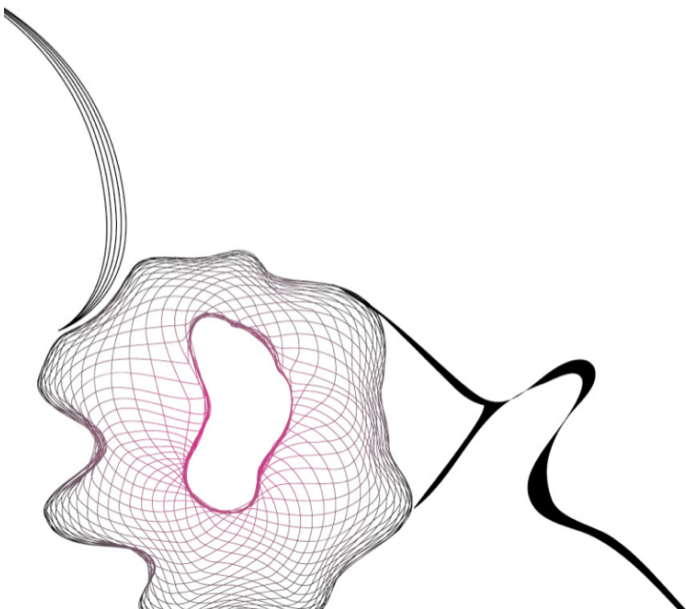


# UNIVERSITY OF TWENTE.

Faculty of Electrical Engineering,  
Mathematics & Computer Science

## Advanced Thermal Modelling of Magnetic Components

Alexandru Savca  
M.Sc. Thesis  
September 24, 2024



---

**Supervisors:**

dr. T. B. Grading<sup>1</sup>  
prof. dr. ir. P. Venugopal<sup>2</sup>

<sup>1</sup>System Integration and Physics Team  
Hitachi Energy Research  
Dättwil, Switzerland

<sup>2</sup>Power Electronics & EMC Group  
University of Twente  
Enschede, The Netherlands

---

# Preface

Before you lies the master thesis “Advanced Thermal Modelling of Magnetic Components.” It has been written to fulfill the graduation requirements of the Power Electronics program at the University of Twente in Enschede, The Netherlands. The thesis was pursued industrially at Hitachi Energy Research in Baden, Switzerland. I was engaged in researching and writing this thesis from March to September 2024.

During my previous projects I have realised that getting out of the comfort zone by exploring and touching new subjects and areas yields valuable experiences. This motivated me to challenge myself again during my graduation project and delve into topics that I was not acquainted with before. I had to learn a new subject from scratch, Finite Element Theory, get acquainted to a new software, COMSOL and strengthen my skills in MATLAB. Also, I have learned that struggling is part of the process. Therefore, this thesis has taught me valuable lessons both professionally and personally.

I would like to thank my supervisors, Dr. Thomas Gradinger and Dr. Prasanth Venugopal, for the excellent guidance, support and encouragement during the process. I would like to thank Ralph Burkart, Thiago Batista, and Ray Heuting for their effort in the official procedures of the collaboration between the institutions. I would also like to thank Arne Schröder for initially granting me the internship opportunity at Hitachi Energy which in turn allowed me to pursue the graduation project.

Finally, I want to thank my family and friends for being there for me. I would also like to thank you, my reader: I hope you enjoy your reading.

Alexandru Savca

Baden, August 1, 2024

# Summary

This thesis presents the development and implementation of advanced thermal models for magnetic components, aimed at achieving both high execution speed and sufficient accuracy. Traditional optimization tools often rely on thermal networks or 1D and 2D cross-section simplifications, which, despite their speed, suffer from limited accuracy in predicting magnetic core temperatures. To address this, magnetic core models utilizing coarse 3D finite-element (FE) models are introduced, developed and implemented in MATLAB. These models are tailored for two types of magnetic cores: bulk (e.g., ferrite) and tape-wound (e.g., amorphous and nanocrystalline). The models are rigorously benchmarked against fine mesh simulations conducted in COMSOL. The isotropic (bulk) models are able to predict volume and surface maximum temperatures as well as average temperatures within 1.25% and the anisotropic (tape-wound) models within 0.8%. Moreover, the execution time of these models is within 0.35 millisecond on a standard laptop (2.6 GHz), making them suitable for optimization processes.

This work significantly improves the accuracy of traditional thermal network models, which have strong trade-offs between accuracy and computational efficiency. The tailored finite-element method (FEM) approach bridges the gap between speed and accuracy, offering a robust and efficient solution for the thermal modeling of magnetic components. This thesis underscores the potential of these models to enhance the optimization of magnetic cores, and, thereby, magnetic components, providing valuable tools for researchers and engineers in the field. Future research may extend these models to components beyond the ones explored in this work and further refine their performance in complex design scenarios.

# Contents

<b>Preface</b>	<b>2</b>
<b>Summary</b>	<b>3</b>
<b>List of Acronyms and Symbols</b>	<b>6</b>
<b>1 Introduction</b>	<b>8</b>
1.1 Background . . . . .	8
1.1.1 Advancements of Magnetic Components . . . . .	8
1.1.2 Applications . . . . .	9
1.1.3 Magnetic Core Types . . . . .	9
1.1.4 Optimization . . . . .	10
1.2 Thermal Modeling . . . . .	11
1.3 Thesis Objectives . . . . .	11
1.4 Structure . . . . .	13
<b>2 Literature Review</b>	<b>14</b>
2.1 Models Based on Thermal Networks . . . . .	14
2.2 Models Based on Neural Networks . . . . .	15
2.3 FE Thermal Models . . . . .	17
2.4 Summary of Previous Literature . . . . .	18
<b>3 Theory</b>	<b>19</b>
3.1 Heat Transfer . . . . .	19
3.1.1 Conduction . . . . .	19
3.1.2 Convection . . . . .	19
3.1.3 Heat Diffusion Equation . . . . .	20
3.2 Numerical Methods . . . . .	21
3.2.1 FDM . . . . .	21
3.2.2 FEA . . . . .	21
3.2.3 FVM . . . . .	22
3.2.4 Suitability of FEM . . . . .	22
3.2.5 Residuals Method - Galerkin FEM Formulation . . . . .	23
<b>4 Methodology Development</b>	<b>25</b>
4.1 1D Elements for Heat Transfer with Heat Generation . . . . .	25
4.1.1 2-node Element . . . . .	25
4.1.2 3-node Element . . . . .	26
4.2 2D Elements for Heat Transfer with Heat Generation . . . . .	26
4.2.1 Theoretical Framework for 2D Elements . . . . .	28
4.3 Iso-parametric Formulation . . . . .	29

4.3.1	Element Integrals . . . . .	30
4.4	3D Elements for Heat Transfer with Heat Generation . . . . .	31
4.4.1	Theoretical Framework for 3D Elements . . . . .	32
4.5	Accuracy of the Formulated FEs . . . . .	36
4.5.1	Initial Tests . . . . .	36
4.5.2	18-node Symmetrical Element . . . . .	39
4.5.3	Non-Dimensional Parameter Accuracy Tests . . . . .	40
4.5.4	Comparison between the two Accuracy Testing Approaches . . . . .	41
<b>5</b>	<b>Model of Isotropic Transformer Cores</b>	<b>44</b>
5.1	COMSOL Benchmark . . . . .	44
5.2	Model Variants . . . . .	44
5.3	Test Cases . . . . .	45
5.4	Models' Accuracy . . . . .	49
5.5	Computational Efficiency . . . . .	50
5.5.1	Comparison of Different Models and Solving Algorithms . . . . .	50
5.5.2	Comparison with Alternative FEM Livelink COMSOL Model . . . . .	51
<b>6</b>	<b>Model of Anisotropic Transformer Cores</b>	<b>53</b>
6.1	Tape-Wound Cores . . . . .	53
6.1.1	COMSOL Benchmark . . . . .	53
6.1.2	3D Models . . . . .	55
6.1.3	2D Model . . . . .	58
6.1.4	Alternative 2D Benchmark . . . . .	59
<b>7</b>	<b>Discussion &amp; Conclusions</b>	<b>62</b>
	<b>References</b>	<b>65</b>
	<b>Appendices</b>	
<b>A</b>	<b>Appendix</b>	<b>68</b>
A.1	Stiffness Matrix Assembling Procedure . . . . .	68
A.2	Interpolation Functions . . . . .	69
A.3	Implementing 27-node in MATLAB . . . . .	70
A.4	Assembling Procedure for Model 2: Three-Element . . . . .	74

# List of Acronyms and Symbols

**Table 1:** List of acronyms

<b>Acronym</b>	<b>Expansion</b>
FEM	Finite Element Method
FE	Finite Element
FEA	Finite Element Analysis
FDM	Finite Difference Method
FVM	Finite Volume Method
BC	Boundary Condition
ODE	Ordinary Differential Equation
PDE	Partial differential equation
T	Temperature
HTC	Heat transfer coefficient
AC	Alternating Current
DC	Direct Current
SST	Solid State Transformers
LFT	Low Frequency Transformers
MFT	Medium Frequency Transformers
PCB	Printed Circuit Board
ANN	Artificial Neural Network

**Table 2:** List of symbols

<b>Symbol</b>	<b>Description</b>
$N$	Interpolation function
$T$	Temperature ( $^{\circ}C$ )
$P$	Power ( $W$ )
$q$	Heat Flux ( $W/m^2$ )
$k$	Thermal conductivity ( $W/mK$ )
$h$	Heat-transfer coefficient ( $W/m^2K$ )
$Q$	Volumetric loss density ( $W/m^3$ )
$c$	Material specific heat ( $J/kgK$ )
$\rho$	Material density ( $kg/m^3$ )
$A$	Area ( $m^2$ )
$L$	Length ( $m$ )
$[k^{(e)}]$	Element stiffness matrix
$\{f^{(e)}\}$	Element force vector
$w$	Element width, corresponding to x coordinate ( $m$ )
$h_e$	Element height, corresponding to y coordinate ( $m$ )
$l$	Element length, corresponding to z coordinate ( $m$ )
$a$	$w/2$
$b$	$h_e/2$
$c$	$l/2$

# Introduction

## 1.1 Background

Transformers have been foundational to the development and evolution of electrical power systems since their inception in the 19th century. Originally designed to enable the efficient long-distance transmission of electrical power, transformers have facilitated the widespread adoption of alternating current (AC) systems, thus shaping the modern electrical grid. Similarly, inductors, originally designed to store energy in magnetic fields and to regulate current in circuits, have played a vital role in filtering signals, managing power supply, and ensuring the stability of AC systems [1] [2].

### 1.1.1 Advancements of Magnetic Components

Line-frequency transformers (Fig.1.1(a)), developed during the rise of electrification in the late 19th century, operate on the principle of electromagnetic induction, as described by Michael Faraday. These transformers are designed to convert high-voltage electricity from power plants to a lower voltage suitable for commercial and residential use. Traditional transformers are passive devices, requiring no external control for operation and function at the standard utility frequencies of either 50 Hz or 60 Hz, depending on regional standards. Countries like the USA and most of Europe operate at these frequencies, with some countries like Japan using both due to historical reasons. Additionally, specialized applications like railway systems may use different frequencies, such as 16.7 Hz, to accommodate older infrastructure designed to minimize electrical arcing in DC machines [1].

The advent of solid-state technology has introduced a significant shift in transformer technology. Unlike traditional transformers, SSTs (Fig.1.1(b)) incorporate power electronics to actively control power flow, thus enabling the interconnection of different AC and DC systems. This is achieved through the use of medium/high frequency transformers (MFTs) that operate at higher switching frequencies independent of the grid frequency. These frequencies can be adjusted internally to optimize performance parameters like size, weight, and efficiency [1]. Inductors have also seen significant advancements in recent years, particularly with the rise of power electronics and optimization tools. Unlike traditional inductors, which are primarily passive components designed to store energy in a magnetic field and oppose changes in current, modern inductors are now being optimized for use in a variety of high-frequency applications. The development of advanced materials, such as nanocrystalline and amorphous alloys, has allowed for inductors with lower core losses and higher efficiency, making them ideal for use in power converters and other energy-efficient devices. The integration of these optimized inductors with SSTs has further enabled the development of more compact and efficient systems, bridging the gap between AC and DC power domains and paving the way for next-generation electrical grids [2] [3].





(a) Grid-frequency transformer.



(b) Modern optimized SST.

**Figure 1.1:** Representation of a grid-frequency transformer and an SST showing the difference in size.

### 1.1.2 Applications

While LFTs have proven extremely reliable and cost-effective, particularly in utility grids where efficiency and simplicity are paramount, SSTs offer enhanced control and flexibility. This makes them particularly suited for modern applications where dynamic load management and integration of renewable energy sources are required. SSTs can manage varying load demands and integrate different energy sources seamlessly due to their ability to control voltage and frequency dynamically.

In traction applications, such as railway systems, SSTs provide a distinct advantage due to their reduced size and weight, which is crucial in on-board applications. The flexibility offered by SSTs allows for more efficient operation within systems that still use older frequencies like 16.7 Hz. This technology has been embraced in several pilot projects by major industry players, demonstrating significant improvements in performance and efficiency.

Looking ahead, the integration of SSTs is set to play a crucial role in the development of smart grids, where power flow needs to be managed more dynamically to accommodate a mix of traditional and renewable energy sources, as well as new technologies like energy storage systems. SSTs' ability to quickly adapt to changes in load and generation offers a significant advantage in these complex systems [1].

### 1.1.3 Magnetic Core Types

Magnetic cores used in SSTs, MFTs and inductors are critical for efficient energy transfer and are typically of three types: bulk ferrites, tape-wound cores (amorphous and nanocrystalline), and stacked materials. Bulk ferrites are ceramic compounds that exhibit high magnetic permeability and low electrical conductivity, minimizing eddy current losses at higher frequencies. These properties make ferrites ideal for high-frequency applications in SSTs and MFTs [4] [5].

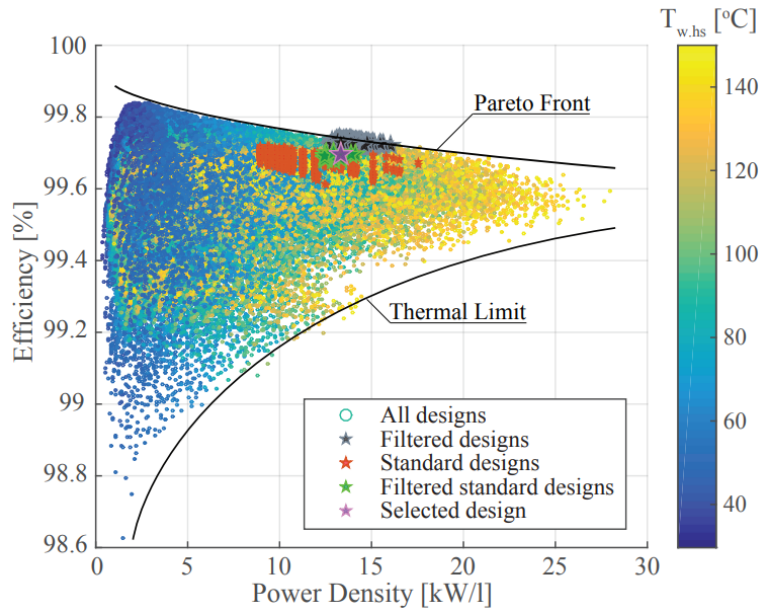
Tape-wound cores, including amorphous and nanocrystalline types, offer superior magnetic properties due to their unique microstructures. Amorphous cores, made by rapid cooling of molten metal, have a disordered atomic structure that reduces core losses and enhances performance at medium frequencies. Nanocrystalline cores, characterized by extremely fine crystalline grains, provide even lower losses and higher saturation magnetization, making them suitable for compact and highly efficient transformer designs [5].

Stacked cores, traditionally made from silicon steel, are constructed by stacking thin sheets of metal to reduce eddy currents and hysteresis losses. While effective in reducing core losses at lower frequencies, advancements in material science have made stacked cores less favorable for high-frequency applications compared to ferrites and advanced tape-wound materials [4] [5].

Each of these core types offers distinct advantages and is selected based on specific requirements of frequency, efficiency, and thermal management in SST and MFT applications.

### 1.1.4 Optimization

Designing magnetic components involves a complex, multi-objective optimization process that must accommodate a variety of specifications, including magnetic, electrical, and thermal considerations. To effectively navigate this complexity, recent research has focused on developing robust design methodologies that compute a vast array of potential designs. This approach helps encompassing the entire design space, allowing for the selection of the most optimal design based on precise specifications. An example of an optimization algorithm result with 2 million MFT designs is shown in Fig. 1.2. Utilizing analytical and semi-analytical models, these methodologies facilitate rapid computation, which is critical for exploring extensive design variations. As a result, the selection of the optimal MFT design requires a high level of confidence in both the design tools and the underlying models, emphasizing the need for accuracy and comprehensive coverage of all relevant transformer aspects. Consequently, these models must be sufficiently precise and cover magnetic, electrical, and thermal aspects of the transformer. Notably, the thermal models require special attention, as they have not been as extensively researched or emphasized in existing studies with the application in magnetic optimization tools [6].



**Figure 1.2:** Optimization results (around two million designs) plotted as efficiency vs. volumetric power density with color map indicating winding hot-spot temperature determined with analytical model [1].

## 1.2 Thermal Modeling

Magnetic optimization tools critically depend on accurate thermal modeling of magnetic cores to enhance design efficiency. In most design optimization processes, thermal models based on electrical networks are utilized. These models compute temperatures at nodal points and via thermal resistance they are able to predict heat flow within devices. Despite their widespread application in various engineering fields, thermal network models often face a significant trade-off between computational speed and model accuracy. For instance, a thermal network model was employed in [6] that approximates 3D distributions, however consumes almost 97% of the total design optimization time. This is because the computational effort of solving the thermal network is approximately proportional to the third power of number of network nodes [6]. Although deemed "sufficiently" accurate for practical applications, these models typically achieve in the vicinity of 10% error in 3D considerations compared to fine mesh precise models [6]. Other works such as in [1] have developed thermal network models that perform with better accuracy with around 2% error, however they are limited to specific 2D assumptions and simplifications and certainly are not largely applicable to all optimization scenarios. The above examples have modelled isotropic cores. To the best of our knowledge, little research has been done for thermal models applied to anisotropic cores. Thermal networks have been applied to model tape-wound cores achieving approximately 6% errors using 2D simplifications [7]. Moreover, these models above are limited in only computing temperature rises and are not able to predict average and surface temperatures. The latter are important for liquid-immersed magnetic components because of the thermal limits of the insulation liquid. Thus, the limitations of the thermal networks can be summarized as follows:

- Temperature distribution inside components is not accurately captured.
- Uncertainty in what the nodal values represent (average or maximum).
- High trade-off between accuracy and computational time.

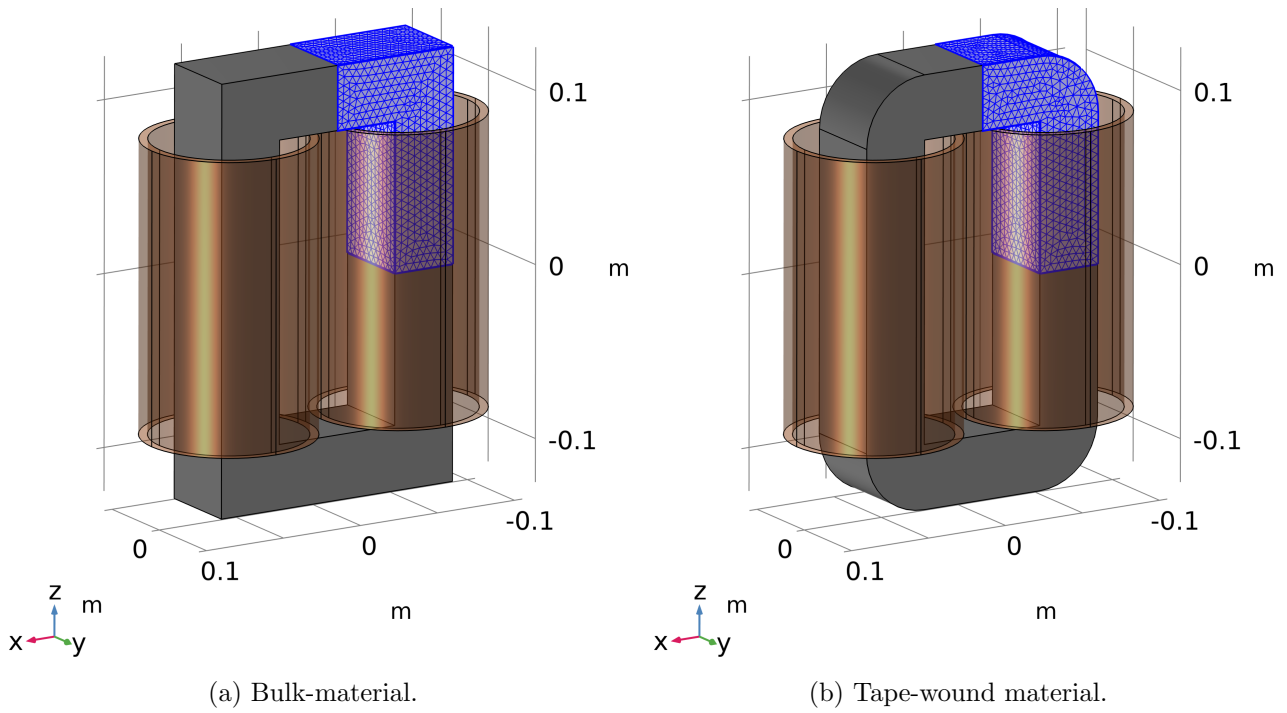
These limitations underscore the need for an innovative modeling approach that is fast, however more accurate and reliable than the present modelling methods.

Alternative thermal modeling techniques, such as the FE method and neural network-based models, have been successfully applied in various contexts beyond magnetic optimization tools. However, despite their advantages, these methods exhibit specific drawbacks that limit their suitability for optimization applications in the field of magnetic optimization. These limitations will be thoroughly examined in the literature review section of this study.

## 1.3 Thesis Objectives

This thesis introduces a novel approach to model bulk and tape-wound transformer cores using tailored FEs. An example of a bulk and tape-wound core and the defined coordinate system is illustrated in Fig. 1.3. This method aims to achieve a comprehensive thermal modeling that accounts for the complete temperature distribution within magnetic cores, offering both speed and precision. The model's performance will be benchmarked against a fine mesh model from COMSOL to ensure its accuracy. The main objectives of this thesis are:

- Develop a fast and accurate physical thermal model for magnetic components, focusing on magnetic cores.
- The implementation of the model should be done in MATLAB and the model should be validated using COMSOL.



**Figure 1.3:** Magnetic core with symmetry about  $x = 0$  and  $z = 0$ , and with meshed computational domain (quarter model, using symmetry) shown in blue.

- Achieve execution times in the order of 5 ms, with inputs including loss density, boundary conditions (BCs), material properties, and geometries, and outputs such as average temperature, maximum temperature, and maximum surface temperature.
- Explore core morphologies, ranging from bulk to tape-wound configurations, with main geometrical configurations being rectangular.

### Research Questions

To achieve these objectives, the research will focus on several key questions:

- What FE type is most suitable for thermal modeling in magnetic optimization tools for different components?

This question seeks to identify the optimal element type, such as hexahedral, multi-node elements, isoparametric elements, 1D, 2D or 3D elements that provide the best balance of accuracy and computational efficiency for modeling thermal behaviors in magnetic components.

- How can the chosen FE be formulated to enhance accuracy and computational speed?

This involves developing a mathematical formulation for the selected FE. The goal is to refine the element's formulation to improve simulation outcomes and efficiency, tailored specifically for thermal analysis in magnetic optimization contexts.

- How can the different FEs be assembled to accurately model a magnetic core?
- What are the verification strategies for assessing the accuracy and speed of the developed FE model in thermal predictions?

This question focuses on defining a systematic approach to verify the FE model's accuracy by comparing its results against established benchmarks such as those from COMSOL simulations. It includes identifying key metrics and scenarios for comparison.

- What improvements can be integrated into the FE model to enhance both its speed and accuracy for use in design optimization of magnetic components?

Beyond initial development and benchmarking, this question explores potential enhancements and optimizations to the FEM model that could further speed up calculations and refine accuracy, making it more effective for practical engineering applications.

- Can the FE models be applied universally across a range of core types like bulk ferrites to tape wound nanocrystalline and amorphous cores?

## 1.4 Structure

The report will be structured as follows:

- In Section 2 the current literature and the state of the art in thermal modeling will be reviewed. Also, the limitations of existing approaches and the reasons for choosing a novel approach will be discussed.
- In Section 3 the necessary theoretical background in heat transfer and finite element analysis (FEA) will be provided.
- In Section 4 a step-by-step model development, focusing primarily on formulating different FEs will be presented.
- In Section 5 the method will be applied to model isotropic magnetic cores.
- In Section 6 models of anisotropic cores will follow.
- In Section 7 the importance of the findings and future recommendations will be discussed. Also, conclusions will be drawn.

# Literature Review

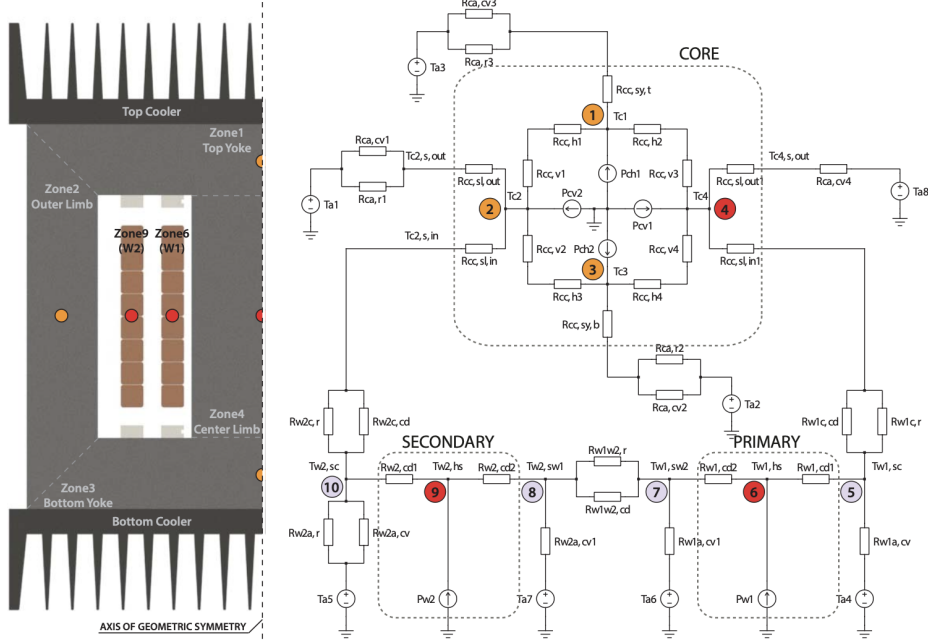
This section examines various techniques for thermal modeling of magnetic components, which are essential for optimizing performance in power electronic systems. The primary focus is on papers applying thermal models to the optimization of magnetic components. However, models with other applications such as planar magnetic components or electrical machines are also reviewed. The techniques for thermal modelling are categorized into three main types: thermal network models, neural network-based models, FEM models. Each category will be explored to highlight the main research, advantages and limitations. This review aims to place the thesis work within the broader research field, emphasizing the innovative approach of the new thermal modeling technique developed.

