Electromagnetics Simulator openEMS for Analysis of RF Components in the IHP OpenPDK

Mustafa Alchalabi, Jan Taro Svejda, Daniel Erni

General and Theoretical Electrical Engineering (ATE), University of Duisburg-Essen (UDE), and Center for Nanointegration Duisburg-Essen (CENIDE), D-47048 Duisburg, Germany

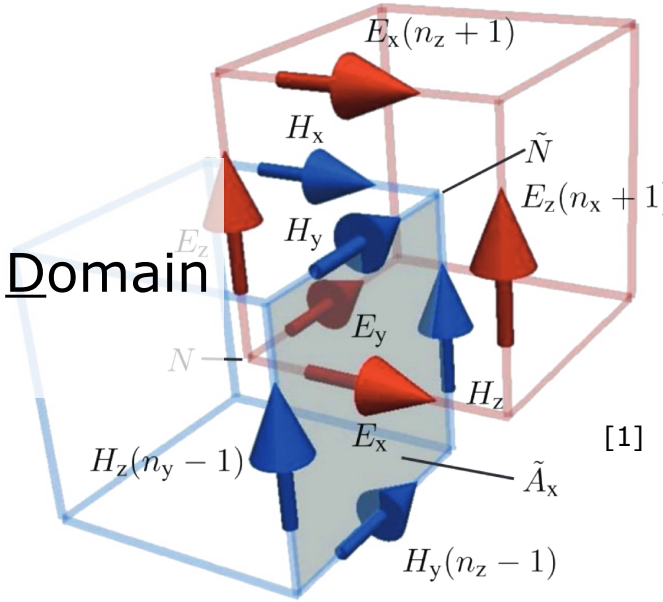
mustafa.alchalabi@uni-due.de

Outline

- What is openEMS?
  - Controlling
  - Analysis
- How is openEMS utilized in the IHP OpenPDK?
  - Input / Output Formats
  - Simulation of Passive RF Structure
- What are the Challenges?
  - Automated Discretization
  - Models of Common Structures
- Summary and Outlook

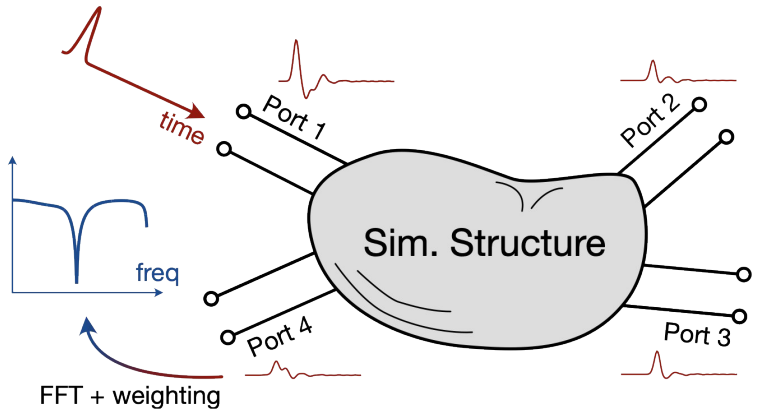
What is openEMS?

- An open source electromagnetic field solver
- EM field calculations with EC-FDTD[1,2] scheme
  - Equivalent Circuit – Finite Differences in Time Domain
  - Current and voltages form update equations



[1]

- Time domain method
  - Broad frequency band covered by one simulation
  - Net response to Gaussian pulse excitation
  - FFT to frequency domain
- Controlled by powerful scripting interfaces
  - Python, or Octave (Matlab)
- Available on Linux, MacOS, Wondows



A Simulation in openEMS

A Simulation in openEMS

- Setup openEMS
  - Drawing unit (µm)
  - Freq. range
  - Domain boundaries
- Setup Materials
- Define structure using primitives like boxes
- Define mesh lines *manually*
  - Ideally along with structure creation

A Simulation in openEMS

Define simulation structure

- Setup openEMS
  - Drawing unit (µm)
  - Freq. range
  - Domain boundaries
- Setup Materials
- Define structure using primitives like boxes
- Define mesh lines *manually*
  - Ideally along with structure creation

Invoke openEMS simulation

- View created structure using AppCSXCAD
- Run openEMS:
  Grab a cup of coffee!

Do post processing steps

A Simulation in openEMS

Define simulation structure

- Setup openEMS
  - Drawing unit (µm)
  - Freq. range
  - Domain boundaries
- Setup Materials
- Define structure using primitives like boxes
- Define mesh lines *manually*
  - Ideally along with structure creation

Invoke openEMS simulation

- View created structure using AppCSXCAD
- Run openEMS:
  Grab a cup of coffee!

