Evaluation of IHP workflow for AWS Palace

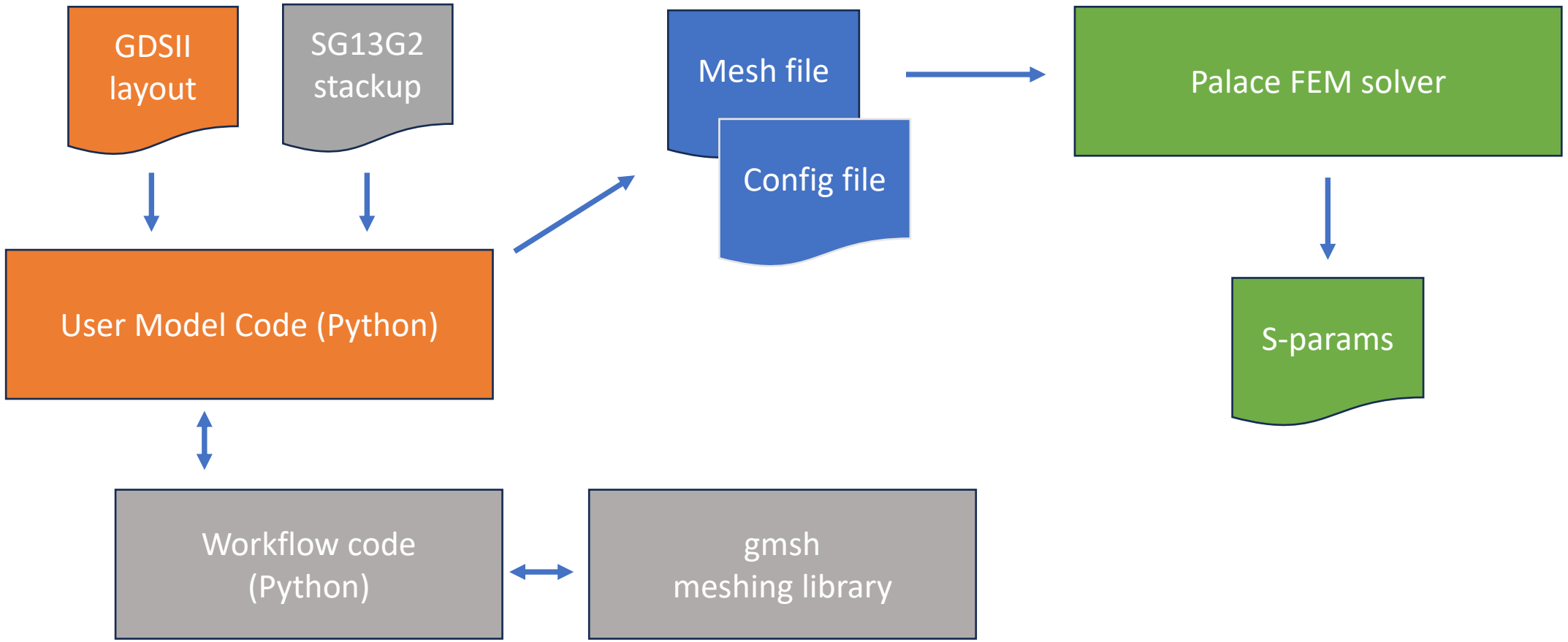
2. October 2025

Volker Mühlhaus

Overview

- Task: develop workflow for Palace FEM, similar to the openEMS flow
- End user can create simulation model with a few simple settings
- Input is GDSII layout + XML stackup
- Hide internal complexity from end user

Overview


GDSII layout
SG13G2 stackup
Mesh file
Config file
Palace FEM solver
User Model Code (Python)
S-params
Workflow code (Python)
gmsh meshing library

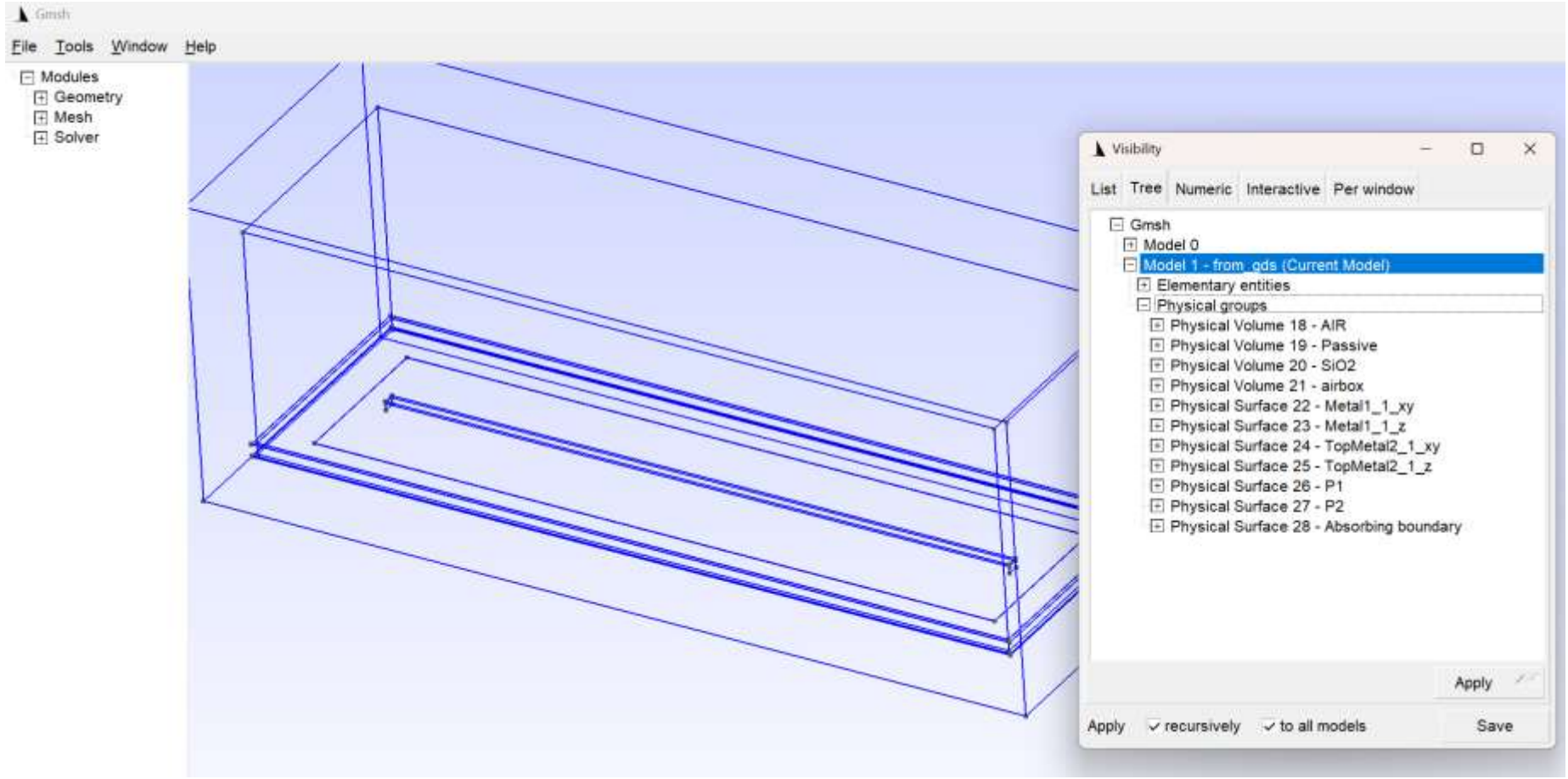
User model code

```
# ==================== input files and path settings ==================

gds_filename = "line_simple_viaport.gds"    # geometries
XML_filename = "SG13G2_nosub.xml"           # stackup

# preprocess GDSII for safe handling of cutouts/holes?
preprocess_gds = False
```

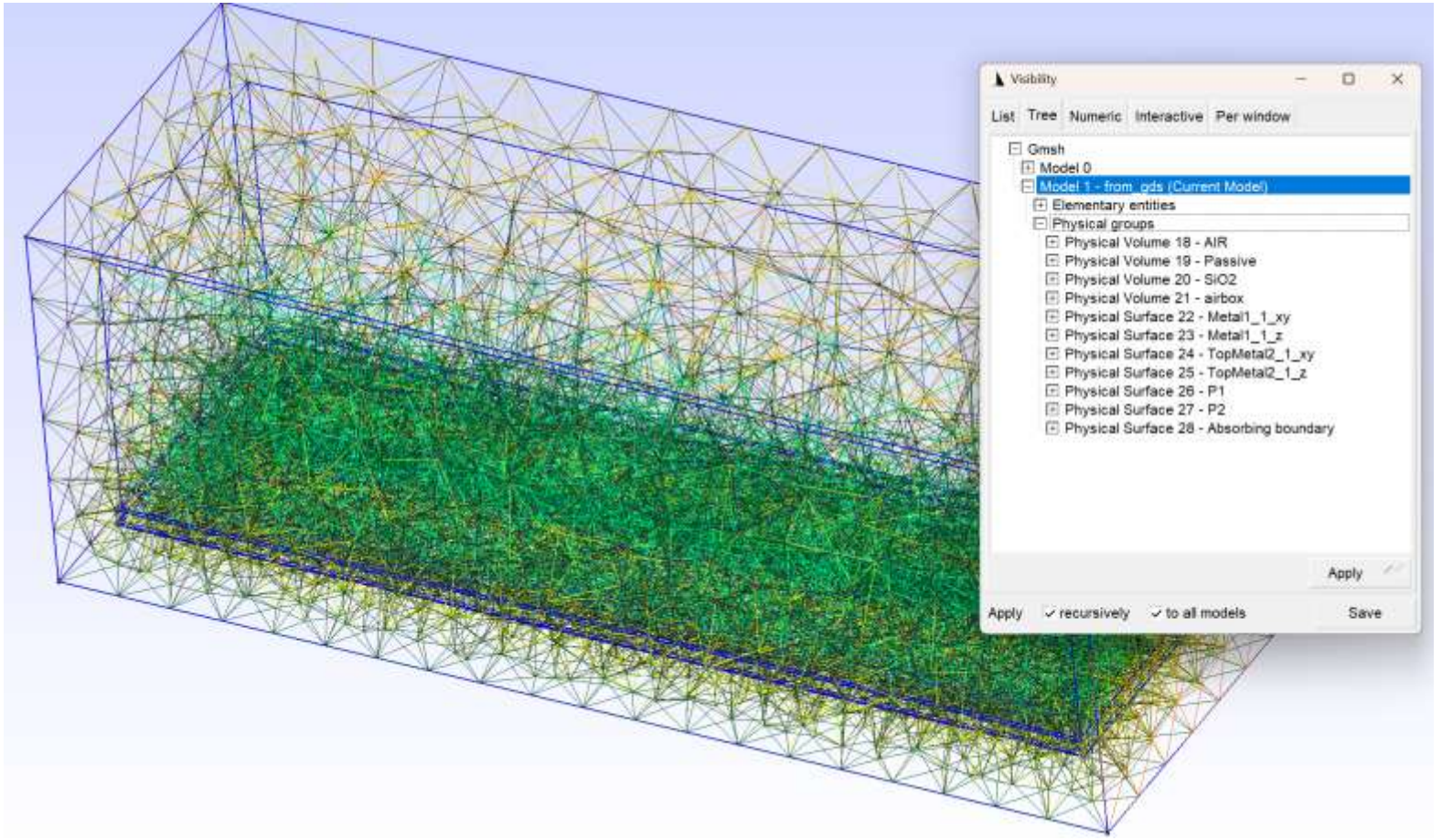
```
settings['unit']   = 1e-6  # geometry is in microns
settings['margin'] = 50    # distance in microns from GDSII geometry boundary to simulation boundary

settings['fstart']  = 0e9
settings['fstop']   = 100e9
settings['fstep']   = 2.5e9

settings['refined_cellsize'] = 2  # mesh cell size in conductor region
settings['cells_per_wavelength'] = 10   # how many mesh cells per wavelength, must be 10 or more

settings['meshsize_max'] = 70  # microns, override cells_per_wavelength
settings['adaptive_mesh_iterations'] = 0
settings['z_thickness_factor'] = 0.33
```

```
simulation_ports = simulation_setup.all_simulation_ports()
# instead of in-plane port specified with target_layername, we here use via port specified with from_layername and to_layername
simulation_ports.add_port(simulation_setup.simulation_port(portnumber=1, voltage=1, port_Z0=50, source_layernum=201, from_layername='Metal1', to_
simulation_ports.add_port(simulation_setup.simulation_port(portnumber=2, voltage=1, port_Z0=50, source_layernum=202, from_layername='Metal1', to_
```

