UNIVERSIDAD PONTIFICIA
ICAI ICADE
COMILLAS
MADRID

ESCUELA TÉCNICA SUPERIOR DE INGENIERÍA (ICAI)

Numerical Simulation of Inductive Heating in Additively Manufactured Aircraft Engine Parts

Autor: Roberto González Celma
Director: Christian Zeller

Madrid
Junio 2017

Roberto
González
Celma

Numerical Simulation of Inductive Heating in Additively Manufactured Aircraft Engine Parts

PONTIFICIA UNIVERSITAS COMILLENSIS · MATRITI ·

AUTHORIZATION FOR DIGITALIZATION, STORAGE AND DISSEMINATION IN THE NETWORK OF END-OF-DEGREE PROJECTS, MASTER PROJECTS, DISSERTATIONS OR BACHILLERATO REPORTS

1. Declaration of authorship and accreditation thereof.

The author Mr. /Ms. Roberto González Celma

HEREBY DECLARES that he/she owns the intellectual property rights regarding the piece of work: Numerical Simulation of Inductive Heating in Additively Manufactured Aircraft Engine Parts3 that this is an original piece of work, and that he/she holds the status of author, in the sense granted by the Intellectual Property Law.

2. Subject matter and purpose of this assignment.

With the aim of disseminating the aforementioned piece of work as widely as possible using the University's Institutional Repository the author hereby **GRANTS** Comillas Pontifical University, on a royalty-free and non-exclusive basis, for the maximum legal term and with universal scope, the digitization, archiving, reproduction, distribution and public communication rights, including the right to make it electronically available, as described in the Intellectual Property Law. Transformation rights are assigned solely for the purposes described in a) of the following section.

3. Transfer and access terms

Without prejudice to the ownership of the work, which remains with its author, the transfer of rights covered by this license enables:

a) Transform it in order to adapt it to any technology suitable for sharing it online, as well as including metadata to register the piece of work and include "watermarks" or any other security or protection system.

b) Reproduce it in any digital medium in order to be included on an electronic database, including the right to reproduce and store the work on servers for the purposes of guaranteeing its security, maintaining it and preserving its format.

c) Communicate it, by default, by means of an institutional open archive, which has open and cost-free online access.

d) Any other way of access (restricted, embargoed, closed) shall be explicitly requested and requires that good cause be demonstrated.

e) Assign these pieces of work a Creative Commons license by default.

f) Assign these pieces of work a HANDLE (*persistent* URL). by default.

4. Copyright.

The author, as the owner of a piece of work, has the right to:

a) Have his/her name clearly identified by the University as the author

b) Communicate and publish the work in the version assigned and in other subsequent versions using any medium.

c) Request that the work be withdrawn from the repository for just cause.

d) Receive reliable communication of any claims third parties may make in relation to the work and, in particular, any claims relating to its intellectual property rights.

5. Duties of the author.

The author agrees to:

a) Guarantee that the commitment undertaken by means of this official document does not infringe any third party rights, regardless of whether they relate to industrial or intellectual property or any other type.

b) Guarantee that the content of the work does not infringe any third party honor, privacy or image rights.
c) Take responsibility for all claims and liability, including compensation for any damages, which may be brought against the University by third parties who believe that their rights and interests have been infringed by the assignment.
d) Take responsibility in the event that the institutions are found guilty of a rights infringement regarding the work subject to assignment.

6. Institutional Repository purposes and functioning.

The work shall be made available to the users so that they may use it in a fair and respectful way with regards to the copyright, according to the allowances given in the relevant legislation, and for study or research purposes, or any other legal use. With this aim in mind, the University undertakes the following duties and reserves the following powers:

a) The University shall inform the archive users of the permitted uses; however, it shall not guarantee or take any responsibility for any other subsequent ways the work may be used by users, which are non-compliant with the legislation in force. Any subsequent use, beyond private copying, shall require the source to be cited and authorship to be recognized, as well as the guarantee not to use it to gain commercial profit or carry out any derivative works.
b) The University shall not review the content of the works, which shall at all times fall under the exclusive responsibility of the author and it shall not be obligated to take part in lawsuits on behalf of the author in the event of any infringement of intellectual property rights deriving from storing and archiving the works. The author hereby waives any claim against the University due to any way the users may use the works that is not in keeping with the legislation in force.
c) The University shall adopt the necessary measures to safeguard the work in the future.
d) The University reserves the right to withdraw the work, after notifying the author, in sufficiently justified cases, or in the event of third party claims.

Madrid, on ..30..... of May, 2017......................., ...

HEREBY ACCEPTS

Signed...

Reasons for requesting the restricted, closed or embargoed access to the work in the Institution's Repository

I declare, under my responsibility, that the Project presented under the title

Numerical Simulation of induction heating in additively manufactured Aircraft Engine Parts

in the Technische Universität München (TUM) and in the ETS of Engineering - ICAI of the Pontificia Comillas University in the academic course 2016-2017 is my original and unpublished authorship and has not been presented prior to other effects. The Project is not plagiarism of another, either totally or partially and the information that has been taken of other documents is duly referenced.

Sign.: Roberto González Celma Date: 30 / 05 / 2017

Authorized delivery of the project supervisor

Christian Zeller

Sign.: M. Sc. Christian Zeller Date: 18 / 05 / 17

.

ESCUELA TÉCNICA SUPERIOR DE INGENIERÍA (ICAI)

Numerical Simulation of Inductive Heating in Additively Manufactured Aircraft Engine Parts

Autor: Roberto González Celma
Director: Christian Zeller

Madrid
Junio 2017

Roberto
González
Celma

Numerical Simulation of Inductive Heating in Additively Manufactured Aircraft Engine Parts

Resumen

Esta tesis se enmarca dentro del ámbito de los procesos de fabricación aditivos implementados por MTU Aero Engines AG en la producción de piezas de motor de aeronaves. En tales procesos, el material se proporciona en forma de polvo y se funde en los lugares elegidos mediante un láser. Con el fin de reducir el riesgo de grietas después del enfriamiento, se mantiene la pieza de trabajo durante el proceso de adición a una temperatura elevada constante mediante dispositivos de calentamiento por inducción. Para obtener la deseada temperatura uniformemente constante, se debe diseñar la geometría de las bobinas de calentamiento teniendo en cuenta la pieza a calentar. Esto se hace incluyendo el diseño de las bobinas en el proceso realimentado de fabricación CAD-CAE-CAM.

En la parte de CAE, se debe simular todo el proceso de calentamiento inductivo, de manera que es posible conocer los campos electromagnéticos creados y el calor suministrado a la pieza de trabajo. Con este propósito, la solución de una forma ligeramente simplificada (se descarta la corriente de desplazamiento) de las ecuaciones de Maxwell, ha sido implementada por el Dr. Guido Dhondt en un nuevo módulo electromagnético para el software de código abierto de Elementos Finitos **CalculiX**. Dicho módulo está todavía en desarrollo, por lo que la tesis que aquí se presenta busca servir de punto de referencia para la verificación de este nuevo módulo implementado.

Previamente a este trabajo, se realizaron algunos intentos de verificación de dicho módulo electromagnético de **CalculiX** sin éxito. Por lo tanto, se necesitaban más pruebas para lograr una verificación completa del módulo. La única manera de verificar un software de Elementos Finitos es comparar sus resultados con uno ya verificado o con soluciones analíticas. En esta tesis existe la ventaja de tener acceso al código fuente de **CalculiX** (es código abierto), y a la teoría electromagnética junto con la del método de elementos finitos (FEM) implementada. La verificación se ha dividido en diferentes pasos sucesivos, cuyas pruebas individualizadas a cada paso, muestran hasta qué punto el código funciona correctamente. Esto no podría haber sido hecho sin el conocimiento mencionado sobre el código fuente y la teoría detrás de él.

CalculiX es un paquete de software de código abierto para cálculos numéricos en 3D usando el FEM. Este software es multiplataforma y ha sido desarrollado por dos empleados de MTU Aero Engines AG, el Dr. Guido Dhondt y el Sr. Klaus Wittig. El

software se divide en dos paquetes bien diferenciados: **CalculiX CrunchiX** y **CalculiX GraphiX**.

Calulix CrunchiX (ccx) es la parte que realiza los cálculos, y **CalculiX GraphiX (cgx)** se encarga del preprocessing y del postprocessing. Indicar que **CalculiX** es compatible con otros paquetes de FEM, tal como ABAQUS. Esto permite utilizar **CalculiX CrunchiX** para realizar los cálculos y otro software para el resto de pasos. Por ejemplo, en varias simulaciones durante esta tesis, se ha utilizado Hypermesh en el proceso previo de mallado.

Para poder diseñar el proceso de verificación, ha habido que familiarizarse con la solución implementada de las ecuaciones de Maxwell dentro del marco de elementos finitos. El FEM aplicado a las ecuaciones de Maxwell proporciona una solución aproximada de ellas, bajo condiciones de contorno específicas. Las ecuaciones diferenciales parciales que deben resolverse pueden ser formuladas mediante campos vectoriales (magnético y electrostático) o a través de potenciales. Las formulaciones potenciales resultan ser más ventajosas cuando se hace uso de métodos de cálculo numérico tal como el FEM, ya que el conjunto de ecuaciones de Maxwell puede reducirse a una o dos ecuaciones diferenciales parciales utilizando potenciales.

El módulo electromagnético de **CalculiX** es capaz de realizar tres tipos de cálculos: Campos electro-estáticos, Campos magneto-estáticos y Campos de corrientes de Foucault. Los tres, en cierto modo, hacen uso de la implementación de la solución de las ecuaciones de Maxwell en el módulo. En el contexto de la presente tesis, son de interés los problemas de campos magneto-estáticos y los problemas de campos de corrientes de Foucault. El objetivo principal de la tesis es verificar el correcto funcionamiento del módulo de electromagnetismo de **CalculiX**. Este módulo será utilizado para realizar simulaciones que modelan el proceso de calentamiento por inducción. Por tanto, sólo se consideran de interés los problemas magnéticos.

El proceso de verificación se compone de tres partes, que corresponden a cómo **CalculiX** obtiene sus resultados. A grandes rasgos, las tres partes se basan en la siguiente ecuación extraída de la formulación P, que es parte de la formulación final implementada en **CalculiX**, la formulación A, V-A-P.

$$\vec{\mathbf{H}} = \vec{\mathbf{T}}_{\mathbf{0}} - \nabla P$$

Cada parte se prueba de forma independiente en un caso simple, de modo que es posible comparar los resultados obtenidos por **CalculiX** con una solución analítica o con resultados experimentales. Por lo tanto, el proceso de verificación se desglosa de la siguiente forma:

- $\vec{\mathbf{T}}_{\mathbf{0}}$
  - Ley de Biot-Savart: Cálculo de los campos magnéticos de acuerdo con la ley de Biot-Savart. Se utilizó una versión pre-desarrollada del código, para poder comparar la solución analítica de Biot-Savart con los valores de $\vec{\mathbf{T}}_{\mathbf{0}}$ obtenidos por **CalculiX**.
  - Mallado de la bobina: Identificación de un mallado estándar de la bobina para usarlo en las simulaciones posteriores. Se estudió los tipos de elementos y los tamaños de elementos (diámetro y longitud) para detectar los más apropiados.
  - Análisis de convergencia: realizado para analizar los factores con mayor influencia en la diferencia entre la solución analítica y la solución de elementos finitos.
- $\vec{\mathbf{H}}$
  - P, el potencial escalar magnético: Se utilizó la versión 2.11 de **CalculiX**. El objetivo era analizar si se calculó bien la corrección realizada por el gradiente del potencial escalar magnético a $\vec{\mathbf{T}}_{\mathbf{0}}$ para obtener el campo magnético en el dominio del aire.
  - Mallado del dominio del aire: Se detectó el mejor tipo de mallado para el dominio del aire.
  - Tamaño del dominio del aire: se probó qué influencia tenía el tamaño del dominio del aire.
  - Análisis de convergencia.
- Pieza de trabajo
  - Tesis del Dr. Kennedy del KTH (Instituto Real Sueco de Tecnología), “Magnetic Fields and Induced Power in the Induction Heating of Aluminum Billets”: la verificación se hizo en esta parte con los datos encontrados en dicha tesis.
  - Campos magnéticos, campos eléctricos inducidos y calor transferido: se analizaron los cálculos de los campos magnéticos dentro de la pieza de trabajo y en el entrehierro, de los campos eléctricos inducidos dentro de la pieza de trabajo que crean las corrientes de Foucault , y del calor transferido a la pieza.
  - Mallado de la pieza de trabajo: Se analizó cuál era la mejor malla para la pieza de trabajo.

- Mallado entre el dominio del aire y el dominio de la pieza de trabajo: se analizó el acoplamiento de las mallas entre dichos dominios, si resultaba mejor un mallado coincidente o un mallado libre.
- Análisis de diferentes casos con piezas de trabajo simplemente conexas y no simplemente conexas.
- Análisis de convergencia.

Las simulaciones realizadas para verificar **CalculiX** se realizan siguiendo el flujo de trabajo estándar en FEM: preprocessing, computation y postprocessing. En los gráficos siguientes, este flujo de trabajo se describe particularizado para esta tesis.

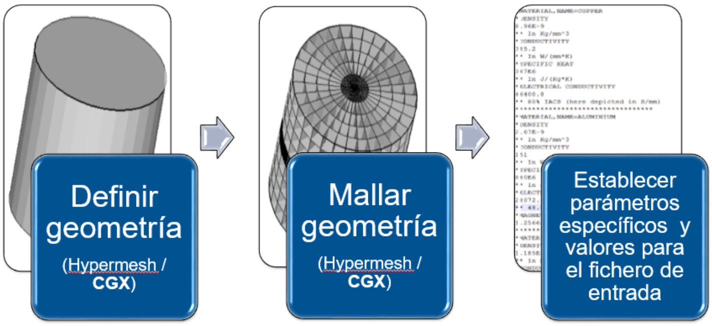

Definir geometría
(Hypermesh / CGX)
Mallar geometría
(Hypermesh / CGX)
Establecer parámetros específicos y valores para el fichero de entrada

Image 0.1: Preprocesado

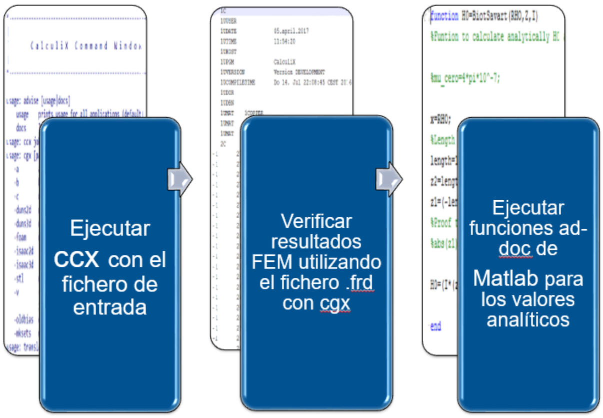

Ejecutar CCX con el fichero de entrada
Verificar resultados FEM utilizando el fichero .frd con cgx
Ejecutar funciones ad-doc de Matlab para los valores analíticos

Image 0.2: Computación y Postprocesado

Se realiza una repetición de estas tareas de simulación hasta que se alcanza la solución analítica, o la convergencia de los resultados (en caso de que no exista una solución analítica).

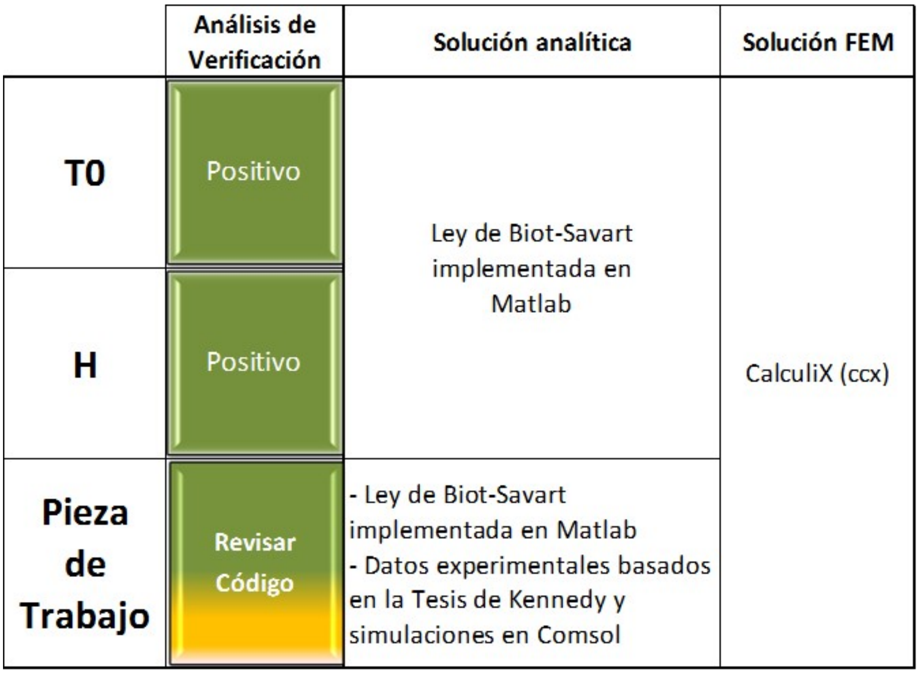
La tesis se ha llevado a cabo en las instalaciones de MTU Aero Engines AG, teniendo acceso a un entorno de desarrollo controlado para realizar todas las pruebas de verificación necesarias. Este entorno incluye el uso de los siguientes paquetes de software:

- Altair HyperMesh (parte del paquete de software Hyperworks): para propósitos de preprocessing.
- **CalculiX**: preprocessing, computation y postprocessing.
- Matlab: postprocessing con funciones y subrutinas ad-doc de Matlab para analizar los resultados.

A pesar de que **CalculiX** es un programa multiplataforma de código abierto, en MTU todo el trabajo con el mencionado software se realizó dentro de un entorno Linux (SUSE). Para realizar las simulaciones había dos posibilidades: localmente, ejecutándose en la workstation asignada, o remotamente, accediendo a los servidores de la empresa. Esta última opción fue de lejos la más utilizada para evitar problemas de rendimiento.

Además, se utilizaron los siguientes lenguajes de programación para diferentes tareas: C, FORTRAN (tanto para el análisis del código fuente de **CalculiX** como para su recompilación cuando fue necesario), lenguaje de programación de Matlab (para funciones y subrutinas ad-doc), y AWK (para cambiar eficientemente el tipo de elemento entre simulaciones a través del manejo de los ficheros de entrada).

Finalmente, las conclusiones del presente trabajo de verificación se resumen en el siguiente cuadro:



	Análisis de Verificación	Solución analítica	Solución FEM
T0	Positivo	Ley de Biot-Savart implementada en Matlab	CalculiX (ccx)
H	Positivo		
Pieza de Trabajo	Revisar Código	- Ley de Biot-Savart implementada en Matlab - Datos experimentales basados en la Tesis de Kennedy y simulaciones en Comsol	

Image 0.3: Tabla de conclusiones

INTENTIONALLY BLANK PAGE

Abstract

This thesis lies within the framework of the additive manufacturing processes deployed by MTU Aero Engines AG in the production of Aircraft engine parts. In such processes, the material is provided in powder form and melted in the chosen locations through a laser. In order to reduce the risk for cracks after cooling down, it has shown advantageous to keep the workpiece during the additive process at a constant high temperature by means of induction heating devices. To obtain that desired uniformly constant temperature, the geometry of the heating coils has to be designed considering the workpiece to be heated. This is done by including the design of the coils in the CAD-CAE-CAM feedback process.

In the CAE part, the entire inductive heating process has to be simulated, so that it is possible to know the electromagnetic fields and the heating rate supplied to the workpiece from the designed coil. On this purpose, the solution of a slightly simplified (displacement current is dismissed) form of the Maxwell´s equations, has been implemented by Dr. Guido Dhondt in a new electromagnetic module for the open source Finite Element software **CalculiX**. The module is still under development, so the thesis here presented seeks to serve as a benchmark for the successful verification of such new implemented module.

Previous to this work, some attempts were carried out to verify the electromagnetic module of **CalculiX** without success. Therefore, further testing is needed to achieve a complete verification of the module. The only way to verify a Finite Element software is to compare its results with one already verified or with analytic solutions. Here in this thesis exists the advantage of having access to the source code of **CalculiX** (easily obtained as it is open source software) and the electromagnetic, and Finite Element Method (FEM) theory behind the implementation of this code. The verification was split up in different successive steps whose individual testing shows until what point the code was working correctly. This could not have been done without the mentioned knowledge about the source code and the theory behind it.

CalculiX is an open source software package for numerical 3D calculations using the FEM. This software is multiplatform and it has been developed by two employees from

MTU Aero engines AG, Dr. Guido Dhondt and Mr. Klaus Wittig. The software is divided in two well differentiated packages: **CalculiX CrunchiX** and **CalculiX GraphiX**.

Calulix CrunchiX (ccx) is the solver that performs the calculation, so it encloses the computational steps. On the other hand, **CalculiX GraphiX (cgx)** is in charge of preprocessing and post processing. However, it must be added that **CalculiX** is compatible with other FEM software, such as ABAQUS. This allows to just make use of **CalculiX CrunchiX** for the calculations and perform the other steps with additional software. For instance, in several simulations during this thesis, Hypermesh was used as a preprocessor to perform the meshing.

In order to have designed the process of verification, first it must got acquainted with the solution of the implemented form of Maxwell's equations within the finite element framework. The FEM applied to Maxwell´s equation delivers an approximate solution of them, under prescribed boundary conditions. The partial differential equations to be solved can be formulated for field quantities (magnetic field, electrostatic field) or for potentials. The potential formulations turn out to be more advantageous when making use of numerical analysis such as the FEM, for the set of Maxwell's equations can be reduced into one or two partial differential equations by using potentials.

The electromagnetic module of **CalculiX** is able to perform three types of calculations: Static electric fields, Static magnetic fields and Eddy current fields. The three of them make use in some extent of the implementation of the solution of Maxwell´s equations in the module. Within the context of the present thesis, Static magnetic fields and Eddy current fields problems are targeted. The main goal of the thesis is to verify the correct operation of the electromagnetism module of **CalculiX**. This module will be used to perform simulations which model the process of inductive heating. Thus, only the magnetic related problems are considered.

Essentially, the verification process is composed of three parts, which correspond to how **CalculiX** obtain its results. Roughly explained, the three parts are based on the following equation extracted from the P-formulation, which is part of the final implemented formulation in **CalculiX**, the A, V-A-P formulation.

$$\vec{\mathbf{H}} = \vec{\mathbf{T}}_{\mathbf{0}} - \nabla P$$

Each part will be tested on his own in a simple case, so that it is possible to compare the results obtained from **CalculiX** with an analytic solution or experimental results. Therefore, the verification process is broken down as follows:

- $\vec{\mathbf{T}}_{\mathbf{0}}$

  - Biot-Savart's law: It was calculated using the law of Biot-Savart for magnetic fields. Here a pre-developed version of the code was used, to be able to compare the analytic solution of Biot-Savart with the $\vec{\mathbf{T}}_{\mathbf{0}}$ values delivered by **CalculiX**.

  - Meshing of the coil: in this part was aimed as well, to identify a standard meshing of the coil for the successive simulations. On account of that, it was studied which element type and which size (Diameter and length) were found to be more advantageous.

  - Convergence analysis: it was carried out to identify which factors had bigger influence in the difference between the analytic solution and the finite element solution.

- $\vec{\mathbf{H}}$

  - P, the magnetic scalar potential: Here the complete version 2.11 of **CalculiX** was used. The target was to analyzed if the correction performed by the gradient of the magnetic scalar potential to $\vec{\mathbf{T}}_{\mathbf{0}}$ to obtain the magnetic field in the air room was well calculated.

  - Meshing of air room: It was found out which was the most advantageous meshing for the air room.

  - Size of air room: it was tested which influence had the size of the air room.

  - Convergence analysis

- Workpiece

  - "Magnetic Fields and Induced Power in the Induction Heating of Aluminum Billets"; Dr. Kennedy's thesis from KTH (Swedish Royal Institute of Technology): the verification was done at this part against the data found in that thesis.

  - Magnetic fields, induced Electric fields and heating rate: The calculations of the magnetic field inside the workpiece and in the air-gap, of the induced electric fields inside the workpiece which creates the Eddy currents and the heating rate to the workpiece, were tested.

  - Meshing of work piece: It was found out which was the most advantageous meshing for the work piece.

  - Coupling between Air mesh and workpiece mesh: it was analyzed whether a coincidence meshing or a free meshing was mode advantageous.

- Analysis of different cases with simple and multiple connected workpieces.
- Convergence analysis

The simulations performed to verify **CalculiX** are made following the standard work-flow in FEM: preprocessing, computation and post processing. In the following graphs, such work-flow is depicted specific for this thesis.

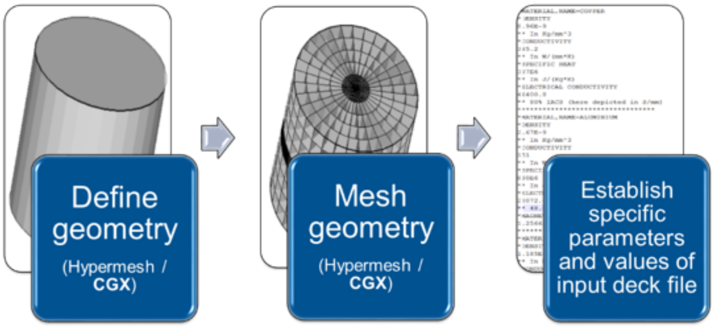


Image 0.1: Preprocessing

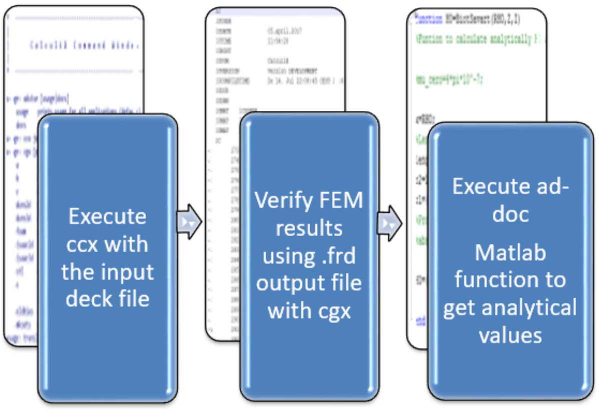

Execute ccx with the input deck file
Verify FEM results using .frd output file with cgx
Execute ad-doc
Matlab function to get analytical values

Image 0.2: Computation and Post-processing

A repetition of these simulation tasks are performed till the analytic solution, or the convergence of the results (in case there is no analytic solution) are reached.

The thesis has been carried out at MTU Aero Engines AG facilities, I have had access to a controlled development environment to perform all necessary verification tests. This environment included the use of the following software packages:

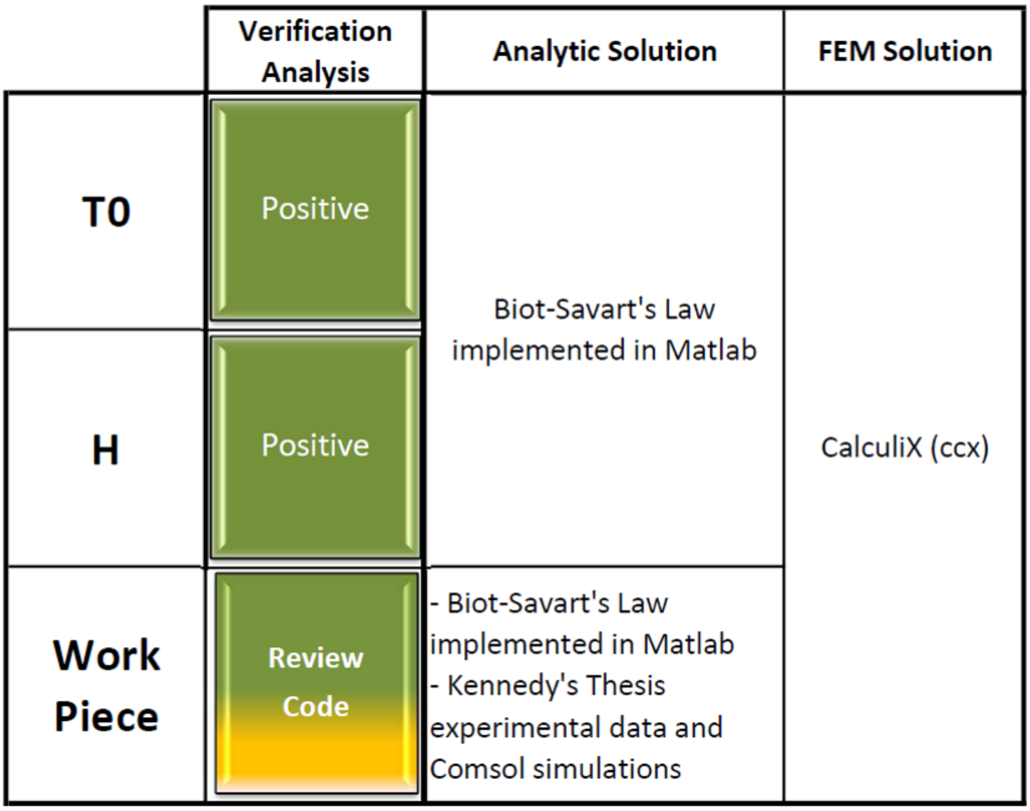
- Altair HyperMesh (part of the software package Hyperworks): for preprocessin g purposes.
- **CalculiX**: preprocessing, computation and postprocessing.

- Matlab: post processing purposes with ad-doc matlab functions and subroutines to analyze the results.

In spite of **CalculiX** is an open source multi-platform program, in MTU all work with the mentioned software was made within a Linux environment (SUSE Linux). To perform the simulations there were two possibilities: locally, executing in the assigned workstation, or remotely, accessing the servers of the company. This last option was by far the most used to avoid performance issues.

In addition to that, the following programming languages were used for different tasks: C, FORTRAN (both for analysis of the source code of **CalculiX** and its recompilation when needed), Matlab programming language (for ad-doc functions and subroutines) and AWK (to change efficiently the element type between simulations through handling of the input deck files).

Finally, the conclusions of the present benchmarking work are summarized in the following table:



	Verification Analysis	Analytic Solution	FEM Solution
T0	Positive	Biot-Savart's Law implemented in Matlab	CalculiX (ccx)
H	Positive		
Work Piece	Review Code	- Biot-Savart's Law implemented in Matlab - Kennedy's Thesis experimental data and Comsol simulations	

Image 0.3: Conclusions table

INTENTIONALLY BLANK PAGE

Numerical Simulation of inductive heating in additively manufactured Aircraft Engine Parts

Roberto González Celma

01.06.2017

INTENTIONALLY BLANK PAGE

Table of Contents

Abbreviations

Abbreviation	Description
2D	two-dimensional space
3D	three-dimensional space
CAE	Computer aided engineering
FE	Finite Element
FEM	Finite Element Method
cgx	**CalculiX** GraphiX
ccx	**CalculiX** CrunchiX
BVP	Boundary value problem
DC	Direct current
AC	Alternate current
IDE	Integrated development environment
SW	Software

Table 0.1: Abbreviation List

Nomenclature

Table with the Greek Alphabet (TIPLER 2010):

Capital letters	Lowercase	Name
Α	$alpha$	Alfa
Β	β	Beta
Γ	γ	Gamma
Δ	δ	Delta
Ε	ε	Épsilon
Ζ	ζ	Dseta
Η	η	Eta
$Theta$	θ	Zeta
Ι	ι	Iota
Κ	κ	Kappa
Λ	λ	Lambda
Μ	μ	Mi N v Ni
Ξ	ξ	Xi
Ο	ο	′Omicron
Π	π	Pi
Ρ	ρ	Ro
Σ	σ	Sigma
Τ	τ	Tau
Υ	υ	Ípsilon
Φ	ϕ	Fi
Χ	χ	Ji
Ψ	ψ	Psi
Ω	ω	Omega

Table 0.1: Greek Alphabet

Symbol	Unit	Description
$\vec{\mathbf{A}}$	$\frac{kg \cdot m}{A \cdot s_2}$	Magnetic Potential Vector
a_i		Generic area which contains the studied point
B	$T = \frac{kg}{A \cdot s^2}$	Magnetic Field
B	$T = \frac{kg}{A \cdot s^2}$	⌈ Magnetic Field ⌊ Magnetic flux density
C_i		Closed path or curve along a_i
D	$\frac{C}{m^2} = \frac{A \cdot s}{m^2}$	Electric displacement Field
$d\vec{\mathbf{S}}$		Differential surface normal vector
dV		Volume differential
E	$\frac{V}{m} = \frac{kg \cdot m}{A \cdot m^3}$	Electric Field
$\vec{\mathbf{F}}$		Generic Vector Function
$\vec{\mathbf{H}}$	$\frac{A}{m}$	Magnetic Field Intensity
I	A	Current
j	$\frac{A}{m^2}$	Current Density
P	A	Magnetic Scalar Potential
S_i	m^2	Generic Surface that encloses a generic volume V_i
$\vec{\mathbf{V}}$	$\frac{kg \cdot m^2}{A \cdot s^3}$	Electric Scalar Potencial
V_i	m^3	Generic Volume that includes spatially the studied point
ε	$\frac{F}{m} = \frac{A^2 \cdot s^4}{kg \cdot m^3}$	Permittivity
μ	$\frac{kg \cdot m}{A^2 \cdot s^2}$	Magnetic Permeability
φ		Generic potential scalar function
σ	$\frac{S}{m} = \frac{A^2 \cdot s^3}{kg \cdot m^3}$	Electrical Conductivity

*Table 0.2: Symbols and Units (based on (*DHONDT *2016b))*

INTENTIONALLY BLANK PAGE

1 Introduction

This document contains the necessary information to be able to successfully undertake the proposed objectives. On the one hand, it looks at the state of science and technology, describing in detail the essential theoretical concepts that must be known to understand the environment and the verification tests to be carried out. On the other hand, it includes a description of performed processes and tests, as well as the executed analyzes.

The working environment of this Master Thesis has been provided by MTU Aero Engines AG, at all levels:

- Development and testing environments (Linux Suse)
- Software Packages (Matlab, Hypermesh, **CalculiX**)
- Office package

The method of work applied has been of the type of analytical and comparative research, that is to say, the analysis criteria have been established, the events to be managed have been defined, the matrix analyzes have been established and applied, and finally, it has been found the corresponding conclusions.

The verification process has included a work-flow for simulations, allowing a working methodical behavior avoiding errors and the generation of data sets to obtain results and conclusions.

INTENTIONALLY BLANK PAGE

2 State of the science

2.1 Fundamentals of magnetic fields in matter

2.1.1 Magnetic classification of matter

The matter can be classified depending on how responds to an interaction with a magnetic field. There is a physical magnitude to measure the degree of non-permanent magnetization that a material obtains in response to a magnetic field (non-permanent because the magnetization disappears as soon as the applied magnetic field does; except for ferromagnetic materials, as it will be explained later on). It is called magnetic permeability (PURCELL et al. 1994):

$$\mu = \frac{\vec{\mathbf{B}}}{\vec{\mathbf{H}}} \tag{2.1}$$

To measure the degree of magnetization of a material, means to determine the capability of the material to support the formation of a magnetic field within it. In order to compare the magnetic behavior of materials, the magnetic permeability as it has been defined, it is known as “absolute ” magnetic permeability, and it can be determined as the product of the “relative ” magnetic permeability of the material, and the magnetic permeability in vacuum:

$$\mu = \mu_r \mu_0 \tag{2.2}$$

Then the materials can be classified according to its relative magnetic permeability (PURCELL et al. 1994).

Diamagnetic materials$(\mu_r < 1)$**:** they have a value of relative magnetic permeability smaller than one. Substances as water, copper, sodium chloride and quartz show a diamagnetic behavior, which means that in presence of a magnetic field, they tend to move where the intensity of the field decreases,i.e., they are repelled by the magnetic fields.

Diamagnetism happens to be a property of every atom and molecule. However, often the opposed behavior is observed. The reason why is that Diamagnetism is overtaken by another more intense effect, which conducts to attraction. Most of the anorganic compounds and practically all of the organic compounds are diamagnetic.

Paramagnetic materials$(\mu_r \sim 1)$**:** they have a value of relative magnetic permeability of approximately one. Paramagnetic are the substances which are attracted where the intensity of the field is higher. It should be remarked that in many cases, such as aluminum and sodium, the paramagnetic effect is not much higher than the common diamagnetic effect.

Ferromagnetic materials$(\mu_r \gg 1)$**:** they have a value of relative magnetic permeability much bigger than one. Ferromagnetic are the substances that behave like the iron or the magnetite. As paramagnetic materials, ferromagnetic materials are attracted where the intensity of the field is higher, but with much higher magnetic forces. Ferromagnetic substances can constitute permanent magnets or be attracted by a magnet with high forces. Of all described type of magnetism, ferro-magnetism is the strongest one, in the sense of creating the strongest magnetic forces.

