An Open-Source FEM Magnetic Toolbox for Calculating Electric and Thermal Behavior of Power Electronic Magnetic Components

Nikolas Förster, Jonas Hölscher, Till Piepenbrock, Philipp Rehlaender,
Oliver Wallscheid, Frank Schafmeister, Joachim Böcker
Paderborn University
Warburger Str. 100
33098 Paderborn
Email: {foerster, piepenbrock}@lea.upb.de
URL: https://lea.upb.de

Keywords

≪Finite-element analysis≫, ≪Flux model≫, ≪Thermal model≫, ≪Cooling≫ ≪Simulation≫.

1 Abstract

Minimizing power losses and the thermal management are important factors in developing magnetics for power electronics. Both are very relevant to maintain the efficiency and the maximum operating temperatures of the core and conductors. To bring this calculation in the development workflow, the FEM (Finite Element Method) Magnetics Toolbox (FEMMT) will be continued. This open-source toolbox helps to automatically simulate magnetics in a guided, standardized format, requiring minimal effort. Based on the magnetoquasistatic simulation, a thermal simulation is set up. Conductor and core losses are taken from the magnetostatic simulation. The losses are homogenized inside the winding and inside the core. Thermal geometries and materials are set by script (e.g., for air gap filling and potting material), to automate the time-consuming drawing process inside the FEM tool. After simulation, the results are read back. With the help of this toolbox, which is publicly and collaboratively developed on GitHub, entire parameter sweeps can be easily performed and the simulation process is significantly speeded up.

2 Introduction

Losses of inductive components in winding and core can be calculated by different analytical or numerical methods [1, 2, 3, 4]. In this context, the question arises if the generated heat can be dissipated appropriately and if the given materials can be operated below the maximum temperature rating. Thus, it can be ensured that, for example, the core material keeps its magnetic properties (Curie temperature), or that the insulation of the conductors does not melt.

FEM simulations are suitable for calculating the local temperature distribution. Thus, hot spots are detected and it is ensured that the maximum temperatures of the different materials are maintained. For this purpose, there are excellent open-source programs [5, 6, 7], which provide interfaces for automation. Nevertheless, the model preparation is usually very time-consuming, as all parts of the magnetic component and the cooling structure have to be drawn manually, relationships and materials have to be established, and thermal boundary conditions have to be assigned. If a large number of setups (e.g., various cooling materials, different sizes of the inductive element and its housing) are to be tested, manual execution is practically infeasible. For that purpose, this paper presents the FEM magnetics toolbox (FEMMT), which can be used as the programming interface of the open-source tool ONELAB [5] to set up standardized magnetic components. The cooling structure is drawn by an abstracted code interface, using Python script language. Once the simulation has finished, the results for temperatures in conductors and core are read back.

The general workflow is shown in Fig. 1. First, a 2D-rotationally symmetric magnetoquasistatic simulation is parameterized according to [9], executed and the results are returned. Then, the thermal materials like air gap material, winding insulation or the potting material are set. After assigning the boundary conditions (ambient temperature and cooling method), the simulation is started. The simulation computes the power flow and temperatures. This paper focuses on the thermal simulation part, starting from the thermal geometry generation, since the magnetostatic part is already described in [9]. The geometry is described by code, so that different setups can be easily calculated and optimized.

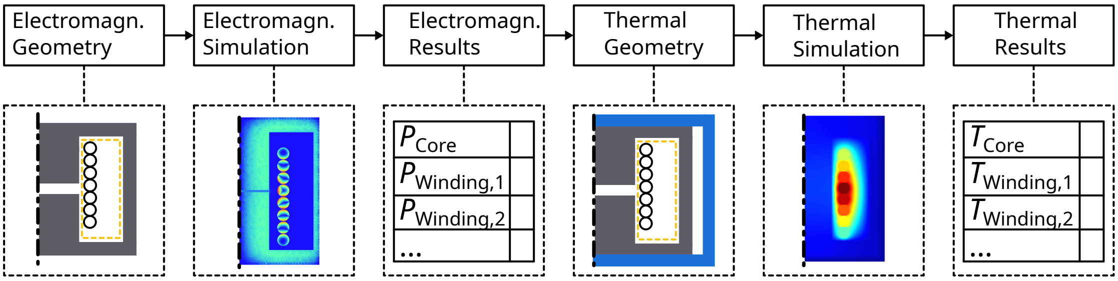


Fig. 1: Workflow to set up a thermal simulation using a magnetoquasistatic pre-simulation

3 Open-Source FEM-Magnetics Toolbox Using Thermal Simulations

This section describes the basics of the simulation tool, such as the abstract code interface for describing the cooling geometry.