Script runs gmsh, step 1: show geometries

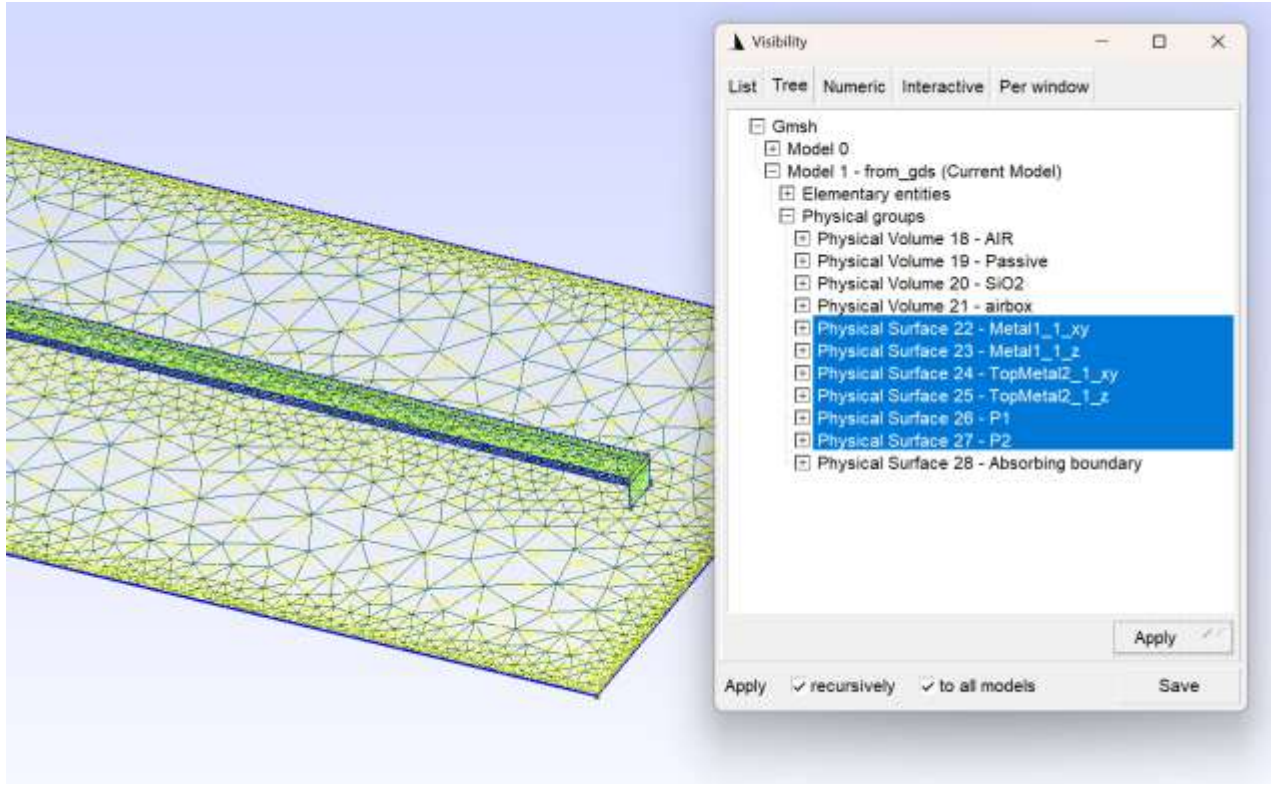


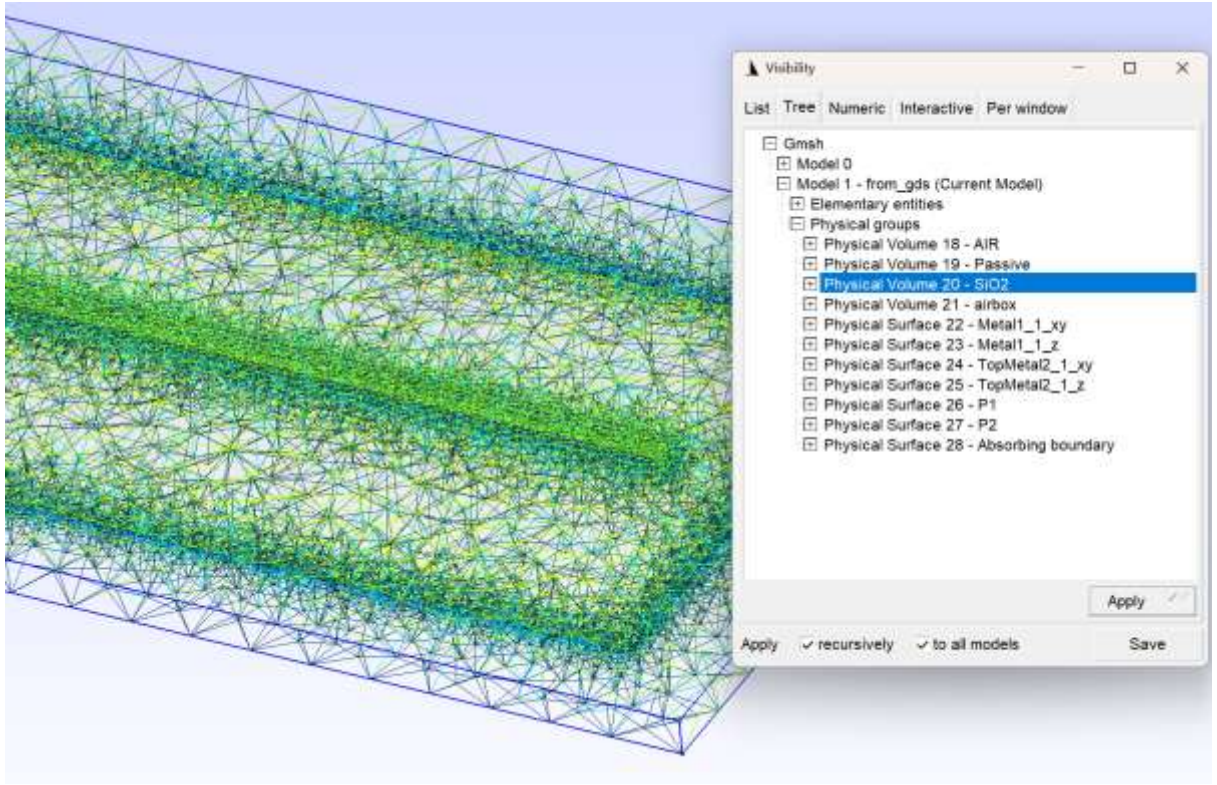
Script runs gmsh, step 2: show mesh



Script runs gmsh, step 2: show mesh

User <u>can</u> look at mesh if he is interested, but no action required!





Model files stored to disk

- Simulation control file (*.json)
- Mesh file (*.msh)
- Script to start simulation

run_sim

Datei Bearbeiten Ansicht

```
#!/bin/bash
run_palace config.json
combine_snp
```

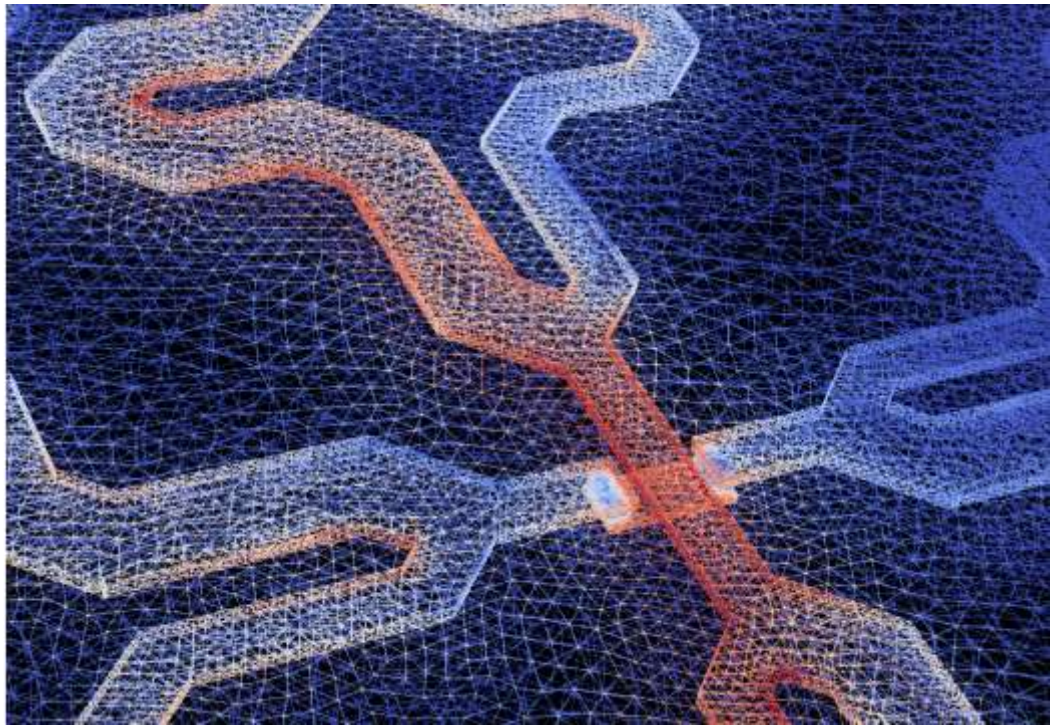
config.json

Datei Bearbeiten Ansicht

```
{
    "Problem": {
        "Type": "Driven",
        "Verbose": 3,
        "Output": "output/palace_line_viaport"
    },
    "Model": {
        "Mesh": "palace_line_viaport.msh",
        "L0": 1e-06,
        "Refinement": {}
    },
    "Solver": {
        "Driven": {
            "MinFreq": 0.1,
            "MaxFreq": 100.0,
            "FreqStep": 2.5,
            "SaveStep": 0,
            "AdaptiveTol": 0.01
        },
        "Linear": {
            "Type": "Default",
            "KSPType": "GMRES",
            "Tol": 1e-06,
            "MaxIts": 400
```

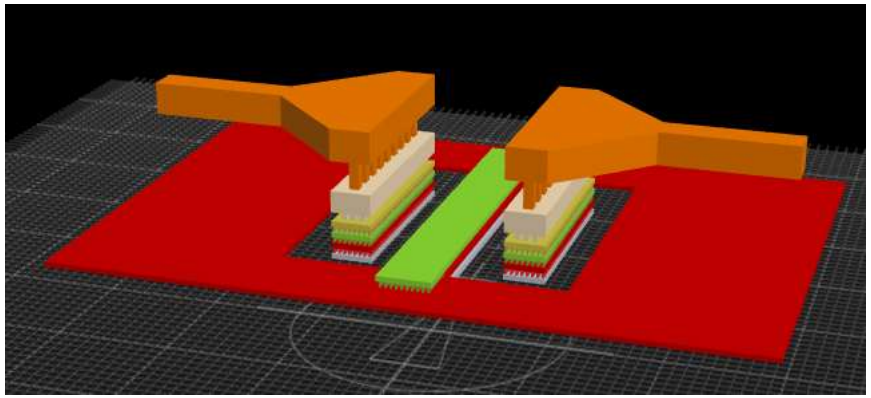
Adaptive Mesh Refinement

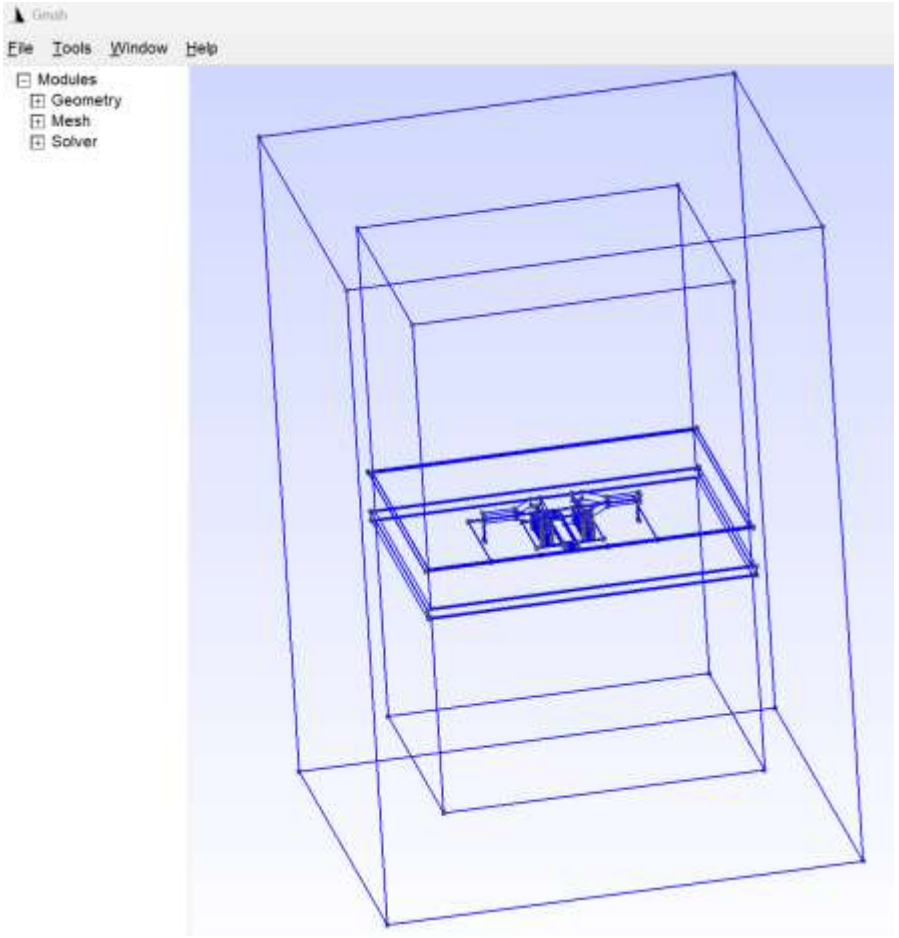
- Palace can do adaptive mesh refinement, enabled in control file
- With latest mesh code, our initial mesh is already very detailed if we specify 2 micron target meshsize
- Coase initial mesh + AMR often slower
- Better go for fine inital mesh?