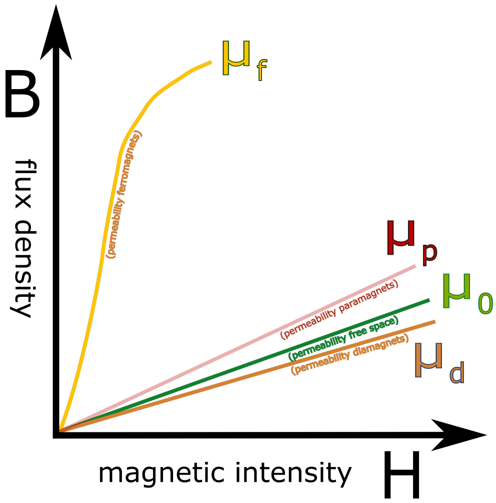

μ f
B
flux density
(permeability ferromagnets)
μ p
(permeability paramagnets)
μ 0
(permeability free space)
(permeability diamagnets)
μ d
magnetic intensity
H

*Image 2.1: Different types of magnetic permeability (image based on (*Zureks 2009*))*

To the classification of the different kinds of magnetism, it should be added an energetic perspective of each type. As it was mentioned, the interactions between ferromagnetic materials and magnetic fields are much more stronger than the ones occurring between paramagnetic or diamagnetic materials and magnetic fields. The reason is that at molecular energy scale, very little energy intervenes in paramagnetic or diamagnetic effects. This matches the reality, as the magnetic fields only have an appreciable influence in ferromagnetic materials. The diamagnetic and paramagnetic effects are difficult to appreciate, and should be measured in a laboratory with instruments sensitive enough.

In the interaction with matter, the magnetic field behaves different than the electric field. At a atomic scale, the atoms and molecules are compounded by particles electrically charged, that move at velocities generally smaller than the speed of light. According to Lorentz force:

$$\vec{\mathbf{F}} = q\vec{\mathbf{E}} + q\vec{\mathbf{v}} \times \vec{\mathbf{B}} \tag{2.3}$$

a magnetic field produces no force upon a stationary electric charge, and upon a mobile electric charge, the magnetic force is proportional to the $\vec{\mathbf{v}}$ of the charge. Therefore, the electric forces have the more influence at a atomic scale, much more than the magnetic forces.

2.1.2 Magnetic dipole moment

Ampere developed a hypothesis to explain the magnetization of matter. According to that theory, the matter from a magnetic point of view, can be modeled as a large group of tiny turns of electric current distributed in the whole substance. The current of each of this turns is able to generate a little magnetic field, which can be represented as the magnetic field generated by a magnetic dipole moment (PURCELL et al. 1994).

The so called magnetic dipole moment of a electric current from a tiny turn is defined as a vector as follows:

$$\vec{\mathbf{m}} = I\vec{\mathbf{a}} \tag{2.4}$$

Upon a magnetic dipole moment a magnetic force can act as consequence of an applied external magnetic field. The direction of the magnetic force will depend on the direction of the magnetic dipole moment and on the direction of the applied magnetic field. In a macroscopic context, the concept of magnetic dipole moment helps to explain the different magnetism types of the matter. For example, a paramagnetic behavior means that some magnetic dipoles moments are aligned in parallel with an external magnetic field, so that the substance will be attracted where this magnetic field is more intense. On the other hand, if the magnetic dipole moments are aligned to the opposite way, the substance will show a diamagnetic behavior, being repelled where the magnetic field is less intense.

Therefore, the orientation of the numerous magnetic dipole moments inside the matter is the responsible of its degree of magnetization. An affirmation that matches Ampere's theory for the magnetization of matter. The orientation of the magnetic dipole moments can be changed for instance with the application of an external magnetic field (process of magnetization). This change in orientation lasts as long as the external magnetic field is applied, except for ferromagnetic materials, which due to its hysteresis cycle of magnetization, its magnetization lasts after the external magnetic field is removed.

2.1.3 Magnetic susceptibility and magnetization

The magnetic susceptibility measures the degree of magnetization of a material in response to a magnetic field. Essentially, it refers to the same concept as the magnetic permeability, only that the magnetic permeability expresses the total magnetization of material and volume.

Before defining mathematically the magnetic susceptibility, it should be introduced the magnetic moment per volume unit, also known as magnetization: $\vec{\mathbf{M}}$; the vector magnetization is calculated as the product of the number of dipoles oriented in a direction per unit volume and the magnetic moment of each of these dipoles. Now the magnetic susceptibility of volume can be introduced as an dimensionless number in the following formula:

$$\vec{\mathbf{M}} = \chi_m \frac{\vec{\mathbf{B}}}{\mu_0} \tag{2.5}$$

A negative value of the magnetic susceptibility of a material points out a diamagnetic behavior, hence a magnetic field that crosses the material will be weakened. On the contrary, a positive value of magnetic susceptibility can mean a paramagnetic behavior, and a magnetic field that crosses the material will be strengthened. Again, this weakening or strengthening of magnetic field across the matter can be explained by the orientation of the magnetic dipoles moments inside the matter. If the dipoles of the matter in its majority are aligned in opposite direction with the external magnetic field, the response is diamagnetic, whereas if the dipoles are aligned with the field, the response is paramagnetic, and the magnetic field is strengthened by the effect of a favorable orientation of the dipoles (matter magnetizes in the same direction of the magnetic field).

However, it should be remarked that for diamagnetic and paramagnetic materials, the magnetic susceptibility has a very small value compared to the unity. Which means that the magnetic field originated by the magnetic dipoles moments inside the matter (as an average of the fields produced by all dipoles), from a macroscopic point of view, is much weaker than an external applied magnetic field. Therefore, in diamagnetic and paramagnetic materials, the applied external magnetic field which acts to orientate the dipoles, can be considered as the same magnetic field that would act in the same space in absence of the material. This is not true for ferromagnetic materials.

The ferromagnetic materials have a positive magnetic susceptibility, and maintain its magnetized state even when the applied magnetic field is removed. The ferromagnetic materials can show states of permanent magnetization, which are known as magnets. It is possible to create artificial magnets magnetizing a ferromagnetic material, and it is possible to demagnetize them applying a enough strong magnetic field opposed to the original direction of magnetization. The cycle of magnetic hysteresis should be take into account to success in those actions.

2.1.4 Vector intensity of magnetic field

The existance of the vector $\vec{\mathbf{H}}$, the intensity of magnetic field, allow to differentiate between free currents and bound currents. The free currents are those which normally are simply known as currents, the ordinary currents of electric conduction, that drive along macroscopic paths, and can be measured with an Ammeter. Then again, the bound currents are related to the magnetic moments at a atomic scale. These are the tiny molecular current turns predicted by Ampere, which are responsible of the magnetization of matter described in last sections. The bound currents can be defined as the curl of the magnetization vector:

$$\vec{\mathbf{J}_{\mathbf{bound}}} = \nabla \times \vec{\mathbf{M}} \tag{2.6}$$

In absence of free currents, $\vec{\mathbf{B}}$ inside the matter as a spatial average, is related to the bound currents in the same way as it is with any other current density:

$$\nabla \times \vec{\mathbf{B}} = \mu_0 \vec{\mathbf{J}_{\mathbf{bound}}} \tag{2.7}$$

If there are free currents acting too, the $\vec{\mathbf{B}}$ produced by these free currents should be added to the magnetic field originated by the magnetized matter:

$$\nabla \times \vec{\mathbf{B}} = \mu_0 (\vec{\mathbf{J}_{\mathbf{free}}} + \vec{\mathbf{J}_{\mathbf{bound}}}) = \mu_0 \vec{\mathbf{J}_{\mathbf{total}}} \tag{2.8}$$

By equation (2.6), the bound current can be expressed as a function of the magnetization vector:

$$\nabla \times \vec{\mathbf{B}} = \mu_0 \vec{\mathbf{J_{free}}} + \mu_0(\nabla \times \vec{\mathbf{M}}) \tag{2.9}$$

$$\nabla \times (\vec{\mathbf{B}} - \mu_0 \vec{\mathbf{M}}) = \mu_0 \vec{\mathbf{J_{free}}} \tag{2.10}$$

Now the vector $\vec{\mathbf{H}}$ can be defined as follows:

$$\mu_0 \vec{\mathbf{H}} = \vec{\mathbf{B}} - \mu_0 \vec{\mathbf{M}} \qquad or \qquad \vec{\mathbf{H}} = \frac{\vec{\mathbf{B}}}{\mu_0} - \vec{\mathbf{M}} \tag{2.11}$$

Then an alternative form of the second Maxwell equation (2.26) is obtained:

$$\nabla \times \vec{\mathbf{H}} = \vec{\mathbf{J_{free}}} \tag{2.12}$$

The vector $\vec{\mathbf{H}}$ is related with the free current similarly as the vector $\vec{\mathbf{B}}$ is with the total current, the bound plus the free current. However, although the divergence of $\vec{\mathbf{B}}$ is always null, the divergence of $\vec{\mathbf{H}}$ is it not necessarily.

Historically the name of magnetic field was booked for $\vec{\mathbf{H}}$. Posterior discoveries have transferred the name to the real magnetic fundamental field, the vector $\vec{\mathbf{B}}$. Even so, the use of $\vec{\mathbf{H}}$ has been maintained as it results advantageous when facing certain kinds of electromagnetic problems. The vector $\vec{\mathbf{H}}$ happens to be useful because in magnetic systems, the magnitude which is easy to measure is the free current. The free current can easily be driven along a wire and measured by an ammeter. As a matter of fact, it is much more difficult to calculate and control the phenomenon of magnetization (vector $\vec{\mathbf{M}}$) inside the materials, and consequently the vector $\vec{\mathbf{B}}$, than to obtain the vector $\vec{\mathbf{H}}$ from the measured free current. In fact, the Ampere's law can be written alternatively with the $\vec{\mathbf{H}}$ vector:

$$\int_C \vec{\mathbf{H}} \cdot d\vec{\mathbf{l}} = \int_S \vec{\mathbf{J_{free}}} \cdot d\vec{\mathbf{a}} = I_{free} \tag{2.13}$$

The relation between $\vec{\mathbf{H}}$ and $\vec{\mathbf{B}}$ is the one indicated by equation (2.1.4). In addition to that, if $\vec{\mathbf{M}}$ is proportional to $\vec{\mathbf{B}}$, then is proportional to $\vec{\mathbf{H}}$ too, and the following equation applies:

$$\vec{\mathbf{M}} = \chi_m \vec{\mathbf{H}} \tag{2.14}$$

For a material that fulfills the last equation, it can be derived:

$$\vec{\mathbf{B}} = \mu_0(\vec{\mathbf{H}} + \vec{\mathbf{M}}) = \mu_0(1 + \chi_m)\vec{\mathbf{H}} = \mu_0\mu_r\vec{\mathbf{H}} = \mu\vec{\mathbf{H}} \tag{2.15}$$

The equation above confirms that if $\vec{\mathbf{M}}$ is proportional to $\vec{\mathbf{H}}$, $\vec{\mathbf{B}}$ is proportional to $\vec{\mathbf{H}}$ as well. The factor of proportionality $1 + \chi_m$ has been mentioned before, and it is known as relative magnetic permeability. Again, this does not apply for ferromagnetic materials.

2.1.5 Potential magnetic vector

The introduction of the potential magnetic vector $\vec{\mathbf{A}}$ can be better understood making a certain kind of analogy with the electric field. The electrostatic field can be expressed as the gradient of a scalar potential. The reason why is that the curl of $\vec{\mathbf{E}}$ is null in every point of the space. The last affirmations satisfy the vectorial identity (A.11). In case of the magneto-static field, as states in the second Maxwell equation (2.26), the curl of $\vec{\mathbf{B}}$ is not always null, so it can not be expressed as the gradient of a scalar potential. The magnetic field instead, can be expressed as the curl of a vectorial function. That can be derived from the vectorial identity (A.10). According to this identity, the divergence of the curl of a vectorial function is equal to null, which is the case of the magnetic field, its divergence is always null by the fourth Maxwell equation (2.28). So again, that allows to express the magnetic field in the following way:

$$\left.\begin{aligned} \nabla \cdot \vec{\mathbf{B}} = 0 \\ \nabla \cdot (\nabla \times \vec{\mathbf{A}}) = 0 \end{aligned}\right\} \qquad \vec{\mathbf{B}} = \nabla \times \vec{\mathbf{A}} \tag{2.16}$$

Therefore, due to analogy with the electric field, the vector $\vec{\mathbf{A}}$ is known as potential magnetic vector. In addition to that, it is important to remark that the last equation (2.16), because of the vectorial identity (A.10), as mentioned, enforces the automatic satisfaction of the fourth Maxwell equation (2.28), which in turn means the satisfaction of the magnetic Gauss's law.

In order to calculate $\vec{\mathbf{A}}$, it should be looked at the relation with the current density $\vec{\mathbf{J}}$:

$$\nabla \times (\nabla \times \vec{\mathbf{A}}) = \mu_0 \vec{\mathbf{J}} \tag{2.17}$$

This equation is a vectorial relation, thus there are three equations to solve, one for each of the three components of the vector. To understand how $\vec{\mathbf{A}}$ is calculated, it turns up useful to develop one of the equations corresponding to one component of the vector, for instance, the one corresponding to component **x** in Cartesian coordinates:

$$\overbrace{\frac{\partial B_z}{\partial y} - \frac{\partial B_y}{\partial z}}^{\mathbf{x}\text{ component of }\nabla\times\vec{\mathbf{B}}} = \frac{\partial}{\partial y}\underbrace{(\frac{\partial A_y}{\partial x} - \frac{\partial A_x}{\partial y})}_{B_z} - \frac{\partial}{\partial z}\underbrace{(\frac{\partial A_x}{\partial z} - \frac{\partial A_z}{\partial x})}_{B_y} = \mu_0 J_x \tag{2.18}$$

Operating the equation above, it can be obtained:

$$\underbrace{-\frac{\partial^2 A_x}{\partial x^2} - \frac{\partial^2 A_x}{\partial y^2} - \frac{\partial^2 A_x}{\partial z^2}}_{-\nabla^2 A_x} + \underbrace{\frac{\partial}{\partial x}(\frac{\partial A_x}{\partial x} + \frac{\partial A_y}{\partial y} + \frac{\partial A_z}{\partial z})}_{\nabla\cdot\vec{\mathbf{A}}} = \mu_0 J_x \tag{2.19}$$

In last equation the negative laplacian of A_x and the divergence of $\vec{\mathbf{A}}$ can be identified.

The magnetic vector potential is not unique, different vectorial functions can satisfy the vectorial identity (A.10) and equation (2.17). For instance, the following vector would be a possible magnetic potential vector, because its $\vec{\nabla}\varphi$ part satisfies the vectorial identity (A.11) as well; this part vanishes when satisfying equation (2.17)):

$$\vec{\mathbf{A}}' = \vec{\mathbf{A}} + \vec{\nabla}\varphi \tag{2.20}$$

To assure the uniqueness of the potential magnetic vector, usually it is of interest to have the $\nabla \cdot \vec{\mathbf{A}}$ defined as null, according to the so called "Coulomb gauge":

$$\nabla \cdot \vec{\mathbf{A}} = 0 \tag{2.21}$$

Then the equation ends up as follows:

$$\frac{\partial^2 A_x}{\partial x^2} + \frac{\partial^2 A_x}{\partial y^2} + \frac{\partial^2 A_x}{\partial z^2} = -\mu_0 J_x \tag{2.22}$$

This last has the form of a Poisson equation, hence it can be solved at the following way:

$$A_x(x_a, y_a, z_a) = \frac{\mu_0}{4\pi}\int \frac{J_x(x_b, y_b, z_b)}{\vec{\mathbf{r_{ab}}}} dv \tag{2.23}$$

$$\vec{\mathbf{A}}(x_a, y_a, z_a) = \frac{\mu_0}{4\pi} \int \frac{\vec{\mathbf{J}}(x_b, y_b, z_b)}{\vec{\mathbf{r_{ab}}}} dv \tag{2.24}$$

2.2 Maxwell's equations

Between 1861 and 1862, James Clerk Maxwell publicized its article "On Physical Lines of Force " (MAXWELL 1862). This article establishes the first historical attempt to develop mathematically a unified theory of the electromagnetism and the light. Maxwell makes use of vectorial calculus in this article to constitute the equations that govern the electromagnetic fields, putting together the laws and theorems enunciated before by Coulomb, Gauss, Ampere and Faraday.

The definition of the electromagnetic fields by Maxwell's equations is essentially complete, and it explains the wavelike nature of the visible light, as an electromagnetic wave which propagates at the speed of light.

Originally, the works of Maxwell produced twenty equations and twenty variables to explain the whole electromagnetism theory. The current form of four equations known academically as the Maxwell's equations, is the work of Oliver Heaviside.

2.2.1 Differential form of Maxwell's equations

The equations can be represented in multiple ways depending on the magnitudes and variables used. Each form of the equations results advantageous according to the electromagnetic problem that is to be solved. Below one of the differential forms is presented:

$$\nabla \times \vec{\mathbf{E}} = -\frac{\partial \vec{\mathbf{B}}}{\partial t} \tag{2.25}$$

$$\nabla \times \vec{\mathbf{B}} = \mu_0 \vec{\mathbf{J}} + \mu_0 \varepsilon_0 \frac{\partial \vec{\mathbf{E}}}{\partial t} \tag{2.26}$$

$$\nabla \cdot \vec{\mathbf{E}} = \frac{\rho}{\varepsilon_0} \tag{2.27}$$

$$\nabla \cdot \vec{\mathbf{B}} = 0 \tag{2.28}$$

This form of Maxwell's equations is written for the electromagnetic fields in vacuum, whose sources are an electrical density charge ρ, and a current density $\vec{\mathbf{J}}$.

Faraday's law The first equation (2.25) is the differential form of Faraday's law. It affirms that a time-varying magnetic field through an electric conductor induces an electric field in it.

Ampere's law The second equation (2.26) includes the Ampere's law in differential form. The first term of the right-hand side of the equation contains the current density, generated by electrons moving inside an electrical conducting material. The second term contains the so called displacement current in dielectric media, which is generated by a time-varying electric field. This last term was introduced by Maxwell in its mathematical formulation. When studying the electromagnetic theory of light, the displacement current term (time-varying electric field) as a counterpart of the magnetic term (time-varying magnetic field) in first equation, implies the existence of electromagnetic waves. However, that happens to not be the purpose of this thesis, and a simplification of this second equation can be made dismissing the displacement current term.

Electric Gauss's law The third equation (2.27) is the electric Gauss's law, which is equivalent to Coulomb's law for electric charges. This law states that the sources of static electric fields are electric charges, and its electric flux lines start and close upon the charge.

Magnetic Gauss's law The fourth equation (2.28) means that the magnetic field is divergence free, which in turn means, that there are not other sources of the magnetic field with exception of the currents, the free magnetic charges do not exist. Therefore, magnetic flux lines close upon themselves.

2.3 Formulation of electromagnetic problems

$$\nabla \times \vec{\mathbf{E}} = -\frac{\partial \vec{\mathbf{B}}}{\partial t} \tag{2.29}$$

$$\nabla \times \vec{\mathbf{H}} = \vec{\mathbf{J}} + \frac{\partial \vec{\mathbf{D}}}{\partial t} \tag{2.30}$$

$$\nabla \cdot \vec{\mathbf{D}} = \rho \tag{2.31}$$

$$\nabla \cdot \vec{\mathbf{B}} = 0 \tag{2.32}$$

2.3.1 Constitutive relations

The constitutive equations are specific to the material used, they are not universal physical laws, they describe the relations between field magnitudes depending on the particular physical properties of the used material. These relations can be linear or nonlinear. The following are the general form of the constitutive equations which emerge in the electromagnetic problems that are about to be discussed:

$$\vec{\mathbf{B}} = \mu_0(\vec{\mathbf{H}} + \vec{\mathbf{M}}) \tag{2.33}$$

$$\vec{\mathbf{J}} = \sigma(\vec{\mathbf{E}} + \vec{\mathbf{E_i}}) \tag{2.34}$$

$$\vec{\mathbf{D}} = \varepsilon_0\vec{\mathbf{E}} + \vec{\mathbf{P}} \tag{2.35}$$

2.3.1.1 Linear Constitutive relations

The linear case is a simplified form of the equations above, where some field magnitudes are proportional to others. The simplification made is showed in the following formulas:

$$\vec{\mathbf{M}} = \chi_m\vec{\mathbf{H}} \to \vec{\mathbf{B}} = \mu_0\mu_r\vec{\mathbf{H}} = \mu\vec{\mathbf{H}} \tag{2.36}$$

$$\vec{\mathbf{E_i}} = 0 \to \vec{\mathbf{J}} = \sigma\vec{\mathbf{E}} \tag{2.37}$$

$$\vec{\mathbf{P}} = \varepsilon_0\chi_d\vec{\mathbf{E}} \to \vec{\mathbf{D}} = \varepsilon\vec{\mathbf{E}} \tag{2.38}$$

χ_m is the magnetic susceptibility and χ_d the electric susceptibility. The magnetic susceptibility has already been defined, but here there is a reminded of its relation with the absolute magnetic permeability μ and the relative magnetic permeability μ_r.

$$\mu = \mu_0(1 + \chi_m) = \mu_0\mu_r \to 1 + \chi_m = \mu_r \tag{2.39}$$

The electric case is analogous:

$$\varepsilon = \varepsilon_0(1 + \chi_d) = \varepsilon_0\varepsilon_r \to 1 + \chi_d = \varepsilon_r \tag{2.40}$$

ε_r is the relative permittivity. Regarding the conductivity σ, it is constant.

2.3.1.2 Nonlinear Constitutive relations

Nevertheless, constitutive relations can be nonlinear. In that case, the parameters described (permeability, conductivity and permittivity) depend on the field magnitudes:

$$\mu = \mu(\vec{\mathbf{H}}, \vec{\mathbf{B}}) \tag{2.41}$$

$$\sigma = \sigma(\vec{\mathbf{E}}, \vec{\mathbf{J}}) \tag{2.42}$$

$$\varepsilon = \varepsilon(\vec{\mathbf{E}}, \vec{\mathbf{D}}) \tag{2.43}$$

2.3.1.3 Homogeneity of material properties

- **Homogeneous material**: so are called the materials whose physical properties are independent of the position in space.
- **Inhomogeneous material**: the properties of these materials depend on the position in space:

  $$\mu = \mu(x, y, z) \qquad \sigma = \sigma(x, y, z) \qquad \varepsilon = \varepsilon(x, y, z) \tag{2.44}$$

  It is possible that the parameters of the constitutive relations depend on the frequency of excitation too:

  $$\mu = \mu(f) \qquad \sigma = \sigma(f) \qquad \varepsilon = \varepsilon(f) \tag{2.45}$$

2.3.1.4 Isotropy of materials

- **Isotropic material**: so are called the materials whose constitutive parameters are independent of the direction of the applied field.
- **Anisotropic material**: constitutive parameters of the material depend on the direction of the applied field. In this case the parameters result to be tensors:

$$\vec{\mathbf{B}} = [\mu]\vec{\mathbf{H}} \tag{2.46}$$

$$\vec{\mathbf{J}} = [\sigma]\vec{\mathbf{E}} \tag{2.47}$$

$$\vec{\mathbf{D}} = [\varepsilon]\vec{\mathbf{E}} \tag{2.48}$$

The most general case would be a situation in which the constitutive relations depend on all the variables described: other magnitude fields, position in space, frequency of the source of excitation and direction of the applied field. For instance, for the magnetic field:

$$\vec{\mathbf{B}} = \vec{\mathbf{B}}\{\vec{\mathbf{H}}, (x, y, z), f\} \tag{2.49}$$

2.3.2 Magnetic case

CalculiX only considers calculations for paramagnetic and diamagnetic materials.

2.4 Classification of Maxwell's equations

Considering a classification of Maxwell's equations, there are five possible electromagnetic problems to be solved. The formulation of the field equations needed in each case is now presented.

Static Magnetic Field As a static case, the time variation is neglected. The time independent current density J_0 generates a time independent magnetic field intensity $\vec{\mathbf{H}}$ and accordingly, a time independent magnetic flux density $\vec{\mathbf{B}}$. The formulation of the equations is as follows:

$$\nabla \times \vec{\mathbf{H}} = \vec{\mathbf{J_0}} \tag{2.50}$$

$$\nabla \cdot \vec{\mathbf{B}} = 0 \tag{2.51}$$

$$\vec{\mathbf{B}} = \begin{cases} \mu_0 \vec{\mathbf{H}}, & \text{in air} \\ \mu_0 \mu_r \vec{\mathbf{H}}, & \text{in magnetically linear material} \\ \mu_0 (\vec{\mathbf{H}} + \vec{\mathbf{M}}), & \text{in magnetically nonlinear material} \end{cases} \tag{2.52}$$

Static Electric Field Here the time variation is neglected too. The electric stationary charges are the sources of the static electric field. They generate an electric intensity field $\vec{\mathbf{E}}$ and accordingly, a flux electric density $\vec{\mathbf{D}}$. These field magnitudes are related in the following way:

$$\nabla \times \vec{\mathbf{E}} = 0 \tag{2.53}$$

$$\nabla \cdot \vec{\mathbf{D}} = \rho \tag{2.54}$$

$$\vec{\mathbf{D}} = \begin{cases} \varepsilon \vec{\mathbf{E}}, & \text{in electrically linear material} \\ \varepsilon_0 \vec{\mathbf{E}} + \vec{\mathbf{P}}, & \text{in electrically nonlinear material} \end{cases} \tag{2.55}$$

$\vec{\mathbf{P}}$ is defined as the polarization vector.

Currents in Conducting Materials A electric current consists of electrons moving along a conducting material. These are the currents described as "free currents " in the last section. The equations needed to describe this phenomenon are time independent:

$$\nabla \times \vec{\mathbf{E}} = 0 \tag{2.56}$$

$$\nabla \cdot \vec{\mathbf{J}} = 0 \tag{2.57}$$

$$\vec{\mathbf{J}} = \begin{cases} \sigma \vec{\mathbf{E}}, & \text{in electrically linear material} \\ \sigma(\vec{\mathbf{E}} + \vec{\mathbf{E_i}}), & \text{in electrically nonlinear material} \end{cases} \tag{2.58}$$

Eddy current field known as "Foucault current field " as well (Eddy Currents in the English bibliography), it is a time varying case. Consequently, electric and magnetic fields are coupled in time. However, in the normal frequency range of use of the coils, the term introduced by Maxwell, known as "displacement current " can be neglected:

$$|\vec{\mathbf{J}}| \gg |\frac{\partial \vec{\mathbf{D}}}{\partial t}| \tag{2.59}$$

Then the problem can be described as "quasi-static ". This concept is used in thermodynamics and it means that the temporal evolution of the magnitudes of the problem does not have large variations.

$$\nabla \times \vec{\mathbf{H}} = \vec{\mathbf{J}} \tag{2.60}$$

$$\nabla \times \vec{\mathbf{E}} = -\frac{\partial \vec{\mathbf{B}}}{\partial t} \tag{2.61}$$

$$\nabla \cdot \vec{\mathbf{B}} = 0 \tag{2.62}$$

$$\vec{\mathbf{B}} = \begin{cases} \mu_0 \vec{\mathbf{H}}, & \text{in air} \\ \mu_0 \mu_r \vec{\mathbf{H}}, & \text{in magnetically linear material} \\ \mu_0 (\vec{\mathbf{H}} + \vec{\mathbf{M}}), & \text{in magnetically nonlinear material} \end{cases} \tag{2.63}$$

$$\vec{\mathbf{J}} = \sigma \vec{\mathbf{E}} \tag{2.64}$$

Wave propagation With this aim Maxwell developed his theory of electromagnetism. The governing equations of this problem are the four Maxwell's equations. In electric engineering applications, the complete set of Maxwell's equations is useful when analyzing waveguides, cavities or resonators at high frequencies.

In the current thesis, only the static magnetic problems and the Eddy current field problems are analyzed in depth.

2.4.1 Magneto statics and Eddy current field problems

Among the described electromagnetic problems in last section, the electromagnetism module of **ccx** (**CalculiX** CrunchiX SW) is able to perform calculations of three of them:

- Static electric fields
- Static magnetic fields
- Eddy current fields

The three of them make use in some extent of the implementation of the solution of Maxwell's equations in the electromagnetism module of **ccx**. As it could be noticed in the description of the field equations in last section, not all electromagnetic problems require from the whole set of Maxwell's equations to be solved. Within the context of the present thesis, static magnetic fields and Eddy current fields problems are targeted. The main goal of the thesis is to verify the correct operation of the the electromagnetism module of **ccx**. This module will be used to perform simulations which model the process of inductive heating. Thus, only the magnetic related problems are considered.

In this section the two problems of interest will be formulated more in detail, adding the interface and boundary conditions to the field equations; hence, describing in this way the whole set of governing equations of each problem.

Firstly, the physical environment of the electromagnetic problems should be introduced. How is this environment is specially important when considering the mathematical domains in which the equations are applied, and the boundary and interface conditions between such domains as well. The environment of the problems is presented here as it is described in the user manual of **ccx** (DHONDT 2016b), under the theory chapter belonging to the electromagnetism module.

The environment described in the manual offers a general framework, aiming to fit within as many specific electromagnetic problems as possible. The environment of these problems is similar, it consists of a delimited portion of space, filled with an electrically nonconducting material such as air, where a coil, fed by an electric current, generates a magnetic field. Besides, inside this space, a material, considered the Work Piece in an experiment, is reached by the magnetic field. The material is usually considered as electrically conductive. The nature of the magnetic field and how the material respond to the interaction with it (in the sense of creating Eddy currents or not), depend on the type of current applied to the coil. Direct current (DC) will generate a magneto static field, whereas alternate current (AC) will produce a time varying magnetic field, which according to Faraday's law, it generates Eddy currents when going through an electrical conductor. This differentiation will lead to two different electromagnetic problems. In both cases, the currents that feed the coil, correspond to the previously described "free currents ".

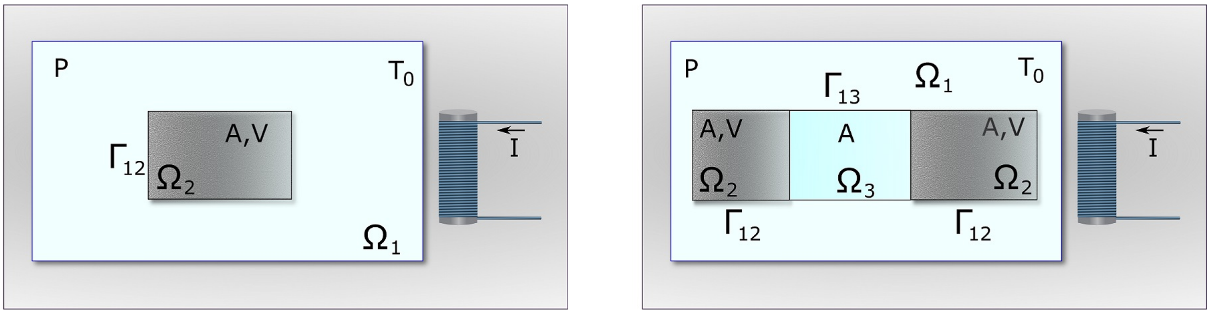
From the description above of the problems' environment, the existence of at least two different regions can be inferred. The regions differentiate each other because of the material present in each one. The materials in turn behave differently before the influence of the magnetic field. Consequently, the magnetic field have to be calculated with different equations in each region, i.e., with different formulations within the context of FEM. The formulations are bonded through interface conditions which are located physically in the the interface between the two regions. Mathematically speaking, each region is designated as a domain. In this thesis the domains are named following the documentation of **ccx** (DHONDT 2016b, P. 244):

- **Domain 1, Air**: The domain filled by a nonconducting material is known as Ω_1. The material chosen in the development of the examples present in this thesis is air.
- **Domain 2, Working Piece**:

- Magneto static problem: it is filled by a magnetic material.
- Eddy current problem: it is filled by an electrically conducting material.

In both cases the domain is known as Ω_2.

The exposed description is now depicted graphically in the group of images 2.2. Likewise, the differentiation between simple connected and multiple connected bodies is made, which it will be explained later on.



(a) Simple connected

(b) Multiple connected

*Image 2.2: Domains of the electromagnetic problem; (images based on (*DHONDT *2016b))*

In the group of images 2.2, it can be noticed that the fed coil is located out of the boxes that represent the domains. The reason why is that the coil is considered separated from the studied environment. The coil excites the system but receives no feedback from it. This can be accomplished in reality by a regulating system, which counteracts any existing disturbance that modifies the current set by the user(DHONDT 2016b, P. 244). However, in the environment of **ccx**, that is achieved through the very code of the program. The code considers that the current along the coil is only set by the user, equations considering a feedback from the system to the current of the coil are not implemented in **ccx**.

2.4.2 Static magnetic fields (DC current in the coil)

The current feeding the coil is direct current (DC). Hence, the time variation of the field magnitudes can be neglected, and the current can be described by a time independent current density $\vec{\mathbf{J}}_0$. The magnetic field produced by this current will be static and independent of time, therefore, it is not coupled with the electric field inside the coil and has no interaction with it.

2.4.2.1 Field equations

The field equations belong to the set of Maxwell's equations are:

$$\nabla \times \vec{\mathbf{H}} = \vec{\mathbf{J_0}}, \qquad in \qquad \Omega_0 \bigcup \Omega_m \tag{2.65}$$

$$\nabla \cdot \vec{\mathbf{B}} = 0, \qquad in \qquad \Omega_0 \bigcup \Omega_m \tag{2.66}$$

It is important to point out that equations (2.65) and (2.66) applies to the whole delimited space of the problem, both to the nonconducting substance (the air), and the conducting material (the Work Piece).

2.4.2.2 Constitutive equations

The magnetic flux density field is calculated differently in the air (nonconducting material) than in the magnetic material, due to the effects of the interaction between magnetic fields and matter. When calculating $\vec{\mathbf{B}}$ inside matter, the effects of magnetization should be considered. As it has been explained, the magnetic field inside the magnetic material will be weakened (diamagnetic behavior) or strengthened (paramagnetic behavior) depending on the type of material from a magnetic point of view.

$$\vec{\mathbf{B}} = \begin{cases} \mu_0 \vec{\mathbf{H}}, & \text{in air}, \Omega_0 \\ \mu_0 \mu_r \vec{\mathbf{H}}, & \text{in magnetically linear material}, \Omega_m \end{cases} \tag{2.67}$$

2.4.2.3 Charge conservation law

The source current density that drives along the coil matches the so called "current continuity equation ". This law in its whole meaning expresses that the variation of currents and charge distribution in relation with space and time depends on each other. However, in the cases studied here, the displacement current ($\frac{\partial \vec{\mathbf{D}}}{\partial t}$) is dismissed, thus now it is only important its meaning of continuity of the current. The following equation comes from the second exposed Maxwell equation (2.30) . Taking the divergence of that equation:

$$\nabla \cdot (\nabla \times \vec{\mathbf{H}}) = \nabla \cdot \vec{\mathbf{J_0}} = 0 \tag{2.68}$$

This meets the vectorial identity (A.10), which states that divergence of the curl of a vectorial field is equal to zero.

2.4.2.4 Boundary and interface conditions

At the time of solving the Maxwell's equations, the found solution must meet the boundary conditions along the boundary surfaces of the problem, and the interface conditions along the interface between different materials. In the cases studied here, the materials differentiate each other because of its degree of magnetization.

Open boundary conditions When applying a numerical method such as FEM, the space of resolution of the faced problem must be delimited. Usually, the open boundaries of the space of the problem are modeled as a sphere of infinite radius. The energy crossing this sphere must be equal to zero, thus the variation of energy of the magnetic field takes place only inside the sphere. This means that the magnetic field must vanish out of the sphere, i.e., at infinity.

Interface conditions the boundary conditions at the interface between different magnetic materials are the following:

- Continuity of the tangential component of the magnetic intensity vector $\vec{\mathbf{H}}$. Mathematically, that is described as follows:

$$\vec{\mathbf{n}} \times (\vec{\mathbf{H}}_{\mathbf{2}} - \vec{\mathbf{H}}_{\mathbf{1}}) = 0 \tag{2.69}$$

- Continuity of the normal component of the magnetic flux density vector $\vec{\mathbf{B}}$:

$$\vec{\mathbf{n}} \cdot (\vec{\mathbf{B}}_{\mathbf{2}} - \vec{\mathbf{B}}_{\mathbf{1}}) = 0 \tag{2.70}$$

2.4.3 Eddy current fields (AC current in the coil)

The current feeding the coil is alternating current (AC), a current varying sinusoidally in time. Therefore, in this case the time variation of the field magnitudes can not be neglected. The time varying current of the coil generates a time varying magnetic field as well. According to Faraday's law, represented in the first exposed Maxwell equation (2.25), a time varying magnetic field going through an electrical conductor, induces an electric field in it. This electric field produces the flowing of Eddy currents in the conductor. Because of the time varying nature of the magnetic field, the generated Eddy currents are time varying as well. Besides, the Eddy currents produce another time varying magnetic field which looks for counteracting the effects of the original magnetic field, the source field of the eddy currents. This is the so called "Lenz's law ".

The Eddy current fields problem is divided in different regions, according to whether there is a conducting material in the region or not. The reason is that only in the region of the conducting material the Eddy currents can arise.

