Simulation of Photo-Sensitive Devices with FDTD Method
What is FDTD method?

- **FDTD** = Finite Difference Time Domain
- **FDTD** method solves Maxwell’s equations on Yee lattice

\[
\nabla \times \vec{E} = -\frac{\partial \vec{B}}{\partial t} - 4\pi \vec{j}_m
\]

\[
\nabla \times \vec{B} = \frac{\partial \vec{E}}{\partial t} + 4\pi \vec{j}_e
\]

Maxwell’s equation

Yee lattice

Applications of FDTD method

- Photo-detectors with submicron fine structure
- LEDs and Lasers with textured surface
- Solar cells
- Photonic crystals
- Waveguide analysis
- Analysis of microwave circuits and antennas
  etc ...
Advanced Feature of FDTD Method

- Capable of tracking time evolution of field pattern.
- Fourier transformation of field can yield transmission/reflection spectrum.
- Single run of simulation can obtain wide range of response spectrum.
- Simple, robust and numerically stable.

http://ab-initio.mit.edu/wiki/index.php/Meep_Tutorial
FDTD Simulation Software
“MEEP” developed at MIT

MEEP (or MEEP) is a free finite-difference time-domain (FDTD) simulation software package developed at MIT to model electromagnetic systems, along with our MPB eigenmode package. Its features include:

- Free software under the GNU GPL.
- Simulation in 1d, 2d, 3d, and cylindrical coordinates.
- Distributed memory parallelism on any system supporting the MPI standard. Portable to any Unix-like system (GNU/Linux is fine).
- Dispersive s(a) (including loss/gain) and nonlinear (Kerr & Pockels) materials. Magnetic permeability µ and electric/magnetic conductivities α.
- PML absorbing boundaries and/or perfect conductor and/or Bloch-periodic boundary conditions.
- Exploitation of symmetries to reduce the computation size — even/odd mirror symmetries and 90°/180° rotations.
- Complete scriptability — either via a Scheme scripting front-end (as in L Cecilia and MPB), or callable as a C++ library.
- Field output in the HDF5 standard scientific data format supported by many visualization tools.

http://ab-initio.mit.edu/wiki/index.php/Meep
Interface between APSYS and MEEP

Crosslight

APSYS

FT data on APSYS FEM mesh

output

dielectric data on regular grid (e.g. 100x100x100)

Map to MEEP's grid

GNU

MEEP

.dll

Simulate propagation of gaussian pulse
Typical configuration of APSYS-FDTD simulation
Role of PML

- PML = Perfectly Matched Layer
- PML is one of the absorbing boundary conditions.
- PML absorbs electromagnetic waves. There is no reflected wave.
- Ideal for simulating open boundaries.
Material Dispersion

- Material dispersion becomes important in photo-sensitive devices, where photo-carriers are generated by absorption of light.

- In MEEP, material dispersion is expressed by a sum of harmonic oscillators.

\[\varepsilon(\omega, \mathbf{x}) = \varepsilon_\infty(\mathbf{x}) + \sum_n \frac{\sigma_n(\mathbf{x}) \cdot \omega_n^2 \Delta \varepsilon_n}{\omega_n^2 - \omega^2 - i \omega \gamma_n}\]
Example of fitting result

Fitting result of dielectric function of a-Si by two oscillator terms.
Limitations of FDTD method

- Grid spacing should be \( \sim \lambda/10 \).
- According to Courant’s stability condition, time step \( \Delta t \) becomes small when FDTD grid spacing becomes small.
- In 3-D simulation, simulation time scales like \( N^4 \), and required memory size scales like \( N^3 \).
- Application is restricted to relatively small size.
APSYS-FDTD Simulation Example 1

2-D Silicon slab

- Configuration of FDTD simulation

![Diagram of FDTD simulation setup](image)
Settings for FDTD

- **fdtd_source** statement is used to specify position, size and component of **gaussian source pulse**.

  ```plaintext
  fdtd_source component=Ex  &&
  center=(5.0 2.0 0.0) size = (10.0 0.0 0.0)
  fdtd_source component=Ey  &&
  center=(5.0 2.0 0.0) size = (10.0 0.0 0.0)
  fdtd_source component=Ez  &&
  center=(5.0 2.0 0.0) size = (10.0 0.0 0.0)
  ```
Settings for FDTD (cont.)

