OpenEMS Simulation Creator
FreeCAD + openEMS

Microwave simulations

by Lubomir Jagos

May 2024

Installation

- openEMS
  - https://www.openems.de/

- FreeCAD
  - Use version 0.20 which is tested to work with this as this GUI application was developed with this one
  - During writing this was tried 0.21 version but there is ERROR with STL export when parts are moved (is fixed in 0.22 which will be released soon)

- OpenEMS Simulation Creator
  - GUI addon to FreeCAD to generates openEMS scripts
  - https://github.com/LubomirJagos/FreeCAD-OpenEMS-Export/releases
  - Examples at https://github.com/LubomirJagos/FreeCAD-OpenEMS-Export-Plugin-Documentation

- Octave
  - https://octave.org/
  - Set directory to openEMS/matlab to be able run scripts

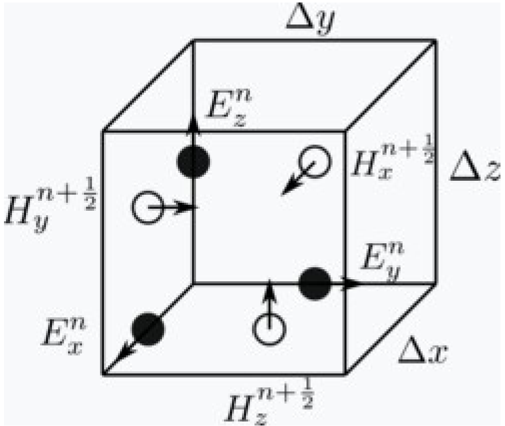
- Python: install version 3.11.6
  - https://www.python.org/downloads/release/python-3116/
  - Install wheels for CSXCAD and openEMS included in downloaded zip folder
  - Install pylab
    - pip install matplotlib
  - It’s good to have some python IDE ie. Spyder
    - https://www.spyder-ide.org/
    - Set in Preferencies python interpreter to installed 3.11.6 version, don’t use spyder included interpreter

- Paraview – for result fields evaluation and doing nice images or animations
  - https://www.paraview.org/download/

FDTD algorithm

- electric and magnetic field is evaluated over rectangular cells (simulation grid)
- E and H field are complex 3D vectors
  - When complex vector is decomposed into x,y,z parts these are also complex (this is just to realize when simulation results are used to calculate something else in scripts)
  - Units for E field are V/m
  - Units for H field are A/m
  - Scalar voltage V is result of curve integral in space between 2 points (voltage probe is in openEMS line)
  - Scalar current A is result of surface integral (at least in openEMS current probe is some surface)

- slower than MoM but can compute structures with dielectric material
- calculates fields in time therefore it evaluate on some frequency band
  - FEM method evaluate structrues on single one frequency for some band evaluation it must be calculate for each frequency separately (but FEM is faster than FDTD)



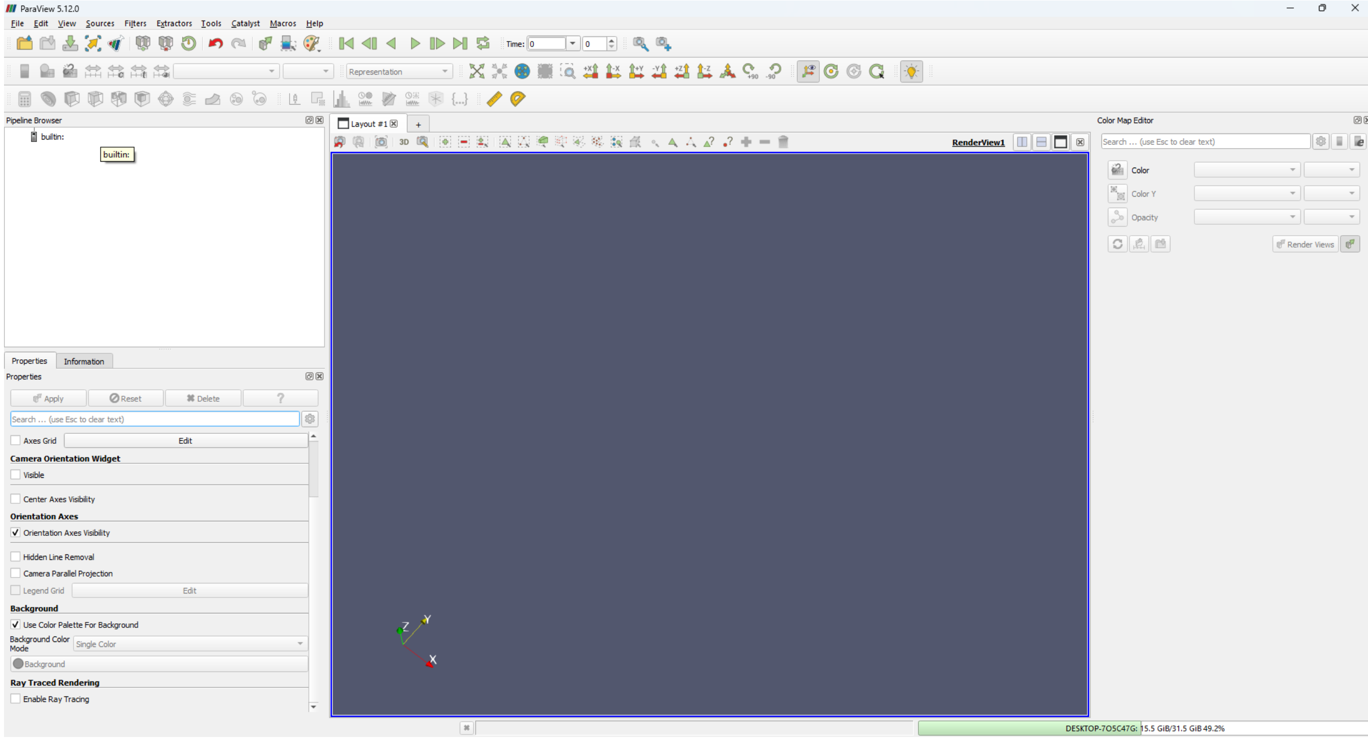
FreeCAD

- open-source 3D CAD
- actively developed
- using python scripting, can be easily extended using python scripts macros
- also when Qt used then GUI app can be added

- good cooperation with KiCAD
- fully scriptable using python
- can run in command line without gui – could be use for parametric simulation scripted in python

Paraview

- general software to visualise scientific data
- can display broad formats
- can be used for EM simulation results (openEMS), fluid simulations results (openFOAM) or else



openEMS

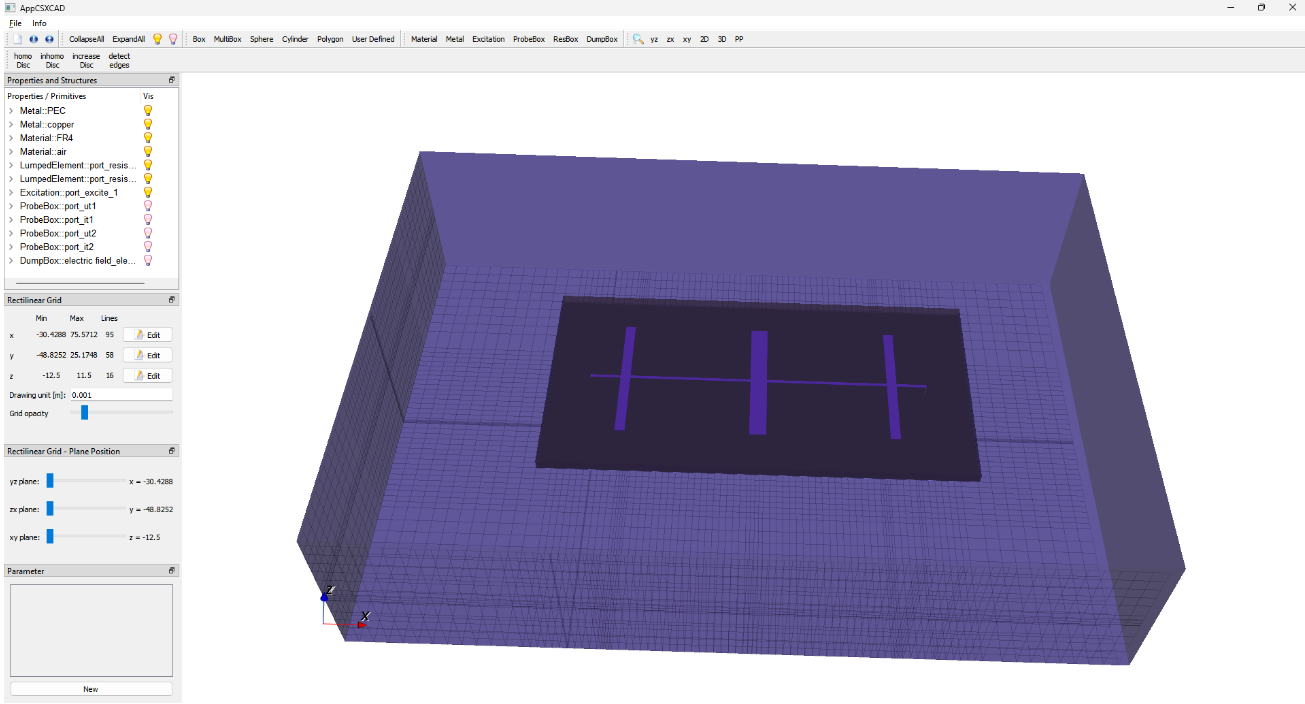
- open-source FDTD EM solver
- developed by Thorsten Liebig
- started in 2010
- there is quite active "community" on github forum
  - https://github.com/thliebig/openEMS-Project/discussions

- available scripting:
  - Octave interface (GUI addon was developed with this interface, this is for sure running right)
    - Generates xml structure definition with simulation settings
  - Python interface (missing some high-level function to create different ports, but these could be easily modelled in FreeCAD)
    - generates xml just for structure
    - uses cython C++ function call directly openEMS core

