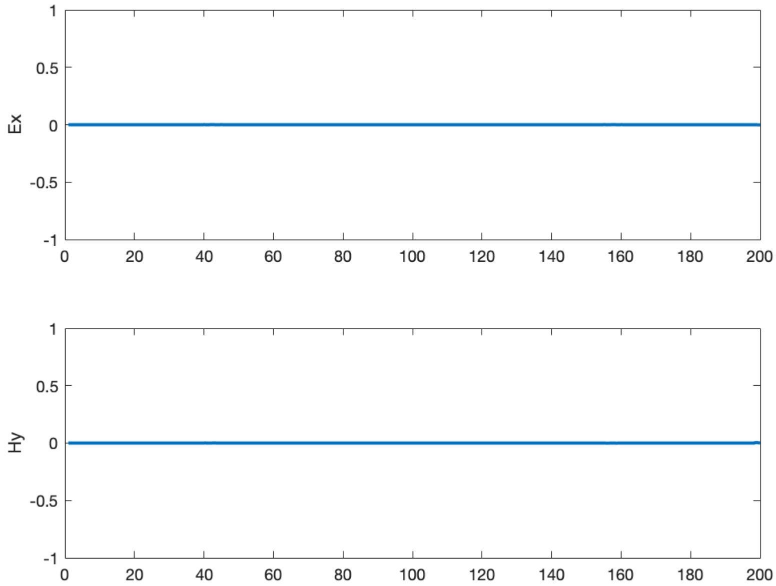

Figure 9: Gaussian Pulse After Hitting Array Boundaries

The program fd1d_1.2 acts similarly to fd1d_1.1; however, in this program the Gaussian wave is absorbed by during the boundary conditions/ array limits. This absorption is shown in Figure 9 where the pulse goes to zero and remains at zero. This confirms the program is working correctly.

4 Problem Set 1.4

4.1 1.4.1

The program fdld_1.3.c simulates a problem partly containing free space and partly dielectric material. Run this program and duplicate the results of Fig. 1.4.

