Finite Difference Time Domain (FDTD) Simulator

Informative session

Saurabh Sant, Dr. sc. ETH
saurabh64sant@gmail.com

SemiVi LLC
Zollikon, Switzerland.

May 10, 2025

Overview

Table of Contents

Salient features

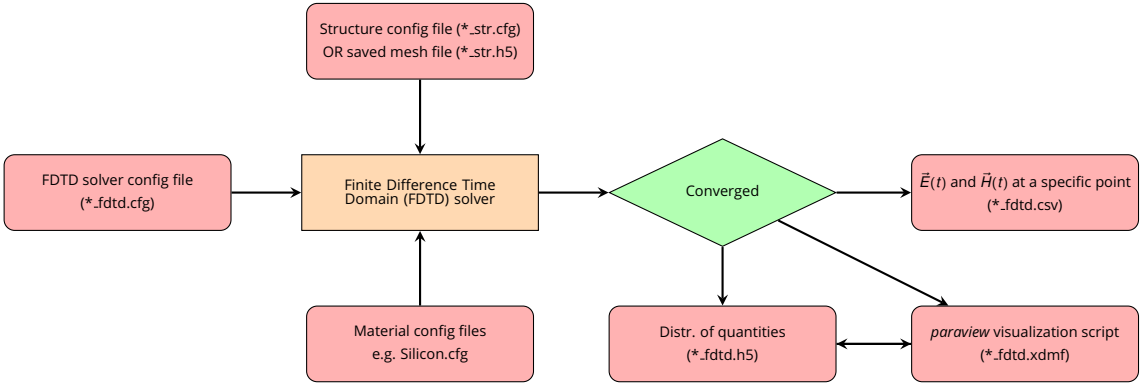
The FDTD solver takes the config file with the required solver inputs and calculates time evolution of electromagnetic waves using finite-difference-time-domain method.

Salient features –

- Materials with wavelength dependent real and imaginary permittivity,
- Dispersive material models – Drude, Debye, Lorentz, and Kerr models.
- Use plane-wave source with uniform beam, Gaussian beam, or mode-beam, also dipole source.
- Apply BCs – reflective, periodic (+ oblique incidence), PML
- *built-in hardware acceleration enabled*.

Table of Contents

FDTD solver interface and config files



To run a FDTD solver simulation...

- Create and save *FDTD solver config file*, *structure config or mesh file*, *material files* (if modified) in the same folder.
- Run simulations using the following command –
  `>> fdtdsolver fdtdsolver SiWG_fdtd.cfg`
- To use *hardware acceleration*, use the following command –
  `>> fdtdsolver gpufdtdsolver SiWG_fdtd.cfg`

Easy-to-use Config file

```
File: {
  Device = "fdtdSiWG_str.cfg";
  Out = "SiWG";
}

Solver: {
  MaximumTime = 15E-14;
  GridType = "TM";
  Settings = ["WavelengthDepIndex"];
}

Source*left: {
  Type = "PlaneWave";
  BeamShape = "ModeBeam";
  Position: ([-0.6, -0.2, 0.], [-0.6, 0.2, 0.]);
  Theta = 90;  Phi = 0; Wavelength = 0.3;
  EffectiveIndex = 3.5; NRise = 5;
  Intensity = 1000; Delay = 1;
  Flags = ["UsePowerMethodModeSolver"];
}

Boundary*xbdr: {
  Axis = ["Xmin", "Xmax"];
  Model = "CPML"; PMLLayers = 20.;
}

Boundary*ybdr: {
  Axis = [ "Ymin", "Ymax", "Zmin", "Zmax"];
  Model = "CPML"; PMLLayers = 20.;
}
```

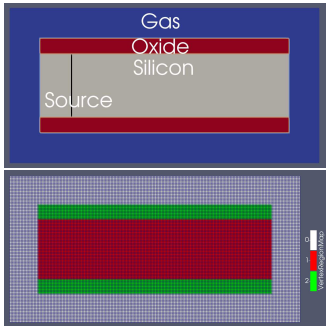


Figure: Structure of the simulated waveguide

- Include a new device structure.
- Define the source window.
- Apply specific boundary conditions.
- Set solver parameters.

Modeling dispersive materials

```
Dispersive*Si: {
  Region = "RegSi";
  Model = "Drude";
  NumberPoles = 3;
  PoleFreq = [2.31E15, 2.18E15, 2.08E15];
  DampFact = [5.5E12, 4.353E12, 10.88E12];
}
```

- `Dispersive`: Defines a dispersive model for `RegSi` region.
- `Model`: Specifies model which has `NumberPoles`.
- `PoleFreq` and `DampFact`: Parameters corresponding each pole are set as lists.
- Dispersive materials are modeled by "Auxiliary Differential Equation" method.

$$\chi(\omega) = \frac{\omega_p^2}{j\omega \cdot (j\omega + g)} \tag{1}$$

$$\vec{P}(\omega) = \epsilon_0 \chi(\omega) \cdot \vec{E}(\omega) \tag{2}$$

$$\vec{\Phi} = j\omega \vec{P}(\omega) = \frac{\epsilon_0 \omega_p^2}{(j\omega + g)} \vec{E}(\omega) \tag{3}$$