- FreeCAD GUI addon is capable to generate python and octave script

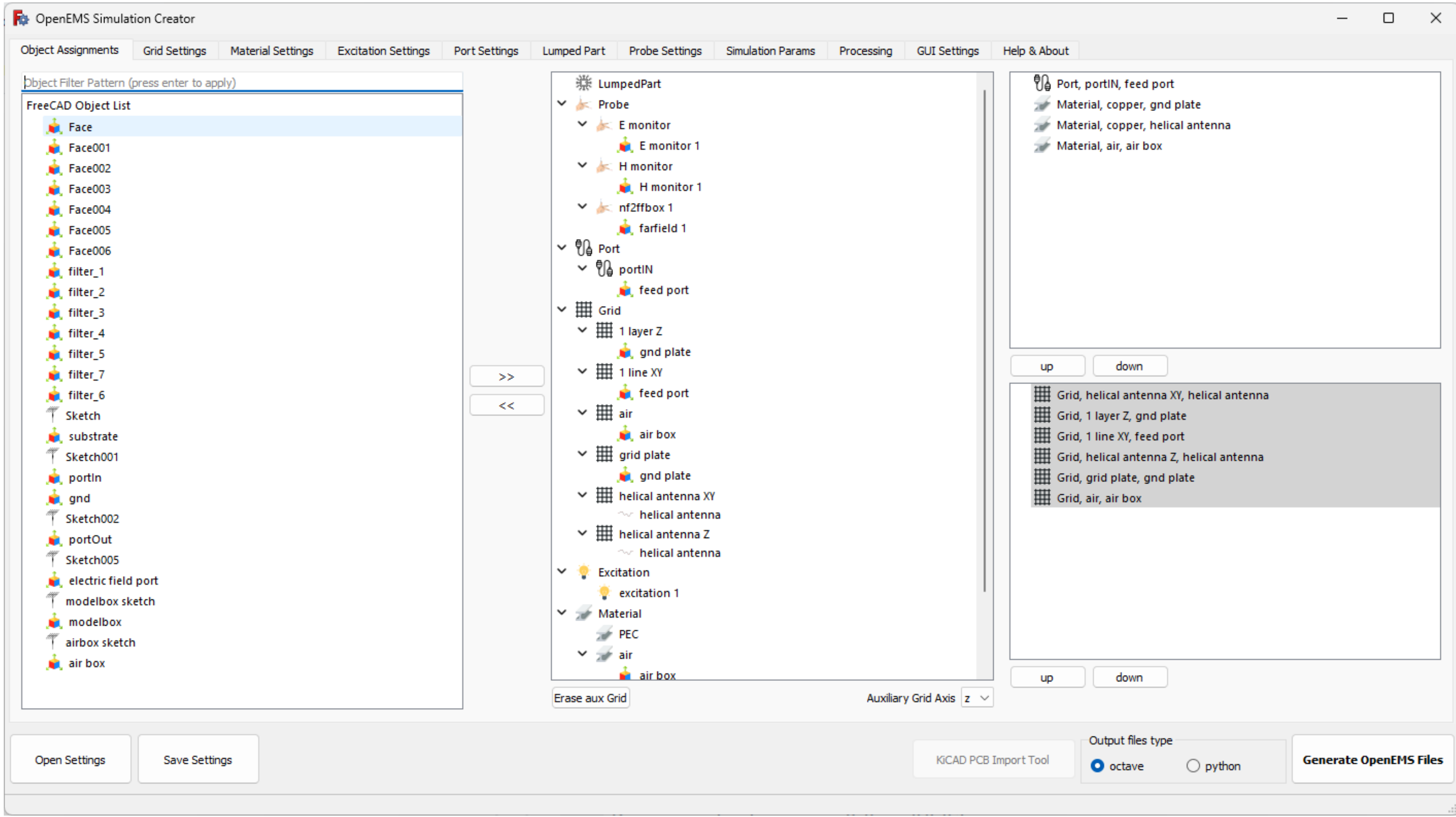
CSXCAD

- part of openEMS
- CAD from Thorsten to visualise structure from XML
- define XML file to describe structure and simulation
- easy
- few geometry primitives (box, cylinder, polygon, ...)
- can be used to display structure and also to modify
- it's possible to create whole simulation in CSXCAD but probably it's easier use scripting approach



OpenEMS Simulation Creator

- GUI addon for FreeCAD or Blender to create openEMS script, simple IDE similar to CST Studio or Ansys HFSS



Simulation setup

- Need to set:
  - Excitation for ports
    - Sinusodial – one frequency
    - Gaussian – range of frequncies
    - User defined E field function – equation is using fparser functions http://warp.povusers.org/FunctionParser/fparser.html#functionsyntax
  - Simulation input port
  - Grid:
    - Rectangular
    - Cylindircal – this is unique feature of openEMS could be used to simulate bend patch antennas or MRI coils
  - Materials:
    - PEC – perfect electric conductor
    - User defined - permitivity, permeability, loss
    - Conducting sheet – same as PEC but on computational C++ code there is some special faster implementation
    - Lorentz material – NOT IMPLEMENTED in GUI but there is scripting interface so could be added into script
    - Debeye material – NOT IMPLEMENTED in GUI but there is scripting interface so could be added into script
  - Ports
    - Passive – to sense E, H fields to be used to evaluate or calculate ie. S params or whatever needed
    - Active – these are wrapping grid nodes where is injected values of excitation, there is excitation common for all ports there could be defined just different time delay for each active port, DIFFERENT EXCITATION FOR PORTS NOT IMPLEMENTED IN openEMS
  - E or H probes
    - Raw values of E or H could be dumped into binary .h5 file
  - Simulation parameters
    - End condition – energy level when in simulation decreas under this level simulation is stopped
    - Maximum number of timesteps – when reached simulation is aborted

Notes to simulation – user experience

- simulation is ideal model
- reality and simulation could differs
  - if too much then model needs to include more real world features
- sometimes results seems wrong but they are right – expectation are wrong and manual study should be done again
- GRID is the MOST IMPORTANT to set
  - it defines calculation nodes

- FR4 is unpredictable material, it's epoxy resin with no exactly defined permitivity, it's like 4.2 – 4.7 no one knows

- don't mix octave and python files, there are little differencies in API like output file naming which can cause that scripts will not run
  - there will be exception thrown in output

- materials in openEMS have priority parameter in case of overlapping material with highest priority is used for node material definition

- if model is done right, slight change of grid shouldn't affect results

- S parameters are independent from excitation port amplitude, so if just S params are object of interest amplitude magnitude of excitation doesn't matter

- try to left lambda/4 space around model, this is common mistake

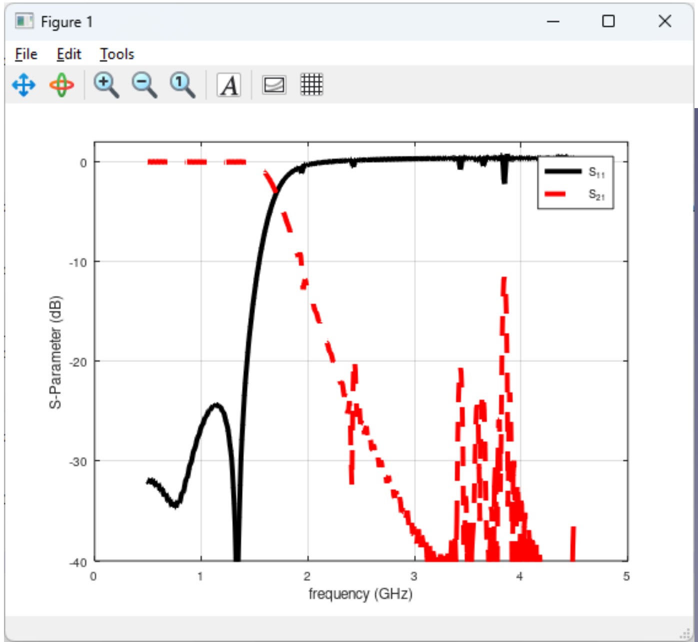
- MAKE SURE THERE ARE LINES CROSSING PORTS, common mistake

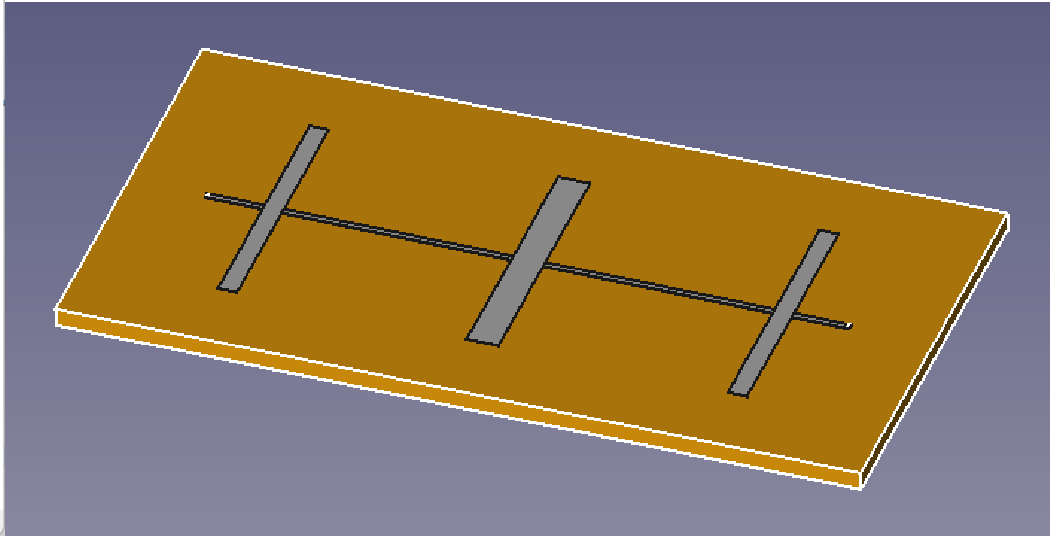
- TOO COARSE MESH can cause numerical instabilities (it's visible when visualise E-field it's values are over 1e6 V/m what is total nonsense)

- SPENT SOME TIME TO LEARN PYTHON doesn't need to be expert but IT WILL CHANGE YOUR LIFE (openEMS, GnuRadio, numpy, sympy, Blender, ...)