FEM-Simulation Tool Details

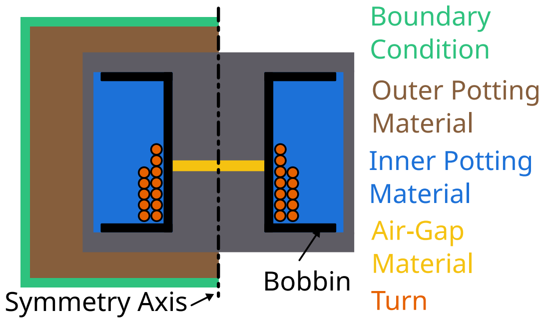
The chosen FEM simulation software is ONELAB [5]. The tool has been chosen as it is controllable via Python, and is an open-source software that is actively developed. It is natively executable on the usual operating systems (Linux/MacOS/Windows). ONELAB contains a self-writable solver, so special features can be taken into account, such as litz wire approximation, eddy currents inside the core for the magnetoquasistatic simulation, and for the thermal simulation the heat conduction and convection can be considered. Based on ONELAB FEM software, the FEM magnetics toolbox (FEMMT) was developed. FEMMT is an interface to ONELAB and includes pre- and post-processing, to control external simulations. It should be noted that the FEMMT is continuously developed on GitHub [10] and welcomes open collaboration.

Abstract Code Interface for Thermal Simulations

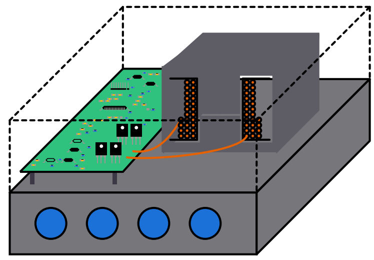
The programming interface of ONELAB is used to automatically create geometries as well as to assign materials. In [9] this is shown how to draw the core including windings, assign the currents, start the simulation and read back the results. For this purpose, a separate code interface is designed, with which the geometries of the inductive elements are transferred with simple commands. This is where the current paper continues.

To be able to describe a thermal simulation, the geometries of the insulations as well as their material properties are necessary. A standardized model including thermal parameters is shown in Fig. 2a. To the rotationally symmetric structure, a winding body can be introduced, as well as insulation between primary and secondary winding. The winding window is filled with inner potting material, and the entire core can be filled with outer potting material. The boundary conditions are set at its outer surfaces. This corresponds, for example, to mounting the inductive element on a heat sink with water cooling as depicted in Fig. 2b.

The simulation results read back contain the power dissipation per turn as well as the power dissipation of the core. Both are now impressed into the thermal simulation. The power dissipation is assumed to be homogeneous for each individual turn as well as for the core. This simplifies the parameter transfer from magnetoquasistatic to thermal simulation.



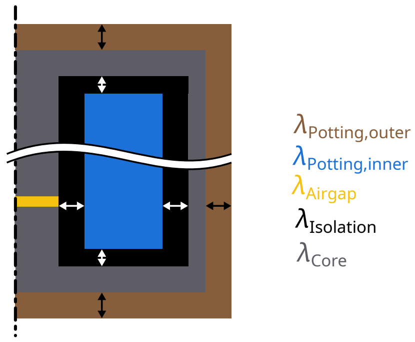
(a) Thermal material definitions



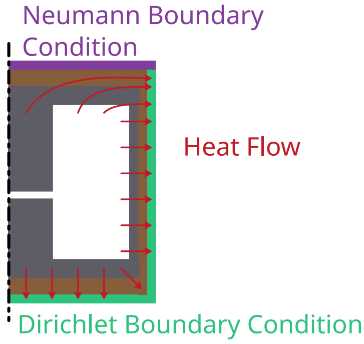
(b) Exemplary power converter based on a cooling plate with liquid cooling