Simulation time, core of medium power amp

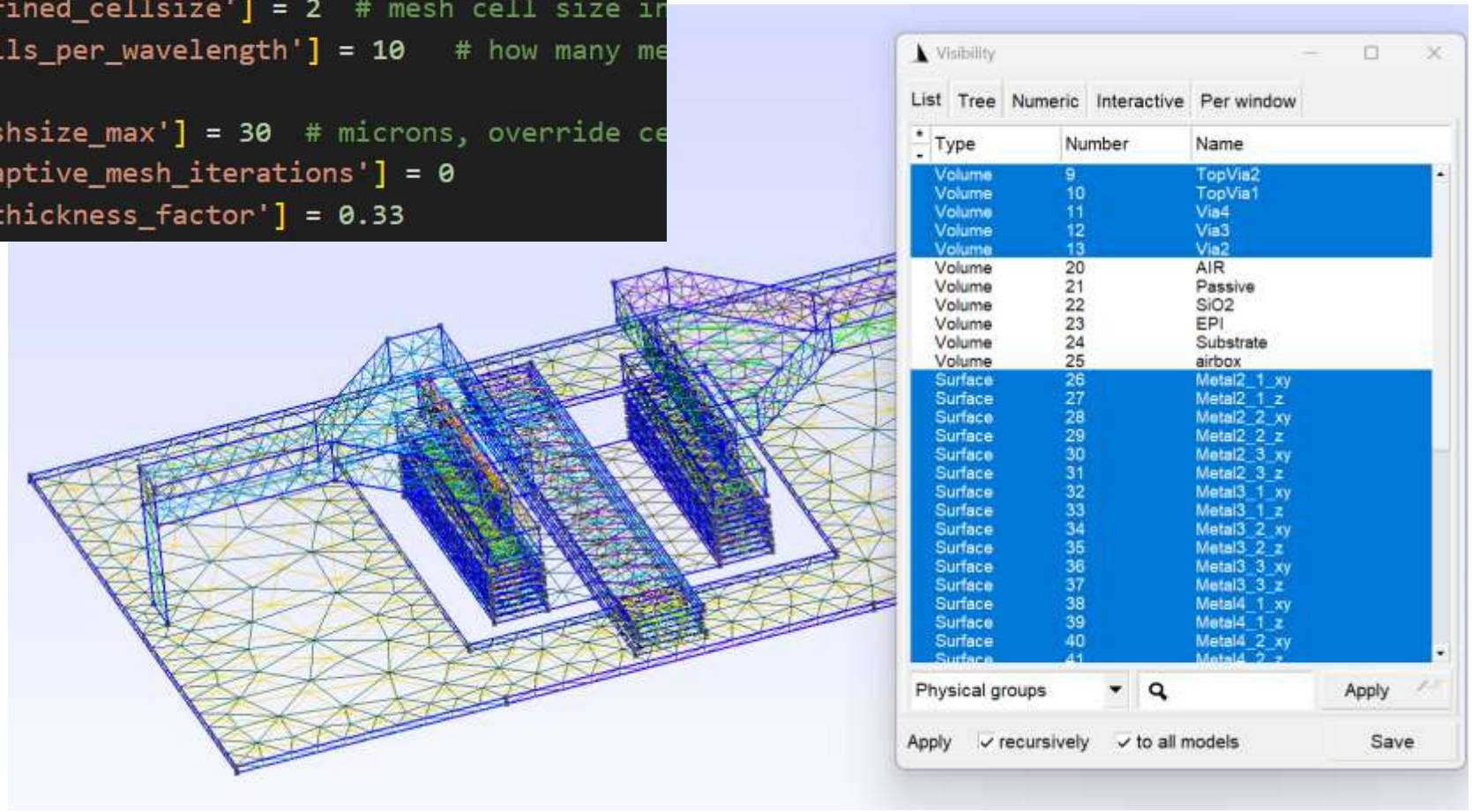
- Example from IHP Analog Academy
- Model has 4 ports



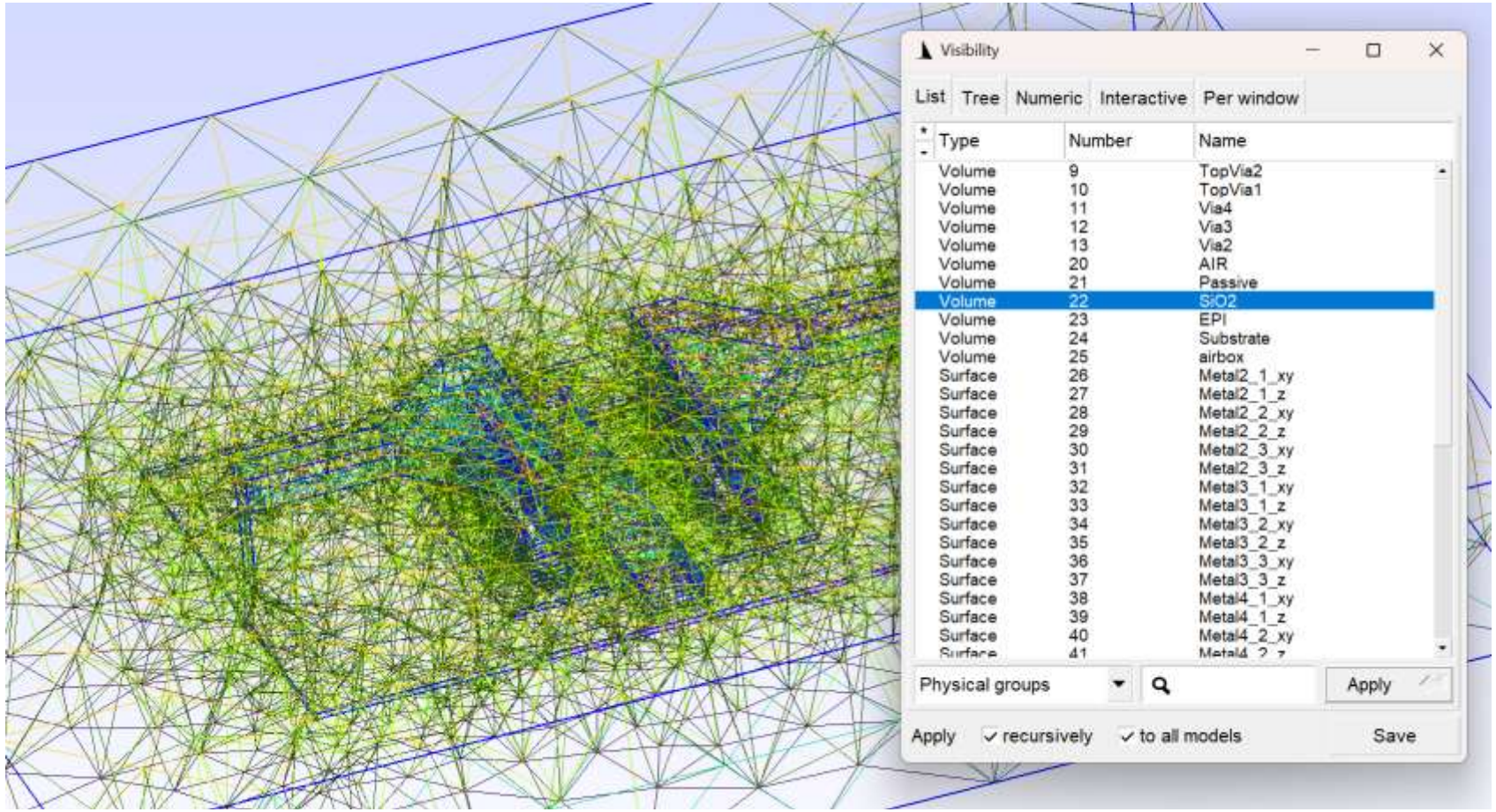


Simulation time, core of medium power amp

```
settings['refined_cellsize'] = 2  # mesh cell size in
settings['cells_per_wavelength'] = 10   # how many me

settings['meshsize_max'] = 30  # microns, override ce
settings['adaptive_mesh_iterations'] = 0
settings['z_thickness_factor'] = 0.33
```

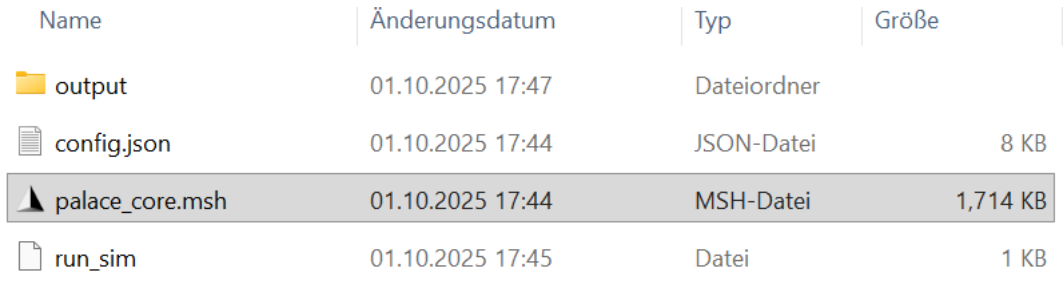


Simulation time, core of medium power amp

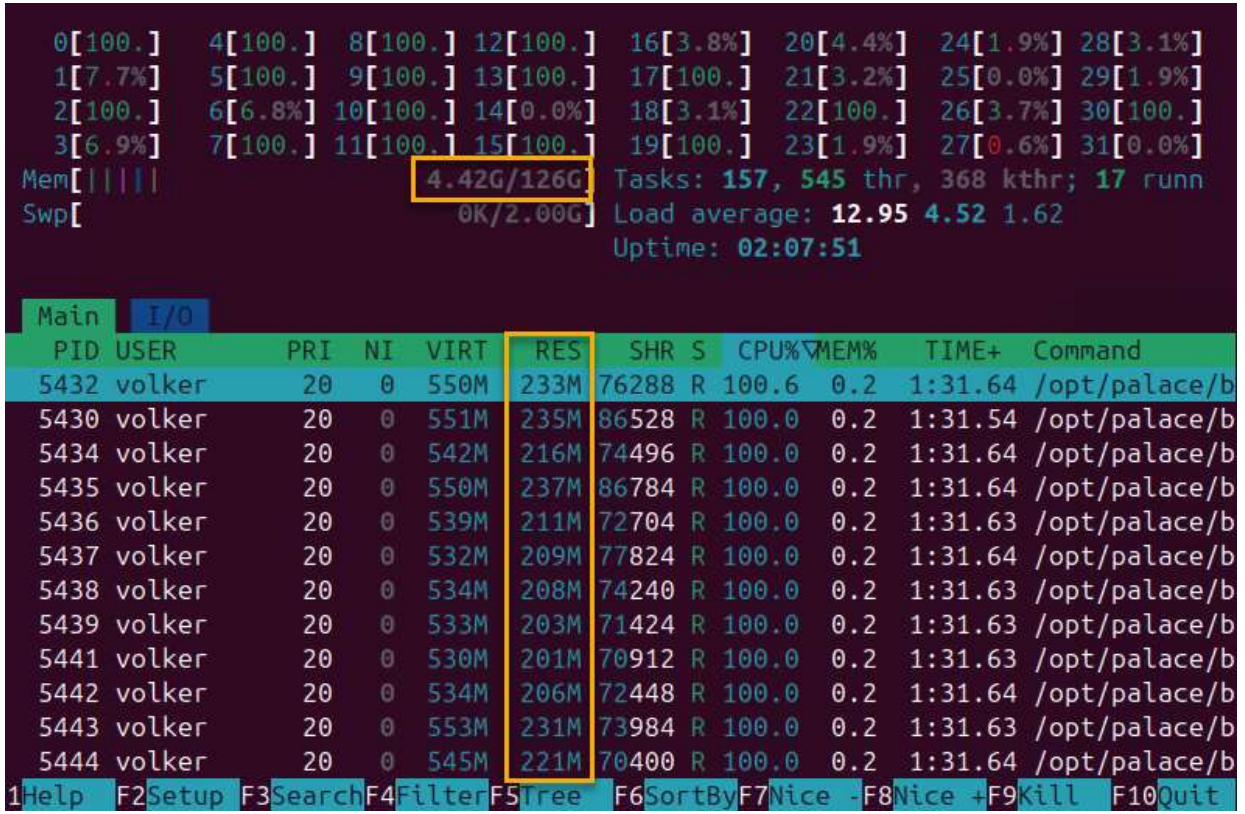


Simulation time, core of medium power amp

- Input mesh file size is 1.7 MB
- Run on 16 cores @ ~230 MB
- 4 ports
- Sweep 1 – 350 GHz , step 1 GHz



Name	Änderungsdatum	Typ	Größe
output	01.10.2025 17:47	Dateiordner	
config.json	01.10.2025 17:44	JSON-Datei	8 KB
palace_core.msh	01.10.2025 17:44	MSH-Datei	1,714 KB
run_sim	01.10.2025 17:45	Datei	1 KB



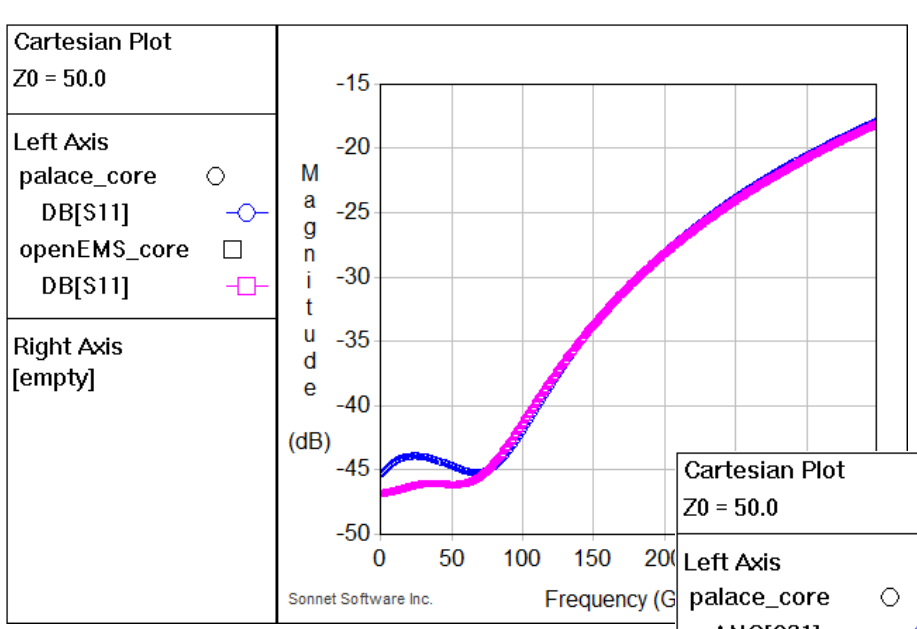
Simulation time, core of medium power amp