Low pass filter

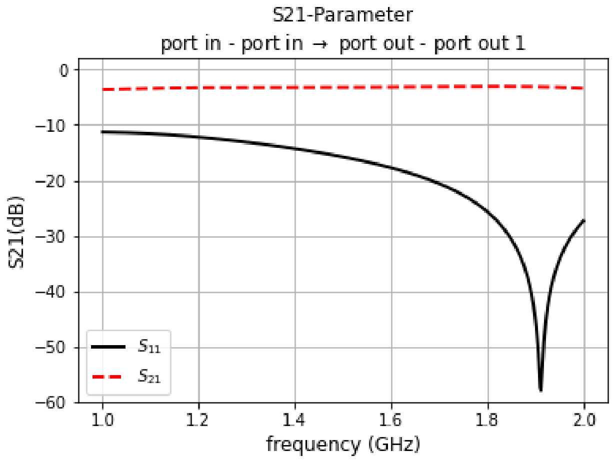
- example of planar low pass filter

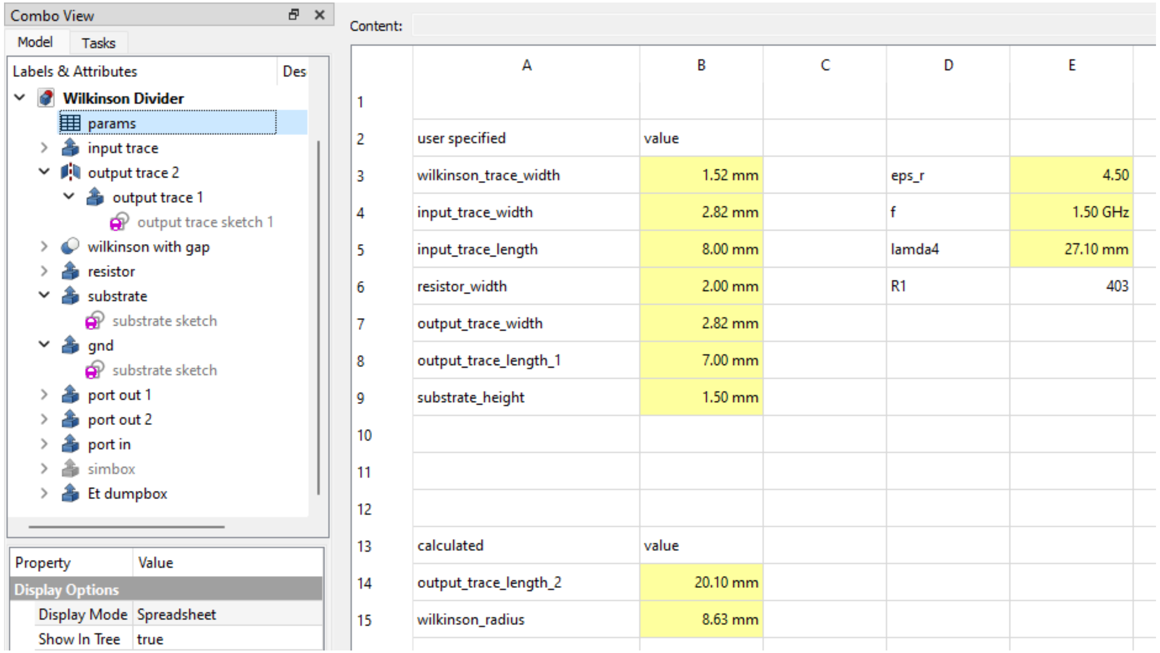


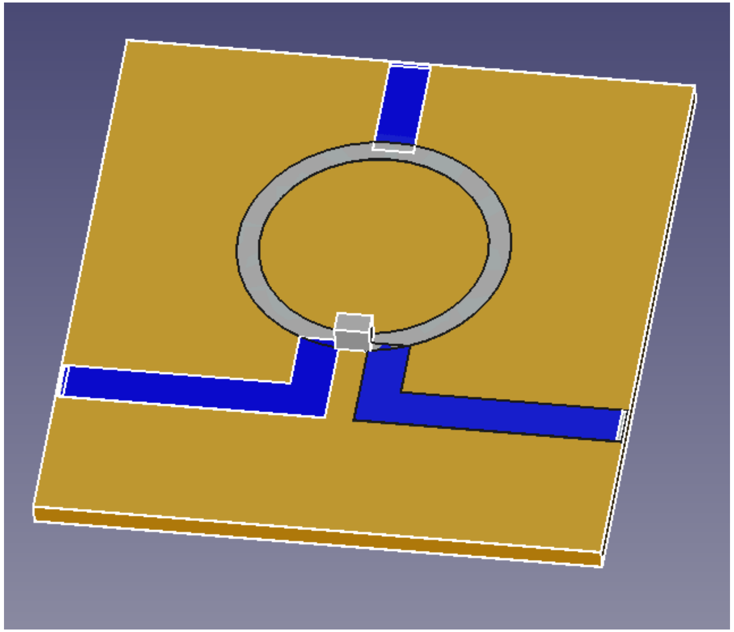


Wilkinson divider

- there is example of usage of parameters from table in FreeCAD

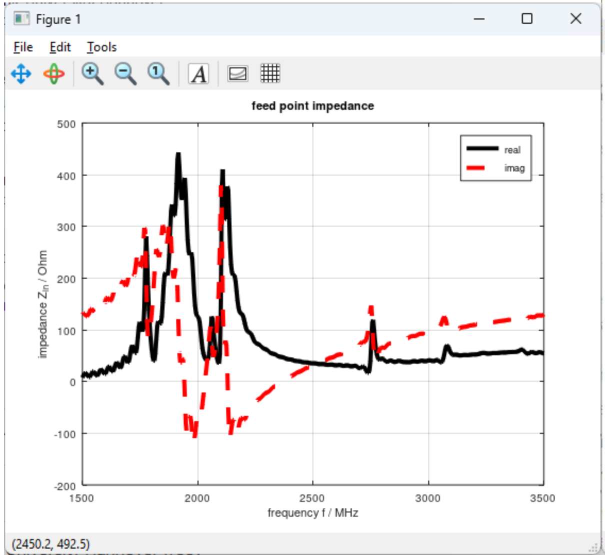


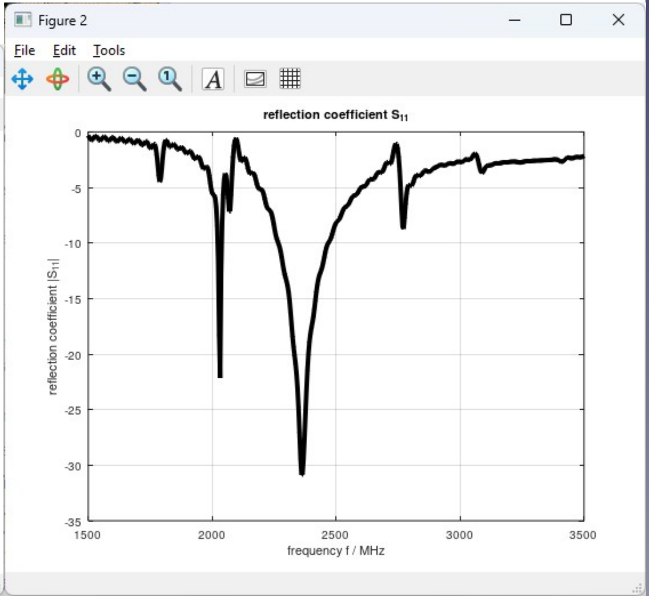


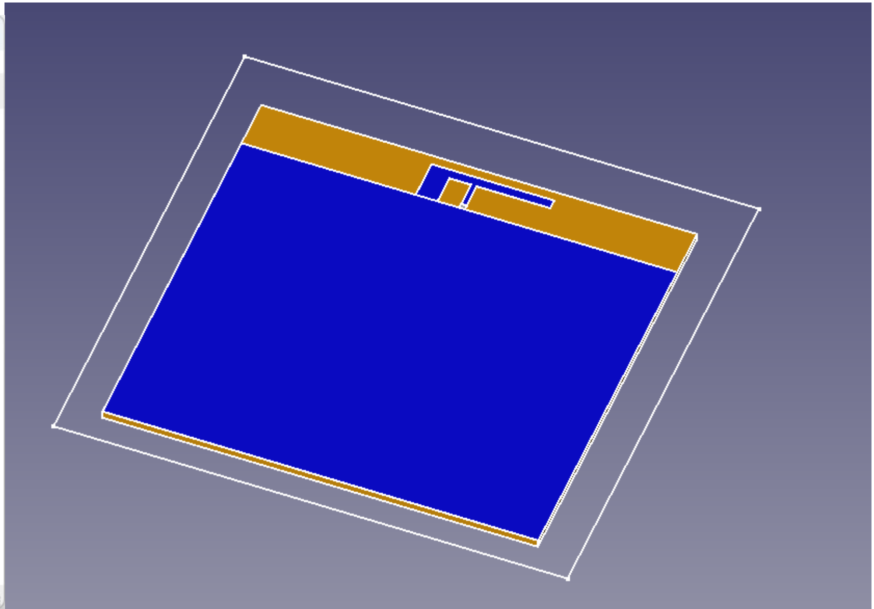


Inverted F antenna

- small antenna doesn't require bottom ground patch

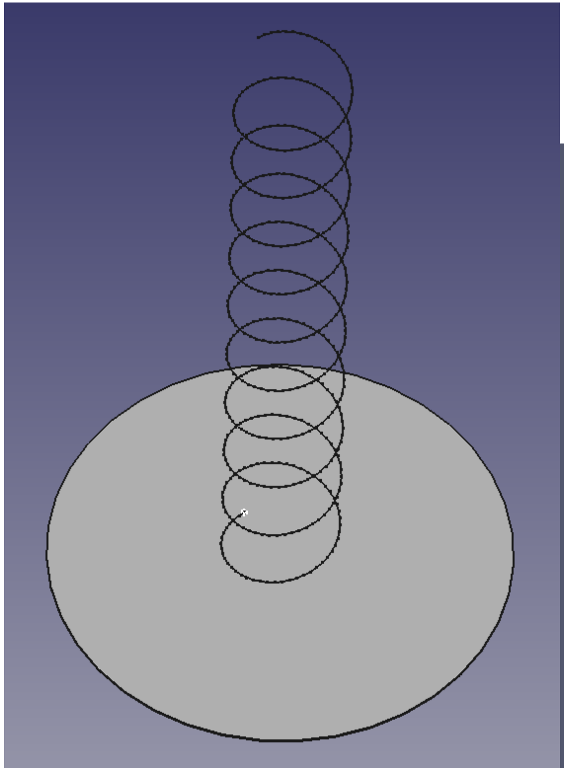


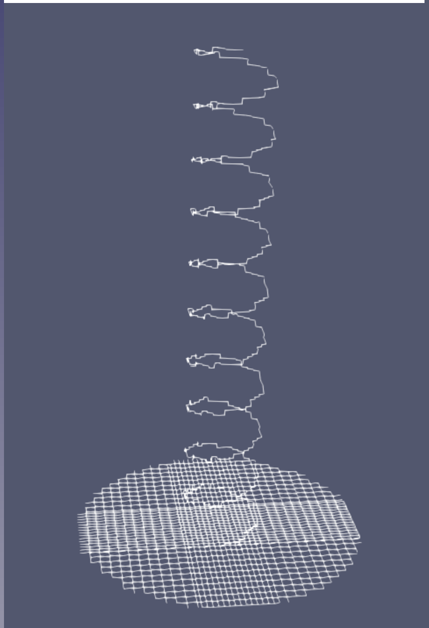


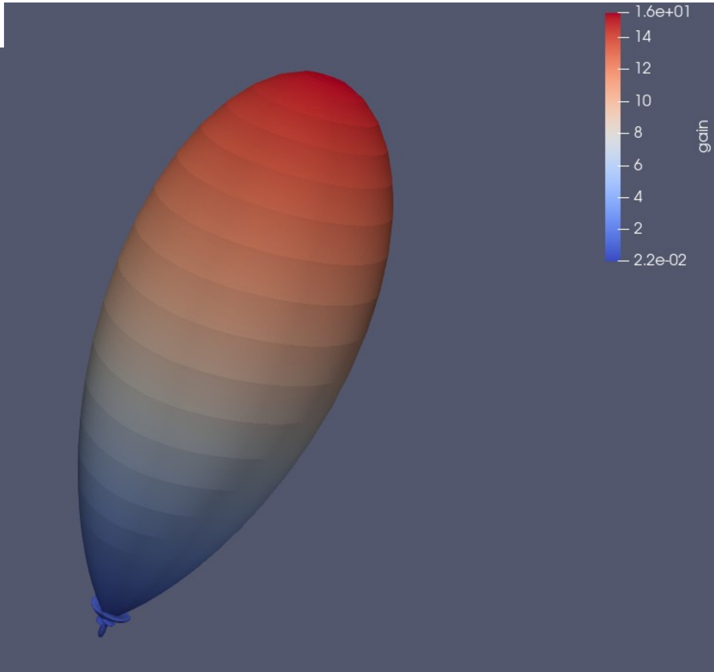


Helical antenna

- wire structure could be also simulated in openEMS but wires must aligned with grid
- contains example of far field diagram generate and evaluate
- for farfield NF2FF probe is used, there is openEMS application which calculate far field from near field
  - antenna phase center should be in middle of NF2FF box probe (but in example it's not)

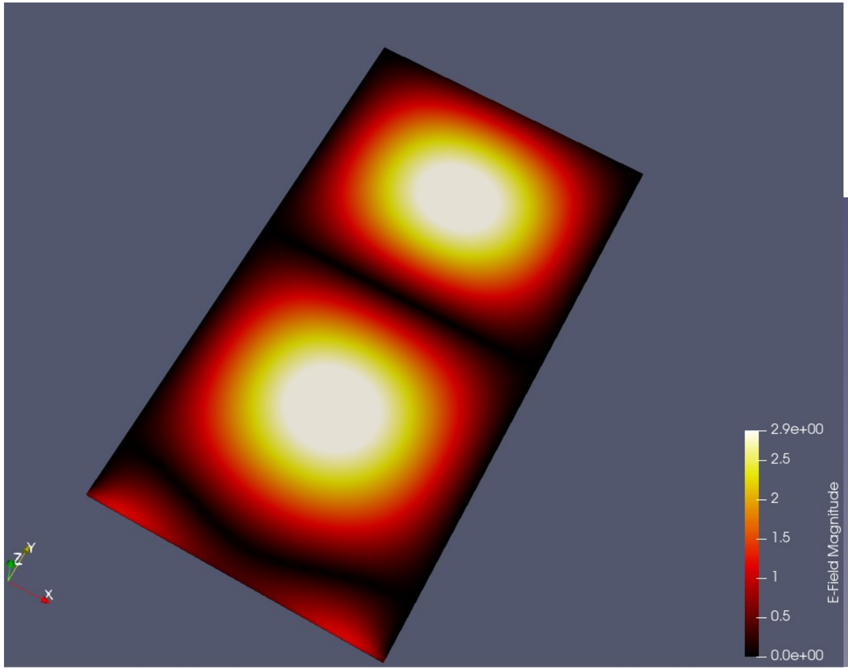


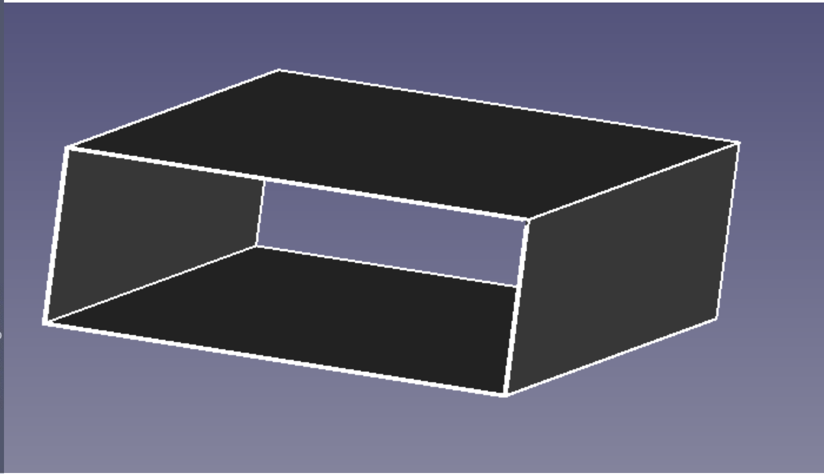




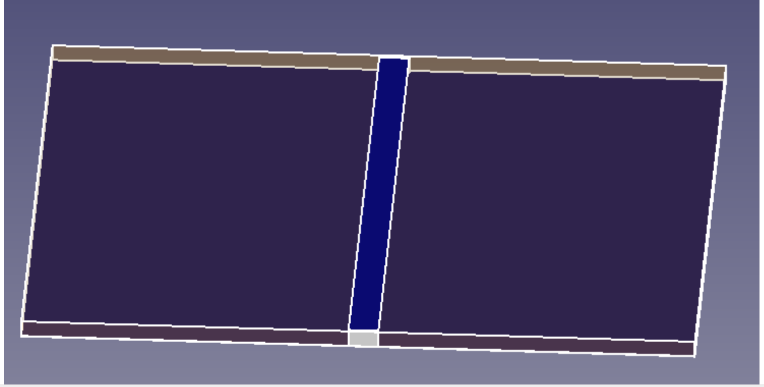
Rectangular waveguide

- great example to visualize E-field
- input ports are excited by modes Texx