## 2.1 Models Based on Thermal Networks

A thermal model based on thermal networks has been employed in [1]. The approach involves modelling the MFT core and the windings by using 2 symmetrical axes. Thus, a 2D half MFT is obtained and shown in Fig. 2.1. In the next step, by assumptions, the critical nodal temperatures are placed in the 2D model. For instance, expected hot spots would be in the middle of the windings and in the middle of the centre limb. Finally, the heat transfer between these points and the outside world is modelled via resistances according to the heat transfer process occurring. It can be noticed in the schematic that the model accounts for convection at the boundaries and radiation between the windings and the core. The heat transfer coefficients used in this model are computed using empirical formulas. The limitations of this model is that the heat transfer in the 3rd dimension is not accounted for. Thus, one cannot account for various 3D cooling cases. Moreover, with a thermal network, one cannot capture the temperature distribution inside the components with a small number of nodes. Consequently, one cannot have an accurate estimation of the maximum temperatures and average temperatures independently.

A similar thermal network model was employed in [6]. Similar symmetry simplifications and assumptions are made, especially in the geometric placement of nodes within the network to represent potential hotspots and surface temperatures essential for accurate heat transfer calculations. From this investigation, another limitation of thermal network models is highlighted: the nodal temperatures are assigned based on assumed hotspot locations—often at the center of component cross-sections. However, there can be scenarios where actual hotspots do not align with these central positions, primarily due to varying surface convection conditions.

A novel approach to thermal modeling is employed in [8] which uses a combination of 2D thermal networks to construct a complex 3D thermal model. This model simplifies the 3D geometry into 2D planes, which are then vertically laminated and interconnected through thermal resistors to simulate heat transfer processes efficiently. The paper highlights the benefits of using such simplified thermal network models in terms of computational efficiency and ease of integration into the design process. However, it also acknowledges the limitations inherent



**Figure 2.1:** Partitioning of the MFT into zones for thermal modeling with respect to geometric symmetry (left) and the detailed static thermal network model of the MFT based on conduction, convection and radiation thermal resistances (nodes without heat injection - gray, expected hot-spot nodes for the given design - red) (right) [1].

in this approach, particularly the challenges in capturing the more complex 3D heat transfer effects with a model based primarily on 2D interactions.

A thermal model was developed in [9] that optimizes the design of planar PCB magnetic components, typically used in high power applications. The model introduced aims to achieve high accuracy with low computational times. This model utilizes purely analytical equations that do not require prior measurements or simulations. This approach significantly reduces computation time—up to 13 times faster than simplified finite element models. The model simplifies the 3D heat transfer challenges into a 2D analytical model. The simplification into a 2D model is justified by the use of high thermal conductivity materials and proximity to cooling solutions that significantly influence thermal resistance paths. Despite its accuracy and fast computational time, the model has limitations such as the exclusion of heat transfer in the third dimension, which could affect the modeling of diverse 3D cooling scenarios. Furthermore, its assumptions about hotspot locations and heat flow may not hold under varying operational conditions that deviate from the norm, potentially leading to inaccuracies in predicting maximum temperatures. Certainly, this model is also limited to highly thermally conductive materials and may not work for other types of materials.

Similar thermal network models have been studied and applied in [10], [11], [12], [13], [14], and [15]. Despite the large application of the thermal network models, the approach is limited by the oversimplifications and assumptions made which result in limited accuracy and high uncertainty for the design optimization tools.

## 2.2 Models Based on Neural Networks

A second thermal approach has been recently studied and is based on neural networks where, an algorithm is trained based on a dataset of simulations performed in a FEM software.

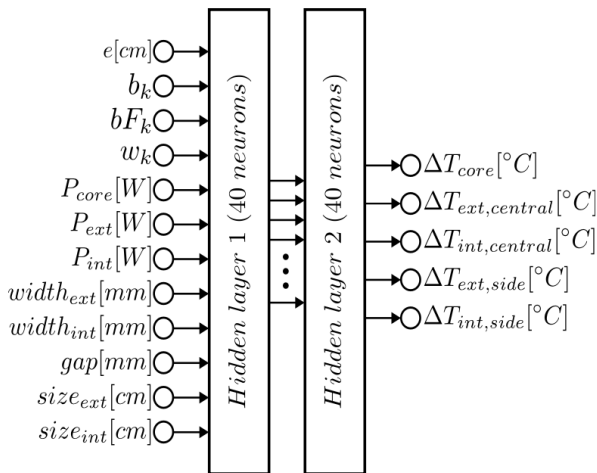
This approach was applied in [16] which uses an Artificial Neural Network (ANN) to analyze

a 50 kW MFT integrated into a Dual Active Bridge. This study bridges the gap between the high accuracy of computationally intensive FEM simulations and the quick application of simpler theoretical models. The research focuses on developing an ANN that is trained with FEM simulation data to reliably predict temperature increments in the transformer’s core and windings. This ANN, designed with a two-hidden-layer architecture (see Fig. 2.2(a)) and utilizing the ReLU activation function, is trained using data derived from FEM simulations. It focuses on predicting temperature increases in transformer components, employing the Adam optimizer and mean squared error for efficient learning. The ANN’s ability to deliver rapid and reliable temperature predictions significantly outpaces traditional FEM simulations, enabling faster iterative design and real-time operational adjustments. As can be seen in Fig. 2.2(b) the ANN model also performs better than the traditional thermal network models.

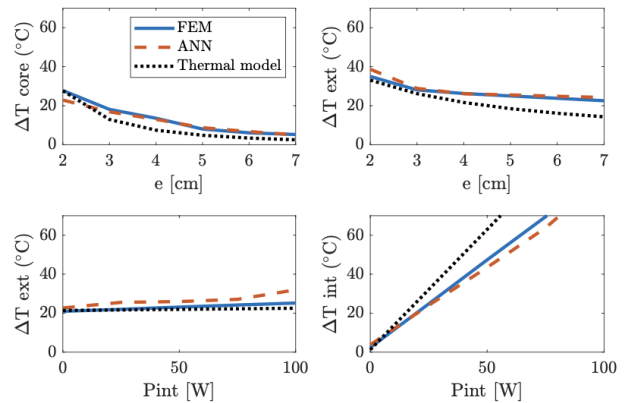
A similar approach was employed in [17] which explores the efficiency and practicality of using ANNs to model the thermal behavior of magnetic components, specifically focusing on medium frequency transformers. The research emphasizes achieving a balance between computational efficiency and the accuracy of thermal predictions necessary for designing power electronics converters. This study elaborates on various techniques to accelerate the generation of training datasets for ANNs without excessively compromising the model’s accuracy. For example, the authors examine the impact of removing fluid dynamics from simulations or ignoring radiation effects, assessing how these adjustments affect the accuracy of the resulting ANN models.

Neural networks have also been applied to thermal modelling in power electronics beyond optimization applications in [18], [19], [20], and [21].

The approach utilizing ANNs shows promising outcomes for application in magnetic design optimization. However, there are two limitations with this method, leading to the conclusion that the physical approach employed in this thesis is more suitable for optimization applications. First the initial training phase is resource-intensive, requiring a substantial amount of data from detailed FEM simulations to ensure comprehensive coverage of potential operating scenarios. It is impossible to cover the huge design space of the optimization tools. Even in this case there can be the situation that a certain design is not covered by the training dataset. In this case the model will produce higher errors and it would require a retraining of the data. On the other hand, a physical model would have the advantage to be more flexible regarding the geometrical aspects of the design as well as the boundary conditions being able to cover the whole design space of the optimization tools.



(a) Chosen architecture for the ANN.



(b) ANNs performance compared to an analytical thermal model.

**Figure 2.2:** Thermal model example based on ANN [16].

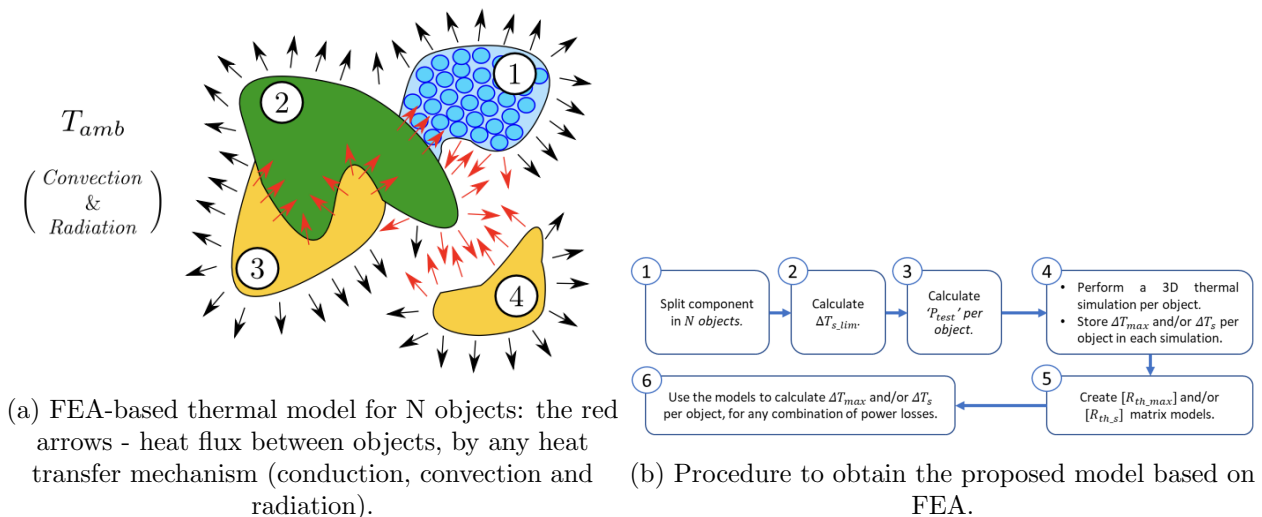


## 2.3 FE Thermal Models

The last category of thermal models involve FE models using FE softwares.

This model in [22] is structured to calculate both the maximum and the average surface temperature of any object within a system, like inductors or transformers, under any given operating condition. The procedure to develop the model is shown in Fig. 2.3(b). It begins with the proposition of modeling a single magnetic component in a system, using FEA to estimate thermal behavior under steady-state conditions. The approach includes splitting the component into representative objects, defining maximum surface temperatures, and calculating a test power loss ( $P_{test}$ ) for each object, necessary for subsequent thermal simulations. For each object, a 3D FE simulation is run, with the object experiencing power losses as per  $P_{test}$ . This allows for the assessment of maximum temperature rise and mean surface temperature rise. It is important to note that while the entire magnetic component is simulated, losses are only applied to one object at a time to isolate its thermal response. The thermal behavior is translated into a coefficients matrix similar to a 'thermal resistance' matrix. This matrix does not directly correspond to physical thermal resistance; rather, it is an expression calculated from the FEA simulation results that reflect the temperature rise over power loss. This model captures the interaction between different objects, including self-heating effects and heat flux interactions as shown in Fig. 2.3(a) The matrix model allows for the calculation of temperature rise for any combination of power losses. Although the model performs fast and very accurately it has the limitation of not being flexible. For instance, the matrix model would have to be re-tuned for each boundary condition situation. Also, it is not clear how accurate the model is for different designs and geometries and if the model has to be re-tuned for various geometries as well.

Another example of a FE model was employed in [23]. The primary use of the thermal model in this research is to simulate and understand the temperature distribution within the transformer under normal operating conditions. Similar thermal modelling studies have been employed in [24], [25], [26], and [27]. These models are based on fine meshes and standard elements which have a high computational time. This makes them unsuitable for optimization tools applications where the design space is usually in the order of hundred of thousands up to several millions.



**Figure 2.3:** Resistance matrix model based on detailed FEM simulations [22].

## 2.4 Summary of Previous Literature

The previous key literature work is summarized and compared for easy referencing in Table 2.1.

**Table 2.1:** Summary table of previous key literature

S/N	Technique Used	Application area	Comp. Time (s)	Highest Errors
1	ANN + 3D FEM [16]	MFT, ferrite core	N.A.	1.8%
2	2D Thermal Network Model + 3D FEM [1]	SST	N.A.	2%
3	High granularity 2D-thermal networks + Experiments [8]	Wireless Charging IPT systems	10	2-3%
4	2D Thermal Network Model + 3D FEM [7]	SST, tape-wound core	N.A.	6%
5	Thermal Matrix + 3D FEM & Experiments [22]	Magnetic Components (windings, cores etc.)	N.A.	-6.4-3.9%
6	3D Thermal Network Model + 3D FEM [6]	SST	70	7.14%
7	Discrete 2D layers to detailed 3D thermal + 3D FEM [28]	Transformer Magnetics	3-6 (coupled mag. +thermal)	3% (sum of losses) -11% (uniform loss dis.)
8	Thermoelectric FEM modelling [29]	Power Transformer	N.A.	11%
9	CFD + Analytical expressions [30]	MHz Planar Magnetics	N.A.	21.4% (secondary windings)

# Theory

In this section the relevant theory is presented for heat transfer principles and numerical methods.

## 3.1 Heat Transfer

Heat transfer theory focuses on understanding how energy moves due to temperature differences within or between substances. It identifies three main modes: conduction, convection, and radiation. Conduction occurs in solids or stationary fluids where a temperature difference is present, causing energy to flow from high to low temperature. Convection involves heat transfer between a surface and a moving fluid, where temperature differences drive energy exchange. Radiation refers to energy emission from surfaces in the form of electromagnetic waves, with net heat transfer occurring between surfaces at varying temperatures [31].

### 3.1.1 Conduction

Conduction, in particular, is the transfer of energy through particle interactions, where more energetic particles pass energy to less energetic ones. This mode is best understood at the atomic and molecular level, as collisions between particles enable energy diffusion from higher to lower temperatures. In solids, conduction involves lattice vibrations, while in electrical conductors, free electron movement also plays a role. Fourier's law provides a mathematical framework for conduction: in a one-dimensional plane wall, the heat flux ( $q_x''$  in  $W/m^2$ ) is directly proportional to the temperature gradient ( $\frac{dT}{dx}$ ) and the thermal conductivity of the material ( $k$  in  $W/(mK)$ ). The minus sign is a consequence of the fact that heat is transferred in the direction of decreasing temperature [31].

$$q_x'' = -k \frac{dT}{dx} \quad (3.1)$$

### 3.1.2 Convection

Convection is a mode of heat transfer that involves two simultaneous mechanisms: energy diffusion via random molecular motion and energy transport through the bulk, or macroscopic, movement of a fluid. In convection, fluid particles move collectively due to a combination of random and bulk motion. This cumulative process is termed convection, while advection refers specifically to the transport caused by the bulk fluid flow. When a fluid is in motion over a surface and there is a temperature difference between the surface and the fluid, heat transfer via convection occurs. A region of the fluid known as the hydrodynamic or velocity boundary layer forms due to the interaction between the fluid and the surface. In this boundary layer, the fluid velocity varies from zero at the surface to a finite value in the outer flow. Additionally, if there is a temperature gradient, a thermal boundary layer forms where the temperature ranges

from that of the surface to that of the outer flow. As long as there is a temperature difference between the surface and the outer flow, heat transfer will occur [31].

The contribution of molecular motion, known as diffusion, is dominant close to the surface where the fluid velocity is relatively low. At the surface-fluid interface, where the velocity is zero, heat transfer happens exclusively through diffusion. In the boundary layer itself, bulk fluid motion carries heat downstream, eventually transferring it to the fluid outside.

Convection heat transfer can be classified based on the type of fluid flow. Forced convection occurs when an external mechanism, such as a fan or pump, drives the flow. For instance, a fan cooling electronic components on a printed circuit board stack is an example of forced convection. On the other hand, free or natural convection occurs due to buoyancy forces driven by density differences in the fluid caused by temperature variations. A typical scenario is the upward airflow caused by heating elements in a vertical array of circuit boards, where the warmer, less dense air rises and is replaced by cooler ambient air.

In reality, mixed convection frequently occurs. This combination of forced and natural convection can arise when weak forced flow coexists with strong buoyancy forces, leading to a secondary flow that can significantly impact heat transfer rates.

In convection, heat transfer is governed by Newton's law of cooling, which states that the convective heat flux is proportional to the difference between surface and ambient, plenum, average or inlet temperatures (depending on the definition of the heat-transfer coefficient (HTC)). The convection HTC ( $h$  in  $W/(m^2K)$ ) depends on fluid flow conditions, surface geometry, and fluid properties, and is key to determining heat transfer rates. Typical values range from  $2 - 100,000 W/(m^2K)$ , depending on the type of convection and fluid used [31].

$$q_x'' = h(T_s - T_\infty) \quad (3.2)$$

In this study, forced air convection will only be considered which usually has a maximum of around  $100 W/(m^2K)$ .

### 3.1.3 Heat Diffusion Equation

A key objective in heat conduction analysis is to determine the temperature field within a medium based on the conditions imposed on its boundaries (i.e. Dirichlet or convective boundary conditions (BCs)). This temperature distribution reveals how temperature changes across different points in the medium, allowing the calculation of the conduction heat flux at any location within the medium or on its surface using Fourier's law.

To determine the temperature distribution, the principle of energy conservation is applied to a differential control volume defined within the medium. By identifying relevant energy transfer processes and using appropriate rate equations, a differential equation is formed, which provides the temperature distribution for given boundary conditions. For a stationary, homogeneous medium where the temperature varies in Cartesian coordinates, a small control volume is considered. The conduction heat rates across the surfaces of this volume are then expanded using a Taylor series to assess the heat rate changes in all three spatial directions.

The analysis also involves evaluating internal energy storage and thermal energy generation within the medium. Energy storage accounts for the rate of thermal energy change due to temperature fluctuations, while energy generation, expressed as a volumetric rate, reflects conversion processes like chemical, electrical, or nuclear transformations. By combining these factors with the conduction rates obtained from Fourier's law, a differential equation known as the heat diffusion equation is formulated [31].

$$\frac{\partial}{\partial x}(k_x \frac{\partial T}{\partial x}) + \frac{\partial}{\partial y}(k_y \frac{\partial T}{\partial y}) + \frac{\partial}{\partial z}(k_z \frac{\partial T}{\partial z}) + q = \rho c_p \frac{\partial T}{\partial t}, \quad (3.3)$$

where  $q$  is the volumetric heat generation rate in  $W/m^2$ . For steady state problems the last term on the right is set to 0 [31].

The thermal modelling approach in this work will involve solving the heat diffusion equation in steady state with FEM theory and convective flux BCs.

## 3.2 Numerical Methods

### 3.2.1 FDM

In numerical analysis, finite-difference methods (FDM) are a class of numerical techniques for solving differential equations by approximating derivatives with finite differences. Both the spatial domain and time domain (if applicable) are discretized, or broken into a finite number of intervals, and the values of the solution at the end points of the intervals are approximated by solving algebraic equations containing finite differences and values from nearby points.

Finite difference methods convert ordinary differential equations (ODE) or partial differential equations (PDE), which may be nonlinear, into a system of linear equations that can be solved by matrix algebra techniques [32].

### 3.2.2 FEA

The FEM or FEA, is a computational technique widely used to approximate solutions to boundary value problems in engineering. A boundary value problem is a mathematical problem where one or more dependent variables must satisfy a differential equation within a known domain and adhere to specific conditions at its boundaries. These problems, often called field problems, are defined over a field that usually represents a physical structure, and the field variables are the dependent variables governed by the differential equation. The boundary conditions specify values of these field variables or their derivatives at the boundary of the field. Depending on the nature of the problem, field variables can include displacement, temperature, heat flux, and fluid velocity [33].

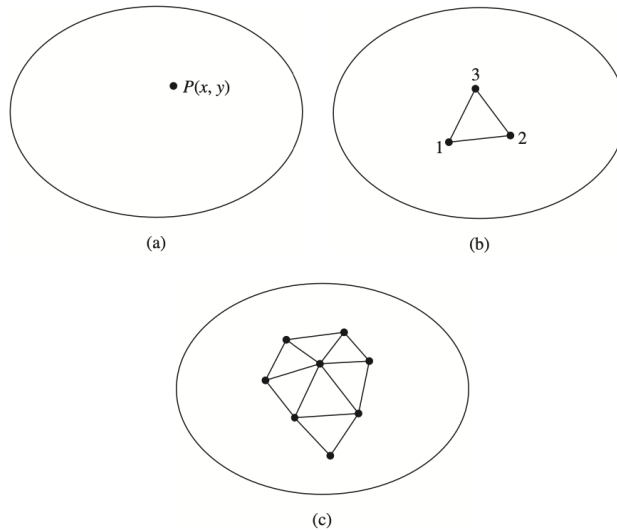
In FEM, the domain of a boundary value problem is discretized into a finite number of smaller subdomains known as elements. These elements collectively form a mesh that approximates the solution over the entire field. Each element is characterized by a set of nodes, which are specific points where field variable values are explicitly calculated. The nodes on the element's boundary connect it to neighboring elements, ensuring that the field variable values at shared nodes remain consistent across elements, thereby maintaining continuity throughout the mesh [33].

A simple two-dimensional domain with a single field variable  $\phi(x, y)$  can be considered (see Fig. 3.1). Each finite element node is assigned an interpolation function (also known as a shape function), which approximates field variable values within the element using known values at the nodes. For instance, a triangular element with three nodes may have a field variable  $\phi$  approximated as:

$$\phi(x, y) = N_1(x, y)\phi_1 + N_2(x, y)\phi_2 + N_3(x, y)\phi_3, \quad (3.4)$$

where  $\phi_1, \phi_2$ , and  $\phi_3$  represent the values at the nodes, and  $N_1, N_2$ , and  $N_3$  are the shape functions that interpolate the values between nodes. The nodal values of the field variable are unknowns determined by solving a system of equations that satisfy the governing differential equation across the domain. The shape functions, usually polynomial expressions, are predetermined based on the nodal arrangement and ensure smooth approximation of the field variable within each element [33].

In comparing the FEM with the FDM, it is important to note that in FEM, the variation of the field variable across the physical domain is inherently integrated into the procedure. With predefined interpolation functions, this variation is explicitly determined for the entire finite



**Figure 3.1:** (a) A general two-dimensional domain of field variable  $\phi(x, y)$ . (b) A three-node finite element defined in the domain. (c) Additional elements showing a partial finite element mesh of the domain [33].

element as part of the problem formulation. In contrast, FDM calculates the field variable only at specific points [33].

### 3.2.3 FVM

In the finite volume method (FVM), volume integrals in a partial differential equation that contain a divergence term are converted to surface integrals, using the divergence theorem. These terms are then evaluated as fluxes at the surfaces of each finite volume. Because the flux entering a given volume is identical to that leaving the adjacent volume, these methods are conservative. The method is used in many computational fluid dynamics packages. "Finite volume" refers to the small volume surrounding each node point on a mesh.

The disadvantage of finite volume in this work is that in order to represent parabolic distribution one would required several elements (because the values in FVM are stored in the centre of volumes). Even then, the interpolation functions are not an inherent part of the problem as seen in FEM. Moreover, the interpolation methods to construct parabolic distributions in FVM are complicated and less-clear to implement [34] [35].

### 3.2.4 Suitability of FEM

Loss-generating regions in magnetic components can often be modeled with good accuracy with a homogeneous volumetric loss density. This applies both to core and windings. In some cases, further simplification is possible because the heat transfer in one direction dominates that in the other directions such that a 1D problem is obtained (see Fig. 1.3(a)). An example is a winding region which has the shape of a thin hollow cylinder, where the heat transfer in radial direction dominates. If we further assume that the radial thickness is small compared to the mean radius of the cylinder, we can neglect the curvature and obtain a 1D planar problem governed by the Poisson equation that describes heat conduction and generation:

$$\frac{\partial}{\partial r} \left( k \frac{\partial T}{\partial r} \right) + Q_{dot} = 0. \quad (3.5)$$

Here,  $T(r)$  is the temperature,  $r$  the radial coordinate, and  $Q_{dot}$  is the (homogeneous) volumetric loss density. Eq. 3.5 yields a parabolic temperature distribution, which is typical for this type

of problem. In case of axial symmetry or 2D problems, the type of temperature distribution is usually similar, i.e., parabolic or close to parabolic. It is this feature that triggers the idea of the suitability of FEM for this work: In case of a (near) parabolic temperature distribution across a loss-generating region, it should be possible to represent that region with high accuracy with a single quadratic finite element over the cross-section. In modelling the core, a 2D second-order finite element representing the core cross-section would allow to solve the heat-conduction and -generation problem with BCs for the heat transfer from surface to ambient. However, usually there is an additional complication: the core is not cooled equally along the magnetic path. For example, the yoke may be cooled better than the legs or vice versa. This would not only require multiple "representative" cross-sections to be modeled with 2D elements, but also an account of the heat transfer along the magnetic path. For example, heat from poorly cooled legs might be transferred to the yokes or vice versa. If an O-core is considered as shown in Fig. 1.3(a) and assume symmetry about  $x = 0$ , a near-parabolic temperature profile in the yoke can be expected. Then, modeling the yoke with a single element in  $x$ -direction should be fairly accurate. The same holds for the leg in case of symmetry about  $z = 0$ . It should be possible to represent the entire core with two or three (depending on the treatment of the corner) 3D quadratic elements. This idea is the basis of the presently studied extremely coarse FE model that is fast enough to be affordable in optimization.

As previously described FDM and FVM are not suitable for this work's purpose. FDM is not suitable for complex geometries and has a point-wise approximation. FVM is more suitable for flow problems where local conservation of quantities is important for accurate results. Similarly, both FDM and FVM do not possess clearly defined interpolation techniques. On the other hand, FEM, offers a range of FEs that allow the choice of the approximation degree thanks to the interpolation techniques. This is the key reason that makes FEM suitable for the application in this work.

### 3.2.5 Residuals Method - Galerkin FEM Formulation

In applying FEM theory to develop the modelling approach a suitable method to formulate finite elements is required. Such a method translates a differential equation into a finite element formulation with certain BCs. This is illustrated in Table 3.1.

**Table 3.1:** Example of a BC problem translated into a FE formulation, see Table 2 for symbols explanations.

Differential Equation Example	FE Formulation Using Galerkin Method
$k_x \frac{d^2 T}{dx^2} + Q = \frac{hP}{A}(T - T_a)$	$k_x A \int_{x_1}^{x_2} \left[ \frac{dN}{dx} \right]^T \left[ \frac{dN}{dx} \right] \{T\} dx + hP \int_{x_1}^{x_2} [N]^T [N] \{T\} dx$ $= A \int_{x_1}^{x_2} Q [N]^T dx + hP T_a \int_{x_1}^{x_2} [N]^T dx + k_x A [N]^T \frac{dT}{dx} \Big _{x_1}^{x_2}$

There are different methods of formulation and they are categorized in two groups:

- Variational methods – approximation is based on the minimization of a functional; Rayleigh-Ritz Method Weighted

- Residual Methods – start with an estimate of the solution and demand that its weighted average error is minimized; Galerkin Method, Least Square Method, Collocation Method, Subdomain Method, Pseudo-spectral Methods.

For this work it does not play a role which method of formulation is chosen because the application of FE is very simple. The choice of between such methods plays a more important role in more in-depth application of FEM particularly in softwares such as COMSOL. It was chosen by preference that the element formulation is done via Galerkin Residuals Method. The advantage of this method is that is mathematically sound and in understanding it one does not require previous knowledge of variational calculus principles as it is for the other methods.

In the following section different elements will be formulated using this method. Explaining the intermediate steps and the theory behind this method is out of the scope of this work. Only the final results will be showcased, however for the interested reader the full theory can be found in [33].



# Methodology Development

In this section the methodology of FE formulation is illustrated. The theoretical framework is shown for the 1D, 2D and 3D cases. Relevant finite elements are implemented in MATLAB and compared to equivalent COMSOL fine-mesh benchmark.

