

MATLAB Partial Element Equivalent Circuit Toolbox for Solving Coupled Electromagnetic-Circuit Problems

Vjosa Shatri^{*a}, Ruzhdi Sefa^b, and Lavdim Kurtaj^c

^{a,b} Department of Fundamental Engineering Subjects, Faculty of Electrical and Computer Engineering, University of Prishtina, Kosovo

^c Department of Automation, Faculty of Electrical and Computer Engineering, University of Prishtina, Kosovo

Accepted 10 September 2013, Available online 01 October 2013, Vol.3, No.4 (October 2013)

Abstract

Performance of Partial Element Equivalent Circuit (PEEC) Toolbox for MATLAB for solving coupled electromagnetic-circuit problems is presented. Geometry of the problems to be solved is described with wire-based language, specifically designed for this toolbox, by using predefined basic elements. Meshing function will discretize geometry, or parts of it, with selected 1D, 2D, 2DS, 3D, or 3Dxyz mesh type, and will generate surface and volume cells as required by PEEC method. Choice of mesh type is trade-off between model size and accuracy. Calculation of partial elements is handled by FastCap2 and FastHenry2. Toolbox offers interface to these applications starting from discretized cells. As known to us, this is first general interface to FastCap2 and FastHenry2, and it can be used also for problems not related to PEEC. Netlist for any standard PEEC circuit model, in retarded or quasi-static version, can be created in format compatible with LTSpice or Multisim circuit solver, where it can be combined with other lumped, linear or nonlinear, passive or active, components for solving combined electromagnetic-circuit problems, with netlist representing electromagnetic part. Naming convention is related to geometry construction elements and enables easy access to any internal point of model. Implementation of delayed quantities is done with LAPLACE controlled sources. Guidelines for proper selection of parameters for LAPLACE sources are given, that result in significant speedup without sacrificing accuracy. Performance of this integrating solution for PEEC solver in frequency- and time-domain is tested and results are in good agreement with results reported elsewhere. Toolbox can serve as platform for testing new approaches for any sub-step of PEEC method, while using others unchanged.

Keywords: Coupled Electromagnetic-Circuit Problems, Geometry Description, Partial Element Calculation, PEEC Toolbox, Wire-Based Language, FFT Window Length, Parallel-Plate Transmission Line.

1. Introduction

Simulating and analyzing electronic circuit operation with Spice-like circuit solver tools is common to electric and electronic engineering community. SPICE (Simulation Programs with Integrated Circuit Emphasis) was public domain from its beginning when it was announced to the world in 1973 (Nagel, L. W. *et al* 1973). Versions of it are core for many circuit simulation programs used in academia for teaching and research, and in industry. Main request from SPICE, especially for industry, is to predict circuit behavior as close as possible to built real circuit, be it in integrated circuit form or board level circuits. Intentions are to accurate predictions up to virtual prototyping level, so real prototypes can be completely bypassed. With ever increasing operation frequency of the new devices to gigahertz range, this intention is harder to fulfill because of electric and magnetic couplings within and between (Daroui, D. *et al* 2012; Rong, A. *et al* 2002),

lumped devices. Even breadboard prototyping, as common method for testing circuit designs, may not be accurate in results when compared to final product with PCB (Printed Circuit Board). At high frequencies using wired device models will not suffice if even wirings itself are not modeled properly. These interactions take us to the field of electromagnetic modeling. To continue using circuit solvers for electromagnetic and combined electromagnetic-circuit problems, Dr. Ruehli developed a method for modeling arbitrary shaped three dimensional (3D) electromagnetic problem as combination of basic circuit building blocks of similar structure, the PEEC (Partial Element Equivalent Circuit) method (Ruehli, A. E. 1974; Ruehli, A. E. *et al* 1992). Blocks are composed from basic circuit elements: resistors, inductances, capacitances, and controlled voltage or current sources. Elements of building block are called "partial" because they represent part of 3D geometry (cell), including its interaction with other building blocks. Cells are created by appropriate discretization of 3D geometry (meshing). Final problem model will be in circuit form and some member of SPICE

*Corresponding author: Vjosa Shatri

family may be used to solve it in time- or frequency-domain. LTspice is freeware SPICE simulator from Linear Technology (LTC) (Linear Technology) has no limitation on node number and can be used to solve model circuit with partial elements.

Calculating capacitive and inductive interactions is classical electromagnetic problem, and serves as basis for partial element calculation. Calculation of partial elements is done by solving Maxwell equations over resulting group of cells. Preferred solution is analytical one (Ekman, J. 2003; Paul, C. R. 2010), but they exist only for specific cells (orthogonal) and configurations between them (Parallel edges and faces). Otherwise numerical integrations are used (Müsing, A. et al 2009). Two software applications developed at MIT (Massachusetts Institute of Technology), FastCap2 (Nabors, K. et al 1999) and FastHenry2 (Kamon, M. et al 1994), solve problems of these type and are used as reference for many other solutions. Using these applications to calculate partial elements requires text-based input files in specific format. There is no general software for crating these files from geometry of the problem, with discretization as required by PEEC method, were there is specific relation (half-cell shift) between surface (capacitive) cells and volume (inductive-resistive) cells.

To solve problems by PEEC method and to create environment for testing different approaches and algorithms for specific aspects of method, PEEC Toolbox for MATLAB was created (Shatri, V. et al 2012). Toolbox supports wire-based language for geometry description designed for this purpose. It integrates FastCap2 and FastHenry2 to calculate partial elements, and LTspice or Multisim (from National Instruments) (Báez-López, D. et al 2011) for circuit simulation. Performance of this approach for solving coupled electromagnetic-circuit problems in time- and frequency-domain is tested on selected problems.

Next section presents PEEC method. Toolbox overview and general flow-diagram with functions and intermediate data is given in Section III. Other two sections cover geometry description with wire-based language and meshing types that can be used. Calculating partial elements and generation of netlist for selected model type is described in Section VI and VII. How to integrate PEEC model for electromagnetic part to lumped part of circuit for creating coupled problem for simulation is covered in Section VIII. In Section IX simulation examples are analyzed and discussed. Finally, Conclusions on performance and possibilities of the Toolbox are given in Section X.

2. PEEC Method

Partial Element Equivalent Circuit method is generalization of common equivalent circuit modeling by defining systematic procedure for getting model of problem to be solved. Model is in form of an electric circuit and SPICE like solvers can be used to analyze it in time or frequency domain. It is synthesized from building blocks of similar structure. Building block is an equivalent

circuit for single cell from discretized (meshed) geometry of the problem. Cells may be rectangular, as in initial model (Heeb, H. et al 1992), or nonorthogonal, but in both cases equivalent circuit of cell retains its structure (Ruehli, A. E. et al 2003; Nitsch, J. et al 2009). PEEC method was developed for solving electromagnetic problems and today is one of standard members of numerical modeling family for this domain.

First step toward PEEC model is geometry description for selected problem. Grid of nodes will be adopted through geometry, generally in three dimensions, but in two or one dimension for simplified cases and for lower order models. Two types of cells are defined over adopted grid nodes, hexahedral volume cells and quadrilateral surface cells. Surface cells are shifted for half length of volume cell in each direction. Assumption for rectangular case is that current density J is constant over volume cell cross-section with length being distance between two nodes for selected current flow direction. Second assumption is that charge density ρ over surface cell is constant. Fig. 1 shows part of conductor with one-dimensional (1D) meshing. Cells are created only in length direction (z -axis). Shifting of surface cells is easily noticed. Two-dimensional (2D) meshing for wider conductors is pictured in Fig.2. Volume cells are defined for two current flow directions, in length direction (z -axis) and in width direction (x -axis). Cells on edges of meshed body will be half in size compared to inner cells (Gope, D. et al 2007). When meshing in three-dimensions (3D), similar 2D meshing is applied in width-height plane (xy plane) also.

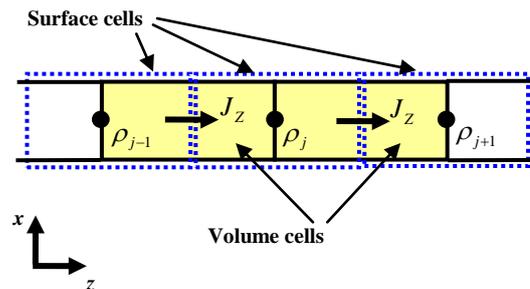


Fig. 1 1D meshing volume cells (solid) and surface cells (dashed) for part of thin conductor.