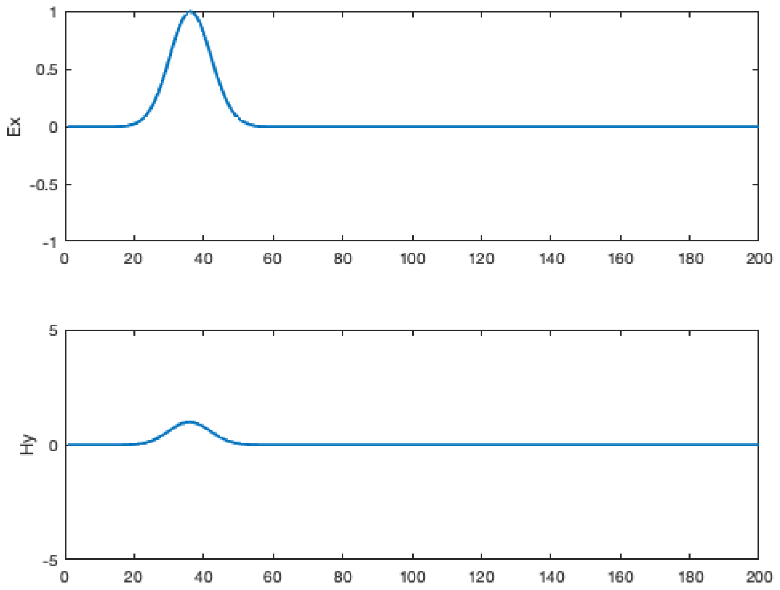

Figure 10: Gaussian Pulse @ T = 100

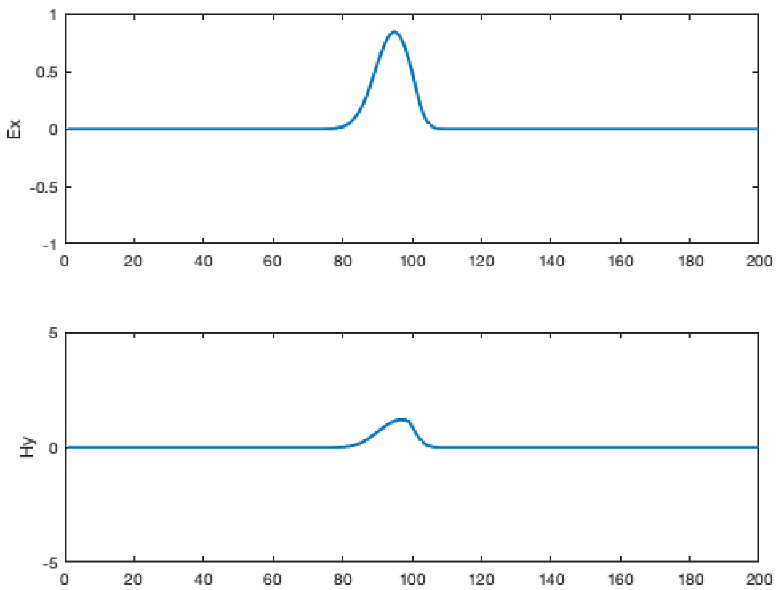

Figure 11: Gaussian Pulse @ T = 220

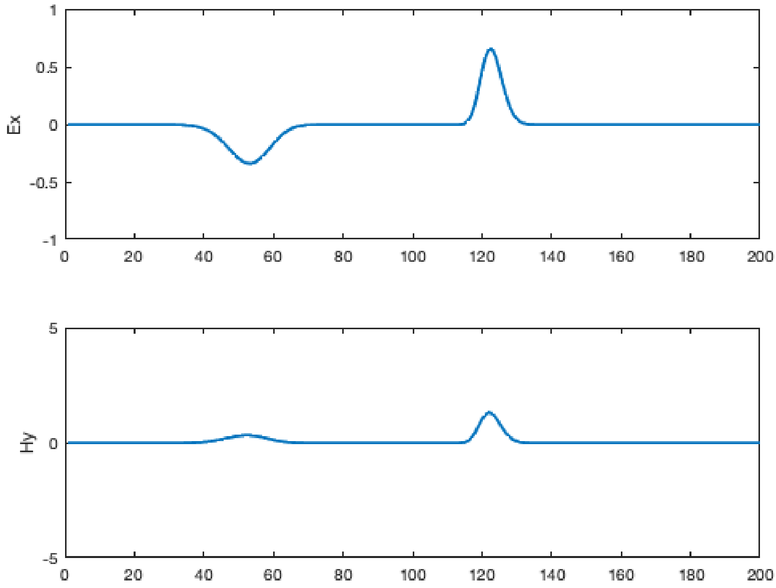

Figure 12: Gaussian Pulse @ T = 320

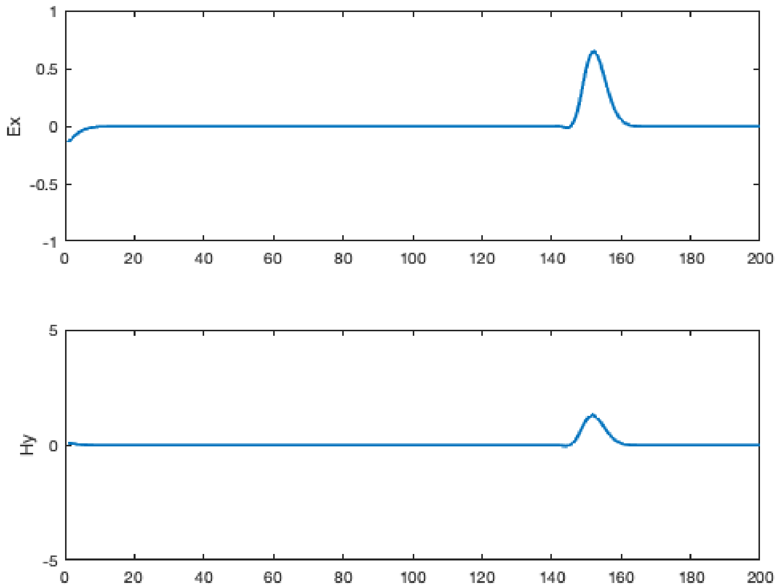

Figure 13: Gaussian Pulse @ T = 440

The figures [Figure 10-13] are duplicates of the images provided in the Sullivan PDF. Given these photos are the same, the code was replicated properly.

