

Topic 4: FEM Magnetics Toolbox

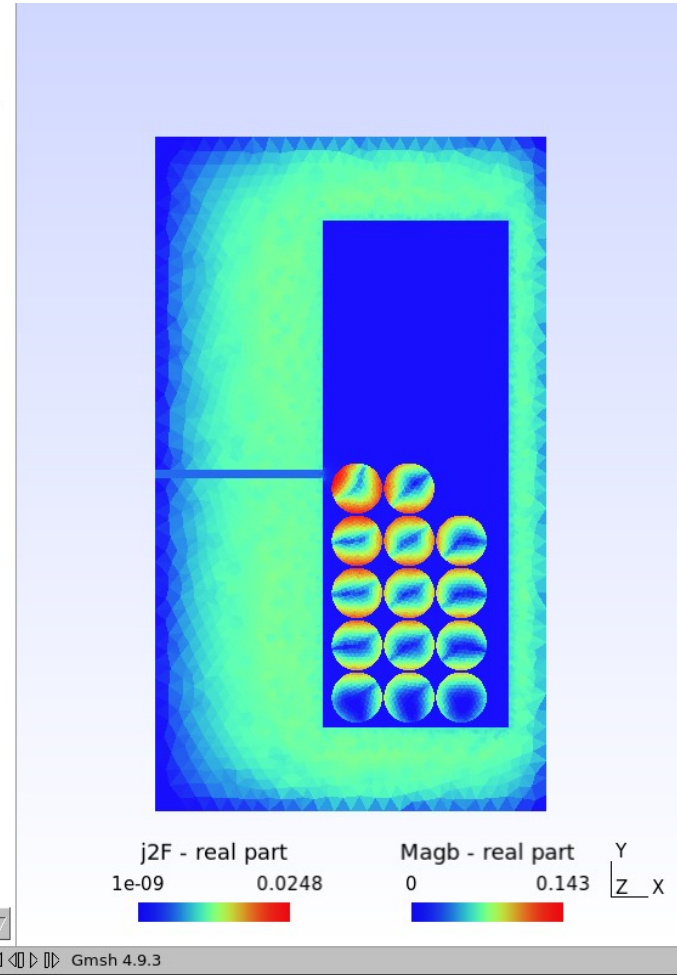
Extension of the FEM Magnetics Toolbox for time domain simulation

```
1 # minimal example for github clone
2 from femmt import MagneticComponent
3 import numpy as np
4
5 component = "inductor"
6
7 if component == "inductor":
8     geo = MagneticComponent(component_type="inductor")
9
10    geo.core.update(type="EI", window_h=0.03, window_w=0.011)
11
12    geo.air_gaps.update(method="center",
13                       n_air_gaps=1,
14                       air_gap_h=[0.0005],
15                       position_tag=[0])
16
17    geo.update_conductors(n_turns=[[14]],
18                         conductor_type=["solid"],
19                         conductor_radii=[0.0015],
20                         winding=["primary"],
21                         scheme=["square"],
22                         core_cond_isolation=[0.0005],
23                         cond_cond_isolation=[0.0001])
24
25    geo.single_simulation(freq=100000, current=[1])
26
27
```

basic_example x

```
Version      : 4.9.3
License      : GNU General Public License
Build OS     : Linux64-sdk
Build date   : 20220104
Build host   : gmsh.info
Build options: 64Bit ALGLIB[contrib] ANW[contrib] Bamg Blas[petsc] E
FLTK version : 1.4.0
PETSc version: 3.14.4 (real arithmetic)
OCC version  : 7.6.0
MED version  : 4.1.0
Packaged by  : geuzaine
Web site     : https://gmsh.info
Issue tracker: https://gitlab.onelab.info/gmsh/gmsh/issues
```

- Modules
 - Geometry
 - Mesh
 - Solver
 - Post-processor



Intro: FEM Magnetics Toolbox

FEM Magnetics Toolbox

Manual Design | Automated Design

Definition | Excitation | Simulation

Simulation Definition
Simulation Type: inductor

Core Definition
Material: N95
Core from database: Manual
2D axisymmetric core parameters
Window Height [m]: 0.03
Core inner diameter [m]: 0.02 | Window Width [m]: 0.011