  - in GUI in list they are ordered from lowest energy to highest – TE10, TE20, TE01, TE11, TE21, TE30, TE31, TE40, TE02





Microstrip simulation



- microstrip dimensions are frequency independent
- dimensions (width) depends on requested characteristic impedance
- it has same impedance over every frequency

For $\frac{W}{h} < 1$ $\quad \varepsilon_{eff} = \frac{\varepsilon_r + 1.0}{2} + \frac{\varepsilon_r - 1.0}{2}\left[\frac{1}{\sqrt{1 + \frac{12h}{W}}} + 0.04\left(1 - \frac{W}{h}\right)^2\right]$

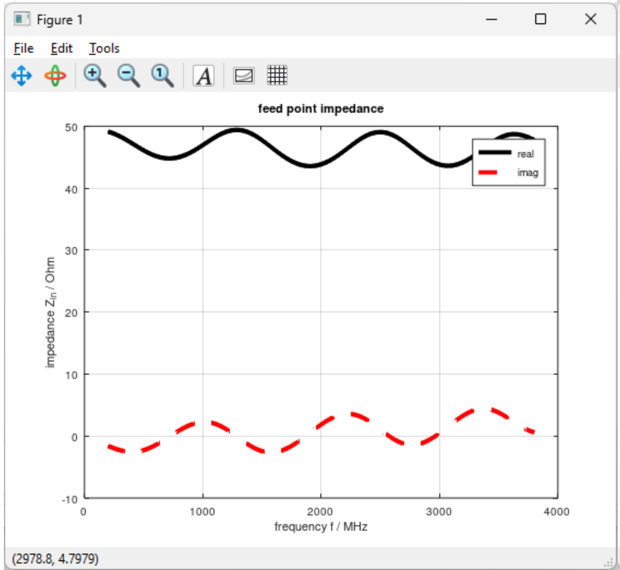
Else $\quad \varepsilon_{eff} = \frac{\varepsilon_r + 1.0}{2} + \frac{\varepsilon_r - 1.0}{2}\left[\frac{1}{\sqrt{1 + \frac{12h}{W}}}\right]$

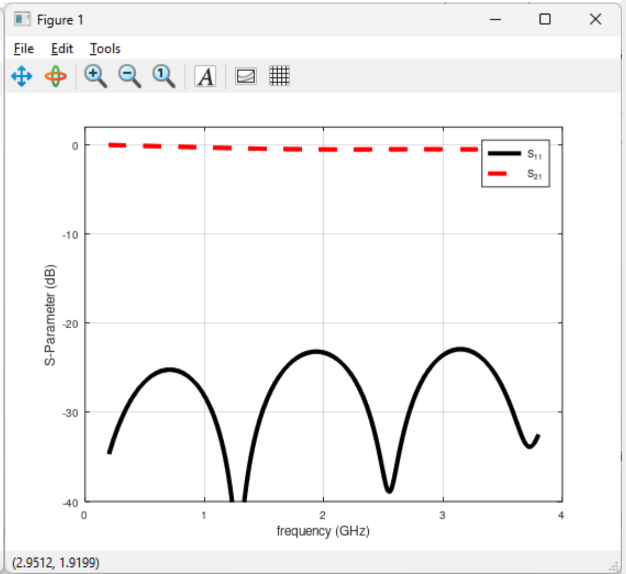
$$Z_o = \frac{120\pi}{2.0\sqrt{2.0}\,\pi\sqrt{\varepsilon_r + 1.0}} \ln\left\{1.0 + \frac{4.0h}{W'}\left[\frac{14.0 + 8.0/\varepsilon_{eff}}{11.0}\frac{4.0h}{W'} + \sqrt{\left(\frac{14.0 + 8.0/\varepsilon_{eff}}{11.0}\right)^2\left(\frac{4.0h}{W'}\right)^2 + \frac{1.0 + 1.0/\varepsilon_{eff}}{2.0}\pi^2}\right]\right\}$$

Where $\quad W' = W + \Delta W'$

$$\Delta W' = \Delta W\left(\frac{1.0 + 1.0/\varepsilon_{eff}}{2.0}\right)$$

$$\Delta W = \frac{t}{\pi}\ln\left[\frac{4e}{(t/h)^2 + \left(\frac{1/\pi}{w/t + 1.1}\right)^2}\right]$$