4.2 1.4.2

Look at the relative amplitudes of the reflected and transmitted pulses. Are they correct? Check them by calculating the reflection and transmission coefficients. (See the appendix at the end of this chapter.)

Reflection Coefficient

$$\Gamma = \frac{\eta_2 - \eta_1}{\eta_2 + \eta_1}$$

Transmission Coefficient

$$\tau = \frac{2\eta_2}{\eta_2 + \eta_1}$$

Calculation for Eta

$$\eta = \frac{\eta_o}{\sqrt{\epsilon_{r1}}}$$

Observed Values

$$\eta_o = 120\pi,\ \epsilon_{r1} = 1,\ \epsilon_{r2} = 4$$

Calculations

$$\eta_1 = \frac{120\pi}{\sqrt{1}} = 120\pi$$

$$\eta_2 = \frac{120\pi}{\sqrt{4}} = 60\pi$$

$$\Gamma = \frac{120\pi - 60\pi}{120\pi + 60\pi} = -\frac{1}{3}$$

$$\tau = \frac{2(60)\pi}{120\pi + 60\pi} = \frac{2}{3}$$

Following calculating these values, the wave propagation observed is correct. After hitting the second medium, the wave reflects, reduces $\frac{1}{3}$, and changes phase demonstrating the negative portion of the calculation. The wave continues propagating at a reduction of $\frac{2}{3}$ showing the transmission coefficient is also correct.

4.3 1.4.3

A type of propagating wave function that is of great interest in areas such as optics is the "wave packet," which is a sinusoidal function in a Gaussian envelope. Modify your program to simulate a wave packet.