Winding 1: Conductor Definition
Winding Material: copper | Wire Type: Litz Wire
Litz from database: Manual
Manual wire parameters
Wire Radius [m]: 0.0015
Implicit: Litz Rad | Fill Factor: 0.6
Number Strands: 600 | Strand Radius [m]: 35.5e-6

Winding Scheme
Number Turns Winding 1: 25
Winding Scheme Winding 1: Square

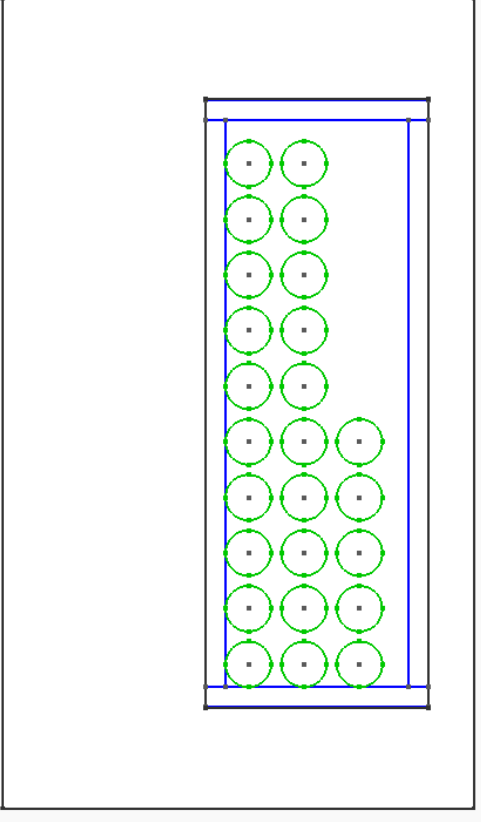
Air Gap Definition
No. of Air Gaps: 0
Method: Percent

Winding Isolation
P2P [m]: 0.0005

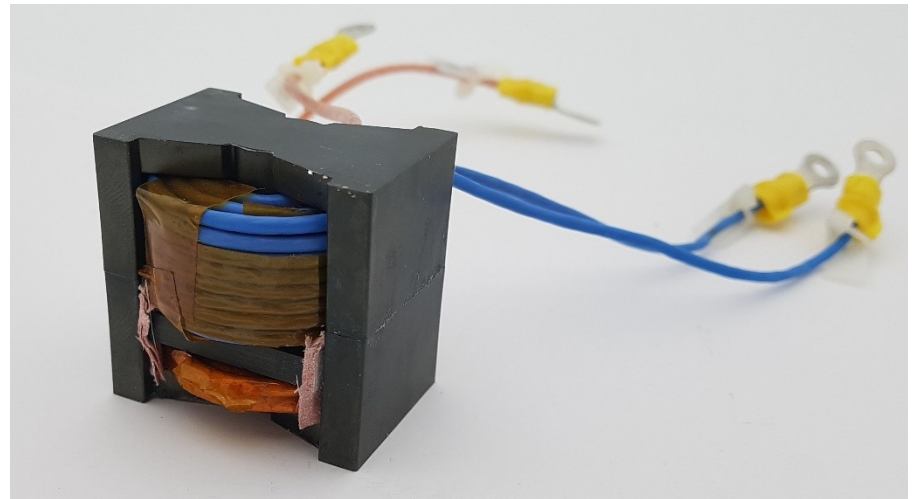
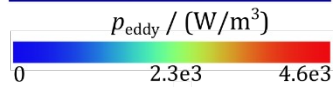
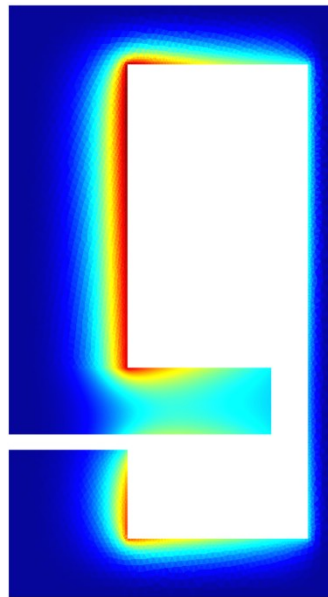
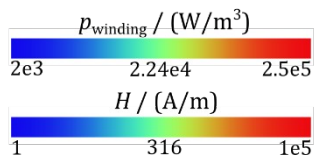
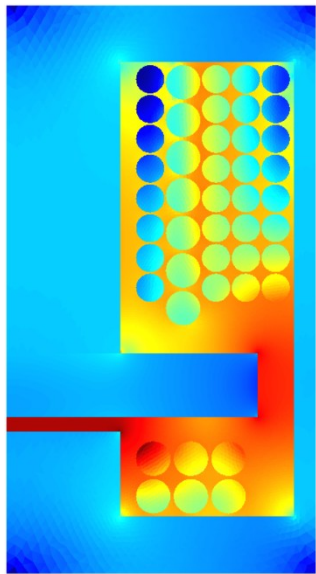
Core Isolation (Bobbin)
Core2Cond top [m]: 0.001 | Core2Cond inner [m]: 0.001
Core2Cond bottom [m]: 0.001 | Core2Cond outer [m]: 0.001