## 4.1 1D Elements for Heat Transfer with Heat Generation

### 4.1.1 2-node Element

For 1D heat conduction with spatially constant thermal conductivity  $k_x$  and volumetric loss density  $Q$ , the following is obtained from Fourier's law:

$$k_x \frac{d^2 T}{dx^2} + Q = 0 \quad (4.1)$$

Next, the Galerkin FEM is applied to obtain the element equations [33]. This is applied to a 2-node 1D linear element. The temperature distribution for such an element can be expressed as:

$$T(x) = N_1(x) T_1 + N_2(x) T_2 \quad (4.2)$$

, where  $T_1$  and  $T_2$  are the temperatures at node 1 and 2 respectively. The Galerkin FEM results in the following residual integrals:

$$\int_{x_1}^{x_2} (k_x \frac{d^2 T}{dx^2} + Q) N_i(x) A dx = 0$$

After integrating by parts, the first term of the equation (for the interested reader, the full steps are shown in [1]), the following is obtained.

$$k_x A \int_{x_1}^{x_2} \frac{dN_i}{dx} \frac{dT}{dx} dx = A \int_{x_1}^{x_2} Q N_i(x) dx + k_x A N_i(x) \frac{dT}{dx} \quad i = 1, 2 \quad (4.3)$$

This equation is of the form:

$$[k] \{T\} = \{f_Q\} + \{f_g\} \quad (4.4)$$

For the very simple 2-node element with the interpolation functions  $N_1(x) = 1 - x/L$  and  $N_2(x) = x/L$  the following results are obtained:

$$[k] = \frac{k_x}{L} \begin{bmatrix} 1 & -1 \\ -1 & 1 \end{bmatrix} \quad (4.5)$$

$$\{f_Q\} = \left[ \begin{array}{c} \frac{QL}{2} \\ \frac{QL}{2} \end{array} \right] \quad (4.6)$$

$$\{f_g\} = k_x \left[ \begin{array}{c} \left. \frac{-dT}{dx} \right|_{x_1} \\ \left. \frac{dT}{dx} \right|_{x_2} \end{array} \right] \quad (4.7)$$

When multiple elements are joined the internal element gradient values in (4.7) are zero (due to opposite sign). At the boundaries the heat fluxes are usually specified, or they are computed using the fixed temperatures or convection flux conditions.

A 1D element with fixed temperature BCs is implemented in MATLAB and the results are shown in Fig. 4.1. The computed nodal values are exact. However, due to the linear nature of the interpolation functions, the interpolated FEM solution differs from the exact solution. The exact solution in (4.8) is determined by integrating (4.1) twice:

$$T(x) = -\frac{Q}{2k}x^2 + \left(\frac{q}{2k}L + \frac{T_2 - T_1}{L}\right)x + T_1 \quad (4.8)$$

With two elements, one can very roughly approximate the parabolic shape. The FEM solution is seen to converge toward the analytical one as the mesh is made finer, which suggests that the FEM formulation is correct, 4.1(b).

### 4.1.2 3-node Element

A 1D 3-node element is a quadratic element. The interpolation functions for such an element are [33]:

$$\begin{aligned} N_1 &= (2s - 1)(s - 1) \\ N_2 &= -4s(s - 1) \\ N_3 &= s(2s - 1) \end{aligned} \quad (4.9)$$

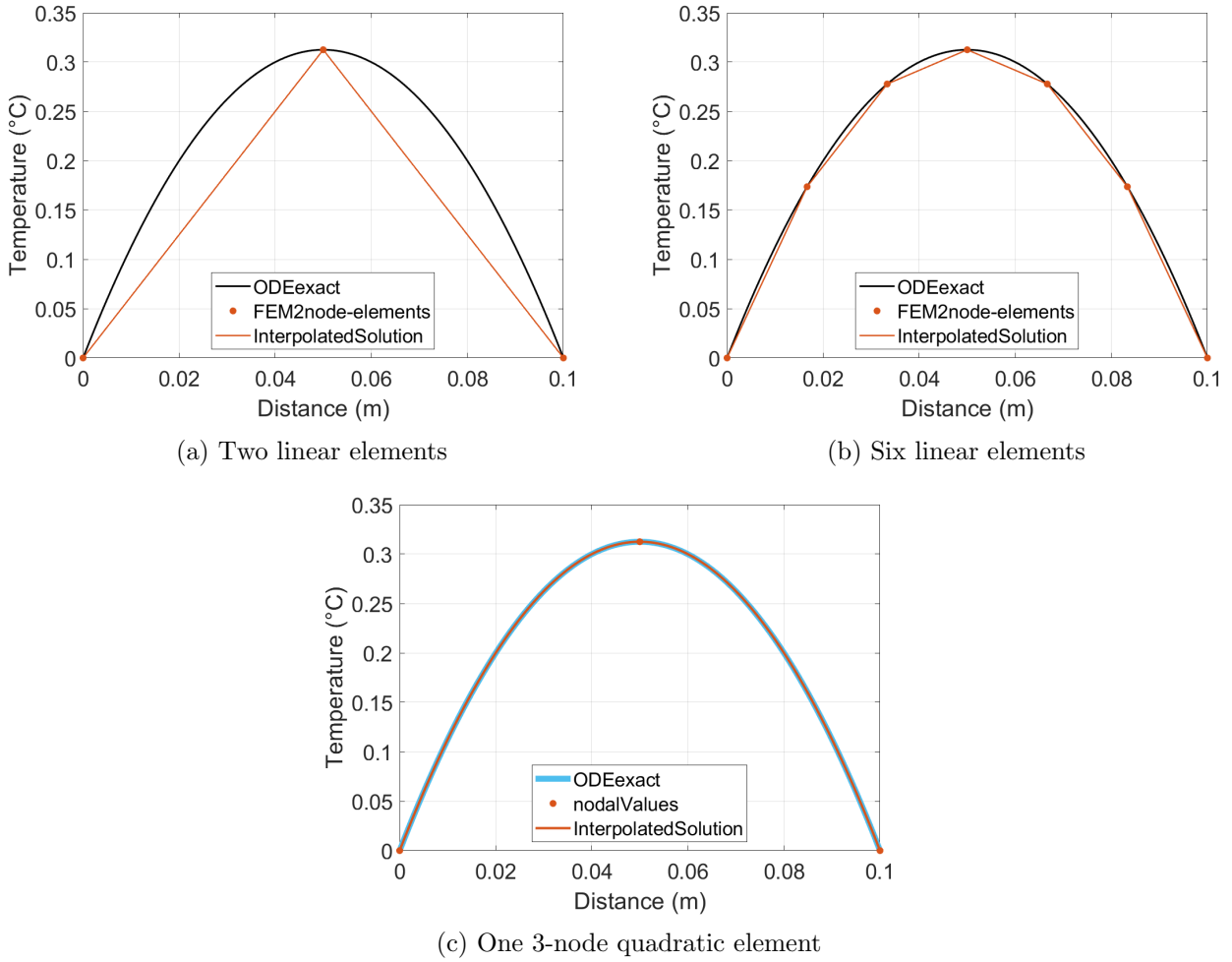
, where  $s = x/L$ . With this element the interpolated FE result is exactly the analytical result as seen in Fig. 4.1(c). This example illustrates that for solving and approximating parabolic shapes a quadratic element is more efficient. This conclusion is also valid for 2D and 3D cases as well as will be seen in the following sections.

## 4.2 2D Elements for Heat Transfer with Heat Generation

The simplest element in two-dimensional FEM is the four-node element. Deriving and expressing the interpolation functions of such an element in the  $x$  and  $y$  coordinates yield algebraically complex expressions. The natural coordinates can thus be changed to normalized coordinates  $r$  and  $s$ , Fig. 4.3 [33].

$$r = \frac{x - \bar{x}}{a} \quad s = \frac{y - \bar{y}}{b} \quad (4.10)$$

The constants  $2a$  and  $2b$  are the length and the width of the element. The interpolation



**Figure 4.1:** Solution for the 1D heat transfer problem with internal heat generation.

functions for such an element are:

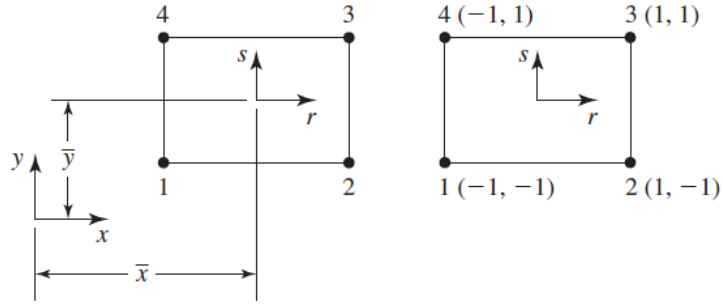
$$\begin{aligned}
 N_1 &= \frac{1}{4} (1 - r) (1 - s) \\
 N_2 &= \frac{1}{4} (1 + r) (1 - s) \\
 N_3 &= \frac{1}{4} (1 + r) (1 + s) \\
 N_4 &= \frac{1}{4} (1 - r) (1 + s)
 \end{aligned} \tag{4.11}$$

such that the value of the interpolation function is 1 at the respective node and 0 at the others. The higher order elements can be categorized in two common families: serendipity and Lagrange [33].

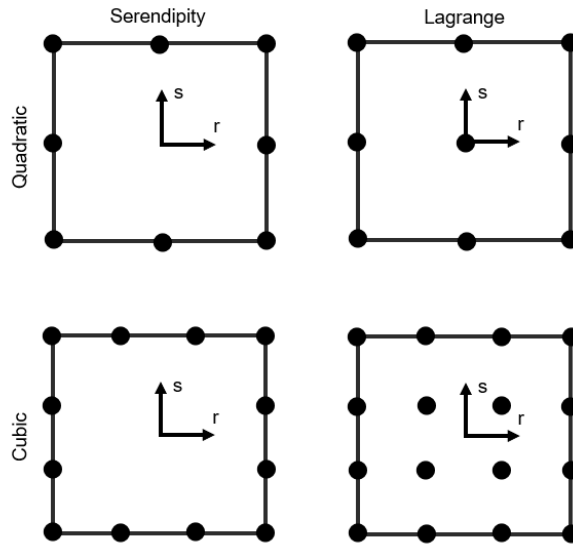
The advantage of the quadratic Lagrange element is the presence of the internal node which allows more accurate temperature rises estimations.

For the quadratic Lagrange element, the following interpolation functions are used [36]:

$$\begin{aligned}
 \text{Corner nodes : } N^i &= \frac{1}{4} (1 + r^i r) r^i r (1 + s^i s) s^i s \\
 \text{Middle edge nodes : } N^i &= \frac{1}{2} (1 - s^2) (1 + r^i r) r^i r, \quad s^i = 0 \\
 \text{Middle node : } N^i &= (1 - r^2)(1 - s^2)
 \end{aligned} \tag{4.12}$$



**Figure 4.2:** A four-node rectangular element showing the translation to natural coordinates [33].



**Figure 4.3:** The quadratic and cubic serendipity and Lagrange family.

The full nine interpolation functions are shown in Appendix A.2.

### 4.2.1 Theoretical Framework for 2D Elements

The 2D theoretical framework is analogous to the 1D case. The heat transfer equation in 2D is:

$$\frac{\partial}{\partial x} \left( k_x \frac{\partial T}{\partial x} \right) + \frac{\partial}{\partial y} \left( k_y \frac{\partial T}{\partial y} \right) + Q = 0 \quad (4.13)$$

The temperature distribution is discretized as:

$$T(x, y) = \sum_{i=1}^M N_i(x, y) T_i = [N]\{T\} \quad (4.14)$$

By applying the Galerkin method to equation (4.13), the following element formulation is obtained:

$$\iint_A \left( k_x \frac{\partial [N]^T}{\partial x} \frac{\partial [N]}{\partial x} + k_y \frac{\partial [N]^T}{\partial y} \frac{\partial [N]}{\partial y} \right) dA T = \iint_A Q [N]^T dA - \oint_S (q_x n_x + q_y n_y) [N]^T dS \quad (4.15)$$

This equation is of the form:

$$[k^{(e)}] \{T^{(e)}\} = \{f_Q^{(e)}\} + \{f_g^{(e)}\} \quad (4.16)$$

where  $[k^{(e)}]$  is the element stiffness matrix,  $\{f_Q^{(e)}\}$  is the element force vector representing internal heat generation and  $\{f_g^{(e)}\}$  is the element nodal force vector associated with heat flux across the element surface area. Note that the term "stiffness matrix" is borrowed from the application of the finite-element method to problems in structural mechanics and is a misnomer in case of thermal problems. The same holds for "force vector".

Transforming left side of equation (4.15) to  $r, s$  coordinates yields:

$$\int_{-1}^1 \int_{-1}^1 \left( k_x \frac{\partial N_i}{\partial r} \frac{\partial N_j}{\partial r} \frac{b}{a} + k_y \frac{\partial N_i}{\partial s} \frac{\partial N_j}{\partial s} \frac{a}{b} \right) dr ds, \quad (4.17)$$

where  $N_i$  is the column interpolation functions vector and  $N_j$  is the row vector [33]. Also,  $i, j = 1 \dots M$ .

### Convective BCs

When applying convective boundary conditions, the flux  $\{f_q^{(e)}\}$  must be in balance with the convection from the segment of concern [33].

$$\{f_g^{(e)}\} = - \oint_S (q_x n_x + q_y n_y) [N]^T dS = - \oint_S h (T^{(e)} - T_a) [N]^T dS \quad (4.18)$$

Equation (4.18) can be rewritten as:

$$\{f_g^{(e)}\} = - \oint_S h [N]^T [N] T dS + \oint_S h T_a \{N\} dS \quad (4.19)$$

And to generalize it:

$$\{f_g^{(e)}\} = - [k_{hS}^{(e)}] T + \{f_{hS}^{(e)}\}, \quad (4.20)$$

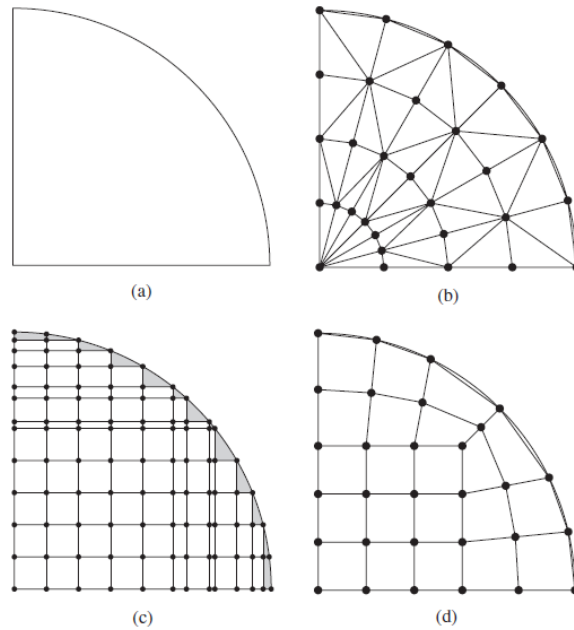
where  $[k_{hS}^{(e)}]$  will add specific terms to the stiffness matrix and  $\{f_{hS}^{(e)}\}$  will contribute to the total force vector [33].

## 4.3 Iso-parametric Formulation

In FEM software in some situations the standard (triangular, rectangular) elements are not efficient to capture irregular geometries such as the rounded corner shape in Figure 4.4.

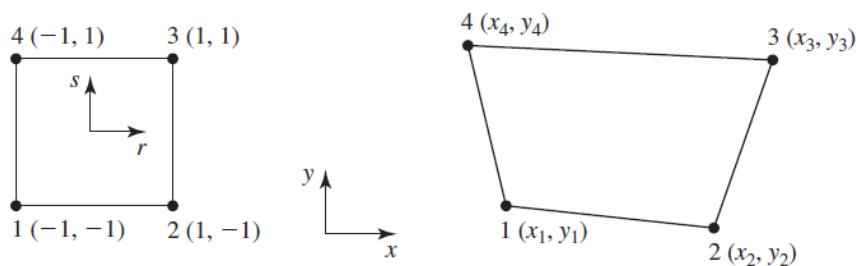
In case of using triangular elements to mesh this object one can notice that the resulting aspect ratio (i.e., the height to base ratio is large). Large aspect ratios increase the inaccuracy of the finite element representation and have a detrimental effect on convergence of finite element solutions. In case of using rectangular elements one can notice that a very fine mesh is required to accurately capture the shape of the object [33].

Another solution is to use quadrilateral elements as shown in Fig. 4.4. The four-node



**Figure 4.4:** (a) A domain to be modeled. (b) Triangular element. (c) Rectangular elements. (d) Rectangular and quadrilateral elements [33].

quadrilateral element is derived from the four-node rectangular element (known as the parent element) via a mapping process. Fig. 4.5 shows the parent element and its natural  $(r, s)$  coordinates and the quadrilateral element in a global Cartesian coordinate system. Since the same interpolation functions are used for both the field variable and description of element geometry, the procedure is known as iso-parametric (constant parameter) mapping. The element defined by such a procedure is known as an iso-parametric element. The mapping process and the element formulation is shown in [33].



**Figure 4.5:** Mapping of a rectangular element to a general quadrilateral element [33].

### 4.3.1 Element Integrals

The integrals in equation (4.15) are a special case for the iso-parametric elements. The left-hand side of the equation requires computation of the partial derivatives of the interpolation

functions with respect to the  $x$  and  $y$  coordinates:

$$\begin{aligned}\frac{\partial N_i}{\partial x} &= \frac{\partial N_i}{\partial r} \frac{\partial r}{\partial x} + \frac{\partial N_i}{\partial s} \frac{\partial s}{\partial x} \\ \frac{\partial N_i}{\partial y} &= \frac{\partial N_i}{\partial r} \frac{\partial r}{\partial y} + \frac{\partial N_i}{\partial s} \frac{\partial s}{\partial y}\end{aligned}\quad (4.21)$$

The partials derivatives with respect to the  $r$  and  $s$  coordinates are:

$$\begin{aligned}\frac{\partial N_i}{\partial r} &= \frac{\partial N_i}{\partial x} \frac{\partial x}{\partial r} + \frac{\partial N_i}{\partial y} \frac{\partial y}{\partial r} \\ \frac{\partial N_i}{\partial s} &= \frac{\partial N_i}{\partial x} \frac{\partial x}{\partial s} + \frac{\partial N_i}{\partial y} \frac{\partial y}{\partial s}\end{aligned}\quad (4.22)$$

In matrix form that is:

$$\begin{bmatrix} \frac{\partial N_i}{\partial r} \\ \frac{\partial N_i}{\partial s} \end{bmatrix} = \begin{bmatrix} \frac{\partial x}{\partial r} & \frac{\partial y}{\partial r} \\ \frac{\partial x}{\partial s} & \frac{\partial y}{\partial s} \end{bmatrix} \begin{bmatrix} \frac{\partial N_i}{\partial x} \\ \frac{\partial N_i}{\partial y} \end{bmatrix}\quad (4.23)$$

The Jacobian operator is:

$$\mathbf{J} = \begin{bmatrix} \frac{\partial x}{\partial r} & \frac{\partial y}{\partial r} \\ \frac{\partial x}{\partial s} & \frac{\partial y}{\partial s} \end{bmatrix} = \begin{bmatrix} \sum_{i=1}^n \frac{\partial N_i}{\partial r} x_i & \sum_{i=1}^n \frac{\partial N_i}{\partial r} y_i \\ \sum_{i=1}^n \frac{\partial N_i}{\partial s} x_i & \sum_{i=1}^n \frac{\partial N_i}{\partial s} y_i \end{bmatrix},\quad (4.24)$$

where  $n$  is the number of the interpolation functions.

If the inverse of the Jacobian matrix can be found, then the partial derivatives in equation (4.22) can be found:

$$\begin{bmatrix} \frac{\partial N_i}{\partial x} \\ \frac{\partial N_i}{\partial y} \end{bmatrix} = [\mathbf{J}]^{-1} \begin{bmatrix} \frac{\partial N_i}{\partial r} \\ \frac{\partial N_i}{\partial s} \end{bmatrix} = \begin{bmatrix} \mathbf{I}_{11} & \mathbf{I}_{12} \\ \mathbf{I}_{21} & \mathbf{I}_{22} \end{bmatrix} \begin{bmatrix} \frac{\partial N_i}{\partial r} \\ \frac{\partial N_i}{\partial s} \end{bmatrix}\quad (4.25)$$

Then the integrals in (4.15) can be transformed as follows:

$$\iint_A \frac{\partial [N]^T}{\partial x} \frac{\partial [N]}{\partial x} dA = \iint_A \left( \mathbf{I}_{11} \frac{\partial [N]^T}{\partial r} + \mathbf{I}_{12} \frac{\partial [N]^T}{\partial s} \right) \left( \mathbf{I}_{11} \frac{\partial N}{\partial r} + \mathbf{I}_{12} \frac{\partial N}{\partial s} \right) |\mathbf{J}| dr ds,\quad (4.26)$$

where the integrand is transformed using the inverse Jacobian matrix terms and the differential area is transformed using the determinant of the Jacobian matrix. The same procedure is followed for the partial derivatives of the interpolation functions with respect to  $y$ . In a 3D case the procedure is analogous to these steps, only the third coordinate is introduced [33].

## 4.4 3D Elements for Heat Transfer with Heat Generation

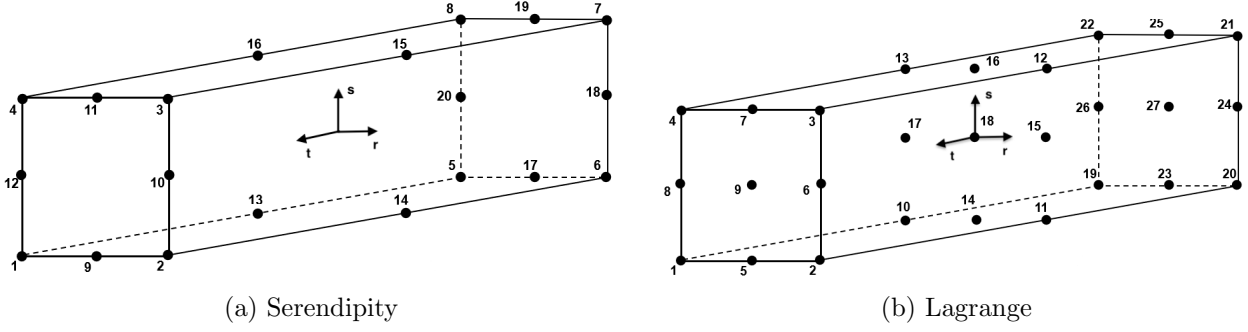
Similarly to the 2D elements, the simplest 3D element is the 8-node linear element. This element can also be defined in the normalized coordinates according to the interpolation functions in Table A.1 in Appendix A.2. The coordinate transformation is realized in a similar fashion as follows:

$$\begin{aligned}r &= \frac{x - \bar{x}}{a} \\ s &= \frac{y - \bar{y}}{b} \\ t &= \frac{z - \bar{z}}{c}\end{aligned}\quad (4.27)$$

Where  $r$ ,  $s$ ,  $t$  range from  $-1$  to  $1$ ,  $2a$ ,  $2b$ ,  $2c$  are the dimensions of the element in the  $x$ ,  $y$ ,  $z$  coordinates,  $w$ ,  $h$ ,  $l$ , respectively, and the coordinates of the element centroid are:

$$\begin{aligned}\bar{x} &= \frac{x_1 - x_2}{2} \\ \bar{y} &= \frac{y_3 - y_2}{2} \\ \bar{z} &= \frac{z_5 - z_1}{2}\end{aligned}\quad (4.28)$$

Similarly to the 2D case, the 3D quadratic Lagrange is more advantageous for heat transfer



**Figure 4.6:** Quadratic serendipity and Lagrange elements

problems with internal heat generation due to the internal node.

The higher-order elements in 3D can also be categorized in serendipity and Lagrange elements. The quadratic versions of the two are represented in Fig. 4.6. The interpolation functions for a 20-node serendipity element are determined using the relations in [36]. For the corner nodes:

$$r_a = \pm 1, s_a = \pm 1, t_a = \pm 1$$

$$N_a = \frac{1}{8}(1 + r_a r)(1 + s_a s)(1 + t_a t)(r_a r + s_a s + t_a t - 2) \quad (4.29)$$

Typical mid-side node:

$$\text{for } r = 0, s_a = \pm 1, t_a = \pm 1$$

$$N_a = \frac{1}{4}(1 - r^2)(1 + s_a s)(1 + t_a t) \quad (4.30)$$

From a practical perspective the 20-node element is flexible and can adapt various BC situations. However, it has limited accuracy when it comes to solving the internal heat generation cases due to the missing mid-element node.

The 27-node element on the other hand, can adapt all BC situations and solve internal heat generation problems due to the present mid-element node. Determining the interpolation functions of this element is straightforward. One can use the interpolation functions of the Lagrange 2D element in 4.12. These can be transposed along the  $t$  axis from 1 to 0 to  $-1$ . So, at  $t = 1$  the nine functions can be multiplied by a function of  $t$  that equals 1 at  $t = 1$  and equals 0 at  $t = 0$ , and  $-1$ . A similar procedure is followed for the functions at  $t = 0$  and  $t = -1$ . This results in the following interpolation functions shown in 4.31 and fully in Appendix A.2.

$$\begin{aligned} \{N_{1..9}\} &= N_{1..9}^{2D} \cdot \frac{1}{2}(1 + t) - \frac{1}{2}(1 - t^2) \\ \{N_{10..18}\} &= N_{1..9}^{2D} \cdot (1 - t^2) \\ \{N_{19..27}\} &= N_{1..9}^{2D} \cdot \frac{1}{2}(1 - t) - \frac{1}{2}(1 - t^2) \end{aligned} \quad (4.31)$$

#### 4.4.1 Theoretical Framework for 3D Elements

The theoretical framework for the 3D case follows in an analogous way from the 2D case by adding a third coordinate and changing the surface integrals to volume integrals. The heat transfer equation in 3D is:

$$\frac{\partial}{\partial x} \left( k_x \frac{\partial T}{\partial x} \right) + \frac{\partial}{\partial y} \left( k_y \frac{\partial T}{\partial y} \right) + \frac{\partial}{\partial z} \left( k_z \frac{\partial T}{\partial z} \right) + Q = 0 \quad (4.32)$$



The temperature distribution is discretized as:

$$T(x, y, z) = \sum_{i=1}^M N_i(x, y, z) T_i = [N] \{T\} \quad (4.33)$$

By applying the Galerkin method to equation (4.32), the following element formulation is obtained [33]:

$$\iiint_V \left( k_x \frac{\partial [N]^T}{\partial x} \frac{\partial [N]}{\partial x} + k_y \frac{\partial [N]^T}{\partial y} \frac{\partial [N]}{\partial y} + k_z \frac{\partial [N]^T}{\partial z} \frac{\partial [N]}{\partial z} \right) dV \{T\} = \quad (4.34)$$

$$\iiint_V Q [N]^T dV - \oint_A (q_x n_x + q_y n_y + q_z n_z) [N]^T dA$$

This equation is also of the form:

$$[k^{(e)}] \{T^{(e)}\} = \{f_Q^{(e)}\} + \{f_g^{(e)}\}, \quad (4.35)$$

where  $[k^{(e)}]$  is the element stiffness matrix,  $\{f_Q^{(e)}\}$  is the element force vector representing internal heat generation and  $\{f_g^{(e)}\}$  is the element nodal force vector associated with heat flux across the element surface area.