PEEC building blocks will be formed by circuit interpretation of electric field integral equation applied on discretized cells, after approximation of unknown current and charge densities with weighted sum (Miscione, G. et al 2006) and the Galarkin solution (Polycarpou, A. C. 2006; Sadiku, M. N. O. 2001; Gibson, W. C. 2008).

$$\varphi_i - \varphi_j = -U_k^{inc} + R_k I_k + j\omega L_{kk} + j\omega \sum_{\substack{n=1 \\ n \neq k}}^{N_V} L_{kn} I_n, \quad (1)$$

and

$$I_i^c = \frac{j\omega}{P_{ii}} \varphi_i - \sum_{\substack{m=1 \\ m \neq i}}^{N_S} \frac{P_{im}}{P_{ii}} I_m^c, \quad (2)$$

where, L_{kk} is partial self-inductance, L_{kn} are partial mutual-inductances, R_k is resistance between nodes of volume

cell, p_{im} are coefficients of potential with $1/p_{ii}$ defined as pseudo-capacitance of node i .

Shifted surface cells are used for calculating derivative approximation. Resulting equations are given above.

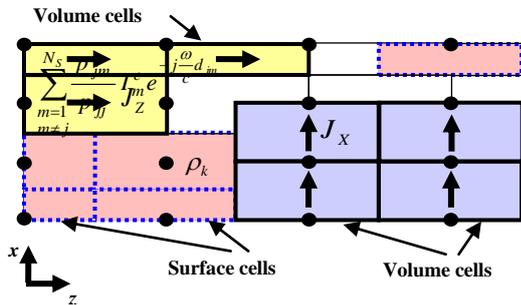


Fig. 2 2D meshing volume cells (solid) and surface cells (dashed) for part of wide-thin conductor. Volume cells for two current directions are shown.

Referent point for node potentials φ_i is infinity. U_k^{inc} represents voltage caused by incident field \mathbf{E}^{inc} in cell i . Cell current is I_k , and current on pseudo-capacitance branch is I_{ic} . Corresponding equivalent circuit for single cell (Nitsch, J. et al 2009) is shown in Fig. 3. Volume cells are called inductive-resistive partitions and surface cells are called capacitive partitions (Ruehli, A. E. 1974). Resistor models ohmic losses, inductor represents self-inductance of volume cell, and current controlled voltage source (CCVS) represents inductive couplings between N_v volume cells. Current controlled current sources (CCCS) represent electric field couplings between N_s surface cells. In retarded PEEC model control currents I_n and I_m^c will be delayed by traveling time between cell centers d_{ij} , as in Fig. 3. If largest size is smaller than $\lambda_{min}/20$, where λ_{min} is smallest wavelength for frequencies of interest, problem can be treated as quasi-static and retardation will not be considered, as in (1) and (2). Since elements of this circuit correspond to discrete parts of the structure they are called partial elements.

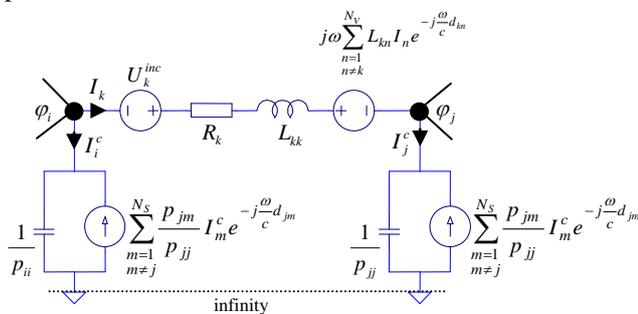


Fig. 3 Retarded model equivalent circuit for single PEEC cell

To complete equivalent circuit for selected problem, values for circuit partial elements must be evaluated. It is computationally intensive process, since calculation of partial inductances requires double volume integration, whereas calculation of coefficients of potential requires double surface integration. Closed-form solutions are

desirable, but they are available only for rectangular cells (Paul, C. R. 2010; Ekman, J. 2003). Numerical integration must be used for calculating them over general geometries. To decrease integration order combined techniques are developed by using closed-form solutions for filaments and integrating over other dimensions (Müsing, A. et al 2009).

System equations will usually be in Modified Nodal Analysis (MNA) formulation, as in SPICE-like solvers (Vladimirescu, A. 1994). Unknown quantities in equivalent circuit model for electromagnetic problems will be branch currents and node potentials with reference to infinity, and solutions may be in time- or frequency-domain.

If field variables are needed they can be calculated by post-processing volume cell currents and node potentials.

3. PEEC Toolbox Overview

Collection of functions for handling PEEC modeling and simulation are written in MATLAB, and are part of PEEC Toolbox (Shatri, V. et al 2012). It covers steps from geometry description to results of simulation. It may serve as platform for testing different implementations for each individual step. Flow-diagram with steps (Ekman, J. 2003; Daroui, D. et al 2010; Yutthagowith, P. 2011; Daroui, D. et al 2012), corresponding functions and intermediate results (structures or files) is given in Fig. 4. Initial version relies on FastCap2 and FastHenry2 (FastFeldSolvers) for calculating partial elements, and LTspice (Linear Technology) or Multisim (Báez-López, D. et al 2011) for circuit simulation.

Toolbox will accept geometry description in text format that follows wire-based language syntax (Shatri, V. et al 2012). Different meshing approaches and resolutions can be selected as tradeoff for accuracy and problem complexity. By calling geometry or mesh viewing functions one can see defined geometry and meshed cells. Meshing is done according to PEEC requirements, i.e. surface cells and volume cells with half-volume cell shift of surface cells. Mesh data are used to generate input files for calling FastCap2 and FastHenry2. Partial elements calculated by them are taken back to MATLAB in matrix form: C_FC, L_FH, and R_FH. Model of the problem in circuit form is formed from partial elements, and generated netlist file will be used in circuit solver along with other lumped circuit elements. Delays for retarded model case are calculated from geometry data and will be added to same netlist file. All couplings between partial elements were implemented with CCCS (Nitsch, J. et al 2009). Netlist format can be compatible with LTspice or Multisim simulation software. Results of simulation in time or frequency domain as circuit quantities are obtained directly.

If field quantities are needed, or data are to be related to part of input geometry data, simulation data can be sent back to MATLAB for post-processing.

New algorithms for any step can be tested by providing appropriate function for that step, while using unchanged functions for other steps. Since only selected step will be

different, it will be suitable for judging relative performance of new approaches.

