0.1 Full FDTD for the 3D Case

- again start with maxwells
 - currents J_x, J_y, J_z were added as they are necessary for voltage and current sources
 - also anisotropy is possible as permittivity, permeability and conductivity no can vary depending on direction

$$
\begin{aligned}
\partial_y E_z - \partial_z E_y &= -\mu_x \partial_t H_x && (1)\\
\partial_z E_x - \partial_x E_z &= -\mu_y \partial_t H_y && (2)\\
\partial_x E_y - \partial_y E_x &= -\mu_z \partial_t H_z && (3)\\
\partial_y H_z - \partial_z H_y &= \varepsilon_x \partial_t E_x + \sigma_x E_x + J_x && (4)\\
\partial_z H_x - \partial_x H_z &= \varepsilon_y \partial_t E_y + \sigma_y E_y + J_y && (5)\\
\partial_x H_y - \partial_y H_x &= \varepsilon_z \partial_t E_z + \sigma_z E_z + J_z \ . && (6)
\end{aligned}
$$

The operator ∂_x is a shorthand for the partial derivative $\frac{\partial}{\partial x}$.
 - no terms are going to be dropped now
 - since form of eqn 1-3 and 4-6 are similar, well only do 1 and 4 and then transfer those results
 - the grid will look like this now (add grid and coordinate system indication = left-handed system here)
 - node is indicated by a 3-tuple (i,j,k) where x..., y..., z...
 - to make the following expressions shorter i,j,k (current node) are taken as default parameters, only the relevant differing parameters are written out, e.g. i+1, j+1, or k+1
 - E(i,j,k) becomes E, E(i,j+1,k) would become E(j+1)
 - now for equation 1, after discretization

$$\frac{E_z^{n+1/2}(j+1) - E_z^{n+1/2}}{\Delta y} - \frac{E_y^{n+1/2}(k+1) - E_y^{n+1/2}}{\Delta z} = -\mu_x \frac{H_x^{n+1} - H_x^n}{\Delta t} \tag{7}$$

 - rearranging to the desired ...

$$H_z^{n+1} = H_z^n - \frac{\Delta t}{\mu_x \Delta y}\left(E_z^{n+1/2}(j+1) - E_z^{n+1/2}\right) + \frac{\Delta t}{\mu_x \Delta z}\left(E_y^{n+1/2}(k+1) - E_y^{n+1/2}\right) \tag{8}$$

 - in turn for equation 4, after discretization

$$\frac{H_z^n(j+1) - H_z^n}{\Delta y} - \frac{H_y^n(k+1) - H_y^n}{\Delta z} = \varepsilon_x \frac{E_x^{n+1/2} - E_x^{n-1/2}}{\Delta t} + \sigma_x \frac{E_x^{n+1/2} - E_x^{n-1/2}}{2} + J_x \tag{9}$$

 - rearranging to ...

$$E_x^{n+1/2} = E_x^{n+1/2} + \frac{2\Delta t}{2\varepsilon_x + \Delta t \sigma_x}\left(\frac{H_z^n(j+1) - H_z^n}{\Delta y} - \frac{H_y^n(k+1) - H_y^n}{\Delta z} - J_x\right) \tag{10}$$

 - write out the results for all 6 equations
 - cfl condition

$$\Delta t \leq \frac{1}{c\sqrt{\frac{1}{(\Delta x)^2} + \frac{1}{(\Delta y)^2} + \frac{1}{(\Delta z)^2}}} \tag{11}$$

 - cell size, depends on simulation wavelength (chosen frequency), as general high-frequency principle no dimension of cell size should be larger than one tenth of wavelength
 - example 2.4ghz leads to 12.5cm wavelength
 - therefore 1.25 cell size is ok

$$\Delta t \leq= \frac{1}{c\sqrt{\frac{3}{(1.25cm)^2}}} = \frac{1.25cm}{\sqrt{3}c} \approx 24ps \tag{12}$$

0.2 Simulations

- i won't write any matlab scripts for this one, would be a waste of effort
  - instead, head to next note on building the grid
  - from that point, focus should be spent on implementing a functioning toolchain
  - toolchain consists of: parametric modeller, grid builder, simulator, result visualizer
  - none of these need to be fancy, parametric modeller only needs to export some boxes for a start, grid builder needs problem configuration and putting in a source, e.g. voltage source, simulator should probably be in grid builder, result visualizer should provide a good way of judging vector fields (plane cuts and 3D view with many lines, thickness/color depends on magnitude or something)