Transforming equation 4.34 to the  $r, s, t$  coordinates:

$$[k^{(e)}] = \int_{-1}^1 \int_{-1}^1 \int_{-1}^1 \left( k_x \frac{\partial N_i}{\partial r} \frac{\partial N_j}{\partial r} \frac{bc}{a} + k_y \frac{\partial N_i}{\partial s} \frac{\partial N_j}{\partial s} \frac{ac}{b} + k_z \frac{\partial N_i}{\partial t} \frac{\partial N_j}{\partial t} \frac{ab}{c} \right) dr ds dt, \quad (4.36)$$

where  $N_i$  is the column interpolation functions vector and  $N_j$  is the row vector. Also,  $i, j = 1 \dots M$ . If  $k$  is constant over the volume, then equation (4.36) can be written as:

$$[k^{(e)}] = k_x \frac{bc}{a} \int_{-1}^1 \int_{-1}^1 \int_{-1}^1 \frac{\partial N_i}{\partial r} \frac{\partial N_j}{\partial r} dr ds dt \quad (4.37)$$

$$+ k_y \frac{ac}{b} \int_{-1}^1 \int_{-1}^1 \int_{-1}^1 \frac{\partial N_i}{\partial s} \frac{\partial N_j}{\partial s} dr ds dt + k_z \frac{ab}{c} \int_{-1}^1 \int_{-1}^1 \int_{-1}^1 \frac{\partial N_i}{\partial t} \frac{\partial N_j}{\partial t} dr ds dt$$

The integrals in equation 4.37 are trivial parts which can be pre-computed using symbolic math and stored. This equation can be rewritten as:

$$[k^{(e)}] = k_x \frac{bc}{a} C_x + k_y \frac{ac}{b} C_y + k_z \frac{ab}{c} C_z \quad (4.38)$$

Thus, the matrices  $C_x, C_y$  and  $C_z$  are called trivial parts of the element stiffness matrix since they solely depend on the predetermined element interpolation functions  $N$ .

## Convective BCs

For applying convective boundary conditions, the flux  $\{f_g^{(e)}\}$  must be in balance with the convection from the area of concern [33].

$$\{f_g^{(e)}\} = \oint_S (q_x n_x + q_y n_y + q_z n_z) [N]^T dS = - \oint_S h (T^{(e)} - T_a) [N]^T dS \quad (4.39)$$

Similarly to the 2D case, equation 4.39 can be rewritten as:

$$\{f_g^{(e)}\} = \oint_S h [N]^T [N] T dS + \oint_S h T_a \{N\} dS \quad (4.40)$$

And to generalize it:

$$\{f_g^{(e)}\} = - [k_{hS}^{(e)}] \{T\} + \{f_{hS}^{(e)}\}, \quad (4.41)$$

where  $[k_{hS}^{(e)}]$  will add specific terms to the stiffness matrix and  $\{f_{hS}^{(e)}\}$  will contribute to the total force vector. Equations 4.35 and 4.41 can be combined in a simplified element formulation,

including convective flux BCs:

$$\left[ k^{(e)} + k_{hS}^{(e)} \right] \{T^{(e)}\} = \{f_Q^{(e)}\} + \{f_{hS}^{(e)}\} \quad (4.42)$$

### Average Temperature

To calculate the average temperature in a 3D element one can use the interpolation functions to first calculate the temperature distribution inside the element.

$$T_{avg} = \frac{\int_V \sum_{i=1}^{27} N_i(r, s, t) \cdot T_i dV}{V} \quad (4.43)$$

Then one can integrate this over the whole volume of the element and divide by the total volume.

An issue with this first formulation is that one would need to integrate the interpolation functions every time the element size changes. This is not optimal for optimization applications. That's why one would prefer a formulation where these integrals can be precomputed and stored to be re-used. To do this one can re-arrange the previous formula.

$$T_{avg} = \frac{\sum_{i=1}^{27} \int_V N_i(r, s, t) dV}{V} T_i = \frac{abc \cdot P_i}{V} T_i \quad (4.44)$$

With this formula one can precompute the integrals  $P_i$  of the interpolation functions since they are pre-determined and re-use them. In fact, this precomputed term is also part of the  $f_Q^{(e)}$  calculation.

### Symmetrical 18-node Element

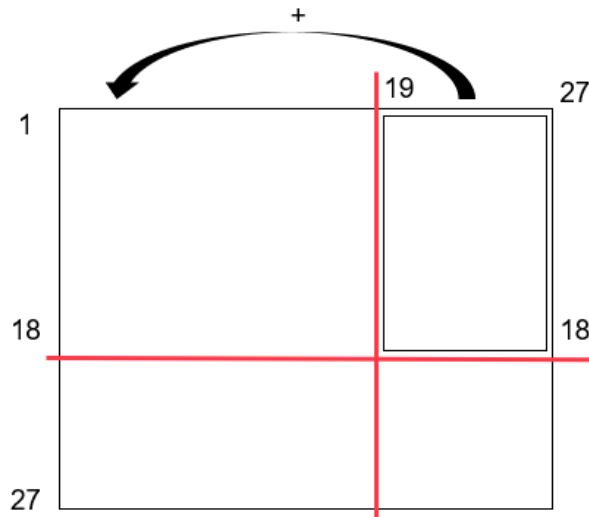
The magnetic core possesses symmetry properties in most cases about the central vertical axis ( $x=0$ , see Fig. 1.3(a)) and in some cases even about both vertical and horizontal ( $z=0$ , see Fig. 1.3(a)) central axes. This means that it is more efficient to account for the symmetrical situation already in the element formulations. A symmetrical element is also in line with the fast modelling approach, because the system of equations and thus the computation times are reduced.

For instance, the 27-node element is symmetrical around  $t = 0$  (Fig. 4.6(b)). This means that the last nine nodes are equal to the first nine nodes. Furthermore, the stiffness matrix and the force vector can be reduced to only 18 nodes. The complexity of the system to be solved will be reduced by nine equations.

The 27-by-27 stiffness matrix can be reduced to 18-by-18 by following the procedure shown in Fig. 4.7. The elements with row numbers from 1 to 18 and column numbers 19 to 27 are added to the elements with row numbers 1 to 18 and column numbers 1 to 9. This means the matrix is reduced to 18-by-18 elements by considering the effect of the last nine symmetrical nodes. Furthermore, the force vector elements from 19 to 27 are removed as well as the unknown temperatures from 19 to 27. The last step to complete the symmetrical element is to adjust the interpolation functions. They must be changed to:

$$\begin{aligned} \{N_{1..9}\} &= N_{1..9}^{2D} \cdot t^2 \\ \{N_{10..18}\} &= N_{1..9}^{2D} \cdot (1 - t^2) \end{aligned} \quad (4.45)$$

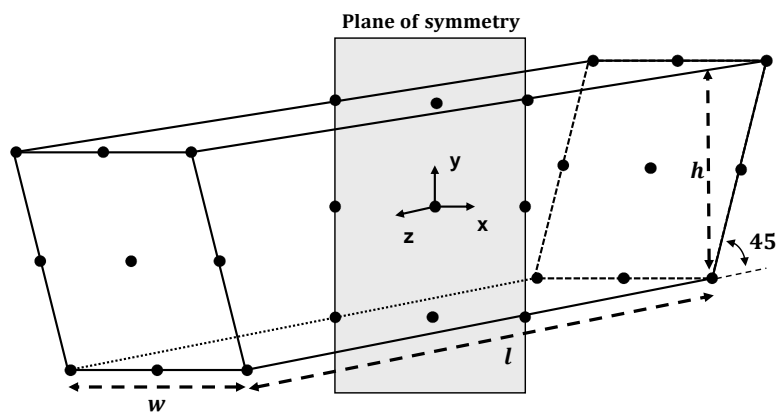
It is important to note that the changed interpolation functions are an essential part of the average temperature calculation for this element. One must not mistake using the 27-node element interpolation functions for the average temperature calculation. Moreover, it is also important to note that the volume has to be reduced by half when calculating the average temperature.



**Figure 4.7:** Reducing the system stiffness matrix due to symmetry of the last nine nodes.

### Iso-parametric 27-node Element

An iso-parametric element is interesting to look at since it would allow the modelling of the full cornered shape of the transformer core with a minimum number of elements. Such an element is depicted in Fig. 4.8. In this element two of the faces are angled at  $45^\circ$  as shown. Due to the symmetry, half of the element is omitted. Two symmetrical elements can then be joined at the angled faces to model one fourth of the transformer core, see Fig. 1.3(a). To solve such an



**Figure 4.8:** Iso-parametric element that facilitates full modeling of the transformer core with a minimum number of elements.

element with convective BCs, the 2D iso-parametric theory in section 4.3.1 is extended to a 3D case by adding a  $3^{rd}$  coordinate.

For an iso-parametric element the resulting integrals in the stiffness matrix and force vectors are dependent on the coordinates and thus on the geometry of the element. Computing the integrals every time the geometry of the element changes is not in line with the fast-modeling approach. A way to overcome this is to write the coordinates of the element nodes in terms of the geometrical variables shown in Fig. 4.8. The resulting stiffness matrix and force vectors after the integrals computation results in expressions in terms of the variables  $w$ ,  $h_e$ ,  $l$ . An example of the first stiffness matrix element is:

**Table 4.1:** Simulation settings for the 2D case.

Object	Square
Size of edge	2 [m]
Internal heat rate, $Q$	1000 [ $\frac{W}{m^3}$ ]
HTC, $h$	200 [ $\frac{W}{m^2K}$ ]
Thermal conductivity, $k$	4 [ $\frac{W}{mK}$ ]
Ambient temperature, $T_a$	0 [ $^{\circ}C$ ]

$$\begin{aligned}
 k_1^{(e)} = k \left( \frac{7h_e^2}{675w} - \frac{23w}{45} + \frac{28h_e l}{675w} - \frac{622lw}{675h_e} - \frac{26l^2w}{45h_e^2} - \frac{26l^3w}{225h_e^3} + \frac{w \cdot \log\left(\frac{1}{l^{\frac{13}{225}}}\right) (2h_e^2 + 3h_e l + l^2)^2}{h_e^4} \right. \\
 \left. + \frac{w \cdot \log\left((2h_e + l)^{\frac{13}{225}}\right) (2h_e^2 + 3h_e l + l^2)^2}{h_e^4} \right)
 \end{aligned} \tag{4.46}$$

Computation of the integrals for an iso-parametric element is algebraically complex as seen by the complex result in 4.46. Obtaining this expression requires using several symbolic simplification commands in MATLAB. The computation time of the stiffness matrix is higher due to the number of operations in each matrix's element. This will be further explored in section 5.5.

## 4.5 Accuracy of the Formulated FEs

### 4.5.1 Initial Tests

#### 4-node Element

To validate the implementation and test the accuracy of the 4-node and the 9-node elements a simple FEM problem will be defined in 2D. Similarly to the 1D case a simple square geometrical shape is chosen. The four edges are exposed to convective boundary conditions and a uniform volumetric heat source is considered inside the object.

Because of the volumetric heat source at least one node is required inside the object. With a single 4-node element the problem cannot be solved. To obtain a temperature rise of the center vs. the edges, one needs to divide the object into at least four-ideally equal-elements and assemble the four stiffness matrixes in a single 9x9 stiffness matrix according to the connection points of the four elements. The same procedure is also applied to the force vector. This is illustrated via an example in Appendix A.1.

As can be noticed by comparison to the 1D case, linear elements are not sufficient to represent parabolic shapes. In Fig. 4.9 the temperature field of the linear element's solution is visualized and compared to a nearly exact solution obtained with a fine mesh in COMSOL. Due to the linear nature of the elements, the temperature field is piecewise linear rather than parabolic. Moreover, the nodal temperatures are overestimated as can be seen by the errors in Table 4.2. It should be noted that the errors for the nodes situated at edges or surfaces (where convection occurs) are computed having the highest nodal temperature (hotspot) as reference. For the linear elements (see Fig. 4.10(a)) these are nodes 1-8 having as reference the value of node 9. This is done to avoid obtaining high errors for nodes with values close to 0 which are

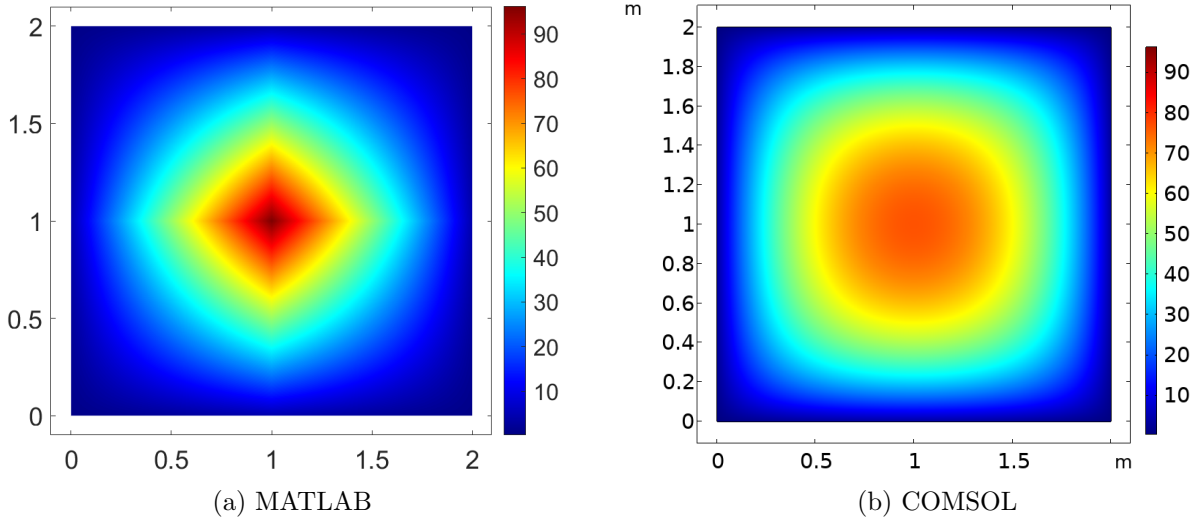
not of interest in the end. For instance, the errors of the low value nodes for the linear elements:

$$Error = \frac{(T_{1-8} - T_{highest}) - (T_{COMS1-8} - T_{highest})}{T_{COMS1-8} - T_{highest}} 100\% \quad (4.47)$$

The more important node is the inner node which for the BCs applied is also the hotspot. The error for this node is defined as ambient to peak:

$$Error = \frac{(T_9 - T_a) - (T_{COMS9} - T_a)}{T_{COMS9} - T_a} 100\% \quad (4.48)$$

This error definitions is kept the same for the subsequent elements.



**Figure 4.9:** Comparison in °C between MATLAB implementation of four linear elements and a COMSOL simulation using a fine meshing.

### 9-node Quadratic Lagrange Element

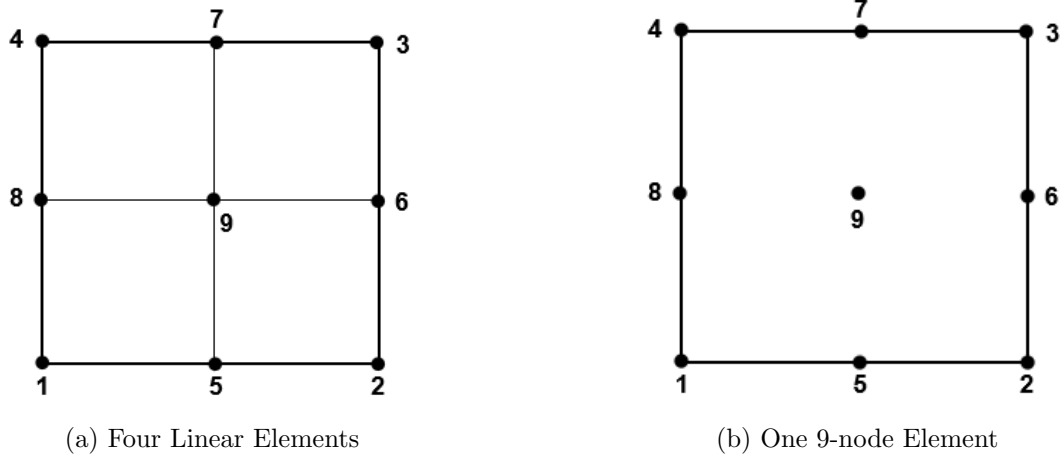
The Lagrange element yields a temperature distribution that is closer to the COMSOL benchmark. Still, the hotspot of the quadratic element is higher than the one in COMSOL with a relative error of 5.5%. This is acceptable for the type of fast model developed in this work. This means that a single quadratic element for the core leg or yoke is sufficient in accuracy.

To illustrate the reason the above error, one can make a 1D cutline plot in the COMSOL benchmark in Fig. 6.2. If one fits a polynomial to the COMSOL plot in Fig. 4.12, a 4<sup>th</sup>-order polynomial will be necessary. This explains the reason for the hotspots mismatch. One cannot solve a 4<sup>th</sup>-order parabolic relationship with a quadratic element exactly. To that end a quartic Lagrange element would be required, however this is unnecessary for the modelling approach followed in this work.

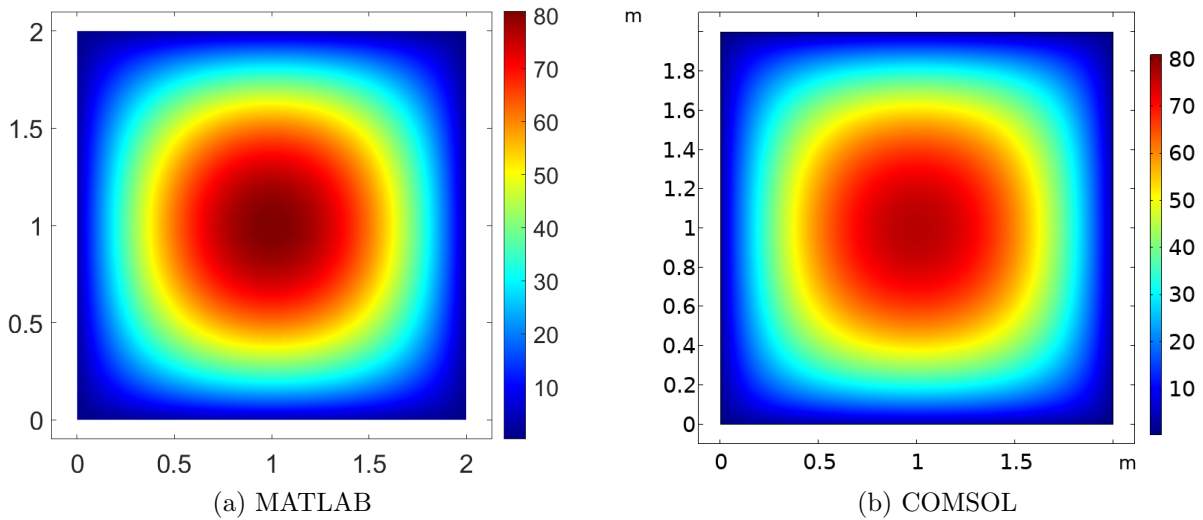
### 27-node Element

The accuracy of the Lagrange 3D element can be assessed, like in the 2D case, by formulating a simple problem and comparing the one element solution to the COMSOL fine-mesh benchmark.

A cube is chosen as the investigated object with the same settings as in Table 1. First, the object is modelled using a single Lagrange quadratic element in MATLAB and next it is compared with an equivalent fine mesh model in COMSOL. The temperature distribution inside the cube can be plotted and visualized using slice functions as seen in Fig. 4.13. The nodal temperature values are compared in Table 4.3. As in the 2D case, the one-quadratic-element solution overestimates the hot-spot. This is the case because of the same reason as in the 2D



**Figure 4.10:** The nodes assigned to the connected linear elements and the single 9-node element.



**Figure 4.11:** Comparison in  $^{\circ}C$  between MATLAB implementation of one quadratic element and a COMSOL simulation using a fine meshing.

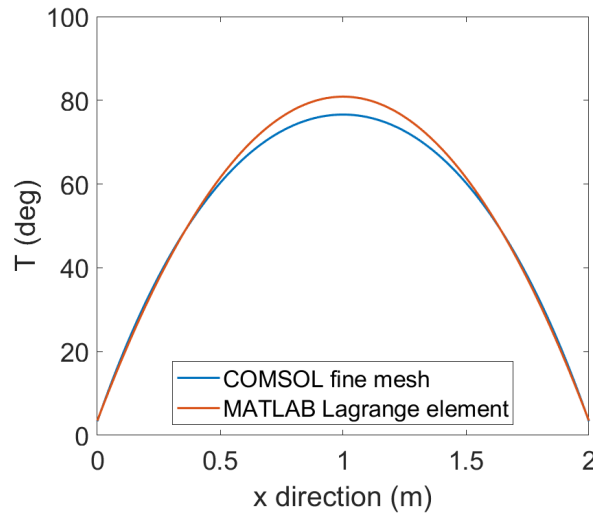
case: the COMSOL solution shows a higher-order relation, while the single element is limited to a quadratic function.

The relative error for the hot-spot in this example is 14.5 %. A higher error is observed compared to the 2D element. This is due to the fact that for a 3D element all external surfaces are exposed to convection and thus for the benchmark, a higher order polynomial distribution is obtained if one makes a cut-line plot. This is not shown here as this is anyways irrelevant for the use purpose of the 3D element. More precisely, in this work, a single element does only have a maximum of four out of six surfaces exposed to convection. The other two surfaces are always adiabatic due to the connectivity to other elements or symmetry considerations.

One important remark to make about the solving procedure is that in computing the element stiffness matrix  $[k^{(e)}]$ , the convective matrix  $[k_{hs}^{(e)}]$  and the force vectors  $\{f_{hs}^{(e)}\}$  and  $\{f_Q^{(e)}\}$  one does not need to re-calculate the integrals every single time. They can be computed once using symbolic math in MATLAB and stored. Then, the problem reduces only to multiplying these with the user input variables such as  $a$ ,  $b$ ,  $c$ ,  $k$ ,  $h$ ,  $T_a$ , choosing which surfaces are exposed to convective heat flux, and lastly solving the system of equations. This is the core of the fast

**Table 4.2:** Comparison between linear elements nodal temperature and fine-mesh COMSOL solution according to the nodes in Fig. 4.10(b).

Node	Four linear elements, °C	One quadratic Lagrange, °C	COMSOL, °C	Relative error linear, %	Relative error quadratic, %
1	0.731	0.515	0.228	0.65	0.37
5	4.268	3.492	3.346	1.25	0.19
9	96.255	80.873	76.598	25.6	5.5



**Figure 4.12:** Comparison of cutline plots in °C in the x-direction of the quadratic element and COMSOL fine mesh result.

modelling approach followed in this work.

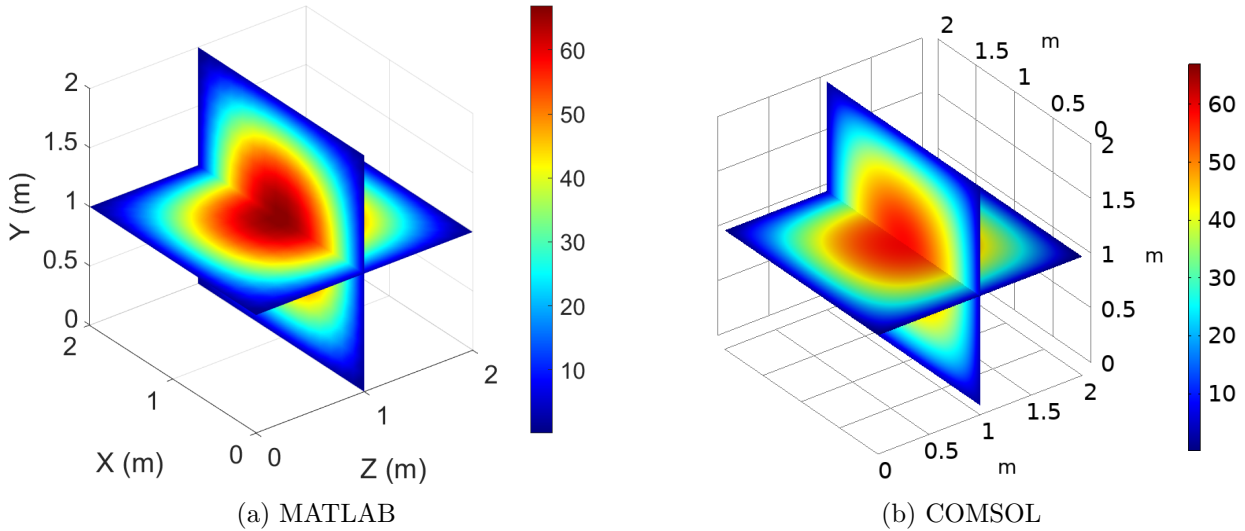
#### 4.5.2 18-node Symmetrical Element

The symmetrical 18 node element was implemented in MATLAB according to the same problem case settings as for the 27-node element. The element was compared with the result of the 27-node element in Fig. 4.7 to check that the expected result is obtained. As can be seen, the symmetric and full versions indeed agree in terms of temperature distribution, suggesting that the symmetric formulation was correctly implemented, see Fig. 4.14.

#### Isoparametric Element

To assess the accuracy of this element a similar problem as for the 27-node element is formulated. For this element convection is applied only to the four straight surfaces. The beveled surfaces will be isolated since in the modelling of the core, convection on these surfaces does not occur. The settings applied are shown in Table 4.4.

The temperature distribution is compared against the COMSOL fine mesh benchmark result. The cut surface plot is made at  $x = 0$ , see Fig. 4.8. The resulting temperature field is shown in Fig. 4.15. It can be noticed by comparison that COMOSL predicts the hotspots to be closer to the inclined edges and not in the middle of the element as the MATLAB single element predicts. Even though the temperature distribution is not captured with high phys-



**Figure 4.13:** 3D visualization of the temperature field in  $^{\circ}\text{C}$  for the single Lagrange element in MATLAB and the COMSOL fine-mesh benchmark.

**Table 4.3:** Comparison of the nodal temperatures between MATLAB one Lagrange element and COMSOL fine mesh benchmark. Note that the other nodes are not shown due to symmetry of the simple problem, see Fig. 4.6.

Node	One quadratic Lagrange, $^{\circ}\text{C}$	COMSOL, $^{\circ}\text{C}$	Relative error, %
1	0.286	0.056	0.39
5	0.537	0.133	0.69
9	3.141	2.77	0.66
18	66.927	58.434	14.5

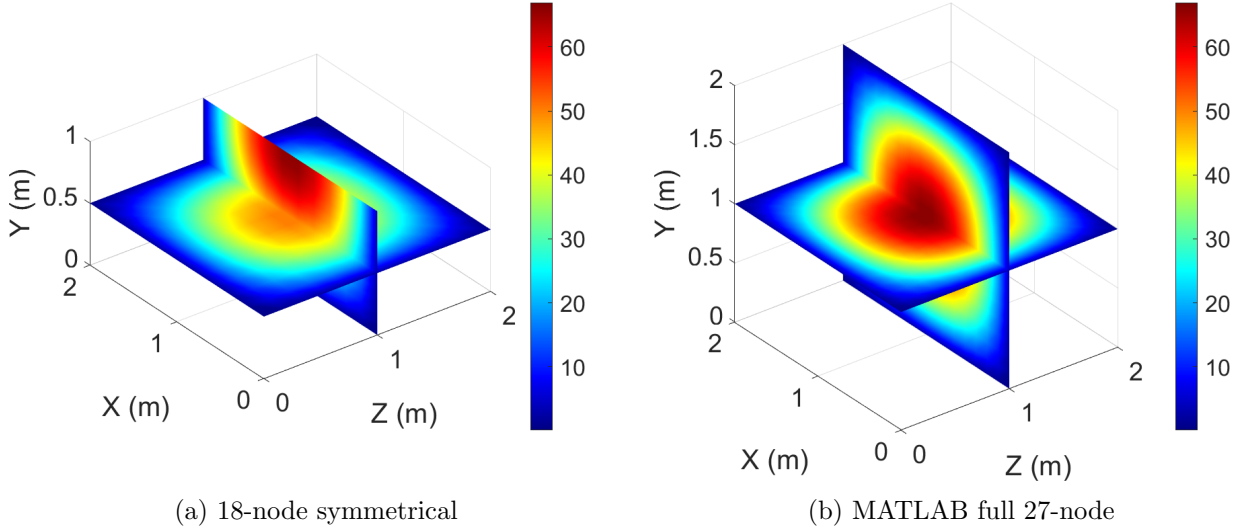
ical accuracy, the temperature rise at the nodes are sufficiently accurate, see Table 4.5. For the nodal numbering one can refer to Fig. 4.6 as the numbering follows the same order for the isoparametric element. For the cold nodes the relative error between the element and COMSOL benchmark is computed with the highest temperature as reference, i.e., 19.925. The highest errors observed is for the mid-surface and mid-element nodes e.g. node 5 and 18: -6.29% and 5.47%. The cause for the deviations is analogous as previously. The fine mesh benchmark result shows a higher order relation. However, as previously mentioned, the errors are sufficient, given the fast modelling approach.