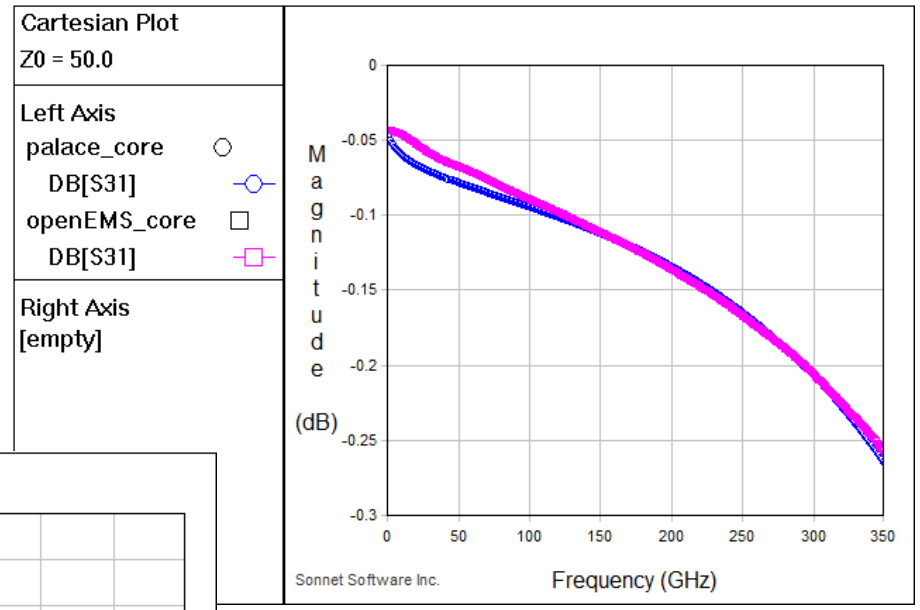
- Adaptive frequency sweep required 26 discrete frequencies
- Total time: 520 s

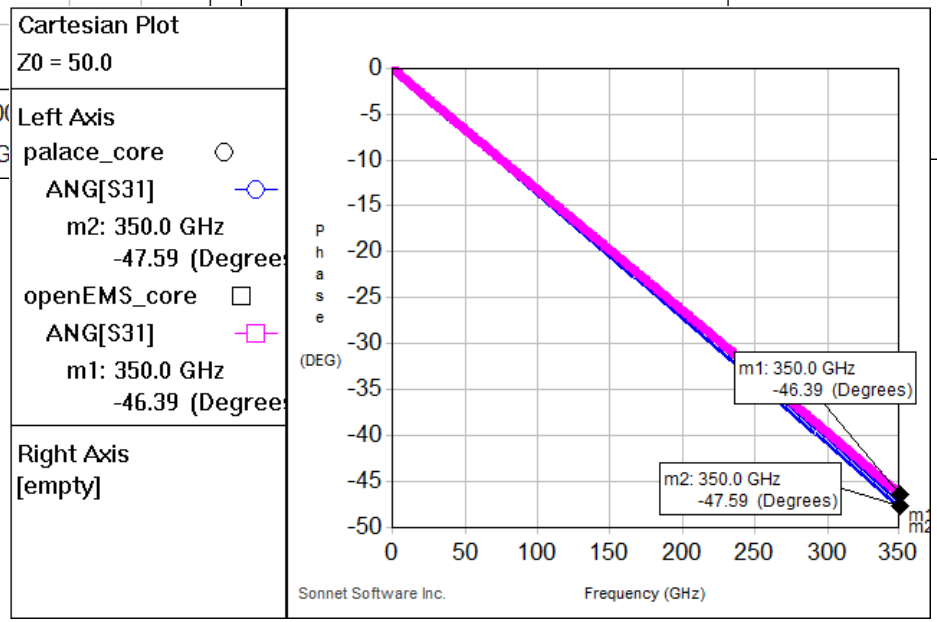
```
============================================================
Initialization                0.064       0.077       0.069
  Mesh Preprocessing          0.268       0.268       0.268
Operator Construction       168.671     218.541     215.407
Linear Solve                 17.117      17.764      17.588
  Setup                      21.977      21.979      21.978
  Preconditioner            128.645     129.081     128.856
  Coarse Solve               18.971      19.632      19.213
PROM Construction             0.620       0.632       0.622
PROM Solve                    2.218       2.478       2.366
Estimation                    0.207       0.231       0.224
  Construction                1.068       1.069       1.068
  Solve                      88.506      88.519      88.511
Postprocessing               20.949      70.828      24.180
Disk IO                       0.200       0.203       0.203
------------------------------------------------------------
Total                       520.594     520.594     520.594

Number of ports:  4
Created combined S-parameter file for  4 ports, filename:  /mnt/pub/IHP/Projekte
/OPDK_2024/Palace_EM/testcases/mpa_core/palace_model/palace_core_data/output/pal
ace_core/palace_core.s4p
```


```
  "GitTag": "v0.14.0-31-g07c452f1",
  "LinearSolver": {
    "TotalIts": 1248,
    "TotalSolves": 26
  },
  "Problem": {
    "DegreesOfFreedom": 221550,
    "MPISize": 16,
    "MeshElements": 32103,
    "MultigridDegreesOfFreedom": [
      42802,
      221550
    ],
    "OpenMPThreads": 1
  }
```

Compare results to openEMS



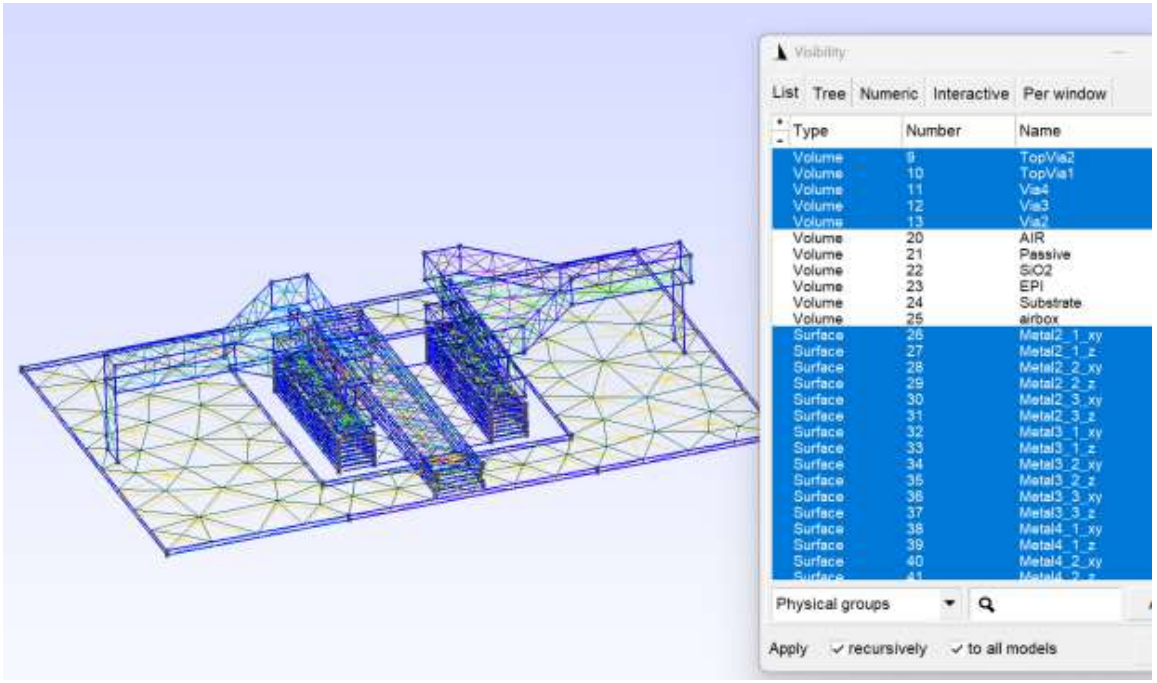




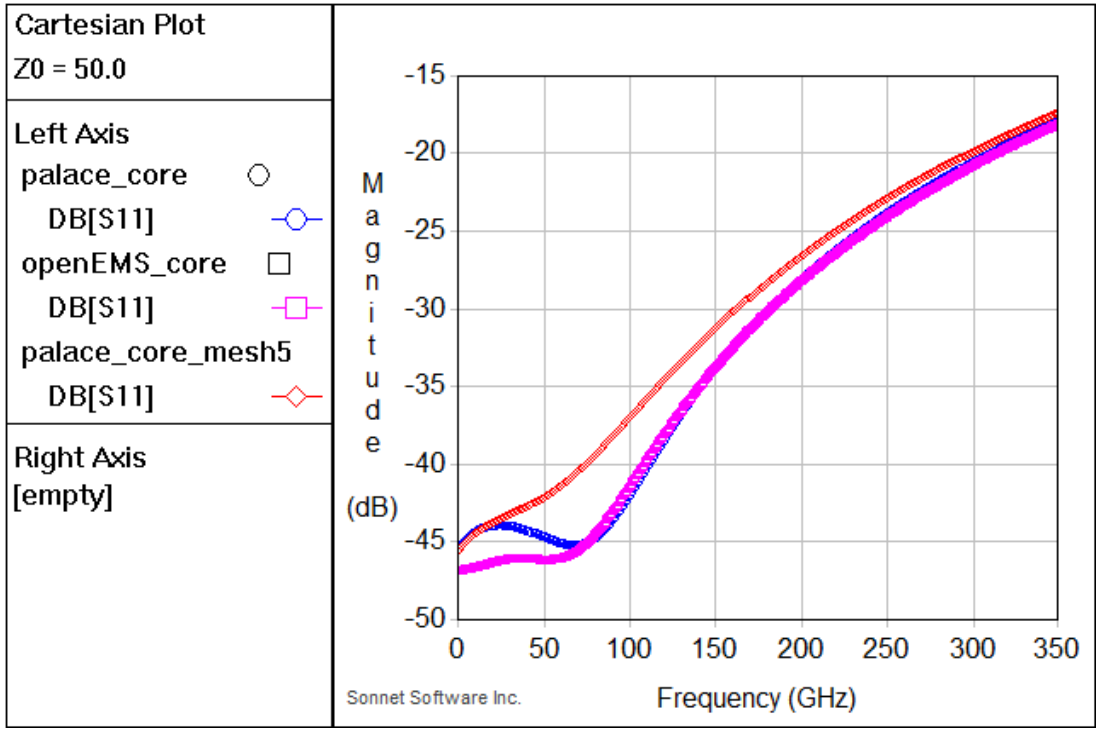
New run with coarser mesh 2µm → 5µm

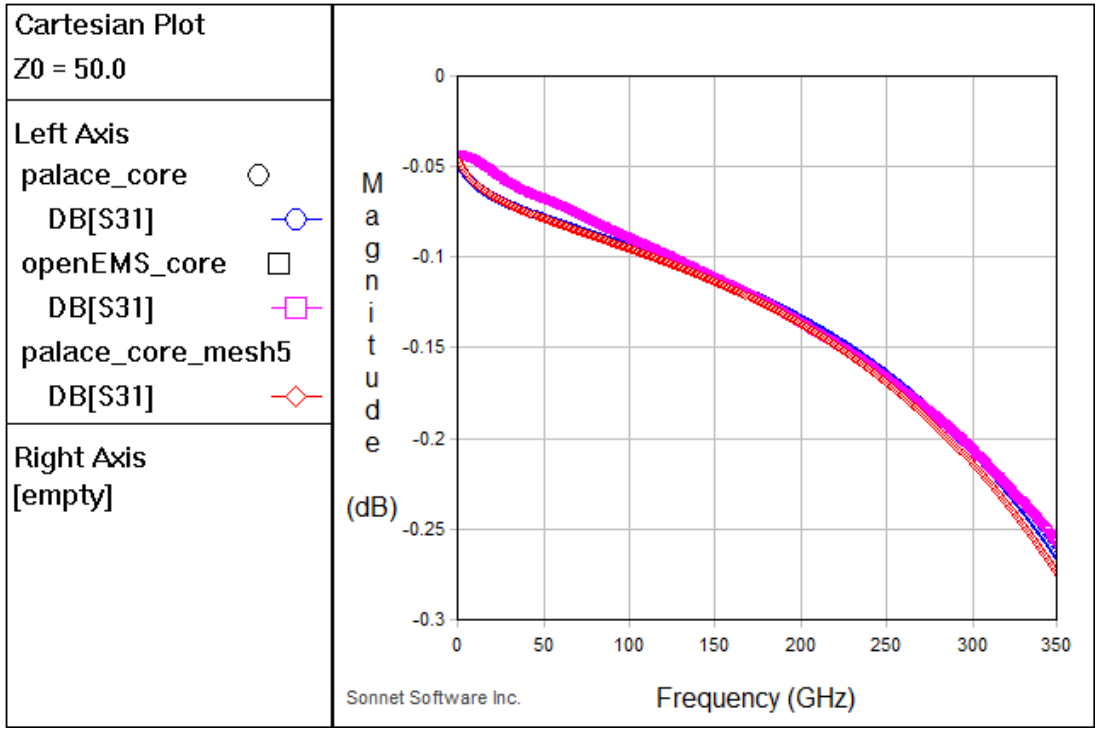
```
settings['refined_cellsize'] = 5
```

- Mesh file size:
  1.7 MB → 0.8 MB
- DegreesOfFreedom:
  221550 → 109664
- Total time:
  520 s → 268 s
- Comparison openEMS total:
  ~ 2800 s