Fig. 2: Cooling boundary condition options

The geometric parameterization is based on the previous work, published in [9]. The standard geometry for thermal parameterization is depicted in Fig. 3a. The following materials can be added: separate material for the air gaps, insulation inside the core (bobbin), inner potting material, and outer potting material. The thermal conductivities of the geometries generated there must be added (thermal conductivity of core, windings, insulations, and potting material). Outside of the core, the thickness of the potting material can be set individually for each side. Fig. 3b shows the parameterizable boundary conditions of



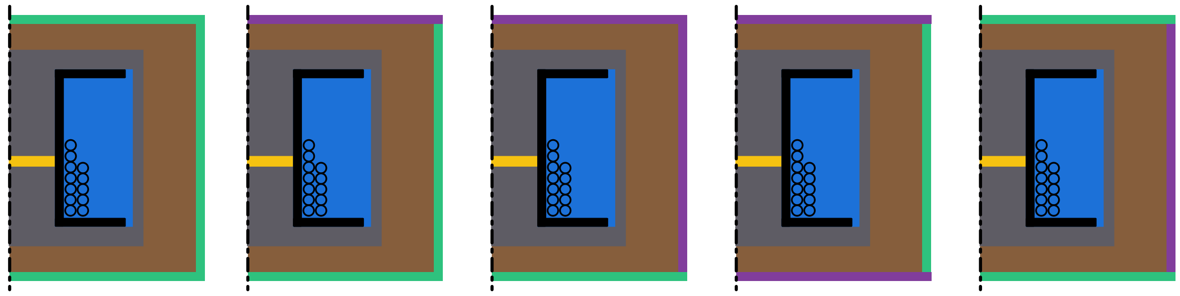
(a) Core parameterization options



(b) Boundary condition parameterization options

Fig. 3: Thermal parameterization options

the solver. The Dirichlet boundary condition sets the function values at the boundary, which corresponds to a fixed temperature. The Neumann boundary condition specifies an isolated environment which requires, e. g. no heat dissipation on that side. The type of boundary condition can be set via a flag in the program code. Possible examples are shown in Fig. 4.



(a) Full cooling

(b) Bottom- and side cooling

(c) Bottom cooling

(d) Side cooling

(e) Top- and bottom cooling

Fig. 4: Cooling boundary condition options

4 Code Example

Listing 1 shows a code example to parameterize the thermal cooling structure. First, the thermal conductivities are described. Then the thickness of the potting material and the temperatures at the boundary conditions are set according to Fig. 3a and Fig. 3b. A flag can be used to determine whether the boundary condition is a Neumann boundary condition or a Dirichlet boundary condition. The inner insulation thickness is taken from the magnetoquasistatic simulation. Then the simulation is started. Please note that the code syntax may change as development progresses. The latest syntax can be found in [10].

This code generates the additional cooling structure inside the ONELAB simulation. The power loss is calculated for every single mesh cell.

```
thermal_conductivity_dict = {
    "air":        0.0263,# unit: W/(m*K)
    "case": {
        "top":    1.54, # unit: W/(m*K)
        "right": 1.54, # unit: W/(m*K)
        "bot":    1.54  # unit: W/(m*K)
    },
    "core": 5, # unit: W/(m*K)
    "winding":    400, # unit: W/(m*K)
    "air_gaps":  180, # unit: W/(m*K)
    "isolation": 0.42  # unit: W/(m*K)
}
# set distance from core to case
case_gap_top =    0.0020 # unit: m
case_gap_right = 0.0025 # unit: m
case_gap_bot =    0.0020 # unit: m

# set boundary temperatures
boundary_temperatures = {
    "value_boundary_top":     20, # unit: degree celcius
    "value_boundary_right":  20, # unit: degree celcius
    "value_boundary_bottom": 20  # unit: degree celcius
}

# 1: Dirichlet boundary condition
# 0: Neumann boundary condition
boundary_flags = {
    "flag_boundary_top":     1,
    "flag_boundary_right":  1,
    "flag_boundary_bottom": 1
}

# start thermal simulation (geo is the simulation object)
geo.thermal_simulation(thermal_conductivity_dict, boundary_temperatures,
    boundary_flags, case_gap_top, case_gap_right, case_gap_bot)
```