### 4.5.3 Non-Dimensional Parameter Accuracy Tests

The previous initial accuracy tests were performed in a way that does not take into account the end application cases of these FEs. To test the accuracy of the elements for the range of cases relevant in this work a non-dimensional parameter can be defined. It is convenient to define this as  $\frac{h}{k/a}$ , the ratio of the heat-transfer coefficient,  $h$  times the edge size,  $a$  of the element over the thermal conductivity,  $k$ . Then, a range of interest can be defined for  $h$  and  $a$ . For our work this is shown in Table 4.6.

Furthermore, the accuracy of the elements is tested as follows. All 4 edges and all 6 surfaces were set to convective heat flux for the 2D and 3D single elements respectively. The isoparametric element has all external surfaces set to convection except the bevelled surfaces which





**Figure 4.14:** Validation of temperature distribution in  $^{\circ}C$  for the 18-node symmetrical element compared against the original 27-node element.

**Table 4.4:** Iso-parametric element simulation settings.

Parameter	Value
$l$	2 (m)
$w$	1 (m)
$h$	1 (m)
Internal heat rate, $Q$	1000 ( $W/m^3$ )
HTC from surface to ambient, $h$	200 ( $W/m^2K$ )
Thermal conductivity, $k$	4 ( $W/mK$ )
Ambient temperature, $T_a$	0 ( $^{\circ}C$ )
BCs	Convection on the four straight surfaces

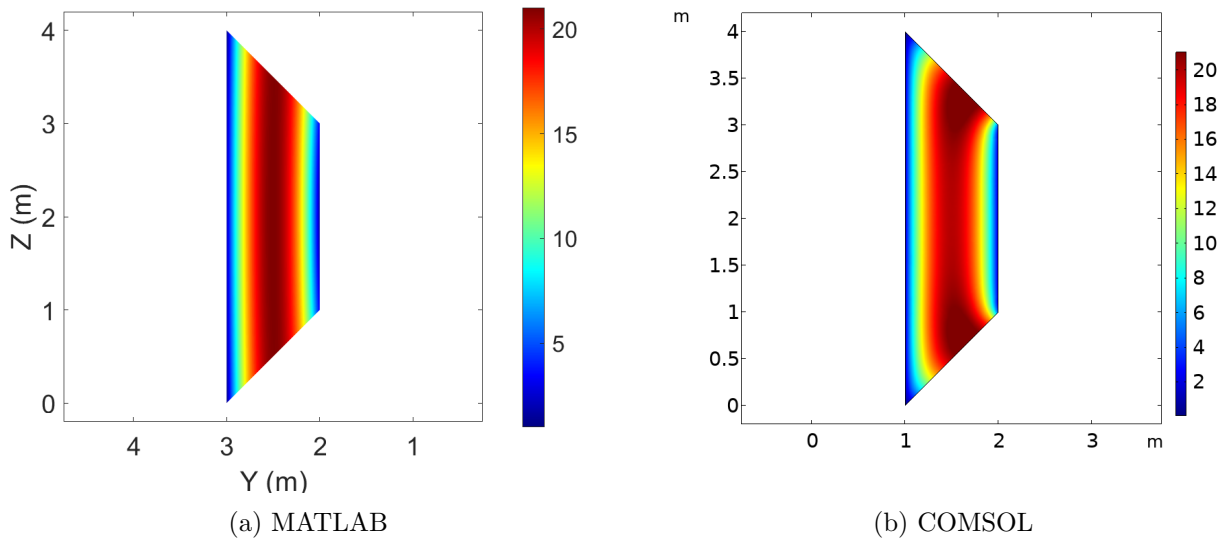
are set to adiabatic. The internal heat generation rate was set to  $500 kW/m^3$ . Then, nodal temperature of the single elements obtained in MATLAB were compared with the temperatures at the same points but simulated with a fine-mesh in COMSOL.

The procedure was applied to the 9-node single element, to the 27-node single element, and to the isoparametric element. For the range of  $\frac{h}{k/a}$  between 0.02-0.4 we get errors (ambient to peak) of 0.0008%-0.05% for the 2D single element, 0.0004%-0.15% for the 3D single element, and 0.027%-0.54% for the isoparametric element. The resulting temperature fields are shown in Fig. 4.16.

The above tests is a proof that it is indeed extremely accurate to model parabolic distributions with single FEs for the non-dimensional range of 0.02-0.4 relevant for this work.

#### 4.5.4 Comparison between the two Accuracy Testing Approaches

While the non-dimensional parameter tests in Sec. 4.5.3 predict the high accuracy ( $< 0.54\%$ ) of the elements for the test cases applied for isotropic cores in subsequent chapters, the initial accuracy tests in Sec. 4.5.1 showcase some limitation cases of these FE. It was seen that for higher HTC values (e.g.,  $200 W/(m^2 K)$ ) the temperature field tends to a 4th order distribution. In this case the single quadratic FE cannot estimate the peaks with high accuracy. Errors of 5.5, 14.5 and -6.29% were observed for the 9-node, 27-node and isoparametric elements respectively.



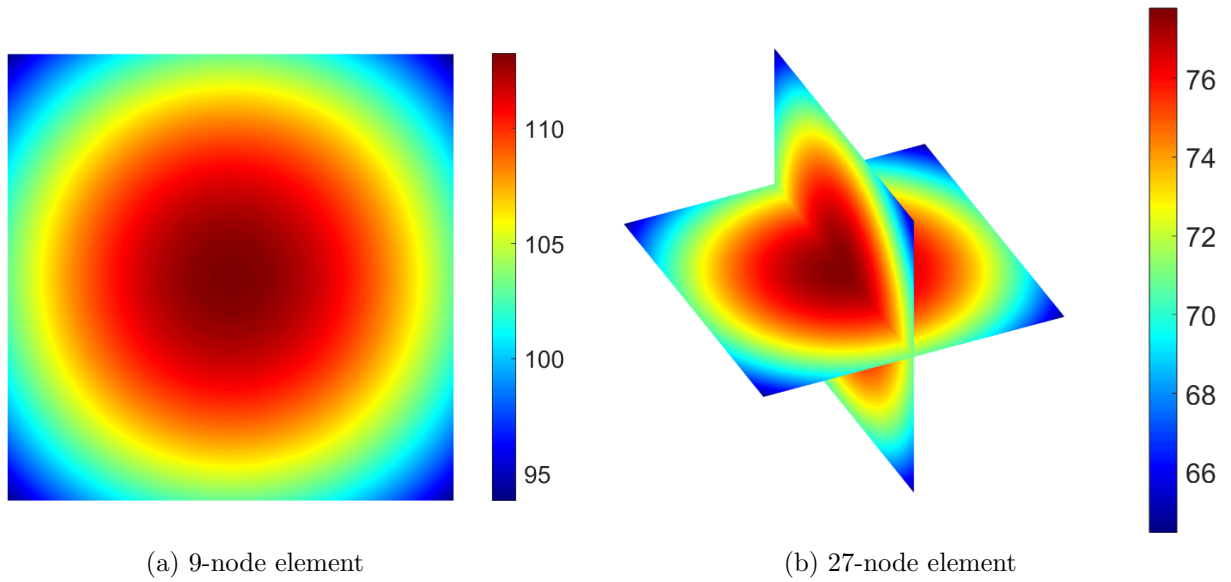
**Figure 4.15:** Temperature distribution in  $^{\circ}C$  along a cut surface plot at half of the isoparametric element width.

**Table 4.5:** Nodal values of single iso-parametric element and COMSOL benchmark.

Node	One iso-parametric element, $^{\circ}C$	COMSOL, $^{\circ}C$	Relative error, %
1	0.3904	0.167	1.13
3	0.2545	0.029	1.13
5	3.027	4.028	-6.29
6	1.725	1.667	0.31
7	0.9978	0.166	4.21
9	20.924	20.249	3.33
10	0.282	0.142	0.71
12	0.315	0.136	0.9
14	1.416	1.656	-1.31
15	1.731	1.653	0.42
16	1.902	1.651	1.37
18	21.016	19.925	5.47

**Table 4.6:** Range of interest of  $\frac{h}{k/a}$

Parameter	Min.	Max.
h-HTC from ambient to surface, $(W/(m^2 K))$	10	50
k-ferrite, $(W/(m K))$	5	5
a-edge length, $(m)$	0.01	0.04
$\frac{h}{k/a}$	0.02	0.4



**Figure 4.16:** The temperature field in  $^{\circ}C$  plotted in MATLAB by interpolating the nodal temperatures obtained for  $\frac{h}{k/a} = 0.04$  (see Table 4.6).

# Model of Isotropic Transformer Cores

## 5.1 COMSOL Benchmark

In Chapter 4 different FEM elements were developed and studied. These elements can be used to model magnetic core shapes. A first component to be modelled is an isotropic transformer bulk core (i.e. ferrite).

We consider the case of a rectangular O-core with symmetry about  $x = 0$  and  $z = 0$ . In this case, it suffices to model an L-shaped core section as shown in Fig. 1.3(a) by the highlighted area. This will be considered the benchmark which will be used to test the accuracy of the later derived models.

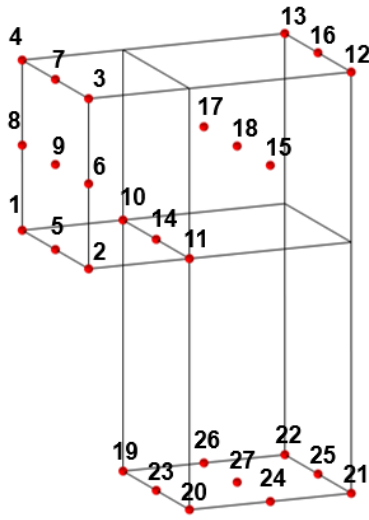
To have a high fidelity in the benchmark results it is important to check the mesh independence of the results. To this end a test case was defined where the benchmark was exposed to convective BCs, see Fig. 5.1(b). This BC case represents one of the cooling situations in a real application.

Important nodes were selected, and temperatures have been probed. The results for two mesh densities i.e., normal and extremely fine setting from COMSOL are shown in Fig. 5.1. As can be seen, for two different meshes the results are almost identical with small discrepancies of around 0.05%. For the sake of having a reliable benchmark result and because the computational time is not an issue for the accuracy tests the finest mesh is chosen for the later use.

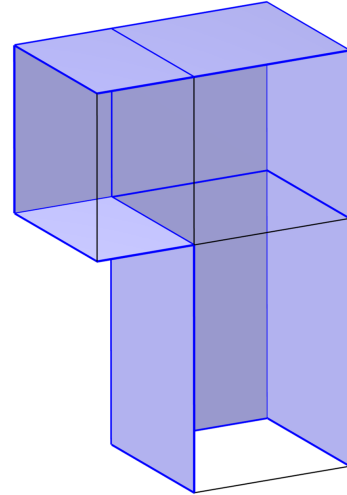
## 5.2 Model Variants

The benchmark core can be modeled in three ways using the elements developed in Chapter 4, see Fig. 5.2.

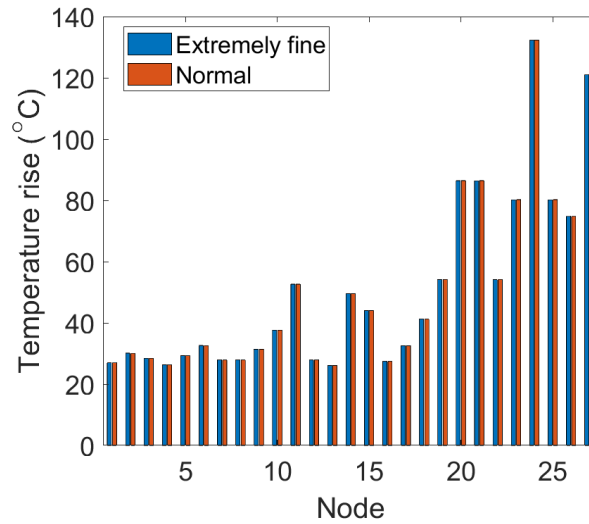
- The non-conformal model is realized using two 18-node symmetrical elements (for the yoke and leg respectively). The two elements are joined as follows: the nodes on the element surfaces facing the corner are connected to each other, even though they are not geometrically co-located. (Hence, the name "non-conformal".) In fact, one could also sketch this model as a single straight bar composed of two elements of the same cross-sections. The advantages of such a rough connection are first having a very simple model which is in line with the fast computation approach. Secondly, one can notice that the total volume included in this model is the same as the benchmark which is important for a later average temperature calculation.
- The three-element model is realized using two 18-node symmetrical elements for the yoke and leg respectively and one 27-node element corresponding to the corner.



(a) The nodes probed.



(b) The BC for the test.



(c) Two mesh densities results.

**Figure 5.1:** Mesh independence test of COMSOL benchmark. Nodes and the resulting rise in temperature.

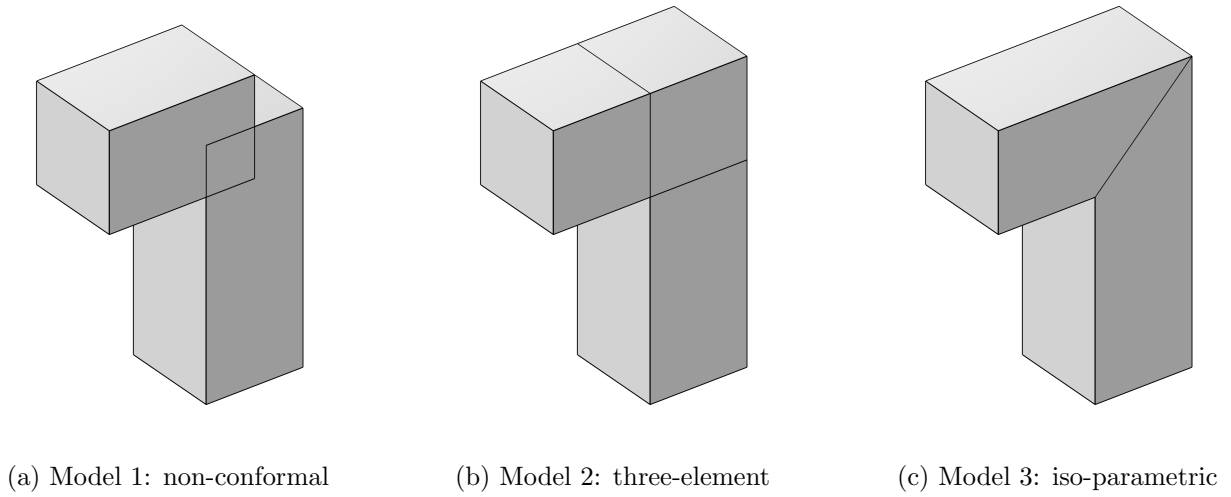
- The iso-parametric model is realized using two 18-node iso-parametric symmetric elements. These elements are obtained from the element defined in section 4.4.1 by following the same procedure as for the 18-node symmetrical element.

The model's stiffness matrixes and force vectors are assembled using the procedure illustrated in A.4.

### 5.3 Test Cases

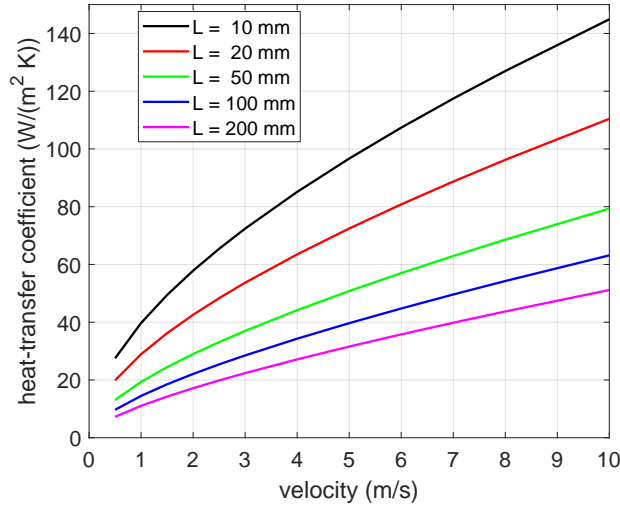
To test the accuracy of the three models above test cases must be defined. Realistic cooling scenarios will be considered as well as HTC,  $k$ , and core geometrical values.

In Fig. 5.3, HTC values are shown for flow of air at 1 bar and 40 °C over a flat plate, as a function of air speed and plate length. We take the flow over a flat plate as a strongly simplified model of the flow along the surface of the core. The HTC values of interest to us are



**Figure 5.2:** The three ways to model a fourth of the isotropic core.

approximately in the range of  $10$  to  $50 \text{ W}/(\text{m}^2\text{K})$ . The first value is obtained for instance for  $1 \text{ m/s}$  and  $200 \text{ mm}$  plate length, while the second value is obtained for  $5 \text{ m/s}$  and  $50 \text{ mm}$  plate length. Test cases are defined for three HTC values:  $10$ ,  $20$ , and  $50 \text{ W}/(\text{m}^2\text{K})$ . Additionally, a typical thermal conductivity for ferrites is  $5 \text{ W}/(\text{mK})$ .

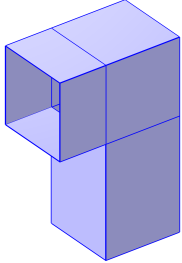
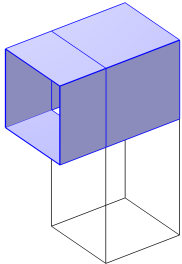
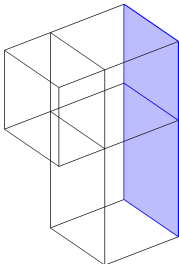
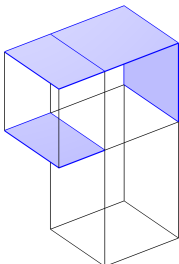
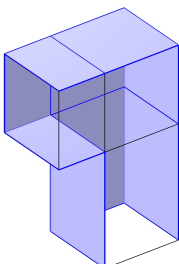
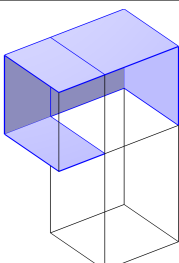


**Figure 5.3:** HTC plots for different flat plate lengths and air velocities at  $1 \text{ bar}$  and  $40 \text{ }^\circ\text{C}$  [34].

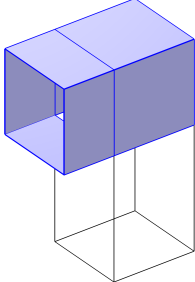
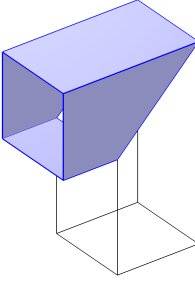
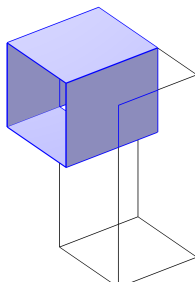
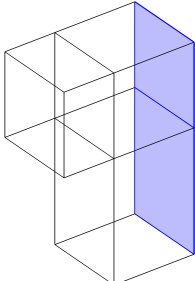
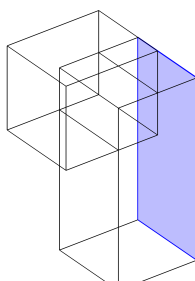
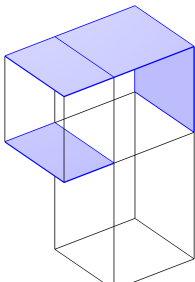
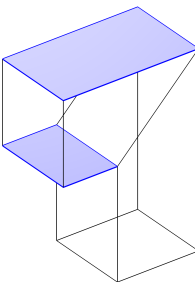
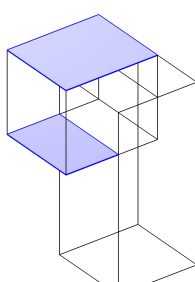
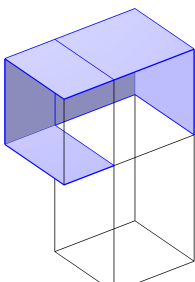
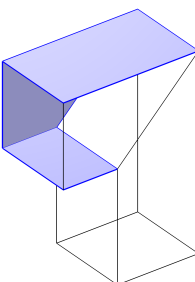
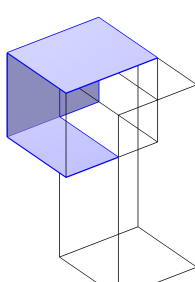
For the geometry of the core, two standard (off-the-shelf) high-power Mn-Zn ferrite cores are chosen, as shown in Fig. 5.5. Depending on the transformer type and the cooling scenario one can have multiple BCs scenarios. One can refer to Fig. 5.4 for general understanding of each BC case. From Fig. 5.4 it can be noticed that the bobbin covers the leg. From this two situations arise: one where the air can still flow on the surface of the core and the other when it is completely covered. Another situation is when multiple cores are stacked on top of each other. This leads to some specific BC case for differed core layers. The most common ones are shown and explained in detail in Table 5.1.

The  $2^{\text{nd}}$  model from Fig. 5.2 can exactly represent all six cases. This is not the case for model 1 and 3. This discrepancy happens for cases 2, 3, 4, and 6 as shown in Table 5.2. One can notice that the area under convective heat flux for models one and three is smaller than the original case. This means that models one and three are limited in representing all BC cases and will show larger deviations for the ones shown in Table 5.2. To summarize the test cases,

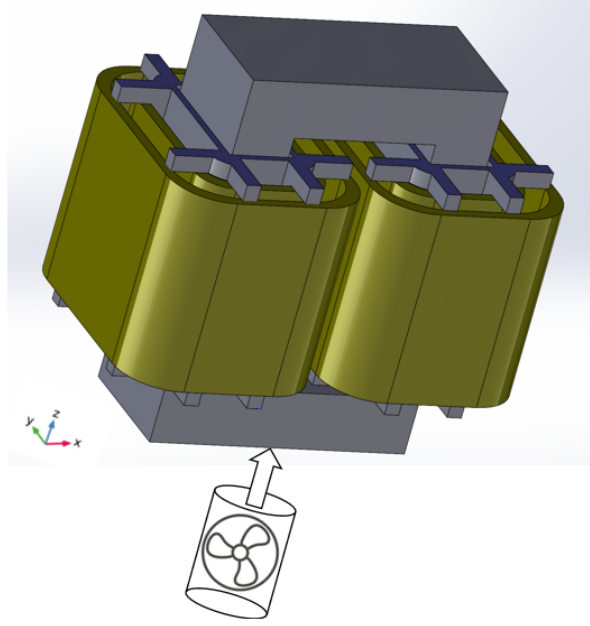
**Table 5.1:** Six different core cooling scenarios showing a short description and the representation of the surfaces under convective heat flux (with light blue).

Case	Description	Representation
1	One core layer, well cooled. Core layer is used to depict one layer of cores in the case of stacked cores. Convection on all external surfaces.	
2	One core layer, bobbin covering leg.	
3	Inner of $\geq$ three core layers, air flow in z-dir., worst case. (in the sense that we neglect heat conduction to the adjacent (in y-direction) core layers).	
4	Inner of $\geq$ three core layers, air flow in y-dir., worst case. (in the sense that we neglect heat conduction to the adjacent (in y-direction) core layers).	
5	Outer core layer of $\geq$ 2 core layers, air flow in z-dir., well cooled.	
6	Outer core layer of $\geq$ 2 core layers, bobbin covering leg.	

**Table 5.2:** The discrepancy of models one and three compared to the BC cases. Notice that the area of convective heat flux for models 1 and 3 is lower than the original case.

Case	Model 3	Model 1
		
	<p data-bbox="600 909 999 976">No discrepancy for model 1 in this case.</p>	
		
		





**Figure 5.4:** Representation of a transformer design with a typical cooling scenario, air flow in the z direction.

three HTC values, two geometries and six BC cases have been defined and chosen. This leads to a total of 36 test cases.

In the next section the three models will be compared against the COMSOL benchmark taking into consideration the 36 test cases defined above.

## 5.4 Models' Accuracy

The next step towards determining the accuracy of the models is to define the important output parameters of these. Typically, optimization tools use the following important output parameters:

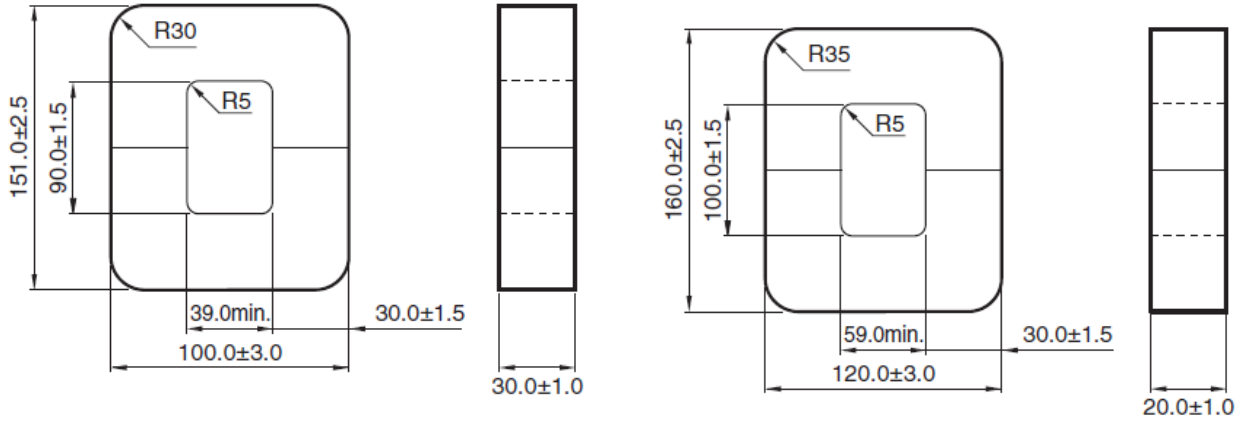
1. Maximum volume temperature ( $T_{max}$ ) (checked against the maximum admissible core-material temperature)
2. Maximum surface temperature ( $T_{surf-max}$ ) (checked against the maximum admissible temperature of the insulation fluid in case of a liquid-immersed core)
3. Average temperature ( $T_{avg}$ ) (used to evaluate the (temperature-dependent) core losses)

These output parameters can be determined for all the three models and the benchmark. Then, the relative error can be computed between them.

First, this will be done using geometry case 1 and an HTCs value of 10, 20 and 50 W/(m<sup>2</sup>K), and for six BC cases as shown in Table 5.1. Fig. 5.6, shows the error plots for all three temperatures.

From these plots, deviations of 30 – 50% for the discrepancy cases listed in Table 5.2: Case 2, 3, 4, and 6 are noticed. On the other hand, model 2 performs with high accuracy. In all the plots above the error of Model 2 is well within 1.25%. From this first test, it can already be noticed that Models 1 and 3 are not sufficiently accurate due to the discrepancies shown in Table 5.2.