Do post processing steps

- Calculate port parameters
  - S-Parameters
  - Voltages, Currents
- Export filed data
  - For visualization in i.e. ParaView
- Calculate power loss densities
- ...

A Simulation in openEMS

```
% Parameteres
addpath('~/opt/openEMS_build/share/openEMS/matlab');
addpath('~/opt/openEMS_build/share/CSXCAD/matlab');

const_c0 = 299792458;   const_mue0 = 1.2566370621219e-6;    const_eps0 = 1/(const_c0^2*const_mue0);
unit_len = 1e-6; % Lengths in unit um

freq_0 = 2e9;   freq_range = [0.0, 4e9];

sub_h = 1524;   sub_epsr = 3.66;    sub_tand = 0.0037;
msl_Zc = [50, 50/sqrt(2)];
for n = 1:2    msl_w(n) = calc_msl_width_by_book_of_pozar(msl_Zc(n), sub_epsr, sub_h);    end
eps_eff = (sub_epsr+1)/2;
lambda_0 = const_c0/freq_0 /unit_len;   lambda_range = const_c0./freq_range /unit_len;
mesh_max_delta = min(lambda_range) /sqrt(sub_epsr) /20;
feed_l = lambda_0 / 2;
```

A Simulation in openEMS

```
% Init FDTD and CSX
FDTD = InitFDTD('EndCriteria', 1e-6, 'Oversampling', 6);
FDTD = SetGaussExcite(FDTD, 0, max(freq_range));
FDTD = SetBoundaryCond(FDTD, {'PML_8', 'PML_8', 'MUR', 'MUR', 'PEC', 'PML_8'});
CSX = InitCSX();
% Define Structure
CSX = AddMetal(CSX, 'cond_top');
CSX = AddMaterial(CSX, 'RO4350B');
CSX = SetMaterialProperty(CSX, 'RO4350B', 'Epsilon', sub_epsr, 'Kappa', sub_tand*2*pi*10e9*sub_epsr*const_eps0);
```

(longer) code block not shown:

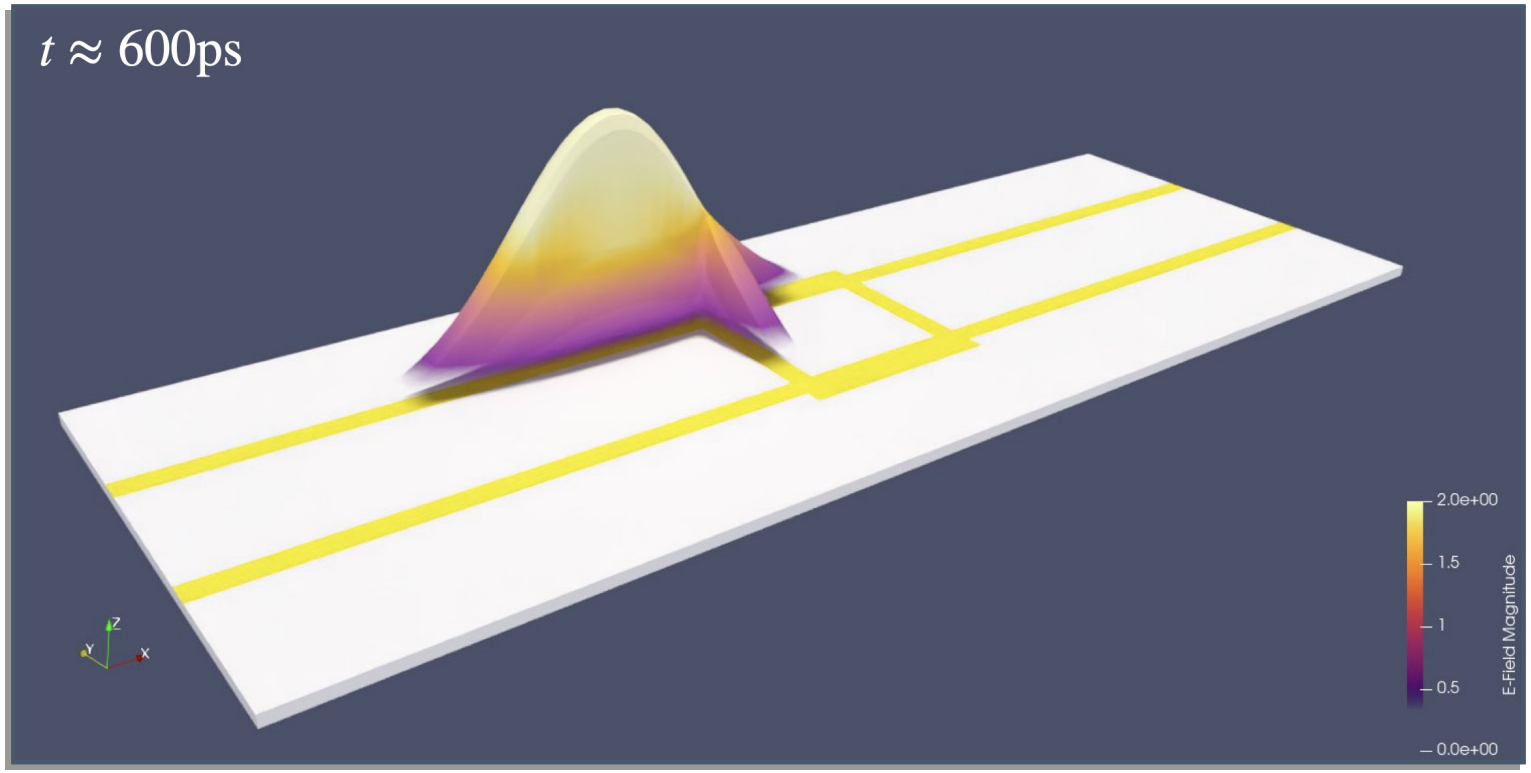
- calculating start and end points of boxes
- adding boxes to materials
- loop for similar parts
- add mesh lines of edges
- smooth mesh

```
start = [mesh.x(1),   mesh.y(1),   0];
stop  = [mesh.x(end), mesh.y(end), sub_h];
CSX = AddBox( CSX, 'RO4350B', 0, start, stop );

mesh.x = AutoSmoothMeshLines(mesh.x, mesh_max_delta, 1.25);
```