Thermal Simulation Details

In order to integrate the heat conduction in the 2D-rotationally symmetric case into the solver, the general heat conduction equation must be simplified and solved. The starting point is the fundamental law of heat conduction:

$$\dot{\vec{q}} = -\lambda(\rho, \vartheta, \vec{r}) \cdot \mathrm{grad}(\vartheta(\vec{r}, t)), \tag{1}$$

where ϑ describes the scalar temperature field which depends on time t and position $\vec{r}$, λ is the heat conductivity of the material and $\dot{\vec{q}}$ is the heat flow density.

In the following, the heat conduction equation derived from Fourier's law in [12] is assumed:

$$\rho \cdot c(\vartheta) \frac{\partial \vartheta}{\partial t} = \mathrm{div}[\lambda(\rho, \vartheta, \vec{r}) \cdot \mathrm{grad}(\vartheta)] + \dot{W}(\vartheta, \vec{r}, t). \tag{2}$$

This equation holds for an isotropic, incompressible material at rest, with material properties λ, c (specific heat capacity), and ρ (mass density of the material). Further, in an electro-magnetic context, the volumetric heat source density $\dot{W}$ equals the locally dissipated power density according to losses. For FEMMT, the following simplifications were made:

- Thermal conductivity λ depends on pressure only in gases and fluids (compressible materials) and thus, this value can be assumed as ρ-independent for solid materials [12].
- Thermal conductivity λ for typical materials (copper, aluminium, ferrite) is nearly temperature-independent as shown in [13], and thus, this value can be assumed as temperature-invariant for solid materials.
- This, thermal conductivity λ is partially homogeneous and can be extracted from the divergence operator.
- The heat source density $\dot{W}$ is assumed to be invariant to temperature and time.
- Only a steady-state temperature field is considered, since the internal heat sources are nearly time-independent due to the high thermal capacitances and the system does not change after settling. Thus, the left side of (2) is zero.

Thus, the heat conduction equation reduces to Poisson's equation:

$$\begin{aligned} 0 &= \lambda \cdot \mathrm{div}(\mathrm{grad}(\vartheta)) + \dot{W}(\vec{r}) \\ \Leftrightarrow 0 &= \lambda \cdot \Delta\vartheta + \dot{W}(\vec{r}) \\ \Leftrightarrow -\dot{W}(\vec{r}) &= \lambda \cdot \Delta\vartheta \end{aligned} \tag{3}$$

which is implemented in the solver. Finally, the starting point for the equations is the same as used in [7]. More details how to program the solver are shown in [8].

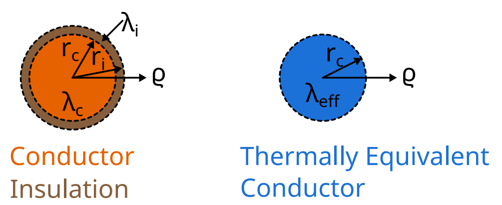
Conductor Insulation Simulation by an Equivalent Thermal Conductivity

Each conductor of an inductive component requires electrical insulation. Typically, the insulation has a much lower thermal conductivity compared to the electrical conductor. If this effect is to be modeled, a very fine mesh in each conductor insulation is necessary due to the small distances. To keep the mesh size, and thus the computation time low, an effective conductivity λ_{eff} of a thermally equivalent conductor is introduced. The equivalent conductor has the effective conductivity of a normal conductor, which consists of conductor and insulation, see Fig. 5a. The effective thermal conductivity is derived from the conductor radius r_{c}, the insulation radius r_{i}, the conductivity of the conductor λ_{c} and the conductivity of the insulation λ_{i} as (4):