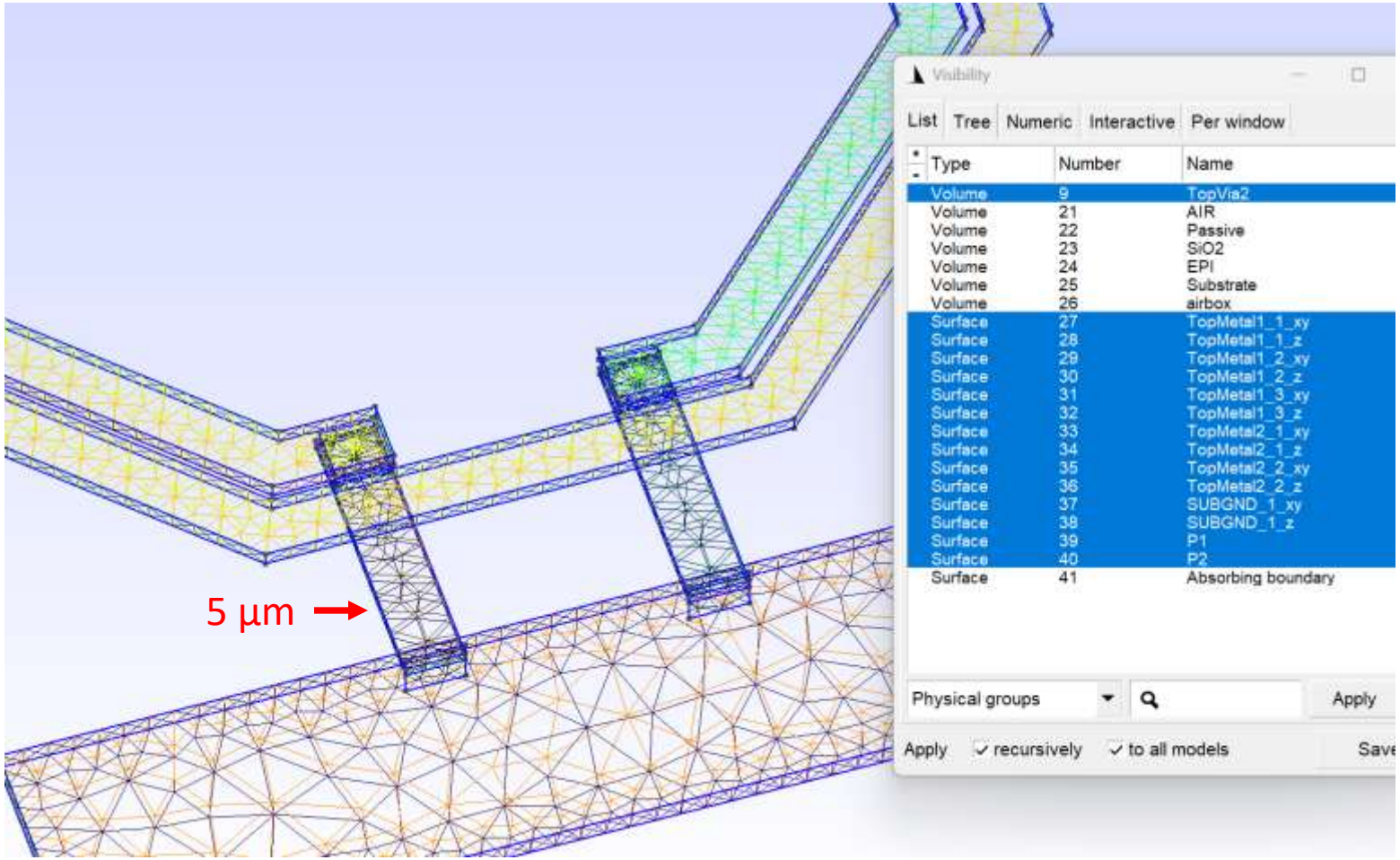
New run with coarser mesh 2µm → 5µm





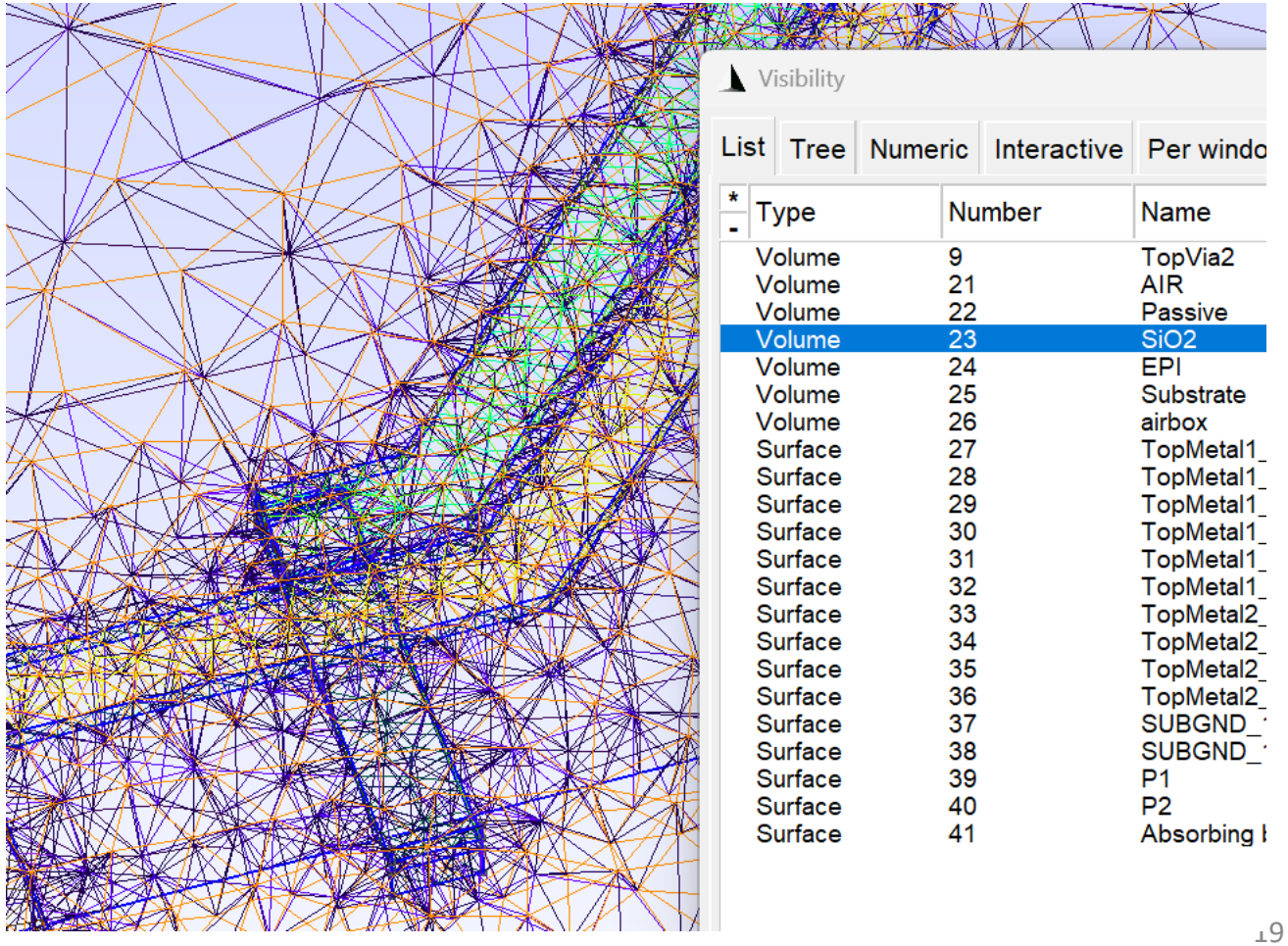
Example Octagon Inductor

- refined_cellsize = 5
- No staircasing issue, regardless of mesh size



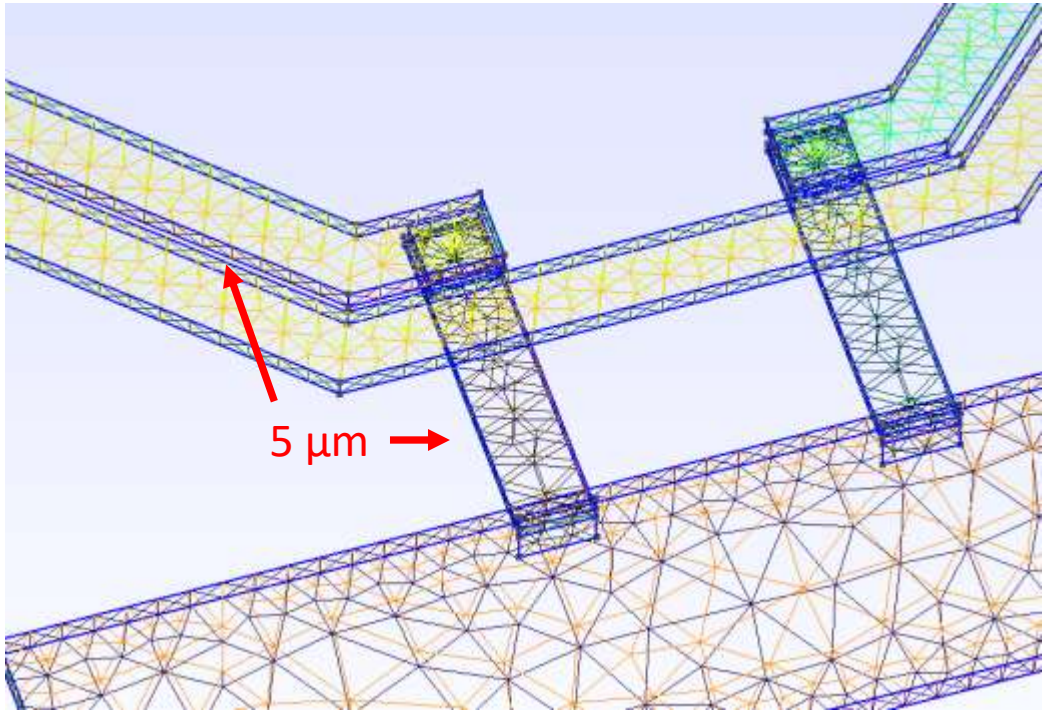
Example Octagon Inductor

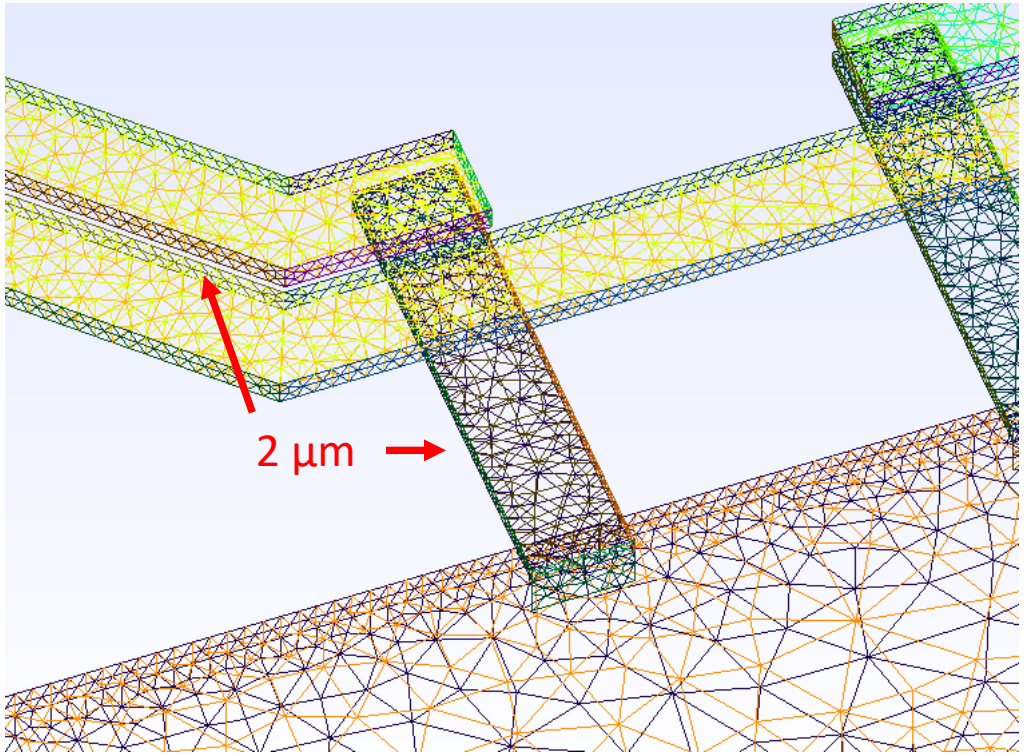
- Mesh is created along the metal traces
- Mesh size must be fine enough to resolve field details in gap between traces
- Total **444 s** for full sweep of 2-port model at 5 µm



