

0.1 Building the Grid

- objective is selecting the material properties (permittivity, permeability, conductivity) for each node in the grid
 - we have a staggered grid with prisms as cells (boxes), this gets inaccurate for certain geometries, think of spheres and cylinders but also structures that are very thin in comparison to the cell size

0.1.1 Structure of a Simulation

- top-down approach, we need broad simulation configuration, that is the boundaries, time step, number of cells
 - then we have collection of objects that have a geometry and materials, these will be given as a scene tree which can contain the CSG-operations union, intersection, and difference
 - the following objects are planned: box, cylinder, ellipsoid (generalization of sphere), custom model (wavefront obj)
 - position/rotation/scale can be given parametrically
 - box: either (corner1, corner2) or (center, extents in x,y,z)
 - sphere: (center, radius in x,y,z) or (corner1, corner2)
 - cylinder: (corner1, corner2) or (center, extents in x,y,z)
 - model: (corner1, corner2) or (center, extents in x,y,z)
 - each one can additionally be rotated, with Euler angles pitch, yaw, roll
 - every object can have one assigned material
 - material consists of permittivity, permeability, conductivity
 - these can be anisotropic in x, y, z
 - side note: data of the whole grid can get very large, several gigabytes even for smaller problems, therefore there definitely needs to be some kind of compression (RLE or sparse matrix approach)

0.1.2 Approach to Building the Grid

- simplest form: fill material properties by directly sampling from the objects at the positions of electric and magnetic fields in each cell
 - laid out: go through every node, determine position of every component of electric/magnetic field, sample from the objects
 - in case there are several overlapping objects: the highest permeability, permittivity, conductivity is picked
 - add image for this
 - there is an error at boundaries, e.g. spheres, cylinders or even prisms with this method
 - increase resolution for no additional space usage by splitting cell into eight sub-cells, sampling them and taking the arithmetic mean, since that is more costly in computation, only do this at boundaries

- other object that comes up quite often are thin plates, e.g. microstrips or other lines on printed circuit boards, conforming the grid to those thin lines would require lots of cells
- faster alternative: align grid with those thin strips and make the respective face of the cell into a high-conductivity material (e.g. for a microstrip)
- thin wires can be done similarly, just make one side of a cell into a highly conductive line
- problem: cell has to conform exactly, either only allow thin stuff on one plane or we need to come up with an algorithm that finds an optimal cell size that aligns with as many thin features as possible, if for example there are several planes in a pcb