Advances (and Surprises) in Electrodynamics
- Time Domain Simulations -

Thorsten Liebig

General and Theoretical Electrical Engineering (ATE)
University of Duisburg-Essen, 47048 Duisburg, Germany

26.05.2011
Outline

1. Motivation
2. Rectangular Waveguide
3. Metamaterial and Plane Waves
Outline

1. Motivation
2. Rectangular Waveguide
3. Metamaterial and Plane Waves
Visualize electromagnetic fields to get a deeper understanding of the different speeds of light.

An inspiration to create your own electromagnetic simulations (maybe by using openEMS)

Just some nice field plots to look at...
Outline

1. Motivation
2. Rectangular Waveguide
3. Metamaterial and Plane Waves
Properties of a rectangular waveguide I

1. TE- and TM-modes propagable
2. Highly dispersive wave propagation \( \beta = \beta(\omega) \)

**TE-10 Mode properties**

- Electric field in y-direction only \( E_y = A \cdot \sin\left(\frac{\pi x}{a}\right) \)
- Magnetic fields in x- and z-direction
Properties of a rectangular waveguide I

(a) XY-plane

(b) XZ-plane

Figure: Field distribution (TE-10 Mode)\(^1\)

\(^1\) pictures provided by Fedor Schreiber
Dispersion and velocities

**Figure:** Dispersion diagram for the rectangular waveguide

**Figure:** Phase- and group-velocity in a rectangular waveguide
Create a rectangular waveguide model using the openEMS-Matlab-Interface:
First step: After defining your parameter etc.
Initialize the FDTD options...

```matlab
1 % initialize the FDTD structure and set the max number of timesteps to "numTS"
2 % and the end criteria to 1e-5
3 FDTD = InitFDTD(numTS, 1e-5):

4 % excite the simulation using a gaussian puls
5 % center frequency: 0.5*(f_start+f_stop)
6 % 20dB cutoff frequency: 0.5*(f_stop-f_start)
7 FDTD = SetGaussExcite(FDTD, 0.5*(f_start+f_stop),0.5*(f_stop-f_start));

8 % define the boundary conditions for x_min, x_max, y_min, y_max, z_min, z_max
9 FDTD = SetBoundaryCond(FDTD, {'PEC','PEC','PEC','PEC','PEC','PML'});

...done!
```
Setup your FDTD mesh & model...

```matlab
10    % setup the FDTD mesh
11    CSX = InitCSX();
12    mesh.x = SmoothMeshLines([0 a], 10);
13    mesh.y = SmoothMeshLines([0 b], 10);
14    mesh.z = SmoothMeshLines([0 length], 10);
15    CSX = DefineRectGrid(CSX, unit, mesh);

16    % setup your excitation
17    CSX = AddExcitation(CSX,'excite',0,[1 1 0]);
18    weight{1} = func_Ex;  % for TE-10 this is '0'
19    weight{2} = func_Ey;  % for TE-10 this is 'sin(0.0031416*x)'
20    weight{3} = 0;
21    CSX = SetExcitationWeight(CSX,'excite', weight);
22
23    start=[mesh.x(1) mesh.y(1) mesh.z(1)];
24    stop  =[mesh.x(end) mesh.y(end) mesh.z(1)];
25    CSX = AddBox(CSX,'excite',0,start,stop);
```

...done.
Setup your dumps...

26  \% e-field dump on an xz-plane (vtk file format)
27  CSX = AddDump(CSX,'Et_xz','FileType',0,'DumpMode',1,'SubSampling','2,2,2');
28  start = [mesh.x(1) b/2 mesh.z(1)];
29  stop = [mesh.x(end) b/2 mesh.z(end)];
30  CSX = AddBox(CSX,'Et_xz',0, start, stop);
31
32  \% e-field dump line along the center-axis of the waveguide (hdf5 file format)
33  CSX = AddDump(CSX,'Et_z','FileType',1,'DumpMode',1,'SubSampling','1,1,1');
34  start = [a/2 b/2 mesh.z(1)];
35  stop = [a/2 b/2 mesh.z(end)];
36  CSX = AddBox(CSX,'Et_z',0, start, stop);

...done.
That’s it! Write file and run...

```matlab
37 WriteOpenEMS([Sim_Path '/' Sim_CSX],FDTD,CSX);
38 RunOpenEMS(Sim_Path, Sim_CSX)
```

Do your post-processing...

```matlab
39 % Read the center-axis dump and display time-domain fields in a matlab figure
40 [Et_field Et_mesh] = ReadHDF5Dump([Sim_Path '/Et_z.h5']);
41 mesh_z = Et_mesh.lines{3};
42 for n=1:numel(Et_field.TD.values)
43    Ey = squeeze(Et_field.TD.values{n}(1,1,:,:)); %reduce dimensions to Ey(z)
44    plot(mesh_z, Ey); %plot Ey over z-coordinates
45    pause(0.1);
46 end
```

to be continued ...
Figure: Paraview: Gaussian pulse in a time-domain numerical simulation
Dispersion and velocities

Another look at the dispersion and velocities:

**Figure:** Dispersion diagram for the rectangular waveguide

**Figure:** Phase- and group-velocity in a rectangular waveguide
Time domain results II

Figure: Gaussian pulse in a time-domain numerical simulation
Outline

1. Motivation
2. Rectangular Waveguide
3. Metamaterial and Plane Waves
Properties of a rectangular waveguide

Metamaterial Properties for Drude type materials

\[ \varepsilon_r = \varepsilon_{r, \infty} \left( 1 - \frac{f_{pe}^2}{f^2} \right) \quad \text{and} \quad \mu_r = \mu_{r, \infty} \left( 1 - \frac{f_{pm}^2}{f^2} \right) \]

with the plasma frequency \( f_{pe} \) and \( f_{pm} \).