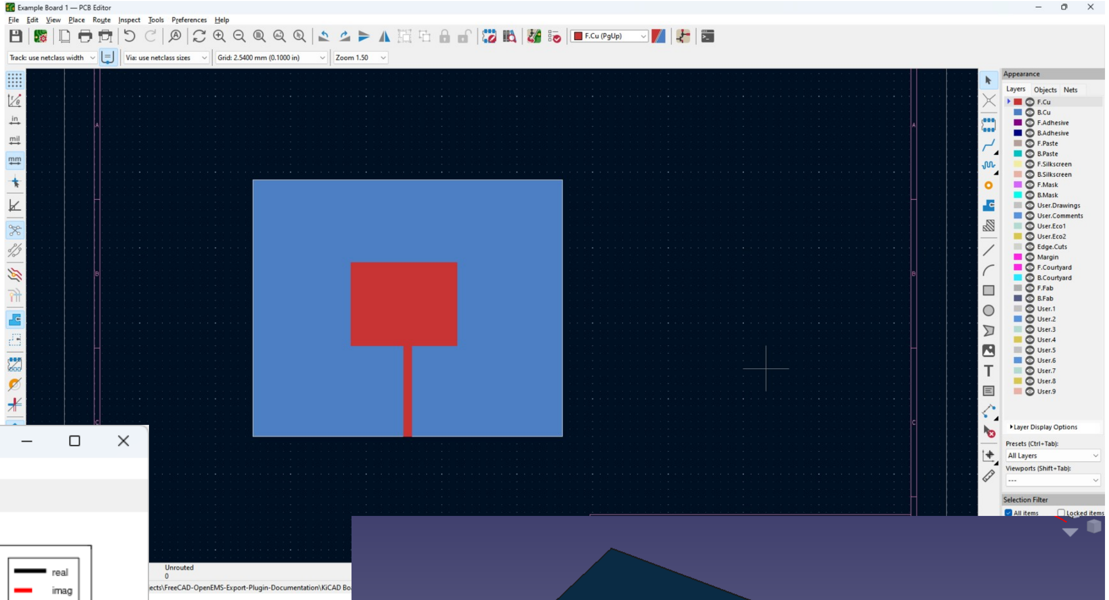


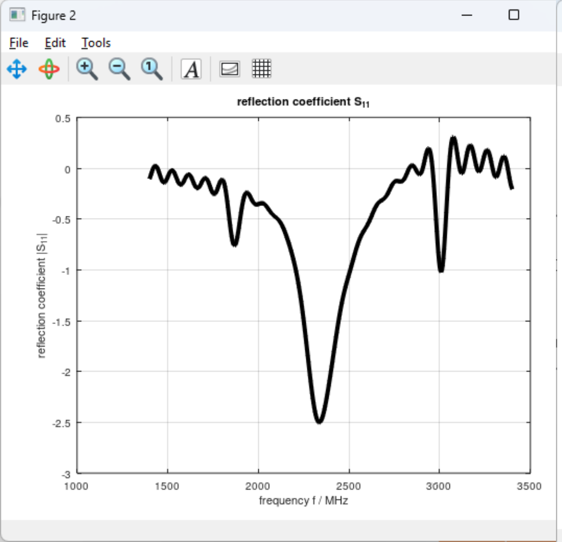


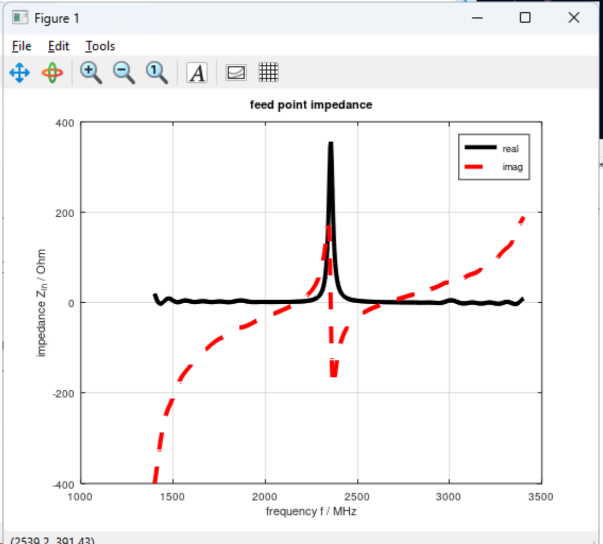
KiCAD board import

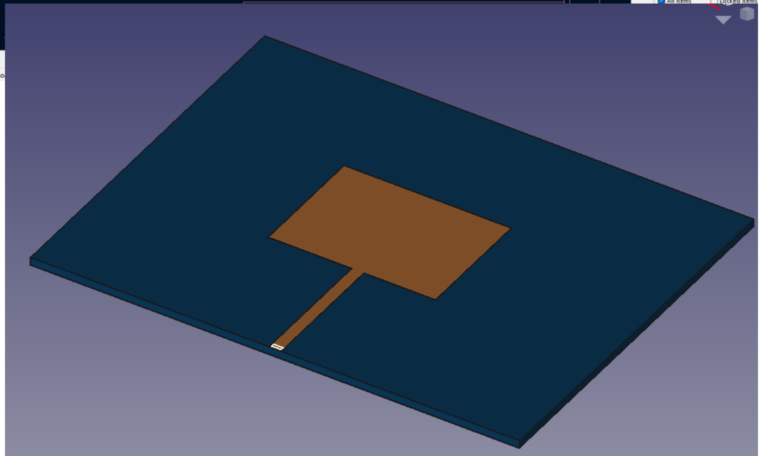
- use polygons
- DON’T USE RECTANGLES they are not imported
- there is tool accesible from main GUI window for KiCAD board importer using fcad_pcb python library
  - https://github.com/realthunder/fcad_pcb

- this one example was random drawn patch antenna in KiCAD therefore S11 is bad (really bad), microstrip should be improved