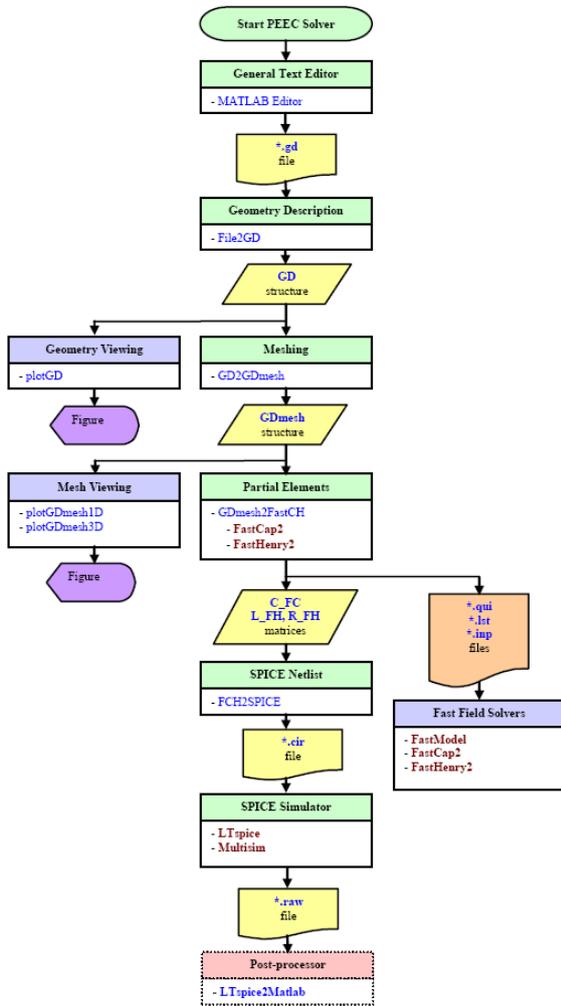


Fig.4 PEEC Toolbox flow-diagram, functions and intermediate data

4. Composing Problem Geometry

First step for analyzing electromagnetic-circuit problems with PEEC method is to describe geometry of the problem. Toolbox supports wire-based language (Shatri, V. et al 2012) with input source file as ASCII text that can be created with general purpose text editors. Geometry of the problem is expressed as number of bodies in space with global coordinate system $Ox_gY_gZ_g$. Each body has its own coordinate system $Ox_bY_bZ_b$ and will be composed from number of elements, also having their own element coordinate system $Ox_eY_eZ_e$. Positions in space are described with homogeneous coordinates (Fu, K. S. et al 1987). This facilitates geometry description by using predefined unity elements that will be scaled and positioned in space by transformations. These actions are described with homogeneous transformation matrixes, and will be created from a set of elementary transformations that have corresponding elementary homogeneous transformation matrix. Elementary transformations used

are rotations around coordinate system axis ($T_{x,\alpha}, T_{x,\phi}, T_{x,\theta}$), translation relative to some coordinate system ($T_{(x,y,z)}$), local scaling (T_{scale_xyz}), and global scaling (T_{scale_global}). Position and orientation of bodies is expressed relative to global coordinate system, whereas position and orientation of elements is expressed relative to body coordinate system.

Wire with square cross-section is basic building block for composing bodies. Width of start cross-section lies along x-axis, height along y-axis and initial wire advance is in direction of z-axis of element coordinate system. Elements are chained according to language syntax, by automatically assigning next element coordinate system. Types of elements that can be used and respective instructions are:

- Lines - **LB, LE, LR,**
- Corners - **TC,** and
- Arcs - **AR.**

Body definition follows standard structure:

```

BS Name ... % Body Start
OR ... % Origin
SW ... % Select Wire
*
* % composing elements
*
BE % Body End
    
```

were **Origin** instruction can be used to set start position for first element relative to body coordinate system, or position for other element if discontinuities are present. Variables and constants can be used to make description easier. They are defined with **VL** instruction (**VariableLocal**).

MATLAB function **plotGD** can be used to see defined geometry and corresponding coordinate systems. It is possible to control which coordinate systems will be shown with axis length and thickness.

5. Meshing

Generation of mesh cells according to PEEC method requirements is done with **GD2GDmesh** function. Meshing strategies can be selected by setting variable **meshType** to desired mesh type. Mode of meshing can be global or local. In global mode whole geometry is meshed with selected mesh type. When local mode is selected, each element individually can be meshed with selected mesh type. Possible choices for mesh type are: 1D, 2D, 2DS, 3D, and 3Dxyz. One dimensional (1D) will make divisions in length direction only (element z-axis). Two dimensional (2D) meshing will make divisions in length and width directions, i.e. element z-axis and x-axis. When 2DS (two dimensional surface) mode is selected surface and volume cells will be created over body surface. Three dimensional (3D) will make divisions in length, width, and height direction. For all previous types element coordinate

system will serve as reference. 3Dxyz will create mesh cells with edges parallel to global coordinate system axis and with faces parallel to global coordinate system planes. Discretization of elements is controlled by three matrixes that specify divisions in length, width, and height directions. Divisions can be uniform or non-uniform.

Meshed structure can be viewed with **plotGDmesh1D** and **plotGDmesh3D** functions. First one can be used for 1D meshing type only, but it offers possibility of scaling down surface and volume cells for highlighting cell boundaries. Second function can be used for any mesh type and will show cells in real size.

Selection of mesh type will influence model size and accuracy of simulation results.

6. Partial Element Calculation

Calculating partial elements is done by calling two external applications (FastFeldSolvers), FastCap2 and FastHenry2, by Windows OLE (Object Linking and Embedding) Automation mechanism for inter-process communication. FastCap2 will be used to calculate partial capacitances (Nabors, K. *et al* 1999; Shatri, V. *et al* 2012), and results will be read back to MATLAB environment as Maxwell capacitance matrix. FastHenry2 is called next to calculate partial inductances and partial resistances (Kamon, M. *et al* 1994). Results will be placed into Inductance and Resistance matrices. Creating input files for these two applications and Automation will be handled by MATLAB function **GDmesh2FastCH**. Other functions for partial element calculation can be used by calling them instead of FastCap2 and FastHenry2.

7. Model Selection and Netlist

Standard model with frequency independent partial elements and with center-to-center retardation can be constructed from calculated partial element matrixes, with retardation computed from GDmesh data structure. Resulting model if all elements are taken into consideration is known as $[R, C^+, L_p, P, \tau]$ model (Nitsch, J. *et al* 2009). If presence of dielectrics is only accounted as additional capacitance within **P**, result will be $[R, L_p, P, \tau]$ model. If largest dimension d_{max} of the problem is small compared to shortest wavelength of interest λ_{min} retardation can be neglected and model will be quasi-static one with notation $[R, L_p, P]$. It may be considered small if $d_{max} \leq \lambda_{min}/20$. For magneto-quasi-static and for stationary currents (Scholz, P. 2010) models will be $[R, L_p]$ and $[R]$. If capacitive couplings dominate the problem $[R, P]$ (or $[R, C]$) model can be used.

Spice netlist file (*.cir) is generated with **FCH2SPICE** function. Parameter and variable values will influence type of model generated. Parameter **MS_LT** sets compatibility of netlist, and **td** selects between retarded or quasi-static model:

MS_LT='LT' - LTspice netlist
MS_LT='MS' - Multisim netlist

td='R' - Retarded model
td='Q' - Quasi-Static model.

Variable **ModType** will select model type:

ModType='RCLP' - $[R, C^+, L_p, P]$
ModType='RLP' - $[R, L_p, P]$
ModType='RL' - $[R, L_p]$
ModType='RP' - $[R, P]$
ModType='R' - $[R]$.

Retarded quantities are generated with Laplace controlled sources. Only for time-domain simulations at LTspice case one can select between Laplace or Delay sources with variable **tdLT** equal to 'L' or 'D' respectively, with 'L' as default value if not specified.

Calling syntax for **FCH2SPICE** function is

```
function [ ] = FCH2SPICE( td, C_FC,
    L_FH, R_FH, GDmesh, MS_LT )
```

It has no output parameters and will generate *.cir file with selected PEEC model type having name

PEECspice_V01.00_LT.cir - for LTspice
PEECspice_V01.00_MS.cir - for Multisim.

8. Coupled Electromagnetic-Circuit Simulation

Model for electromagnetic part of the problem will be represented by netlist file in selected format. It will be included in circuit part together with other lumped components and, if available, measuring instruments.

When using LTspice as circuit solver, netlist is added to simulation environment with .include directive, Fig. 5. Parameter will be file name of *.cir file with its path name. Access to internal nodes of the model will be with ports that have selected node name. Any number and type of lumped elements, linear or nonlinear, can be added to form desired circuit part that is coupled to electromagnetic part through declared ports. Netlist file for electromagnetic part of the problem at Fig. 5 is PEECspice_V01.00_LT.cir with directory path name C:\PEEC_Toolbox\. Four internal nodes are accessed by ports with names matching desired node names:

N0101010101, N0101310101 - Body 01
N0202010101, N0202310101 - Body 02.

Long node naming convention (**Nbbeesswwhh**) was used to make easy reference to bodies and their components (elements and cells). Node to the infinity (not shown in this figure) is named **Ninf**.

Any type of simulation can be done directly from LTspice graphical user interface. Waveform of any quantity can be viewed with waveform viewer. Simulation data are stored in file.