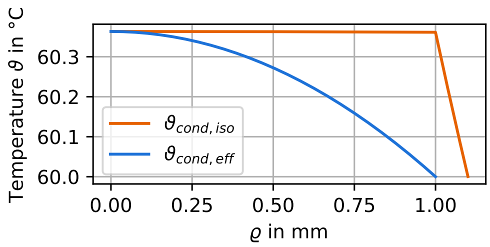
$$\lambda_{\mathrm{eff}} = \frac{1}{2 \cdot \frac{\ln(r_{\mathrm{i}}/r_{\mathrm{c}})}{\lambda_{\mathrm{i}}} + \frac{1}{\lambda_{\mathrm{c}}}}. \tag{4}$$

The core temperature of the thermally equivalent conductor corresponds to the core temperature of the real conductor (see Fig. 5b). For this example, a conductor with radius of $1.0\,\mathrm{mm}$ and an insulation radius of $1.1\,\mathrm{mm}$ is used. The conductor heat conductivity is given with $\lambda_{\mathrm{c}} = 400\,\frac{\mathrm{W}}{\mathrm{mK}}$ and the insulation heat conductivity is $\lambda_{\mathrm{i}} = 0.42\,\frac{\mathrm{W}}{\mathrm{mK}}$. The outer temperature is $60\,°\mathrm{C}$ and the dissipated electrical power density P_0 is given. For $10\,\mathrm{W}$ dissipated power in the conductor, the power density is $P_0 = \frac{P}{r_{\mathrm{c}}^2 \cdot \pi \cdot l} = \frac{10\,\mathrm{W}}{1\,\mathrm{mm}^2 \cdot \pi \cdot 1\,\mathrm{m}} = 3183\,\frac{\mathrm{kW}}{\mathrm{m}^3}$. The formula for the effective temperature is derived in [13]:

$$\vartheta_{\mathrm{cond,eff}}(\varrho) = -\frac{\varrho^2 P_0}{4\lambda_{\mathrm{eff}}} + \frac{r_{\mathrm{c}}^2 P_0}{2} \cdot \left(\frac{\ln(\frac{r_{\mathrm{i}}}{r_c})}{\lambda_{\mathrm{i}}} + \frac{1}{2\lambda_{\mathrm{c}}}\right) + T_0. \tag{5}$$



(a) Conductor with insulation is summarized to an thermally equivalent conductor



(b) Temperature distribution inside the conductor with insulation and the thermally equivalent conductor

Fig. 5: Description of an conductor with an effective thermally conductivity

A conductor made of stranded wires can be converted into a solid wire conductor beforehand with the help of the fill factor F. The conductor radius for stranded wires results to

$$r_{\mathrm{c}} = r_{\mathrm{i}} \cdot \sqrt{F}. \tag{6}$$

The core temperature of the stranded wire can then be calculated using (4) and (5).

This example shows, how little influence the insulation has on the temperature increase within the conductor ($0.37\,\mathrm{K}$). The influence is almost negligible. For a litz wire using the same insulation diameter and a fill factor $F = 0.5$, the temperature rise is $0.79\,\mathrm{K}$.