Update Preview

Gmsh Visualisation



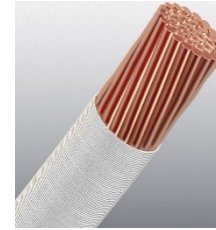
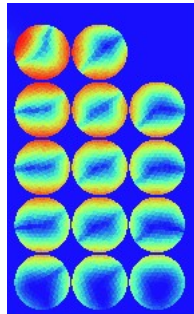
Previous Work Example



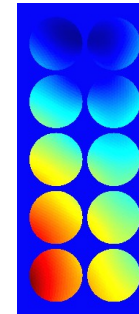
windings



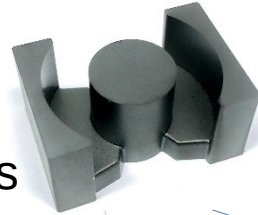
solid



strands



skin effect and proximity effect



(ferrite) cores

type

empirical

physical

frequency domain

approach

Steinmetz equation

$$P = \frac{1}{T} \int_0^T k_i \left| \frac{dB}{dt} \right|^a (\Delta B^{b-a}) dt$$

eddy currents

$$p_{eddy} = -\omega \Im(\underline{\underline{\epsilon}}) |\underline{E}|^2$$

hysteresis

$$p_{hysteresis} = -\omega \Im(\underline{\underline{\mu}}) |\underline{H}|^2$$

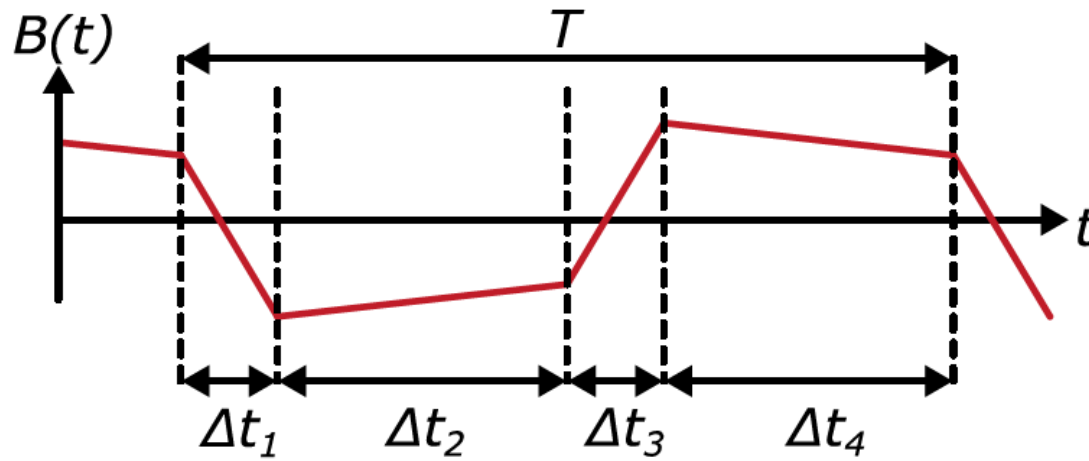
data

$$k_i, a, b$$

$$\underline{\underline{\epsilon}} = \epsilon' - j\epsilon''$$

$$\underline{\underline{\mu}} = \mu' - j\mu''$$

Example: Dual Active Bridge



in PE often non-sinusoidal flux:

- for **linear** material: **superposition**
- For **non-linear** material: **time domain**