- **fdtd_model** statement is used to configure FDTD simulation.

```plaintext
fdtd_model
export_var = density
wavel_range = [0.55,0.65] &&
PML_thickness = 1
boundary_type = [1,0,1] &&
buffer_y = [2,2]
nb_wavel = 10
nb_mesh = [20,150,0] &&
extra_time = 0
auto_dt = 20
auto_dt2 = 5
auto_finish = yes &&
watch_point1 = [5,0.5,0]
```

Here, the duration of FDTD simulation is determined automatically by measuring magnitude of electric field.
Settings for FDTD (cont.)

- **fdtd\_dispersion** statement is used to give material dispersion for particular material.

```plaintext
fdtd\_dispersion mater=1 order=2 &&
eps0=2.33019 &&
omega1=2.83138 gamma1=0.24013 delta\_eps1=2.39085 &&
omega2=3.38094 gamma2=0.60544 delta\_eps2=7.41185
```

\[
\varepsilon(\omega, x) = \varepsilon_\infty(x) + \sum_n \frac{\sigma_n(x) \cdot \omega_n^2 \Delta \varepsilon_n}{\omega_n^2 - \omega^2 - i \omega \gamma_n}
\]
Propagation of Gaussian pulse

Click the picture to animate
How long should we run FDTD?

- Crosslight implemented “Watch and compare” method to judge if wave is fully decayed.

Watch point can be placed at any location.
How long should we run FDTD? (cont.)

- Magnitude of electric field at each watch point is averaged over certain time duration.
- Decay of electric field is measured by comparing current averaged value and initial averaged value of electric field magnitude.

$\Delta t_1$ should be long enough so as to catch propagating wave at every watch points.
Relative energy density

1D slice

Optical energy decreases exponentially along incident path.
Optical generation rate (log scale)
How could we obtain spectrum data of relative energy density out of FDTD?

1st step: Wave propagation in empty space
- Input pulse
- Empty FDTD simulation
- FFT to get $E_0(x,y,z,\omega)$

2nd step: Wave propagation with device
- Input pulse
- FDTD simulation
- FFT to get $E(x,y,z,\omega)$

$$|E(x,y,z,\omega)|^2$$
$$|E_0(x,y,z,\omega)|^2$$
$$\sim D(x,y,z,\omega)$$
APSYS-FDTD Simulation Example 2

2-D Silicon slab half-covered by Al layer

Configuration of **FDTD** simulation
Propagation of Gaussian pulse

Click the picture to animate
Relative energy density
APSYS-FDTD Simulation Example 3

Focusing effect in 3-D Photodetector with lens shaped window
Optical Generation Rate (log scale)

Focusing effect can be observed clearly
Modeling 3D Texture

Complicated 3D geometry can be defined by connecting multiple 2D slices.

3D texture data generated by CAD or 3DCG software may be imported by APSYS-FDTD package in the future.
Pyramidal texture for Si solar cell
Relative Energy Density

Pyramidal texture enhances transmission of incident light.

3D View

2D Slice at center

1D Slice at center
Optical Generation Rate (log scale)

3D View

2D Slice at Center
Comparison of Flat and Textured surface

Textured surface yields high intensity light distribution around center of device.
System Requirement

- 2GB memory recommended for 3D simulation.
- **64-bit** CPU and OS is recommended if user deals with big structure.
- **FDTD** simulation time varies from few minutes to few hours depending on problem. Obviously, faster CPU is better.
Summary

- **APSYS** and **FDTD(MEEP)** now work together seamlessly.

- 2D and 3D structures were used to demonstrate capability of **APSYS-FDTD** package.

- 64-bit CPU and OS may be necessary to carry out complicated 3D structure. **APSYS** is 64-bit ready software.