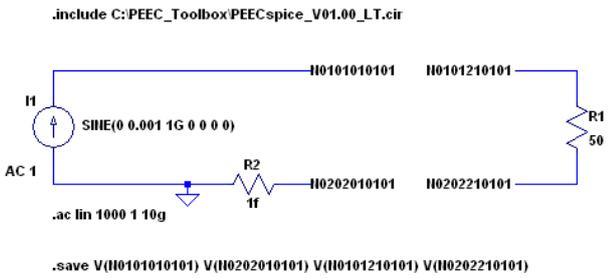


Fig. 5 Simulation of generated netlist along with additional lumped elements in LTspice

By default all circuit data are stored that may result in huge files. To decrease amount of data spice directive .save can be used to save only selected node voltages and device currents, e.g.

```
.save V(N0101010101) V(N0202010101) I(R1)
```

When post-processing is needed files need to be saved in raw format and they can be transferred to MATLAB environment with **LTSPICE2MATLAB** function (Wagner, P. 2009).

For Multisim case, netlist is included in simulation environment by adding "Arbitrary SPICE Block" and loading it with generated netlist. Like in LTspice case, access to internal nodes of the model will be by ports names matching desired internal node name as is shown in Fig. 6. In this case short node naming convention (Nnnn) was used. Node to the infinity is named **NE00**. Transferring Multisim simulation data to MATLAB has not been tested yet.

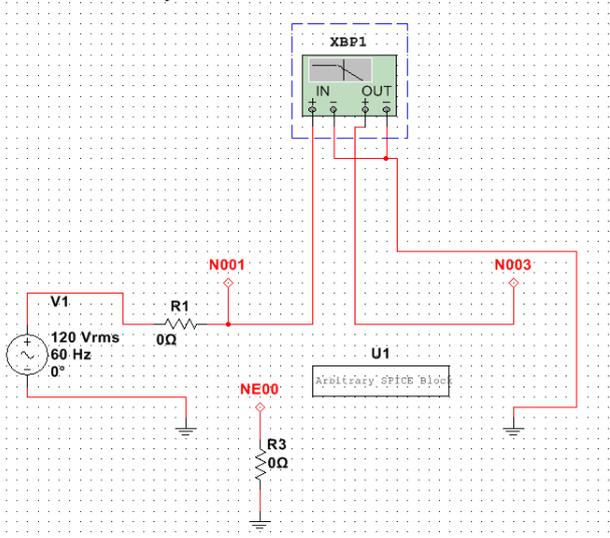


Fig. 6 Simulation of generated netlist along with additional lumped elements in Multisim

9. Simulation Examples

Two examples that follow will be used for testing PEEC Toolbox performance. Example from Fig. 1 will serve as illustration for all steps and intermediate data, from input text file that describes problem geometry to simulation

results. Only LTspice implementation and results will be shown.

2.1 Thin conductor with 1D meshing

This example has one body with one element 10mm long. Wire cross-section is 1mmx0.2mm. MATLAB editor was used to write input file that describes geometry:

```

%Thin Wire - 1D Meshing
BS C01 0.000 0.000 0.000 DX 0
SW S 0.001 0.000200 C
LR 0.010
BE
    
```

Body with name C01 (**BS C01**) starts from origin (0.000 0.000 0.000) of global coordinate system and initial progress (body Z-axis) will be along global X-axis (DX 0). Wire for composing body is of square cross-section (S parameter of **SW** instruction) with dimensions 1mmx0.2mm. (0.001 0.000200, width and height parameters of **SW** instruction are always in meters). View point for cross-section is its center (C parameter). First body composing element (in this example also the only one) is **LR** (Line Relative) with 0.010m length. Body definition was ended with **BE** instruction. Text was saved in file with name ThinWire_10mm_1mm_200um.gd.

Input file parsing according to wire-based language is done with **File2GD** function. Parsing results are stored into GD structure. By using **plotGD** function one can see composed geometry and corresponding coordinate systems: global, body and element. Geometry for this example is shown in Fig. 7.

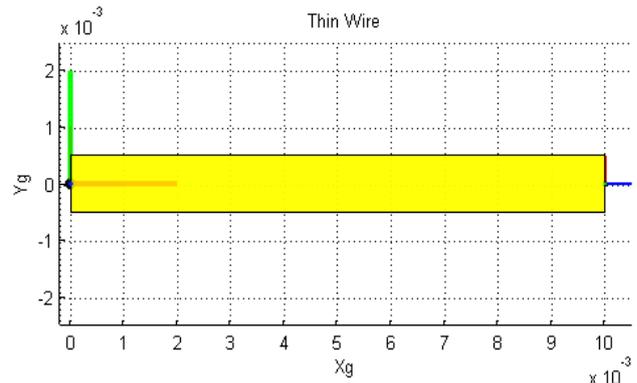


Fig. 7 Defined geometry for Thin Wire example.

Meshing is done by calling **GD2GDmesh** function with GD structure as parameter. It will return new structure GDmesh with mesh data in it. Since source file has no meshing data specified, default values will be used: 1D meshing with one segment in each direction. Fig. 8 shows meshed structure, where surface cells are colored in yellow and volume cells are colored in green.

Both of them are made partly transparent so one can see inner space Surface cells are made smaller in length direction to highlight boundary between cells and to make partly clearer view of volume cell, whereas volume cells are made thinner (like filaments) with length unchanged.

There is only one volume cell, and there are two half-size surface cells on conductor edges.

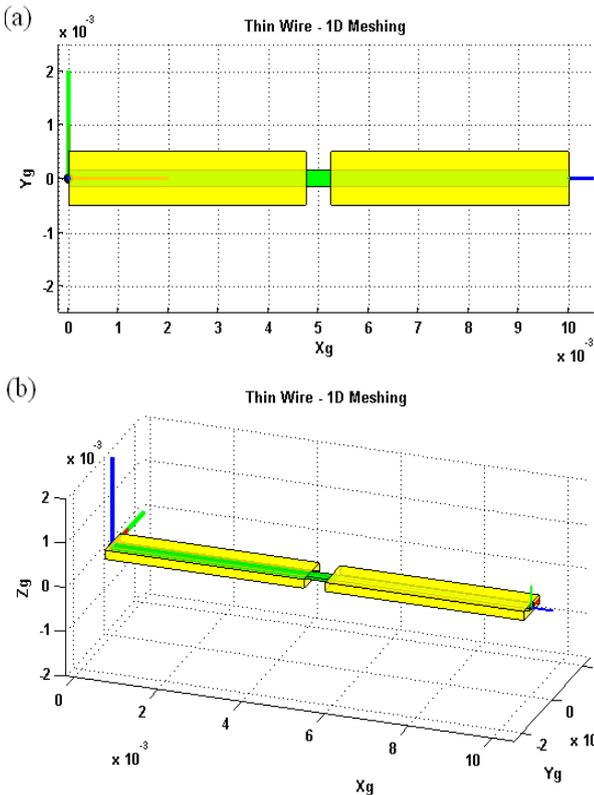


Fig. 8 1D Meshing with one volume cell and two (half-size) surface cells for Thin Wire example: (a) normal view and (b) oblique view. Surface cells are made shorter in length to highlight boundary, and volume cell is made thinner.

Fig. 8 is generated with **plotGDmesh1D**. Figure with full-size cells can be generated with **plotGDmesh3D**. Finer meshing can be controlled by corresponding variables, as in Fig. 9 where number of volume segments in length direction was set to 5 with **VL** instruction:

```
VL nl=5
```

that must be located before the element to be meshed, **LR** in this case.

Variable values and its effect will remain unchanged until new value is assigned, even through other bodies and their composing elements, or until overridden by some higher level feature. Volume cell boundaries are visible through transparent surface cells and they can be distinguished at middle of inner surface cells.

Function **GDmesh2FastCH** will calculate partial elements for meshed geometry described with **GDmesh** structure. Partial capacitances calculated by **FastCap2** are returned in **C_FC** matrix with **nSxnS** dimension, where **nS** is number of surface cells. Partial inductances and partial resistances calculated by **FastHenry2** are returned in **L_FH** matrix with **nVxnV** dimension and **R_FH** vector of **nVx1** dimension, where **nV** is number of volume cells.