A Simulation in openEMS

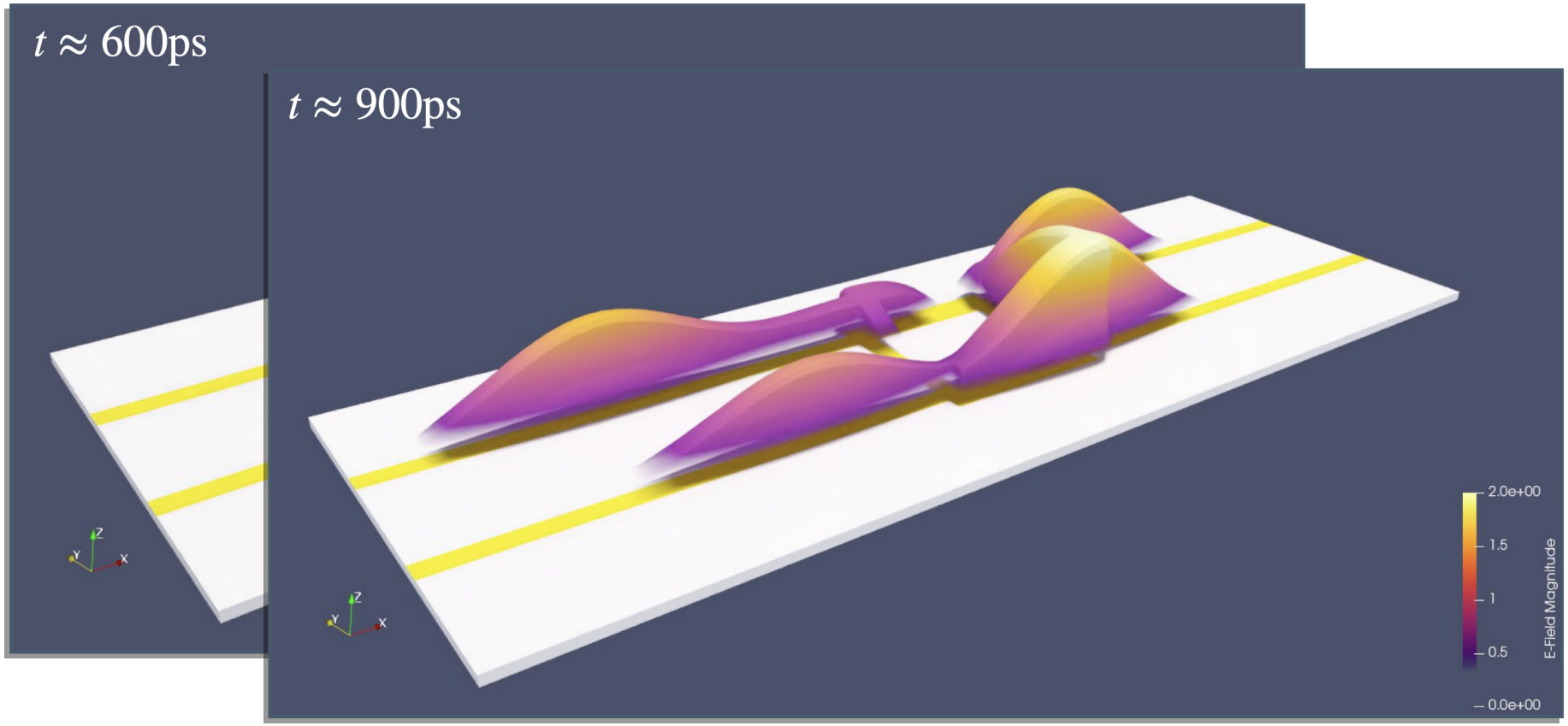
```
 ----------------------------------------------------------------------
 | openEMS 64bit -- version v0.0.35-110-g782a738
 | (C) 2010-2023 Thorsten Liebig <thorsten.liebig@gmx.de>  GPL license
 ----------------------------------------------------------------------
        Used external libraries:
                CSXCAD -- Version: v0.6.2-124-gb5919a6
                hdf5   -- Version: 1.14.1
                          compiled against: HDF5 library version: 1.14.1-2
                tinyxml -- compiled against: 2.6.2
                fparser
                boost  -- compiled against: 1_81
                vtk -- Version: 9.2.6
                       compiled against: 9.2.6

Create FDTD operator (compressed SSE + multi-threading)
FDTD simulation size: 83x83x178 --> 1.22624e+06 FDTD cells
FDTD timestep is: 5.58967e-12 s; Nyquist rate: 30 timesteps @2.98169e+09 Hz
Excitation signal length is: 1026 timesteps (5.73501e-09s)
Max. number of timesteps: 1000000000 ( --> 974659 * Excitation signal length)
Create FDTD engine (compressed SSE + multi-threading)
Running FDTD engine... this may take a while... grab a cup of coffee?!?
[@        4s] Timestep:          203 || Speed:   61.8 MC/s (1.985e-02 s/TS) || Energy: ~1.03e-18 (- 0.00dB)
…
[@       52s] Timestep:         3157 || Speed:   74.7 MC/s (1.641e-02 s/TS) || Energy: ~4.52e-20 (-52.49dB)
Time for 3157 iterations with 1226242.00 cells : 52.75 sec
Speed: 73.39 MCells/s
```