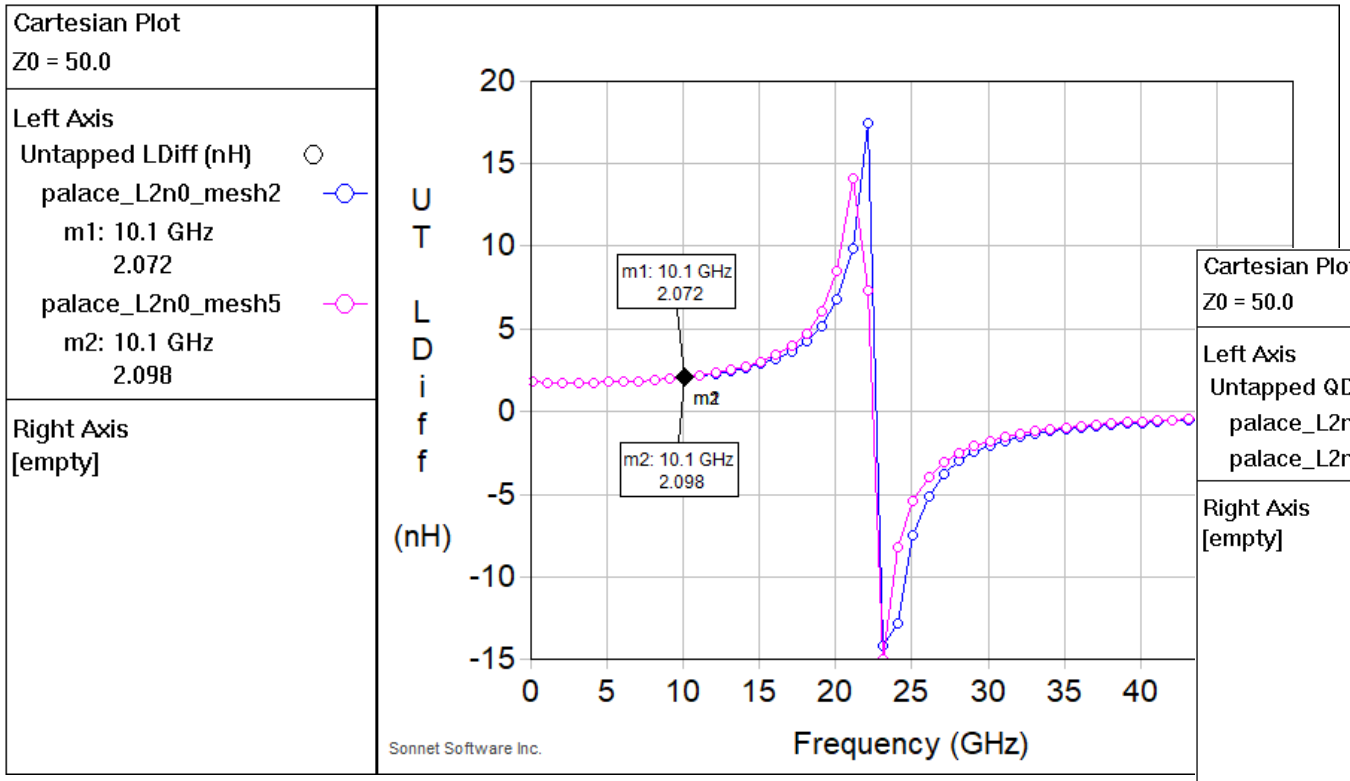
Octagon refined_cellsize 5 µm vs. 2µm

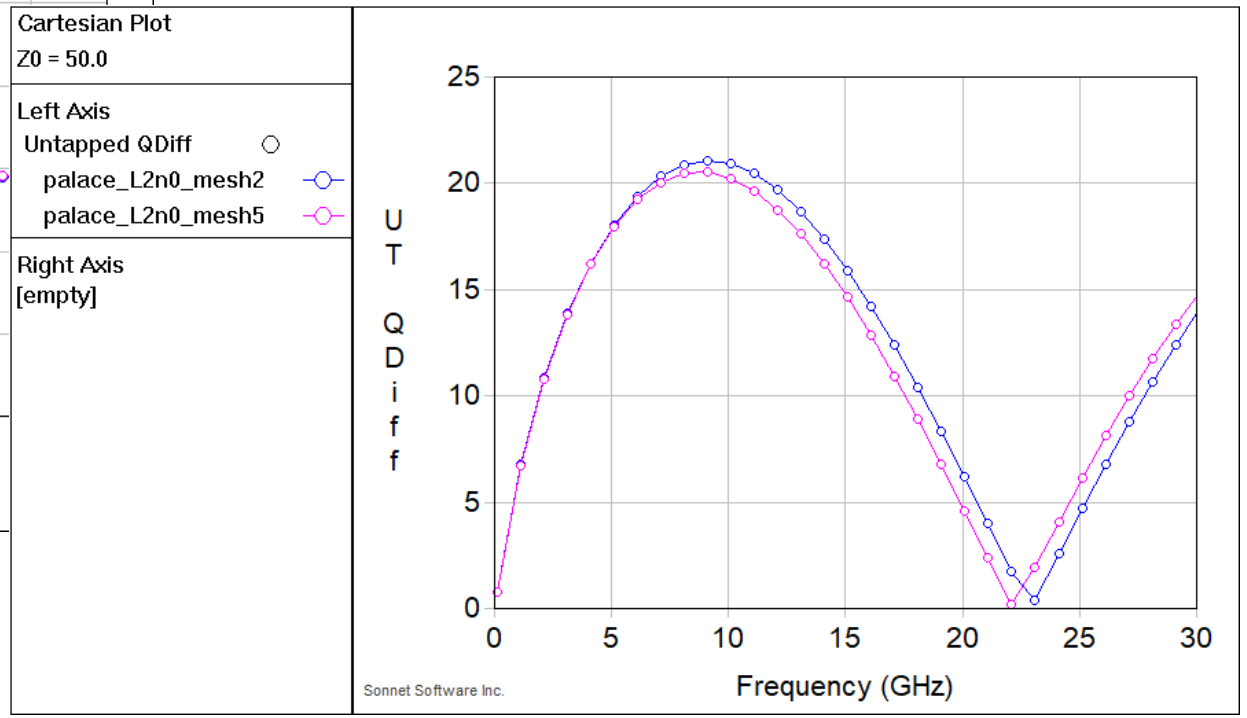
- **2.9 MB** → **9 MB** mesh file size
- **444 s** → **1193 s** time for full S-params