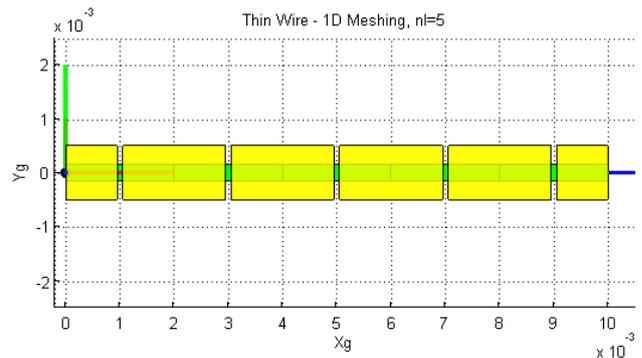


Fig. 9 1D Meshing with five volume cells ($nl=5$) and six surface cells (four with full-size and two on edges with half-size) for Thin Wire example.

Retarded model in form of Spice netlist for this example is given in Fig. 10. Circuit will have two self-capacitances, two **LAPLACE** current-controlled current-sources that represent capacitive couplings, and branch with self-inductivity and resistance of volume cell. Each branch has one 0 volt voltage source that supplies controlling current for couplings. Model has only one volume cell, therefore no inductive couplings are present. This file resulted after calling **FCH2SPICE** function. Colored view is obtained by opening netlist file with **LTspice**.

Node names with underscore symbol represent branch internal nodes. Usually these nodes will not be accessed from additional external circuit part. Nodes of interest for this wire would be first and last node in length direction: **N0101010101** and **N0101020101**.

```
* C:\PEEC_Toolbox\PEECspice_U01.00_LT.cir
* LTspice format

* Self-Capacities - Cii
vUc0101010101 N0101010101 N0101010101c 0.00
cC0101010101_00000001 N0101010101c Ninf
+ +01.0068609546248100e-013
vUc0101020101 N0101020101 N0101020101c 0.00
cC0101020101_00000000 N0101020101c Ninf
+ +01.0068609546248093e-013

* indKP=true -> YES Capacitive Couplings - Pij
* LAPKmin=0.000
bI0101010101_0101020101 Ninf N0101010101c
+ I=i(vUc0101020101)*(+01.9712205988073872e-001)
+ LAPLACE= exp(-s*(+03.333333333333335e-011))
bI0101020101_0101010101 Ninf N0101020101c
+ I=i(vUc0101010101)*(+01.9712205988073855e-001)
+ LAPLACE= exp(-s*(+03.333333333333335e-011))

* Self-inductivities - Lii
vU0101010101_0101020101 N0101010101 N0101010101_0101020101r
+ 0.00
rR0101010101_0101020101 N0101010101_0101020101r N0101010101_0101020101l
+ +08.6206896551724039e-004
lL0101010101_0101020101 N0101010101_0101020101l N0101020101
+ +06.6909258385833728e-009

* Inductivity Couplings - Kij
* indKL=true -> YES Indictivity Couplings - Lij
* LAPKLmin=0.000

.end
```

Fig. 10 Netlist for retarded PEEC model for Thin Wire example with one volume cell and two surface cells (1D Meshing)

Thin wire can be one of real wires for connecting circuit lumped elements. Sample circuit in **LTspice** for simulating coupled electromagnetic-circuit problem with retarded PEEC model of Thin Wire is given in Fig. 11. "Real" wire

is used for connecting R1 and R2 and is represented by two ports, together with included PEEC model netlist file. Node names correspond to discretization with one volume cell were last node number in length direction is 02 (N0101020101). Similar circuit connected with ideal wire (R3-R4) is shown in same Figure.

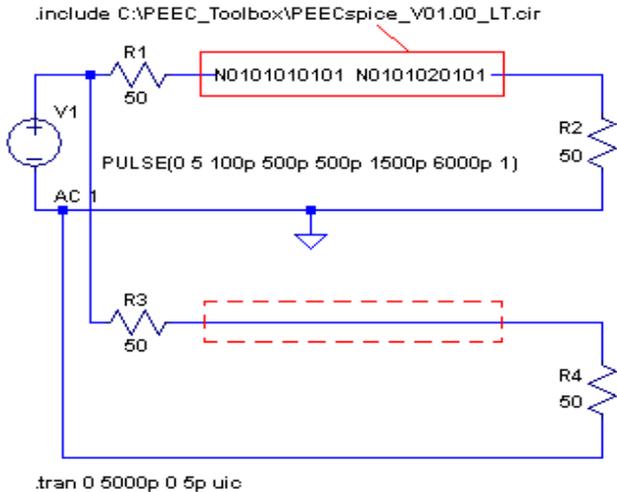


Fig. 11 LTspice circuit for testing influence of real wire connection (model of Thin Wire example included in *.cir file) in contrast to ideal connection.

Transient response for pulse excitation is shown in Fig. 12. Blue line gives ideal response when influence of connection trace is not taken into account. Two other traces are for cases with Thin Wire 1D PEEC model, red for one volume cell and green for ten volume cells, and almost overlap. Later case manifested late time instabilities (Garret, J. et al 1997). Simulation got stabilized by including damping resistor in parallel with partial self-inductances (Ekman, J. et al 2004; Nitsch, J. et al 2009). This option is offered by LTspice and no change to PEEC model has been done.

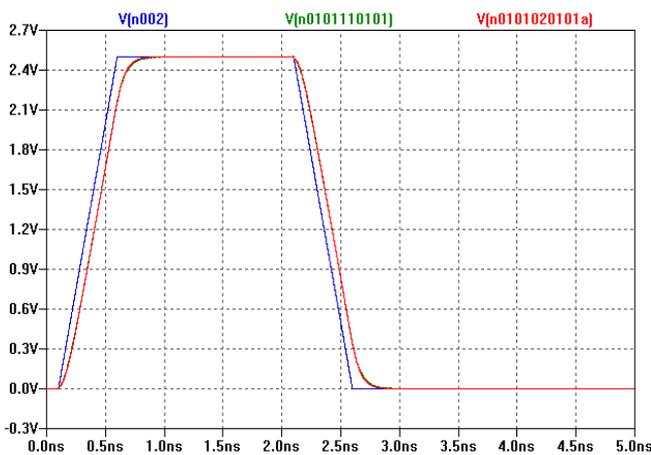


Fig. 12 Transient response for ideal connection (blue), and for connection with Thin Wire from example modeled with one volume cell (red) and with ten volume cells (green).

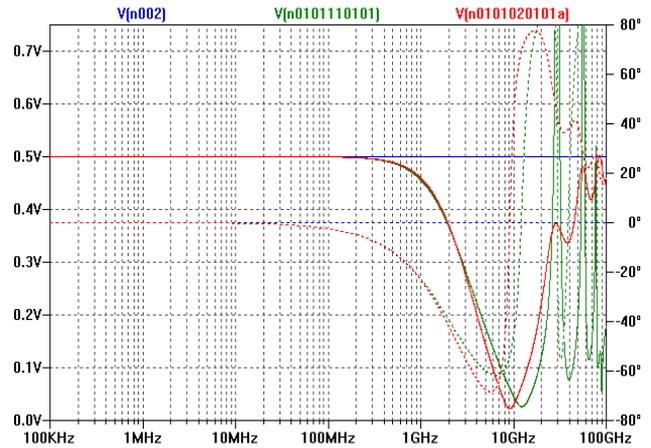


Fig. 13 Frequency response for ideal connection (blue), and for connection with Thin Wire from example modeled with one volume cell (red) and with ten volume cells (green). Both, magnitude (solid line) and phase (dashed line) are plotted.

Fig. 13 shows frequency response for same cases. According to rule of thumb, condition relating discretization length Δl and maximum frequency of interest f_{max} (Nitsch, J. et al 2009; Ekman, J. 2003)

$$\Delta l \leq \frac{c}{20 \cdot f_{max}} \quad (3)$$

gives f_{max} equal to 1.5GHz for first case (red), and around this frequency response starts getting separated more noticeably from response of second case (green) used as reference. Discretization length for second case is 1mm giving 15GHz as maximum frequency of interest. Constant 20 in (3) denominator normally is around 10-20 with preference to upper values (Ekman, J. et al 2004) for more accurate results, and expectations for reliable response are up to 3GHz for first case and up to 30GHz for second case.

