

Physics 640
October 14, 2011

Notes for reports:

Introduction is important in the report. For example, the essential ingredients of the project should be at least touched on. For example, in Project 2, the FDTD algorithm for 1D wave equation should be presented and derived.

For example, understanding of the algorithm is essential to “reverse” the direction of travel for the wave. It can be done by changing the location of the second wave initialization (we initialized the wave at two time instants).

Project 3 options (all with animations): a) and b) or a) and c)

- a) **1) Run FDTD2D for some input file**
2) Implement FDTD1D for EM waves with ABC and implement two and three media, should see reflection(s) and transmission(s) (FDTD can handle any number of media we may have). Implement a “perfectly absorbing layer”.
- b) **Use existing FDTD2D with Mur ABC but implement a correct mode for waveguide coupling. You can implement two parallel waveguides into the FDTD code, then feed in a correct mode**
- c) **Extend your FDTD1D with ABC to 2D, implement and visualize in Matlab**

1) Direct extension of FDTD 1D to 2D, surface waves, with ABC

How to derive the algorithm for 2D surface waves:

$$1D \frac{\partial^2 y}{\partial x^2} = \frac{1}{v^2} \frac{\partial^2 y}{\partial t^2} \rightarrow$$

$$\frac{\partial F^n(i)}{\partial x} \cong \frac{F^n(i + 1/2) - F^n(i - 1/2)}{\delta}$$

$$\frac{\partial F^n(i)}{\partial t} \cong \frac{F^{n+1/2}(i) - F^{n-1/2}(i)}{\delta t}$$

$$\frac{F^{n+1}(i) + F^{n-1}(i) - 2F^n(i)}{(\delta t)^2} = c^2 \frac{F^n(i+1) + F^n(i-1) - 2F^n(i)}{\delta^2}$$

$$\rightarrow F^{n+1}(i) = 2F^n(i) - F^{n-1}(i) + \left(\frac{c\delta t}{\delta}\right)^2 [F^n(i+1) + F^n(i-1) - 2F^n(i)]$$

$$2D \frac{\partial^2 y(x, z, t)}{\partial x^2} + \frac{\partial^2 y(x, z, t)}{\partial z^2} = \frac{1}{v^2} \frac{\partial^2 y(x, z, t)}{\partial t^2}$$

$$\frac{\partial F^n(i)}{\partial x} \cong \frac{F^n(i + 1/2) - F^n(i - 1/2)}{\delta} \Rightarrow \frac{\partial^2 F^n(i)}{\partial x^2} \cong \frac{F^n(i+1) - F^n(i) - F^n(i) + F^n(i-1)}{\delta^2}$$

$$\frac{\partial F^n(i)}{\partial t} \cong \frac{F^{n+1/2}(i) - F^{n-1/2}(i)}{\delta t} \Rightarrow \frac{\partial^2 F^n(i)}{\partial t^2} \cong \frac{F^{n+1}(i) - F^n(i) - F^n(i) + F^{n-1}(i)}{(\delta t)^2}$$

$$\frac{y^n(i+1, k) - 2y^n(i, k) + y^n(i-1, k)}{(\delta x)^2} + \frac{y^n(i, k+1) - 2y^n(i, k) + y^n(i, k-1)}{(\delta z)^2} = \frac{1}{v^2} \frac{y^{n+1}(i, k) - 2y^n(i, k) + y^{n-1}(i, k)}{(\delta t)^2}$$

What are the ABC we will use? Use constant flux ABC

$$f(t) \Big|_{x=(N+1)\Delta x} = f(t-T) \Big|_{x=N\Delta x}$$

How many time instants we will need to initialize? 2, 4, 6?