The other cases of the second core geometry have been investigated in a similar fashion. Same errors have been seen for models 1 and 3. For model 2 the errors are also within 1.25%.



(a) First geometry-smaller core window.

(b) Second geometry-bigger core window, thinner core.

**Figure 5.5:** Two of-the-shelf ferrite core geometries used for the accuracy tests. The dimensions are shown in mm.

To sum up, model 2 performs the best, predicting temperature rises  $T_{max}$ ,  $T_{surf-max}$  and  $T_{avg}$  having errors within 1.25%. The higher errors for models 1 and 3 reveal their limitation in representing various cooling scenarios precisely. In formulating and assembling a model for a certain object, one has to think about the possible BCs. This gives an indication of the minimum number of elements required. In this specific case it was seen that for representing all BC cases accurately model 2 that has three finite elements is more suitable and the other models having only two elements are limited.

## 5.5 Computational Efficiency

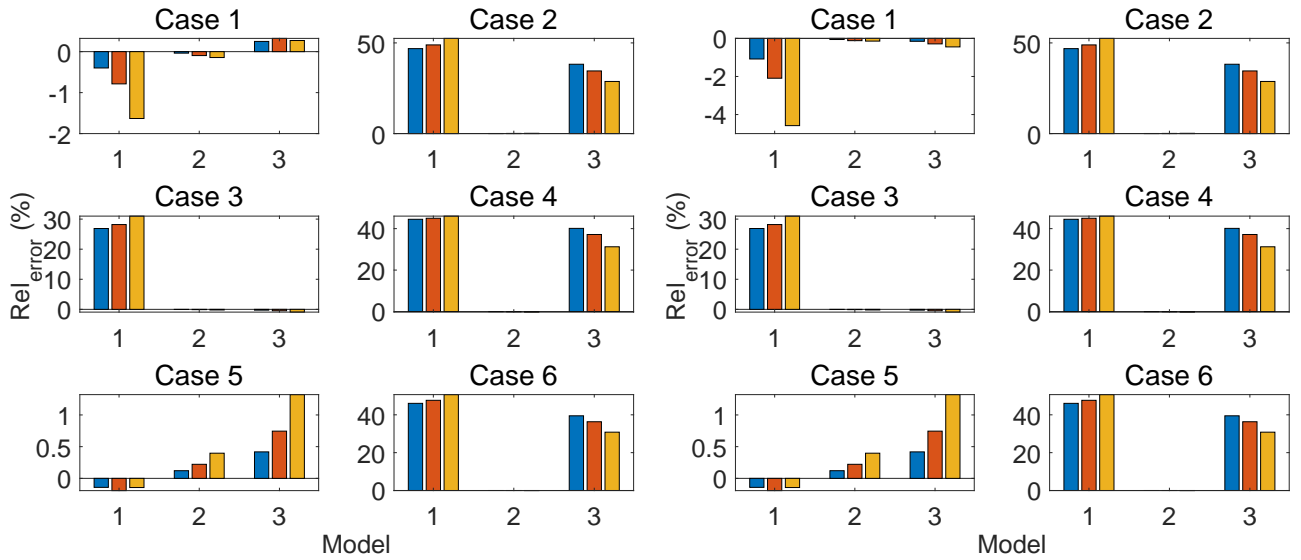
### 5.5.1 Comparison of Different Models and Solving Algorithms

The time efficiency is another important aspect of the developed models. To assess and compare the time efficiency the following steps were taken. First, each model has been written into a function keeping only the essential computations. Then the models are iterated 5000 times and the total time is recorded. This is necessary because MATLAB cannot accurately estimate low times. Then the total time is divided by 5000 to get the time per iteration. The models time efficiency can be seen in Fig. 5.7.

One can see that model 1 performs the best since it solves a smaller linear system of 27 equations. Model 2 has a higher time because it solves 45 equations. Model 3 is more time consuming due to the large number of algebraic operations to compute the iso-parametric stiffness matrix (see eq. (4.46)).

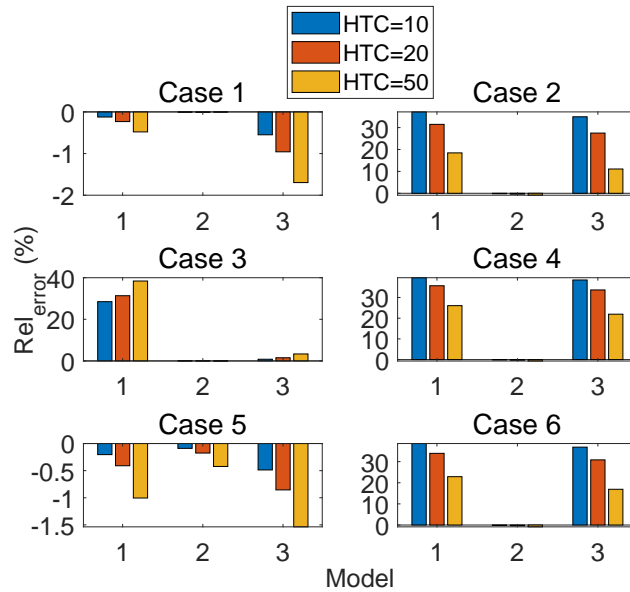
Another important aspect is the method used to solve the linear equations system. In Matlab there are several ways to solve a linear system written in matrix-vector form. If  $X$  is the unknown vector,  $A$  the system matrix and  $b$  the system vector, one can use the slash (/) or backslash (\) operator. This algorithm chooses the solving method based on the system properties. Alternatively, one can use the linsolve function to specify the desired method.

When the systems of equations is solved using slash (/) or backslash (\), the operator factorizes the coefficient matrix  $A$  and uses this matrix decomposition to compute the solution. However, each subsequent time similar system of equations with a different  $b$  is solved, the operator computes the same decomposition of  $A$ , which is a redundant computation. This is a real case in the optimization tools where the same design is analyzed for different operating



(a) Error plots for  $T_{max}$

(b) Error plots for  $T_{surf-max}$



(c) Error plots for  $T_{avg}$

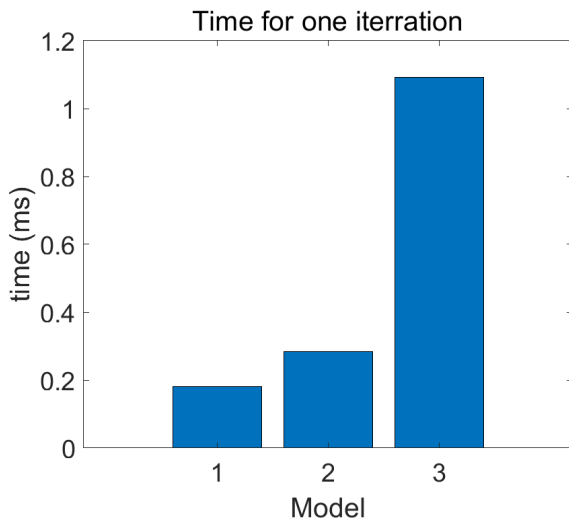
**Figure 5.6:** Error plots for the three important temperatures, of the isotropic core (the same legend holds for all subfigures).

points of volumetric losses  $Q$ . In such a case only the force vector will change and the stiffness matrix will remain the same. The solution to this problem is to pre-compute the decomposition of  $A$ , and then reuse the factors to solve for the different values of  $b$ . This is known as LU decomposition. The difference between using and not using a LU decomposition in the case of same force vector is shown in Fig. 5.7(b).

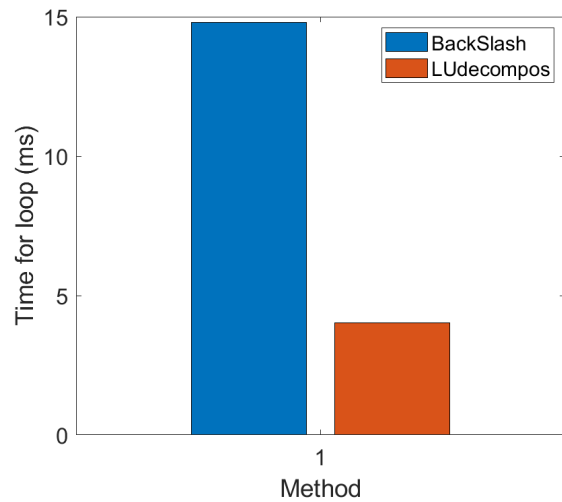
As a final remark, the values in the plots from Fig. 5.5 serve mainly as an illustration for comparison rather than being exact values. The estimated time is relative and varies depending on the device' cores being used at the moment of running the code as well as on the device where the code is run.

### 5.5.2 Comparison with Alternative FEM Livelink COMSOL Model

The alternative to formulating and implementing FE models from scratch in MATLAB would be using an FEM software such as COMSOL to create the same very coarse models and run them



(a) Models' computational time in ms.



(b) Illustration of the computational efficiency of using LU decomposition for different operating points: different force vectors and same stiffness matrix.

**Figure 5.7:** Computational efficiency of the models and two different methods to solve linear systems.

via LiveLink simulation. This unfortunately is not a viable option for optimization applications as it is computationally expensive. A single run of model 2 implemented in COMSOL and run in MATLAB via LiveLink takes about 4.68 s. In contrast it was seen that the same model formulated directly in MATLAB takes approximately 0.3 ms. Thus, the methodology proposed in this work is 15600 times more efficient than the readily available LiveLink solutions.

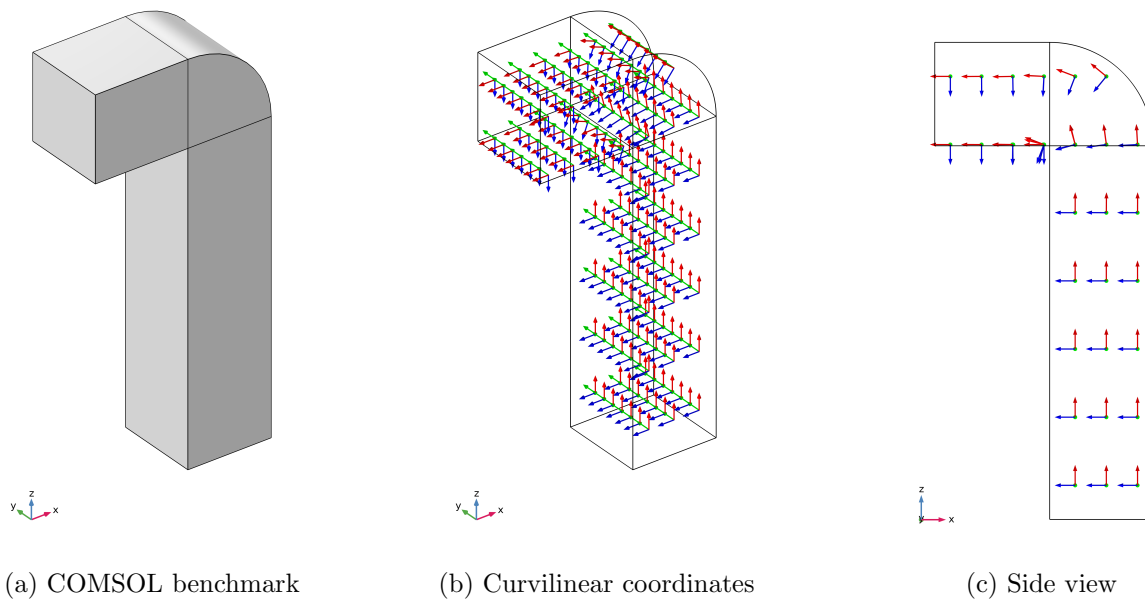
# Model of Anisotropic Transformer Cores

## 6.1 Tape-Wound Cores

In certain frequency ranges and for certain applications (chokes, pulse applications, current transformers), nanocrystalline or amorphous material is the most suitable. These materials are provided in form of tapes. The core is made by winding the tape around a mandril that has the size and shape of the core window. Hence the name tape-wound.

### 6.1.1 COMSOL Benchmark

To create models for these types of cores a new benchmark must be designed in COMSOL.



**Figure 6.1:** COMSOL benchmark model for tape-wound cores. Curvilinear coordinate system used to define the anisotropic thermal conductivity.

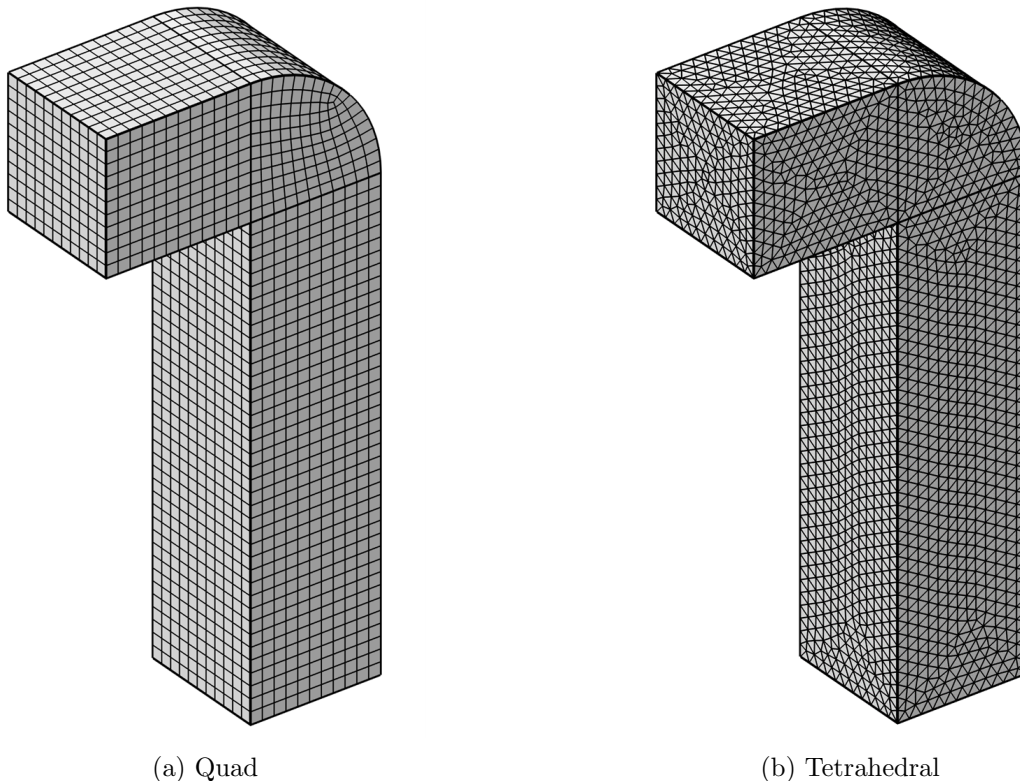
In Fig. 6.1 the designed benchmark is shown. The main geometrical difference compared to the isotropic benchmark is the curvature radius in the outer corner. It can be noticed that the inner corner curvature radius is neglected due to the negligible dimension in comparison with the outer one.

The more important difference is the thermal conductivity anisotropy. To model this anisotropy of the geometrical shape in Fig. 6.4(a) one can use an adaptive curvilinear coordinate system in COMSOL which follows the rounded corner shape as can be seen in Fig.

6.1(c). In a real tape-wound core, due to the electrical insulation between the layers that also has a low thermal conductivity, the effective thermal conductivity is much lower normal to the laminations than within them. Hence, there is a low  $k$  in the direction of the blue arrows ( $z$  direction of the curvilinear coordinates), Fig. 6.1(c).

Another alternative to the curvilinear coordinates is that, one could create a local cylindrical coordinate system for the corner element alone, with the origin situated exactly at the inner corner point. Then, the thermal conductivity can be set to a low value in the  $r$  direction for the corner element and in  $z$  and  $x$  direction for the yoke and leg respectively. Both methods work, however, curvilinear coordinates is the recommended method by COMSOL hence it was chosen in this work.

For this anisotropic case a mesh study was performed. Two mesh types have been compared: tetrahedral and quad. It was found by investigation (an analytical case was solved for both meshes) that the quad mesh gives the exact analytical result. The tetrahedral mesh gives a negligible error in the order of 0.1%. It is assumed that the slight discrepancies have to do with the same reason as stated in section 4.3: due to large height-width ratios the tetrahedral mesh introduces inaccuracies and convergence issues. Another possible reason for the more accurate performance of the quad mesh is that the interpolation functions of the elements are exactly aligned with the solution of the analytical problem investigated (case 1 and 3 in table 6.1). The quad mesh has been checked for different densities and no significant deviations have been found between them. Even though both mesh types can be used, the quad mesh is preferred because it passes the analytical test with no error, thus providing more certainty for the latter simulations.



**Figure 6.2:** Two mesh options for the anisotropic benchmark

Depending on the thermal conductivity normal to the laminations (blue arrows,  $z$  direction of the curvilinear coordinates, Fig. 6.1(c)), the heat transfer of the anisotropic benchmark case can either be represented by a PDE in 2D or 3D. In the scenario where  $k$  has a non-zero value in the direction normal to the laminations then the heat transfer occurs in all three dimensions

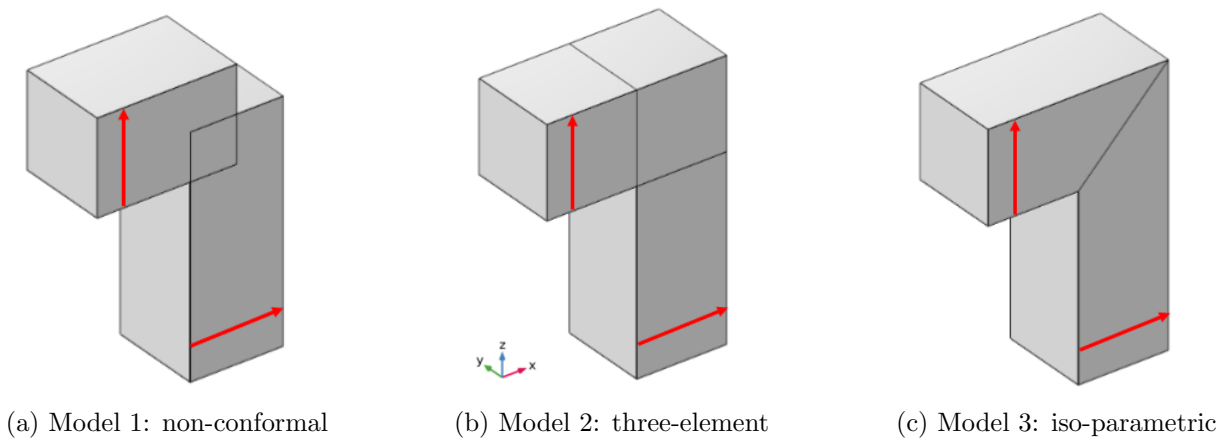
and thus 3D elements should be used for the model. For the sake of simplicity this will be shown for  $k=0$  in section 6.1.2. The disadvantage of this approach is that a simple model, such as model 2 for the isotropic case does not exactly cover the geometry of the benchmark. In order to do so, a specific element type should be formulated for the rounded corner element.

In the other scenario, a very simple approach is to consider  $k=0$  in the direction normal to the laminations and have a 2D heat transfer PDE. In this case one can model the benchmark with simple and fast 2D elements. This will be shown in section 6.1.3.

## 6.1.2 3D Models

### Model Variants

The benchmark can be modeled using the same models developed in Chapter 5 with adjustments of the thermal anisotropy for each element. For all the models, the yoke element has a low  $k$



**Figure 6.3:** Anisotropic models indicating with red arrows the direction where  $k=0$  for each element.

in the  $z$  direction and the leg element in the  $x$  direction, see Fig 6.3. For model 2 the corner element is isotropic in all directions. This is a simplification made to avoid the complexity of introducing cylindrical coordinates in the corner element formulation.

### Test Cases

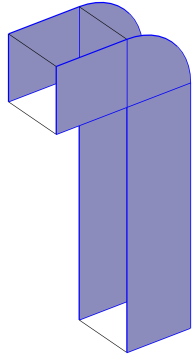
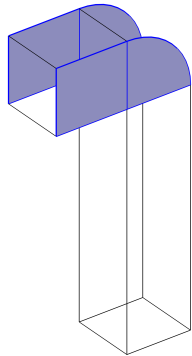
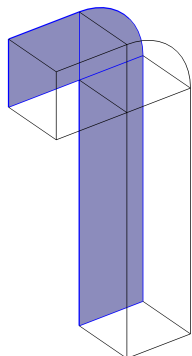
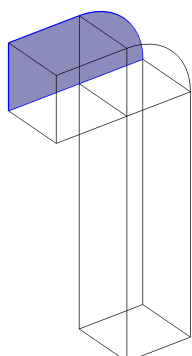
The BC test case for the anisotropic case are depicted in Table 6.1. As for the HTC values, the same will be used as for the isotropic case: 10, 20, 50  $W/m^2K$ . A typical thermal conductivity for nanocrystalline and amorphous materials is 9  $W/(mK)$ . Considering that no significant differences have been noticed between the two different geometries in the isotropic case, only one geometry is investigated. The core window height is 0.65 m and the width is 0.2 m. The cross-section is rectangular with the edge of 0.09 m.

### Models' Accuracy

The error of the different models and BC cases are shown in Fig. 6.5. Cases 1 and Case 3 are rather simple 1D problems which can be solved analytically. Case 1 depicts a parabolic temperature distribution and Case 3, half-parabolic one. This explains the high accuracies in Fig. 6.5 ( $<0.2 \cdot 10^{-9}\%$ )

For cases 2 and 4 we can see higher errors ( $<25\%$ ) because the model has a higher cooled surface compared to the benchmark but also an isotropic corner. Both lead to a better cooling

**Table 6.1:** Test cases implying different cooling BCs. The blue surfaces represent the convective heat transfer BCs. Notice that compared to the isotropic cases, the convection flux on surfaces in the laminations plane has been set to adiabatic. This is a simplification made because ideally the thermal conductivity in the direction normal to the laminations is 0 and there is no heat transfer occurring in this direction.

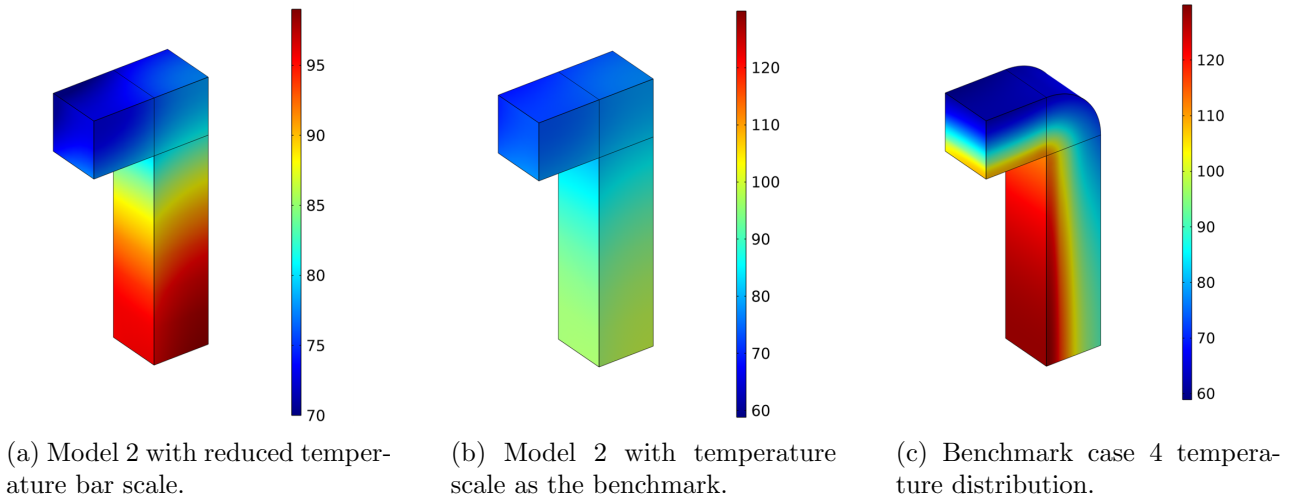
Case	Description	Representation
1	One core layer, well cooled. Convection on all surfaces except for the surfaces that connect to the other part of the integral core.	
2	One core layer, bobbin covering leg.	
3	Outer core layer of $\geq 2$ core layers, air flow in z-dir., well cooled.	
4	Outer core layer of $\geq 2$ core layers, bobbin covering leg.	



and a lower hot spot. This can be fixed by applying a correction factor to the HTC coefficient of the corner convective surface. This factor is the convective area on the corner element divided by the convective area on the benchmark rounded corner:

$$Corr_{factor} = \frac{A_{cool-Mat}}{A_{cool-Corner}} = \frac{pi}{4} = 1.27 \quad (6.1)$$

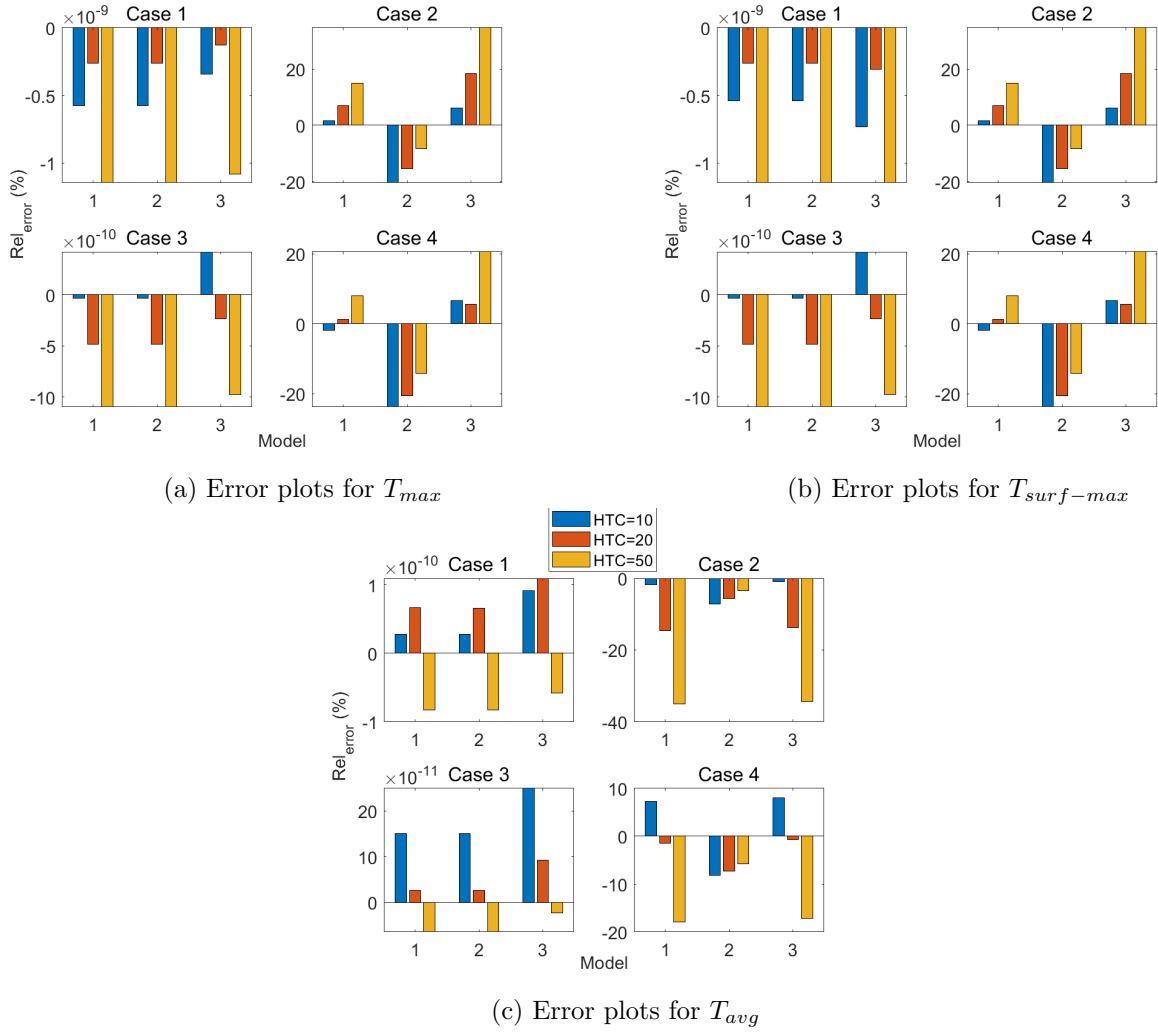
As a result of this, the errors are reduced to  $< 16\%$ . There is still a significant error because model 2 has a changed physics (T distribution) compared to the benchmark due to the isotropic corner. This distribution is seen in Fig. 6.4. In the benchmark model the hot spot is in the inner part of the leg. This is because the anisotropy is present in the whole L core. If one imagines one sheet section of lamination on the inner part of the leg this will be cooled by a sheet section in the inner yoke. However, if one looks at a sheet section on the outer part of the leg this gets cooled by a sheet section of the yoke plus a sheet section of the rounded corner. This explains why the outer part of the leg is better cooled and the inner part gets hotter. On the contrary, in model 2 we have used an isotropic corner which hinders the same behavior as in the benchmark. One notices that the hotspot is now on the outer part of the leg. Also, the isotropic corner leads to an overall higher cooling for model 2.