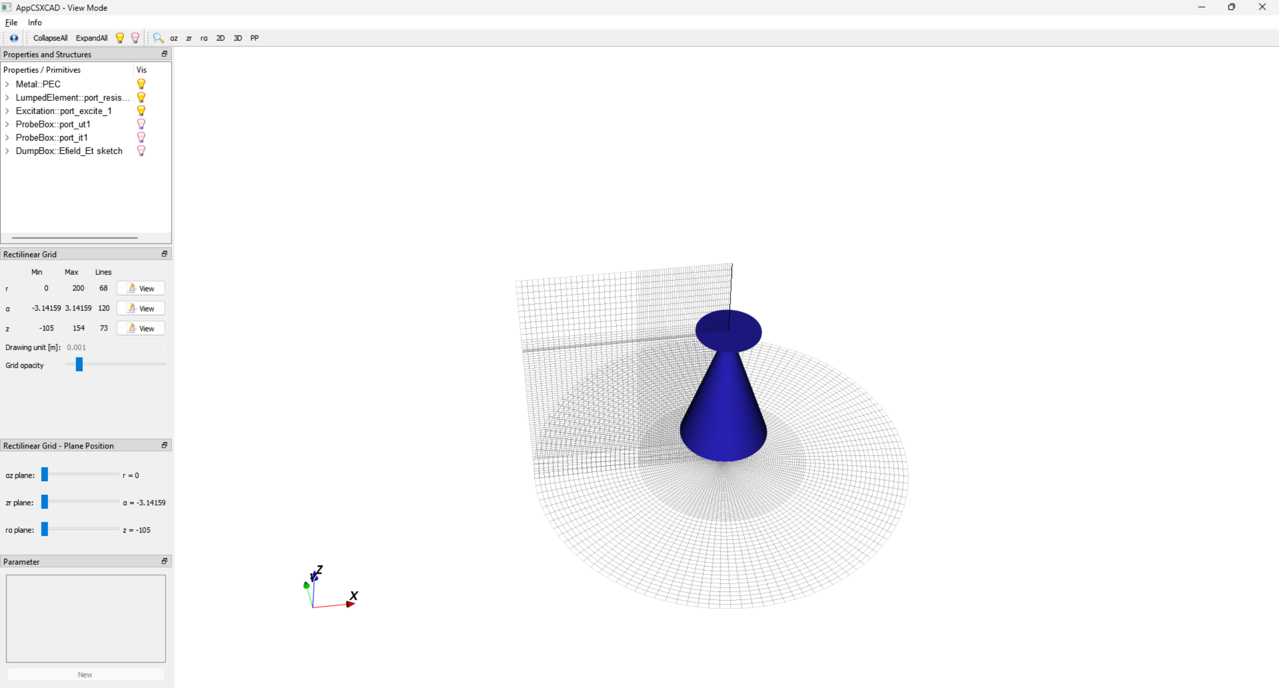


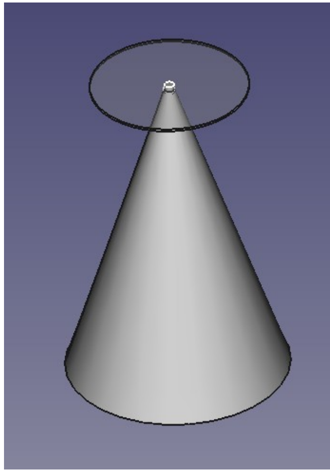




Discone antenna example

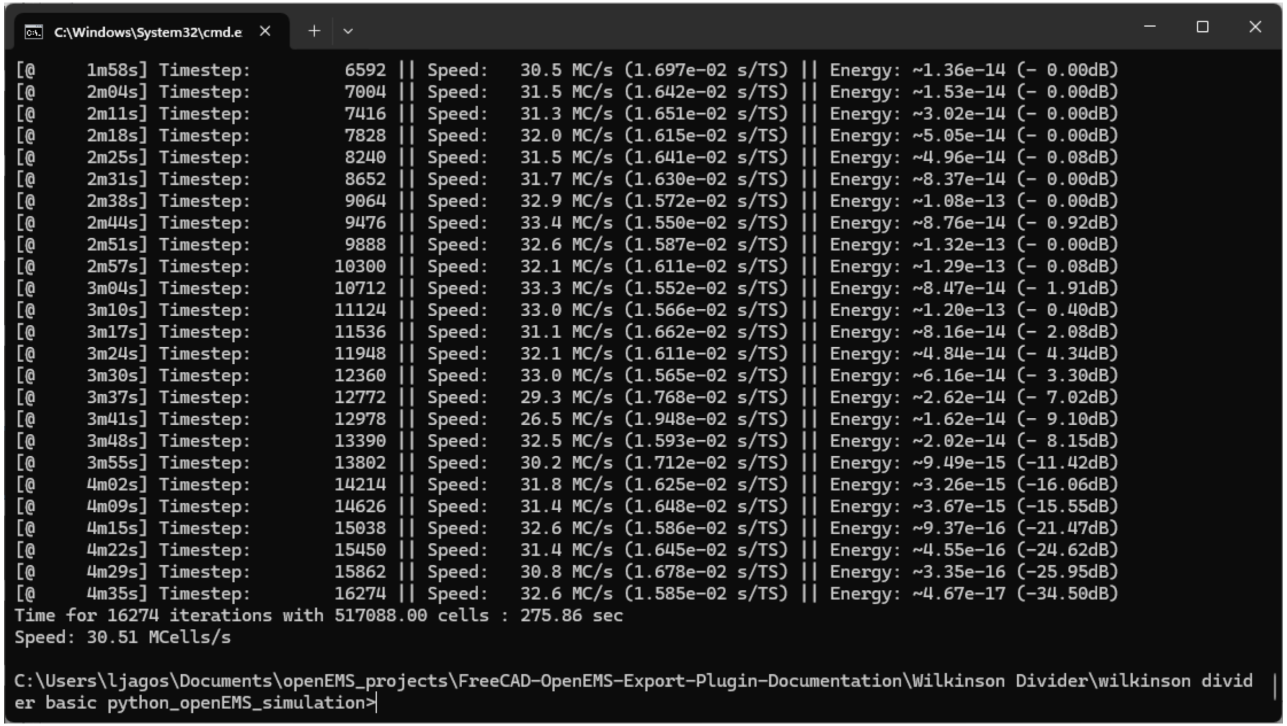
- example of cylindrical coordinate system
- there is something wrong and numerical instability occurs – EXAMPLE NOT FINISHED





Simulation process

- energy should be continualy decreasing
- if energy is not decreasing or not changing there is probably something wrong in simulation
- if energy is still from beginning 0.00dB most likely source port is not excited or connected to strcuture – good check model



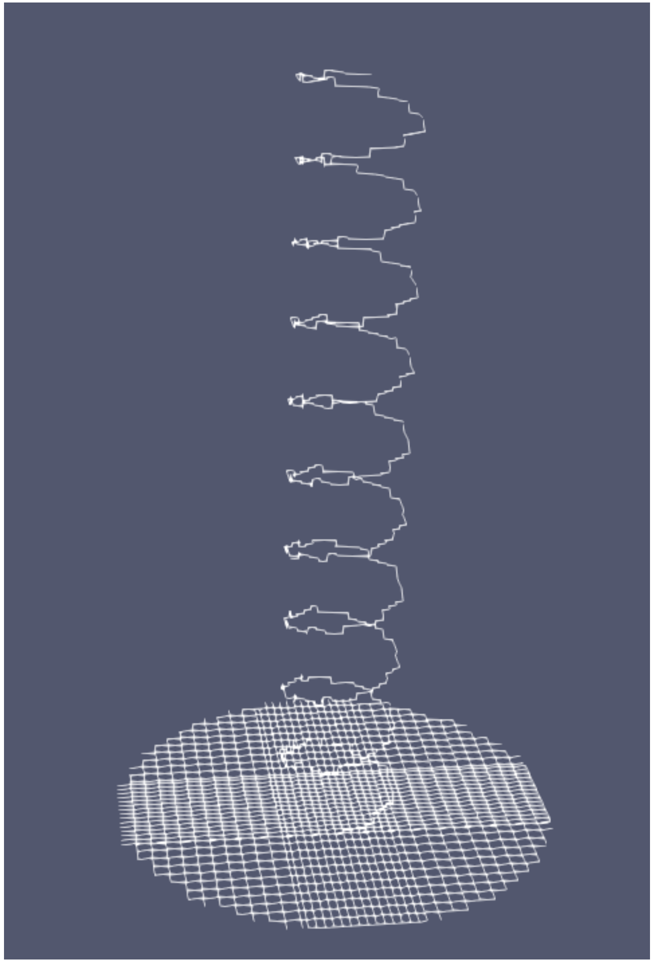
Simulation start

- simulation start, there is information about calculated timestep and also how long is excitation signal in timesteps, it's important to have at least maximum number of timesteps 2x greater than excitation signal
- number of cells is also important information, more cells means looong simulation

```
Create FDTD operator (compressed SSE + multi-threading)
FDTD simulation size: 181x152x82 --> 2.25598e+06 FDTD cells
FDTD timestep is: 2.12825e-12 s; Nyquist rate: 460 timesteps @5.10729e+08 Hz
Excitation signal length is: 13461 timesteps (2.86483e-08s)
Max. number of timesteps: 30000 ( --> 2.22866 * Excitation signal length)
openEMS::SetupFDTD: Warning, max. number of timesteps is smaller than three times the excitation.
	You may want to choose a higher number of max. timesteps...
Create FDTD engine (compressed SSE + multi-threading)
Warning: No primitives found in property: PEC!
Running FDTD engine... this may take a while... grab a cup of coffee?!?
[@	13s] Timestep:	115 || Speed:   18.8 MC/s (1.197e-01 s/TS) || Energy: ~7.72e-19 (- 0.00dB)
[@	22s] Timestep:	230 || Speed:   31.2 MC/s (7.241e-02 s/TS) || Energy: ~1.19e-18 (- 0.00dB)
```

Model check

- there is switch for openEMS -debug-PEC which generate PEC material into file then structure could be checked using Paraview, good to debug when something is wrong with model
- also possible to export other dielectric materials to see how it looks like in grid



Noticebly people using openEMS

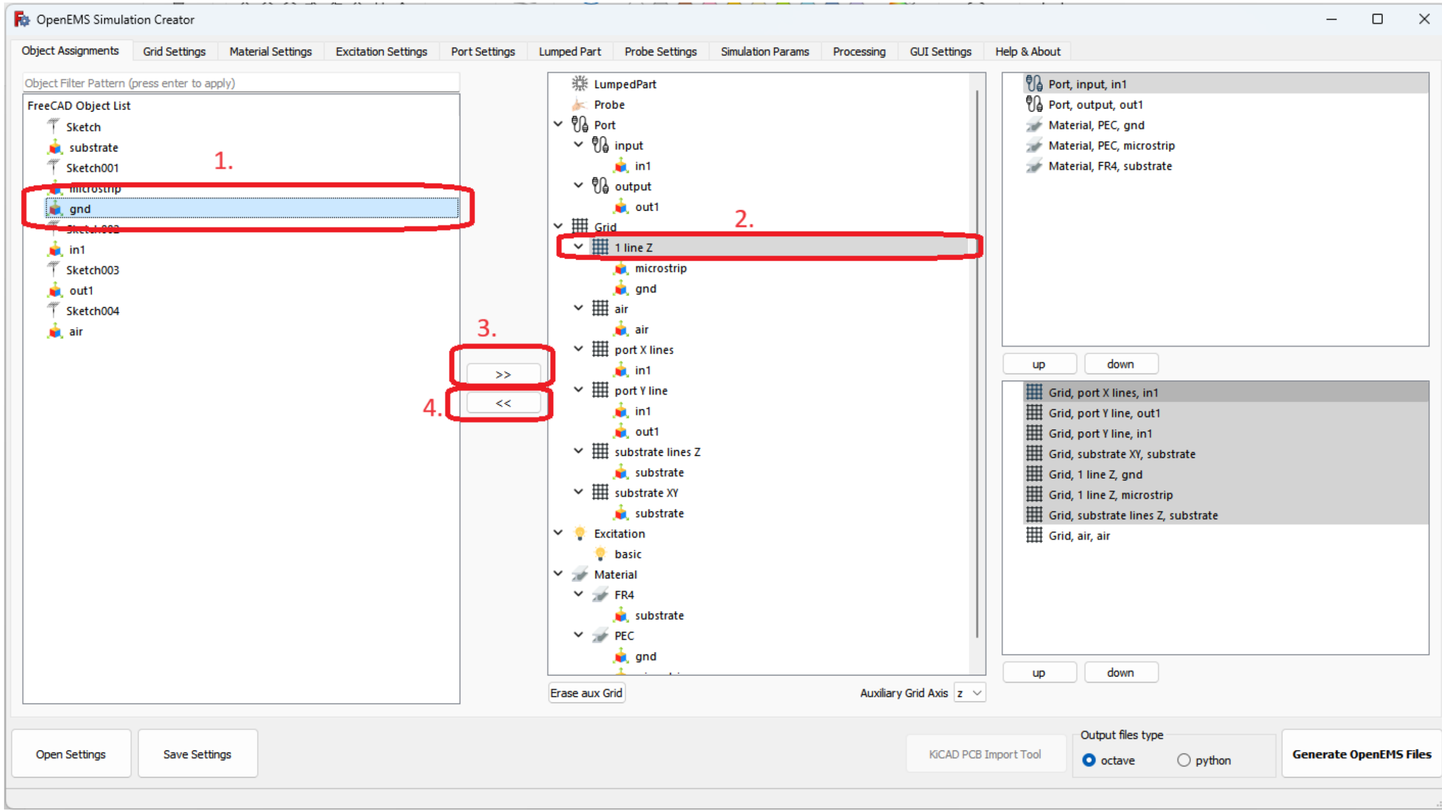
- Thorsten Liebig – author of openEMS
- Gadi Lahav – israeli RF designeer, he added inductance lumped element support to openEMS
- Apostolos Spanakis Misirlis – greek RF engineer
- Samer Aldhaher – UK electrical engineer, great electronics animation creator
- Volker Mühlhaus – german RF Specialist
- Leibniz University Hannover – students using openEMS for semestral projects

Thanks for attention.