- Both \( \varepsilon_r \) and \( \mu_r \) are frequency dependent and can be negative for frequencies below their respective plasma frequencies.
- In case of \( \varepsilon_r < 0 \) and \( \mu_r < 0 \) the refractive index is negative as well.
Material Dispersion

Figure: Dispersion diagram

Figure: Material constants and refractive index $n$
Phase and group velocity

Figure: Phase and group velocity
Simulate a plane wave in openEMS:

1. \% initialize the FDTD structure and set the max number of timesteps to "numTS"
2. \% and the end criteria to 1e-5
3. \texttt{FDTD} = \texttt{InitFDTD}(numTS, 1e-5):

4. \% excite the simulation using a gaussian pulse
5. \texttt{FDTD} = \texttt{SetGaussExcite}(FDTD, f0, f0/sqrt(2));

6. \% define the boundary conditions for x_min, x_max, y_min, y_max, z_min, z_max
7. \texttt{FDTD} = \texttt{SetBoundaryCond}(FDTD, \{'PMC', 'PMC', 'PEC', 'PEC', 'MUR', 'MUR'\});

...done!
Motivation
Rectangular Waveguide
Metamaterial and Plane Waves

Properties
openEMS model

Model in openEMS II

Setup your FDTD mesh & model:

8  % setup the FDTD mesh
9  CSX = InitCSX();
10  mesh.x = -width/2 : mesh_res : width/2;
11  mesh.y = -height/2 : mesh_res : height/2;
12  mesh.z = -length/2 : mesh_res : length/2;
13  CSX = DefineRectGrid(CSX, 1e-3,mesh);

14  % setup your excitation
15  CSX = AddExcitation(CSX,'excite',0,[0 1 0]);  % excite E_y
16
17  start=[-width/2 -height/2 mesh.z(3)];
18  stop =[-width/2 height/2 mesh.z(3)];
19  CSX = AddBox(CSX,'excite',0,start,stop);

20  % apply drude material
21  CSX = AddProperty(CSX,'LorentzMaterial','drude');
22  CSX = SetPropertyArgs(CSX,'LorentzMaterial','drude','Property','Epsilon',2);
23  CSX = SetPropertyArgs(CSX,'LorentzMaterial','drude','Property','Mue' ,2);
24  CSX = SetPropertyArgs(CSX,'LorentzMaterial','drude','PlasmaFrequency','Epsilon',MTM.f0);
25  CSX = SetPropertyArgs(CSX,'LorentzMaterial','drude','PlasmaFrequency','Mue' ,MTM.f0);
26  start=[mesh.x(1) mesh.y(1) -MTM.length/2];
27  stop =[-mesh.x(end) mesh.y(end) MTM.length/2];
28  CSX = AddBox(CSX,'drude',10,start,stop);
Model in openEMS III

Setup your dumps:

30 \%
31 e-field dump on an xz-plane (vtk file format)
32 CSX = AddDump(CSX,'Et_xz','FileType',0,'DumpMode',1,'SubSampling','2,2,2');
33 start = [mesh.x(1) 0 mesh.z(1)];
34 stop = [mesh.x(end) 0 mesh.z(end)];
35 CSX = AddBox(CSX,'Et_xz',0 , start,stop);

36 \%
37 e-field dump line along the center-axis (x=y=0) (hdf5 file format)
38 CSX = AddDump(CSX,'Et_z','FileType',1,'DumpMode',1,'SubSampling','1,1,1');
39 start = [0 0 mesh.z(1)];
40 stop = [0 0 mesh.z(end)];
41 CSX = AddBox(CSX,'Et_z',0 , start,stop);
That’s it! Write file and run...

```matlab
WriteOpenEMS([Sim_Path '/' Sim_CSX],FDTD,CSX);
RunOpenEMS(Sim_Path, Sim_CSX)
```

Do your post-processing...

```matlab
% Read the center-axis dump and display frequency-domain fields in a matlab figure
[Et_field Et_mesh] = ReadHDF5Dump([Sim_Path '/Et_z.h5'], 'Frequency', [f_m f0 f_p]);
mesh_z = Et_mesh.lines{3};
phase = linspace(0,360,51);
phase = phase(1:end-1);
for n=1:numel(Et_field.FD.values) % loop through all frequencies
    Ey = squeeze(Et_field.FD.values{n}(1,1,:,:)); %reduce dimensions to Ey(z)
    for p = phase %loop through phase
        plot(mesh_z, real( Ey * exp(1j*p*pi/180) )); %plot Ey over z-coodinates
        pause(0.1);
    end
end
```
Time domain results

Figure: Gaussian pulse in a time-domain numerical simulation
Material Dispersion

A closer look at the dispersion and materials constants:

Figure: Dispersion diagram

Figure: Material constants and refractive index.
The perfect lens - The wave front...

Figure: The wave front travels through the lens. Source: www.trnmag.com/Pendry-perfect-lens-diagram.gif
The perfect lens - Refocus Effect?

Figure: The perfect lens effect? Source: www.trnmag.com/Pendry-perfect-lens-diagram.gif
For further information:

www.ate.uni-due.de

www.openEMS.de

(under construction)

Thank you for your attention!