**Figure 6.4:** Comparison plots between model 2 and benchmark with BC case 4.

### Alternative 3D Model for Error Improvement

To improve the 3D model 2 even more one could formulate an iso parametric element for the corner as shown in Fig. 6.6 In this configuration the upper left most edge is shifted on to the curvature radius of the equivalent benchmark corner. This way the error in the volume and surface in the corner is decreased. One can check this by making a 3-element mesh model in COMSOL and comparing it with the benchmark. This was done for BC case 4. Now, the error is decreased from 25% to less than 10%. As mentioned previously, one would not be able to completely remove the inaccuracy when an isotropic corner element is used. This alternative model with an iso parametric corner is only exemplified in COMSOL, however it was not formulated in MATLAB. Such an element could be formulated similarly to the iso-parametric tapered element in section 4.3. An even better solution or perhaps the best would be to formulate a half cylinder element in cylindrical coordinates which would represent the geometrical shape of the corner exactly. Such a formulation is hinted in [33] and could be part of the future work.

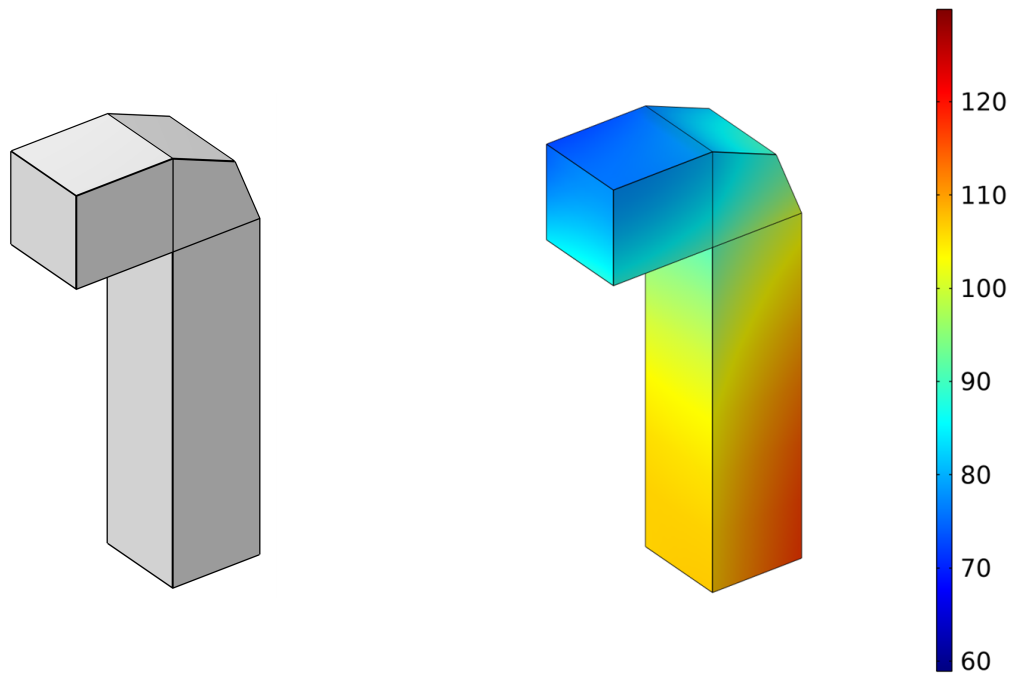


**Figure 6.5:** Error plots for the three important temperatures, of the anisotropic core (the same legend holds for all subfigures).

### 6.1.3 2D Model

In case of the tape-wound cores the thermal conductivity in the direction normal to the laminations is much lower compared to the other two directions. In such a case where  $k$  can be considered 0, the problem can be reduced to a purely 2D nature. To model accurately the benchmark in Fig. 6.4(c) in 2D one can model the hot and the cold surfaces independently using 2D 9-node elements. This is illustrated in Fig. 6.7. The  $T_{max}$  and  $T_{surf-max}$  is determined using the hot surface. The  $T_{avg}$  is calculated by averaging both hot and cold surfaces.

The accuracy of the 2D model can be determined for BC case 2 and 4 in table 6.1. Case 1 and 3 can be left out as they can be purely modelled analytically. The errors of these models are within 4.2%, much lower than the previous 3D models thanks to using the assumption of  $k=0$  normal to the laminations. A first question that comes out is: Why are the 2D models less accurate compared to the 3D models for the isotropic cores? Should they not be very accurate as well to model the parabolic temperature field on the hot surface of the anisotropic benchmark for instance? It is believed that the 2D models should perform with higher accuracy and that the present errors of  $< 4.2\%$  could be mostly because of a benchmark artefact which is not yet understood. Because the mesh independence of the benchmark results was confirmed, one could run more simulations and investigate the impact of different settings for the curvilinear coordinates on the 3D benchmark result. It is believed that the artefact has to do with the use



(a) Implementation of model 2 in COMSOL with a new corner element. (b) Temperature distribution of the new alternative for model 2.

**Figure 6.6:** An alternative for model 2 for accuracy improvement.

of the curvilinear coordinate system whose effects are not fully understood.

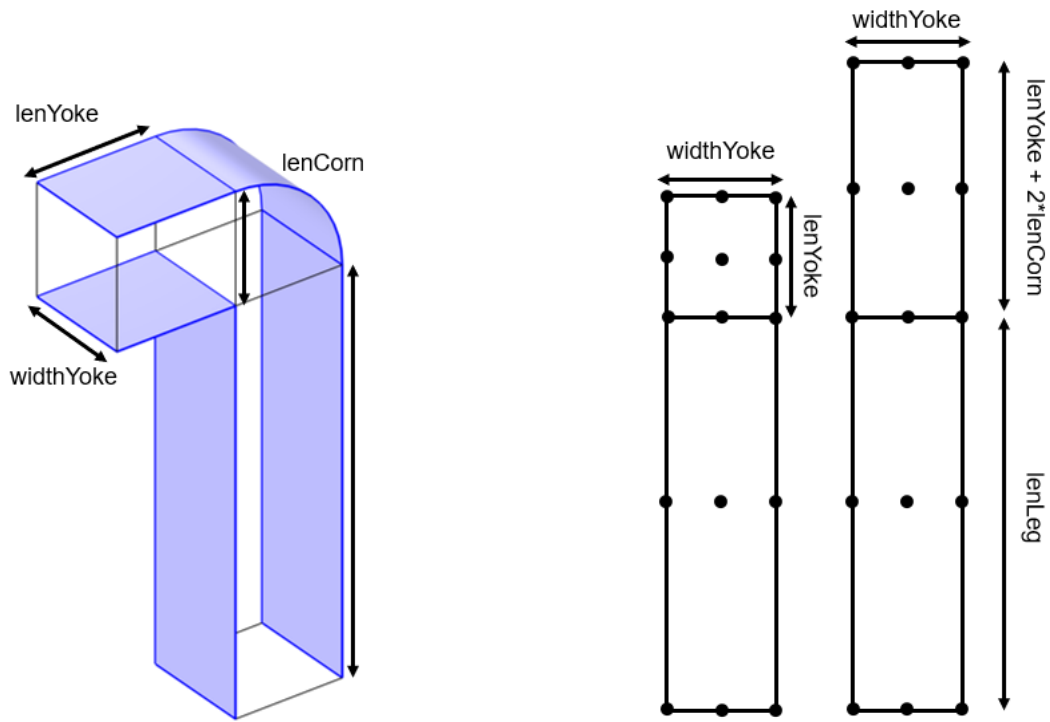
#### 6.1.4 Alternative 2D Benchmark

Thanks to the property of  $k=0$  in the direction normal to the laminations one could even simplify the COMSOL benchmark from a 3D to a 2D problem. In this case the benchmark will be a fine-mesh of the hot and cold surfaces independent from each other. This is shown in Fig. 6.10 for the hot surface as an example.

If one uses this 2D finely-meshed benchmark to test the accuracy of the 2D two element models, high accuracy, within 0.8% error, is obtained. The errors are plotted from peak to ambient in Fig. 6.9. Again, this confirms that with only several FEs one can model parabolic or close to parabolic temperature fields.

Compared to the 3D benchmark, the 2D benchmark is a conservative option because it is only valid for  $k=0$  or very low values of  $k$ . Still, since there is more confidence in the 2D benchmark results it is preferred to have this as a comparison for our 2D coarse mesh models. As future work, one could develop a more confident 3D benchmark for the tape-wound material. Moreover, one could upgrade the 3D coarse models with a rounded corner element in cylindrical coordinates as discussed and benchmark it against the 3D model. The advantage of such progress is that one could take into account cases where  $k$  has a finite value instead of treating it as 0.

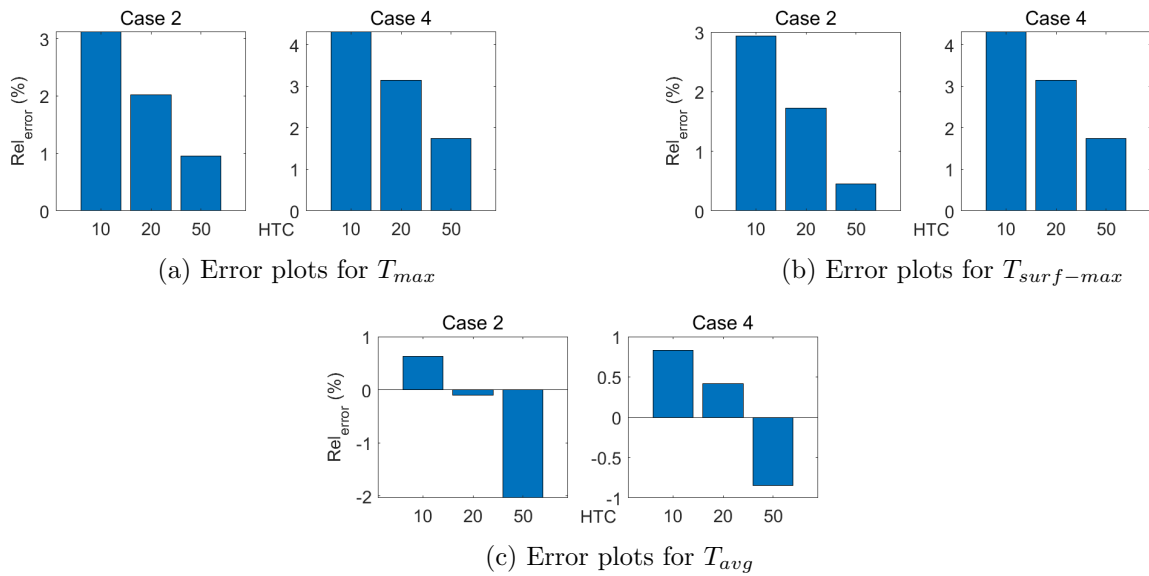
The 2D models are a simple solution, however conservative, as they are only valid when very low thermal conductivity in the direction normal to the laminations is considered.



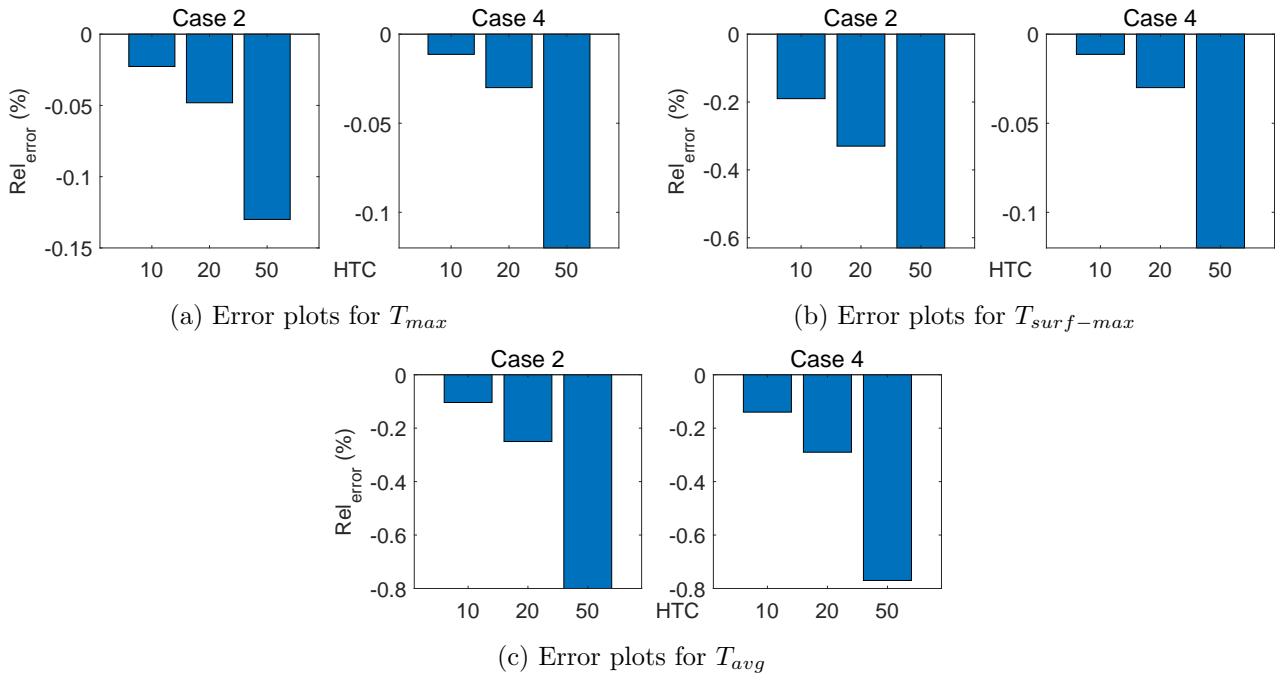
(a) 3D benchmark surfaces to be modelled with 2D elements

(b) The 2D models.

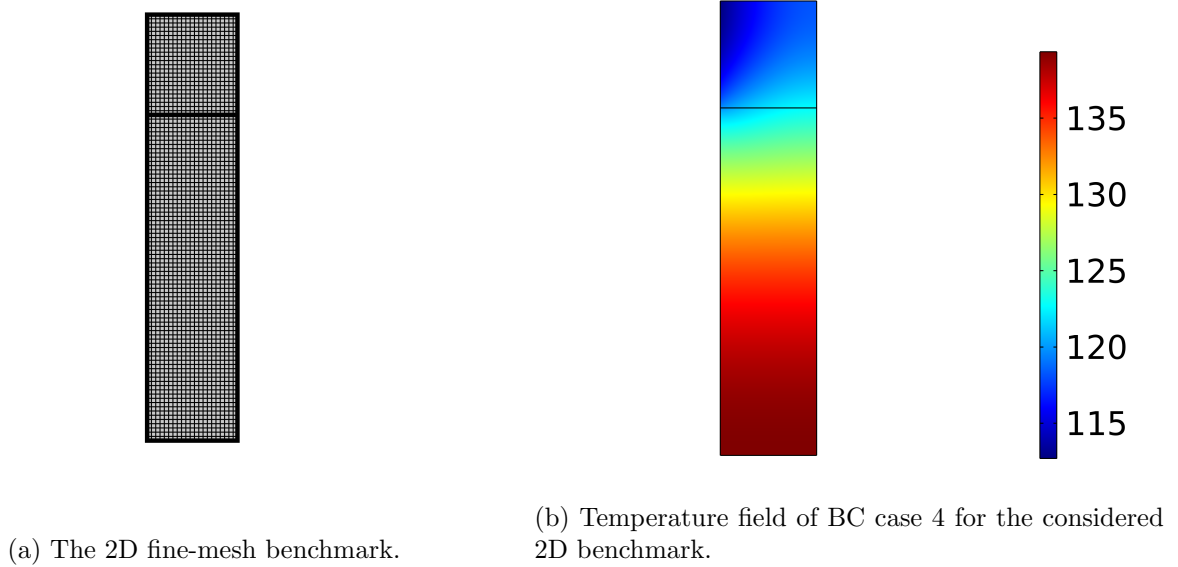
**Figure 6.7:** 2D coarse model of the 3D anisotropic benchmark.



**Figure 6.8:** Error plots for the three important temperatures of the 2D model approach for the BC case 2 and 4 benchmarked against the 3D COMSOL model.



**Figure 6.9:** Error plots for the three important temperatures of the 2D model approach for the BC case 2 and 4 benchmarked against the alternative 2D COMSOL model in Fig. 6.10 .



(a) The 2D fine-mesh benchmark.

(b) Temperature field of BC case 4 for the considered 2D benchmark.

**Figure 6.10:** An alternative 2D benchmark for the developed 2D models.

# Discussion & Conclusions

The development of accurate and fast thermal models for magnetic transformer cores is crucial for the optimization of power electronics. This work introduces a novel method leveraging FE theory to model magnetic cores efficiently with a minimal number of elements, addressing both bulk and tape-wound core types. This section discusses the implications, advantages, and potential improvements of the proposed method, and their significance in power electronics optimization.

## Summary of the Proposed Method

The thermal modelling method in this work can be generalized for any application in the following steps:

### 1. Phase 1: Problem Definition and Discretization

- (a) Write the DE of the physical phenomenon to be modelled (e.g. heat diffusion equation).
- (b) Transform the DE into a FE formulation using one of the methods in 3.2.5

### 2. Phase 2: Elements Selection and Formulation

- (c) Choose suitable finite element types according to the geometry to be modelled and the BC cases. Possible elements have been shown in this work for 1D, 2D and 3D scenarios. Think how many elements are required to accurately model the geometry and account for various BC cases. Also, consider using minimum number of elements to avoid unnecessary computational effort.
- (d) Formulate the chosen elements resulting in writing the interpolation functions  $N$ .
- (e) Compute the symbolic part of element stiffness matrix and force vectors (the integrals which do not depend on geometrical constants or other non-trivial parameters) by using symbolic math and store these variables for re-use purposes. As an example for the 3D case, see eq. 4.38.

### 3. Phase 3: Assembly and Solution

- (f) Compute the full element stiffness matrix and force vectors using the non-trivial parameters such as geometrical constants, thermal conductivity, HTC etc.
- (g) Label element nodes and assemble the element stiffness matrix and force vector into a global stiffness matrix and force vector.
- (h) Solve for the unknown nodal values (e.g. temperature).

Phase 1 is the initial problem definition where the basis is set. Phase 2 is the problem modelling where all the decisions about the element types and model formulations are made. Phase 3 is the final computational stage. While phases 1 and 2 serve as a modelling and formulation stage which is time consuming, phase 3 yields the final fast model which is the one to be implemented by a power electronics optimization tool.

## Advantages of the Proposed Method

The primary advantage of the method presented in this thesis is its efficiency and accuracy in modeling thermal behaviors of magnetic cores using a minimal number of finite elements. This is achieved by formulating specific elements tailored to the geometry of the components. Standard rectangular quadratic Lagrange elements in 2D (9-nodes) and 3D (27-nodes) were used, along with a reduced 18-node formulation for 3D cases due to inherent symmetry conditions. An iso-parametric element with an irregular trapezoidal shape was also demonstrated, showing benefits in reducing the number of elements required. For bulk-type magnetic cores, a simple O-core shape was modeled using quadratic Lagrange and symmetrical tailored elements, showing three possible ways to represent the core. The second model, benchmarked against a fine mesh COMSOL model, achieved high accuracy with errors within 1.25%. This demonstrates the method's capability to provide precise thermal analysis with significantly reduced computational effort. Similarly, for tape-wound cores, the method was applied successfully using two approaches. While 3D models showed high deviations compared to COMSOL benchmarks due to necessary simplifications, the 2D models, which are appropriate when thermal conductivity normal to the laminations is low, achieved errors within 0.8%. This validates the applicability of the proposed method in scenarios where the simplifications align with the physical properties of the materials involved.

The above results present significant advancements for optimization tools. While the present thermal modelling techniques have limited accuracy the new approach presented in this work has shown significant improvements. For instance, in [1] a thermal network analytical model is presented for bulk cores which achieves accuracy of 2%, however limited to cases where 2D simplifications are possible. If 3D scenarios are employed for such models, errors of 10% are obtained [6]. In [7] thermal network models have been applied to tape-wound cores with errors of approximately 6%. Moreover, for all examples above, the errors are obtained for only one parameter which is denoted as "temperature rise", its meaning being vague. In contrast, the new approach presented in this work achieves errors within 1.25% and 0.8% for bulk and tape-wound cores respectively. To have a fair comparison of accuracy between the presented models and the literature one would need to have identical benchmarking cases and situations. Still, the new approach presents a high accuracy for modelling bulk and tape-wound materials and is able to distinguish between three significant temperature parameters: volume and surface maximum, and average. At the same time, the models are fast, can cover huge design spaces, hence, they are suitable for design optimization applications.

All in all, the improvements in accuracy and complexity of the thermal models above present important steps to maximizing design efficiency of transformers and to advance the current state of the power electronic converters.

## Computational Efficiency

A significant benefit of this method is its computational efficiency. By pre-computing and re-using key element parameters such as stiffness matrices and force vectors, the models can execute within 0.28 ms on average-specification computers. Another step taken to improve the computational efficiency was reducing the element size to a faster 18-node element. Similarly,

it was shown that the method to solve the linear system is also important. LU decomposition saves significant computational effort when different operating points are investigated.

This efficiency considerations is particularly advantageous in optimization tools, where multiple iterations are required to find the optimal transformer design in terms of size, efficiency, and cost.

## Limitations and Future Work

While the proposed method demonstrates substantial improvements in modeling efficiency and accuracy, there are inherent limitations. For instance, the 3D models for tape-wound cores showed high deviations due to simplifications. Future work could focus on enhancing these models by incorporating more complex element formulations that better capture the anisotropic thermal properties of tape-wound cores. Another area for improvement is the integration of these models into broader optimization frameworks. While the models are currently validated against COMSOL benchmarks, incorporating real-world data and operational conditions could further refine their accuracy and applicability. For instance, how can the current models be adapted to cover real scenarios such as when only part of the leg area is covered by the windings. This would probably require some heat-transfer coefficient corrections/adaptable algorithms or extending the models with additional FEs. Additionally, extending the method to model other types of magnetic components, such as planar magnetic components and inductors, could broaden its utility in power electronics design.

Moreover, the method can be extended to isotropic core of different types and shapes such as E-type cores or toroidal shapes. Also, the tape-wound cores could also be extended to various shapes and types such as shell and toroidal. The stacked core types have not been explored yet and the proposed method could be extended to these core types as well.

Considering the focus are of this work: mainly magnetic cores, the thermal models should be further extended to incorporate other relevant components of transformers and inductors, windings, since they represent an important part of the optimized design.

## Conclusion

In conclusion, the novel FE-based method for modeling magnetic transformer cores introduced in this work offers significant advancements in accuracy. By formulating and utilizing a minimal number of tailored finite elements, this method can effectively model both bulk and tape-wound cores, making it a valuable tool for optimizing magnetic components designs. The work provides engineers and researchers with a quick and effective method to set-up coarse FE models without deep understanding of FEM. This is especially useful in applications where FEM softwares cannot be used because of their computational inefficiency. Future work should aim to address the current limitations and explore further applications of the method to enhance its robustness and versatility in power electronics optimization.



# Bibliography

- [1] M. Mogorovic, “Modeling and design optimization of medium frequency transformers for medium-voltage high-power converters,” *Faculté des sciences et techniques de l’ingénieur Laboratoire d’électronique de puissance Programme doctoral en energie pour l’obtention du grade de Docteur ès Sciences*, August 2019.
- [2] R. W. Erickson and D. Maksimovic, *Fundamentals of Power Electronics*, 2nd ed. Springer Science & Business Media, 2001.
- [3] S. Kulkarani and S. Khaparde, *Inductors and Transformers for POver Electronics*, 1st ed. CRC Press, 2012.
- [4] Wikipedia, *Ferrite core*. Wikipedia: Available: [https://en.wikipedia.org/wiki/Ferrite\\_core](https://en.wikipedia.org/wiki/Ferrite_core), 2023.
- [5] mag inc, *Powder Cores, Ferrite Cores, & Tape Wound Cores*. [Online]. Available: <https://www.mag-inc.com/products/powder-cores>, 2023.
- [6] A. Fouineau, M.-A. Raulet, M. Guillet, F. Sixdenier, and B. Lefebvre., “A medium frequency transformer design tool with methodologies adapted to various structures,” *2020 Fifteenth International Conference on Ecological Vehicles and Renewable Energies (EVER)*, vol. 38, pp. 40–53, September 2020.
- [7] S. James, W. Yiren, C.-L. Gerardo, and F. Andrew, “Rapid thermal analysis of nanocrystalline inductors for converter optimization,” *IEEE Journal of Emerging and Selected Topics in Power Electronics*, vol. 8, no. 3, pp. 2276 – 2284, 2020.
- [8] A. Delgado, L. Clavero, M. Vasic, P. Alou, M. Bakic, and T. Wijekoon, “3d thermal modeling of inductive power transfer coils based on basic thermal network for optimization analysis,” *2023 IEEE Applied Power Electronics Conference and Exposition (APEC)*, 06 2023.
- [9] L. C. Ordonez, A. Delgado, P. A. Cervera, and T. Wijekoon, “Fast and accurate analytical thermal modeling for planar pcb magnetic components,” *IEEE TRANSACTIONS ON POWER ELECTRONICS*, vol. 38, pp. 7480–7491, June 2020.
- [10] K. Lai-Dac, Y. Lembeye, and B. Sarrazin, “Two free air convection and radiation thermal models for planar magnetic components,” *IEEE Transactions on Power Electronics*, pp. 7480–7491, 2011.
- [11] M. Slavkova and K. Milanov, “Analytical model for thermal analysis of toroidal chokes and transformers,” in *Proceedings of the 14th Electrical Engineering Faculty Conference (BulEF)*, 2022, pp. 1–4.