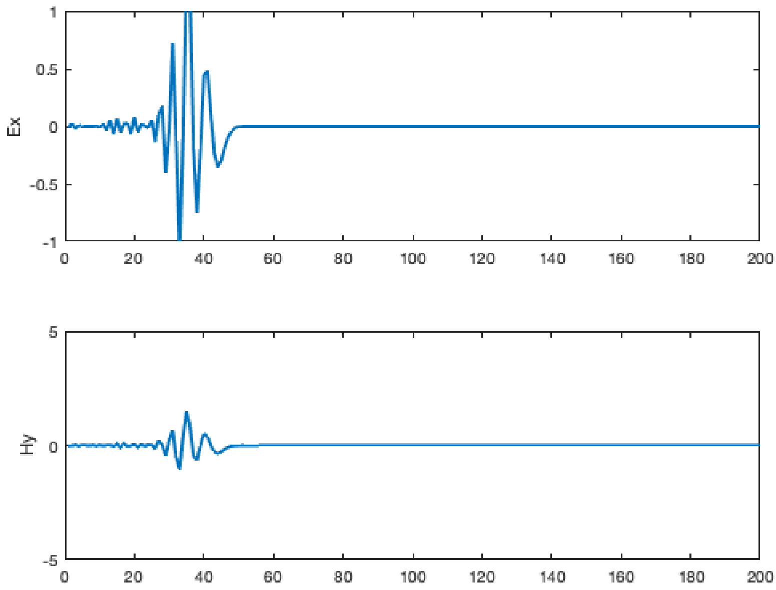

Figure 14: Wave Packet Propagation

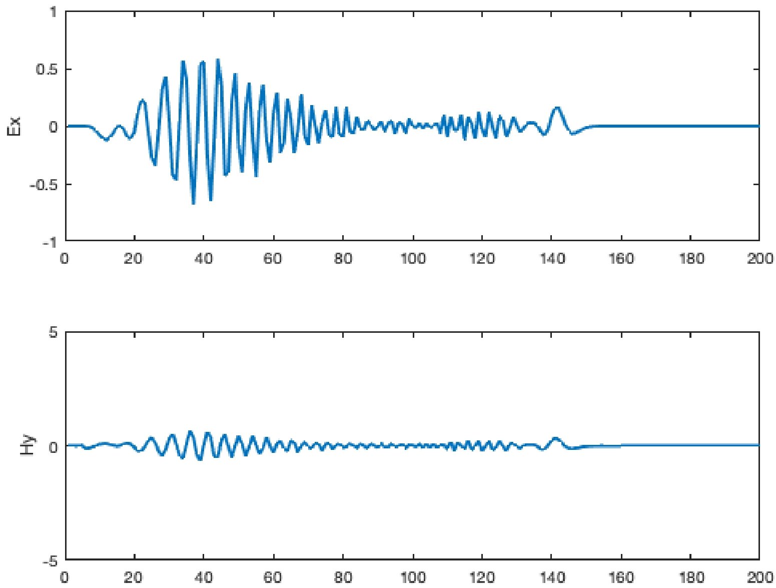

Figure 15: Wave Packet after Reflection/ Transmission off Second Medium

In Figure 14 & Figure 15 a wave packet is shown propagating. The wave packet interferes with the second medium, reflects back, and transmits. I expected more reduction in the reflection amplitude. This may be due to the wave packet having constructive interference with itself though. I also expected more transmission amplitude, but I am not sure of an explanation for this.

5 Appendix

5.1 1.1.1

```
clear all; close all; clc;

%Aaron Rosen - Fields & Waves II
%Project 3 - An Introduction to the Finite Difference Time Domain


%Rebuilding  FD1D_1.1.c in MATLAB
KE = 200;
Ex = zeros(KE, 1);
Hy = zeros(KE, 1);
kc = KE/2; %Center of problem space
t0 = 40; %Center of the incident pulse
spread = 12; %Width of the incident pulse
NSTEPS = 1; %Number of Steps
T = 0;


while (NSTEPS > 0)
        %Enter the Steps

        prompt = "NSTEPS---> ";
        NSTEPS = input(prompt);
        disp(NSTEPS);
        n = 0;


        for n = 1:NSTEPS
        T = T+1;
        %Calculating the Ex Field
              for k = 2:KE
                  Ex(k) = Ex(k) + 0.5*(Hy(k-1) - Hy(k));
              end
        %Gaussian Pulse in the middle
              pulse = exp(-0.5*((t0-T)./spread).^2);
              Ex(kc) =pulse;
              disp(t0-T);
              disp(Ex(kc));
        %Calculate the Hy field
```

```
            for k = 1:KE-1
                Hy(k) = Hy(k) + 0.5*(Ex(k) - Ex(k+1));
            end


            figure(1)
            subplot(2,1,1)
            plot(Ex, 'LineWidth', 2)
            ylim([-1 1])
            ylabel('Ex')

            subplot(2,1,2)
            plot(Hy, 'LineWidth', 2)
            ylim([-1 1])
            ylabel('Hy')

            for k = 1:KE
                disp(Ex(k));
                disp(Hy(k));
            end



      %
        end
end
```