Thin Wire retarded PEEC model with ten volume cells has 10+1 self-capacities, 10 self-inductivities, (10+1)*10 current-controlled LAPLACE current-sources for implementation of capacitive couplings and 10*(10-1) current-controlled LAPLACE current-sources for implementation of inductivity couplings. Computer used for simulation was Laptop with Intel Pentium M processor, 1.73GHz clock, and 512MB RAM. Transient response duration was 5ns and maximum time step was set to 5ps, which results with at least 1000 time steps. Simulation time was around 40 seconds with LTspice "Normal" solver, where 28 seconds were spent calculating the impulse responses from Laplace functions. When "Alternate" solver was used times were around 72 and 53 seconds respectively. Simulation results and time of simulation remained same regardless of integration method selected from methods offered by LTspice.

Simulated frequency response covered the range from 100kHz to 100GHz, i.e. six decades, with 1000 points per decade and 6000 frequencies points in total. Simulation

lasted around 5-6 seconds for all frequency points. Damping resistors did not influence simulation.

2.2 Parallel-plate transmission line fed differentially

Next example used for testing PEEC solver is parallel-plate transmission line. Plates are 50mm in length, 20µm wide, and 1µm thick. Separation between plates is 20µm. Plates are meshed by 20 cells each in length direction only. Input file for describing problem geometry is:

```
% Parallel Plate Transmission Line

VL nl=20
BS C01 0 0 0.000010 DX 0
SW S 0.000020 0.000001 C
LR 0.050
BE

VL nl=20
BS C02 0 0 -0.000010 DX 0
SW S 0.000020 0.000001 C
LR 0.050
BE
```

Resulting meshed structure (surface cells) is shown in Fig. 14, where width and height are scaled up for better viewing. Time for meshing was 4 seconds. Second VL line will give us possibility for different meshing for each body. Corresponding netlist for retarded model is put on Fig. 15. For sake of brevity, for group of similar cells, only data for first and last line are given, with three dots (...) representing the other. Calculation of partial elements by **GDmesh2FastCH** lasted 7 seconds, while time needed for **FCH2SPICE** function to generate netlist was 12 seconds.

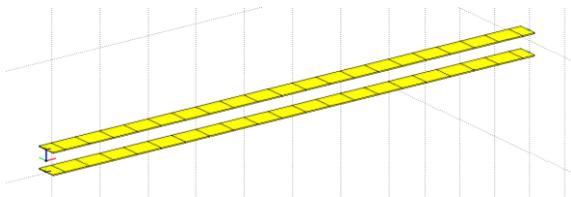


Fig. 14 Surface cells for parallel-plate transmission line. Its length is 50mm, plate cross-section is 20µmx1µm (w.h), and distance between plates is 20µm. Plates are discretized by 20 volume cells and 21 surface cells (19 full- and 2 half-size surface cells).

LTspice circuit for simulating transmission line model is shown on Fig. 16. Current source is used for differential excitation and line is terminated with 50Ω resistor (Ekman, J. et al 2004; Ruehli, A. E. et al 2003). Input ports are named N0101010101 and N0202010101, whereas output ports as N0101210101 and N0202210101. Bold numbers are used to highlight mesh nodes in length direction used for input and output, with 01 as first node of element and 21 as last one in this case. Generally last node of element is one more than number of divisions per

element, i.e. 20+1. Time- and frequency-domain simulations can be performed on same circuit.

```
* C:\PEEC_Toolbox\PEECspice_U01.00_LT.cir
* LTspice Format

* Self-Capacities - Cii
vUc0101010101 N0101010101 N0101010101c 0.00
cC0101010101 00000001 N0101010101c Minf
+ +01.2687495996875046e-014
...

vUc0101210101 N0101210101 N0101210101c 0.00
cC0101210101 00000061 N0101210101c Minf
+ +01.2687495998238846e-014
...

vUc0202010101 N0202010101 N0202010101c 0.00
cC0202010101 00000904 N0202010101c Minf
+ +01.2687495996874589e-014
...

vUc0202210101 N0202210101 N0202210101c 0.00
cC0202210101 00001764 N0202210101c Minf
+ +01.2687495998238693e-014
...

* indKP=true -> YES Capacitive Couplings - Pij
* LAPKMin=0.000
bI0101010101_0101020101 Ninf N0101010101c
+ I=i(vUc0101020101)*(+06.8322294802243808e-002)
+ LAPLACE= exp(-s*(+08.333333333333353e-012))
+ window=2.000000000000001e-010 nfft=65536
...

bI0202210101_0202200101 Ninf N0202210101c
+ I=i(vUc0202200101)*(+06.8322293926660393e-002)
+ LAPLACE= exp(-s*(+08.333333333333353e-012))
+ window=2.000000000000001e-010 nfft=65536
...

* Self-inductivities - Lii
vU0101010101_0101020101 N0101010101 N0101010101_0101020101r 0.00
rR0101010101_0101020101 N0101010101_0101020101r N0101010101_0101020101
+ +02.1551724016395477e+000
lL0101010101_0101020101 N0101010101_0101020101 N0101010101_0101020101
+ +02.9869469875818987e-009
...

vU0202200101_0202210101 N0202200101 N0202200101_0202210101r 0.00
rR0202200101_0202210101 N0202200101_0202210101r N0202200101_0202210101
+ +02.1551724016395477e+000
lL0202200101_0202210101 N0202200101_0202210101 N0202200101_0202210101
+ +02.9869469944192686e-009
...

* Inductivity Couplings - Kij
* indKL=true -> YES Inductivity Couplings - Lij
* LAPKMin=0.000
bI0101010101_0101020101_0101020101_0101030101
+ N0101020101 N0101010101_0101020101
+ I=i(vU0101020101_0101030101)*(+01.1580585438237033e-001)
+ LAPLACE= exp(-s*(+08.333333333333353e-012))
+ window=2.000000000000001e-010 nfft=65536
...

bI0202200101_0202210101_0202190101_0202200101
+ N0202210101 N0202200101_0202210101
+ I=i(vU0202190101_0202200101)*(+01.1580585438237033e-001)
+ LAPLACE= exp(-s*(+08.333333333333353e-012))
+ window=2.000000000000001e-010 nfft=65536
...

.end
```

Fig. 15 Parallel-plate transmission line PEEC model. For brevity, only a part of netlis is given. Missing part of similar lines is marked with three dots (...)

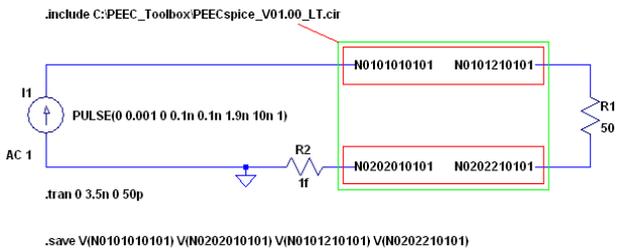


Fig. 16 LTspice circuit for simulating transmission line model excited differentially with current source and terminated with resistor.

Configuration for time-domain simulations was chosen to be as in (Ruehli, A. E. et al 2003). Current source will generate trapezoidal current pulse with 100ps rise-time, 1.9ns on-time, 100ps fall-time, and 1mA on-current.

Plotted waveforms are for transmission line input voltage and for voltage on load resistor. Fig. 17 gives LTspice waveforms for retarded PEEC transmission line model. Parameters used by Laplace controlled sources are window=0.2ns (window length) and nfft=4096 (number of Fast Fourier Transform data points). Engine solver was "Normal" and integration method "modified trapezoidal". All coupling between partial elements were used, and number of Laplace CCCS for this example (20 volume cells per segment) is

Capacitive couplings

$$(P_{ij}): (2*(20+1))*(2*(20+1)-1)$$

Inductive couplings

$$(L_{ij}): (2*20)*(2*20-1)$$

Total: 3282 CCCS.

Simulation time for calculating the impulse responses from Laplace functions, and matrix initialization was around 79 seconds. After that solution speed went with ~2ps/s. Waveform are in good agreement with waveforms from (Ruehli, A. E. et al 2003).

Solution waveforms for quasi-static transmission line model are shown in Fig. 18. Simulation conditions are same as for retarded model case. Since all frequency components are same for both models, waveforms in Fig. 18 are phase distorted versions of waveforms in Fig. 17. Phase distortion is caused by not taking into account travel time for coupling signals. Relation between both waveforms is similar to relation of waveforms given in Fig. 5 of (Heeb, H. et al 1992). Solution time for quasi-static case is ~5 seconds.