- [12] L. Nie, J. Yang, and K. Tang, “Thermal network modeling of high frequency insulated core transformers,” *IEEE Transactions on Applied Superconductivity*, vol. 32, no. 6, September 2022.
- [13] Z. Shen, Y. Shen, and H. Wang, “Thermal modelling of planar transformers considering internal power loss distribution,” in *IEEE Transactions*, 2019, pp. 7480–7491.
- [14] Y. Chen, Z. Zhang, C. Zhang, W. Ying, Y. Li, and Q. Yang, “Comparison between thermal-circuit model and finite element model for dry-type transformer,” in *2019 22nd International Conference on Electrical Machines and Systems (ICEMS)*, Tianjin, China, 2019, pp. 1–5.
- [15] K. Górecki and K. Górski, “Non-linear thermal model of planar transformers,” in *European Microelectronics Packaging Conference (EMPC)*. Warsaw, Poland: IMAPS / EMPC, September 2017, pp. 1–5.
- [16] D. Molinero, D. Santamargarita, E. Bueno, M. Marrón, and M. Vasic, “Artificial neural network based thermal model for a three-phase medium frequency transformer,” *IEEE International Symposium on Industrial Electronics*, 2023.
- [17] D. Santamargarita, G. Salinas, D. Molinero, E. J. Bueno, and M. Vasic, “Tradeoff between accuracy and computational time for magnetics thermal model based on artificial neural networks,” *IEEE Journal of Emerging and Selected Topics in Power Electronics*, vol. 11, no. 6, pp. 5658–5670, 2023.
- [18] Y. Wang, Y. Yang, and W. Chen, “A neural network model for electro-thermal effect affected hot carrier injection reliability in n-type nanoscale mosfets,” in *2022 Asia-Pacific International Symposium on Electromagnetic Compatibility (APEMC)*, 2022, pp. 652–654.
- [19] A. K. Gupta and P. Mathur, “Accurate density prediction of nanofluids using an optimized artificial neural network for enhanced heat transfer and energy systems,” in *2023 10th International Conference on Electrical Engineering, Computer Science and Informatics (EECSI)*, 2023, pp. 613–617.
- [20] Y. Zhang, Z. Wang, H. Wang, and F. Blaabjerg, “Artificial intelligence-aided thermal model considering cross-coupling effects,” *IEEE Transactions on Power Electronics*, vol. 35, no. 10, pp. 9998–10 002, 2020.
- [21] X. Yang, J. Tan, B. Zhou, D. Hua, G. Tang, and T. Qiao, “Thermal-stress analysis of 3d-ic based on artificial neural network,” in *2019 IEEE 4th International Conference on Cloud Computing and Big Data Analysis (ICCCBDA)*, 2019, pp. 105–110.
- [22] G. S. Lopez, “Thermal modelling of high-frequency magnetic components for power electronics by finite element analysis,” Ph.D. dissertation, Universidad Politecnica de Madrid, Madrid, 4 2020, section 3.2: FEA-based static thermal model.
- [23] S. Maity, B. Chatterjee, S. Maur, S. Chatterjee, and S. Dalai, “Electromagnetic and thermal modelling of a dry-type transformer using femm and ansys maxwell,” in *IEEE 2nd International Conference on Industrial Electronics: Developments & Applications (ICIDeA)*. Aizawl, India: IEEE, 2023.
- [24] Y. Chen, Q. Yang, C. Zhang, Y. Li, and X. Li, “Thermal network model of high-power dry-type transformer coupled with electromagnetic loss,” *IEEE Transactions on Magnetics*, vol. 58, no. 11, pp. 1–5, 2022.

- [25] Y. Jiawen, Y. Hu, J. Li, and X. Han, “3d fem simulation study of a single-phase three-limb transformer under dc bias,” in *2023 2nd International Conference on Power Systems and Electrical Technology (PSET)*, 2023, pp. 131–136.
- [26] Y. B. Zhang, Y. L. Xin, T. Qian, X. Lin, W. H. Tang, and Q. H. Wu, “2-d coupled fluid-thermal analysis of oil-immersed power transformers based on finite element method,” in *2016 IEEE Innovative Smart Grid Technologies - Asia (ISGT-Asia)*, 2016, pp. 1060–1064.
- [27] E. Mechkov, R. Tzeneva, V. Mateev, and I. Yatchev, “Thermal analysis using 3d fem model of oil-immersed distribution transformer,” in *2016 19th International Symposium on Electrical Apparatus and Technologies (SIELA)*, 2016, pp. 1–3.
- [28] L. Clavero, A. Delgado, P. Alou, M. Bakic, and T. Wijekoon, “Improved analytical core temperature prediction based on estimation of the non-uniform flux distribution,” in *2023 25th European Conference on Power Electronics and Applications (EPE'23 ECCE Europe)*, 2023, pp. 1–8.
- [29] R. Bouhaddiche, S. Bouazabia, and I. Fofana, “Thermal modelling of power transformer,” in *2017 IEEE 19th International Conference on Dielectric Liquids (ICDL)*, 2017, pp. 1–4.
- [30] Z. Shen, Y. Shen, B. Liu, and H. Wang, “Thermal coupling and network modeling for planar transformers,” in *2018 IEEE Energy Conversion Congress and Exposition (ECCE)*, 2018, pp. 3527–3533.
- [31] T. L. Bergman, A. S. Lavine, F. P. Incropera, and D. P. DeWitt, *Fundamentals of Heat and Mass Transfer*. John Wiley & Sons, 2011.
- [32] G. Christian, R. Hans-G., and S. Martin, *Numerical Treatment of Partial Differential Equations*, 1st ed. Springer Science Business Media, 2007.
- [33] D. V. Hutton, *Fundamentals of Finite Element Analysis*. Oxford: McGraw-Hill, 2011.
- [34] V. H. Atlas, Ed., *VDI-Gesellschaft Verfahrenstechnik und Chemieingenieurwesen*, 2nd ed. Springer, 2010.
- [35] E. Fontes, *COMSOL Blog: FEM vs. FVM*, 1st ed. COMSOL, 2018.
- [36] O. Zienkiewicz, *The Finite Element: Its Basis and Fundamentals*. Oxford: McGraw-Hill, 1967.

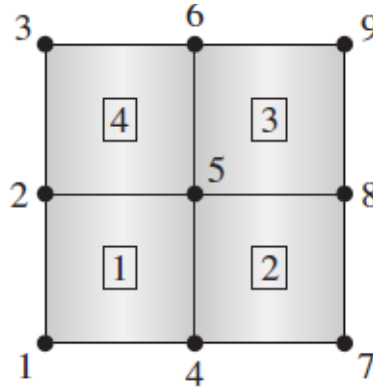
# Appendix

## A.1 Stiffness Matrix Assembling Procedure

For solving an FEM problem, the stiffness matrixes of each element must be assembled in a global stiffness matrix. This stands also for the force vector.

This method illustrates the assembling procedure of the four 4-node elements in section 4.5.1.

The 2 by 2 square is divided in four elements with the following node numbers as shown in:



**Figure A.1:** The four linear elements and the nodal numbers [33].

Element 1 has on a general form the following stiffness matrix:

$$k_{e1} = \begin{bmatrix} k_{11}^1 & k_{12}^1 & k_{13}^1 & k_{14}^1 \\ k_{21}^1 & k_{22}^1 & k_{23}^1 & k_{24}^1 \\ k_{31}^1 & k_{32}^1 & k_{33}^1 & k_{34}^1 \\ k_{41}^1 & k_{42}^1 & k_{43}^1 & k_{44}^1 \end{bmatrix} \quad (\text{A.1})$$

A global matrix is then initiated that has the rows and the columns indexed from 1 till 9. First the nodal numbers are placed in a vector in an anti-clockwise direction as following:

$$L_i = [ i_1 \quad i_4 \quad i_5 \quad i_2 ] \quad (\text{A.2})$$

A clockwise order can also be chosen but then this must be respected for all elements. Next an index matrix is initiated using the vector above as following.

$$L_{ii} = \begin{bmatrix} i_{11} & i_{14} & i_{15} & i_{12} \\ i_{41} & i_{44} & i_{45} & i_{42} \\ i_{51} & i_{54} & i_{55} & i_{52} \\ i_{21} & i_{24} & i_{25} & i_{22} \end{bmatrix} \quad (\text{A.3})$$

Then element  $k_{11}^1$  will take the place  $i_{11}$  in the global stiffness matrix. Element  $k_{12}^1$  will take the entry  $i_{14}$  and so on.

In MATLAB this can be executed with the following lines:

```
1 Li = [1 4 5 2];
2 K(L1,Li) = ke1;
```

The resulting global matrix is then:

$$K = \begin{bmatrix} k_{11}^1 & k_{14}^1 & 0 & k_{12}^1 & k_{13}^1 & 0 & 0 & 0 & 0 \\ k_{41}^1 & k_{44}^1 & 0 & k_{42}^1 & k_{43}^1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ k_{21}^1 & k_{24}^1 & 0 & k_{22}^1 & k_{23}^1 & 0 & 0 & 0 & 0 \\ k_{31}^1 & k_{34}^1 & 0 & k_{32}^1 & k_{33}^1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \quad (\text{A.4})$$

In a similar manner, the other element stiffness matrixes are added to the global matrix. The same analogous procedure is to be followed for assembling the global force vector.

## A.2 Interpolation Functions

**Table A.1:** Interpolation functions for a linear 8-node and quadratic 20-node element.

	8-node linear element	20-node quadratic serendipity element
N1	$\frac{1}{8}(1-r)(1-s)(1+t)$	$-\frac{1}{8}(r-1)(s-1)(t+1)(r+s-t+2)$
N2	$\frac{1}{8}(1+r)(1-s)(1+t)$	$-\frac{1}{8}(r+1)(s-1)(t+1)(r-s+t-2)$
N3	$\frac{1}{8}(1+r)(1+s)(1+t)$	$\frac{1}{8}(r+1)(s+1)(t+1)(r+s+t-2)$
N4	$\frac{1}{8}(1-r)(1+s)(1+t)$	$\frac{1}{8}(r-1)(s+1)(t+1)(r-s-t+2)$
N5	$\frac{1}{8}(1-r)(1-s)(1-t)$	$\frac{1}{8}(r-1)(s-1)(t-1)(r+s+t+2)$
N6	$\frac{1}{8}(1+r)(1-s)(1-t)$	$-\frac{1}{8}(r+1)(s-1)(t-1)(r-s+t+2)$
N7	$\frac{1}{8}(1+r)(1+s)(1-t)$	$-\frac{1}{8}(r+1)(s+1)(t-1)(r+s-t-2)$
N8	$\frac{1}{8}(1-r)(1+s)(1-t)$	$-\frac{1}{8}(r-1)(s+1)(t-1)(r-s+t+2)$
N9		$\frac{1}{4}(r^2-1)(s-1)(t+1)$
N10		$-\frac{1}{4}(s^2-1)(r+1)(t+1)$
N11		$-\frac{1}{4}(r^2-1)(s+1)(t+1)$
N12		$\frac{1}{4}(s^2-1)(r-1)(t+1)$
N13		$-\frac{1}{4}(t^2-1)(r-1)(s-1)$
N14		$\frac{1}{4}(t^2-1)(r+1)(s-1)$
N15		$-\frac{1}{4}(t^2-1)(r+1)(s+1)$
N16		$\frac{1}{4}(t^2-1)(r-1)(s+1)$
N17		$-\frac{1}{4}(r^2-1)(s-1)(t-1)$
N18		$\frac{1}{4}(s^2-1)(r+1)(t-1)$
N19		$\frac{1}{4}(r^2-1)(s+1)(t-1)$
N20		$-\frac{1}{4}(s^2-1)(r-1)(t-1)$

**Table A.2:** Interpolation functions for 2D 9-node element and 3D 27-node element.

	2D 9-node element	3D 27-node element
$N_1$	$\frac{1}{4}(1-r)(1-s)rs$	$\frac{1}{8}(1-r)(1-s)(1+t)rst$
$N_2$	$-\frac{1}{4}(1+r)(1-s)rs$	$-\frac{1}{8}(1+r)(1-s)(1+t)rst$
$N_3$	$\frac{1}{4}(1+r)(1+s)rs$	$\frac{1}{8}(1+r)(1+s)(1+t)rst$
$N_4$	$-\frac{1}{4}(1-r)(1+s)rs$	$-\frac{1}{8}(1-r)(1+s)(1+t)rst$
$N_5$	$-\frac{1}{2}(1-r^2)(1-s)s$	$-\frac{1}{4}(1-r^2)(1-s)(1+t)st$
$N_6$	$-\frac{1}{2}(1-s^2)(1+r)r$	$-\frac{1}{4}(1-s^2)(1+r)(1+t)rt$
$N_7$	$\frac{1}{2}(1-r^2)(1+s)s$	$\frac{1}{4}(1-r^2)(1+s)(1+t)st$
$N_8$	$-\frac{1}{2}(1-s^2)(1-r)r$	$-\frac{1}{4}(1-s^2)(1-r)(1+t)rt$
$N_9$	$(1-r^2)(1-s^2)$	$\frac{1}{2}(1-r^2)(1-s^2)(1+t)t$
$N_{10}$		$\frac{1}{4}(1-r)(1-s)(1-t^2)rs$
$N_{11}$		$-\frac{1}{4}(1+r)(1-s)(1-t^2)rs$
$N_{12}$		$\frac{1}{4}(1+r)(1+s)(1-t^2)rs$
$N_{13}$		$-\frac{1}{4}(1-r)(1+s)(1-t^2)rs$
$N_{14}$		$-\frac{1}{2}(1-r^2)(1-s)(1-t^2)s$
$N_{15}$		$-\frac{1}{2}(1-s^2)(1+r)(1-t^2)r$
$N_{16}$		$\frac{1}{2}(1-r^2)(1+s)(1-t^2)s$
$N_{17}$		$-\frac{1}{2}(1-s^2)(1-r)(1-t^2)r$
$N_{18}$		$(1-r^2)(1-s^2)(1-t^2)$
$N_{19}$		$-\frac{1}{8}(1-r)(1-s)(1-t)rs$
$N_{20}$		$\frac{1}{8}(1+r)(1-s)(1-t)rst$
$N_{21}$		$-\frac{1}{8}(1+r)(1+s)(1-t)rst$
$N_{22}$		$\frac{1}{8}(1-r)(1+s)(1-t)rst$
$N_{23}$		$\frac{1}{4}(1-r^2)(1-s)(1-t)st$
$N_{24}$		$\frac{1}{4}(1-s^2)(1+r)(1-t)rt$
$N_{25}$		$-\frac{1}{4}(1-r^2)(1+s)(1-t)st$
$N_{26}$		$\frac{1}{4}(1-s^2)(1-r)(1-t)rt$
$N_{27}$		$-\frac{1}{2}(1-r^2)(1-s^2)(1-t)t$

### A.3 Implementing 27-node in MATLAB

In this appendix the theory in section 4.4.1 will be translated into MATLAB code. This serves as a recipe for formulating and implementing other type of elements in MATLAB code.

The process involves the symbolic MATLAB tool and starts by writing down the interpolation functions, in this case for the 27-node, shown in Table A.2.

```

1  %% write interpolation functions, 27-node quadratic
2  syms r s t
3
4  N = [1/4*(1-r)*(1-s)*r*s*(1/2*(1+t)-1/2*(1-t^2));
5       1/4*(1+r)*(s-1)*r*s*(1/2*(1+t)-1/2*(1-t^2));
6       1/4*(1+r)*(1+s)*r*s*(1/2*(1+t)-1/2*(1-t^2));
7       1/4*(r-1)*(1+s)*r*s*(1/2*(1+t)-1/2*(1-t^2));
8       1/2*(1-r^2)*(s-1)*s*(1/2*(1+t)-1/2*(1-t^2));
9       1/2*(1-s^2)*(1+r)*r*(1/2*(1+t)-1/2*(1-t^2))...];

```

The next step is the computation of the stiffness matrix. This is done in several steps.

First, the partial derivatives of the interpolation functions in 4.37 with respect to  $r$ ,  $s$ , and  $t$  are each assigned to a different variable. Then, each separate triple integral can be assigned to a separate variable as well.

```

1  %% Calculation of stiffness matrix
2  diffN_r = diff(N,r);
3  diffN_s = diff(N,s);
4  diffN_t = diff(N,t);
5
6  matrix_diffN_r = diffN_r*diffN_r';
7  matrix_diffN_s = diffN_s*diffN_s';
8  matrix_diffN_t = diffN_t*diffN_t';
9
10 int_r = double(int(int(int(matrix_diffN_r,r,-1,1),s,-1,1),t,-1,1)
    );
11 int_s = double(int(int(int(matrix_diffN_s,r,-1,1),s,-1,1),t,-1,1)
    );
12 int_t = double(int(int(int(matrix_diffN_t,r,-1,1),s,-1,1),t,-1,1)
    );

```

The variables  $int_r$ ,  $int_s$  and  $int_t$  can then be stored since their computation is long. The final step is to multiply (after loading them in another function or code e.g. which is used for optimization purposes) these variables with the respective constants and sum them to obtain the stiffness matrix.

```

1  %% Calculation of stiffness matrix
2  k_e = k*b*c*int_r/a + k*a*c*int_s/b + k*a*b*int_t/c;

```

The following step is to calculate the convective flux matrix,  $k_{hS}^{(e)}$  and vector,  $f_{hS}^{(e)}$ . They are represented by closed surface integrals (see eq. 4.40) meaning that it has to be split and the flux must be calculated on each surface. It is then convenient to assign it to a separate variable. This begins with first assigning the right nodes to the specific variable named after the surface it represents (for reference see Fig. A.2). Then, the known coordinate values -1 or 1 is substituted in the interpolation functions. Furthermore, the integrals are calculated and stored as previously explained. The final step is to compute the convective matrix and vector by multiplication with the relevant input variables.

```

1  %% Calculation of convective flux matrix and vector
2  % predefine a vector of symbolic zeroes
3  h_s = N;
4  h_s(2:end)= 0;
5  h_s= h_s(2:end);
6  h_s(deg_free)= 0;
7
8  hs_pos_r = h_s; % this is a column vector of 27 symbolic zeros
9  hs_neg_r = h_s;
10 hs_pos_s = h_s;
11 hs_neg_s = h_s;
12 hs_pos_t = h_s;
13 hs_neg_t = h_s;
14
15 % Define faces for convective bcs
16

```

```

17 pos_r_face = [2 6 3 11 15 12 20 24 21];
18 neg_r_face = [1 8 4 10 17 13 19 26 22];
19 pos_s_face = [4 7 3 13 16 12 22 25 21];
20 neg_s_face = [1 5 2 10 14 11 19 23 20];
21 pos_t_face = [4 8 1 7 9 5 3 6 2];
22 neg_t_face = [22 26 19 25 27 23 21 24 20];
23
24 % substitute the known coordinate values in the interpolation
   functions
25
26 hs_pos_r(pos_r_face) = subs(N(pos_r_face),r,1);
27 hs_neg_r(neg_r_face) = subs(N(neg_r_face),r,-1);
28 hs_pos_s(pos_s_face) = subs(N(pos_s_face),s,1);
29 hs_neg_s(neg_s_face) = subs(N(neg_s_face),s,-1);
30 hs_pos_t(pos_t_face) = subs(N(pos_t_face),t,1);
31 hs_neg_t(neg_t_face) = subs(N(neg_t_face),t,-1);
32
33 % Calculate the integrals for the convective matrix; these are
   the ones to be stored and re-used for each subsequent
   calculation.
34
35 int_pos_r = double(int(int(hs_pos_r*hs_pos_r',s,-1,1),t,-1,1));
36 int_neg_r = double(int(int(hs_neg_r*hs_neg_r',s,-1,1),t,-1,1));
37 int_pos_s = double(int(int(hs_pos_s*hs_pos_s',r,-1,1),t,-1,1));
38 int_neg_s = double(int(int(hs_neg_s*hs_neg_s',r,-1,1),t,-1,1));
39 int_pos_t = double(int(int(hs_pos_t*hs_pos_t',r,-1,1),s,-1,1));
40 int_neg_t = double(int(int(hs_neg_t*hs_neg_t',r,-1,1),s,-1,1));
41
42 % Final step to caclulate convective flux matrix
43
44 pos_r = b*c*int_pos_r;
45 neg_r = b*c*int_neg_r;
46 pos_s = a*c*int_pos_s;
47 neg_s = a*c*int_neg_s;
48 pos_t = b*a*int_pos_t;
49 neg_t = b*a*int_neg_t;
50
51 % In this case all external surfaces are exposed to convective
   flux. In case some surfaces are adiabatic they should for
   exemple be multiplied with 0.
52
53 k_hs_e = h*(pos_r+neg_r+pos_s+neg_s+pos_t+neg_t);
54
55 % Calculate integrals for convective flux vector; these should
   also be stored
56
57 int_pos_rF = double(int(int(hs_pos_r,s,-1,1),t,-1,1));
58 int_neg_rF = double(int(int(hs_neg_r,s,-1,1),t,-1,1));
59 int_pos_sF = double(int(int(hs_pos_s,r,-1,1),t,-1,1));
60 int_neg_sF = double(int(int(hs_neg_s,r,-1,1),t,-1,1));

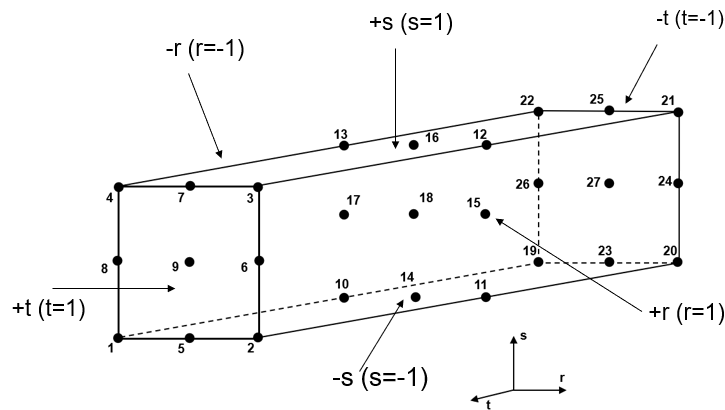
```



```

61 int_pos_tF = double(int(int(hs_pos_t,r,-1,1),s,-1,1));
62 int_neg_tF = double(int(int(hs_neg_t,r,-1,1),s,-1,1));
63
64 pos_r = b*c*int_pos_rF;
65 neg_r = b*c*int_neg_rF;
66 pos_s = a*c*int_pos_sF;
67 neg_s = a*c*int_neg_sF;
68 pos_t = b*a*int_pos_tF;
69 neg_t = b*a*int_neg_tF;
70
71 % In this case all external surfaces are exposed to convective
    flux. In case some surfaces are adiabatic they should for
    exemple be multiplied with 0.
72
73 f_hs_e = h*T_a*(pos_r+neg_r+pos_s+neg_s+pos_t+neg_t);

```



**Figure A.2:** The names of each surface for the example of 27-node implementation in MATLAB.

The following step is to calculate the internal heat generation rate vector by first calculating the integrals and then the constants multiplications.

```

1 % calculate integrals of internal heat gen vector; these should
  be stored.
2
3 Q_int = double(int(int(int(N,r,-1,1),s,-1,1),t,-1,1));
4
5 fq_e = a*b*c*Q*Q_int;

```

The final step is the calculation of the nodal temperatures using one of the algorithm in MATLAB. First the total systems matrix is calculated by adding the stiffness matrix together with the convective flux matrix. The total force vector is the sum of the heat generation and convective flux vectors. Finally, the backslash operator calculates the nodal temperatures.

```

1 %% Calculate nodal temperatures
2
3 K_e = k_e + k_hs_e;
4 F_e = f_hs_e + fq_e;
5
6 T_e = K_e\F_e;

```

## A.4 Assembling Procedure for Model 2: Three-Element

To create model 2, three elements must be assembled. Some considerations come into play when doing so. First, the yoke element must be rotated 180 degrees around s axis (see Fig. 4.6). This is because of the symmetry considerations: the plane of symmetry for this element must be situated at nodes 10-18. Similarly, the leg element is rotated 90 degrees around the x axis for the same reason.

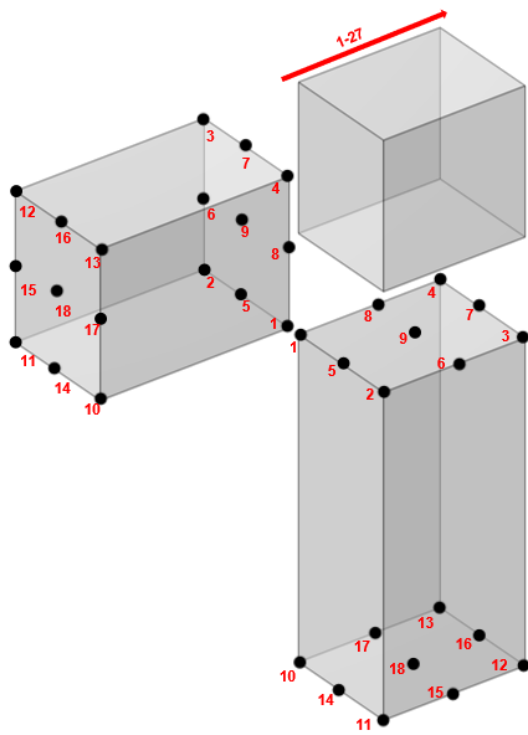
The result of these rotations is shown in Fig. A.3(a) where the local node numbering is shown for each element. The corner element requires no rotation, and its local node numbering is the same as the one in Fig. 4.6.

To assemble these three elements into a global stiffness matrix, a global nodal numbering is required. This can be seen in Fig. A.3(b). This way one can related each local node to a global node and thus realize the assembling of the global stiffness matrix.

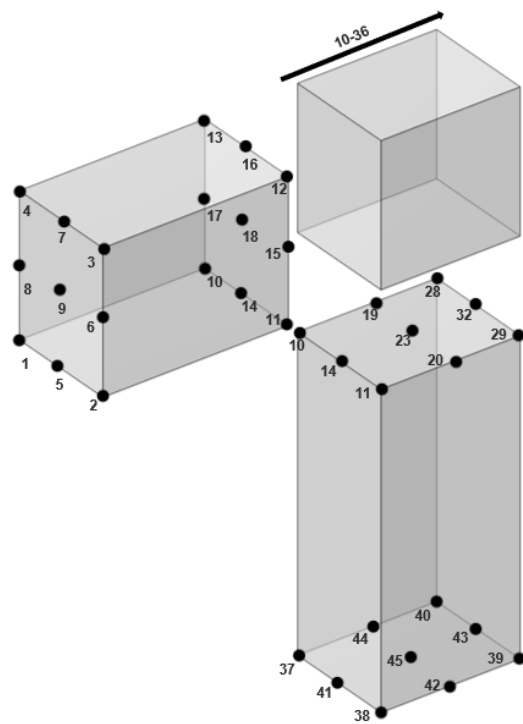
In MATLAB code this can be realized using the following approach:

```
1 %% Sample Matlab code
2 %Set-up nodal connectivity vectors
3 pointsYoke = [11 10 13 12 14 17 16 15 18 2 1 4 3 5 8 7 6 9];
4 pointsLeg = [10 11 29 28 14 20 32 19 23 37:45];
5
6 % Preallocate the K matrixes
7 K0 = zeros(45,45);
8 K1 = zeros(45,45);
9 K2 = zeros(45,45);
10
11 % Fill in the yoke matrix in the global matrix
12 K0(1:18,1:18) = K_yoke(pointsYoke,pointsYoke);
13
14 % Fill in the corner matrix in the global matrix
15 K1(10:36,10:36) = K_corner;
16
17 % Fill in the leg matrix in the global matrix
18 K2(pointsLeg,pointsLeg) = K_leg;
19
20 % Sum all contributions
21 K = K0 + K1 + K2;
```

Figure 27: Nodes numbering for model 2: local numbering (left) and global numbering (right).



(a) Local node numbering.



(b) Global node numbering.

**Figure A.3:** Nodes numbering for model 2: local numbering and global numbering.