2.4.3.1 Field equations

Here the governing equations are split up according to the region of the problem that is being analyzed. The part corresponding to the nonconducting material goes as follows:

$$\nabla \times \vec{\mathbf{H}} = \vec{\mathbf{J_0}} \quad in \quad \Omega_1 \tag{2.71}$$

$$\nabla \cdot \vec{\mathbf{B}} = 0 \quad in \quad \Omega_1 \tag{2.72}$$

In the conducting region, where the Eddy currents arise, the Faraday's law has to be taken into account, and the governing equations happen to be almost the complete set of Maxwell's law equations:

$$\nabla \times \vec{\mathbf{H}} = \vec{\mathbf{J}} \quad in \quad \Omega_2 \tag{2.73}$$

$$\nabla \cdot \vec{\mathbf{B}} = 0 \quad in \quad \Omega_2 \tag{2.74}$$

$$\nabla \times \vec{\mathbf{E}} = -\frac{\partial \vec{\mathbf{B}}}{\partial t} \quad in \quad \Omega_2 \tag{2.75}$$

2.4.3.2 Constitutive equations

Similar to the previous electromagnetic problem, the constitutive equations are different according to the region of the problem:

$$\vec{\mathbf{B}} = \begin{cases} \mu_0 \vec{\mathbf{H}} & \text{in air}, \Omega_1 \\ \mu_0 \mu_r \vec{\mathbf{H}} & \text{in magnetically linear material}, \Omega_2 \end{cases} \tag{2.76}$$

A equation must be added in the conducting region due to the arise of a electric field that maintains the Eddy currents:

$$\vec{\mathbf{J}} = \sigma \vec{\mathbf{E}} \tag{2.77}$$

2.4.3.3 Charge conservation law

The current continuity equation applies both for the source currents in the coil and the Eddy currents produced in the conductive material. Besides in case of Eddy currents, this equations assures that these currents close upon themselves.

$$\nabla \cdot \vec{\mathbf{J}} = 0 \tag{2.78}$$

2.4.3.4 Boundary and interface conditions

Open boundary conditions they are similar to the magneto-statics case.

Interface conditions the nonconducting region and the conducting region are coupled through the interface Γ_{12}. The boundary conditions in the interface are similar to the last problem, but with one addition due to the arise of the Eddy currents:

- Continuity of the tangential component of the magnetic intensity vector:

$$\vec{\mathbf{n}} \times (\vec{\mathbf{H_2}} - \vec{\mathbf{H_1}}) = 0 \tag{2.79}$$

- Continuity of the normal component of the magnetic flux density vector: $\vec{\mathbf{B}}$:

$$\vec{\mathbf{n}} \cdot (\vec{\mathbf{B_2}} - \vec{\mathbf{B_1}}) = 0 \tag{2.80}$$

- None of the Eddy currents can flow orthogonal to the interface:

$$\vec{\mathbf{J_2}} \cdot \vec{\mathbf{n_2}} = 0 \tag{2.81}$$

2.5 Potential formulations of Maxwell's equations

This chapter, fully based on (KUCZMANN 2009), shall help to locate the Maxwell's equations into the FE framework. The Finite Element Method (FEM) applied to Maxwell's equations, delivers an approximate solution of them, under prescribed boundary conditions. The partial differential equations to be solved can be formulated for field quantities or for potentials. The potentials formulations turn out to be more advantageous when making use of numerical analysis such as the FEM, for the set of Maxwell's equations can be reduced into one or two partial differential equations by using potentials. The boundary and interface conditions can also be represented by potentials.

2.5.1 Potential formulations in static magnetic fields problems

The formulations presented in this section apply for both the nonconducting region (air) and the conducting region, or where magnetic material is located.

2.5.1.1 Reduced magnetic scalar potential, the P-formulation

Under this formulation, the magnetic intensity vector $\vec{\mathbf{H}}$ is divided in two parts as follows:

$$\vec{\mathbf{H}} = \vec{\mathbf{T_0}} + \vec{\mathbf{H_m}} \tag{2.82}$$

$\vec{\mathbf{T_0}}$ is the so called impressed current vector potential. $\vec{\mathbf{H_m}}$ can be represented by the gradient of a scalar potential.

The division splits the magnetic intensity vector in a rotational part and a non-rotational part.

$$\nabla \times \vec{\mathbf{T_0}} = \vec{\mathbf{J_0}} \tag{2.83}$$

$$\nabla \times \vec{\mathbf{H_m}} = 0 \tag{2.84}$$

The curl of the impressed current vector potential matches the value of the current density. While $\vec{\mathbf{H_m}}$ is defined as non-rotational. This division according to the curl of each part, fulfills the second Maxwell's equation (2.30):

$$\nabla \times \vec{\mathbf{H}} = \nabla \times (\vec{\mathbf{T_0}} + \vec{\mathbf{H_m}}) = \nabla \times \vec{\mathbf{T_0}} + \cancel{\nabla \times \vec{\mathbf{H_m}}} = \vec{\mathbf{J_0}} \tag{2.85}$$

It should be pointed out that the two parts can be calculated separately. $\vec{\mathbf{T_0}}$ should satisfy the equation (2.83) and should be built up from the known source current density $\vec{\mathbf{J_0}}$. Then $\vec{\mathbf{H_m}}$ can be described as the gradient of a scalar potential.

⇒**Determine the impressed current vector potential** $\vec{\mathbf{T_0}}$ can be calculated through different methods. Here the method presented is the one implemented in **CalculiX**. It consists of applying the Biot-Savart's law, $\vec{\mathbf{T_0}}$ is calculated as the static magnetic intensity field produced in free space.

⇒**Determine the non-rotational part** According to the vectorial identity (A.11), valid for any scalar function φ, the non-rotational part $\vec{\mathbf{H_m}}$ can be determined from the negative gradient of a magnetic scalar potential P, i.e., because of the fact of being defined as non-rotational vectorial field, $\vec{\mathbf{H_m}}$ can be expressed as the gradient of a scalar function, known as the magnetic scalar potential in this case.

$$\vec{\mathbf{H_m}} = -\vec{\nabla}\mathbf{P} \tag{2.86}$$

Equation (2.82) can be rewritten as follows:

$$\vec{\mathbf{H}} = \vec{\mathbf{T_0}} - \vec{\nabla}\mathbf{P} \tag{2.87}$$

The magnetic scalar potential P, is known in this formulation as "reduced magnetic scalar potential ", due to the source term finds itself hidden in the impressive current vector potential $\vec{\mathbf{T_0}}$.

Through the linear constitutive equation (paramagnetic and diamagnetic cases), the magnetic flux density $\vec{\mathbf{B}}$ can be obtained:

$$\vec{\mathbf{B}} = \mu_0(\vec{\mathbf{T_0}} - \vec{\nabla}\mathbf{P}) \tag{2.88}$$

From the last equation, a generalized Laplace-Poisson equation can be derived. On this purpose, the fourth Maxwell equation should be applied (2.28):

$$\nabla \cdot \vec{\mathbf{B}} = \nabla \cdot (\mu_0\vec{\mathbf{T_0}}) - \nabla \cdot (\mu_0\vec{\nabla}\mathbf{P}) = 0 \tag{2.89}$$

The last formula constitute the field equation of the P-formulation for magneto-static cases.

It should be added that the P-formulation works well if the magnetic permeability of the materials dealt is not very high. Under this formulation, for low permeabilities, the value obtained for $\vec{\mathbf{H}}$ is very similar to the value of the impressed current vector potential T_0, the source magnetic field. In **CalculiX** that should not be a problem, as the implemented formulations are thought to be applied only in diamagnetic or paramagnetic material; ferromagnetic materials, which have a high magnetic permeability, are put aside.

When working with materials with higher permeabilities, the value of T_0 tend to match the value of ∇P. That may cause to arise a cancellation error in the formula (2.87).

2.5.1.2 The magnetic vector potential, the A-formulation

It was spotted in a previous section that the potential magnetic vector $\vec{\mathbf{A}}$ is not unique by nature in its solution. The Coulomb gauge resolves this problem, assuring that the solution under this formulation is unique.

$\Rightarrow$**Coulomb gauge** In 2D problems the formula of the Coulomb gauge $\nabla \cdot \vec{\mathbf{A}} = 0$, is satisfied automatically (it can be demonstrated mathematically applying the divergence of the curl of $\vec{\mathbf{A}}$). However, that is not the case in 3D, where the uniqueness of $\vec{\mathbf{A}}$ is not directly demonstrated. The are two ways to assure the uniqueness of $\vec{\mathbf{A}}$, each delivers a different A-formulation:

- **Gauged formulation:** it consists of the implicit enforcement of the Coulomb gauge $\nabla \cdot \vec{\mathbf{A}} = 0$
- **Ungauged formulation:** it consists of applying a numerical technique insensitive to Coulomb gauge.

The gauged formulation is the one implemented in **CalculiX**, so it will be the one explained. Furthermore, it should be noticed that the gauged formulations deliver a one unique solution for the potentials, whereas the ungauged formulations result in an infinite number of solutions for the potentials. Nevertheless, the field magnitudes calculated from the potentials are always unique, as the solution of Maxwell's equations is unique too.

⇒**Gauged formulation** The lack of uniqueness of the potential magnetic vector leads to numerical difficulties when solving 3D problems. The uniqueness can be prescribed implicitly by the Coulomb gauge. This process consists of the modification of the following equation:

$$\nabla \times \vec{\mathbf{B}} = \mu_0 \vec{\mathbf{J_0}} \rightarrow \nabla \times (\frac{1}{\mu_0} \nabla \times \vec{\mathbf{A}}) = \vec{\mathbf{J_0}} \tag{2.90}$$

The modification starts with the definition of the functional of the last equation. The functional is defined as a linear case:

$$\Im(\vec{\mathbf{A}}) = \frac{1}{2} \int_{\Omega} \frac{1}{\mu} |\nabla \times \vec{\mathbf{A}}|^2 d\Omega - \int_{\Omega} \vec{\mathbf{J_0}} \cdot \vec{\mathbf{A}} d\Omega \tag{2.91}$$

The principle of minimum energy dictates that the functional has a minimum value only if the $\vec{\mathbf{A}}$ taken as solution satisfies the equation (2.90) and the corresponding boundary conditions of the problem. This minimum value provides a stable configuration of the magnetic vector potential. Once that was explained, the functional can now be modified by applying the Coulomb gauge:

$$\Im(\vec{\mathbf{A}}) = \frac{1}{2} \int_{\Omega} \frac{1}{\mu} |\nabla \times \vec{\mathbf{A}}|^2 d\Omega - \int_{\Omega} \vec{\mathbf{J_0}} \cdot \vec{\mathbf{A}} d\Omega + \frac{1}{2} \int_{\Omega} \frac{1}{\mu} (\nabla \cdot \vec{\mathbf{A}})^2 d\Omega \tag{2.92}$$

This time, the functional will have a minimum value only if $\vec{\mathbf{A}}$ satisfies too the Coulomb gauge and the corresponding appended boundary conditions. This is the implicit enforcement of Coulomb gauged that was mentioned. If the functional keeps being operated, it allows to combine the equation (2.90) and $\nabla \cdot \vec{\mathbf{A}} = 0$, into one partial differential equation:

$$\nabla \times (\frac{1}{\mu} \nabla \times \vec{\mathbf{A}}) - \nabla (\frac{1}{\mu} \nabla \cdot \vec{\mathbf{A}}) = \vec{\mathbf{J_0}} \tag{2.93}$$

This equation constitutes the field equation of the A-formulation, and its solution delivers a vector $\vec{\mathbf{A}}$ unique according to Coulomb gauge.

2.5.1.3 Combination of the magnetic vector potential and the magnetic scalar potential, the A-P-formulation

The combination of P-formulation with the A-formulation results in an advantageous formulation. A general approach would be to use the A-formulation in the whole domain of the problem, nevertheless, it is more favorable to use both the potential magnetic vector $\vec{\mathbf{A}}$ and the reduced magnetic potential P. The strategy to follow in this combined formulation, consists of applying the reduced magnetic scalar potential in the nonconducting region (air), and the potential magnetic vector in the conducting region or where the magnetic material is located. In this way, the number of unknowns in the nonconducting region (air) is reduced regarding the use of the A-formulation on it. However, it should be borne in mind the addiction of extra interface conditions between the two regions, due to the two formulations are coupled through such conditions.

The field equations are the combination of the last discussed formulations:

$$\nabla \cdot (\mu_0 \vec{\mathbf{T_0}}) - \nabla \cdot (\mu_0 \vec{\nabla} \mathbf{P}) = 0 \qquad in \qquad \Omega_1 \tag{2.94}$$

$$\nabla \times (\frac{1}{\mu} \nabla \times \vec{\mathbf{A}}) - \nabla (\frac{1}{\mu} \nabla \cdot \vec{\mathbf{A}}) = \vec{\mathbf{J_0}} \qquad in \qquad \Omega_2 \tag{2.95}$$

2.5.2 Potential formulations in Eddy current fields problems

The formulation here presented applies only to the Eddy current region, i.e., the region where the conducting material is located.

2.5.2.1 The magnetic vector potential and electric scalar potential, the A,V-formulation

When equation (2.16) is substituted in the first Maxwell equation (2.25), the following equation is obtained:

$$\nabla \times \vec{\mathbf{E}} = \frac{\partial}{\partial t} \nabla \times \vec{\mathbf{A}} = -\nabla \times \frac{\partial \vec{\mathbf{A}}}{\partial t} \tag{2.96}$$

This can be done, because the curl, a derivation by space, can be interchanged with a derivation by time. If the last equation is operated, the following is obtained:

$$\nabla \times (\vec{\mathbf{E}} + \frac{\partial \vec{\mathbf{A}}}{\partial t}) = 0 \tag{2.97}$$

Once more in this case, it results convenient to look at a vectorial identity. The identity (A.11) is similar in form to the last equation. That means the non-rotational vector $\vec{\mathbf{E}} + \frac{\partial \vec{\mathbf{A}}}{\partial t}$, defined between the brackets in the last equation, can be expressed as the gradient of a scalar potential. This potential happens to be the electric scalar potential:

$$\vec{\mathbf{E}} + \frac{\partial \vec{\mathbf{A}}}{\partial t} = -\vec{\nabla}\mathbf{V} \rightarrow \vec{\mathbf{E}} = -\vec{\nabla}\mathbf{V} - \frac{\partial \vec{\mathbf{A}}}{\partial t} \tag{2.98}$$

In order to arrive to one of the field equations of this formulation, a development of formulas is needed starting from the following equation, one of the forms of the second Maxwell equation (2.30) (Ampere's law in differential form):

$$\nabla \times \vec{\mathbf{H}} = \vec{\mathbf{J}} \tag{2.99}$$

Supposing the conducting material as magnetically linear $\vec{\mathbf{B}} = \mu\vec{\mathbf{H}}$ and with equation $\vec{\mathbf{J}} = \sigma\vec{\mathbf{E}}$, it is obtained:

$$\nabla \times (\frac{1}{\mu}\nabla \times \vec{\mathbf{A}}) = \sigma\vec{\mathbf{E}} \tag{2.100}$$

By substituting $\vec{\mathbf{E}}$ with equation (2.98):

$$\nabla \times (\frac{1}{\mu}\nabla \times \vec{\mathbf{A}}) = \sigma(-\vec{\nabla}\mathbf{V} - \frac{\partial \vec{\mathbf{A}}}{\partial t}) \rightarrow \nabla \times (\frac{1}{\mu}\nabla \times \vec{\mathbf{A}}) + \sigma\vec{\nabla}\mathbf{V} + \sigma\frac{\partial \vec{\mathbf{A}}}{\partial t}) = 0 \tag{2.101}$$

And that is one of the field equations of this formulation. The second field equation is obtained by substituting in the charge conservation equation (2.37), the equation $\vec{\mathbf{J}} = \sigma\vec{\mathbf{E}}$ and successively the equation (2.98):

$$\nabla \cdot \vec{\mathbf{J}} = 0 \rightarrow \nabla \cdot (\sigma\vec{\mathbf{E}}) = 0 \rightarrow -\nabla \cdot (\sigma\vec{\nabla}\mathbf{V} + \sigma\frac{\partial \vec{\mathbf{A}}}{\partial t}) = 0 \tag{2.102}$$

In this formulation there are two field equations because there are two unknown functions as well, the vectorial function $\vec{\mathbf{A}}$ and the scalar funtion V. Moreover, it is interesting to point out, that equation (2.101) comes from the development of the differential form of

Ampere's law itself, and equation (2.102) comes from the development of the divergence of Ampere's law.

The two field equations described up to now constitute the ungauged form of this formulation. The reason why is the lack of uniqueness of vector $\vec{\mathbf{A}}$ under this form. Similar to the case of static magnetic fields, under the A-formulation, the Coulomb gauge is needed to assure the uniqueness of vector $\vec{\mathbf{A}}$.

⇒**Gauged formulation** By application of Coulomb gauge, the term $-\nabla(\frac{1}{\mu}\nabla \cdot \vec{\mathbf{A}})$ should be appended to equation (2.101). In this way, the field equations of A, V-formulation for Eddy current field problems, whose solution is unique according to Coulomb gauge, are the following:

$$\nabla \times (\frac{1}{\mu}\nabla \times \vec{\mathbf{A}}) + \sigma\vec{\nabla}\mathbf{V} + \sigma\frac{\partial\vec{\mathbf{A}}}{\partial t} - \nabla(\frac{1}{\mu}\nabla \cdot \vec{\mathbf{A}}) = 0 \tag{2.103}$$

$$-\nabla \cdot (\sigma\vec{\nabla}\mathbf{V} + \sigma\frac{\partial\vec{\mathbf{A}}}{\partial t}) = 0 \tag{2.104}$$

2.5.2.2 Coupling static magnetic and Eddy current fields problems

In previous sections, potential formulations for both the nonconducting region and the conducting region have been developed. In the conducting region, the formulations differ each other whether the problem faced is a static magnetic field problem or an Eddy current field problem. The problem faced in inductive heating is an Eddy current field problem, which requires to analyze as well the nonconducting area. In order to accomplish a global analysis, a potential formulation is needed, which considers globally the two different regions of the problem (conducting and nonconducting), but at the same time, applies individually for each region the suitable formulation. Because of that, the formulation defined should include a coupling between the already defined formulations in the last sections.

The A, V-formulation for the conducting region can be coupled with the A-formulation for the nonconducting region. However, because of the same reason pointed out with A-*P*-formulation, to reduce the number of unknowns in the nonconducting region (air), it results more advantageous to use the reduced magnetic scalar potential, what defines a A, V-*P*-formulation. This last formulation will now be explained, together with an improved formulation of it, the A, V-A-*P*-formulation. To understand the improvement, it turns necessary before beginning with the formulations to clarify the concept of simple and multiple connected bodies.

Simple connected body Topologically speaking, a body is simple connected if it is path connected, which means that any path drawn between any two points contained in the body, will be contained in the body as well. Besides, the path can be transformed, for instance, being reduced to a point, without leaving the body. In 2D, a example of simple connected body is a disk; in 3D, an example would be a sphere.

Multiple connected body these bodies are connected, they have its matter built up together, but they are not simply connected. A path in these bodies can not be reduced in any way to a point without leaving the body. An example in 2D is a circle, and in 3D, a ring would be a multiple connected body, or not simply connected.

2.5.2.3 Gauged A,V-P formulation

This formulation can be applied only if the conducting region is a simple connected domain. In the conduction region, a A,V-formulation is applied, while in the nonconducting region, as it was said, aiming to reduce the number of unknowns, the reduced magnetic scalar potential P is used instead of the magnetic vector potential $\vec{\mathbf{A}}$. Therefore, the formulation is a modification of the A,V-A formulation. The following are the field equations:

$$\nabla \cdot (\mu \nabla \vec{\mathbf{P}}) = \nabla \cdot \mu \vec{\mathbf{T_0}} \qquad in \quad \Omega_1 \tag{2.105}$$

$$\nabla \times (\frac{1}{\mu} \nabla \times \vec{\mathbf{A}}) + \sigma \nabla \vec{\mathbf{V}} + \sigma \frac{\partial \vec{\mathbf{A}}}{\partial t} - \nabla(\frac{1}{\mu} \nabla \cdot \vec{\mathbf{A}}) = 0 \qquad in \quad \Omega_2 \tag{2.106}$$

$$-\nabla \cdot (\sigma \nabla \vec{\mathbf{V}} + \sigma \frac{\partial \vec{\mathbf{A}}}{\partial t}) = 0 \qquad in \quad \Omega_2 \tag{2.107}$$

2.5.2.4 Gauged A,V-A-P formulation

As it was stated, the last described formulation can only be applied to a simple connected domain of the conducting region. The A,V-A-P formulation is a modification of A,V-P formulation, in order to be able to apply it to multiple connected domains. Therefore, the A,V-A-P formulation can be used when the conducting region is a multiple connected domain.

The multiple connected bodies are assorted so because they have holes in them that do not belong to the body. In the faced Eddy current field problem, these holes belong to the nonconducting region. The strategy of the A,V-A-P formulation is to apply a

A,V formulation to the conducting region, the multiple connected body, and at the same time, apply the potential magnetic vector $\vec{\mathbf{A}}$ to such holes, which are part of the nonconducting region, but are spatially located close to the conducting region. The idea is to transform a multiple connected body into a simple connected body through the use of the formulation. To do that, the nonconducting domain is split up in two parts domains, one remains as the original nonconducting domain minus the other part. This last other part is the holes, which if are added to the multiple connected body (the conducting domain), make it simple connected.

Consequently, under this formulation the problem is divided in three domains: the conducting domain and the two parts of the nonconducting domain. A,V formulation is applied in the conducting domain. Vector $\vec{\mathbf{A}}$ is applied in the part of the nonconducting region, that if added to the conducting region, makes it simple connected. Lastly, P-formulation is applied in the rest of the nonconducting region. The equations of the formulation are the following:

$$\nabla \cdot (\mu \vec{\nabla} \mathbf{P}) = \nabla \cdot \mu \vec{\mathbf{T_0}} \qquad in \quad \Omega_1 \tag{2.108}$$

$$\nabla \times (\frac{1}{\mu} \nabla \times \vec{\mathbf{A}}) + \sigma \vec{\nabla} \mathbf{V} + \sigma \frac{\partial \vec{\mathbf{A}}}{\partial t} - \nabla (\frac{1}{\mu} \nabla \cdot \vec{\mathbf{A}}) = 0 \qquad in \quad \Omega_2 \tag{2.109}$$

$$-\nabla \cdot (\sigma \vec{\nabla} \mathbf{V} + \sigma \frac{\partial \vec{\mathbf{A}}}{\partial t}) = 0 \qquad in \quad \Omega_2 \tag{2.110}$$

$$\nabla \times (\frac{1}{\mu} \nabla \times \vec{\mathbf{A}}) - \nabla (\frac{1}{\mu} \nabla \cdot \vec{\mathbf{A}}) = \vec{\mathbf{J_0}} \qquad in \quad \Omega_3 \tag{2.111}$$

The formulation here presented is the gauged version, which means like in the previous formulations, that the solution of vector $\vec{\mathbf{A}}$ under this formulation is unique according to the Coulomb gauge ($\nabla \cdot \vec{\mathbf{A}} = 0$).

3 State of the technology

This chapter helps to understand how **CalculiX** is implemented. It is fully based in (DHONDT 2016b).

3.1 Implementation in CalculiX

In the electromagnetic field computation problems dealt in this thesis the partial differential equations to be solved are elliptic and parabolic types, and their order is two.

3.2 Strong form

3.2.0.1 Field equations

$$\nabla \cdot (\mu(\vec{\mathbf{T}}_{\mathbf{0}} - \vec{\bigtriangledown}\mathbf{P})) = 0 \tag{3.1}$$

$$\nabla \times \frac{1}{\mu}(\nabla \times \vec{\mathbf{A}}) - \nabla \frac{1}{\mu} \nabla \cdot \vec{\mathbf{A}} + \sigma \frac{\partial \vec{\mathbf{A}}}{\partial t} + \sigma \vec{\bigtriangledown}\mathbf{V} = 0 \tag{3.2}$$

$$\nabla \cdot \sigma(\frac{\partial \vec{\mathbf{A}}}{\partial t} + \vec{\nabla}\mathbf{V}) = 0 \tag{3.3}$$

$$\nabla \times \frac{1}{\mu}(\nabla \times \vec{\mathbf{A}}) - \nabla \frac{1}{\mu} \nabla \cdot \vec{\mathbf{A}} = 0 \tag{3.4}$$

3.2.0.2 Boundary and interface conditions

$$\mu(\vec{\mathbf{T}}_{\mathbf{0}} - \vec{\nabla}\mathbf{P}) \cdot \vec{\mathbf{n}}_{\mathbf{1}} + (\nabla \times \vec{\mathbf{A}}) \cdot \vec{\mathbf{n}}_{\mathbf{2}} = 0 \tag{3.5}$$

$$(\vec{\mathbf{T}}_{\mathbf{0}} - \vec{\nabla}\mathbf{P}) \times \vec{\mathbf{n}}_{\mathbf{1}} + \frac{1}{\mu_1}(\nabla \times \vec{\mathbf{A}}) \times \vec{\mathbf{n}}_{\mathbf{2}} = 0 \tag{3.6}$$

$$(\frac{\partial \vec{\mathbf{A}}}{\partial t} + \vec{\nabla}\mathbf{V})_2 \cdot \vec{\mathbf{n_2}} = 0 \tag{3.7}$$

$$\vec{\mathbf{A}} \cdot \vec{\mathbf{n_2}} = 0 \tag{3.8}$$

$$\vec{\mathbf{B}} \cdot \vec{\mathbf{n_1}} = 0 \tag{3.9}$$

3.3 Weak form

Weighting functions

$$\delta\vec{A} \tag{3.10}$$

$$\delta V \tag{3.11}$$

$$\delta P \tag{3.12}$$

The equation (3.2) is multiplied by the weighting function $\delta\vec{A}$:

$$\nabla \times \frac{1}{\mu}(\nabla \times \vec{\mathbf{A}})\delta\vec{A} - \nabla\frac{1}{\mu}\nabla \cdot \vec{\mathbf{A}}\delta\vec{A} + \sigma\frac{\partial \vec{\mathbf{A}}}{\partial t}\delta\vec{A} + \sigma\vec{\nabla}\mathbf{V}\delta\vec{A} = 0 \tag{3.13}$$

The vectorial identities (A.6) and (A.7) are used to transform the last equation. In the following, the terms of equation (3.13) which are developed through the vectorial identities are presented; the suitable form of the vectorial identity that is being applied is presented as well.

The vectorial identity (A.6) is used to develop the second term of the left hand-side in equation (3.13):

$$\nabla \cdot (\varphi\vec{\mathbf{F}}) = \varphi\nabla \cdot \vec{\mathbf{F}} + (\vec{\nabla}')\vec{\mathbf{F}} \to -(\vec{\nabla}')\vec{\mathbf{F}} = \varphi\nabla \cdot \vec{\mathbf{F}} - \nabla \cdot (\varphi\vec{\mathbf{F}}) \begin{cases} \varphi = \frac{1}{\mu}\nabla\vec{\mathbf{A}} \\ \vec{\mathbf{F}} = \delta\vec{\mathbf{A}} \end{cases} \tag{3.14}$$

$$-\nabla(\frac{1}{\mu}\nabla \cdot \vec{\mathbf{A}})\delta\vec{A} = \frac{1}{\mu}(\nabla \cdot \vec{\mathbf{A}})(\nabla \cdot \delta\vec{\mathbf{A}}) - \nabla \cdot (\frac{1}{\mu}(\nabla \cdot \vec{\mathbf{A}})\delta\vec{\mathbf{A}}) \tag{3.15}$$

Now the vectorial identity (A.7) is used in the first term of the left hand-side in equation (3.13):

$$\nabla\cdot(\vec{\mathbf{F}}\times\vec{\mathbf{G}})=(\nabla\times\vec{\mathbf{F}})\vec{\mathbf{G}}-\vec{\mathbf{F}}(\nabla\times\vec{\mathbf{G}})\rightarrow\vec{\mathbf{F}}(\nabla\times\vec{\mathbf{G}})=(\nabla\times\vec{\mathbf{F}})\vec{\mathbf{G}}-\nabla\cdot(\vec{\mathbf{F}}\times\vec{\mathbf{G}})\begin{cases}\vec{\mathbf{F}}=\delta\vec{\mathbf{A}}\\ \vec{\mathbf{G}}=\dfrac{1}{\mu}\nabla\times\vec{\mathbf{A}}\end{cases}\tag{3.16}$$

$$-\nabla\times\frac{1}{\mu}(\nabla\times\vec{\mathbf{A}})\delta\vec{\mathbf{A}}=(\nabla\times\delta\vec{\mathbf{A}})(\frac{1}{\mu}(\nabla\times\vec{\mathbf{A}}))-\nabla\cdot(\delta\vec{\mathbf{A}}\times(\frac{1}{\mu}\nabla\times\vec{\mathbf{A}}))\tag{3.17}$$

Putting all together, the following equation is obtained:

$$\begin{aligned}(\nabla\times\delta\vec{\mathbf{A}})(\frac{1}{\mu}(\nabla\times\vec{\mathbf{A}}))-\nabla\cdot(\delta\vec{\mathbf{A}}\times(\frac{1}{\mu}\nabla\times\vec{\mathbf{A}}))+\frac{1}{\mu}(\nabla\cdot\vec{\mathbf{A}})(\nabla\cdot\delta\vec{\mathbf{A}})&\\ -\nabla\cdot(\frac{1}{\mu}(\nabla\cdot\vec{\mathbf{A}})\delta\vec{\mathbf{A}})+\sigma\frac{\partial\vec{\mathbf{A}}}{\partial t}\delta\vec{A}+\sigma\vec{\nabla}\mathbf{V}\delta\vec{A}=0&\end{aligned}\tag{3.18}$$

$$\begin{aligned}\int\limits_{\Omega_2}(\nabla\times\delta\vec{\mathbf{A}})(\frac{1}{\mu}(\nabla\times\vec{\mathbf{A}}))dV-\int\limits_{\Omega_2}\nabla\cdot(\delta\vec{\mathbf{A}}\times(\frac{1}{\mu}\nabla\times\vec{\mathbf{A}}))dV+\int\limits_{\Omega_2}\frac{1}{\mu}(\nabla\cdot\vec{\mathbf{A}})(\nabla\cdot\delta\vec{\mathbf{A}})dV&\\ -\int\limits_{\Omega_2}\nabla\cdot(\frac{1}{\mu}(\nabla\cdot\vec{\mathbf{A}})\delta\vec{\mathbf{A}})dV+\int\limits_{\Omega_2}\sigma\delta\vec{A}(\frac{\partial\vec{\mathbf{A}}}{\partial t}+\vec{\nabla}\mathbf{V})dV=0&\end{aligned}\tag{3.19}$$

In the second and fourth term of the last equation, the Gauss's theorem (A.13) of the divergence can be applied to transform the volume integrals in surface integrals. Here it is important to remember that domain Ω_2 or the conducting region is defined without free boundaries, all of its boundary consists of an interface(Γ_{12}) with domain Ω_1, as it is completely surrounded by the nonconducting region.

$$\begin{aligned}\int\limits_{\Omega_2}(\nabla\times\delta\vec{\mathbf{A}})(\frac{1}{\mu}(\nabla\times\vec{\mathbf{A}}))dV-\int\limits_{\Gamma_{12}}(\delta\vec{\mathbf{A}}\times(\frac{1}{\mu}\nabla\times\vec{\mathbf{A}}))\cdot d\vec{\mathbf{S_2}}+\int\limits_{\Omega_2}\frac{1}{\mu}(\nabla\cdot\vec{\mathbf{A}})(\nabla\cdot\delta\vec{\mathbf{A}})dV&\\ -\int\limits_{\Gamma_{12}}(\frac{1}{\mu}(\nabla\cdot\vec{\mathbf{A}})\delta\vec{\mathbf{A}})\cdot d\vec{\mathbf{S_2}}+\int\limits_{\Omega_2}\sigma\delta\vec{A}(\frac{\partial\vec{\mathbf{A}}}{\partial t}+\vec{\nabla}\mathbf{V})dV=0&\end{aligned}\tag{3.20}$$

At this point it should be cleared out that $d\vec{\mathbf{S_2}}$, known as differential surface vector, is a vector with differential module, whose direction aims out and normal to the surface of domain Ω_2, thus it can be written too as follows:

$$d\vec{\mathbf{S_2}} = \vec{\mathbf{n_2}} dS_2 \tag{3.21}$$

$\vec{\mathbf{n_2}}$ is an unit vector, normal to the surface of domain Ω_2. Hence, the surface integrals of equation (3.20) can be written too:

$$-\int\limits_{\Gamma_{12}} (\delta\vec{\mathbf{A}} \times (\frac{1}{\mu}\nabla \times \vec{\mathbf{A}})) \cdot \vec{\mathbf{n_2}} dS_2 - \cancel{\int\limits_{\Gamma_{12}} (\frac{1}{\mu}(\nabla \cdot \vec{\mathbf{A}})\delta\vec{\mathbf{A}}) \cdot \vec{\mathbf{n_2}} dS_2} \tag{3.22}$$

To simplify the second of these surface integrals, the boundary conditions can be reviewed. Specifically, the boundary condition (3.8) must be satisfied by the weighting functions as well, which means:

$$\delta\vec{\mathbf{A}} \cdot \vec{\mathbf{n_2}} = 0 \tag{3.23}$$

Applying the last boundary condition in equation (3.20) cancels the second surface integral.