Please discuss this in your report

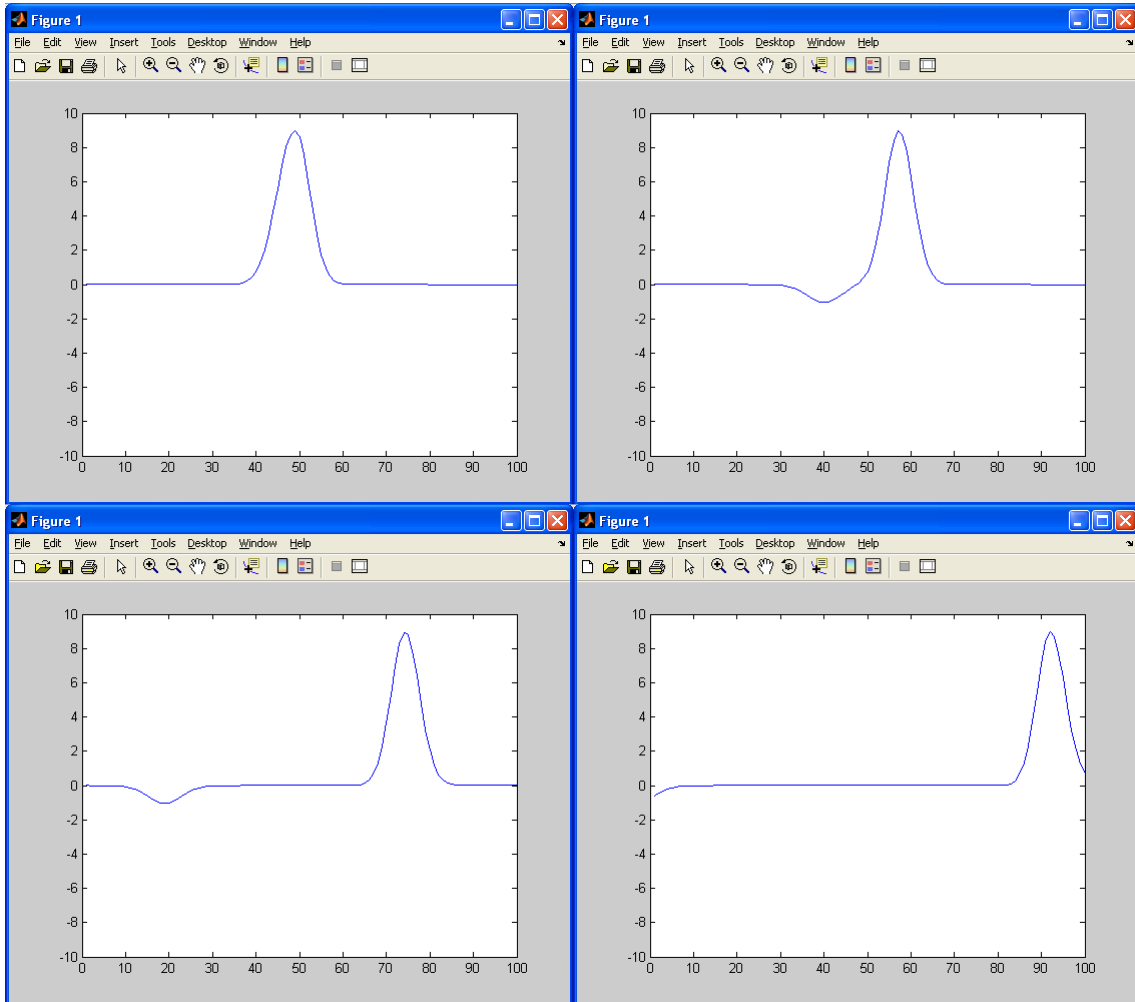
- 2) FDTD1D for EM waves with ABC and implement a least two media, should see reflection and transmission (FDTD can handle any number of media we may have)

What happens at the boundary between two different media?

The FDTD algorithm (with little extension from the version we wrote) can handle multiple media through different values for eps (relative dielectric constant) and mu (relative magnetic constant). Reflection and transmission is coded in the Maxwell's equation and our algorithm is an approx. of those equations.