Following slides are about usage of GUI addon in FreeCAD

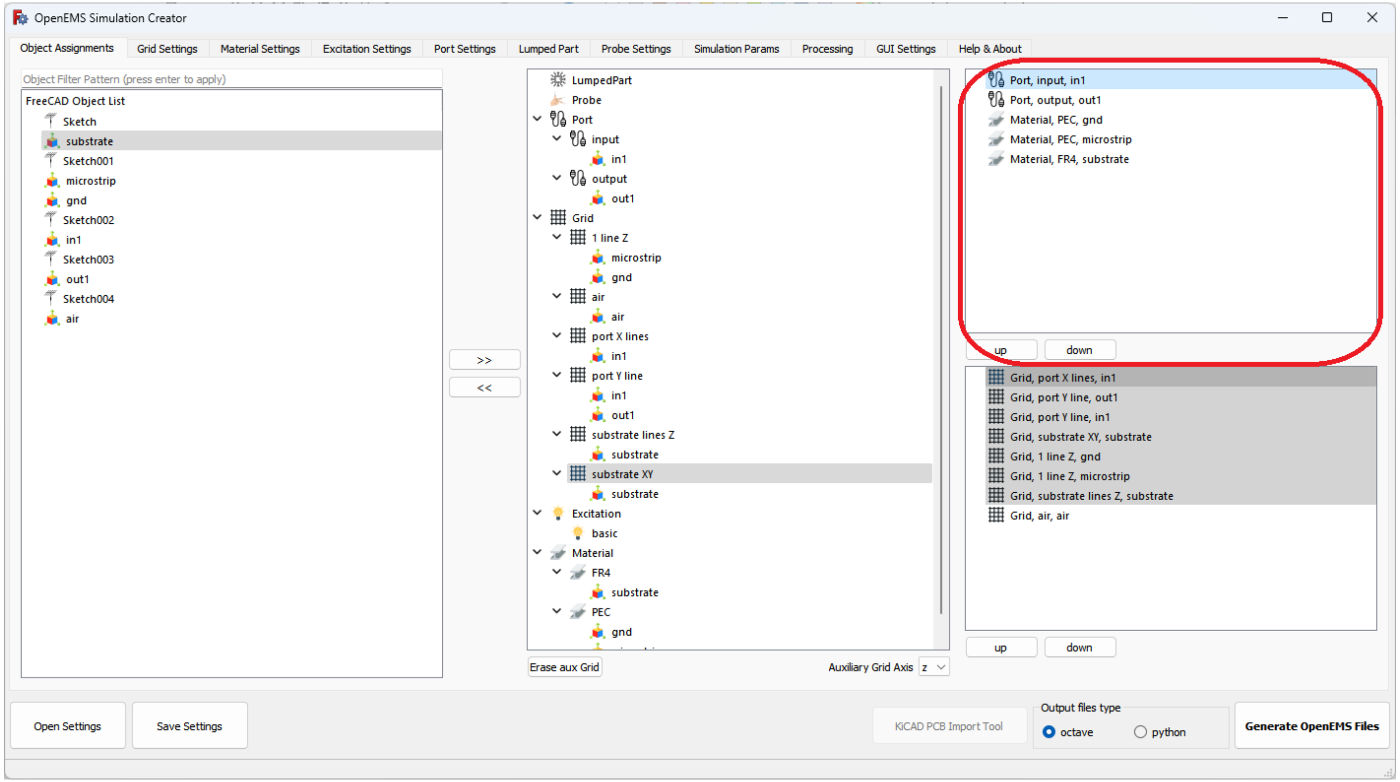
GUI addon – object assignment

1) select object
2) select category (grid, material, port, probe or lumped part)
3) use assign button (>>) to add object under some category (if it's not possible to add object to some category or there is some problem there is error message)
4) if object should be unassigned select object in category and click on unassign button (<<)



GUI addon – material priorities

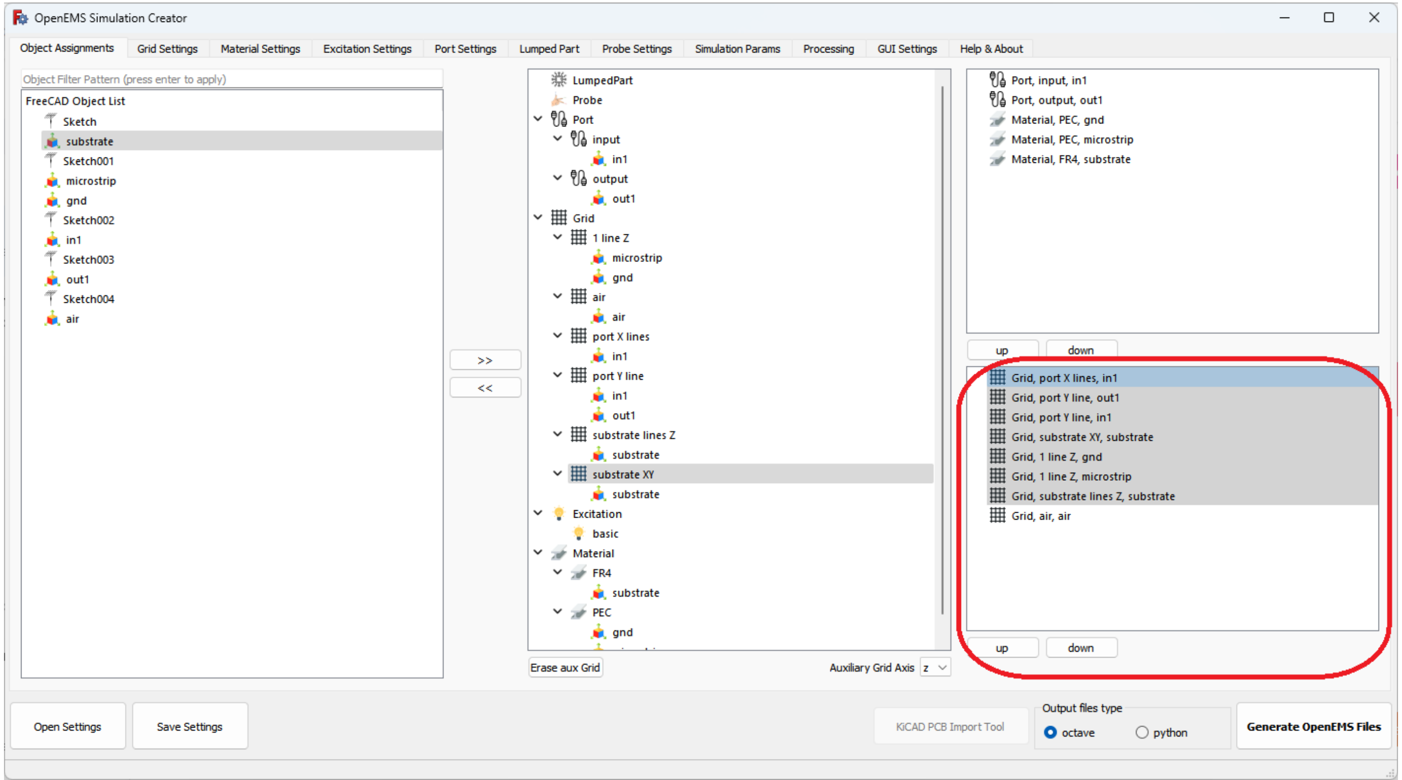
- if two objects with different material properties overlapped material with higher priority is used for nodes which lay inside object
- good practice is to have ports at top to be sure that excited port is really used



GUI addon – gridlines priorities

- gray grid items are deleting gridlines inside object boundaries, evaluation is going from bottom to top so first bottom items delete gridlines and add their gridlines and after it's going to item up, again delete gridlines and add lines
- therefore top items are last applied
- white grid items are not deleting gridlines inside assigned object boundaries and are just adding lines

- grid item can be selected and moved up and down using buttons under it
- good practice is to hav air gridlines which are most coarse at the bottom

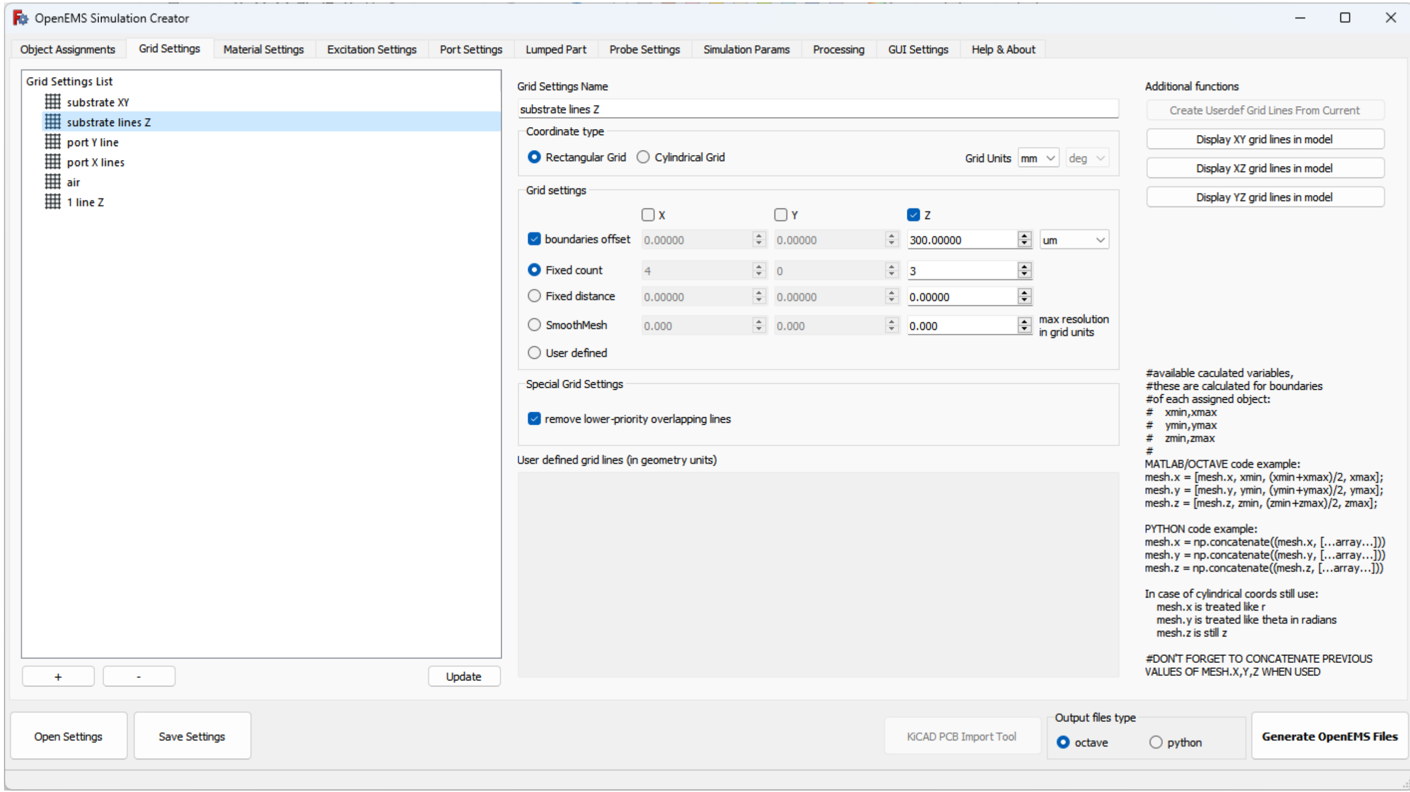


GUI addon – grid

- boundaries offset – moves gridlines inside object boundaries, by specified distance, this is to ensure that lines will be inside STL imported object due there is some rounding defined in GUI addon on 14 decimal places, so STL boundaries are evaluated in GUI addon to picometer precision