Now the focus is on the first surface integral of equation (3.20). By vectorial identity (A.9), the integrand of the first surface integral can be changed so:

$$(\delta\vec{\mathbf{A}} \times (\frac{1}{\mu}\nabla \times \vec{\mathbf{A}})) \cdot \vec{\mathbf{n_2}} = \delta\vec{\mathbf{A}} \cdot ((\frac{1}{\mu}\nabla \times \vec{\mathbf{A}}) \times \vec{\mathbf{n_2}}) \tag{3.24}$$

The integrand can still be changed by considering boundary equation (3.6):

$$\delta\vec{\mathbf{A}} \cdot ((\frac{1}{\mu}\nabla \times \vec{\mathbf{A}}) \times \vec{\mathbf{n_2}}) = -\delta\vec{\mathbf{A}} \cdot ((\vec{\mathbf{T_0}} - \vec{\nabla}\mathbf{P}) \times \vec{\mathbf{n_1}}) = \delta\vec{\mathbf{A}} \cdot ((\vec{\mathbf{T_0}} - \vec{\nabla}\mathbf{P}) \times \vec{\mathbf{n_2}}) \tag{3.25}$$

The first surface integral from equation (3.20) can now be expressed:

$$\int\limits_{\Gamma_{12}} (-\delta\vec{\mathbf{A}} \cdot ((\vec{\mathbf{T_0}} - \vec{\nabla}\mathbf{P}) \times \vec{\mathbf{n_2}}) dS_2 = \int\limits_{\Gamma_{12}} (\delta\vec{\mathbf{A}} \cdot (\vec{\mathbf{T_0}} \times \vec{\mathbf{n_2}}) - \delta\vec{\mathbf{A}} \cdot (\vec{\nabla}\mathbf{P} \times \vec{\mathbf{n_2}})) dS_2 \tag{3.26}$$

Again, the vector identity (A.9) can be useful to change the last surface integral if it is applied to the second term of the integrand:

$$\int\limits_{\Gamma_{12}} (\delta\vec{\mathbf{A}} \cdot (\vec{\mathbf{T_0}} \times \vec{\mathbf{n_2}}) - (\delta\vec{\mathbf{A}} \times \vec{\nabla}\mathbf{P}) \cdot \vec{\mathbf{n_2}}) dS_2 \tag{3.27}$$

The equation can still be developed. By using the vector identity (A.7), a different way to express the second term of the integrand is obtained. Here, to change the vectorial identity as it is needed, a property of the vectorial product has to be used:

$$\begin{gathered}\nabla \times (\varphi \vec{\mathbf{F}}) = \varphi \nabla \times \vec{\mathbf{F}} + (\vec{\nabla}' \times \vec{\mathbf{F}}) \rightarrow \nabla \times (\varphi \vec{\mathbf{F}}) = \varphi \nabla \times \vec{\mathbf{F}} \overbrace{-(\vec{\mathbf{F}} \times \vec{\nabla}')}^{\vec{\mathbf{F}} \times \vec{\mathbf{G}} = -\vec{\mathbf{G}} \times \vec{\mathbf{F}}} \rightarrow \\ (\vec{\mathbf{F}} \times \vec{\nabla}') = \varphi \nabla \times \vec{\mathbf{F}} - \nabla \times (\varphi \vec{\mathbf{F}}) \begin{cases} \varphi = P \\ \vec{\mathbf{F}} = \delta \vec{\mathbf{A}} \end{cases}\end{gathered} \tag{3.28}$$

$$(\delta \vec{\mathbf{A}} \times \vec{\nabla} \mathbf{P}) \cdot \vec{\mathbf{n_2}} = P(\nabla \times \delta \vec{\mathbf{A}})\vec{\mathbf{n_2}} - (\nabla \times (P \delta \vec{\mathbf{A}}))\vec{\mathbf{n_2}} \tag{3.29}$$

Finally, the surface integral looks like:

$$\int_{\Gamma_{12}} (\delta \vec{\mathbf{A}} \cdot (\vec{\mathbf{T_0}} \times \vec{\mathbf{n_2}}) - P(\nabla \times \delta \vec{\mathbf{A}})\vec{\mathbf{n_2}} + (\nabla \times (P \delta \vec{\mathbf{A}}))\vec{\mathbf{n_2}}) dS_2 \tag{3.30}$$

The surface referred by the last integral is supposed to be closed. A integral of closed surface open the possibility to apply the Stokes' theorem (A.14), which in turn, allows to eliminate a term of the surface integral. By Stokes' theorem:

$$\begin{aligned}&\int_{\Gamma_{12}} (\delta \vec{\mathbf{A}} \cdot (\vec{\mathbf{T_0}} \times \vec{\mathbf{n_2}}) - P(\nabla \times \delta \vec{\mathbf{A}})\vec{\mathbf{n_2}} + (\nabla \times (P \delta \vec{\mathbf{A}}))\vec{\mathbf{n_2}}) dS_2 = \\ &\quad \int_{\Gamma_{12}} (\delta \vec{\mathbf{A}} \cdot (\vec{\mathbf{T_0}} \times \vec{\mathbf{n_2}})) dS_2 - \int_{\Gamma_{12}} (P(\nabla \times \delta \vec{\mathbf{A}})\vec{\mathbf{n_2}}) dS_2 + \int_{\Gamma_{12}} ((\nabla \times (P \delta \vec{\mathbf{A}}))\vec{\mathbf{n_2}}) dS_2 = \\ &\qquad \int_{\Gamma_{12}} (\delta \vec{\mathbf{A}} \cdot (\vec{\mathbf{T_0}} \times \vec{\mathbf{n_2}})) dS_2 - \int_{\Gamma_{12}} (P(\nabla \times \delta \vec{\mathbf{A}})\vec{\mathbf{n_2}}) dS_2 + \underbrace{\int (P \delta \vec{\mathbf{A}}) \cdot d\vec{\mathbf{l}}}_{Stokes} = \\ &\qquad\qquad \int_{\Gamma_{12}} (\delta \vec{\mathbf{A}} \cdot (\vec{\mathbf{T_0}} \times \vec{\mathbf{n_2}}) - \int_{\Gamma_{12}} (P(\nabla \times \delta \vec{\mathbf{A}})\vec{\mathbf{n_2}}) dS_2\end{aligned} \tag{3.31}$$

This was the last simplification practiced in the first surface integral of equation (3.20). The equation (3.20) constitutes one of the governing equations belonging to the final weak form, that is implemented in **CalculiX**:

$$\int_{\Omega_2} (\nabla \times \delta\vec{\mathbf{A}})(\frac{1}{\mu}(\nabla \times \vec{\mathbf{A}}))dV - \int_{\Gamma_{12}} (\delta\vec{\mathbf{A}} \cdot (\vec{\mathbf{T_0}} \times \vec{\mathbf{n_2}}) + \int_{\Gamma_{12}} (P(\nabla \times \delta\vec{\mathbf{A}})\vec{\mathbf{n_2}})dS_2$$
$$+ \int_{\Omega_2} \frac{1}{\mu}(\nabla \cdot \vec{\mathbf{A}})(\nabla \cdot \delta\vec{\mathbf{A}})dV + \int_{\Omega_2} \sigma\delta\vec{A}(\frac{\partial\vec{\mathbf{A}}}{\partial t} + \vec{\nabla}\mathbf{V})dV = 0 \quad (3.32)$$

The equation (3.3) is multiplied by the weighting function δV:

$$\nabla \cdot \sigma(\frac{\partial\vec{\mathbf{A}}}{\partial t} + \vec{\nabla}\mathbf{V})\delta V = 0 \quad (3.33)$$

Taking into account the vectorial identity (A.6), the last equation can be expressed distinctively:

$$\nabla \cdot (\varphi\vec{\mathbf{F}}) = \varphi\nabla \cdot \vec{\mathbf{F}} + (\vec{\nabla}') \cdot \vec{\mathbf{F}} \rightarrow \varphi\nabla \cdot \vec{\mathbf{F}} = \nabla \cdot (\varphi\vec{\mathbf{F}}) - (\vec{\nabla}') \cdot \vec{\mathbf{F}} \begin{cases} \varphi = \delta V \\ \vec{\mathbf{F}} = \sigma(\frac{\partial\vec{\mathbf{A}}}{\partial t} + \vec{\nabla}\mathbf{V}) \end{cases} \quad (3.34)$$

$$\nabla \cdot (\sigma\delta V(\frac{\partial\vec{\mathbf{A}}}{\partial t} + \vec{\nabla}\mathbf{V})) - \nabla\delta\vec{\mathbf{V}} \cdot \sigma(\frac{\partial\vec{\mathbf{A}}}{\partial t} + \vec{\nabla}\mathbf{V}) = 0 \quad (3.35)$$

$$\int_{\Omega_2} (\nabla \cdot (\sigma\delta V(\frac{\partial\vec{\mathbf{A}}}{\partial t} + \nabla\vec{\mathbf{V}})))dV - \int_{\Omega_2} (\nabla\delta\vec{\mathbf{V}} \cdot \sigma(\frac{\partial\vec{\mathbf{A}}}{\partial t} + \nabla\vec{\mathbf{V}}))dV = 0 \quad (3.36)$$

The first volume integral of last equation can become a surface integral through the application of Gauss theorem (A.13) of divergence:

$$\underbrace{\int_{\Gamma_{12}} (\sigma\delta V(\frac{\partial\vec{\mathbf{A}}}{\partial t} + \vec{\nabla}\mathbf{V}))\vec{\mathbf{n_2}}dS_2}_{Gauss} - \int_{\Omega_2} (\nabla\delta\vec{\mathbf{V}} \cdot \sigma(\frac{\partial\vec{\mathbf{A}}}{\partial t} + \nabla\vec{\mathbf{V}}))dV = 0 \quad (3.37)$$

Then the obtained surface integral can be canceled by application of the boundary condition (3.7). Lastly, another governing equation belonging to the final weak form is accomplished:

$$\underbrace{\int_{\Gamma_{12}} (\sigma\delta V(\frac{\partial\vec{\mathbf{A}}}{\partial t} + \vec{\nabla}\mathbf{V}))\vec{\mathbf{n_2}} dS_2}_{Boundary\ (3.7)} - \int_{\Omega_2} (\nabla\delta\vec{\mathbf{V}} \cdot \sigma(\frac{\partial\vec{\mathbf{A}}}{\partial t} + \nabla\vec{\mathbf{V}}))dV = 0 \rightarrow$$

$$\int_{\Omega_2} (\nabla\delta\vec{\mathbf{V}} \cdot \sigma(\frac{\partial\vec{\mathbf{A}}}{\partial t} + \vec{\nabla}\mathbf{V}))dV = 0 \quad (3.38)$$

The equation (3.1) is multiplied by the weighting function δP:

$$\nabla \cdot (\mu(\vec{\mathbf{T}}_{\mathbf{0}} - \vec{\nabla}\mathbf{P}))\delta P = 0 \quad (3.39)$$

$$\nabla \cdot (\varphi\vec{\mathbf{F}}) = \varphi\nabla \cdot \vec{\mathbf{F}} + (\vec{\nabla}') \cdot \vec{\mathbf{F}} \rightarrow \varphi\nabla \cdot \vec{\mathbf{F}} = \nabla \cdot (\varphi\vec{\mathbf{F}}) - (\vec{\nabla}') \cdot \vec{\mathbf{F}} \begin{cases} \varphi = \delta P \\ \mu(\vec{\mathbf{T}}_{\mathbf{0}} - \vec{\nabla}\mathbf{P}) \end{cases} \quad (3.40)$$

$$\nabla \cdot (\delta P\mu(\vec{\mathbf{T}}_{\mathbf{0}} - \vec{\nabla}\mathbf{P})) - \delta\vec{\mathbf{P}} \cdot \mu(\vec{\mathbf{T}}_{\mathbf{0}} - \vec{\nabla}\mathbf{P}) = 0 \quad (3.41)$$

$$\int_{\Omega_1} (\nabla \cdot (\delta P\mu(\vec{\mathbf{T}}_{\mathbf{0}} - \vec{\nabla}\mathbf{P})))dV - \int_{\Omega_1} (\delta\vec{\mathbf{P}} \cdot \mu(\vec{\mathbf{T}}_{\mathbf{0}} - \vec{\nabla}\mathbf{P}))dV = 0 \quad (3.42)$$

Gauss theorem (A.13) can be applied in the first volume integral, transforming it in surface integrals:

$$\underbrace{\int_{\Gamma_1} (\delta P\mu(\vec{\mathbf{T}}_{\mathbf{0}} - \vec{\nabla}\mathbf{P}))\vec{\mathbf{n_1}} dS + \int_{\Gamma_{12}} (\delta P\mu(\vec{\mathbf{T}}_{\mathbf{0}} - \vec{\nabla}\mathbf{P}))\vec{\mathbf{n_1}} dS}_{Gauss} - \int_{\Omega_1} (\delta\vec{\mathbf{P}} \cdot \mu(\vec{\mathbf{T}}_{\mathbf{0}} - \vec{\nabla}\mathbf{P}))dV = 0 \quad (3.43)$$

Domain Ω_1, the nonconducting region, is enclosed by two surfaces, Γ_1 and Γ_{12}. Γ_1 is the outer surface of the domain Ω_1, whereas Γ_{12} is the interior surface, the interface with the other domains. The surface integral of the outer surface should vanish, as it does the normal component of the calculated magnetic field according to (3.9). That surface integral along the outer surface of Γ_1, contains the calculation of $\vec{\mathbf{B}}$ in the outer surface, therefore, in consonance with (3.9), the surface integral should vanish. The new equation obtained, with only the surface integral of the interface, is as follows:

$$\int_{\Gamma_{12}} (\delta P\mu(\vec{\mathbf{T}}_{\mathbf{0}} - \vec{\nabla}\mathbf{P}))\vec{\mathbf{n}}_{\mathbf{1}} dS - \int_{\Omega_1} (\delta\vec{\mathbf{P}} \cdot \mu(\vec{\mathbf{T}}_{\mathbf{0}} - \vec{\nabla}\mathbf{P}))dV = 0 \tag{3.44}$$

The boundary condition (3.5) allows to keep developing the equation, and obtain the final form of the governing equation belonging to the weak form:

$$\begin{aligned} -\int_{\Gamma_{12}} (\delta P(\nabla \times \vec{\mathbf{A}}))\vec{\mathbf{n}}_{\mathbf{2}} dS - \int_{\Omega_1} (\delta\vec{\mathbf{P}} \cdot \mu(\vec{\mathbf{T}}_{\mathbf{0}} - \vec{\nabla}\mathbf{P}))dV = 0 \rightarrow \\ \int_{\Gamma_{12}} (\delta P(\nabla \times \vec{\mathbf{A}}))\vec{\mathbf{n}}_{\mathbf{2}} dS + \int_{\Omega_1} (\delta\vec{\mathbf{P}} \cdot \mu(\vec{\mathbf{T}}_{\mathbf{0}} - \vec{\nabla}\mathbf{P}))dV = 0 \end{aligned} \tag{3.45}$$

Finally, all the governing equations belonging to the final weak form implemented in **CalculiX** can be gathered:

$$\int_{\Gamma_{12}} (\delta P(\nabla \times \vec{\mathbf{A}}))\vec{\mathbf{n}}_{\mathbf{2}} dS + \int_{\Omega_1} (\delta\vec{\mathbf{P}} \cdot \mu(\vec{\mathbf{T}}_{\mathbf{0}} - \vec{\nabla}\mathbf{P}))dV = 0 \tag{3.46}$$

$$\begin{aligned} \int_{\Omega_2} (\nabla \times \delta\vec{\mathbf{A}})(\frac{1}{\mu}(\nabla \times \vec{\mathbf{A}}))dV - \int_{\Gamma_{12}} (\delta\vec{\mathbf{A}} \cdot (\vec{\mathbf{T}}_{\mathbf{0}} \times \vec{\mathbf{n}}_{\mathbf{2}}) + \int_{\Gamma_{12}} (P(\nabla \times \delta\vec{\mathbf{A}})\vec{\mathbf{n}}_{\mathbf{2}}) dS_2 \\ + \int_{\Omega_2} \frac{1}{\mu}(\nabla \cdot \vec{\mathbf{A}})(\nabla \cdot \delta\vec{\mathbf{A}})dV + \int_{\Omega_2} \sigma\delta\vec{A}(\frac{\partial\vec{\mathbf{A}}}{\partial t} + \vec{\nabla}\mathbf{V})dV = 0 \end{aligned} \tag{3.47}$$

$$\int_{\Omega_2} (\nabla\delta\vec{\mathbf{V}} \cdot \sigma(\frac{\partial\vec{\mathbf{A}}}{\partial t} + \nabla\vec{\mathbf{V}}))dV = 0 \tag{3.48}$$

This weak form, according to FE theory, is still mathematically equivalent to the strong form, as long as an infinite number of weighting functions are allowed. In fact, the weak form is no approximation, only the correct functions solutions fulfill the weak form exactly.

The weak form equations still deal with infinitely many functions, i.e., the problem dealt is still continuous. Therefore, the number of unknowns must be reduce to a finite number, to a discrete problem, in order to get so an approximate solution. The next step to get the finite element formulation that is implemented in **ccx**, is to perform a discretization of the weak form through the introduction of the shape functions. This discretization aims to get approximate solutions of the equations, which certainly, is the basis of the FEM.

3.4 Discretization; Galerkin's method

The implementation of the weak form in **ccx** goes through its approximate numeric resolution, on that purpose, following the standard FEM a discretization of the weak form is performed; the Bubnov-Galerkin's method is applied by introducing the shape functions both as weighting functions and trial functions (the trial functions are the functions used for expansion of the approximate solution, the ones which are substituted in the primary field). The Bubnov-Galerkin's method known simply as Galerkin's method, considers the same interpolation functions, the shape functions, for both the real functions (trial functions) and the virtual functions (weighting functions). This method is opposed to the Petrov-Galerkin scheme.

Regarding the relation between Galerkin's method and weighted residual method, from previous sections, it has been stated that the weighted residual method minimizes the residual of the dealt partial differential equations (in the present case, the partial differential equations are the potential formulations of the required Maxwell's equations). Indeed, the best approximation for the potentials, chosen as primary variables in this finite element formulation, is obtained when the integrals of the residuals of the partial differential equations multiplied by a weighting function (it is also known as " test function ") over the problem domain is zero. The weighting functions can be arbitrary chosen, but as it was mentioned, Galerkin's method dictates that the same shape functions are selected for both the weighting functions and the trial functions.

It is important to remark that the use of the Galerkin's method for the primary variables (the potentials) and the weighting field, reduces the still continuous problem described by the weak form to a discrete one, i.e., a discrete problem with a countable or " finite " number of unknowns for the primary field. So it is possible to calculate an approximate solution by the FEM.

The interpolation functions or shape functions now introduced are standard. They are constituted from Lagrange polynomials. A detailed description is referred here (DHONDT 2004, P. 67).

Due to the vectorial nature of much of the terms of the integrals constituting the governing equations in its weak form, tensorial calculus is needed to perform the substitution of the shape functions in both the weighting fields and the primary fields (remember the primary fields are compound by the potentials whose solution is being searched) in all domains. As follows, the development of the equations (3.46), (3.47) and (3.48) is showed. In each equation the terms of each integral will be dealt separately in order to make clear the tensorial calculations.

3.4.1 Governing equation evolution

Using standar shape functions , equations (3.46), (3.47) and (3.48) evole to matrices (3.49), (3.50), (3.51), (3.52), (3.53), (3.54), (3.55), (3.56), (3.57) and (3.58) (For deep details, step by step, A.3):

$$[K_{AA}]_{e(iK)(jM)} = \int_{(V_{0e})_{\Omega_2}} [\frac{1}{\mu}\varphi_{i,L}\delta_{KM}\varphi_{j,L} - \varphi_{i,M}\varphi_{i,K} + \varphi_{i,K}\varphi_{j,M}]dV_e \tag{3.49}$$

$$[K_{AP}]_{e(jM)(i)} = \int_{(A_{0e})_{\Gamma_{12}}} \varphi_i\varphi_{j,L}n_{1,K}\epsilon_{KLM}dA_e \tag{3.50}$$

$$[K_{PA}]_{e(i)(jM)} = \int_{(A_{0e})_{\Gamma_{12}}} \varphi_i\epsilon_{KLM}\varphi_{j,L}n_{1K}dA_e \tag{3.51}$$

$$[K_{PP}]_{e(i)(j)} = -\int_{(V_{0e})_{\Omega_1}} \varphi_{i,K}\mu\varphi_{j,K}dV_e \tag{3.52}$$

$$[M_{AA}]_{e(iK)(jM)} = \int_{(V_{0e})_{\Omega_2}} \varphi_i\sigma\varphi_j\delta_{K,M}dV_e \tag{3.53}$$

$$[M_{Av}]_{e(iK)(j)} = \int_{(V_{0e})_{\Omega_2}} \varphi_i\sigma\varphi_{j,K}\delta_{K,M}dV_e \tag{3.54}$$

$$[M_{vA}]_{e(j)(iM)} = \int_{(V_{0e})_{\Omega_2}} \varphi_{j,K}\sigma\varphi_i\delta_{KM}dV_e \tag{3.55}$$

$$[M_{vv}]_{e(i)(j)} = \int_{(V_{0e})_{\Omega_2}} \varphi_{i,K}\sigma\varphi_{i,K}\delta_{KM}dV_e \tag{3.56}$$

$$[F_A]_{e(jM)} = -\int_{(A_{0e})_{\Gamma_{12}}} \epsilon_{KLM}\varphi_j\vec{\mathbf{T}}_{0L}n_{1K}dA_e \tag{3.57}$$

$$[F_p]_{e(i)} = -\int_{(V_{0e})_{\Omega_1}} \varphi_{i,K}\mu\vec{\mathbf{T}}_{0K}dV_e \tag{3.58}$$

4 Goals and procedures

Goal of the thesis The main goal of the present thesis is to verify the correct performance of the module for electromagnetic calculations from **CalculiX** Software (DHONDT 2016a).

CalculiX is an open source FE software that covers most of the practical areas of physics, counting more than twenty different modules. The electromagnetism module is currently in verification tests, among which this thesis is produced. **CalculiX** SW has two packages:

- CalculiX Solver (DHONDT 2016b)
- CalculiX Graphics (WITTIG 2017)

The **CalculiX Solver** package is called **CalculiX CrunchiX (ccx)**, this module has implemented the solution of Maxwell's equations within the framework of the FEM. Such implementation was carried out by Dr. Guido Dhondt. To test the implementation, several numerical simulations of different electromagnetic problems will be performed.

The **CalculiX Graphics** package is called **CalculiX GraphiX (cgx)**, this module has been implemented to generate /display finite elements (FE) and results coming from **ccx**.

Therefore, this thesis aims to constitute a proper benchmark for the electromagnetic module of **CalculiX CrunchiX (ccx)**. Once the module is verified successfully, it can be used to perform trustfully numerical simulations of electromagnetic problems which include coils of complex geometries. These simulations are an important step inside the CAE process to design the coils used in inductive heating.

Procedure The thesis has been divided into two well differentiated parts: one is theory focused and it corresponds with the state of art of the work, where the concepts needed to understand how the simulations work are described. The other part focus in describing the results obtained in the performed simulations. Even though, to relate such results with the benchmarking of the electromagnetic module, continuous references are used to the theory described in this chapters:

- State of the science: Chapter 2

- State of the technology: Chapter 3

The verification process is composed of three parts, which correspond to how **CalculiX** obtain its results. Each part will be tested in a simple case, so that it is possible to compare the results obtained from the simulations performed by **CalculiX** with an analytic solution:

- T0: Chapter 5
- H: Chapter 6
- Work piece: Chapter 7

Looking at **ccx** code, the following comments are of interest to understand the behavior of the program and create strategy to analyze it (those comments are directly extracted from the program B.1.2)

- calculating the initial quasi-static magnetic intensity due to the coil current.
- calculate the current density in the coils: in this section *nload*, *nforc*, *nbody* and *nam* are set to zero. The electrical potential is supposed to be given (in the form of a temperature), the current is calculated (in the form of heat flux) by thermal analogy
- the coil current is assumed to be applied at once, i.e., asstep loading. The calculation, however, is a quasi-static calculation.
- calculates the magnetic intensity due to currents in the phi-domain of an electromagnetic calculation-Variables for multi-threading procedure
- subroutine for multi-threading of *biotsavart* (a technique by which a single set of code can be used by several processors at different stages of execution.)(*biotsavartmt*)
- ASSIGNS THE DOMAIN A NODE BELONGS TO, TO THIS NODE (for electromagnetic calculations, only nodes not belonging to shells)
- CALCULATES THE MAGNETIC INTENSITY due to currents in the phi-domain of an electromagnetic calculation (**biotsavart.f**)
- currents are supposed to be modeled by shell elements
- mapping T0 from the phi domain onto the border of the A and A-V domains
- *** creating the A.n MPC
- identifying the interfaces between the A and A-V domains and the phi-domain

- nodewise storage of the primary variables determination which derived variables have to be calculated
- electromagnetic calculation is linear: should not be taken into account in the convergence check (only thermal part is taken into account)
- the calculation of the electromagnetic fields is (quasi)linear, i.e., the solution of the equations is the fields, only the temperature calculation is nonlinear, i.e., the solution of the equations is a differential temperature
- calculating the electromagnetic fields and temperatures only the temperature calculation is differential
- electromagnetic calculation is linear: should not be taken into account in the convergence check (only thermal part is taken into account)
- CALCULATING T0 ON THE INTERFACE BETWEEN A OR A-V DOMAINS and the P-domain. At the start of the routine h0 is available in the complete phi-domain. Using the MPCs developed in *tiedcontact* to tie the phi-values at the border of the A or A-Vdomains to those in the phidomain, the h0 values are calculated in a similar way
- looking for MPCs tying phi between the A or A-V domains and the phi-domain

Once the FE potential formulation of the Maxwell's equations implemented in **ccx** has been checked as such, i.e., the check of the mathematical development of the FE formulation, including the discretization and the assembly of the matrices; then, the implementation needs to be tested.

To test the implementation means to check if each of the calculations performed by **ccx** for each node are done correctly, within the limits characteristically of the FEM. Therefore, the desired verification is global, it must be tested that **ccx** is able to provide correct results for every assessed point.

Taking advantage that **ccx** is able to perform different type of electromagnetic calculations, it turns up wiser not to start the testing with the Eddy currents case, which is the one of interest in the context of inductive heating, but with a much simpler case, the magneto-statics one. The magneto-statics cases have less solved variables to analyze and so facilitate the testing. Besides, the testing is carried out taking into account the developed FE formulation and the way **ccx** solves the variables.

INTENTIONALLY BLANK PAGE

5 Verification of $\vec{\mathbf{T}}_0$

$\vec{\mathbf{T}}_0$ was introduced under the P-formulation as the impressed current vector potential 2.5.1.1, hence it is considered as a magnetic field intensity type of quantity vector, measured in A/mm. In the context of a FEM software there are different possibilities to calculate it, the one chosen to implement in **ccx** is the application of Biot-Savart's law. That is the most intuitive way and it turns up good for the process of verification, for it allows a comparison between the results provided by **ccx** and an analytical solution according to Biot-Savart's law.

This is the standard approach in FEM software when there is an analytic solution to compare with. However, such standard approach is usually done when the applied FEM software is already tested, and what it really pretends is to test the results of a simulation, comparing them with an analytical solution, so that the meshing can be better adjusted; i.e. the common convergence FEM analysis is done but focusing against the analytical solution. For example, that would be the case in structural mechanics when analyzing a pure bending beam problem, the approximate FEM solution is compared with the analytical ones of Bernoulli or Timoshenko, in order to adjust the meshing.

In spite of the analogies, here it should be borne in mind, that the comparison is made with the main purpose of testing the implementation of $\vec{\mathbf{T}}_0$ in **ccx**. On the other hand, taking advantage of the procedure, the meshing will be analyzed too. For this reason, in the current section like in the others belonging to the verification process, occasionally the testing intentions will get mixed up with standard FEM analysis.

5.1 Objectives

The objectives pursued with the current section can be summarized in the following:

- **Verification of $\vec{\mathbf{T}}_0$**: the implementation of $\vec{\mathbf{T}}_0$ in **ccx** will be tested. In order to get a successful verification, the results provided by **ccx** for $\vec{\mathbf{T}}_0$ must be close to the solution provided by the analytical Biot-Savart's law. The sought verification is global, i.e. it should be verified that **ccx** is able to calculate properly $\vec{\mathbf{T}}_0$ for any possible assessed point.

- **Meshing of the coil**: as it was already stated, taking advantage of the verification process, a standard meshing of the coil, or standard tips for a good meshing of the coil are sought too. This way, not only a meshing for the coil can be advised, but the found suitable meshing can be used in the subsequent simulations of the other parts of the verification process. It has been studied which element type and which element size result more advantageous, and if its performance results plausible.
- **Convergence analysis**: to carry out a convergence analysis is part of a standard FEM work-flow, here it has been performed to identify which factors have bigger influence in the difference between the analytic solution and the values provided by **ccx** for $\vec{\mathbf{T}}_0$.

The procedure to accomplish such objectives follows a work-flow that all parts of the verification share. First, the situation is analyzed to conceive a FE model susceptible of being used in successive simulations. Then, maintaining the bases of the FE model but varying some characteristics (the meshing or parameters of the input deck C.1), different simulations are carried out and their results analyzed in Matlab.

Again, it should be remembered that due to the duality verification-convergence of the results, ubiquitous in this thesis, many times the described objectives of the section are exposed together without making a clear difference between them, as many times they are achieved with the same analysis; for instance, a converge analysis which gathers the results of $\vec{\mathbf{T}}_0$ from different simulations, can be used to verify $\vec{\mathbf{T}}_0$ itself, and to obtain a suitable meshing for the coil.

5.2 Calculation of $\vec{\mathrm{T}}_0$ by ccx

To design a suitable simulation for testing $\vec{\mathbf{T}}_0$, first it is useful to remember how exactly **ccx** calculates it.

It has been said that **ccx** calculates $\vec{\mathbf{T}}_0$ through the Biot-Savart's law. On the other hand, **ccx** requires that the source of the magnetic field, the coil, must be meshed with shell elements. **ccx** makes use of the Biot-Savart's law to integrate the contributions of the current flowing through such shell elements, and so, get the magnetic intensity vector. $\vec{\mathbf{T}}_0$ is calculated directly in the nodes of the air domain or domain Ω_1 (DHONDT 2016b), as everything surrounding the coil was free space ($\mu = \mu_0$); if there is a work piece (a different domain), it is not considered, i.e. $\vec{\mathbf{T}}_0$ is not calculated in the nodes of the space occupied by the work piece. It should be remembered that each domain has its own meshing and corresponds to a region of the electromagnetic problems analyzed in this thesis.

Hence, the calculation is analogous to that typical of the application of Biot-Savart's law in theoretical problems, such as the analytic calculation of a magnetic field in the vicinity of a straight wire of finite length, along which an electric current flows. The difference lies in how the integrals are treated. By **ccx**, the integrals of the source current must be solved through numerical integration. In theoretical problems, it is common that those integrals can be solved analytically.

The boundary conditions does not affect the calculation of $\vec{\mathbf{T}}_0$, again, it is calculated in free space. However, it should be added that it is calculated too on the external faces of the other domains in contact with domain Ω_1(DHONDT 2016b, P. 254).

The numerical integration is costly in terms of computational time, thus these calculations are parallelized when **ccx** is executed(DHONDT 2016b, P. 254).

To sum up, it should be considered that $\vec{\mathbf{T}}_0$ is calculated in the nodes of the air domain, through numerical integration applied to the integrals defined by Biot-Savart's law, and such integration is performed along the shell elements of the coil.

5.3 Magneto-static problem; benchmark problem

CalculiX is able to face a broad series of electromagnetic problems: electrostatic, magneto-static and magnetic induction (Eddy current fields). Between them, the magneto-statics are the ones of interest in this section. The reason is how **ccx** calculates $\vec{\mathbf{T}}_0$, as it was explained above, when calculating it, only the domain of the air is considered. Thus the calculation of $\vec{\mathbf{T}}_0$ itself resembles a normal magneto-static problem, in which knowing the source electric current, the magnetic field is calculated spatially; regardless of if the faced problem is for instance a magnetic induction one, **ccx** performs the calculation of $\vec{\mathbf{T}}_0$ as a magneto-static case.

Therefore, it makes sense to test the implementation of $\vec{\mathbf{T}}_0$ with simulations in which the problem faced is a magneto-static one. The idea is to test that **ccx** is able to calculate $\vec{\mathbf{T}}_0$ properly in every node of the meshing of domain Ω_1.

In magneto-static problems the calculated magnetic field do not vary in time, as consequence of the source current of the magnetic field, which is DC current, and which neither varies in time. In the performed simulations for $\vec{\mathbf{T}}_0$, the current that feeds the coil is always DC.

5.4 Conception of the FE models

The FE models built up for $\vec{\mathbf{T}}_0$ are based on the original examples for the electromagnetic module, which come along with the installation of **CalculiX** (*induction.inp*). The models are built with the sake of simplicity, the focus is on the post-processing part, where the results are analyzed in Matlab and it is actually when the verification can be accomplished. The models follow a general design, between them only varies the meshing, and occasionally the size of the built geometries, although their shape is always the same. They look like this:

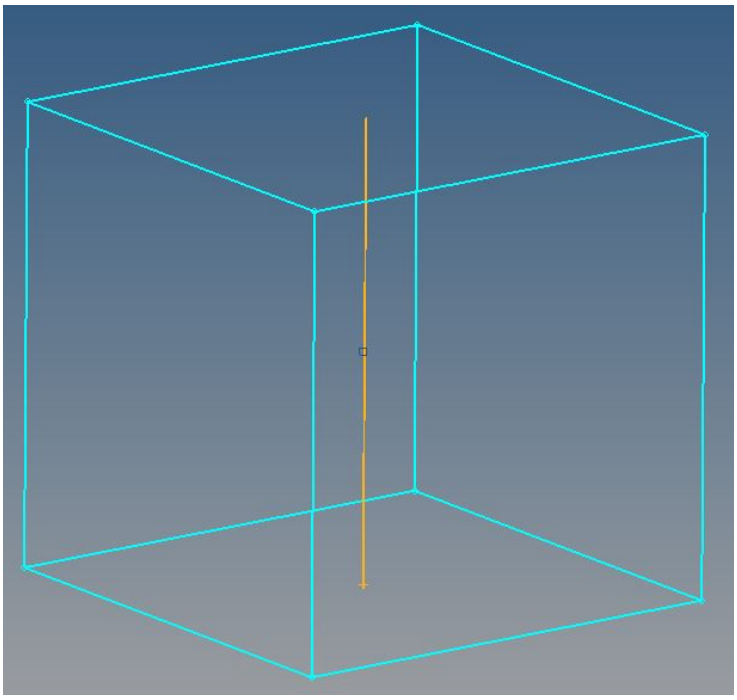
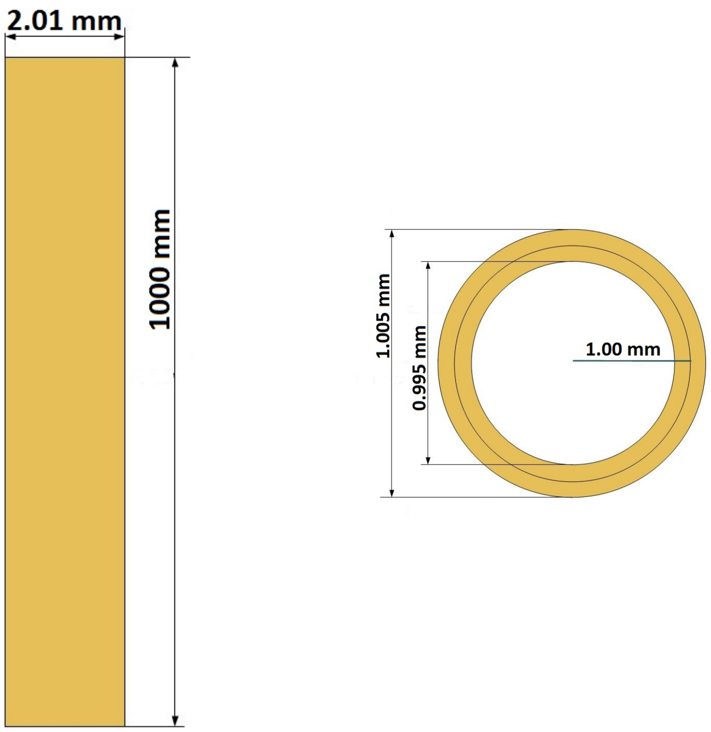


Image 5.1: FE model

In the language of Hypermesh (HYPERMESH 2016) and (ALTAIR 2015), each of the models have two components:

- **Coil**: it is modeled as a straight wire of finite length, so it not exactly a coil, but it will still be called like that to keep the same language for referring to the components of all simulations during the thesis. In Hypermesh, the geometry used to model it, is a cylindrical surface, which is logic as it must be meshed with shell elements. The dimensions of the coil are the same in all models of this $\vec{\mathbf{T}}_0$ verification part, only the meshing changes. The measures of the coil were taken from the original example of **CalculiX**, 1mm as radius of the cylinder and 1000 mm long. Spatially, the coil is located along the Z axis, being symmetric distributed respect to the XY plane. The origin of coordinates can be defined by Hypermesh.

- **Air**: corresponding to domain Ω_1, the air is modeled as a cube which surrounds the coil. The geometry used is a cube volume, which is meshed by volume elements. Similar to the coil, the initial measures of the cube were taken from the original example from **CalculiX**, a cube of 1000 mm of each side. The geometry of the cube is built in a way that the center of it matches the origin of coordinates. Thus the coil crosses the cube along the Z axis. Due to the process of verification, not only the meshing is changed, but the size of the cube too. Spatially, the position of the components is maintained despite a change in cube's size, the coil is always left as it was above described, and the cube is expanded from its center, maintaining it at the origin of coordinates.



(a) Wire's length

(b) Cross section area

Image 5.2: Measures of the Coil

When changing the cube's size, only an expansion can take place, for the size of the coil always remains equal, and it can not be that the environment surrounding the coil, the air, is smaller than the coil itself.

Both the coil and domain Ω_1 have their own meshing. The meshes are considered independently of each other, and there is no necessity of coincident meshes. It is not important that domain Ω_1 and the coil are in contact, in fact, domain Ω_1 contains the coil. As it was stated in the objectives of the current section, the focus is on the meshing of the coil. It is primordially analyzed.

Once the meshing is done, in Hypermesh set of nodes and elements are created. Usually, they are the following in all FE models:

- **Nall**: set of all nodes existing in the FE model.
- **Ncoil**: it is made up of all the nodes belonging the meshing of the coil.
- **Nair**: it is made up of all the nodes belonging the air.
- **Eall**: set of all elements existing in the FE model.
- **Ecoil**: set of elements belonging the coil.
- **Eair**: set of elements belonging the air.
- **Nplus**: it is made up of the nodes located on the upper edge of the coil.
- **Nminus**:it is made up of the nodes located on the lower edge of the coil.
- **Setx**: it is made up of nodes belonging to the air and located along the X axis.
- **Sety**: it is made up of nodes belonging to the air and located along the Y axis.

The name of the sets are depicted as they were normally designated in Hypermesh. The purpose of creating them is to allow the specification of constraints and parameters at the input deck, through such sets.