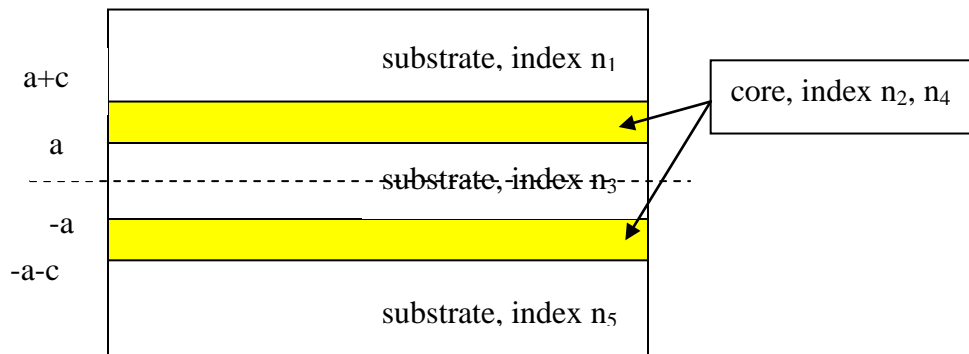
At the outer boundaries use the "constant flux ABC"

Describe different options for δt in your report; also what if we go from a higher to a lower index medium; etc.



- 3) Use existing FDTD2D with Mur ABC but implement a correct mode for waveguide coupling. You can implement two parallel waveguides into the FDTD code, then feed in a correct mode (Team 2)

What is the waveguide?



Physics requires only certain modes to exist within the two core layers (continuity of Maxwell equations at the separations). These modes can be solved using Mathematica to simplify a symbolic equation, or we can provide you with the numbers. Once you have the mode, the FDTD2D example code will propagate it properly within the core layers.

$$x > a + c \quad E_y = A_1 e^{-\gamma_1(x-a-c)}$$

$$a < x \leq a + c \quad E_y = A_2 \cos k_2 \left(x - a - \frac{c}{2} \right) + A_3 \sin k_2 \left(x - a - \frac{c}{2} \right)$$

$$-a < x < a \quad E_y = A_4 e^{-\gamma_3 x} + A_5 e^{\gamma_3 x}$$

$$-a - c \leq x \leq -a \quad E_y = A_6 \cos k_4 \left(x + a + \frac{c}{2} \right) + A_7 \sin k_4 \left(x + a + \frac{c}{2} \right)$$

$$x < -a - c \quad E_y = A_8 e^{-\gamma_5(x+a+c)}$$

$$\gamma_i = \sqrt{\beta^2 - n_i^2 k_o^2}, \quad (i = 1, 3, 5)$$

$$k_j = \sqrt{n_j^2 k_o^2 - \beta^2}; \quad (j = 2, 4)$$

How do I get the amplitudes A_i ?

From the continuity of E_y at the four boundaries: 4 equations;

Let's look at the simpler situation of $n_2=n_4$; $n_1=n_3=n_5 \rightarrow \gamma_1 = \gamma_3 = \gamma_5$; $k_2=k_4$

This Mathematica ["notebook"](#) tries to solve for these 4 equations to get the right mode to feed into the FDTD2D program. May need your modification to produce final correct results.

What is the coupling?

For example at initial time, more energy is fed into one waveguide, after sufficient time evolution, the energy goes into the other waveguide or "it is coupled into the other waveguide"

Visualization: in 2D we can do intensity plot (using color code to express intensity along the third direction), raised surface plot (using the third dimension to express the third direction)

From the continuity of the fields and its derivatives at the boundaries the following values for the propagation constant along z can be obtained: $\beta_a=7.27e6$ and $\beta_s=7.95e6$, for the antisymmetric and symmetric modes, respectively. Then the following amplitudes can be obtained (see table below), using these data:

```
freq=3e14;
speedc=3e8;
rk0=2*pi*freq/speedc;
sizeX=2.5e-8;
n1=1;
n2=1.5;
```

The width of each waveguide is 10 cells, the separation between them is 10 cells, the substrate is 35 cells each side.