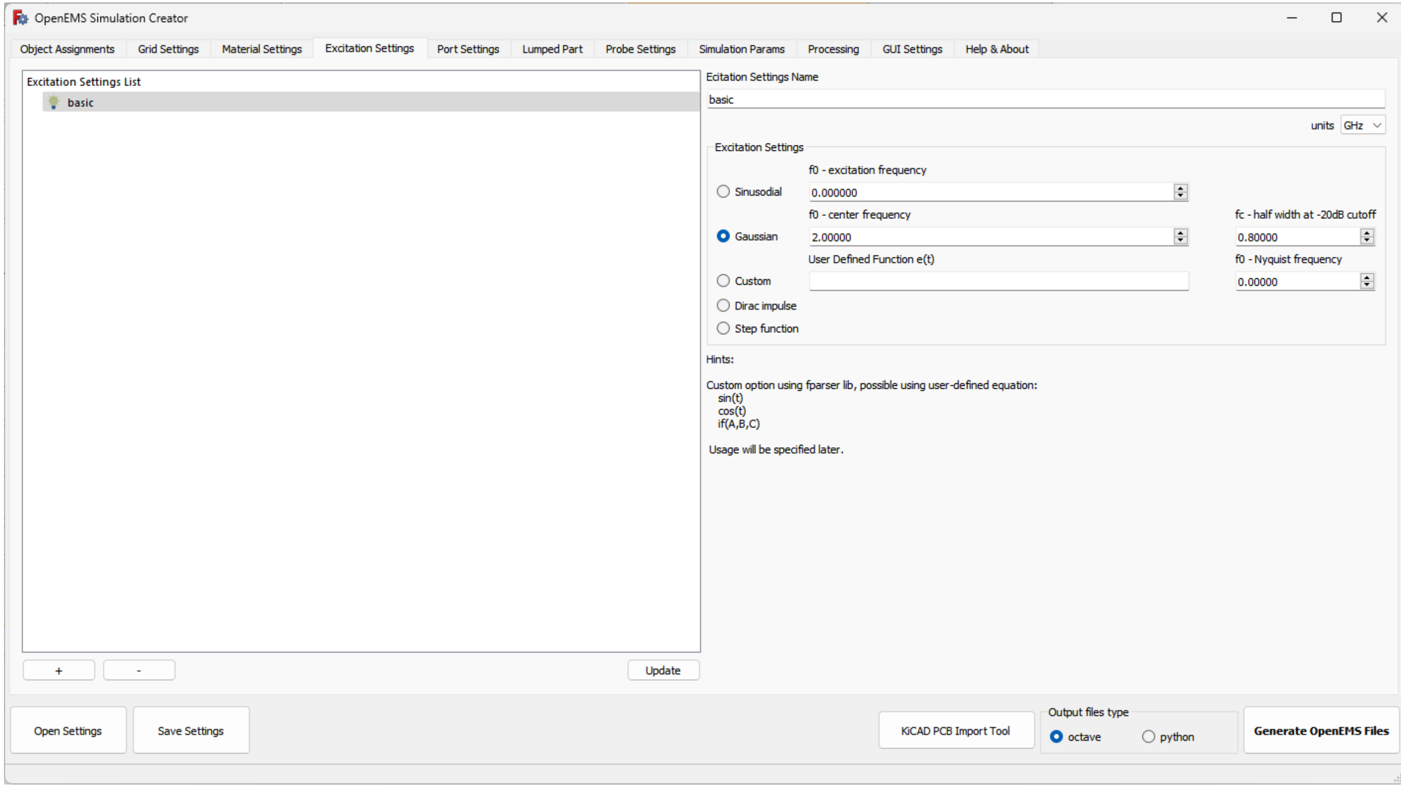
- smooth mesh creates gridlines how it should be disperzed into defined boundaries used in lot of orignal script openEMS examples
- this is little different as all object assigned into smoot mesh item are evaluated together
- smooth mesh is good to try to see what it's doing

- code from user defined is directly inserted into script therefore octave or python code must be used if choosed



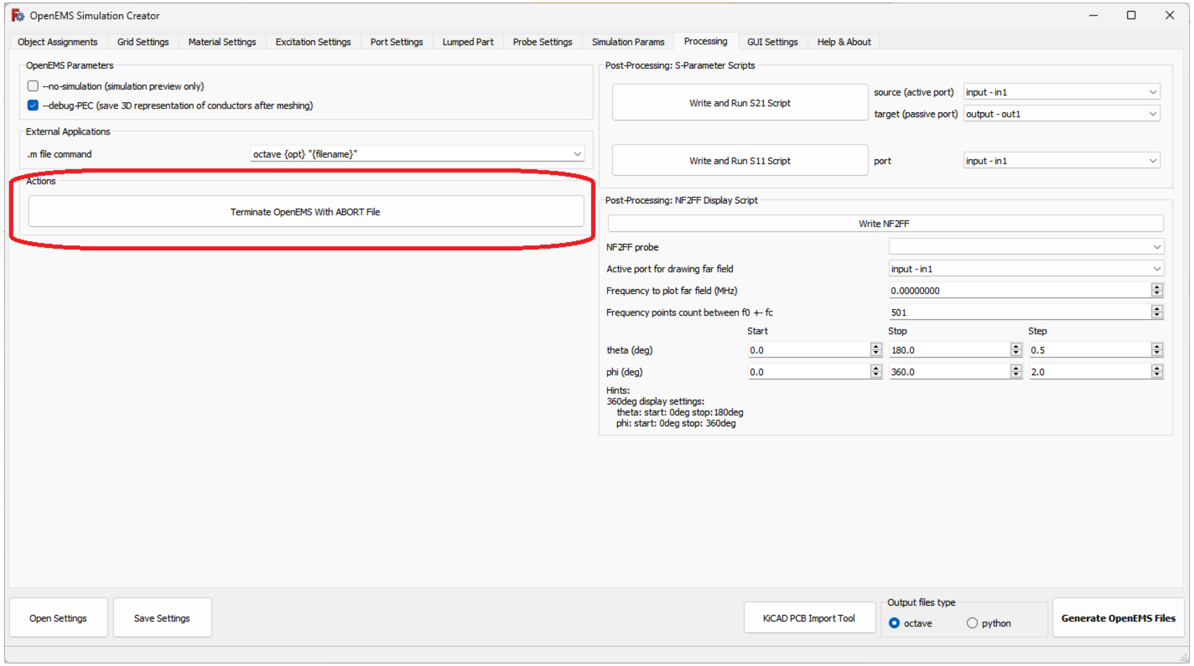
GUI addon – excitation

- only one excitation allowed in openEMS simulation
- port delay excitation not implemented in GUI, must be done in script manualy if needed



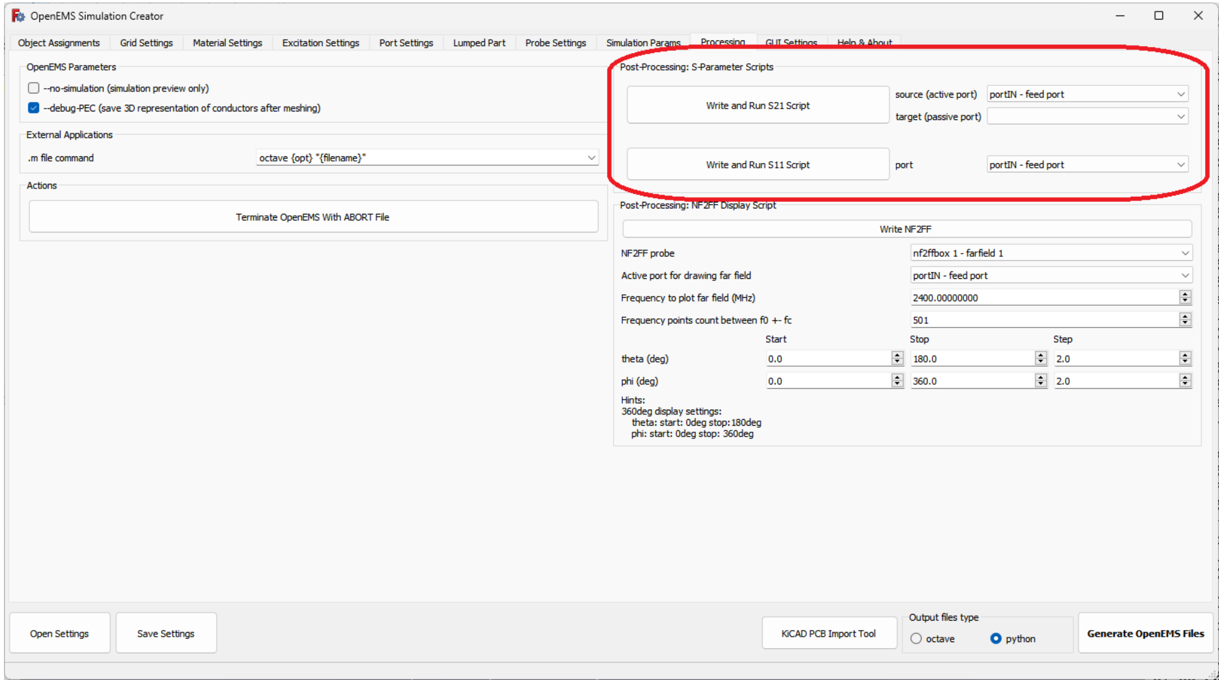
GUI addon – abort simulation

- when simulation is running it could be aborted by adding file named ABORT into simulation folder
- this is done by clicking on button in GUI



GUI addon – S11, S21 scripts

- on postprocessing tab there are buttons to generate S11 and S21 evaluate scripts, they display graph
- S11 for some port also draw input impedance
- could be used to inspire how to evaluate results from openEMS result files



GUI addon – far field display script

- there is in postprocessing button to generate far field script
- it uses NF2Ff (near field to far field) application which is part of openEMS
- it calculates far field from E, H fields on NF2FF box which is defined around structure