5 Verification

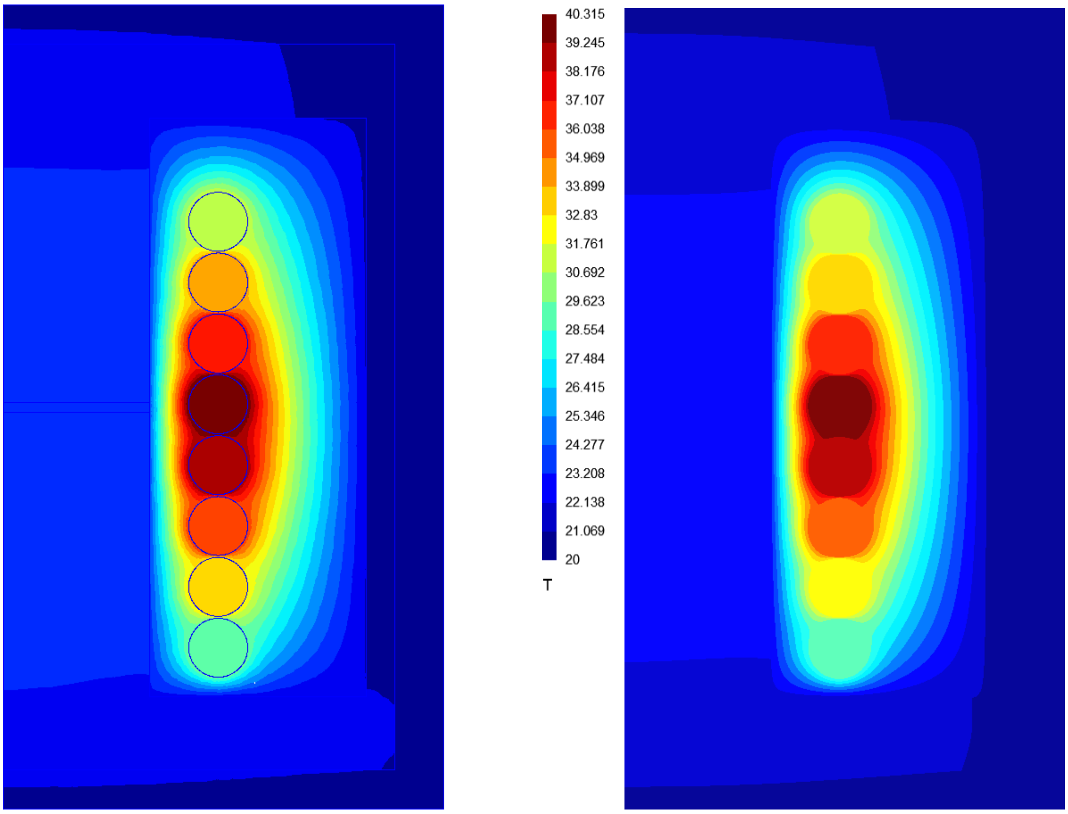
In a first step, the verification of the results is provided by a 2D rotationally symmetric reference simulation with the 2D FEM simulation tool FEMM to ensure the solver is implemented correctly. Further, the results are verified by measurements.

Verification by 2D Reference Simulation

To verify the automated simulation in FEMMT, a comparison simulation is set up in FEMM. A PQ40/40 N95 core is used for this purpose. The coil being investigated is set up by a winding with eight turns of solid wire with a radius of $1.5\,\mathrm{mm}$. The power dissipation of core and windings come from a magnetoquasistatic simulation performed before. A $3\,\mathrm{A}$ sinusodial peak current at $100\,\mathrm{kHz}$ flows through the winding. The boundary conditions on all sides are Dirichlet boundary conditions with a temperature of $20\,°\mathrm{C}$. Fig. 6a shows the validation simulation for FEMM, Fig. 6b the simulation with FEMMT. The temperature distribution and peak temperatures match well.

Temperature Measurement Validation

In order to verify the results in practice, an experimental setup is constructed. This is shown schematically in Fig. 7a and in reality in Fig. 8a. In [9] the power dissipation of a coil is determined calorimetrically and compared with the simulation. The same coil with the same data is used, extended by some temperature sensors which are distributed inside the turns (see Fig. 7b. The coil is placed in potting material. The temperature of the container is approximately constant due to the thick aluminum



(a) FEMM simulation

(b) FEMMT simulation

Fig. 6: Temperature field calculation using FEMM and FEMMT

walls. Thus, a Dirichlet boundary condition can be assumed. Since only heat conduction, but not convection, was considered in the simulation, a styrofoam is put over the arrangement, which reproduces the Neumann boundary condition (perfect thermal insulation). Cooling is provided by an overdimensioned heat sink with a pressure chamber and a fan to enable an almost constant heat sink temperature. With the assistance of a resonant circuit, excited by a half-bridge, a sinusoidal current is injected into the coil. In the steady state, the measured temperature field is compared with the simulated temperature field.

Fig. 8b shows the comparison between the FEMMT simulation and the measurement results of the lab setup. The sensors are named according to Fig. 7b.