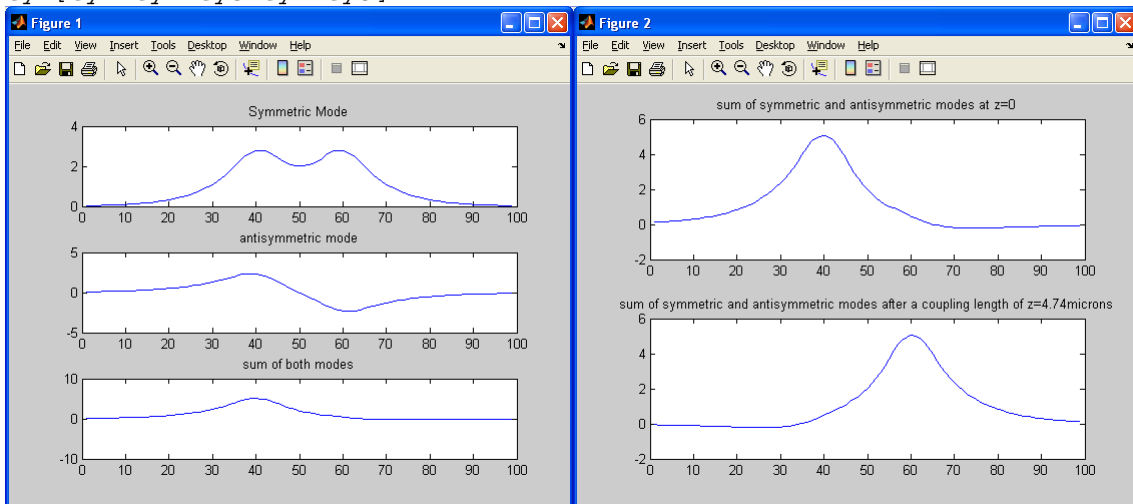
Antisymmetric $\beta_a=7.27e6$	Symmetric $\beta_s=7.95e6$
$a_1 = 1$ $a_2 = 1.1474$ $a_3 = -0.2352$ $a_4 = 0.6990$ $a_5 = -0.7103$ $a_6 = -1.1474$ $a_7 = -0.2352$ $a_8 = -1$	$a_1 = 1$ $a_2 = 1.3754$ $a_3 = 0.1846$ $a_4 = 0.5192$ $a_5 = 0.4850$ $a_6 = 1.3754$ $a_7 = -0.1846$ $a_8 = 1$

Something like this can be used to define the sum of these modes:

```

i1=1:1:35;
ey1=a1*exp(gamma5s*(i1-35).*size)+a1a*exp(gamma5a*(i1-35).*size);
i2=36:45;
ey2=a2*cos(rk2s*(i2-40).*size)+ a3*sin(rk2s*(i2-
40).*size)+a2a*cos(rk2a*(i2-40).*size)+a3a*sin(rk2a*(i2-40).*size);
i3=46:55;
ey3=a4*exp(-gamma5s*(i3-50).*size)+ a5*exp(gamma5s*(i3-
50).*size)+a4a*exp(-gamma5a*(i3-50).*size)+ a5a*exp(gamma5a*(i3-
50).*size);
i4=56:65;
ey4=a6*cos(rk2s*(i4-60).*size)+a7*sin(rk2s*(i4-
60).*size)+a6a*cos(rk2a*(i4-60).*size)+a7a*sin(rk2a*(i4-60).*size);
i5=66:99;
ey5=a8*exp(-gamma5s*(i5-65).*size)+a8a*exp(-gamma5a*(i5-65).*size);
i=[i1 i2 i3 i4 i5];
ey=[ey1 ey2 ey3 ey4 ey5];

```



After a coupling length of $z=4.74$ microns, the sum mode is coupled into the other waveguide as seen in the figure above.

Within the `fdtd2d.f`, may need to make this correction in the *function num2str* to have it compiled:

```

character*2 function num2str(i)
num2tr='XX'  -> num2str='XX'

```