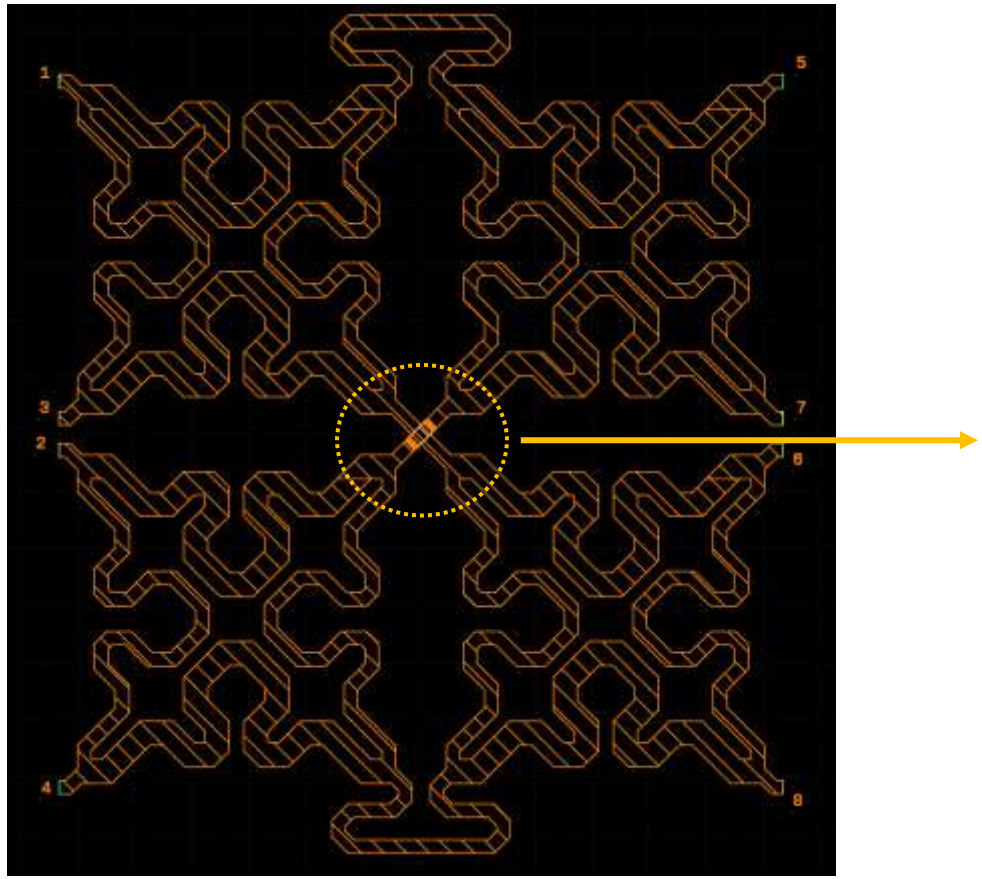
Octagon refined_cellsize 5 µm vs. 2µm

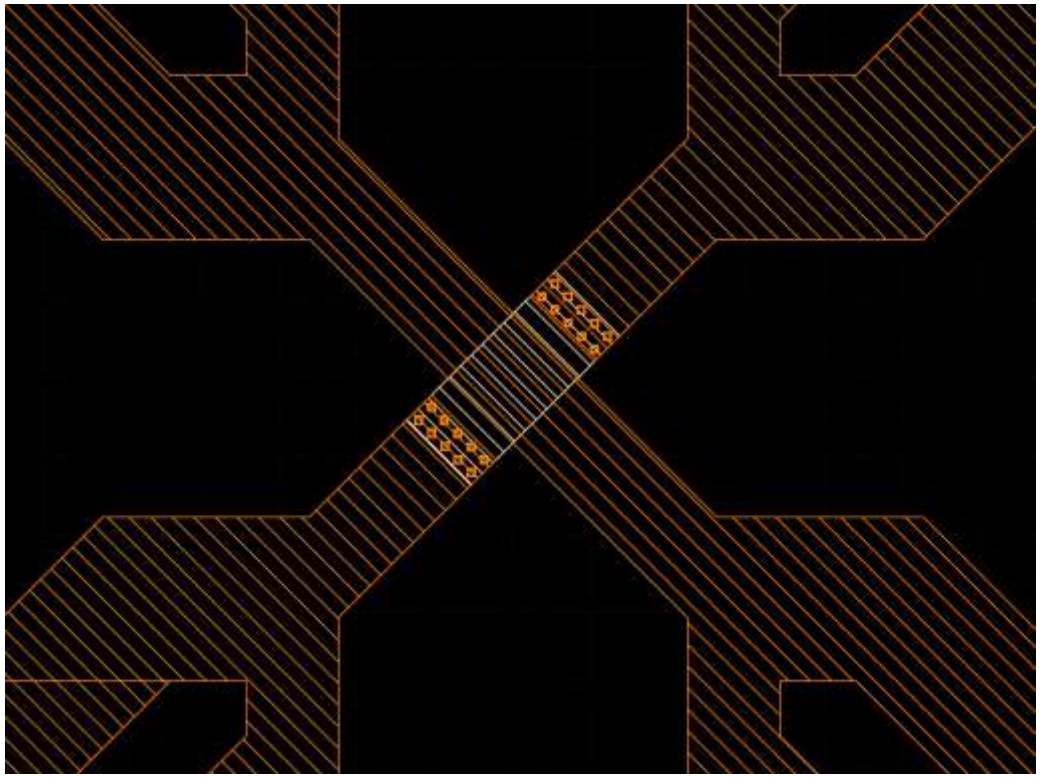


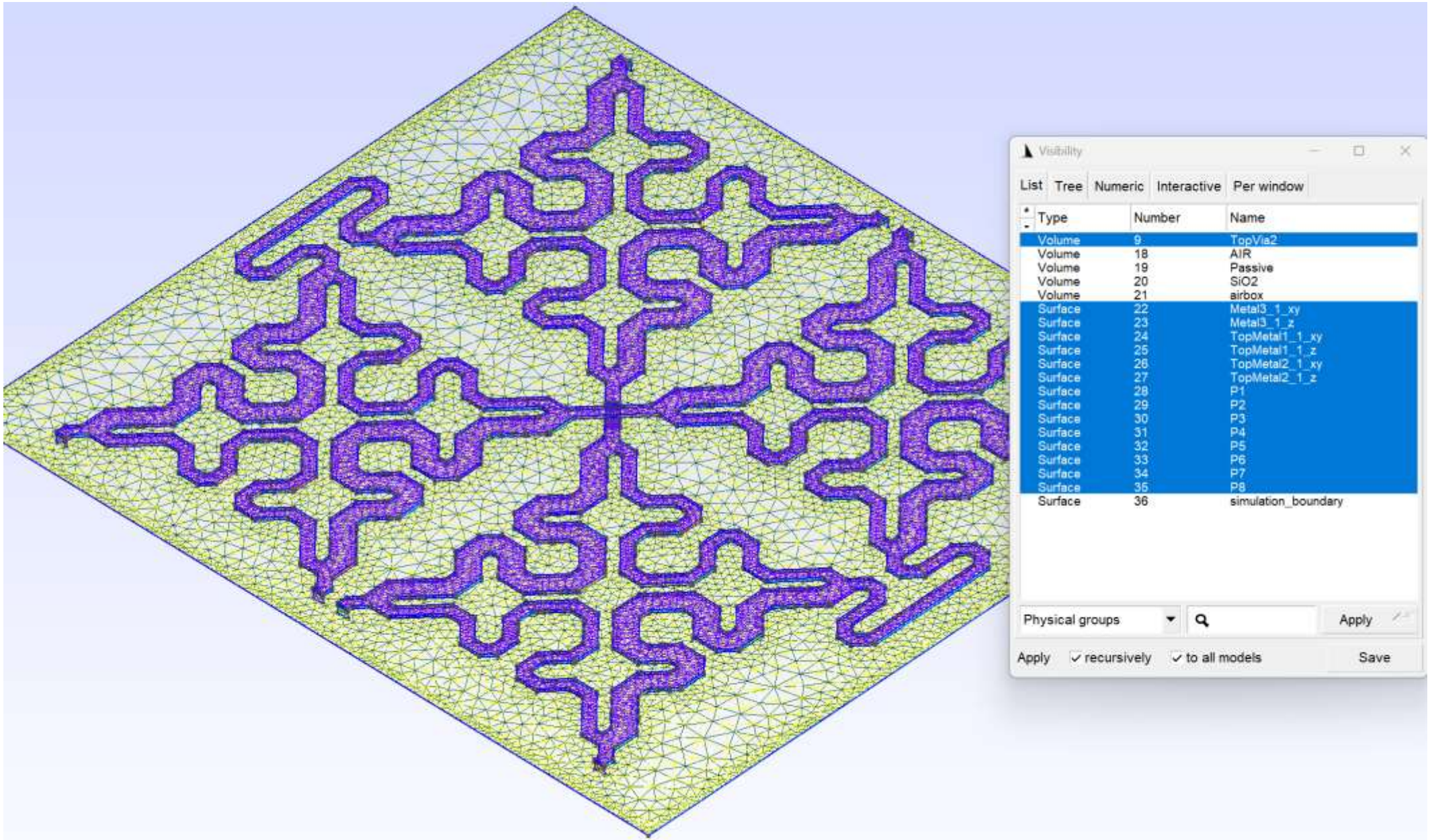


Butler Matrix by Ardavan Rahimian, TO 07/2025

- Total 8 ports, design frequency 93 GHz



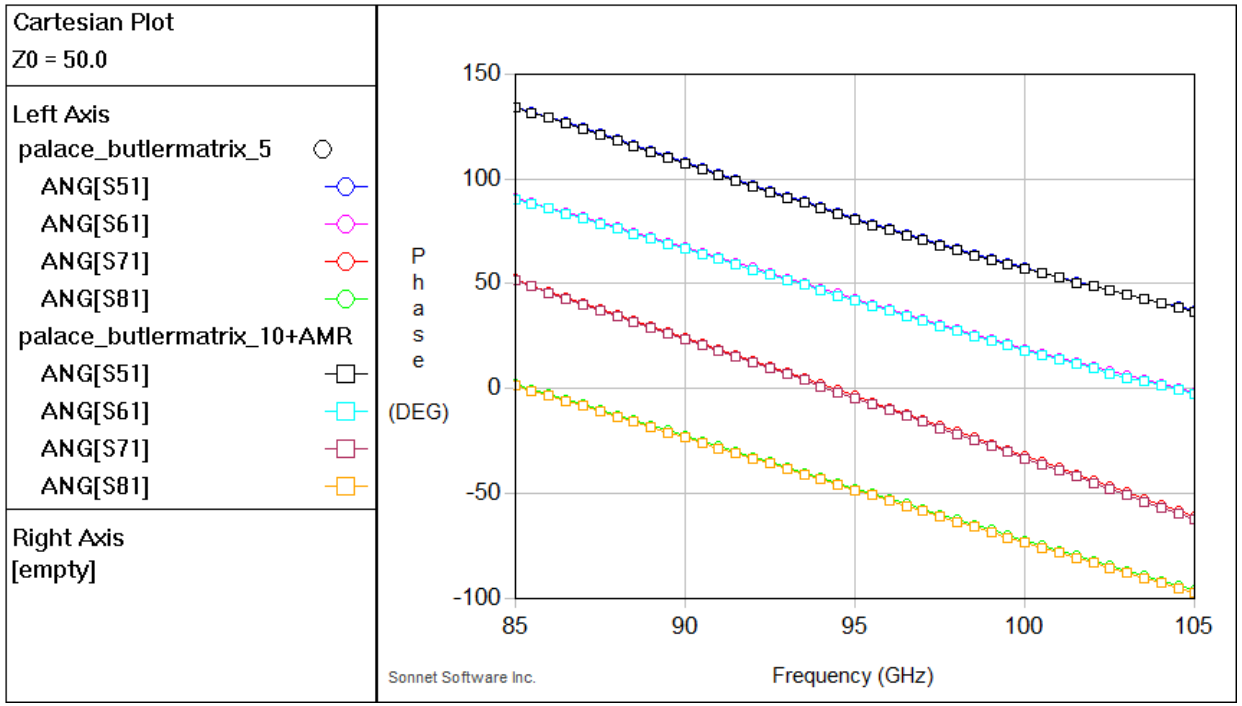


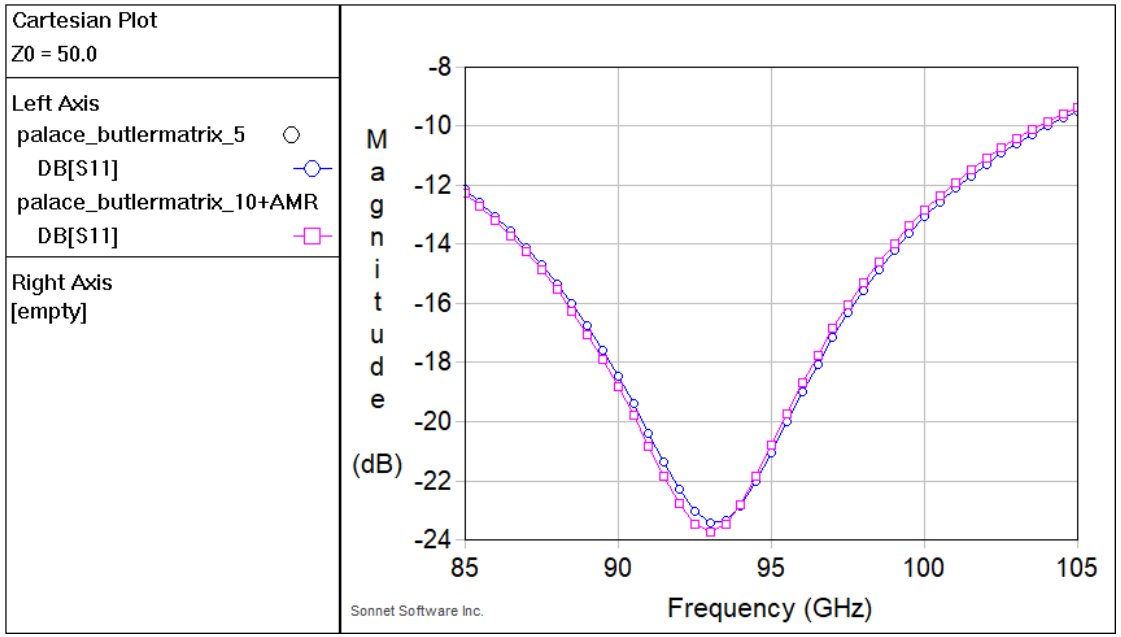

Visibility
List
Tree
Numeric
Interactive
Per window
Type
Number
Name
Volume 9 TopVia2
Volume 18 AIR
Volume 19 Passive
Volume 20 SiO2
Volume 21 airbox
Surface 22 Metal3_1_xy
Surface 23 Metal3_1_z
Surface 24 TopMetal1_1_xy
Surface 25 TopMetal1_1_z
Surface 26 TopMetal2_1_xy
Surface 27 TopMetal2_1_z
Surface 28 P1
Surface 29 P2
Surface 30 P3
Surface 31 P4
Surface 32 P5
Surface 33 P6
Surface 34 P7
Surface 35 P8
Surface 36 simulation_boundary
Physical groups
Apply
Apply
recursively
to all models
Save

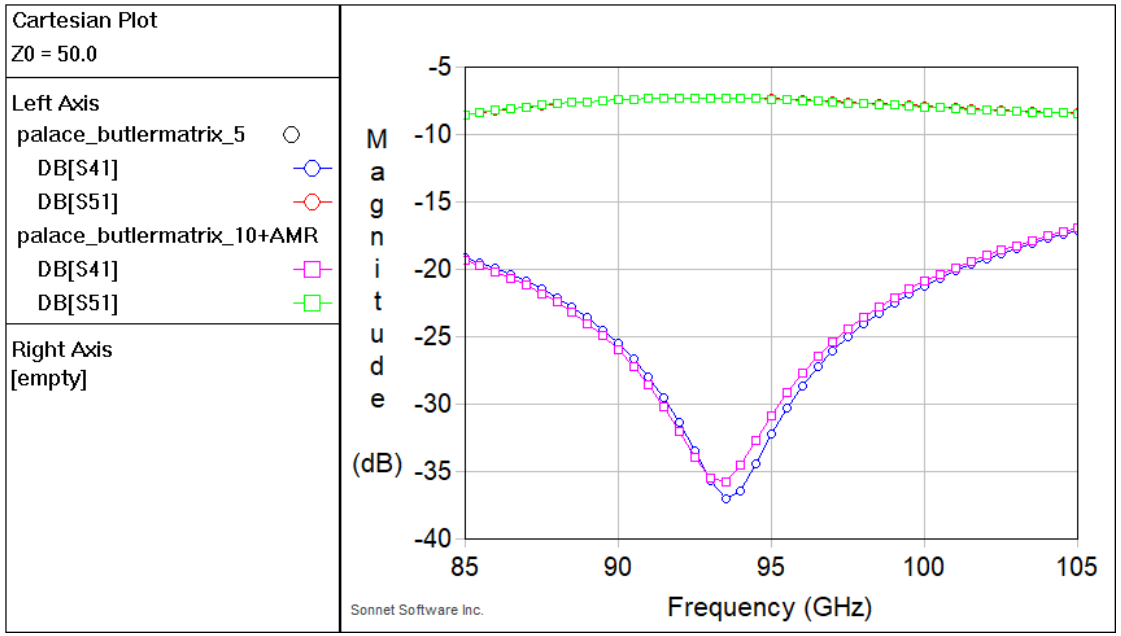
Model size

- Mesh file size at 5µm target size: **21 MB,** RAM required: **~ 45 GB**
- Simulation frequency range: **85 – 105 GHz**, step 0.5 GHz
- Only port 1 excitation, we get S11, S21, S31, S41, S51, S61, S71, S81
- Simulation time using initial mesh **@ 5µm**:
  23 min for adaptive sweep using 6 frequencies, 1 port
  DOF = 2517230
- Simulation time **@ 10µm + 1 adaptive refinement**:
  31 min for adaptive sweep using 6 frequencies, 1 port
  DOF = 1230198, 1652802