A Simulation in openEMS



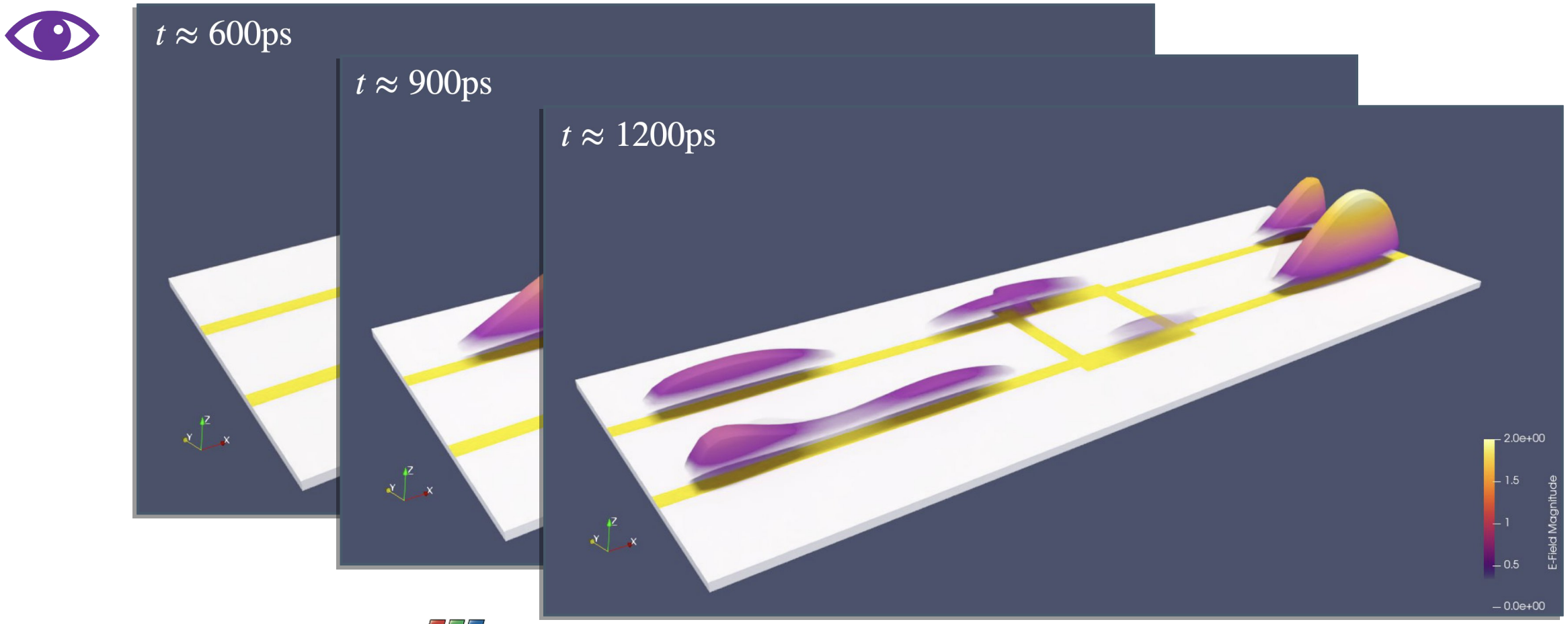
visualized with ParaView

A Simulation in openEMS



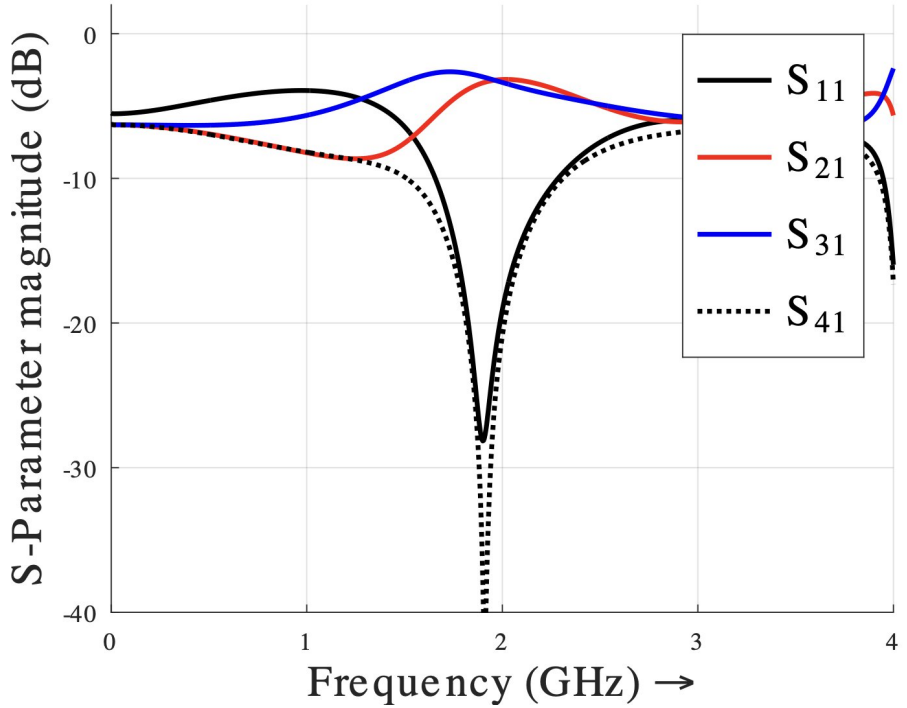
visualized with ParaView