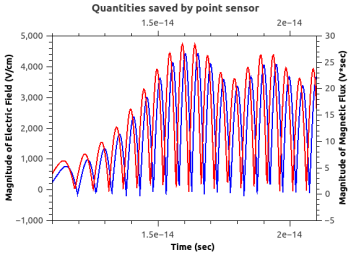
Substituting $j\omega \rightarrow \frac{\partial}{\partial t}$,

$$g\vec{\Phi} + \frac{\partial \vec{\Phi}}{\partial t} = \epsilon_0 \omega_p^2 \vec{E} \tag{4}$$

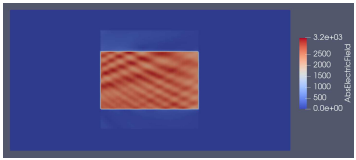
Results Visualization

```
Movie*XYEField: {
  Size = "640x480"; Plane = "XY";
  Intercept = 0; Quantities = ["ElectricFieldZ"];
}

PointSensor*X0pt: {
  Position = [0.05, 0.025, 0.];
  Quantities = ["AbsElectricField", "AbsMagneticFlux"];
}

TimeAverage*AvgMid: {
  TimeBegin = 2E-15; TimeEnd = 4E-15;
  Quantities = ["AbsElectricField", "AbsMagneticFlux"];
  Position = ([-0.35, -0.35, 0.], [0.35, 0.35, 0.]);
  TimeSteps = 1;
}
```

- `Pointsensor`: Saves temporal data at given point in a csv file.
- `Movie`: Saves temporal data at the cross-section as a movie.
- `TimeAverage`:Stores temporal average of the data.
- Saves an *xdmf* script for visualization in *paraview*.



(a) Data of Point Sensor



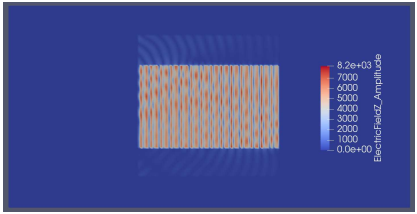
(b) Time average of $|\vec{E}|$

Figure: Magnitude and Phase of E_z

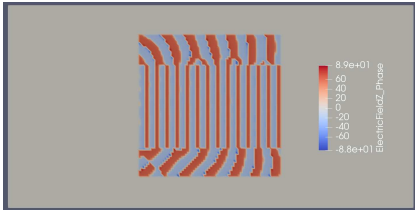
Results Visualization

```
PhaseCalculator*Ph: {
  Time = 2E-15;
  Quantities = ["ElectricFieldZ", "MagneticFluxZ"];
  Position = ([-0.35, -0.35, 0.], [0.35, 0.35, 0.]);
}

Detector*det: {
  StepX = 5;  StepY = 5; StepZ = 5;

  Tolerance = 1E0; StartTime = 2E-15;
  Position = ([-0.35, -0.35, -0.26], [0.35, 0.35, 0.26]);
  Quantities = ["ElectricFieldZ"];
}
```

- `PhaseCalculator`: Calculates magnitude and phase of the data at a time-point. Also, saves an *xdmf* script for visualization in *paraview*.
- `Detector`: Checks if the given quantity has stabilized over time. When yes, stops the simulation.



(a) Magnitude of Electric Field



(b) Phase of Electric Field

Figure: Magnitude and Phase of E_z

Conclusions

- Finite-difference time domain (FDTD) solver supports,
  - Materials with wavelength dependent real and imaginary permittivity,
  - Dispersive material models – Drude, Debye, Lorentz, and Kerr models.
  - Use plane-wave source with uniform beam, Gaussian beam, or mode-beam, also dipole source.
  - Apply BCs – reflective, periodic (+ oblique incidence), PML
- Hardware acceleration enabled in FDTD calculations.
- Open to collaborations with our customers / partners.

Table of Contents

Easy-to-install licenses

For ordering a license, along with the name and the organization details, please also provide –

- For the node-locked licenses: Ethernet mac address of the client machine on which the software will run.
  OR
- For the server licenses: Ethernet mac address of the server machine at the client organization.

If you purchased one or more node-locked licenses, you will receive the following license file by secured email.

1. ActiveLicense_NL_<id>_<Info>.lic, where <id> stands for license id and <Info> stands for customer identification in short.

Copy the license file to `/var/local/oesoft/licenses/` on the machine whose mac-address has been provided and change its access rights to `660`.

If you purchased one or more server licenses, you will receive the following license files by secured email.

1. ActiveLicense_SE_<id>_<Info>.lic
2. ServerLicense_<id>_<Info>.lic
3. ClientLicense_<id>_<Info>.lic

Copy the first two license files to `/usr/share/oesoft/licenses/` on the *server machine* whose mac-address has been provided. Copy the ClientLicense*.lic file to *all the* client machines at the same location. You don't need to change the access rights.

The End

Questions? Comments?