5.5 Analytic solution: Biot-Savart's law

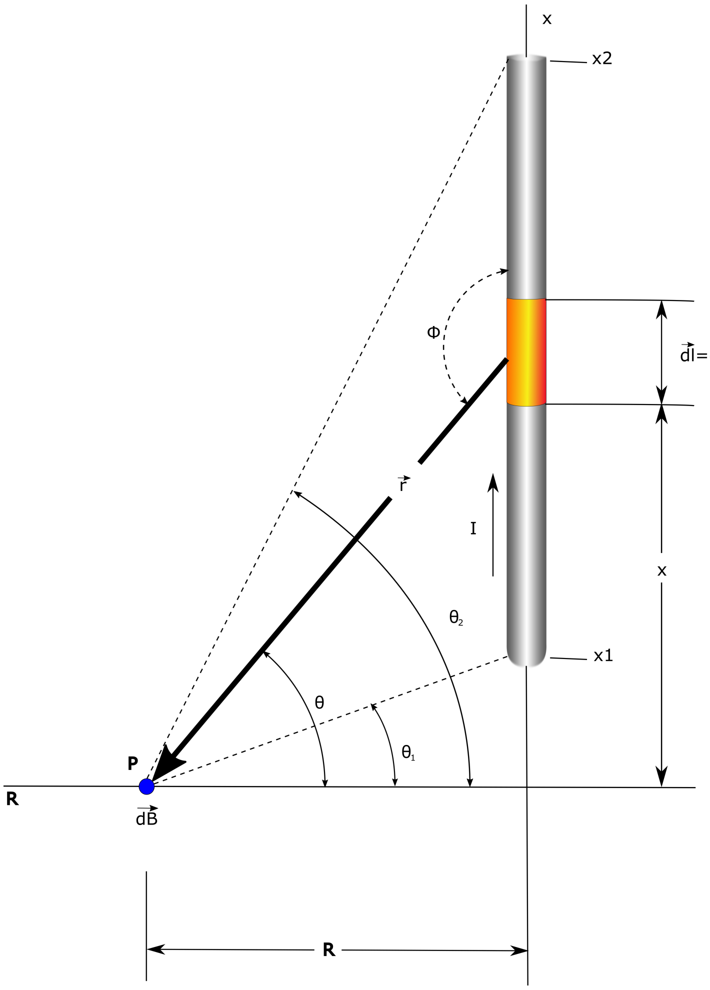


*Image 5.3: Biot-Savart's law for a finite straight filament(image based on (*TIPLER *2010, P. 927))*

$$d\vec{\mathbf{B}} = \frac{\mu_0}{4\pi} \cdot \frac{Id\vec{\mathbf{l}} \times \vec{\mathbf{r}}}{r^2} \tag{5.1}$$

$$dB = \frac{\mu_0}{4\pi} \cdot \frac{I \cdot dl \cdot sen(\phi)}{r^2} = \frac{\mu_0}{4\pi} \frac{Idl \cdot cos(\theta)}{r^2} \tag{5.2}$$

$$tg(\theta) = \frac{x}{R} \rightarrow x = Rtg(\theta) \quad \vec{\mathbf{dl}} = dx\vec{\mathbf{i}} \tag{5.3}$$

$$dx = R \cdot \frac{1}{cos^2(\theta)} d\theta = R \cdot \frac{1}{\left[\frac{R}{r}\right]^2} d\theta = R \frac{r^2}{R^2} d\theta = \frac{r^2}{R} d\theta \tag{5.4}$$

$$dB = \frac{\mu_0}{4\pi} I \frac{r^2 d\theta}{R r^2} \cdot cos(\theta) = \frac{\mu_0}{4\pi} \frac{I d\theta cos(\theta)}{R} \tag{5.5}$$

$$B = \int_{\theta=\theta_1}^{\theta=\theta_2} \frac{\mu_0}{4\pi} \frac{I cos(\theta)}{R} d\theta = \frac{\mu_0}{4\pi} \cdot \frac{I}{R} \int_{\theta_1}^{\theta_2} cos(\theta) d\theta \tag{5.6}$$

$$B = \frac{\mu_0}{4\pi} \cdot \frac{I}{R} \left[sen(\theta)\right]_{\theta_1}^{\theta_2} \tag{5.7}$$

$$B = \frac{\mu_0}{4\pi} \cdot \frac{I}{R} (sen(\theta_2) - sen(\theta_1)) \qquad \textbf{Biot-Savart Law} \tag{5.8}$$

This equation has been implemented in Matlab C.1.6 to calculate analytically T_0, and representing results graphically as image 5.4.

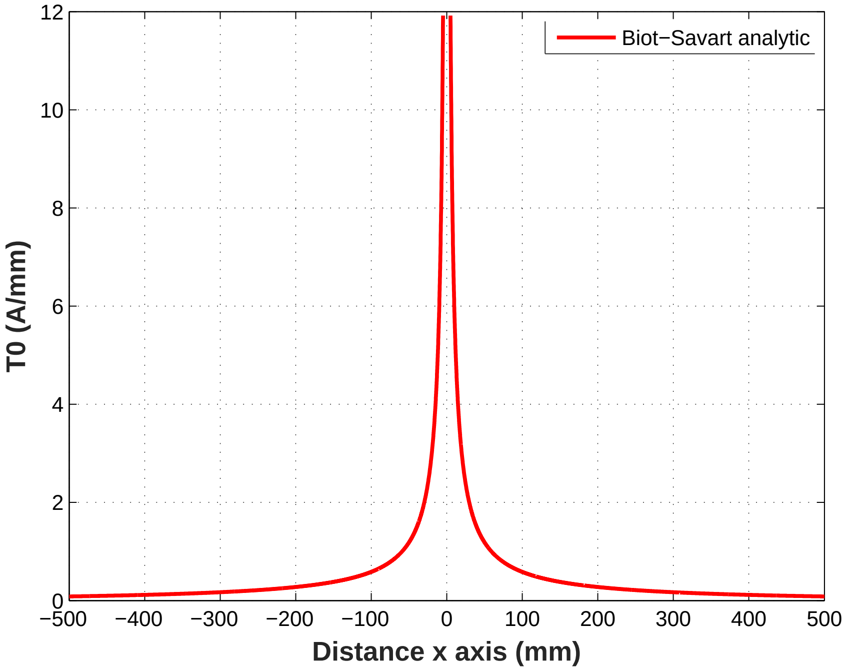


Image 5.4: Biot-Savart analytic

5.6 Feeding of the coil

$\vec{\mathbf{T}}_{\mathbf{0}analytic}$, that will be defined later on, is calculated according to two assumptions that might seem contradictory at first glance: Electric current can be approximated analytically as running through an infinitely narrow round wire, also known as filament. This is required to calculate the magnetic field analytically through the version of Biot-Savart's law explained previously. In spite of its unidimensional nature, the filament is required to have a cross section area. In this case, the $\vec{\mathbf{T}}_{\mathbf{0}analytic}$ of equation (5.9) is calculated with Biot-Savart's law bearing in mind such supposition.

The hypothesis above described for the electric current is not entirely considered, instead in the analytical calculations of $\vec{\mathbf{T}}_{\mathbf{0}analytic}$, the analytical electric current is considered equivalent to the one running through an infinitely narrow round wire, but a wire which has the same cross section area as the wire of the FE models.

5.7 Convergence analysis for meshing of the coil

The first step towards the verification of $\vec{\mathbf{T}}_\mathbf{0}$ is to learn how to mesh the coil properly. As it has been stated, the calculation of $\vec{\mathbf{T}}_\mathbf{0}$ is done through a numerical integration which adds up all the contributions of the electric current flowing along the coil. Thus, it is important that the meshing of the coil is suitable done, so that all those contributions are well considered; otherwise the verification turns out difficult, as it is hard to say whether there is an error in the way of meshing, or actually, it is a problem on how the software **ccx** calculates the results.

Nevertheless, the way to analyze the meshing goes through $\vec{\mathbf{T}}_\mathbf{0}$ itself. As it happens in a normal FEM work-flow, to improve a meshing, a repetition of simulations is performed till the analytic solution, or convergence of the results (in case there is no analytic solution) is reached. In the current case, there is an analytic solution according to Biot Savart's law to compare with $\vec{\mathbf{T}}_\mathbf{0}$ obtained values. Hence, this analysis of the coil's meshing is included in the testing, as much of the calculations and analysis performed help out to verify $\vec{\mathbf{T}}_\mathbf{0}$ too.

As it has been mentioned, **ccx** requires the coil to be meshed with shell elements. The available options in **CalculiX** match the shell elements existing by Abaqus: S4R, S4, S8R and S8. The " S "stands for shell elements and the number indicates the amount of nodes contained in each shell element. The " R"indicates reduced integration, which means less Gauss or integration points in the integration scheme.

Images 5.5 and 5.6 represent first order S4 elements and second order S8 elements. By **CalculiX**, they are known as qu4 and qu8, respectively.

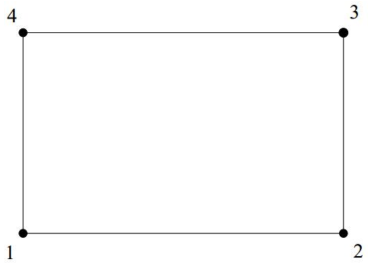


Image 5.5: S4 shell element (qu4, type 9)

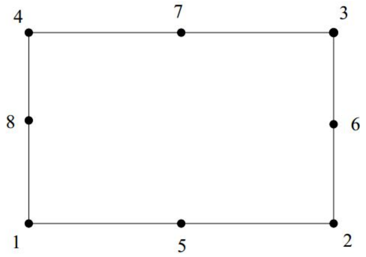


Image 5.6: S8 shell element (qu8, type 10)

Second order S8 elements make use of quadratic shape functions, in opposition to the S4 elements, which use linear shape functions.

The calculation of $\vec{\mathbf{T_0}}$ depends entirely on the meshing of the coil. The numeric integration which provides the value of $\vec{\mathbf{T_0}}$ in the nodes of the air domain, will be as much accurate as the discretization of the geometry of the coil is (the meshing of the coil). Due to such integration, the meshing of the coil must be regular and homogeneous, there can not be elements of different sizes or shapes; otherwise, it could occur that a particular part of the coil contributes more than others to the integration, and the accuracy would be harmed.

The meshing is performed in Hypermesh. It is possible to use the " auto-mesh " option to get an homogeneous meshing with squared shell elements (all elements of equal element size and squared), however, it is been sought the best way of meshing for the coil, considering all possible combinations and influences in the calculation of $\vec{\mathbf{T_0}}$. Therefore, it was decided to split up the meshing analysis in three parts according to three defined meshing parameters:

- Analysis of circumference's length divisions of the coil/wire: parameter of the mesh which sets with how many element divisions is built the circumference of the cross section of the wire.

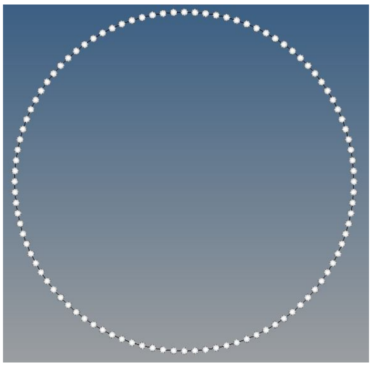


Image 5.7: Circumference's length divisions

- Analysis of drag divisions of the coil/wire: parameter of the mesh which sets how many elements there are along the length of the wire.

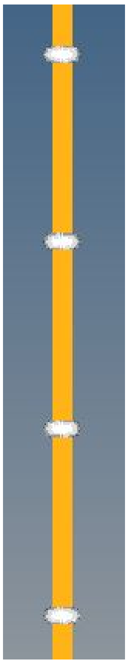


Image 5.8: Drag divisions

- Analysis of the element type: S4R, S4, S8R or S8. It should be mentioned that S8 is not supported by version 13 of Hypermesh, but it is supported by **CalculiX**.

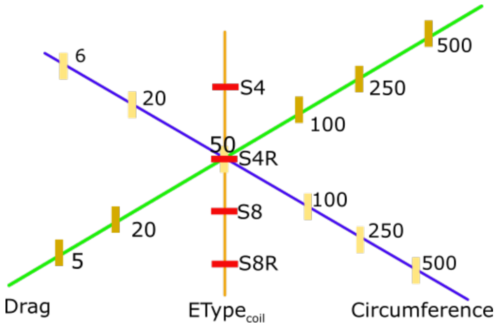


Image 5.9: Graphical summary Convergence analysis

Usually, having a regular and homogeneous meshing, i.e. with all elements with the same size and shape, there are two parameters to adjust between different attempts of simulations: the element size and the element type. Looking at the three parameters defined above, circumference's length divisions and drag divisions relate to the element size, while the one left is the element type itself. It is done like this because the meshing must be maintained uniform and regular, with no elements of different shapes or sizes, but it must be analyzed separately the influence of the number of element divisions in the cross section of the wire, and along the length of the wire; these are the two ways to vary the element size keeping at the same time the needed uniform mesh. The elements will not be squared in shape, but rectangular.

To analyze the meshing of the Air is not targeted in the actual section, thus the meshing of the air domain will be kept as it is in the example of **CalculiX** " induction.inp ". By such example the elements used are volume elements known in Abaqus as C3D8 (Continium 3D-4 node elements). These brick elements are named in **CalculiX** he8 and they look like this:

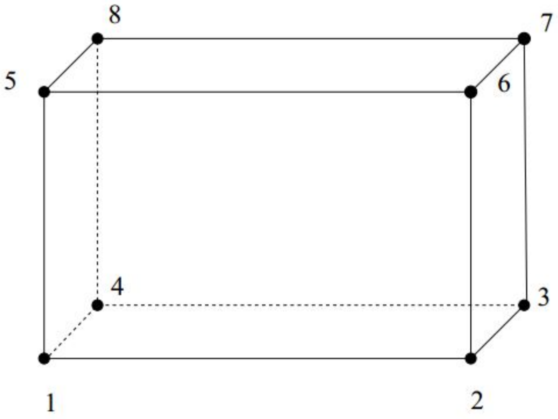


Image 5.10: C3D8 brick element (he8, type 1)

A convergence analysis has been carried out for each of the defined mesh parameters for the coil. In each analysis a mesh parameter is altered while the other two remain equal. So it is possible to check individually the influence of each of the mesh parameters in the meshing and in the calculation of $\vec{\mathbf{T}}_0$.

The desired verification of $\vec{\mathbf{T}}_0$ is global, which means that $\vec{\mathbf{T}}_0$ must be analyzed in each of the nodes where is calculated, the nodes of the air domain:

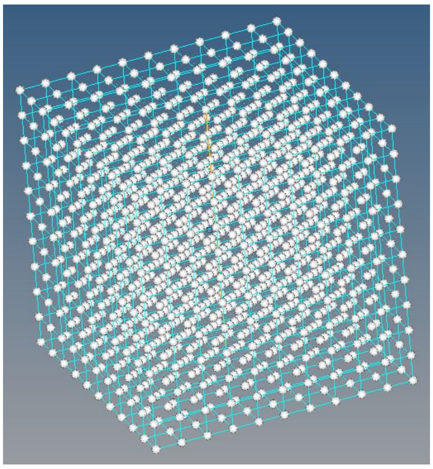


Image 5.11: Nodes air domain

This is difficult to depict in a convergence analysis, which tend to be local (limited to a node or a small number of nodes) in typical FEM analysis. In order to keep the global approach, the convergence analyses are performed through mean values that reflect the information of all nodes. A version of the analytical Biot-Savart's law was implemented as a Matlab function, capable of calculating the magnetic field's modulus anywhere from given Cartesian coordinates. For each node, the value of $\vec{\mathbf{T}}_0$ delivered by **ccx** is compared with the analytic one in Matlab, recording the differences in every node of the air domain. To be able to analyze such quantity of data during the thesis, several graphical approaches are used. In the current convergence analyses, the differences between the FE calculated values and the analytical ones, are represented in convergence graphs through mean values, in particular, through the mean relative deviation.

The definition of the relative deviation in this section for $\vec{\mathbf{T}}_0$, regarding the analytical value by Biot-Savart, is as follows:

$$Relative\ \ deviation(T_0) = \frac{|T_{0_{ccx}} - T_{0_{analytic}}|}{T_{0_{analytic}}} \tag{5.9}$$

The mean of the relative deviations of each node is used in the next convergence graphs, as a first approach to display the information obtained from the comparison between the solutions provided by **ccx** and the analytical solutions. It is clear that a mean quantity can lose information because of the data distribution (for instance, much higher differences between the solutions in the vicinity of the coil could rise the total mean value), but on the other hand, it is a measure of the total relative deviation that can be extracted subsequently from each performed simulation to constitute a convergence graph. Definitely, it is a quicker way to analyze the meshing of the coil and find a suitable one, instead of analyzing straightforward all the available data, without having secured the coil's meshing.

Aside from the mean relative deviation, the mean absolute deviation is calculated as consequence (numerator of the relative deviation), and the maximum values of both the relative deviation and the absolute deviation are calculated too. These additional calculations are not represented in the graphs of the present section, but they can be found in Appendix D.1. They were not added to the section as they only reinforce the conclusions obtained from the analysis of the mean relative deviation.

5.7.1 Coil circumference's length analysis

In this convergence analysis, the number of elements in the cross section of the wire is modified whereas the other two mesh parameters, the drag divisions and the element type, remain equal. The table 5.1 gathers the meshing information regarding each of the simulations performed:

$EType_{Air}$	$ESize_{Air}(cm)$	$EType_{Coil}$	Circumference	Drag	$I_{Analytic}(A)$	$I_{FE}(A)$
C3D8	10	S4R	6	50	374.47784	357.60000
C3D8	10	S4R	20	50	374.47784	372.97680
C3D8	10	S4R	50	50	374.47784	374.28800
C3D8	10	S4R	100	50	374.47784	374.40720
C3D8	10	S4R	250	50	374.47784	374.43700
C3D8	10	S4R	500	50	374.47784	374.58600

Table 5.1: Circumference's length analysis simulations; meshing data.

"EType " stands for element type and " ESize " for element size. The meshing of the air domain is regular and homogeneous, all C3D8 elements have the same size and shape. It is so for all performed simulations in this analysis.

As it was explained before, for the meshing of the coil, the element size parameter is managed through two others: the number of element divisions in the cross section of the wire (circumference divisions), and the number of element divisions along the length of the wire (drag divisions). Here only the circumference divisions are modified.

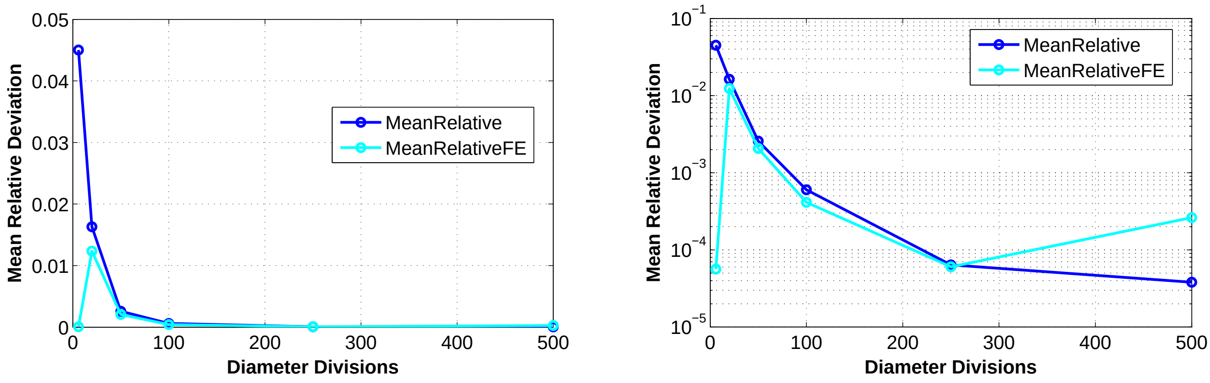
Lastly, in the table, two electric currents are incorporated, the analytic electric current, and the so called I_{FE}, which stands for the current that actually goes through the meshing of the wire from the FE model, that as it can be seen in the table, does not match the analytic one.

The circumference's length divisions set the number of elements present in the cross section of the wire. The number is important because it determines the quality of the discretization of the wire's geometry. The more divisions, the better, because a high number of divisions (thus a high number of elements in the cross section) assures that the meshing matches the round shape of the wire's cross section. Shell elements are being used, and they do not count with a round shape, hence, the approximation of the built meshing to the perfect shape of a round circle, is better as long as there are an enough number of elements which have a small side in the wire's cross section.

The convenient discretization of the wire's cross section affects directly the calculation of $\vec{\mathbf{T}}_0$ because of the magnetic field's source, the electric current. **ccx** performs the numerical integration to calculate $\vec{\mathbf{T}}_0$ adding up all the contributions of each element belonging to the wire's mesh. The contributions depend on the electric current density of each element. The electric current density is defined by the electric potential, the electrical conductivity (both introduced in the input deck) and the very geometry of the coil (wire in the actual case). Therefore, the accuracy in the calculation of $\vec{\mathbf{T}}_0$ is also directly related to the convenient calculation of the electric current by **ccx**. To get an electric current in the simulation, similar to the one that would flow through a real round wire, the meshing of the cross section must approach the perfect round shape of the wire. This issue is regulated by the mesh parameter " circumference's length divisions ", and it is illustrated in the convergence graphs of the actual section.

Here arises again one of the persistent difficulties of this thesis, to wonder about, if this is more of a problem related with the user (who should be able to carry out a suitable meshing), or it is a problem in the calculations of **ccx**. The verification of $\vec{\mathbf{T}}_0$ seeks to demonstrate the first affirmation.

The group of images 5.12 set out two graphs which represent the very same convergence analysis, just the one on the left has a linear scale in the Y axis, whereas the one on the right has a logarithmic scale to improve the view. The X axis represents the number of circumference's length divisions, the defined mesh parameter; the Y axis represents the mean value of all relative deviations from every node of the air domain. The relative deviation is defined between the analytic Biot-Savart's value and the value given by **ccx** (5.9).



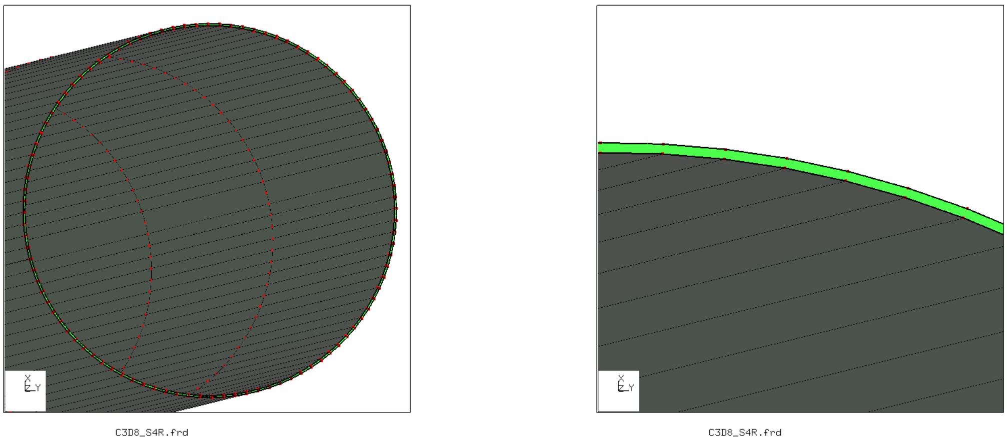
(a) Linear Y axis

(b) Logarithmic Y axis

Image 5.12: Convergence analysis; evaluation of circumference's length divisions

Each dot of the graphs represents the mean relative deviation of a performed simulation. The simulations differ between each other only in the circumference's length divisions, which is incremented gradually in each simulation. All the information regarding the meshing of each simulation is found in table 5.1. In the graphs there are depicted two kinds of relative deviations, one called in the legend simply " MeanRelative " and the other one " MeanRelativeFE ". They differ in how $\vec{T}_{0analytic}$ (the analytic solution) of equation (5.9) is calculated, thus obtaining different relative deviations values.

To calculate $\vec{T}_{0analytic}$, the version of Biot-Savart's law previously explained is used. In such calculation the value of the electric current through the wire is required. **CalculiX** does not provide the electric current as a result, instead it provides a FEM-fashion solution, the electric current density $\vec{J}$ in each element. The problem is that in the calculation of $\vec{J}$(A/mm) influences the cross section area of the wire, and thus the accuracy with it is meshed.



(a) Zoom out

(b) Zoom in

Image 5.13: Meshed Cross section area

Therefore, how the wire's cross section area is considered analytically affects the calculation of $\vec{\mathbf{T}}_{0\,analytic}$ and consequently the calculated relative deviations too. The differentiation between " MeanRelative " and " MeanRelativeFE " just seeks to point out this issue:

- **" MeanRelative "** : this option considers the wire's cross section area ideally rounded and calculates analytically the electric current through the wire, from the given $\vec{\mathbf{J}}$ (A/mm) by **CalculiX** and the ideal cross section area. So indirectly, this option considers that the meshing of the cross section area is good enough and ignores any error derived from the discretization of the geometry.

$$R = 1\ \ mm \quad Thick_{shell} = 0.01\ \ mm \tag{5.10}$$

$$R_1 = R - \frac{Thick_{shell}}{2} \tag{5.11}$$

$$R_2 = R + \frac{Thick_{shell}}{2} \tag{5.12}$$

$$Area_{analytic} = \pi \cdot R_2^2 - \pi \cdot R_1^2 \tag{5.13}$$

$$I_{analytic} = \vec{\mathbf{J}}_{ccx} \cdot Area_{analytic} \tag{5.14}$$

- **" MeanRelativeFE "** : this option considers the wire's cross section area as it is really built in the FE model, i.e. it considers the meshed cross section, not the analytical equivalent that the meshing seeks to approach. Such considerations has been named after " FE " during the thesis. $Area_{FE}$

$$Area_{FE} = Divisions_{Circumference} \cdot Element_{size} \cdot Thick_{shell} \tag{5.15}$$

$$I_{FE} = \vec{\mathbf{J}}_{ccx} \cdot Area_{FE} \tag{5.16}$$

Both of the options consider when calculating analytically the solution for the magnetic field, that the electric current can be approximated as running through an infinitely narrow wire, a filament (in order to able able to use the Biot-Savart's expression); and at the same time, the filament in spite of its unidimensional nature, is considered to have a cross section area, so it is possible to calculate the electric current from the given results by CalculiX, $\vec{\mathbf{J}}_{ccx}$.

To sum up, " MeanRelative " represents the comparison with the pure analytic solution, while " MeanRelativeFE " considers besides the particularizations of the FE model.

Comming back to images 5.12 Now the graphs are analyzed according to the differentiation performed before. The convergence analysis shows successful, for the values of the means relative deviations converge when the circumference's length divisions is incremented. In a convergence analysis, that is indication of having found a suitable meshing, with which the simulation provides accurate results. It should be remembered that the quantity analyzed is a mean relative deviation respect to the analytic solution, so the desired convergence it is at a value the most possible close to zero, which means perfect accuracy, perfect convergence with the analytic solution.

The “ MeanRelative ” values converge at 0.000601 approximately ($\sim$ 0.06%) at 100 circumference's length divisions. From 100 divisions the mean relative deviation keep shrinking but at much lower rate. This is graphically well seen at the plot with the linear scale, and in detail at the plot with the logarithmic scale. It is not by chance, that the simulations of $\vec{\mathbf{T}}_0$ get better when the number of elements in the cross section area is incremented. The more elements, the more close is the meshing to the real round shape of the wire and consequently, the more close are the results of the numerical integration by **ccx** to the analytic solution. Therefore, it can be extracted as a conclusion for the coils, that the round shapes begin to be well meshed from 100 divisions at the cross area section.

Regarding the “ MeanRelativeFE ” values, they follow the “ MeanRelative ” ones but being always more accurate. The exceptions are at the beginning with 6 divisions, where the “ MeanRelativeFE ” value does not follow the “ MeanRelative ” one, and at the end with 500 divisions, where the “ MeanRelative ” value is more accurate than the “ MeanRelativeFE ” value. The higher accuracy in general of “ MeanRelativeFE ” values respect to “ MeanRelative ” ones, is due to how this option calculates the analytic solution. “ MeanRelativeFE ” considers the cross section area to calculate analytically the electrical current, as it is in the FE model when it is meshed. Therefore, it approximates better the current density $\vec{\mathbf{J}}_{ccx}$ that **CalculiX** is using to calculate $\vec{\mathbf{T}}_0$. Because **ccx** calculates $\vec{\mathbf{T}}_0$ according to Biot-Savart, but with the volumetric conductor expression (more realistic than the filament conductor expression), which requires the electric current density, not the electric current. So “ MeanRelativeFE ” has always a smaller relative deviation independently of how good is the meshing, because its comparison is made with an analytic solution that considers the discretization of the geometry.

Particularly curious are the cases of the first simulation and the last one. In the first simulation with 6 divisions, an hexagonal meshing, the “ MeanRelativeFE ” option is where best approximates

5.7.2 Coil's drag analysis

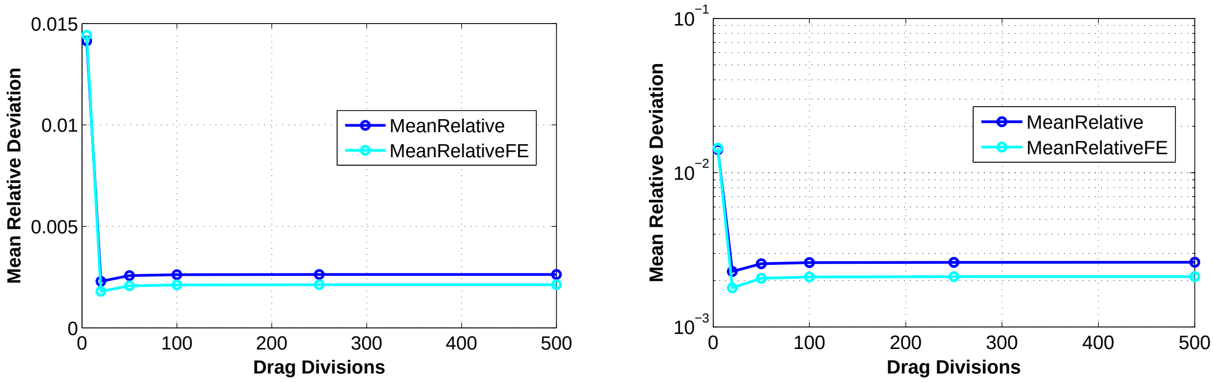
In this convergence analysis, the number of elements along the wire's length (Z axis direction) is modified, whereas the other two mesh parameters, the circumference divisions and the element type, remain equal. Table 5.2 gathers the meshing information regarding each of the simulations performed:

$EType_{Air}$	$ESize_{Air}(cm)$	$EType_{Coil}$	Circumference	Drag	$I_{Analytic}(A)$	$I_{FE}(A)$
C3D8	10	S4R	50	5	374.47784	374.28800
C3D8	10	S4R	50	20	374.47784	374.28800
C3D8	10	S4R	50	50	374.47784	374.28800
C3D8	10	S4R	50	100	374.47784	374.28800
C3D8	10	S4R	50	250	374.47784	374.28800
C3D8	10	S4R	50	500	374.47784	374.28800

Table 5.2: Drag analysis simulations; meshing data.

This time, both analytically defined electrical currents are equal in each simulation because the number of circumference divisions remains untouched. The drag divisions do not affect the discretization of the cross section area, instead they set the number of divisions lengthwise at the wire, which it does affect the integral calculation of $\vec{\mathbf{T}}_0$ by **ccx**. Again, here the purpose is to demonstrate through a convergence analysis that $\vec{\mathbf{T}}_0$ can be well calculated by **ccx** when a suitable meshing is used, and that there is no problem with the software.

The group of graphs 5.14 set out the convergence analysis. As before, to get a better view a linear scale and a logarithmic scale have been included. Like in all graphs concerning $\vec{\mathbf{T}}_0$, the " MeanRelative " and " MeanRelativeFE " differentiation is included to illustrate the permanent issue about the discretization of the cross section area and its consideration by the analytical solution. Even thought, the analysis is focused in the " MeanRelative " values, which make the comparison with the pure analytic solution.



(a) Linear Y axis

(b) Logarithmic Y axis

Image 5.14: Convergence analysis; evaluation of drag divisions

Comment on image 5.14 In each simulation the drag divisions is incremented to see how reacts the mean relative deviation. From 50 divisions, the mean relative deviation converges at 0.00257 approximately ($\sim$ 0.257%). Thereafter, the increment of the drag divisions does not translate in gains of accuracy. On the contrary to the circumference's length divisions, here a high number of drag divisions is not required, between 20 and 50 is enough to calculate accurately $\vec{\mathbf{T}}_{\mathbf{0}}$.

5.7.3 Coil's Element type analysis

In this last convergence analysis it is contemplated the influence of the element type from the shell elements belonging to the coil/wire. The other two mesh parameters, the circumference divisions and the drag divisions, remain equal. The table 5.3 gathers the meshing information regarding each of the simulations performed:

$EType_{Air}$	$ESize_{Air}(cm)$	$EType_{Coil}$	Circumference	Drag	$I_{Analytic}(A)$	$I_{FE}(A)$
C3D8	10	S4R	50	50	374.47784	374.28800
C3D8	10	S4	50	50	374.47784	374.28800
C3D8	10	S8R	50	50	374.47784	374.28800
C3D8	10	S8	50	50	374.47784	374.28800

Table 5.3: Drag analysis simulations; meshing data.

Again, as the circumference divisions remains untouched in all simulations, both analytical electric currents stay equal. Additionally, it should be pointed out how the comparison between first order elements, S4R and S4, and second order elements, S8R and S8, is made. In Hypermesh there is an option to perform a change from first order elements to second order elements. When performing that change it is important to consider two details:

- Half number of divisions at the cross section area: the second order elements such as S8R and S8 make use of quadratic shape functions. In order to do that it is necessary to have more nodes in each side of the element, which can be checked at images 5.5 and 5.6. Therefore, if the comparison is to be made with a simulation with first order elements and 50 divisions at the cross section area, 25 divisions must be set before making the change to second order elements. So the number of elements at the cross section remains at 25, but due to the mid-side nodes of the second order elements, the number of divisions at the cross section turn to 50. This issue can be observed in the group of images 5.15, 5.16 and 5.17.

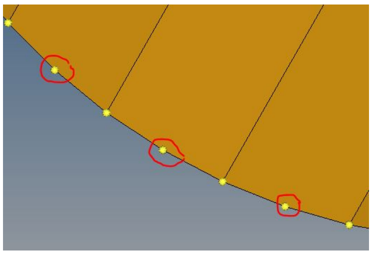


Image 5.15: Mid-side nodes at the cross section area I

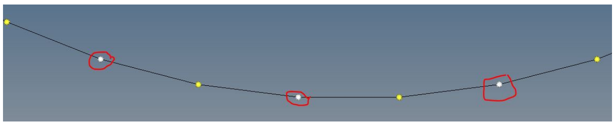


Image 5.16: Mid-side nodes at the cross section area II

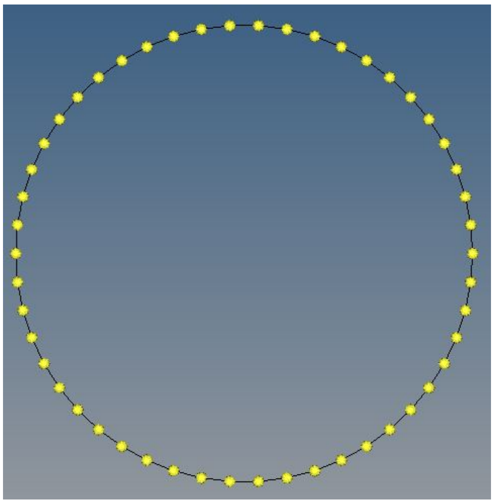
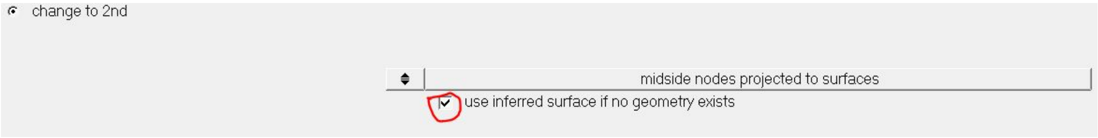


Image 5.17: 50 divisions, but 25 elements S8R

- Mid-side nodes projected to surfaces: this option of Hypermesh should be used instead of the another available option " mid-side nodes at exact midpoint ". And it should be used, with the additional option " use inferred surface if no geometry exists " checked, as it appears in image 5.18.



● change to 2nd

midside nodes projected to surfaces

☑ use inferred surface if no geometry exists

Image 5.18: Mid-side nodes projected to surfaces

The option " mid-side nodes at exact midpoint " creates the new nodes (which transform the elements into elements of second order) exactly at the midpoint located between the old nodes. This wastes the positive effect in accuracy of second order elements, because the meshing done this way does not approach the sought round shape.

Besides, the option of inferred geometry should be checked, because the method used in Hypermesh to mesh the wire, consists of meshing the base, and then drag the elements upwards, so actually, no geometry exists along the wire, just at the base.

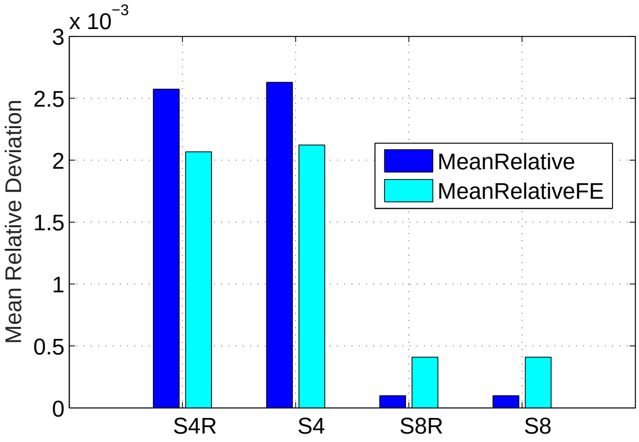


Image 5.19: Evaluation of Element Type

Comment on image 5.19 Four simulations have been carried out, each with a meshing for the coil/wire built by each of the four available shell elements. The graph does not look like as the previous others, for the mesh parameter analyzed here is not a quantifiable one. A graph bar was chosen to depict clearly the information obtained.

