FastMaxwell

Introduction

FastMaxwell is an open-source, full-wave, 3D, electromagnetic solver for efficient electrical extraction of interconnect structures above substrate. FastMaxwell is based suited for the extraction of manhatten-type electrical structures described by large aspect ratio's as is typically the case with interconnects, power grids, inductors and transmission lines. The code is accelerated via the dyadic substrate Green's function, efficient discrete complex image method, efficient full-wave integration routines and an iterative precorrected fast Fourier transform matrix vector product. The code has been tested on a variety of large scale (up to 300,000 unknowns) structures.

External Packages

Within FastMaxwell we use a variety of packages:

1. Pfft++: Adopted to suit our implementation, this package is embedded within our implementation and does not require external download or special installation procedure.
2. SuperLU 3.0: for solving large sparse linear systems in order to compute the preconditioner (requires separate installation steps)
3. itpp: (requires separate installation steps)
4. fftw: (requires separate installation steps)
5. BLAS: (requires separate installation steps)
6. LAPACK: (requires separate installation steps)

Installation

The complete installation instructions are included in the README file. This is a general summary of the steps

1. install fftw v2.1.5 (http://www.fftw.org/)
2. install blas
3. install LAPACK
4. install itpp
5. install SuperLU: we strongly encourage using the same BLAS library while installing itpp and SuperLU. Otherwise we have observed possible compilation errors
6. unzip and untar FastMaxwell: there are 4 different directories
    1. src: includes the main extractor source files and the iterative solver
    2. pfft: includes the precorrected-fft related files
    3. examples: includes some example input files
    4. bin: includes the executable under the appropriate directory
    5. util: includes the config.guess, share.cpp.makefile and share.cc.makefile.
7. In the directory FastMaxwellv2.0 edit makefile, def.makefile and util/share.cpp/makefile and util/share.cc.makefile to point the correct location of the libraries and include files of the external packages
8. type make

Usage

Before giving the details of writing the input file, we just make the general remark that FastMaxwell's modeling ideology is based on the partial element equivalent circuit or similarly the same spirit adopted in FastHenry. Structures best modeled using FastMaxwell are Manhatten-like geometries described by large aspect ratios. In such structures the current can be safely assumed 1-directional and the direction of the current flow is indicated as the length direction the perpendicular plane is the cross sectional plane. Initial preparation should include:

1. A set of axis should be imposed on the structure, such that the substrate occupies the complete lower half space characterized by (z<0). The x- and y- axis should be chosen as to coincide with the interconnect structure alignment.
2. The structure should be first discretized into a set of segments (also referred to as filaments), we recommend the filament size to be smaller than $\frac{\lambda}{20}$.
3. The cross of every segment is then discretized into smaller filaments. It is important to make sure that the filaments near the corners of the segment are small (typical cross section dimensions are skin_depth/3). For efficient simulation the filaments toward the center of the conductor can have larger dimensions.

Input File Structure

The input file is composed of a variety of commands detailed in this section:

Default: Specifies important "common" information such as:

```
.defaults w=10.000000e-06 h=4.000000e-06
+ sigma=5.880000e+07
+ fnw=4 fnh=4 frw=1.2 frh=1.2 nl=8
+ layer_epir=4.2 layer_sigma=7.41
+ pnw=1 pnh=1 pfracw=0.10000 pfrach=0.10000 pfracl=0.10000 nep
```

Parameter	Description
nl	number of discretizations along the length of the segment
w	Width of the conductor
h	Conductor thickness
fnw	Number of filaments per width
fnh	Number of filaments per thickness
frw, (frh)	Ratio (>1) of the widths (thicknesses) of successive filaments. Filaments near the corner are smaller
layer_epir	Dielectric constant
layer_sigma	Conductivity
pnw, (pnh)	number of panels per filament, if 1 is specified then number is chosen according to default accuracy settings
pfracw, (pfrach, pfracl)	Ratio (>1) of the widths (thicknesses, length) of successive panels. Panels near the corner are smaller Note that width is associated with panels discretizing the top and bottom sides of the conductor and thickness is associated with panels discretizing the sides of the conductor.