Parameters for Laplace sources have great influence on solution time and results of simulation. LTspice Laplace controlled source (b LTspice element - arbitrary behavioral voltage or current sources) performance is controlled by two parameters: window length and number of FFT data points. Window length is to be chosen such as to accommodate impulse response of Laplace function part of source. Usual artifacts of FFT (Chu, E. 2008) may appear when calculating impulse response by inverse FFT of discrete set of frequency response points. Laplace function for all coupling sources is pure time delay, with impulse response being single pulse at specified delayed time position. According to this, window length must be longer than largest time delay for problem to be analyzed. If largest distance is d_{max} , then window length must be

$$window > \frac{d_{max}}{c}, \tag{4}$$

where c is speed of electromagnetic wave propagation. For transmission line problem d_{max} is length of diagonal, and is of almost same length as line itself. Window length would be

$$window > \frac{0.05}{3 \cdot 10^8} s = 0.167 ns, \tag{5}$$

and adopted window length is 0.2ns, as can be seen from netlist in Fig. 15. To minimize picket fencing artifacts,

"fence openings" positions should be as close as possible to dominant excitation frequencies. The reciprocal of window length is frequency resolution, and also position of first "fence opening". Number of FFT data points times frequency resolution will set highest frequency considered, and at the same time positions of consecutive "fence openings" up to highest frequency. Number of FFT data points for netlist in Fig. 15 is 65536. If window and nfft parameters are not set, LTspice will make a guess for them. Guessed values were window=7ns and nfft=32768, with results similar to the case with window=0.2ns and nfft=1024

$$32768 / (7/0.2) = 936 \approx 1024,$$

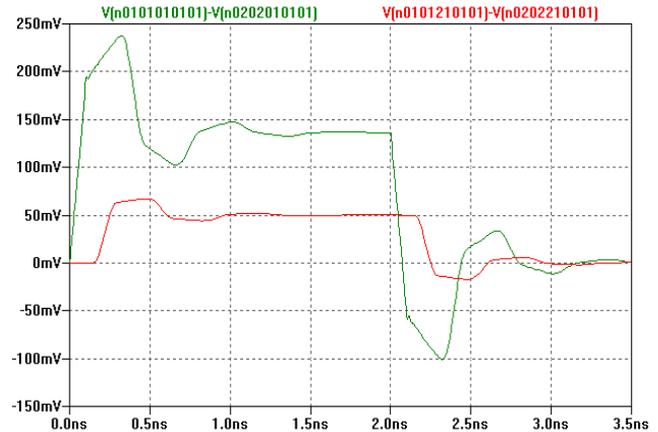


Fig. 17 LTspice input voltage (green) and output voltage (red) waveforms for retarded PEEC transmission line model with window=0.2ns (window length) and nfft=4096 (number of FFT data points) as parameters for Laplace controlled sources. Engine solver was "Normal" and integration method "modified trapezoidal".

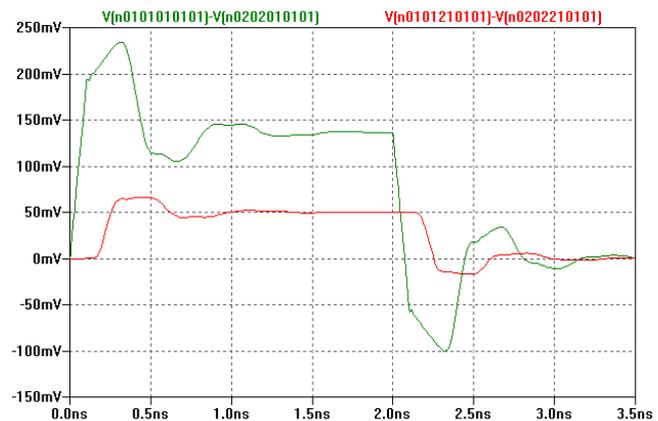


Fig. 18 LTspice input voltage (green) and output voltage (red) waveforms for quasi-static PEEC transmission line model. Engine solver was "Normal" and integration method "modified trapezoidal".

but it executes slower and requests more memory for storing impulse responses.

Times for matrix initialization and simulation speed for selected values of nfft are given in Table I.

Frequency domain simulations were performed for frequencies from 1Hz up to 10GHz at 1000 point. Simulations were completed in about 17 seconds. Frequency responses for real and imaginary part of output voltage for retarded PEEC transmission line model are shown in Fig. 19. Solid line represents real part, whereas dashed line represents imaginary part. Imaginary part agrees with results of (Ekman, J. et al 2004). Quasi-static frequency response is same to response in Fig. 19, only zero-cross in range 8GHz to 9GHz is slightly shifted to the left, $f_{zCR}=8.65902\text{GHz}$ and $f_{zCQ}=8.63986\text{GHz}$. This version of Toolbox shows also artifacts with non-overlapping cells. Fig. 20 shows simulation for quasi-static PEEC transmission line model, with 10 volume cells for upper conductor and 20 volume cells for bottom conductor, showing artifact around 8.5GHz (encircled part).

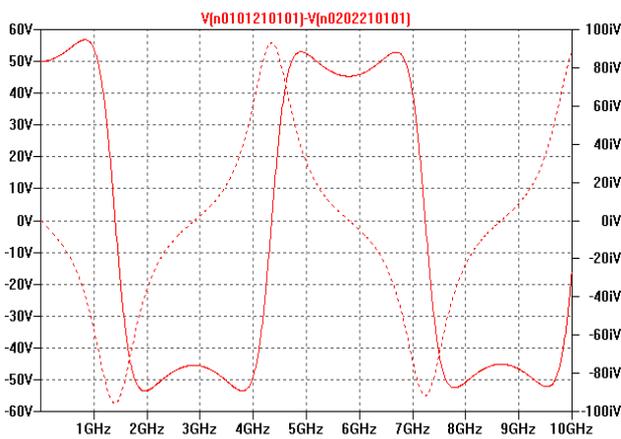


Fig. 19 Frequency responses for real (solid line) and imaginary part (dashed line) of output voltage for retarded PEEC transmission line model.

Table I: Time for calculating impulse responses and matrix initialization, and simulation speed for selected number of FFT data points. Window length was 0.2ns.

Number of FFT data points	Time calculating impulse responses [s]	Simulation speed [fs/s]
256	35	30000
512	40	15000
1024	44	7200
2048	57	3400
4096	87	1560
8192	161	630
16384	356	305
32768	845	134
65536	1765	43

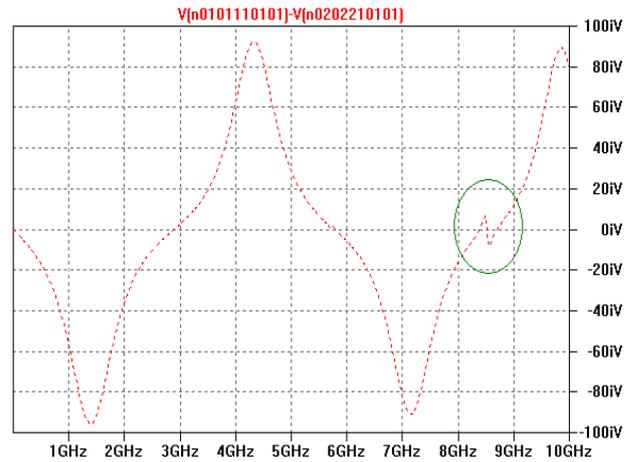


Fig. 20 Frequency response for imaginary part of output voltage for quasi-static PEEC transmission line model, with 10 volume cells for upper conductor and 20 volume cells for bottom conductor, showing artifact around 8.5GHz (encircled part).

10 Conclusions

Prime objective of this work was to create platform that can be used during research on problems related to PEEC solvers. MATLAB was selected as integrating platform, and small number of functions, directly related to one of sub-steps for PEEC modeling and simulation, will enable easy access even for novice in the field.

Contribution to this direction is also wire-based language that is simple, but in the same time can be used to describe wide range of problem geometries. It features chaining property that will construct bodies from list of basic elements by automatically joining them, including branching in any direction in three dimensional space. Language can be easily extended with new elements with specific behavior regarding connection with neighboring elements in chain and meshing.