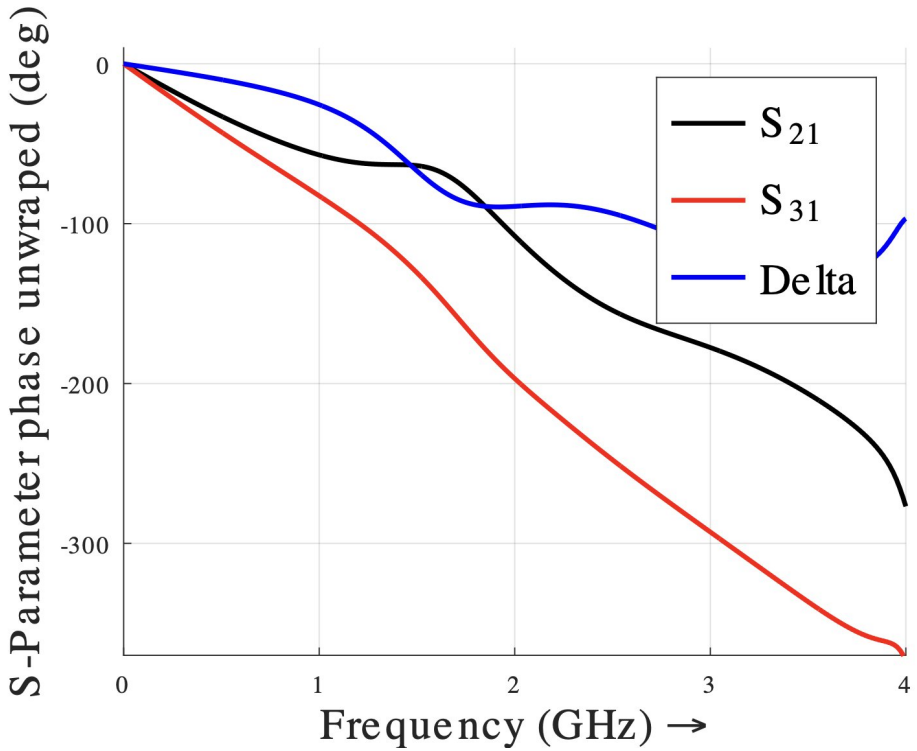
A Simulation in openEMS



visualized with ParaView

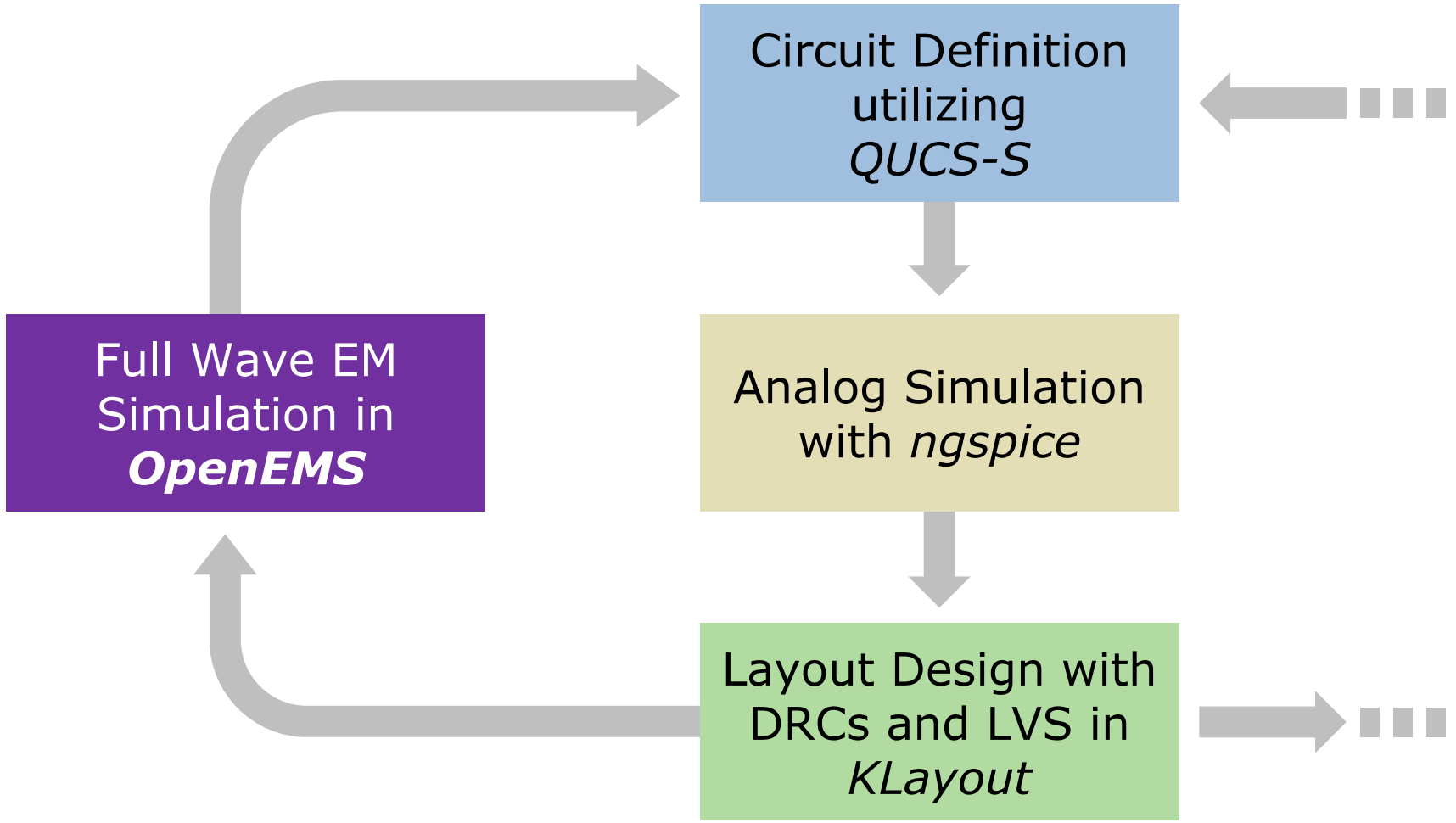
A Simulation in openEMS



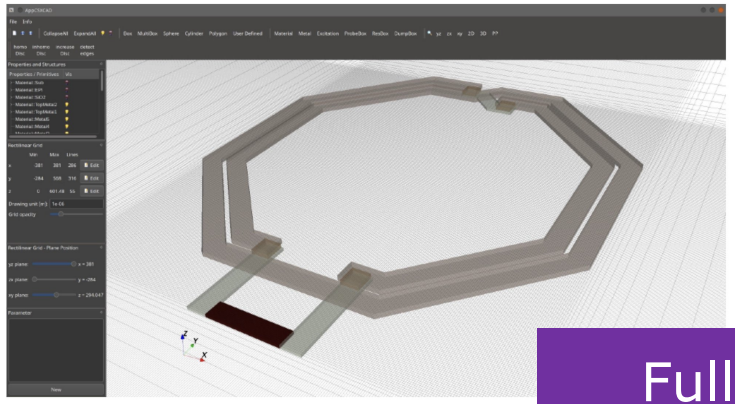


- Comparison with / Extraction for circuit simulation model

OpenEMS in OpenPDK



OpenEMS in OpenPDK



Example simulation:
Inductor coil with two turns

Inductance?
Parasitic elements?