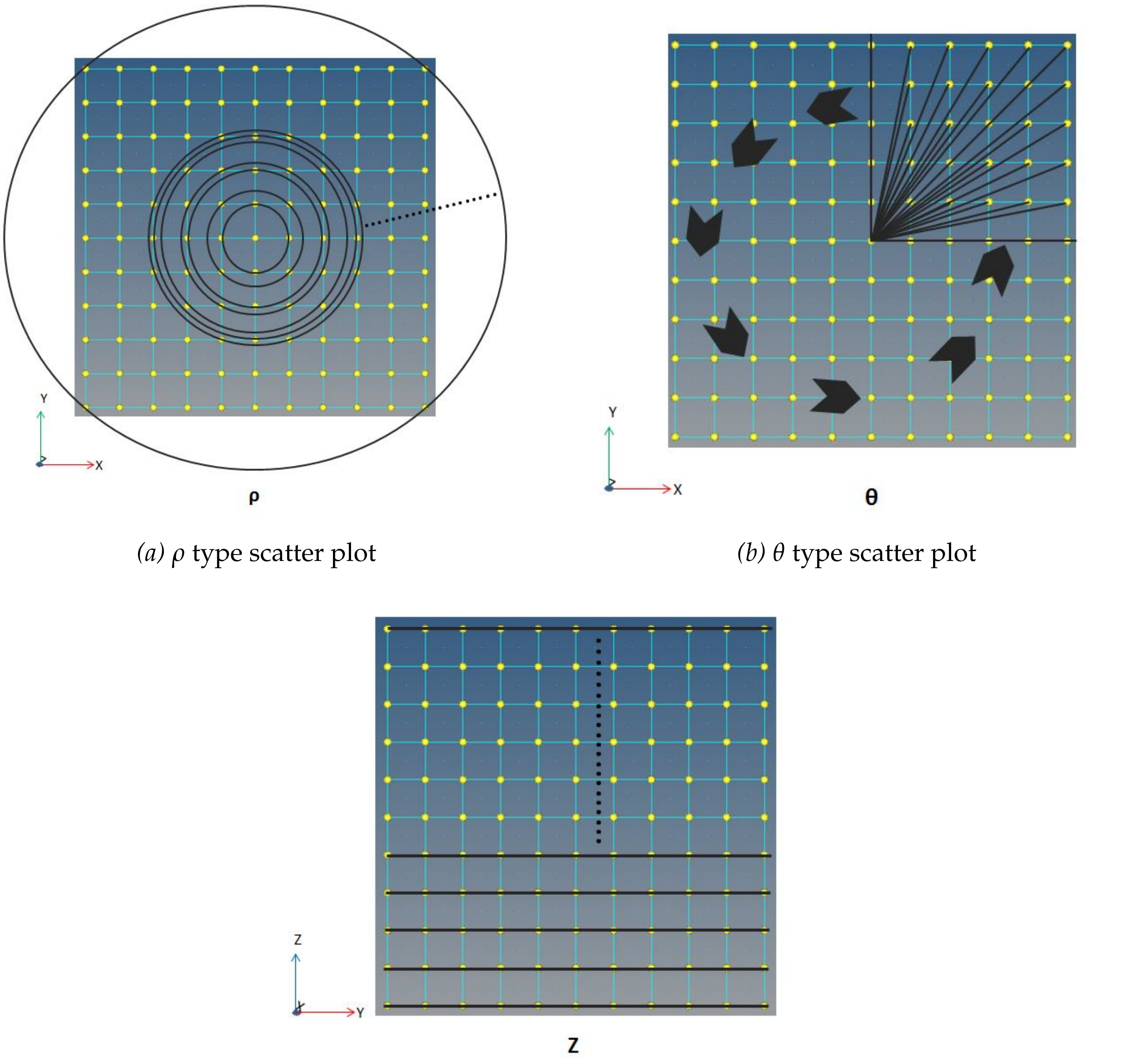
For the mean relative deviation of all nodes in each simulation, the first order elements (S4R and S4) reach an accuracy of approximately 0.2%, while the second order elements (S8R and S8) of approximately 0.0098%. This seems logical, because second order elements improve the numerical integration and provide results closer to the analytical solution of $\vec{\mathbf{T}}_0$, which indeed can be seen at the graph. If the details mentioned before about the change in order of the elements are not considered, the advantage given by the second order elements is wasted, and they do not approach better the analytic solution, for the meshing does not approach well the sought round shape of the wire conductor. So, as important as the chosen element type is the way of meshing.

Another outcome of the analysis, is the difference between simulations with elements that have reduced integration points, and simulations with elements with normal number of integration points. In both cases of first order and second order elements, the simulations with elements with reduced integration result slightly more accurate. Hence, such slightly improvement is not enough to point out the case as an event of locking.

Lastly, it is remarkable that the differentiation between ′MeanRelative′ and ′MeanRelativeFE′ values works as usual with first order elements, being more accurate the ′MeanRelativeFE′ ones, but it does not with second order elements, as it can be appreciated at the graph.

5.8 Cylindrical coordinates Scatter plots; different meshing

So far, the sought global verification has been depicted graphically through mean quantities. That is a way ideal to compare simulations, for there are available quantities that summarize all information. The scatter plots were design to avoid such summary, and display the information as it exactly is.



(a) ρ type scatter plot

(b) θ type scatter plot

(c) Z type scatter plot

Image 5.20: ScattePlot ρ, θ and Z Information

Due to the cylindrical shape of the wire and the natural null divergence of the magnetic field, to perform the scatter plots it is preferred the cylindrical coordinate system (ρ, θ, Z) instead of the Cartesian (X,Y,Z) one. Three types of scatter plots are performed, corresponding each type with one of the cylindrical coordinates. It should be remembered that the coordinates origin, for the the cylindrical coordinate system too, is located in the center of the room size, matching the middle of the wire.

In the previous section, in each successive simulation the wire/coil's mesh was made finer. Now, a good enough meshing for the wire was chosen according to the results of last section, and it is kept in all simulations. On the other hand, the air's meshing is modified in each simulation. The air is meshed in a regular and homogenous way, with all volumetric elements of the same size. It is so because it is sought the verification of the calculations of **ccx**, the biased meshes are used when it seeks more accuracy in a particular place, here the target is to analyze the accuracy globally. Therefore, the air's meshing has two mesh parameters in the successive simulations, the element type and the element size.

Before moving on with more simulations to analyze the air's meshing, the three types of scatter plots are presented in images 5.21, 5.22 and 5.23, to see how such scatter plots look like. In the captions of the images the type of scatter plot is indicated, as well as the element type and the element size of the air's meshing. The 'MeanRelative' and 'MeanRelativeFE' differentiation is added to the plots, just this time it is depicted as 'Relative' and 'RelativeFE', because the plotted relative deviations are nodal, not mean values of all nodes as in the previous section.

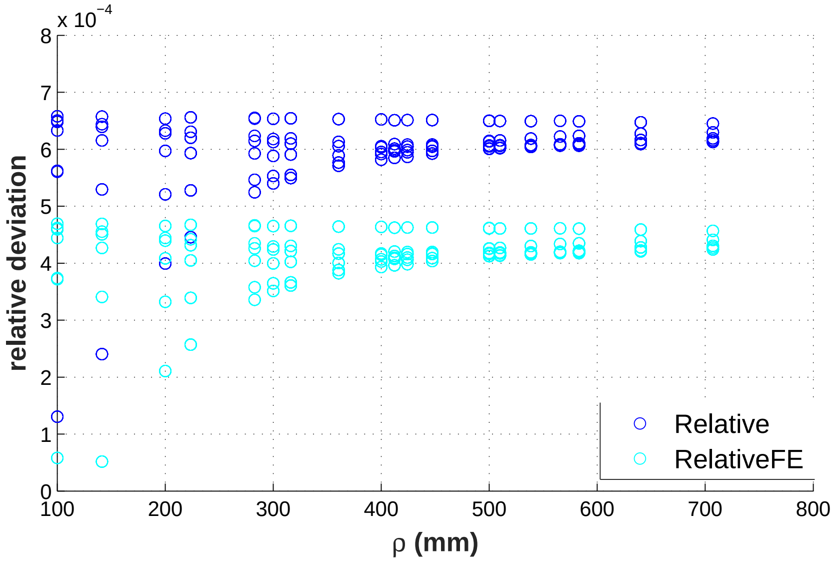


Image 5.21: Scatter plot ρ; C3D8 10cm

In the graph 5.21 a constant deviation between $\vec{T}_{0ccx}$ and $\vec{T}_{0analytic}$ is seen, as many nodes have the same value of deviation in spite of the radial distance. On the other hand, between 100 mm and 500 mm approximately, there are some nodes which do not show such constant behavior of the deviation, instead they are more accurate the closer to 100 mm.

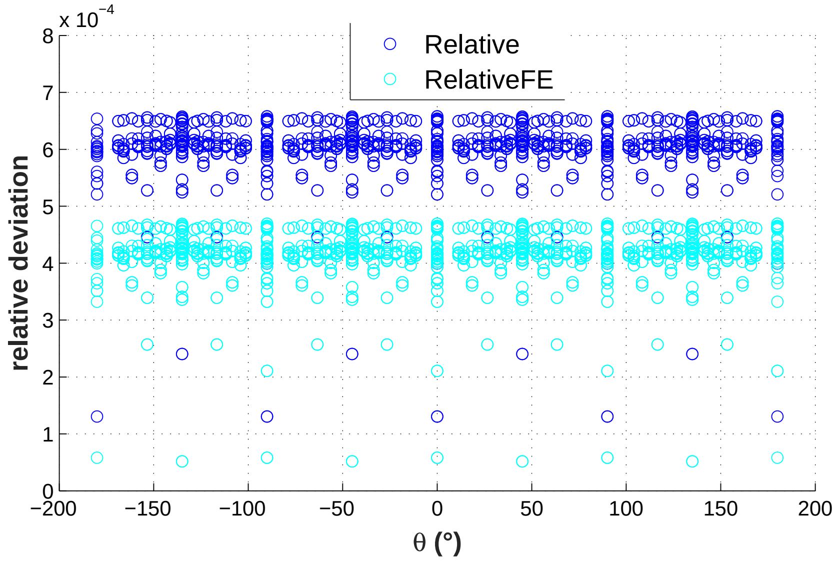


Image 5.22: Scatter plot θ; C3D8 10cm

In image 5.22, the plotted nodal deviations form four equal shapes, each defined in intervals of 90°([-180°,-90°],[-90°,0°],[0°,90°],[90°,180°]). That means the nodal relative deviations repeat their values in each of the four intervals, i.e., the nodal relative deviation is equally distributed each 90°. This can be seen as an indication that **ccx** works well, as the nodal deviation between $\vec{\mathbf{T}}_{\mathbf{0}ccx}$ and $\vec{\mathbf{T}}_{\mathbf{0}analytic}$ only varies within a 90°quadrant, being the same in all quadrants. Such behavior is in consonance with the null divergence of the magnetic field.

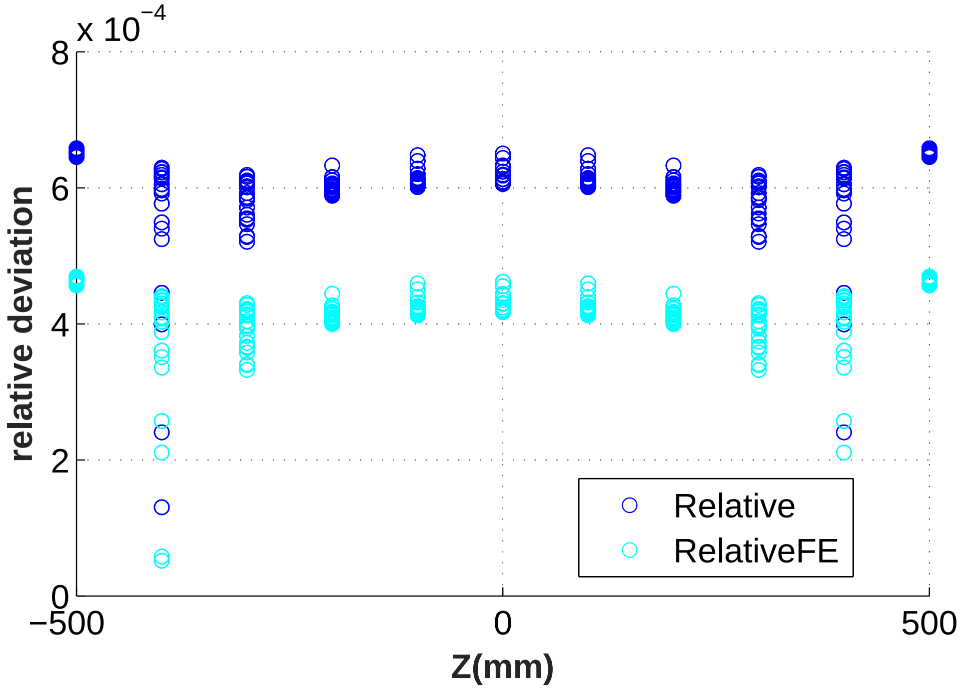


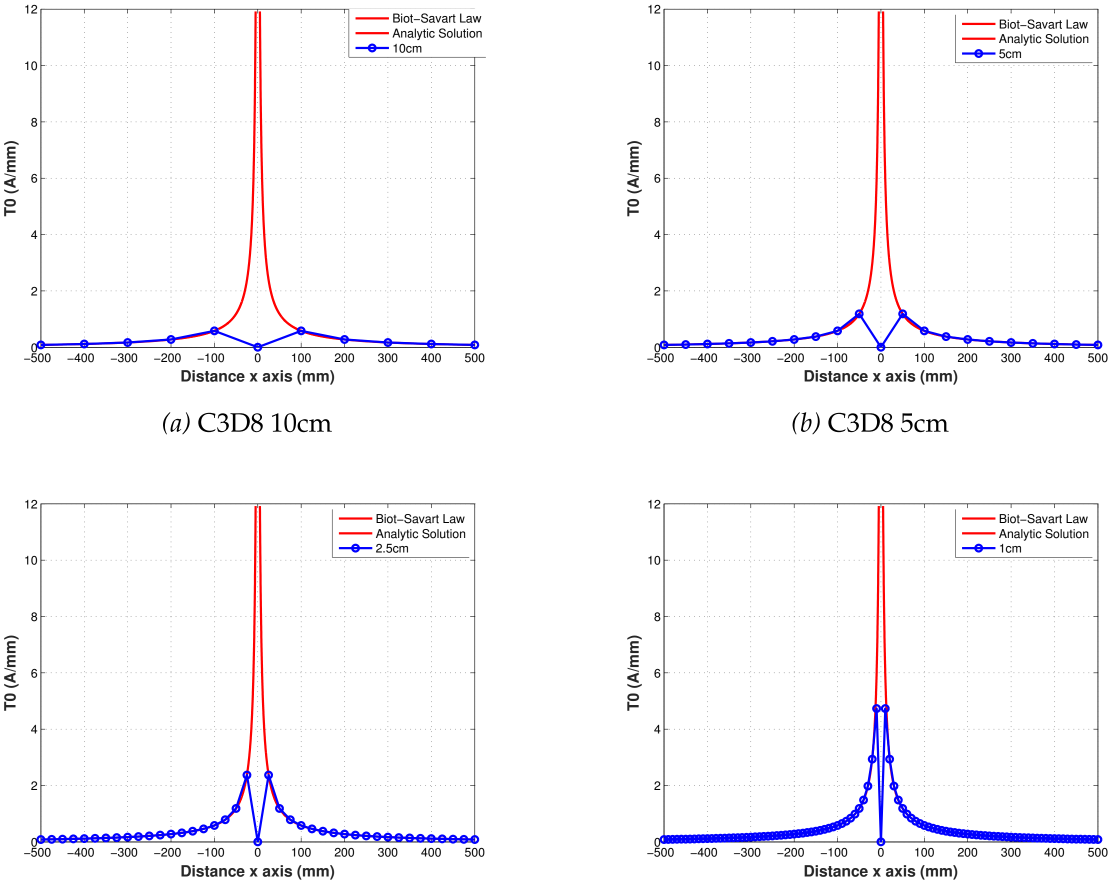
Image 5.23: Scatter plot Z; C3D8 10cm

Lastly, in image 5.23, it can be seen that the nodal relative deviations are distributed symmetrically with respect to XY plane(Z=0). In addition to that, it is remarkable that the distribution of the nodal relative deviations for each Z coordinate is not equal. Regarding the analysis of the air's mesh, the table 5.4 has the meshing information of all performed simulations to that end. As it was mentioned, the meshing for the wire is always the same and just the element size of the air's meshing is changed. The element type of the air's meshing is of no interest here, for it does not affect the calculation of $\vec{\mathbf{T_0}}_{ccx}$; which is a nodal calculation, independently of the element type(**ccx** simply calcultes $\vec{\mathbf{T_0}}$ at the coordinates of the nodes).

$EType_{Air}$	$ESize_{Air}(cm)$	$EType_{Coil}$	Circumference	Drag	$I_{Analytic}(A)$	$I_{FE}(A)$
C3D8	10	S4R	100	50	374.47784	374.40720
C3D8	5	S4R	100	50	374.47784	374.40720
C3D8	2.5	S4R	100	50	374.47784	374.40720
C3D8	1	S4R	100	50	374.47784	374.40720

Table 5.4: Evaluation of air's meshing simulations; meshing data.

In the air's meshing analysis, before focusing in the global perspective with the scatter plots, a local point of view has been considered. In mosaic 5.24 there are four graphs grouped, where for each air's meshing contained in table 5.4, a local comparison between the analytic solution by Biot-Savart and the values of $\vec{\mathbf{T_0}}$ provided by **ccx** is represented. It is local because the comparison is made at a group of air nodes locates along X axis (y=0,z=0). The group of nodes grows as the air's meshing gets finer, the idea is to accomplish a local graphical comparison similar to a convergence analysis.



(a) C3D8 10cm

(b) C3D8 5cm

(c) C3D8 2.5cm

(d) C3D8 1cm

Image 5.24: Local analysis; comparison between $\vec{\mathbf{T}}_{\mathbf{0}ccx}$ *and* $\vec{\mathbf{T}}_{\mathbf{0}analytic}$

The results of **ccx** follow the analytical solution in accordance with the magnitude of the deviations observed in previous sections. To appreciate the deviations, a zoom in must be performed. This has been done at image 5.25 from graph (a) of 5.24, where it can be seen that the small deviations exist.

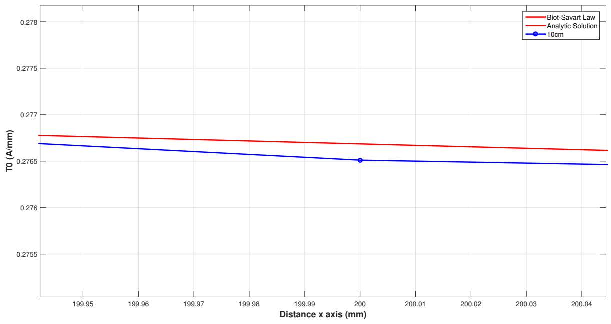


Image 5.25: Zoom at graph (a) from mosaic 5.24

$\vec{\mathbf{T}}_{\mathbf{0}ccx}$ is calculated very accurately along the entire X axis, with the exception of the vicinity of the wire. Boundary conditions regarding the size of the air room do not affect the calculation of $\vec{\mathbf{T}}_{\mathbf{0}ccx}$, as it can be see in graphs of 5.24, the node values of $\vec{\mathbf{T}}_{\mathbf{0}ccx}$ close to outer faces of the air room (x=-500 or x=500) are as well calculated as the node values located in the middle of the air room, and that happens independently of the mesh size (just the finer the mesh the more nodal values are calculated). Nevertheless, as it was mentioned, and now it is when the convergence analysis turns interesting, in the vicinity of the wire, it can be seen that the finer the mesh, as it is a regular mesh with all elements equally sized, the more nodal values are able to follow in such vicinity the analytic solution; however, in the finest mesh depicted in graph (d), it can be seen that the closest nodal values to the origin, where the wire is located and flows the source electric current, begin to diverge from the analytical solution.

Therefore, the graphs 5.24 confirm two facts which will be better analyzed in a global way with the scatter plots, however, by now:

- The calculation of $\vec{\mathbf{T}}_{\mathbf{0}ccx}$ is not affected by boundary conditions of the air room, **ccx** simply calculates statically $\vec{\mathbf{T}}_{\mathbf{0}}$ at the coordinates of the air nodes; the finer the mesh, the more points to calculate, but a refinement of the air′s meshing does not mean a direct improvement in the accuracy of the calculation of $\vec{\mathbf{T}}_{\mathbf{0}ccx}$, as it does a refinement of the coil′s meshing (where it flows the electric current). $\vec{\mathbf{T}}_{\mathbf{0}ccx}$ is calculated according to Biot-Savart′s law, thus the accuracy in the calculation is set by how good are considered the contributions of the shell elements through the electric current flows, which is determined by the coil′s meshing. The air′s meshing just locate spatially a higher or smaller number of air nodes (depending on how fine is the mesh), its unique role in the calculation of $\vec{\mathbf{T}}_{\mathbf{0}ccx}$ is to determinate the coordinates where ccx must calculate $\vec{\mathbf{T}}_{\mathbf{0}ccx}$. Because of that, nodes belonging to coarser meshes, if happen to exist in finer meshes with the same coordinates, are going to have the same value of $\vec{\mathbf{T}}_{\mathbf{0}ccx}$, and hence, the same deviation with respect to the analytic solution. This can better visualized in image 5.26, which groups all the information of image 5.24.
- Where $\vec{\mathbf{T}}_{\mathbf{0}ccx}$ does lose accuracy with respect to the considered analytic solution is in the vicinity of the coil/wire. In image 5.26 can be seen that the closest nodes to the origin, belonging to the finest mesh, diverge of the analytic solution. This can be because of the way that is considered the analytic solution (filament current solution I instead of volumetric current solution $\vec{\mathbf{J}}$), or actually **ccx** is not able to calculate $\vec{\mathbf{T}}_{\mathbf{0}}$ in such close points to the origin. The scatter plot should shed light on this issue.

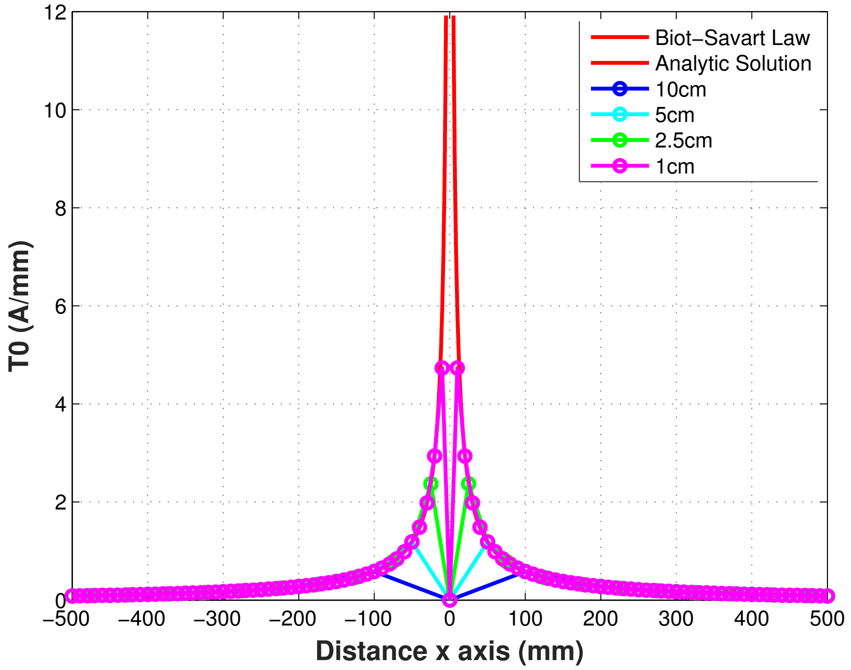
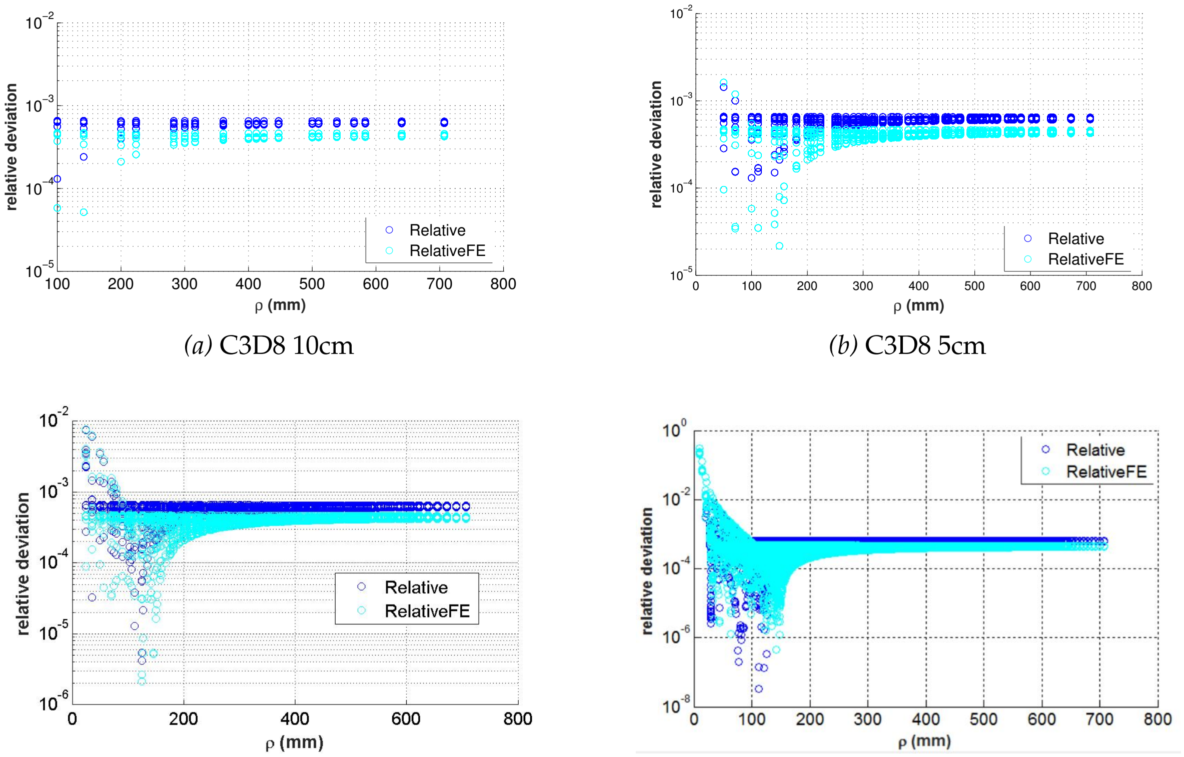


Image 5.26: Local analysis; Image 5.24 all together

Coming back to the scatter plots, the three types of scatter plots have been made for each of the simulations indicated in table 5.4. They are grouped by the used cylindrical coordinate in three mosaics: 5.27 for ρ, 5.28 for θ and 5.29 for Z. Each mosaic has four scatter plots corresponding with the four air meshes tried out. The mosaic resembles a convergence analysis, as in each of its graphs the air's meshing gets finer.



(a) C3D8 10cm

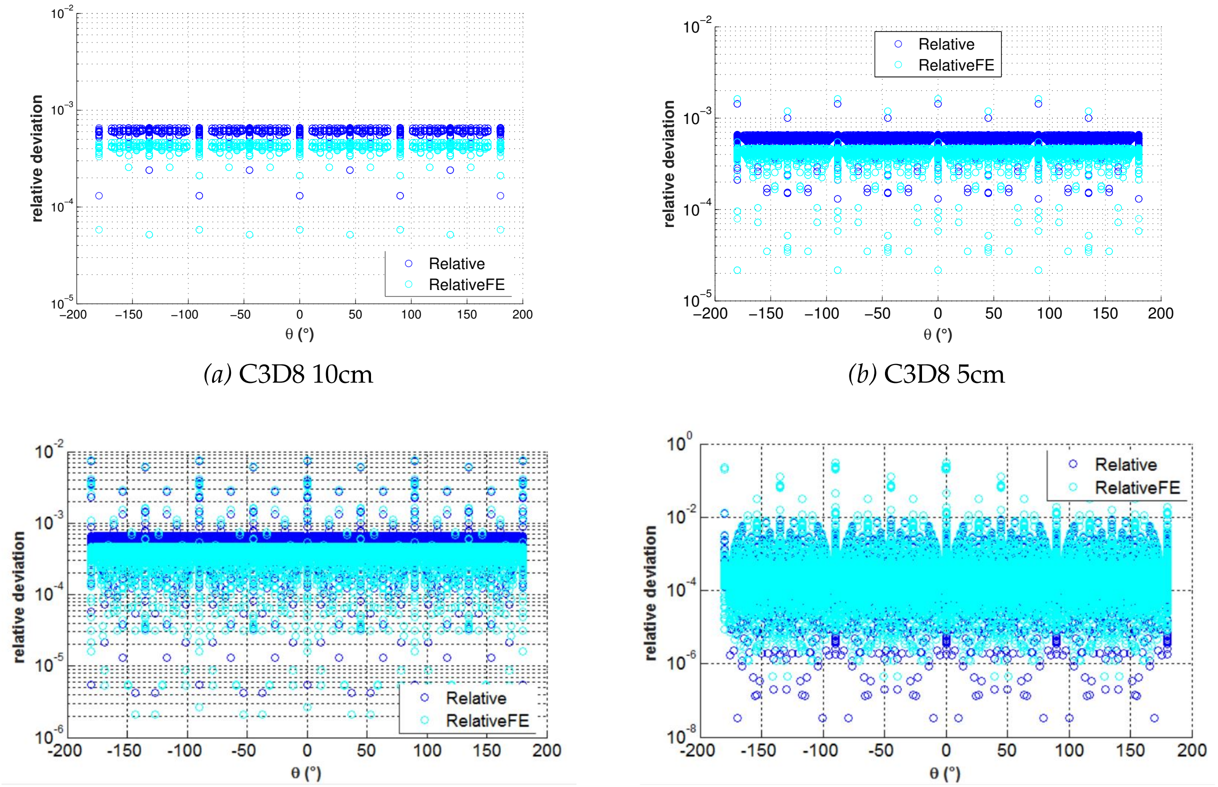
(b) C3D8 5cm

(c) C3D8 2.5cm

(d) C3D8 1cm

Image 5.27: Scatter plots ρ; Evaluation of air's meshing; Logarithmic scale

The graphs of image 5.27 confirm what was derived from the previous analysis. The shape built by the nodal deviations of the graphs is constant, as the mesh gets finer, such shape is defined by more nodal values. Hence, nodal deviations with constant values exist, which are independent of the radial distance. In addition to that, the nodal deviations close to the origin get less accurate. This issue increases as the air's mesh gets finer.



(a) C3D8 10cm

(b) C3D8 5cm

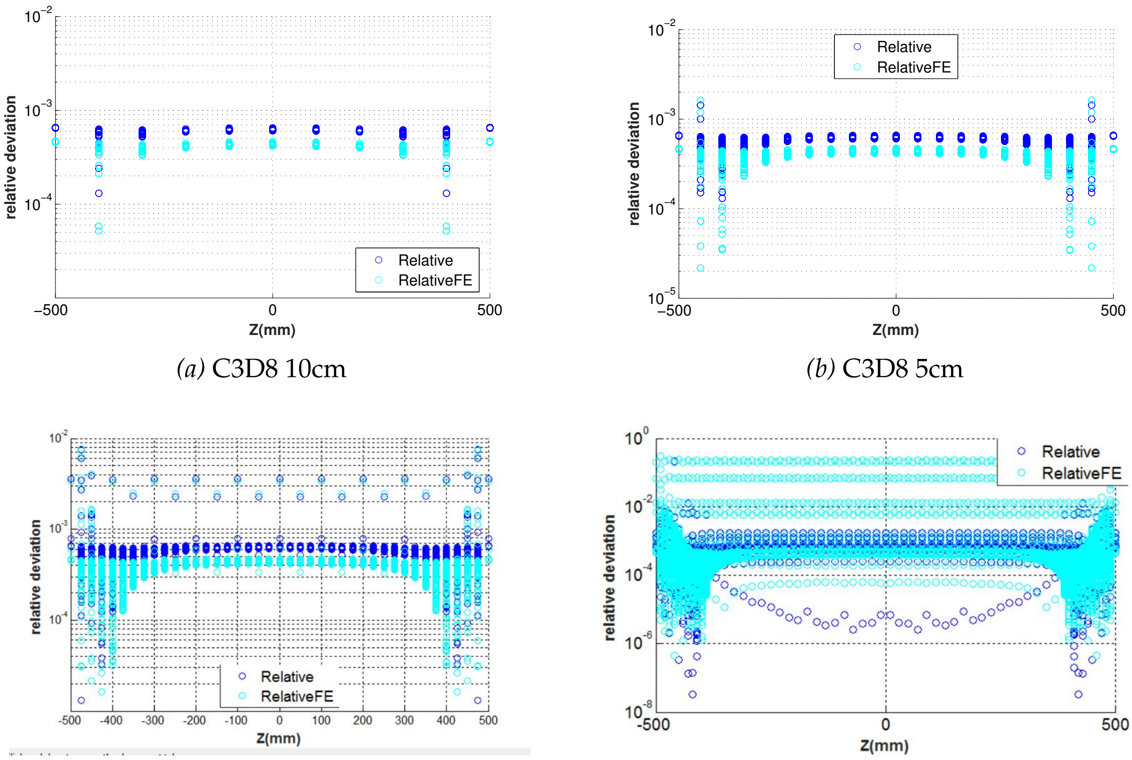
(c) C3D8 2.5cm

(d) C3D8 1cm

Image 5.28: Scatter plots θ; Evaluation of air's meshing; Logarithmic scale

In image 5.28, similar to image 5.27, a shape is maintained along the graphs, built as more nodes are added due to a finer mesh. Likewise, the finer the mesh, the more nodes are located close to the origin, which decreases the accuracy of the nodal deviations; although, this issue is difficult to see only in image 5.28, it has to be observed together with image5.27.

The mentioned similarity between quadrants in θ type scatter plots is also observed in image 5.28.



(a) C3D8 10cm

(b) C3D8 5cm

(c) C3D8 2.5cm

(d) C3D8 1cm

Image 5.29: Scatter plots Z; Evaluation of air's meshing; Logarithmic scale

Lastly, in image 5.29 the same comments about the shape of the graphs as in images 5.27 and 5.28 can be applied. However, it is remarkable that the graph (d), the one with the finest mesh, does not maintain the shape entirely.

The symmetry with respect to plane XY (Z=0) is maintained in all graphs from image 5.29.

INTENTIONALLY BLANK PAGE

6 Verification of $\vec{\mathbf{H}}$

Once $\vec{\mathbf{T_0}}$ has been tested, the next step to verify equation (3.1) is to analyze P, the magnetic scalar potential. The boundary and interface conditions influence in the calculation of the gradient of the magnetic scalar potential. Due to how **ccx** performs this calculation, it turns more advantageous to focus the verification in the final result provided, the magnetic field $\vec{\mathbf{B}}$. However, instead of $\vec{\mathbf{B}}$, it was decided to focus the verification on $\vec{\mathbf{H}}$. Both vectors are related by the magnetic permeability through equation (2.1). The reason is related with the previous chapter, where $\vec{\mathbf{T_0}}$ was analyzed as a magnetic field intensity, not a magnetic flux density or simply magnetic field. So to facilitate the required references to the previous analysis of $\vec{\mathbf{T_0}}$, it was decided to keep with a magnetic field intensity such as $\vec{\mathbf{H}}$.