3Dxyz meshing type results with classical PEEC model were analytical expressions for calculation of partial elements can be used. Other types will result with more compact models, but generally relative position between cells can be arbitrary (nonparallel). Interface for FastCap2, used to calculate partial capacitances, by design can handle non-orthogonal cells, including arbitrary shapes for non PEEC related problems. Interface for FastHenry2 uses filament approximation of non-orthogonal cells. Usage of homogenous coordinates and corresponding transformation matrixes when creating this Toolbox carries all information needed to specify position and orientation of volume cells for FastHenry2.

Retarded and quasi-static full models are generated. Results of simulation with LTspice are reported in this paper. Simulations in frequency-domain are handled easily. Time-domain simulations can be improved significantly with proper choice of parameters for LAPLACE functions. Guideline for selecting window length in relation to problem dimensions is given with (4), were adopted value is slightly more than lowest bound. Up

to 4096 FFT data point, time for calculating impulse responses increases slowly with doubling data points, around 2.5 times for 16 times increase in number of FFT data points (256 to 4096). After this point time doubles when doubling FFT data points. Feasibility of this Toolbox, including integrated applications, is demonstrated with good agreement of results obtained with results reported elsewhere.

Research will continue by including other physical domains in PEEC solver for modeling and solving multiphysics problems. Non-orthogonal cells and models for joining elements with discontinuous grid point is second direction of research on PEEC method.

Acknowledgment

Special thanks go to Mike Engelhardt, from Linear Technology, for his readiness and instant support on overcoming problems we faced when using LTspice to solve problems with hundreds of thousands of LAPLACE controlled sources.

References

- Bález-López, D., and Gguerrero-Castro, F. E. (October 25, 2011), Circuit Analysis with Multisim, Moran & Claypool Publishers.
- Chu, E. (2008), Discrete and Continuous Fourier Transforms: Analysis, Applications and Fast Algorithms, Chapman & Hall/CRC.
- Daroui, D., and Ekman, J. (2010), Parallel Implementation of the PEEC method, *The Applied Computational Electromagnetics Society Journal*, vol. 25, no. 5, pp. 410-422.
- Daroui, D., and Ekman, J. (2012), Performance analysis of parallel non-orthogonal PEEC-based solver for EMC applications, *Progress in electromagnetic Research B*, vol. 41, pp. 77-100.
- Ekman, J. (2003), Electromagnetic modeling using the partial element equivalent circuit method, Ph.D. Theses, EISLAB, Luleå, Sweden.
- Ekman, J., and Antonini, G. (09 August 2004), *IEEE International Symposium on Electromagnetic Compatibility (EMC 2004)*, Silicon Valley, California.
- FastFeldSolvers, www.fastfieldsolvers.com.
- Fu, K. S., Gonzalez, R. C., and Lee, C. S. G. (1987), Robotics: Control, Sensing, Vision, and Intelligence, McGraw-Hill, Inc.
- Garret, J., Ruehli, A. E., and Paul, C. (1997), Accuracy and stability advancements of the partial element equivalent circuit model, in *Proc. of the International Symposium on EMC*, Zurich, Switzerland, March 1997, pp. 529-534.
- Gibson, W. C. (2008), *The Method of Moments in Electromagnetics*, Chapman and Hall/CRC.
- Gope, D., Ruehli, A., and Jandhyala, V. (May 2007), Solving low-frequency EM-CKT problems using the PEEC method, *IEEE Trans. on Advanced Packaging*, vol. 30, no. 2.
- Heeb, H., and Ruehli, A. E. (November 1992), Three-dimensional interconnect analysis using partial element equivalent circuits, *IEEE Trans. on Circuits and Systems: Fundamental Theory and Applications*, vol. 39, no. 11.
- Kamon, M., Tsuk, M. J., and White, J. K. (September 1994), FASTHENRY: A multipole-accelerated 3-D inductance extraction program, *IEEE Trans. on Microwave Theory and Techniques*, vol. 42, no. 9.
- Linear Technology, LTspice IV, <http://www.linear.com/designtools/software>.
- Miscione, G., and Antonini, G. (April 2006), An integral method based approach for EM analysis, EMC Laboratory, Dipartimento di Ingegneria Elettrica e dell'Informazione, Università degli Studi di L' Aquila, Poggio di Roio, Italy, Tech. Rep. R.06-87.
- Müsing, A., Ekman, J., and Kolar, J. W. (March 2009), Efficient calculation of non-orthogonal partial elements for the PEEC method, *IEEE Trans. on Magnetics*, vol. 45, Issue: 3, pp. 1140-1143.
- Nabors, K., and White, J. (1999), Multipole-accelerated 3-D capacitance extraction algorithm for structures with conformal dielectrics, *29th ACM/IEEE Design Automation Conference*.
- Nagel, L. W., and Pederson, D. O. (Apr. 1973), SPICE (Simulation Program with Integrated Circuit Emphasis), Memorandum No. ERL-M382, University of California, Berkley.
- Nitsch, J., Gronwald, F., Wollenberg, G., (2009), Radiating Non-Uniform Transmission Line Systems and the Partial Element Equivalent Circuit Method, John Wiley & Sons, Ltd.
- Paul, C. R. (2010), Inductance: loop and partial, Hoboken, New Jersey: John-Wiley & Sons, Inc.
- Polycarpou, A. C. (, October 25, 2006), Introduction to the Finite Element Method in Electromagnetics, Moran & Claypool Publishers.
- Rong, A., and Cangellaris, A. C. (2002), Robust multi-GHz electromagnetic analysis of high-speed interconnects and integrated passives, *WSEAS International Conference on Electronics, Control & Signal Processing and E-Activities*, Singapore.
- Ruehli, A. E. (March 1974), Equivalent circuit models for three-dimensional multiconductor systems, *IEEE Trans. on Microwave Theory and Techniques*, vol. MTT-22, no.3.
- Ruehli, A. E., and Heeb, H. (July 1992), Circuit models for three-dimensional geometries including dielectrics, *IEEE Trans. on Microwave Theory and Techniques*, vol. 40, no.7.
- Ruehli, A. E., Antonini, G., Esch, J., Ekman, J., Mayo, A., and Orlandi, A. (May 2003), Nonorthogonal PEEC formulation for time- and frequency-domain EM and circuit modeling, *IEEE Trans. on Electromagnetic Compatibility*, vol. 45, no. 2.
- Sadiku, M. N. O. (2001), Numerical techniques in electromagnetics, second edition, CRC Press LLC.
- Scholz, P. (2010), Analysis and numerical modeling of inductively coupled antenna systems, Ph.D. dissertation, Institut für Theorie Elektromagnetischer Felder (TEMF), Technische Universität Darmstadt, Deutschland.
- Shatri, V., Sefa, R., and Kurtaj, L. (2012), Partial element equivalent circuit (PEEC) toolbox for MATLAB, integrating FastCap2 and FastHenry2 to calculate partial elements, and Multisim or LTspice for circuit simulation, in *Proc. of 12th WSEAS International Conference on Systems Theory and Scientific Computation (ISTASC'12)*, Istanbul, Turkey, August 21-23, 2012, pp. 163-169.
- Shatri, V., Sefa, R., and Kurtaj, L. (2012), Relative position between mesh cell edges and interrelation with accuracy when using FastCap2 for capacitance calculations of 3-D PEEC-like cells, in *Proc. of 12th WSEAS International Conference on Systems Theory and Scientific Computation (ISTASC'12)*, Istanbul, Turkey, August 21-23, 2012, pp. 213-218.
- Vladimirescu, A. (1994), *The SPICE book*, John Wiley and Sons, Inc.
- Wagner P. (26 April 2009), Fast import of compressed binary .raw files created with LTspice circuit simulator, Available: <http://www.mathworks.com/matlabcentral/fileexchange/23394-fast-import-of-compressed-binary-raw-files-created-with-ltspice-circuit-simulator>
- Yutthagowith, P. (2011), Implementation of a Transmission Line model with the PEEC Method for Lightning Surge Analysis, *6th IASME/WSEAS International Conference on Energy & Environment*, Cambridge, UK.