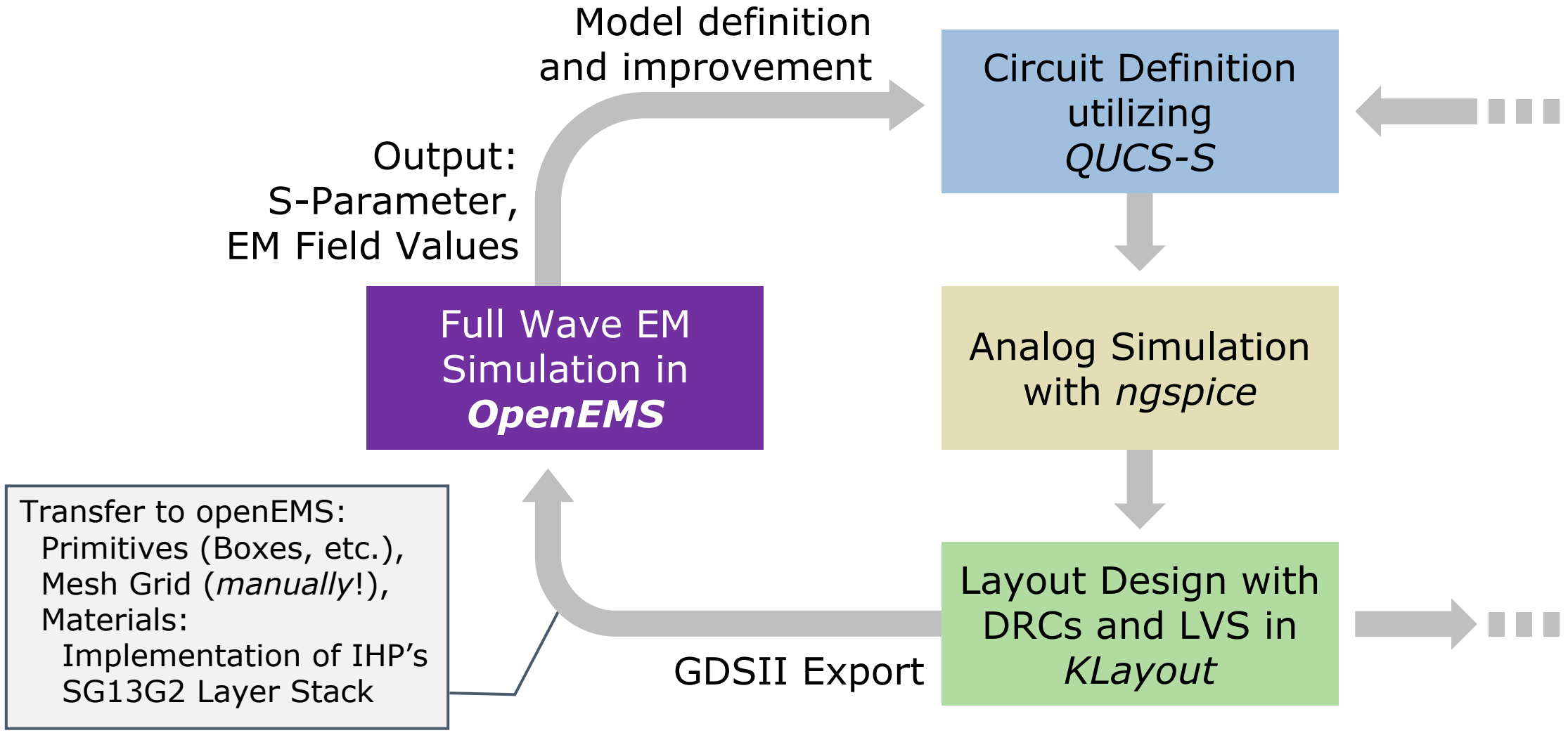
Circuit Definition utilizing *QUCS-S*

Analog Simulation with *ngspice*

Layout Design with DRCs and LVS in *KLayout*

Full Wave EM Simulation in ***OpenEMS***

OpenEMS in OpenPDK



Future of openEMS

- Challenges of openEMS' integration to OpenPDK
  - GDSII import and SG13G2 layer stack implementation
  - Manual mesh grid generation
  - Uniform export of simulation results
    (file naming, file types, figures of interest, etc.)

Future of openEMS

- Challenges of openEMS' integration to OpenPDK
  - GDSII import and SG13G2 layer stack implementation → available in IHP's openPDK GIT-repo.
  - Manual mesh grid generation
  - Uniform export of simulation results
    (file naming, file types, figures of interest, etc.)

Future of openEMS

- Challenges of openEMS' integration to OpenPDK
  - GDSII import and SG13G2 layer stack implementation → available in IHP's openPDK GIT-repo.
  - Manual mesh grid generation
  - Uniform export of simulation results (file naming, file types, figures of interest, etc.)

- Project DI–DEMICO
  - DE:Sign Enablement for Millimetre-wave Integrated Circuits using Open-Source-Tools
  - Project duration May-2024 to April-2027
  - 4 Partners: TU Dresden PSN, TU Dresden IFTE, IHP GmbH, Uni DuE ATE
  - Funded by the German BMBF

Development of automated mesh creation