6.1 Objectives

The objectives sought to accomplish in the actual section can be summarized in the following:

- **Verification of $\vec{\mathbf{H}}$**: the implementation of $\vec{\mathbf{H}}$ in **ccx** will be tested. Again the desired verification is global, **ccx** must provide reasonable values of $\vec{\mathbf{H}}$ in any assessed point.
- **Meshing of the air room**: in the previous chapter the coil's meshing was extensively analyzed, now it is followed suit with the air's meshing.
- **Convergence analysis**: this analysis was carried out with respect to coil's meshing in the previous chapter. Here the analysis is perform with respect to air's meshing.
- **Size of the air room**:the air room's size does not affect the calculation of $\vec{\mathbf{T_0}}$, but it does affect the calculation of $\vec{\mathbf{H}}$, due to the boundary conditions at the room's limits. It will be analyzed the effect of the air room's size in $\vec{\mathbf{H}}$.

$$Relative \;\; deviation(H) = \frac{|H_{ccx} - T_{0_{ccx}}|}{T_{0_{ccx}}} \tag{6.1}$$

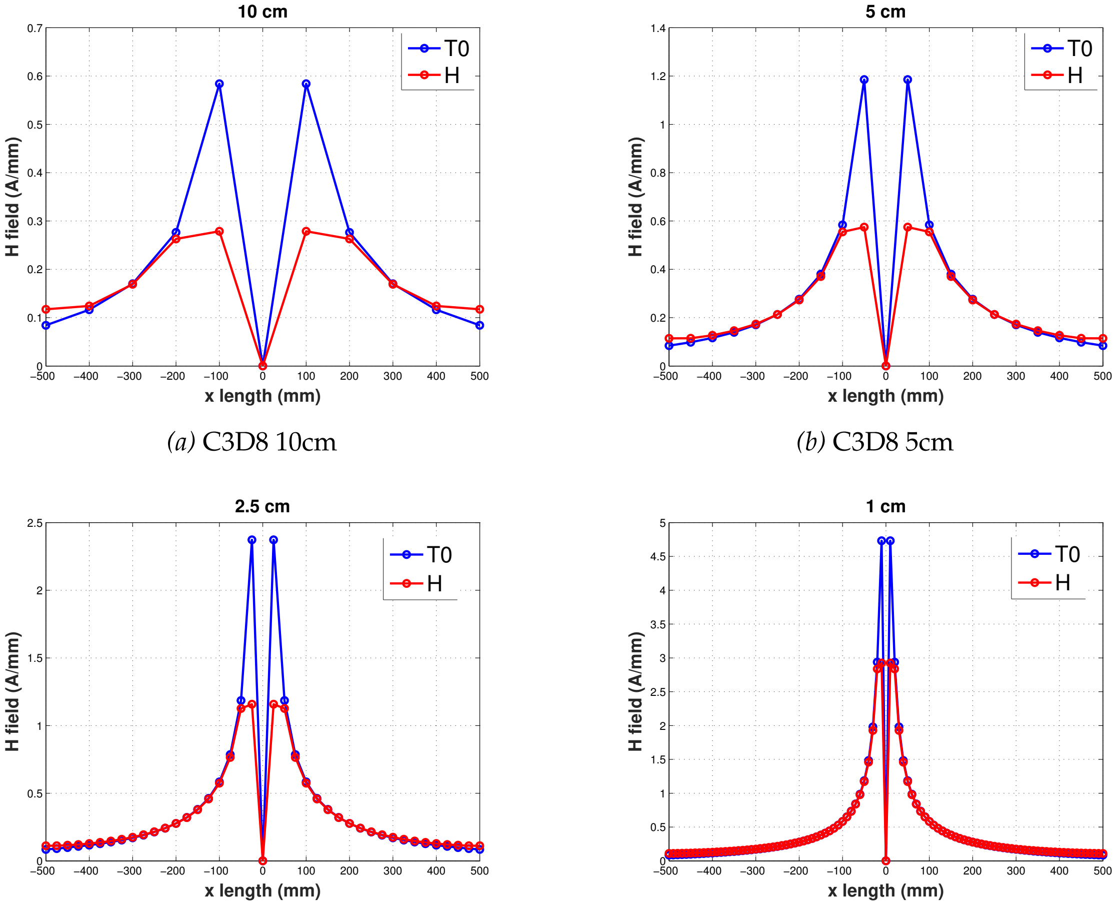
6.2 Conception of the FE models

The FE models built up for $\vec{\mathbf{H}}$ are based on the original examples for the electromagnetic module, which come along with the installation of **CalculiX** (*induction.inp*). The models are built with the sake of simplicity, the focus is on the post-processing part, where the results are analyzed in Matlab and it is actually when the verification can be accomplished. The models follow a general design, between them only varies the meshing, and occasionally the size of the built geometries, although their shape is always the same.

$EType_{Air}$	$ESize_{Air}(cm)$	$EType_{Coil}$	Circumference	Drag	$I_{Analytic}(A)$	$I_{FE}(A)$
C3D8	10	S4R	100	50	374.47784	374.40720
C3D8	5	S4R	100	50	374.47784	374.40720
C3D8	2.5	S4R	100	50	374.47784	374.40720
C3D8	1	S4R	100	50	374.47784	374.40720

Table 6.1: Evaluation of air's meshing simulations; meshing data.

Likewise in previous chapter, first a local analysis is carried out of a group of nodes located along the X axis (y=0,z=0). The analysis is performed for all simulations of table 6.1. The resulting graphs are grouped in image 6.1. Again the idea is to accomplish a local graphical comparison similar to a convergence analysis, but this time between the $\vec{\mathbf{T}}_{\mathbf{0}}$ and $\vec{\mathbf{H}}$ calculated by ccx. In the magneto-static without workpiece case analyzed here, the accuracy of $\vec{\mathbf{H}}_{ccx}$ is better as closer are its values to $\vec{\mathbf{T}}_{\mathbf{0}ccx}$.



(a) C3D8 10cm

(b) C3D8 5cm

(c) C3D8 2.5cm

(d) C3D8 1cm

Image 6.1: Local analysis; comparison between $\vec{\mathbf{T}}_{\mathbf{0}ccx}$ and $\vec{\mathbf{H}}_{ccx}$

On the contrary to $\vec{\mathbf{T}}_{\mathbf{0}}$, the calculation of $\vec{\mathbf{H}}$ by **ccx** is affected by the boundary conditions of the air room. That can be seen in image 6.1 in the values of $\vec{\mathbf{H}}$ of the nodes located close to the outer faces of the air room. In addition to that, the calculated $\vec{\mathbf{H}}$ values lose accuracy when they get close to the origin.

Both issues are not comparable with what happens with $\vec{\mathbf{T}}_{\mathbf{0}ccx}$, for the calculation of $\vec{\mathbf{T}}_{\mathbf{0}ccx}$ is independent of air room boundary conditions, and $\vec{\mathbf{T}}_{\mathbf{0}ccx}$ is inaccurate only in the vicinity of the coil, independently of the meshing. On the other hand, both issues related with $\vec{\mathbf{H}}_{ccx}$ can be dealt managing the meshing. As it can be seen in image 6.1 along the four graphs, as the air's mesh gets finer, both effects (inaccuracy at the outer faces of the air room and close to the origin) are reduced. Therefore, the refinement of the air's mesh is important to be considered in order to get more precision for $\vec{\mathbf{H}}_{ccx}$. This can be better appreciated in image 6.2.

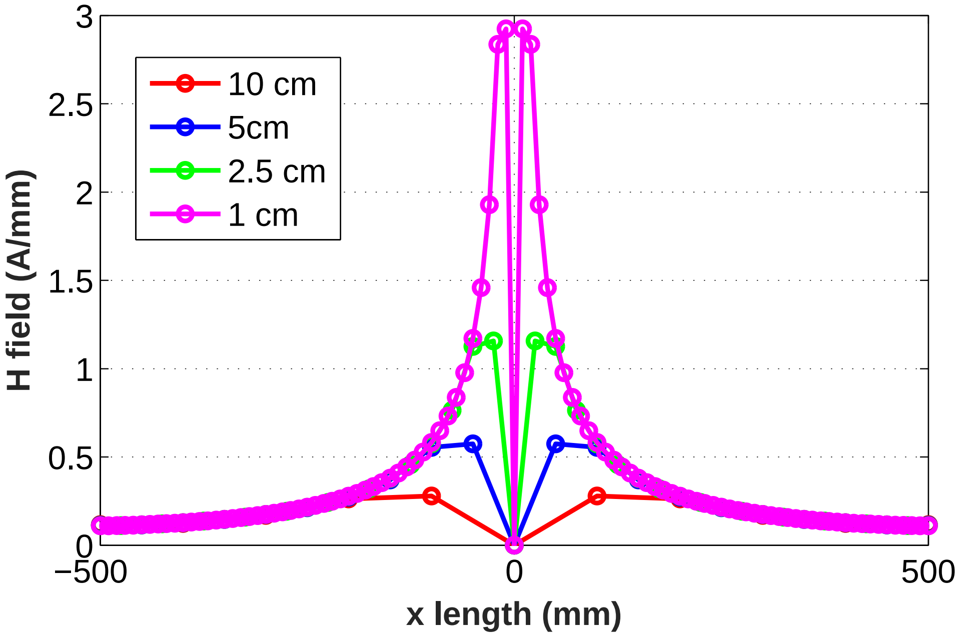


Image 6.2: Local analysis; Image 6.1 all together

6.3 Cylindrical coordinates normalized Scatter plots; different air domain mesh size

Firstly, to see how $\vec{\mathbf{H}}_{ccx}$ is represented in scatter plots, the three types are carried out for the first simulation of table 6.1: image 6.3 for ρ, image 6.4 for θ and image 6.5 for Z.

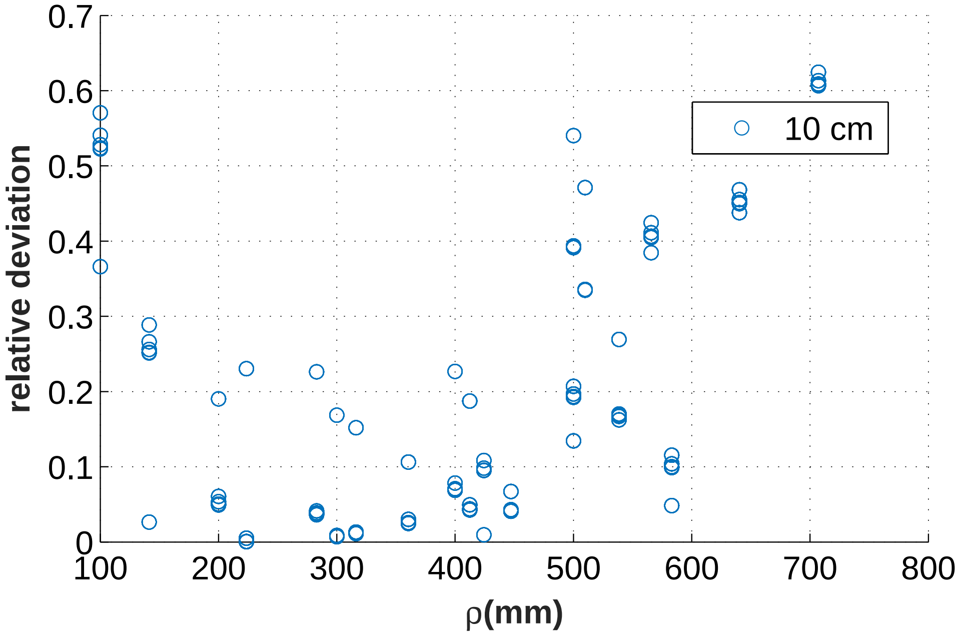


Image 6.3: Scatter plot ρ; C3D8 10cm

As the local analysis predicted, it can be seen in image 6.3 that $\vec{\mathbf{H}}_{ccx}$ is only enough accurate in the middle of the air room. This is due to the effect of the air room boundary conditions.

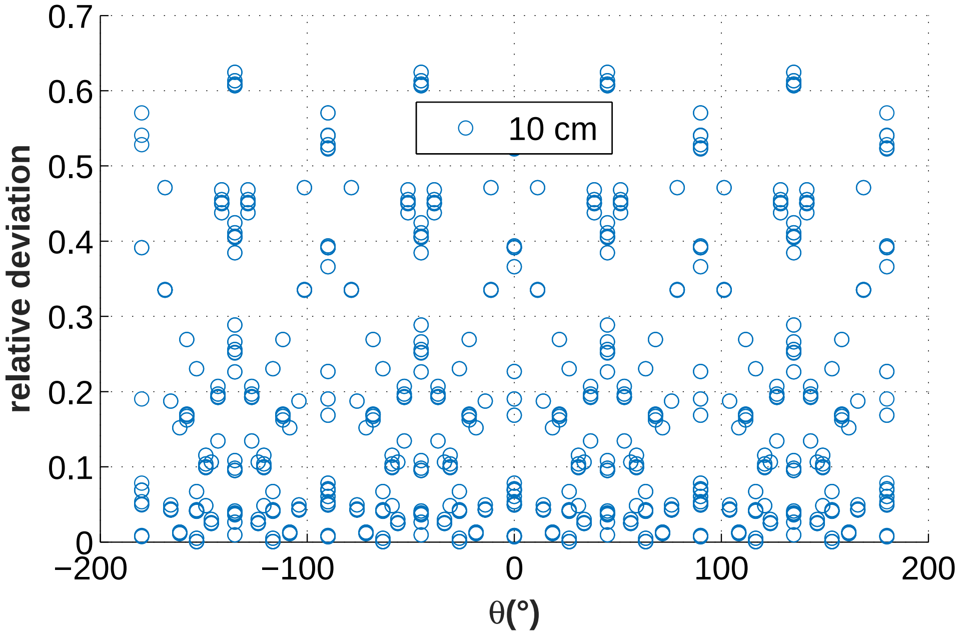


Image 6.4: Scatter plot θ; C3D8 10cm

Similar to $\vec{\mathbf{T}}_{\mathbf{0}ccx}$ case, the distribution of the nodal relative deviations is equal by quadrants.

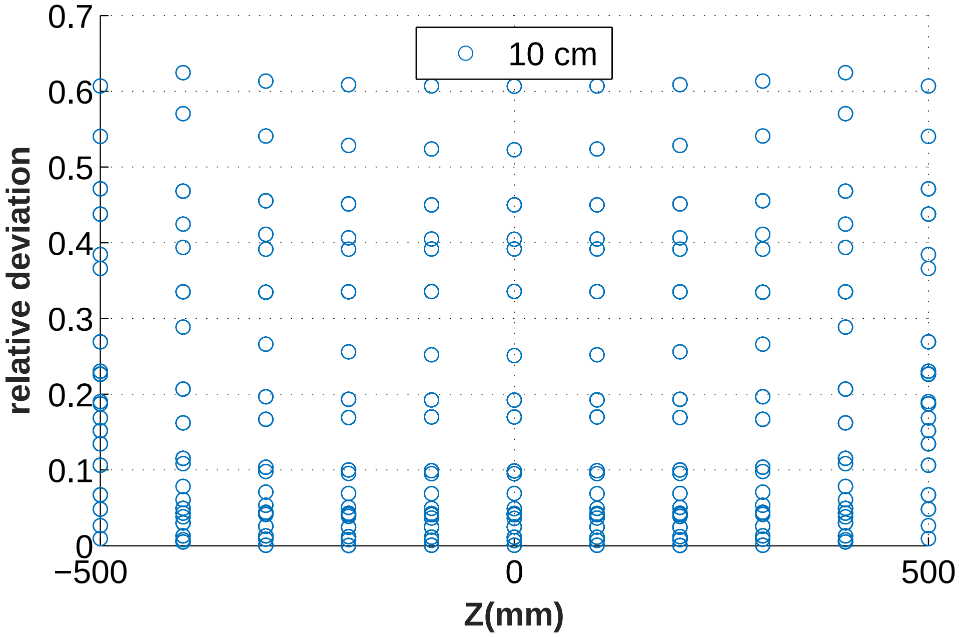
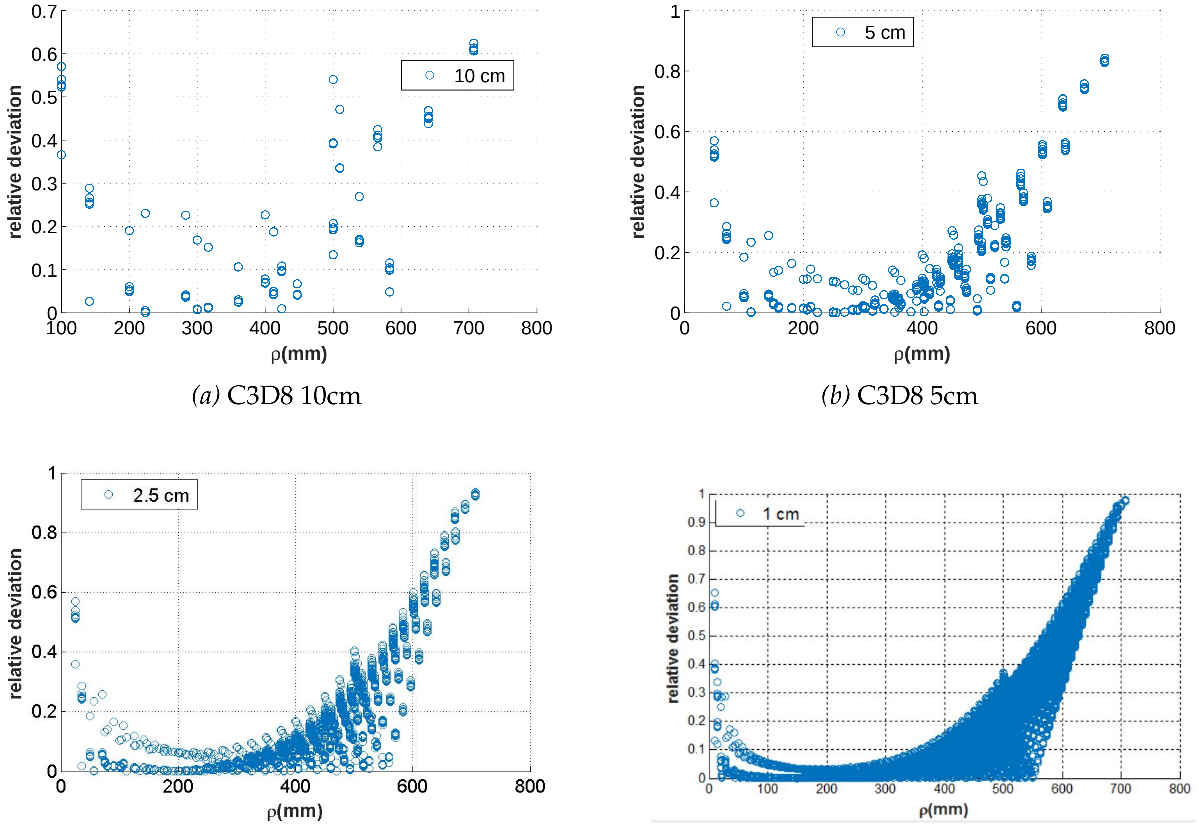


Image 6.5: Scatter plot Z; C3D8 10cm

Similar as well to $\vec{T}_{0ccx}$ case, the distribution of the nodal relative deviations is symmetric with respect to plane XY(Z=0). However, the distribution of the deviations in each Z level is more homogeneous in this case. The reason is that in each Z level there are both accurate nodes (located in the middle of the air room) and inaccurate nodes (located in the outer faces of the air room and close to the origin).

Now the convergence analysis for the air's meshing is performed completely. Scatter plots of each kind are depicted making each time the mesh finer.



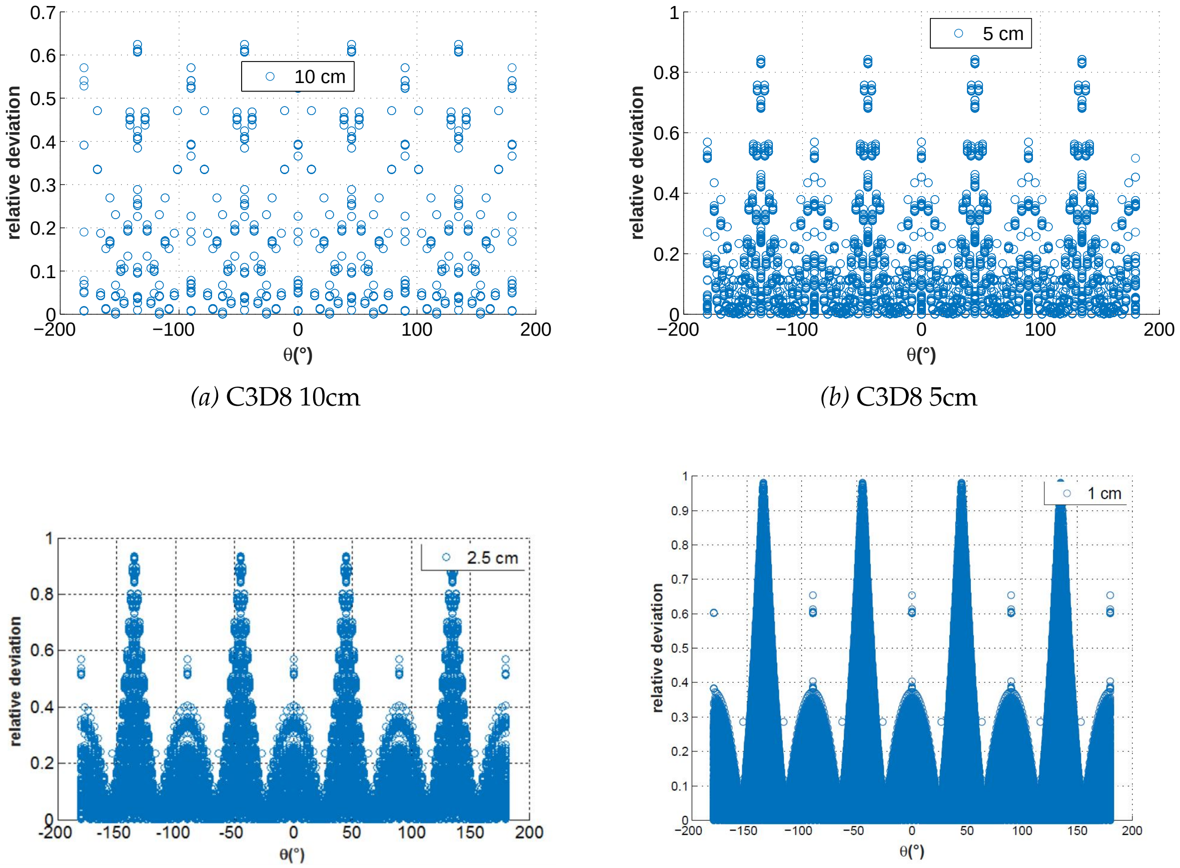
(a) C3D8 10cm

(b) C3D8 5cm

(c) C3D8 2.5cm

(d) C3D8 1cm

Image 6.6: Convergence analysis; Cylindrical Scatter plot ρ; different meshing



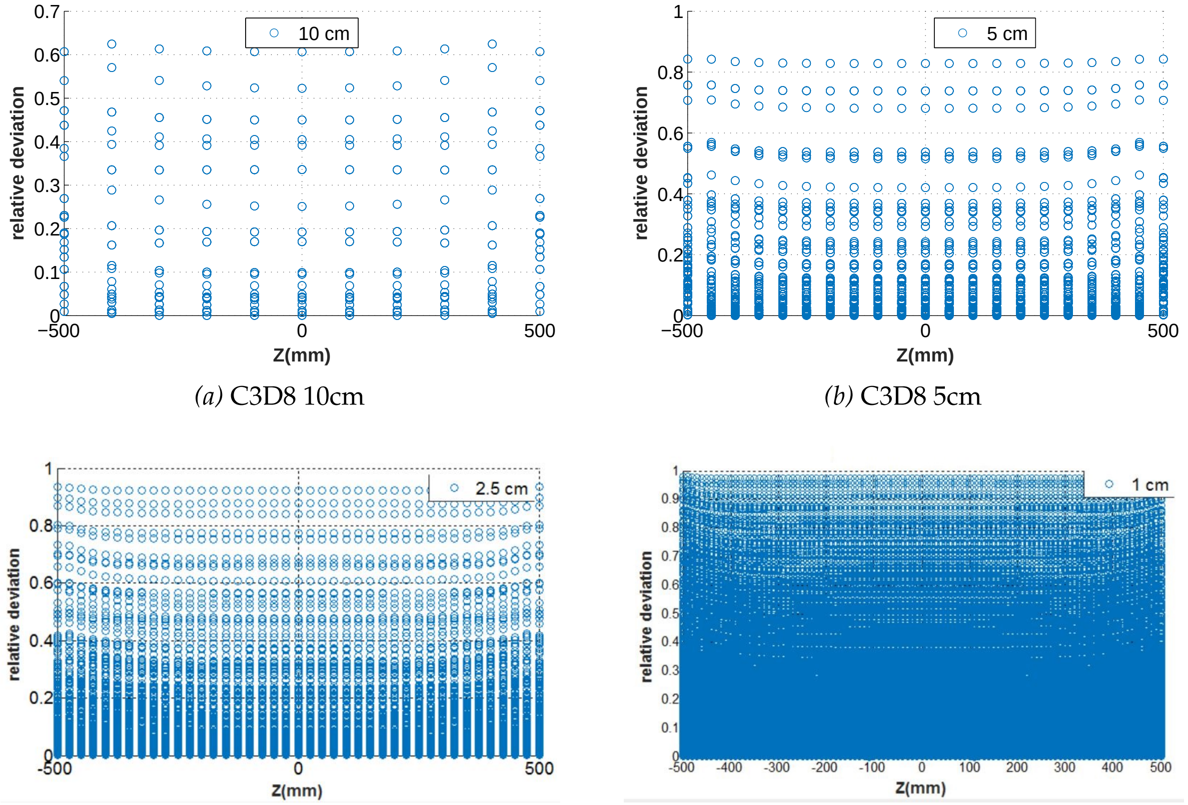
(a) C3D8 10cm

(b) C3D8 5cm

(c) C3D8 2.5cm

(d) C3D8 1cm

Image 6.7: Convergence analysis; Cylindrical Scatter plot θ; different meshing



(a) C3D8 10cm

(b) C3D8 5cm

(c) C3D8 2.5cm

(d) C3D8 1cm

Image 6.8: Convergence analysis; Cylindrical Scatter plot Z; different meshing

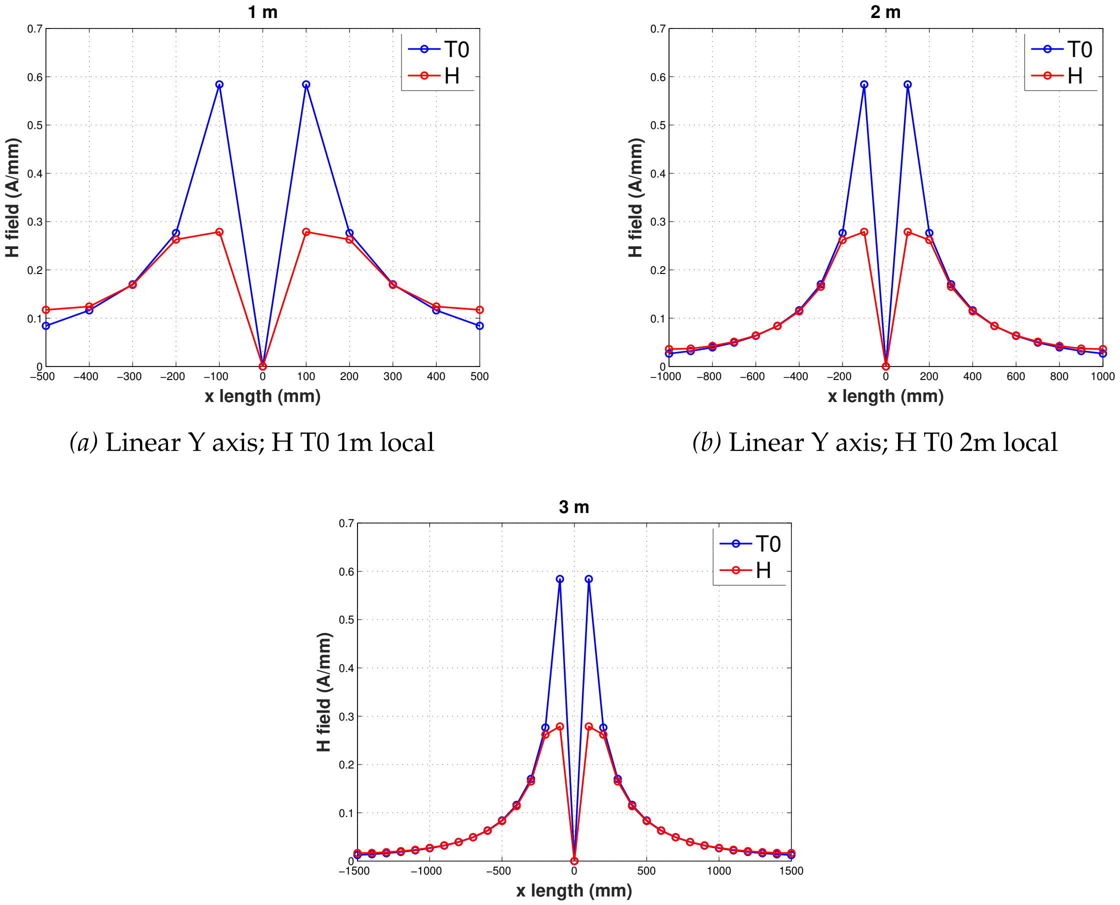
6.4 Cylindrical coordinates normalized Scatter plots; different air domain size

In this section, in order to be able to compare the simulations with different sizes, a normalization is carried out. The normalization only affects the ρ graphs and the Z graphs.

$CubeSize_{Air}(m)$	$EType_{Air}$	$ESize_{Air}(cm)$	$EType_{Coil}$	Circumference	Drag
1	C3D8	5	S4R	100	50
1.5	C3D8	5	S4R	100	50
2	C3D8	5	S4R	100	50
3	C3D8	5	S4R	100	50
4	C3D8	5	S4R	100	50

Table 6.2: Different sizes.

As usual, first a local analysis is performed.

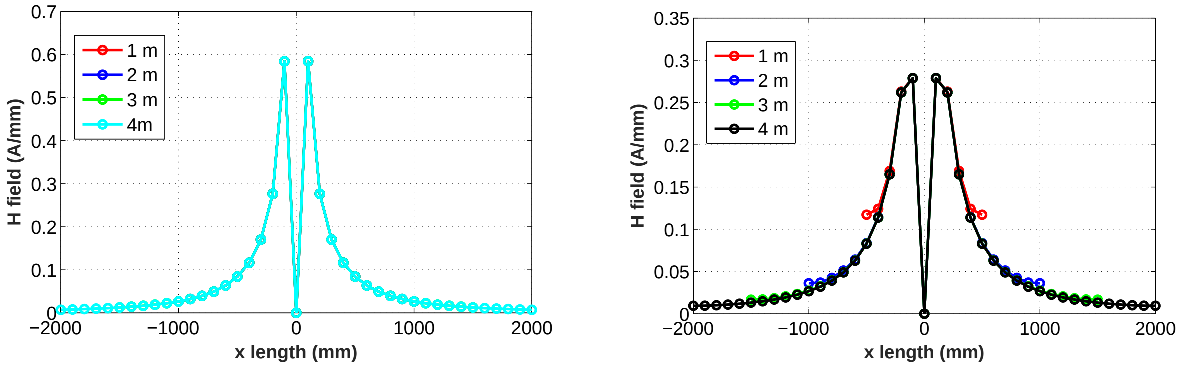


(a) Linear Y axis; H T0 1m local

(b) Linear Y axis; H T0 2m local

(c) Linear Y axis; H T0 3m local

Image 6.9: Convergence analysis; BiotSavart evaluation for different sizes



(a) Linear Y axis; T0 AllSizes local

(b) Linear Y axis; H AllSizes local

Image 6.10: Convergence analysis; BiotSavart evaluation for different sizes

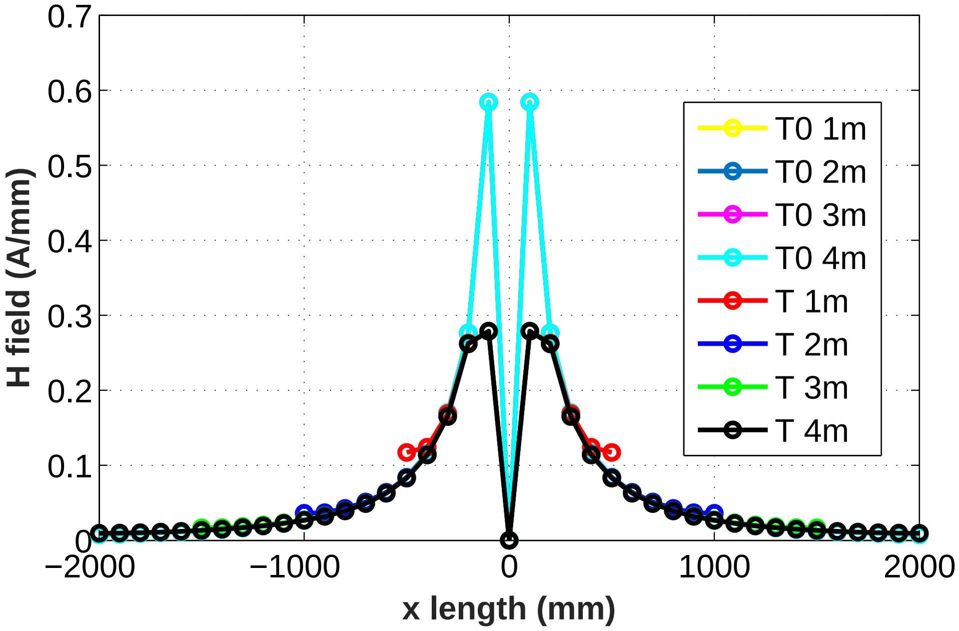
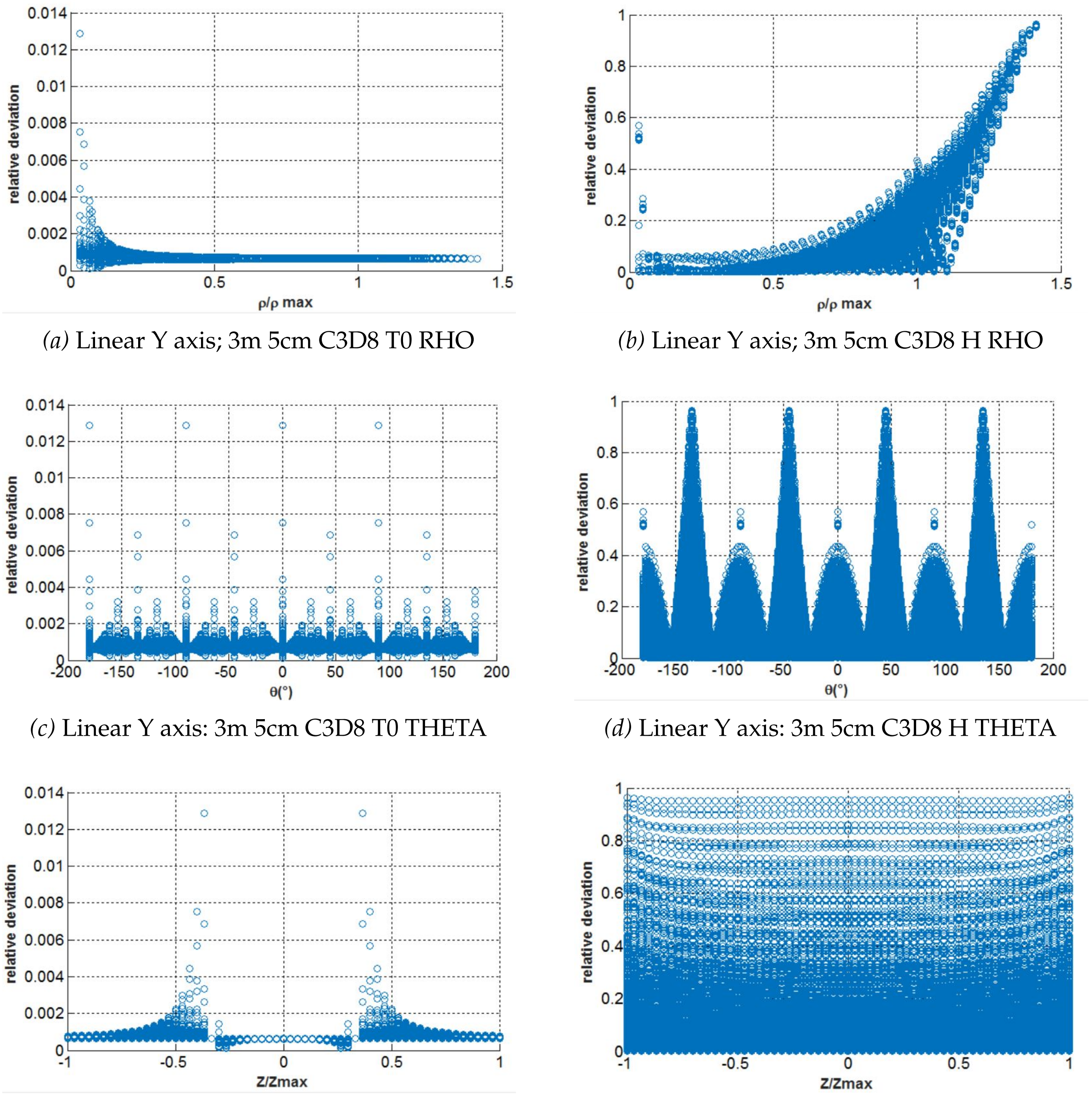


Image 6.11: Evaluation of room size

The room size affects the definition of the boundary conditions. A cube of 3m side was chosen to repeat the analysis performed up to now. Both $\vec{\mathbf{T}}_0$ and $\vec{\mathbf{H}}$ graphs are shown.