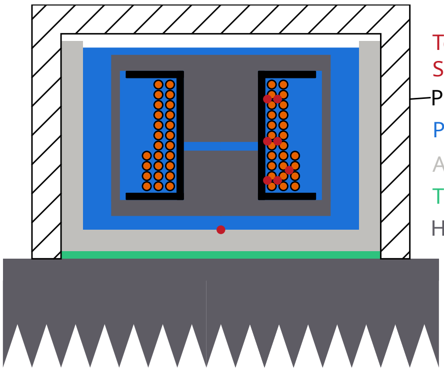
Even the simulation consideres a 2D rotationally symmetric inductor, what differs from the real PQ40/40 core, the results are matching well.

6 Using the Toolbox for Optimization

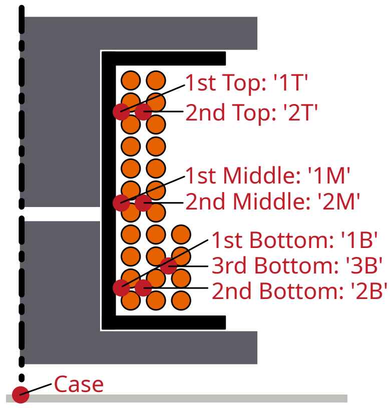
In the following, it is shown how the toolbox can be used to optimize magnetic components with regard to various criteria, see also Fig. 9. First, the user enters the possible geometry and current shapes for the magnetoquasistatic solver. The next step includes an analytical pre-calculation using a reluctance model to sort out parameter sets, which do not fit the goal-parameters (e.g., inductance values). The remaining geometries are built and simulated automatically with the help of the FEM module in ONELAB. Such an optimization is shown in [14]. In the next step, different cooling systems can be simulated and configurations with too high core- or conductor temperatures can be automatically sorted out.

7 Summary and Outlook

In order to easily create standard magnetic and thermal setups in FEM simulations, this paper presents an open-source FEM magnetics toolbox. The toolbox provides methods for drawing different cores, air

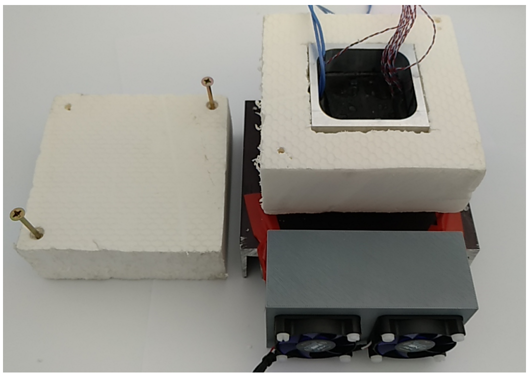


(a) Measurement and setup scheme

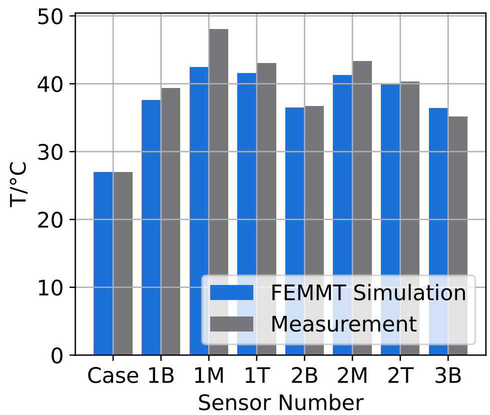


(b) Temperature sensor placement

Fig. 7: Inductor temperature verification setup



(a) Lab setup



(b) Comparison FEMMT simulation vs. measurement

Fig. 8: Inductor temperature verification measurement

gaps, winding schemes and winding types including its thermal behaviour. The built-in control of the FEM simulation tool is based on Python. The toolbox provides the basis to automate the design process of magnetic components with FEM simulations.

In the near future, the toolbox functions will be expanded in the direction of usability and optimization. Currently, work on a graphical user interface is in progress. An integrated material database will provide data on demand for certain standard materials, such as complex core parameters, electrical and thermal conductivities. A thermal equivalent circuit will be introduced according to [11]. Various optimization routines are to be integrated directly into the toolbox in order to be able to perform the design according to different criteria.