Integrate an existing code example into our framework

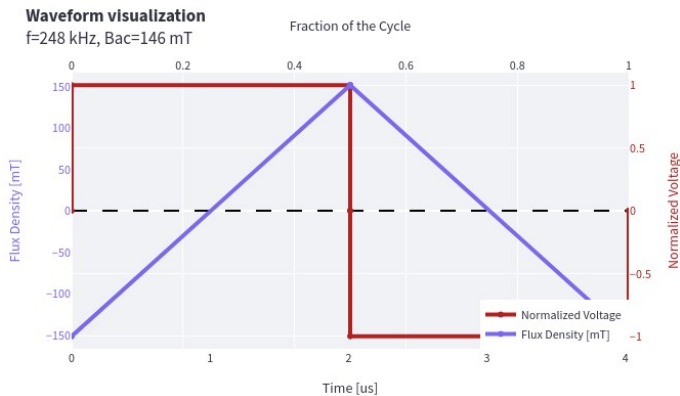
```
Operation {  
  CreateDirectory[resPath];  
  If(Flag_FrequencyDomain)  
    Generate[A]; Solve[A]; SaveSolution[A];  
  Else  
    InitSolution[A]; // provide initial condition  
    TimeLoopTheta[TimeInit, TimeFinal, DeltaTime, 1.]{  
      // Euler implicit (1) -- Crank-Nicolson (0.5)  
      Generate[A]; Solve[A];  
      If(NbrRegions[Vol_NL_Mag])  
        Generate[A]; GetResidual[A, $res0];  
        Evaluate[ $res = $res0, $iter = 0 ];  
        Print[{$iter, $res, $res / $res0},  
          Format "Residual %03g: abs %14.12e rel %14.12e"];  
        While[$res > NL_tol_abs && $res / $res0 > NL_tol_rel &&  
          $res / $res0 <= 1 && $iter < NL_iter_max]{  
          Solve[A]; Generate[A]; GetResidual[A, $res];  
          Evaluate[ $iter = $iter + 1 ];  
          Print[{$iter, $res, $res / $res0},  
            Format "Residual %03g: abs %14.12e rel %14.12e"];  
        }  
      EndIf  
      SaveSolution[A];  
    }  
  EndIf  
}
```



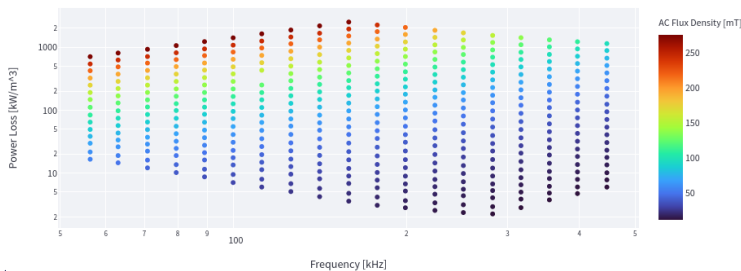
This project

Getting material data for time domain simulations

- Magnet database, contains lot of real-world measurements
- Use neuronal networks / machine learning to get the data for different wave forms



Flux Density vs Frequency and Power Loss



This project: time domain simulation



Goals

- Extend the FEMMT project with the capability for time domain simulations
- Use MagNet database to provide material data for the time domain simulation
- Validate the simulations with a commercial FEM tool (ANSYS Maxwell or Comsol Multiphysics)

This Project: time domain simulation



Skills (you can learn / improve)

- FEM (open source environment ONELAB)
- Neuronal networks / machine learning
- Python
- Version Control System (Git)
- Power Electronics

Deadlines / Organisation

- Make appointment for mandatory interview via mail to piepenbrock@lea.upb.de until **Tuesday Apr 11th, 8am**
- Interviews take place on **Wednesday Apr 12th**
- If you have time limits for April 12th, please include that in the email. We will try to take it into account.
- Possible topics in interview:
 - Power electronics
 - Law of induction
 -

Contact:

Till Piepenbrock, M.Sc.

Research Assistant

Email: piepenbrock@lea.upb.de

Power Electronics and Electrical Drives

Paderborn University

D-33098 Paderborn, Germany

Web: lea.upb.de

GENERAL INFORMATION

- **Self managed group work**
- **You are responsible for your results**
- **9 credits (= 270 h workload)**
- **Time range \approx 6 month with 10 h per week**
- **Meetings are held every week**
- **Not every applicant can be admitted to the project, since the number of participants is limited**

Questions?