5.2 1.1.2

```
clear all; close all; clc;

%Aaron Rosen - Fields & Waves II
%Project 3 - An Introduction to the Finite Difference Time Domain


%Rebuilding  FD1D_1.2.c in MATLAB
KE = 200;
Ex = zeros(KE, 1);
Hy = zeros(KE, 1);

kc = KE/2;
k1 = kc - 20;
k2 = kc +20;
```

```
t0 = 40;
spread = 12;
T = 0;
NSTEPS = 1;

while (NSTEPS > 0)

    prompt = "NSTEPS ---> ";
    NSTEPS = input(prompt);
    disp(NSTEPS);

    for n = 1:NSTEPS
        T = T+1;
        %Calculate the Ex Field
        for k=2:KE
            Ex(k) = Ex(k) + 0.5*(Hy(k-1) - Hy(k));
        end

        %Guassian Pulse
        pulse = exp(-0.5*((t0-T)/spread).^2);
        Ex(k1) = Ex(k1) + pulse;
        Ex(k2) = Ex(k2) + pulse;
        disp(t0-T);
        disp(Ex(k1));
        disp(Ex(k2));

        %Calculate the Hy Field
        for k=1:KE-1
            Hy(k) = Hy(k) + 0.5*(Ex(k)-Ex(k+1));
        end

        figure(1)

        subplot(2,1,1)
        plot(Ex, 'LineWidth', 2)
        ylim([-2 2])
        ylabel('Ex')

        subplot(2,1,2)
        plot(Hy, 'LineWidth', 2)
        ylim([-2 2])
        ylabel('Hy')
```

```

        for k = 1:KE
            disp(Ex(k));
            disp(Hy(k));
        end
            % writematrix(Ex, Ex);


              %writematrix(Hy,  Hy);

        end
end
```

5.3 1.3.1

```
clear all; close all; clc;

%Aaron Rosen - Fields & Waves II
%Project 3 - An Introduction to the Finite Difference Time Domain


%Rebuilding  FD1D_1.2c in MATLAB
KE = 200;
Ex = zeros(KE, 1);
Hy = zeros(KE, 1);

kc = KE/2;
t0 = 40;
spread = 12;
T = 0;
NSTEPS = 1;

ex_low_m1 = 0;
ex_low_m2 = 0;
ex_high_m1 = 0;
ex_high_m2 = 0;

while (NSTEPS > 0)
    n = 0;
    prompt = "NSTEPS--->";
    NSTEPS = input(prompt);
    disp(NSTEPS);

```

```
    for n = 1:NSTEPS
        T = T+1;

        %Calculate the Ex Field
        for k = 2:KE
            Ex(k) = Ex(k) + 0.5*(Hy(k-1) - Hy(k));
        end

        %Put a Gaussian Pulse in the Middle
        pulse = exp(-0.5*((t0-T)/spread)^2);
        Ex(kc) = Ex(kc) + pulse;
        disp(t0-T);
        disp(Ex(kc));

        %Define Boundary Conditions
        Ex(1) = ex_low_m2;
        ex_low_m2 = ex_low_m1;
        ex_low_m1 = Ex(2);
        Ex(KE-1) = ex_high_m2;
        ex_high_m2 = ex_high_m1;
        ex_high_m1 = Ex(KE-2);


        %Calculate the Hy Field
        for k = 1:KE-1
            Hy(k) = Hy(k) + 0.5*(Ex(k) - Ex(k+1));
        end

         figure(1);
        subplot(2,1,1)
        plot(Ex, 'LineWidth', 2)
        ylim([-1 1])
        ylabel('Ex')

        subplot(2,1,2)
        plot(Hy, 'LineWidth', 2)
        ylim([-1 1])
        ylabel('Hy')

        for k=1:KE
            disp(Ex(k));
            disp(Hy(k));
```

```
        end
    end
end
```

5.4 1.4.1

```
clear all; close all; clc;

%Aaron Rosen - Fields & Waves II
%Project 3 - An Introduction to the Finite Difference Time Domain


%Rebuilding FD1D_1.3c in MATLAB
KE = 200;
Ex = zeros(KE, 1);
Hy = zeros(KE, 1);
CB = zeros(KE, 1);



kc = KE/2;
t0 = 40;
spread = 12;
T = 0;

NSTEPS = 1;

ex_low_m1 = 0;
ex_low_m2 = 0;
ex_high_m1 = 0;
ex_high_m2 = 0;

for k = 2:KE
    CB(k) = 0.5;
end

die = "Dielectric starts @ ---> ";
kstart = input(die);
disp(kstart);
eps = "Epsilon --->";
epsilon = input(eps);
disp(epsilon);