Implementation of common structure models

Summary and Outlook

- EC-FDTD based openEMS is suitable for simulations of RF components also in the mm-wave range
- Within the openPDK a GDSII import to openEMS is implemented utilizing the SG13G2 layer stack
- In order to be able to use openEMS effectively, automated discretization is required
  - Development within the project DI-DEMICO
- Furthermore, standard structures should be mapped as models in order to simplify designs
  - Modeling of IP blocks (e.g. pad frames, transmission lines) planned within project DI-DEMICO

Thank you for your Attention

Feel free to clone, fork and contribute:
https://www.openems.de

ATE
General and Theoretical Electrical Engineering

UNIVERSITÄT DUISBURG ESSEN
Open-Minded

References:

[1] Liebig Thorsten, Rennings Andreas, Held Sebastian, and Erni Daniel, "openEMS – a free and open source equivalent-circuit (EC) FDTD simulation platform supporting cylindrical coordinates suitable for the analysis of traveling wave MRI applications," *Int. J. Numer. Model.*, vol. 26, no. 6, pp. 680–696, 2013, doi: 10.1002/jnm.1875.

[2] A. Rennings, "Elektromagnetische Zeitbereichssimulationen innovativer Antennen auf Basis von Metamaterialien," Dr.-Ing. Dissertation, Universität Duisburg-Essen, Fak. IW / ATE, Duisburg, DE, 2008.