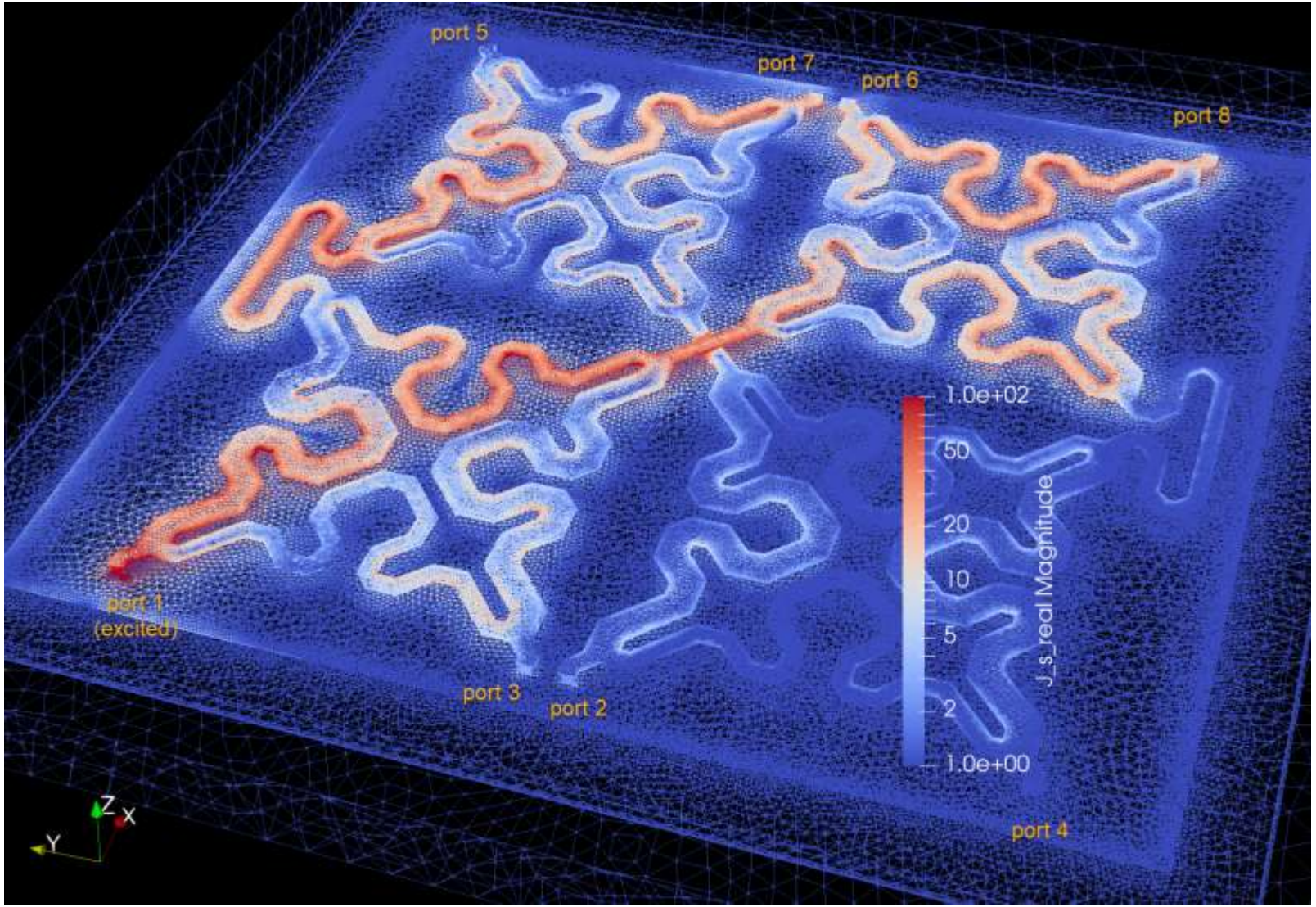
Mesh strategies compared



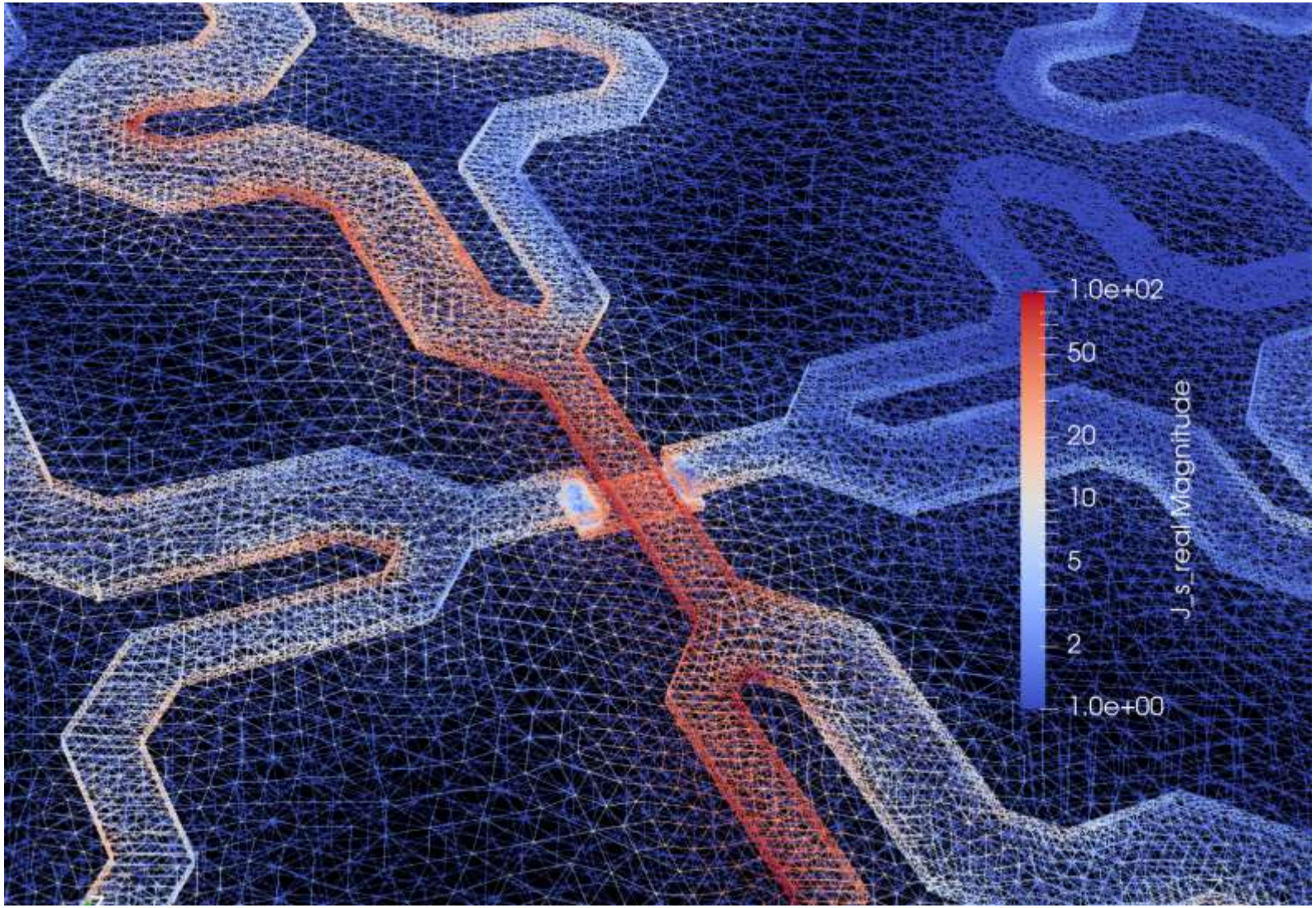




Current density 93 GHz, feed at port 1

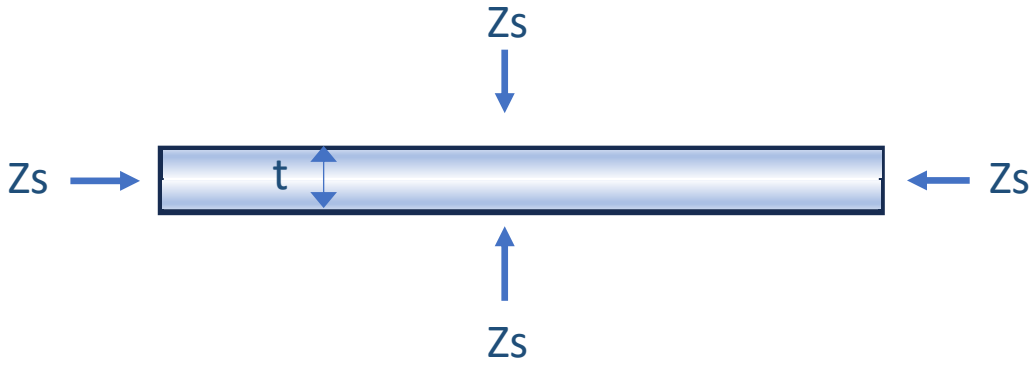


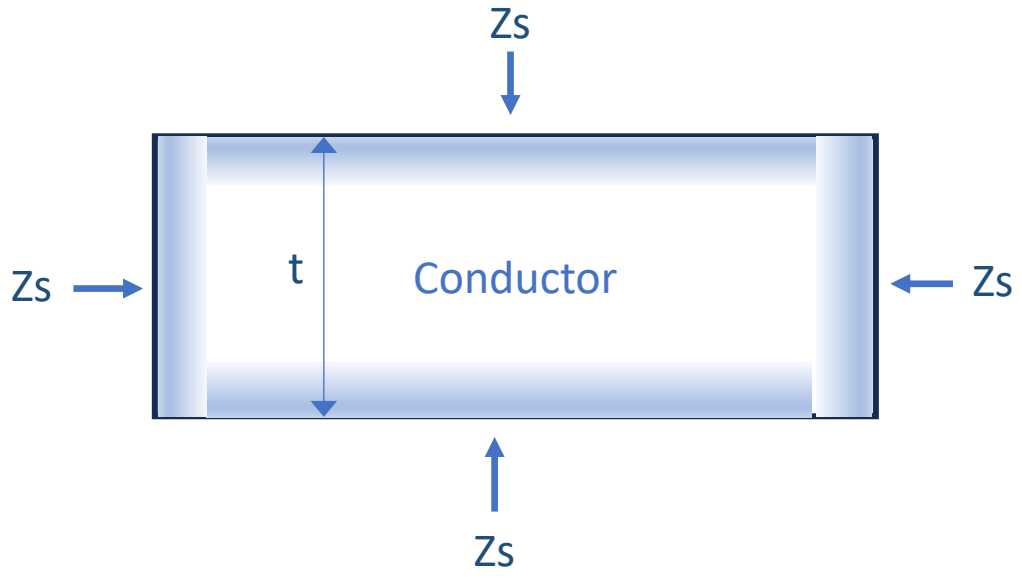
Current density 93 GHz, crossover detail



Metal loss model: surface impedance

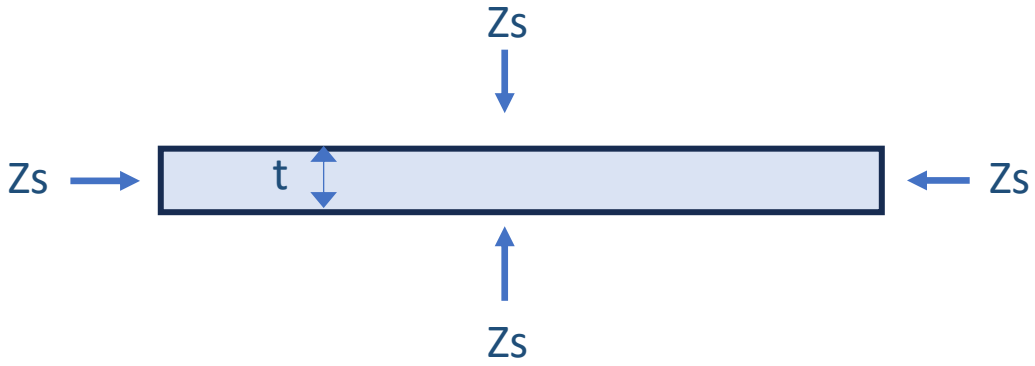
- Metals modelled as hollow bodies with surface impedance
- Surface impedance in Palace from conductivity and thickness
- In skin effect regime at high freq, current flows on all sides (edges)

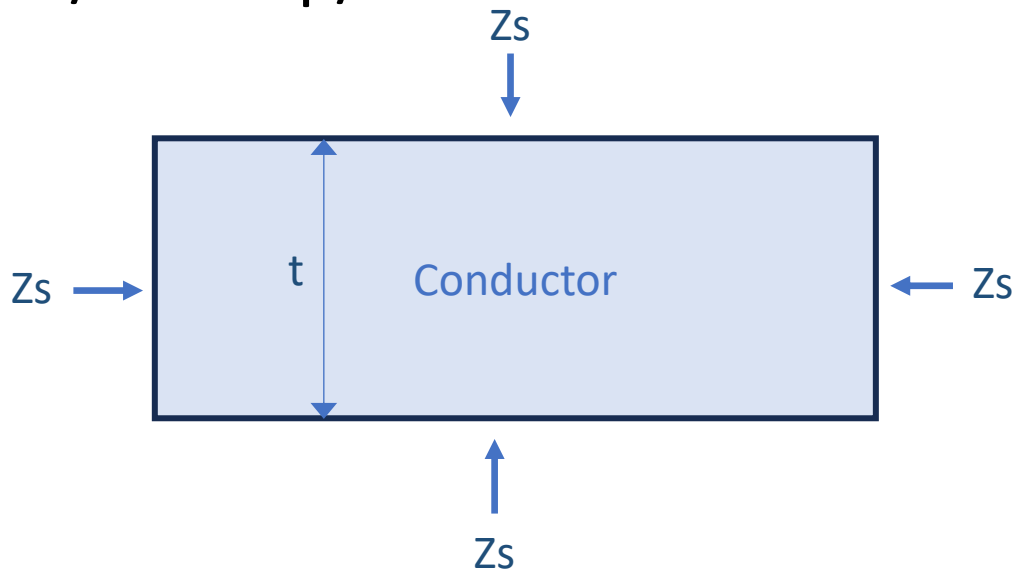




Metal loss model: surface impedance

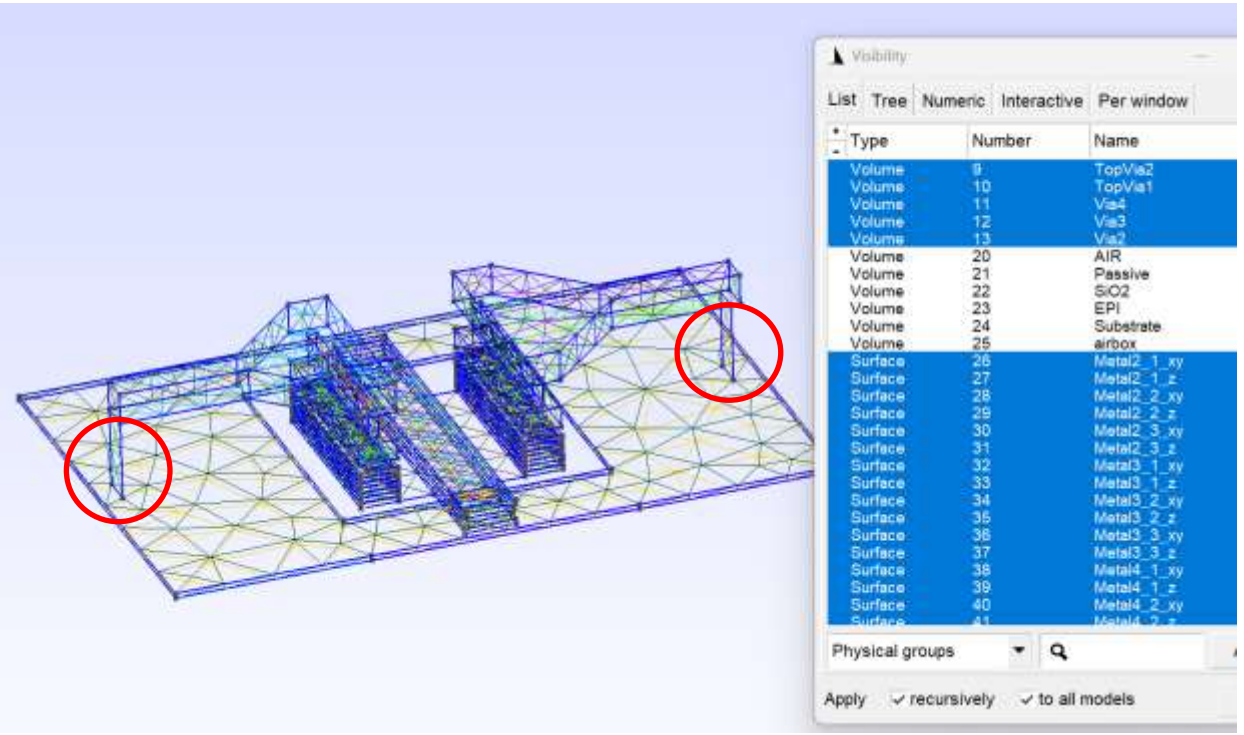
- At lower frequency, physical cross section is the limit, not skin depth
- Palace Zs formula assumes that we have 2 sides of thickness/2 each
- Using same Zs on side walls over-estimate the total cross section!
- In present code, side walls set to 1/3 of top/bottom thickness





Other issues

- Palace can not solve at 0 Hz DC, so we use a low frequency value.
- Lumped ports have finite size -> this adds some small extra length



Software versions

- Palace version used for this document: v0.14
- Palace installation method used:
  container for Apptainer environment, run on Ubuntu 24.04.02 LTS
- Simulation host: AMD Ryzen 9-7950X (16 cores) with 128 GB RAM
- Code base: Python scripts as of October 01, 2025