```

```
for k = kstart:KE
    CB(k) = 0.5/epsilon;
end



while (NSTEPS > 0)
    n = 0;
    prompt = "NSTEPS--->";
    NSTEPS = input(prompt);
    disp(NSTEPS);

    for n = 1:NSTEPS
        T = T+1;



        %Calculate the Ex Field
        for k = 2:KE
            Ex(k) = Ex(k) + CB(k)*(Hy(k-1) - Hy(k));
        end


            %Define Boundary Conditions
        Ex(1) = ex_low_m2;
        ex_low_m2 = ex_low_m1;
        ex_low_m1 = Ex(2);
        Ex(KE-1) = ex_high_m2;
        ex_high_m2 = ex_high_m1;
        ex_high_m1 = Ex(KE-2);



        %Put a Gaussian Pulse in the Middle
        pulse = exp(-0.5*((t0-T)/spread)^2);
        Ex(6) = Ex(6) + pulse;
        disp(t0-T);
        disp(Ex(6));
```

```



        %Calculate the Hy Field
        for k = 1:KE-1
            Hy(k) = Hy(k) + 0.5*(Ex(k) - Ex(k+1));
        end




        figure(1);
        subplot(2,1,1)
        plot(Ex, 'LineWidth', 2)
        ylim([-1 1])
        ylabel('Ex')



        subplot(2,1,2)
        plot(Hy, 'LineWidth', 2)
        ylim([-5 5])
        ylabel('Hy')



        for k=1:KE
            disp(Ex(k));

        end
    end
end
```

5.5 1.4.3

```
clear all; close all; clc;

%Aaron Rosen - Fields & Waves II
%Project 3 - An Introduction to the Finite Difference Time Domain


%Rebuilding  FD1D_1.3c in MATLAB
KE = 200;
Ex = zeros(KE, 1);
```

```
Hy = zeros(KE, 1);
CB = zeros(KE, 1);



kc = KE/2;
t0 = 40;
spread = 12;
T = 0;

NSTEPS = 1;

ex_low_m1 = 0;
ex_low_m2 = 0;
ex_high_m1 = 0;
ex_high_m2 = 0;

for k = 2:KE
    CB(k) = 0.5;
end

die = "Dielectric starts @ ---> ";
kstart  = input(die);
disp(kstart);
eps =  "Epsilon  --->";
epsilon = input(eps);
disp(epsilon);

for k = kstart:KE
    CB(k) = 0.5/epsilon;
end



while (NSTEPS > 0)
    n = 0;
    prompt = "NSTEPS--->";
    NSTEPS = input(prompt);
    disp(NSTEPS);

    for n = 1:NSTEPS
        T = T+1;
```

```



        %Calculate the Ex Field
        for k = 2:KE
            Ex(k) = Ex(k) + CB(k)*(Hy(k-1) - Hy(k));
        end




        %Put a Gaussian Pulse in the Middle
        %pulse = exp(-0.5*((t0-T)/spread)^2);
        %Ex(6) = Ex(6) + pulse;
        %disp(t0-T);
        %disp(Ex(6));

        %wave packet
        if T<20
        pulse2 = (sin((T)*2*pi) - sin((T)*1.1*2*pi));
        Ex(6) = (Ex(6) + pulse2);

        else

        Ex(6) = 0;
        Ex(k) = 0;

        end


            %Define Boundary Conditions
        Ex(1) = ex_low_m2;
        ex_low_m2 = ex_low_m1;
        ex_low_m1 = Ex(2);
        Ex(KE-1) = ex_high_m2;
        ex_high_m2 = ex_high_m1;
        ex_high_m1 = Ex(KE-2);
```

```


        %Calculate the Hy Field
        for k = 1:KE-1
            Hy(k) = Hy(k) + 0.5*(Ex(k) - Ex(k+1));
        end




        figure(1);
        subplot(2,1,1)
        plot(Ex, 'LineWidth', 2)
        ylim([-1 1])
        ylabel('Ex')



        subplot(2,1,2)
        plot(Hy, 'LineWidth', 2)
        ylim([-5 5])
        ylabel('Hy')



        for k=1:KE
            disp(Ex(k));

        end
    end
end
```