Nodes: used to identify the beginning and the end of conductor segments

```
Nx      x=      y=      z=
```

Segments: The set of nodes are connected together by segments (current is assumed to flow in the direction of the segments)

```
Ex      Nx      Ny      nl=      w=      h=
```

nl is the number of discretizations along the length of the segment, w (h) is the width (thickness) of the segments and overrides the default value. It is important to understand that the length of the segment is determined by the direction connecting the two nodes and that the width and thickness are determined by the right hand rule.

Equiv: Used in case of short filaments or small connections such as vias to identify an electrical short circuit between the nodes without physical support and thereby significantly simplifies the simulation without significantly affecting the accuracy of the simulation.

```
.equiv  Nx      Ny
```

L: Used to define the substrate layer

```
L layer_epir=11.70000 layer_sigma=7.41 starth=0.00000
```

where layer_epir is the dielectric constant and layer_sigma is the conductivity of the lossy substrate and starth is the z-coordinate of the substrate. It is important to note that the substrate is always considered to be infinite in the x-y direction and to extend in the lower half-space.

External: Defines external excitation port

```
.external       Nx      Ny
```

Resistance: The code permits connecting lumped elements between defined nodes. This can be used to model a load, the input impedance of the source or short resistive vias.

```
Rdriver         Nx      Ny      val
```

Command Summary

Command	Explanation
.units	m
.default	
Nx x= y= z=	defines a node at position (x,y,z)
Ex Nx Ny	
.equiv	virtual short circuit
.external Nx Ny	external excitation
Rx Nx Ny val	resistance between nodes Nx and Ny
Layer	defines substrate
*	used to indicate a comment
.end	terminates the input file

Input File Example

An example discretization of the three turn inductor shown in figure 1 is demonstrated below:

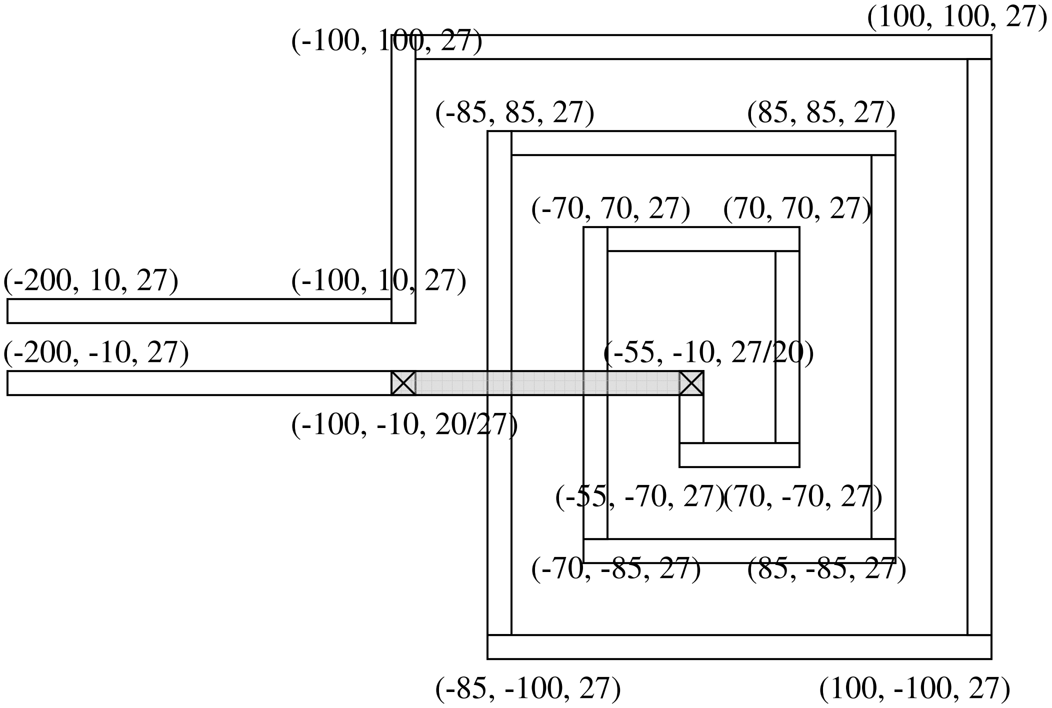

Figure.1 Inductor top view. Dimensions are in um. (-50, -10, 27/20) indicates two nodes with the same x and y coordinates but at two different z levels.