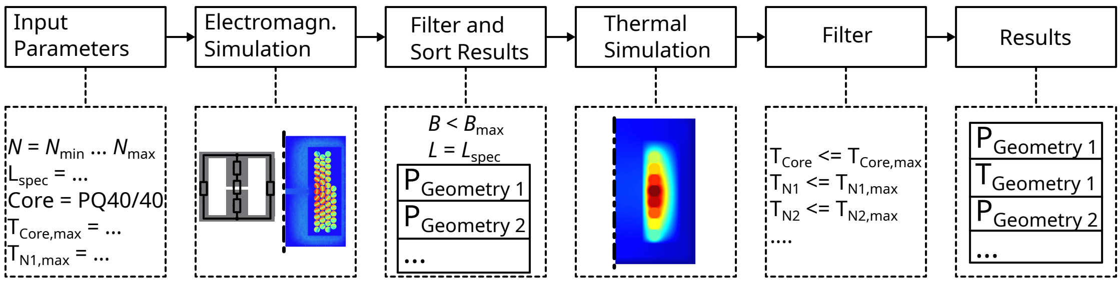


Fig. 9: Usage of FEMMT for an optimization

References

[1] R. Unruh, F. Schafmeister and J. Böcker, "11kW, 70kHz LLC Converter Design with Adaptive Input Voltage for 98 % Efficiency in an MMC," 2020 IEEE 21st Workshop on Control and Modeling for Power Electronics (COMPEL), 2020.

[2] M. J. Jacoboski, A. de Bastiani Lange and M. L. Heldwein, "Closed-Form Solution for Core Loss Calculation in Single-Phase Bridgeless PFC Rectifiers Based on the iGSE Method," in IEEE Transactions on Power Electronics, vol. 33, no. 6, pp. 4599-4604, June 2018, doi: 10.1109/TPEL.2017.2775106.

[3] J. B. Goodenough, "Summary of losses in magnetic materials," in IEEE Transactions on Magnetics, vol. 38, no. 5, pp. 3398-3408, Sept. 2002, doi: 10.1109/TMAG.2002.802741.

[4] J. Mühlethaler, "Modeling and multi-objective optimization of inductive power components", 2012.

[5] Christophe Geuzaine et al.: ONELAB, `https://onelab.info/`, Accessed Thursday 30th June, 2022.

[6] Alexandre Halbach: Sparselizard, `https://sparselizard.org/`, Accessed Thursday 30th June, 2022.

[7] David Meeker: FEMM, `https://www.femm.info`, Accessed Thursday 30th June, 2022.

[8] C. Geuzaine. „GetDP: a general finite-element solver for the de Rham complex“. In: PAMM Volume 7 Issue 1. Special Issue: Sixth International Congress on Industrial Applied Mathematics (ICIAM07) and GAMM Annual Meeting, Zürich 2007.

[9] Nikolas Förster, Till Piepenbrock, Philipp Rehlaender, Oliver Wallscheid, Frank Schafmeister and Joachim Böcker. "An Open-Source FEM Magnetics Toolbox for Power Electronic Magnetic Components". PCIM Europe 2022; International Exhibition and Conference for Power Electronics, Intelligent Motion, Renewable Energy and Energy Management. 2022.

[10] LEA Power Electronics and Electrical Drives: FEM Magnetics Toolbox, `https://github.com/upb-lea/FEM_Magnetics_Toolbox`.

[11] Manfred Albach: Induktivitäten in der Leistungselektronik, ISBN: 978-3-658-15080-8, 2017.

[12] Hans Dieter Baehr and Karl Stephan: Wärme- und Stoffübertragung, ISBN: 978-3-662-58441-5, 2019.

[13] Jonas Hölscher. "Automatisierte thermische Analyse ausgewählter Transformatoren und Induktivitäten". Bachelor Thesis. Paderborn University 2022.

[14] Till Piepenbrock. „Automated FEM Transformer Design for a Dual Active Bridge“. Master Thesis. Paderborn University 2022.