IndS_2.5_200_10_5.inp

```
*indn_fine: first line is a comment

***********************
* default parameters
***********************
.units m
.defaults w=10.000000e-06 h=4.000000e-06
+ sigma=5.880000e+07
+ fnw=4 fnh=4 frw=1.2 frh=1.2 nl=8
+ layer_epir=4.2 layer_sigma=7.41
+ pnw=1 pnh=1 pfracw=0.10000 pfrach=0.10000 pfracl=0.10000 nep

***********************
* Nodes
***********************
```

```
N0 x=-200.0000e-06 y=10.000000e-06 z=27.000e-06

N1 x=-100.0000e-06 y=10.000000e-06 z=27.000e-06
N2 x=-100.0000e-06 y=+100.000000e-06 z=27.000e-06
N3 x=+100.0000e-06 y=+100.000000e-06 z=27.000e-06
N4 x=+100.0000e-06 y=-100.000000e-06 z=27.000e-06

N5 x=-85.0000e-06 y=-100.000000e-06 z=27.000e-06
N6 x=-85.0000e-06 y=+85.000000e-06 z=27.000e-06
N7 x=+85.0000e-06 y=+85.000000e-06 z=27.000e-06
N8 x=+85.0000e-06 y=-85.000000e-06 z=27.000e-06

N9 x=-70.0000e-06 y=-85.000000e-06 z=27.000e-06
N10 x=-70.0000e-06 y=+70.000000e-06 z=27.000e-06
N11 x=+70.0000e-06 y=+70.000000e-06 z=27.000e-06

N12 x=+70.0000e-06 y=-70.000000e-06 z=27.000e-06
N13 x=-55.0000e-06 y=-70.000000e-06 z=27.000e-06
N14 x=-55.0000e-06 y=-10.000000e-06 z=27.000e-06
N15 x=-55.0000e-06 y=-10.000000e-06 z=20.000e-06
N16 x=-100.0000e-06 y=-10.000000e-06 z=20.000e-06
N17 x=-100.0000e-06 y=-10.000000e-06 z=27.000e-06
N18 x=-200.0000e-06 y=-10.000000e-06 z=27.000e-06


***********************
* Segments
***********************

E0 N0 N1
E1 N1 N2
E2 N2 N3
E3 N3 N4
E4 N4 N5
E5 N5 N6
E6 N6 N7
E7 N7 N8
E8 N8 N9
E9 N9 N10
E10 N10 N11
E11 N11 N12
E12 N12 N13
E13 N13 N14 nl=4
E14 N14 N15 nl=2
E15 N15 N16 nl=4
E16 N16 N17 nl=2
```

```
E17 N17 N18

*.equiv N12 N13
*.equiv N14 N15


**************
* substrate info
**************
L layer_epir=11.70000 layer_sigma=7.41 starth=0.00000

**************
* driver port
**************
.external N0 N16
*Rdriver   N1 N4 1.000000e-09
.end
```

Running Command

```
./FastMaxwell [options]
```

[options] =-iben
-i: input file name
-b: begin frequency
-e: end frequency
-n: number of frequency points

Example:

```
./FastMaxwell –i ind.inp –b 1e6 –e 1e11 –n 100
```

This command will simulate the structure in the input file ind.inp in the frequency band from 1e6 to 1e11 with 100 frequency points distributed logarithmically.

Output file format

The main output file is named output.mat. This file is composed of three different columns: the first is the frequency, the second is the real part of the input impedance and the third is the imaginary part of the input impedance.

MATLAB Interface: The output file can be imported to MATLAB via:

```
load –ascii output.mat
```