(a) Linear Y axis; 3m 5cm C3D8 T0 RHO

(b) Linear Y axis; 3m 5cm C3D8 H RHO

(c) Linear Y axis: 3m 5cm C3D8 T0 THETA

(d) Linear Y axis: 3m 5cm C3D8 H THETA

(e) Linear Y axis: 3m 5cm C3D8 T0 Z

(f) Linear Y axis; 3m 5cm C3D8 H Z

Image 6.12: Convergence analysis; BiotSavart evaluation for different sizes

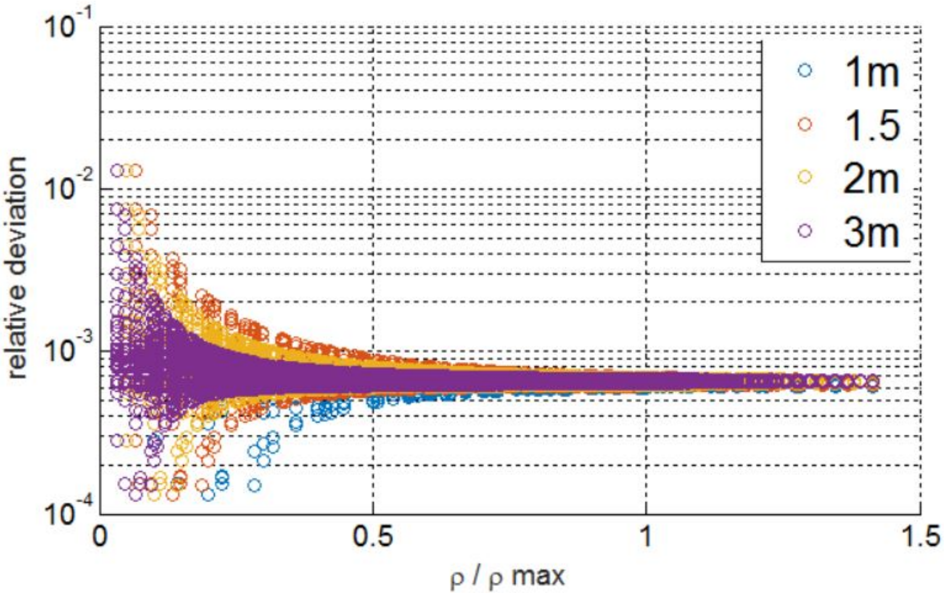


Image 6.13: ρ scatter plot; different sizes $\vec{\mathbf{T}}_0$

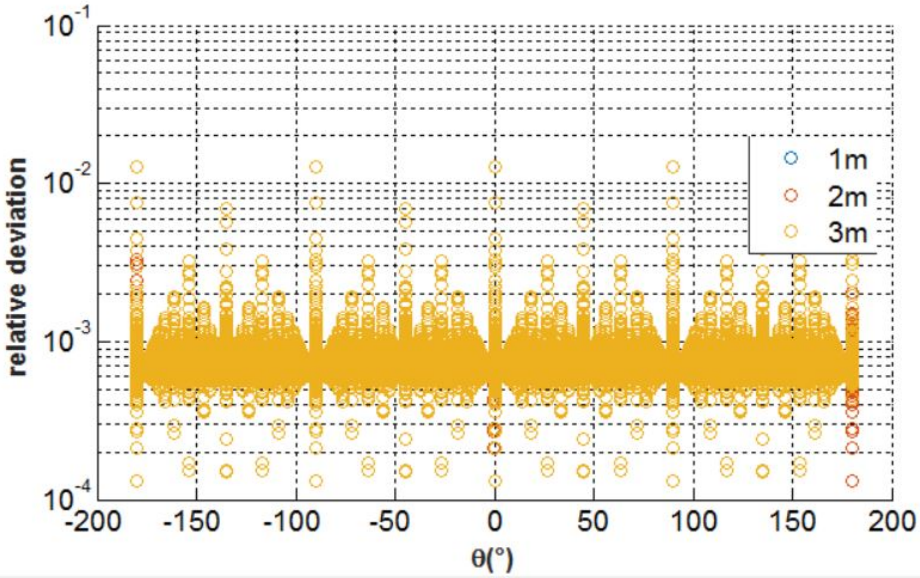


Image 6.14: θ scatter plot; different sizes $\vec{\mathbf{T}}_0$

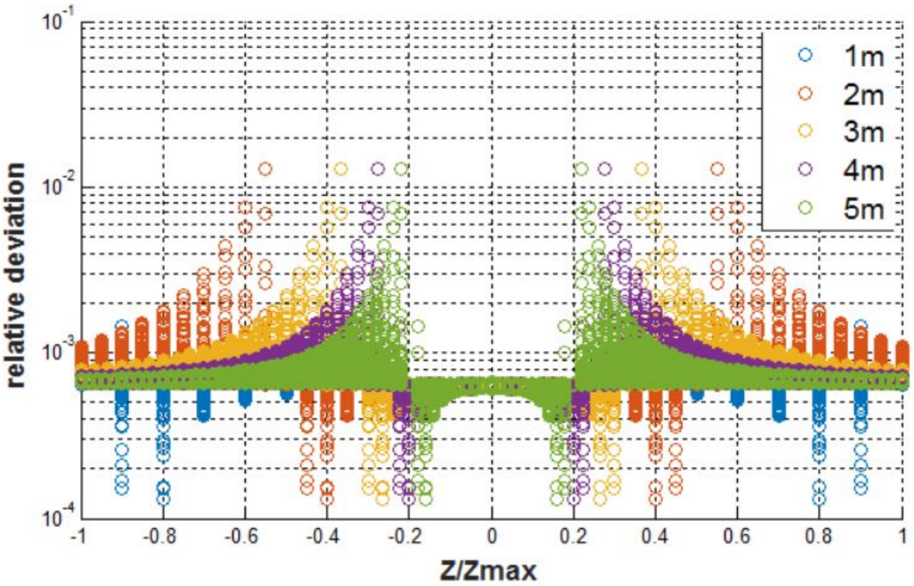


Image 6.15: Z scatter plot; different sizes $\vec{\mathbf{T}}_{\mathbf{0}}$

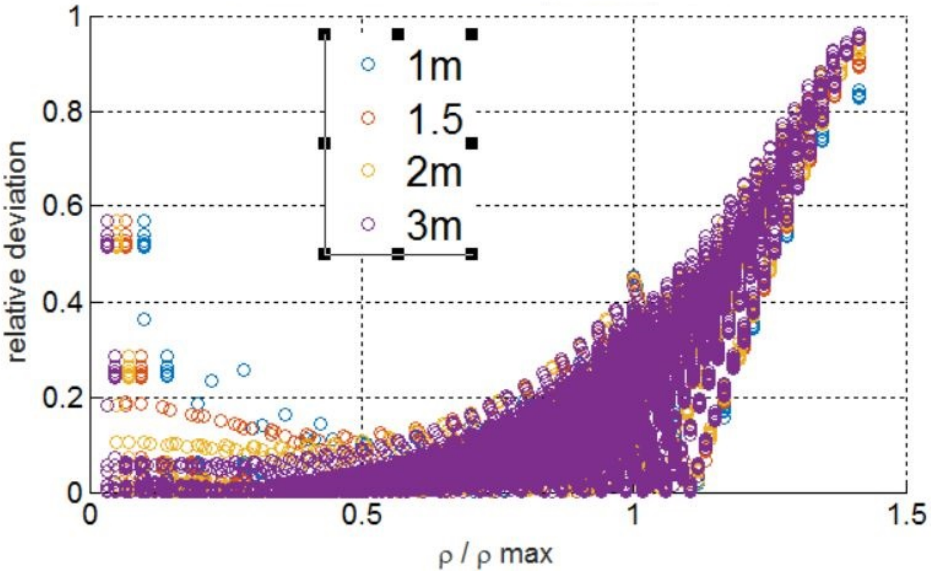


Image 6.16: ρ scatter plot; different sizes $\vec{\mathbf{H}}$

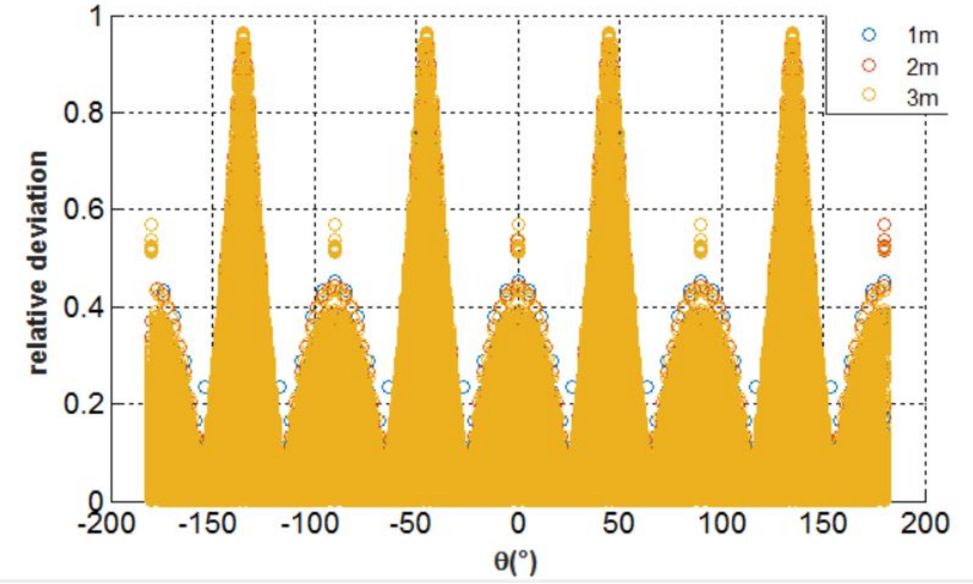


Image 6.17: θ scatter plot; different sizes $\vec{\mathbf{H}}$

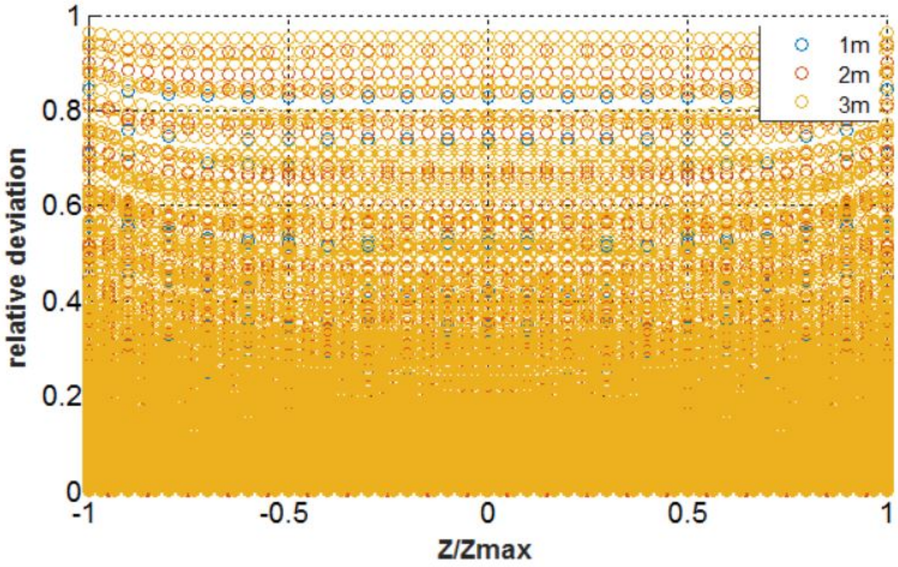


Image 6.18: Z scatter plot; different sizes $\vec{\mathbf{H}}$

Lastly, the upper boundary of the scatter plots for different sizes is depicted. So the maximum deviation for each analyzed room size can be appreciated.

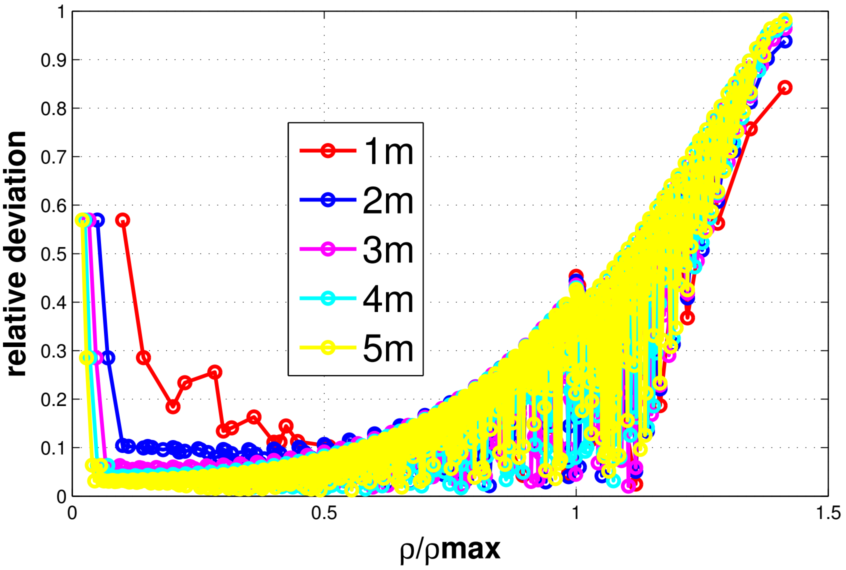


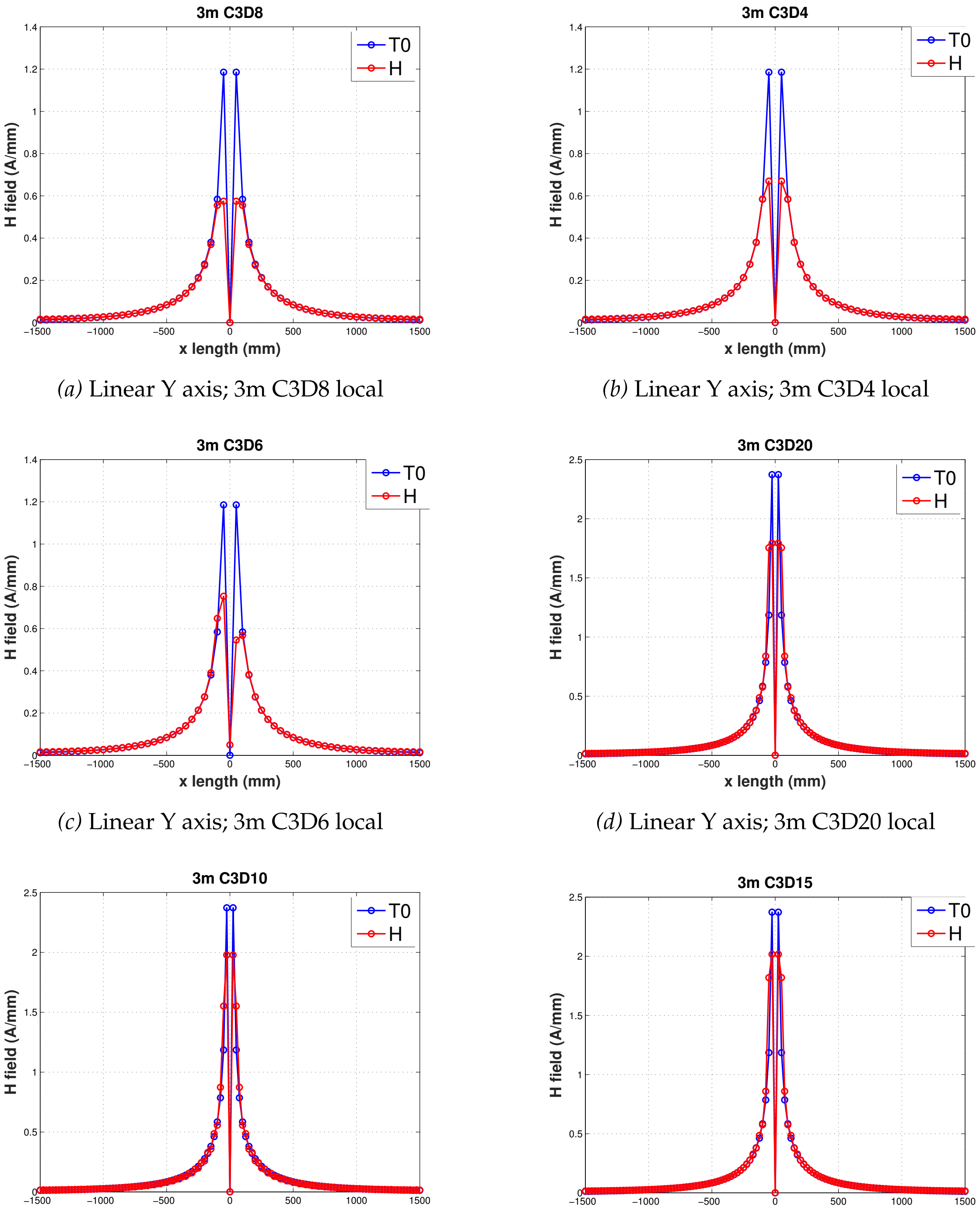
Image 6.19: Upper boundaries of different room sizes

6.5 Element type analysis

In this section, a local analysis of the influence of the element type is shown, as well as a global analysis in form of a bar graph which summarizes the results of the performed simulations.

$CubeSize_{Air}(m)$	$EType_{Air}$	$ESize_{Air}(cm)$	$EType_{Coil}$	Circumference	Drag
3	C3D8	5	S4R	100	50
3	C3D4	5	S4R	100	50
3	C3D6	5	S4R	100	50
3	C3D20	5	S4R	100	50
3	C3D10	5	S4R	100	50
3	C3D15	5	S4R	100	50

Table 6.3: Element type analysis



(a) Linear Y axis; 3m C3D8 local

(b) Linear Y axis; 3m C3D4 local

(c) Linear Y axis; 3m C3D6 local

(d) Linear Y axis; 3m C3D20 local

(e) Linear Y axis; 3m C3D10 local

(f) Linear Y axis; 3m C3D15 local

Image 6.20: Convergence analysis; BiotSavart evaluation for different sizes

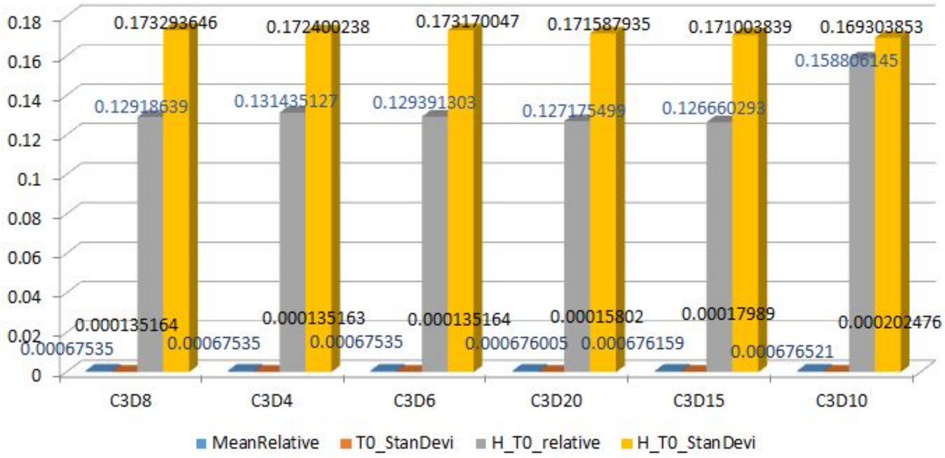


Image 6.21: H T0 Element Type Data table graph

INTENTIONALLY BLANK PAGE

7 Verification of Work Piece

The calculations of $\vec{\mathbf{H}}$ (consequently $\vec{\mathbf{B}}$ too) and $\vec{\mathbf{T}}_{\mathbf{0}}$ by **ccx** have been verified through examples where magneto-static problems with no work piece involved are solved. Such verifications only activates the first equation (3.1) of the gauged A,V-A-P potential formulation implemented in **CalculiX** (speaking about the strong form); this equation represents the null value of the magnetic field divergence, and it is only applied to the air domain or domain Ω_1. Then, three equations of the formulation are left to be verified. These three equations are activated entirely when facing electromagnetic problems with time-variable quantities and a work piece involved, such as the Eddy current problem described in this thesis. Induction heating constitutes a case of Eddy current problem, just the complication of thermal variables is added.

The equations (3.2) and (3.3) are applied to the work piece domain or domain Ω_2. The equation (3.4) is applied in cases with multiple connected bodies to domain Ω_3. This last domain, is the domain that if added to the work piece, transforms it into a simple connected body (initially domain Ω_3 is part of the air domain Ω_1).

Therefore, to go forward with the full verification of the implementation of Maxwell's equations in **CalculiX**, an induction heating problem is considered. The thesis "Magnetic Fields and Induced Power in the Induction Heating of Aluminum Billets "(KENNEDY 2013b) was recommended by Dr. Dhondt to that end. This thesis has verified experimentally a proposed semi-empirical short coil correction factor. Here, the interest is not in the purpose, but in the process followed in such thesis to verify the correction factor. In that process several induction heating problems with different coils and different aluminum workpieces were faced. The problems were simulated with FE models and experimentally performed too. Hence, that thesis provides simulated and experimental data of induction heating problems in addition to the analytic semi-empirical formula which seeks to verify. All can be considered as tools to help in the verification of **CalculiX**. The idea is to replicate one of the FE models used in the referred thesis, and check whether the results provided by **ccx** are close or not to the simulations performed by Dr. Kennedy, to his experimental data, or even to the analytic solution given by the semi-empirical correction factor.

7.1 Objectives

The actual section pursues the following goals:

- To select as benchmark problem for the verification of **CalculiX** one of the induction heating trials carried out in Dr. Kennedy's thesis.
- To build a running FE model of the chosen problem capable of being used in successive simulations with parametrized meshes.
- Meshing of the work piece: in previous sections it has been analyzed the meshing of the coil and the air domain. Now the meshing of the workpiece is analyzed.
- Analysis of the magnetic field, the induced electric field and the induced power.
- Analysis of simple connected and multiple connected work pieces.

7.2 Magnetic Fields and Induced Power in the Induction Heating of Aluminium Billets

(KENNEDY 2013b) is not a standalone thesis, but it is made from three supplements. Neither it is exactly a summary of all three supplements, it is more like a framework to gather and explain the results obtained in them. Indeed, in words of Dr.Kennedy, the main thesis "provides the reader with sufficient background to appreciate the following supplements " (referring to the three supplements) (KENNEDY 2013b, P. IX). Although the main thesis is very explanatory itself, the supplements must be reviewed individually for the sake of **CaluliX**'s verification, because in them important data of the simulations and experimental trials performed can be found.

- **Supplement 1**: Improved Short Coil Correction Factor for Induction Heating of Billets (KENNEDY 2012). Here the induction heating experiments with water-cooled aluminum billets are introduced, as well as the obtained experimental data.
- **Supplement 2**: Analytical and FEM Modeling of Aluminum Billet Inductions Heating with experimental verification (KENNEDY 2013a). This supplement is particularly important because it presents the 2D axial symmetric FE models built in COMSOL used to validate the semi-empirical short coil correction factor. The results of these models will help to verify the performance of **ccx**. However, from the Dr.Kennedy's point of view, this supplement seeks to validate the very COMSOL FE models with analytical and experimental data, so that they are reliable to be used themselves as a tool of validation.

- **Supplement 3**: Empirical verification of a short coil correction factor (KENNEDY 2014). It gathers the data of supplements 1 and 2 plus new additional data. It summarizes all the obtained results and presents the final frequency corrected short coil correction factor.

7.2.1 Short Coil

The shape factor of a coil is defined as follows:

$$Shape\ \ factor : \frac{D_{cs}}{l_c} \tag{7.1}$$

D_{cs} stands for "current sheet" diameter and l_c is the coil's length. A current sheet is an idealized solenoidal coil that has no internal volume. Its electrical current is a surface current, for the the sheet is infinitely thin. The spacing between the turns of the coil is considered infinitely thin too. The supposition of the current sheet is made, so that the diameter of a real solenoidal coil can be defined as the equivalent current sheet diameter, taken on the center line of the conductor.

The coils used in cylindrical geometry induction heating systems, the ones of interest here, have usually a shape factor between 0.5 and 2, and consequently are defined as "short" ones; but what does "short" means in this context?. Taking up equation (2.26), which describes the Ampere's law in its differential form, it can be expressed in its integral form too (the most recognizable form):

$$\int\limits_C \vec{\mathbf{B}} \cdot d\vec{\mathbf{l}} = \mu_0 \cdot \mu_r \int\limits_S \vec{\mathbf{J}} \cdot d\vec{\mathbf{a}} = \mu \cdot I_{enclosed} \tag{7.2}$$

Before moving on, it should be pointed out that, as it was done previously, the term regarding the displacement current is dismissed.

The feasibility of the Ampere's law to calculate accurately the magnetic field inside a solenoidal empty coil (air core) determines whether the coil is defined as "long" or as "short". Typically, a very long coil can be considered theoretically infinite (far from the edges) in order to use equation (7.2) to calculate the magnetic field. In such coils, the magnetic field can be considered funneled to its interior, homogeneously distributed and parallel to the coil axis, which allows to apply the following:

$$B_\infty \cdot l_c = \mu \cdot N_c \cdot I_c \rightarrow B_\infty = \frac{\mu \cdot N_c \cdot I_c}{l_c} \tag{7.3}$$

In equation (7.3), l_c stands for a particular length of a very long coil or infinite coil (length located far from the edges), and N_c is the number of turns of the coil existing in such length.

Then again, Ampere's law can not be so applied in shorter or finite coils, where the magnetic flux inside them is not homogeneously distributed and it is not parallel to the coil axis. Besides, in short coils the exterior magnetic flux is non-zero.

7.2.2 Nagaoka coeficient κ_N

If the inductance of a cylindrical air core coil is known, the magnetic field in the coil axis direction can be calculated through the following equation:

$$L_c = \frac{\phi_c \cdot N_c}{I_c} \rightarrow L_c = \frac{A_c \cdot N_c \cdot B_z}{I_c} \tag{7.4}$$

ϕ_c is the magnetic flux through the coil and A_c is the area of the coil that the magnetic flux goes through.

Lorenz developed a formula to calculate the inductance of the equivalent current sheet of a solenoidal coil (**Kennedy.2013**). Nagaoka reformulated Lorenz's equation using the concept of shape factor. In addition to that, Nagaoka introduces a short coil correction factor with respect of the calculations for long and infinite coils. The short coil correction factor is referred as κ_N (**Kennedy.2013**). Then κ_N allows to calculate B_z in the following way:

$$B_z = \frac{\kappa_N \cdot \mu \cdot N_c \cdot I_c}{l_c} \rightarrow B_z = \kappa_N \cdot B_\infty \tag{7.5}$$

Therefore, the Nagaoka coefficient indicates according to the shape factor of a coil, what percentage has this coil of the z-component magnetic fulx density of the equivalent long coil.

7.2.3 Workpiece and modified Nagaoka coeficient κ_N^*

The introduction inside the coil of a workpiece made of a non-ferromagnetic material, such as aluminum, causes a change in the distribution of the magnetic flux density within the coil, funneling most of flux lines to the air-gap. This change is depicted in 7.1, where it can be seen that the introduction of the workpiece, forces a constant number of lines of flux to occupy a lower air-gap annular area.

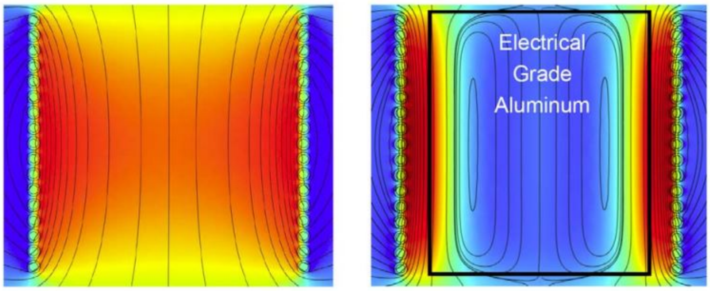


*Image 7.1: Magentic flux density distribution without work piece (air core) and with work piece; extracted from (*Kennedy *2013b, P. 24)*

The coil is fed by AC current, which produces a time varying magnetic field. According to equation (2.25), that describes the Faraday's law of magnetic induction, the alternate magnetic field induces an electric current (Eddy currents) when going through an electric conductor, the workpiece in this case. These Eddy currents, that are time varying too, create in turn another time varying magnetic field, which opposes the source magnetic field of the Eddy currents (Lenz's law). This is the reason why the magnetic flux density is concentrated in the air-gap, the response magnetic field produced by the Eddy currents cancels the flux density inside the workpiece but reinforces the flux density in the air-gap, increasing its value.

Up to what level within the workpiece the Eddy currents are created is determined by the electromagnetic penetration depth, which depends on the electrical conductivity of the workpiece, on its magnetic permeability, and on the frequency of the source alternate magnetic field (which is in phase with the AC current that feeds the coil):

$$\delta_w = \sqrt{\frac{\rho_w}{\pi \mu f}} \tag{7.6}$$

At the surface of the workpiece, according to (Kennedy 2013b, P. 16), the magnetic field intensity is so defined:

$$H_z = \frac{\kappa_N^* \cdot N_c \cdot I_c}{l_c} \tag{7.7}$$

κ_N^* is a modified Nagaoka short coil correction factor, whose definition and validation are the primary targets of Dr.Kennedy's thesis. This correction factor considers the influences of the geometries of both coil and workpiece, their electrical properties, and the frequency of the AC current applied to the coil.

Reviewing all said, it is clear that a coupling exists between the magnetic flux within the coil and the workpiece. This coupling is of high importance in induction heating, for it controls the efficiency of energy transfer between the coil and the workpiece. Northrup describes such coupling as follows (KENNEDY 2013b, P. 23):

$$C = \left[\frac{D_w}{D_c}\right]^2 \cdot 0.8 \tag{7.8}$$

D_w is the outer diameter of the workpiece and D_c is the inner diameter of the coil.

Vaughan and Williamson dealt with the coupling, with an empirically modified Nagaoka short coil correction factor (KENNEDY 2013b, P. 23). They validated empirically the factor at 10 Khz frequency with non-ferromagnetic and high conductive cylindrical workpieces.

$$\kappa_N^* = \kappa_N \cdot \left[1 - \left[\frac{D_w}{D_c}\right]^2\right] + \left[\frac{D_w}{D_c}\right]^2 \tag{7.9}$$

The equation (7.9) present the following traits:

- As usual in this thesis, it assumes for both the coil and workpiece a uniform cylindrical shape.
- The equation was tested at high frequencies because it is formulated for that purpose. It assumes insignificant penetration of the magnetic field into both the coil and the workpiece, which is reasonable for high frequencies and large coils and workpieces. In this assumption is similar to Northrup's coupling factor.
- It constitutes an empirically modified Nagaoka coefficient which is accounted for the fractional volume interior to the coil occupied by the workpiece. Situation that ends in a change of the distribution of the magnetic flux density.

Dr.Kennedy proposes in his thesis a way more accurate to describe the coupling, or in other words, a new short coil correction factor. This new factor is convenient for analytical modeling of cylindrical geometry induction heating systems at both high and low frequency.

$$\kappa_N^* = \kappa_N \cdot \left[1 - \left[\frac{D_w - \delta_w}{D_c + \delta_c}\right]^2\right] + \left[\frac{D_w - \delta_w}{D_c + \delta_c}\right]^2 \tag{7.10}$$

It is defined by his author as "a semi-empirical one dimensional correction factor, to correct for the two dimensional variation in the magnetic flux of a short coil, i.e. radial and axial variation " . The proposed factor adds to the correction of the fractional volume occupied by the workpiece of (7.9), a frequency correction for low frequencies. At low frequency meaningful electromagnetic penetration exists both into the workpiece and the coil, and the assumptions made for (7.8) and (7.9) are not valid. So Kennedy's factor accounts for the significant penetration of the lines of flux into the coil and the workpiece at low frequencies. The resulting estimation with (7.10) of the air-gap magnetic flux density is consequently lower than the one obtained with (7.9).

During Dr.Kennedy's thesis the analytical calculations made using this new factor have been validated. These validations are important for the verification of CalculiX because they represent sources of tested data of induction heating problems.

- Experimental measurements of the magnetic field: The factor (7.10) required empirical verification due to its semi-empirical nature; κ_N, δ_w and δ_c are analytical, but the applied volumetric correction because of the different consideration of low frequencies is empirical.
- Axial symmetric FE models: they are built in COMSOL and result ideal to compare their results with the ones obtained by ccx.
- Induced power in the workpiece (heating rate).
- Total reactive power of the coil-workpiece system.

Experimental data and its corresponding simulated data with the validated COMSOL models are available from cases of four different coils and 3 different workpieces. The cases are analyzed with high and low electric current and high and low cooling water flow. To the purpose of the present thesis (software verification, not empirical validation), one case is chosen in order to build the equivalent FE model in CalculiX. The details are exposed in the following section.

7.3 Conception of the FE model; benchmark problem

The FE model built on CalculiX replicates one of the experimental trials carried out in (KENNEDY 2013b). The data of the workpiece used can be found in table 7.1 and the data of the coil in table 7.2. In addition to that, images are shown that represent the chosen model.

	Workpiece 1
Alloy	A356
Diameter, mm	75.0
Length, mm	130.0
Measured IACS Electrical Conductivity, %	48.4
Penetration depth δ_W (mm) at 50 Hz and 293 K	13.43

*Table 7.1: Experimental workpiece 1; from (*Kennedy *2013b, P. 27)*

	Short Coil 1
Average Diameter, mm	132
Height, mm	106
Diameter to Height Ratio	1.24
Number of Turns	16
Short Coil Correction Factor	0.641
Electrically Determined IACS Conductivity, %	80
Penetration depth δ_c (mm) at 50 Hz and 293 K	10.45
Total Length of Leads, cm	66
Coil Resistance at Maximum Current Including Leads, Ω	0.01047
Coil Inductance, μH	26.38
Mod. Nagaoka Coefficient k_N^* for Work Piece 1	0.720

*Table 7.2: Experimental short coil 1; from (*Kennedy *2013b, P. 28)*

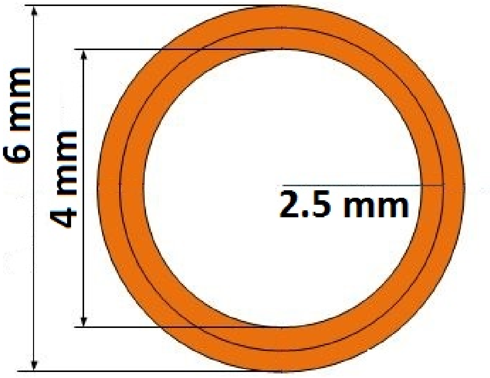


Image 7.2: Coil cross section area

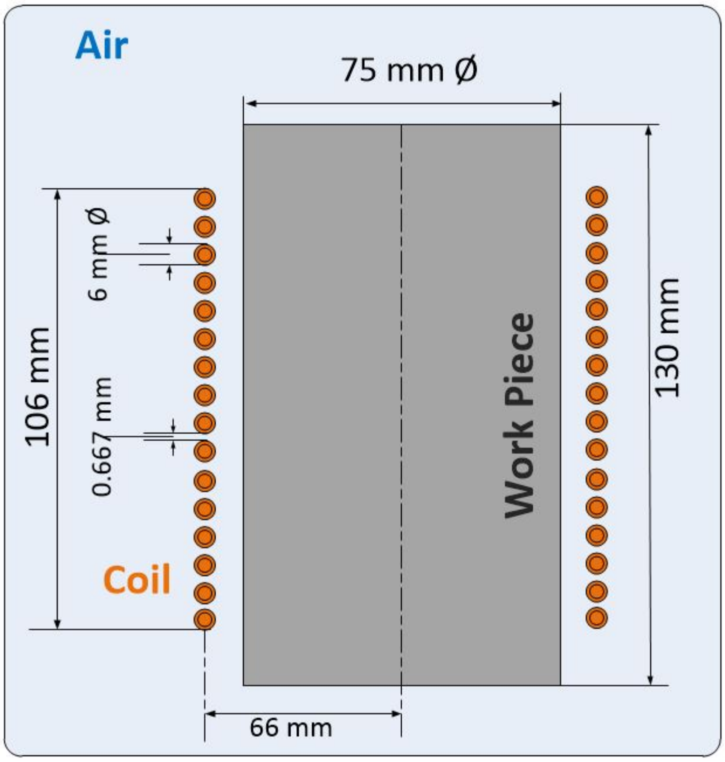


Image 7.3: Coil-Workpiece system; simple connected workpiece (image based on (Kennedy 2014, p. 29))

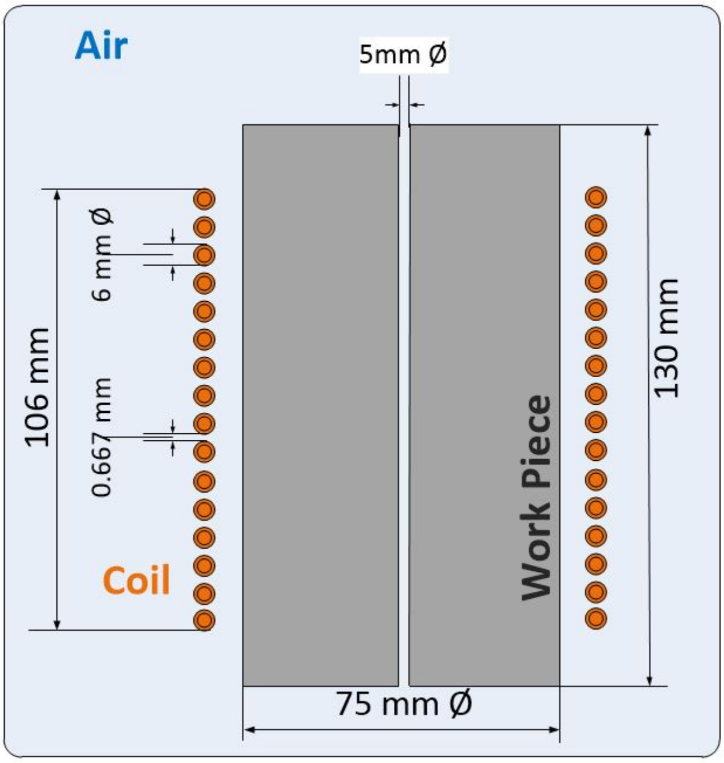


Image 7.4: Coil-Workpiece system; multiple connected workpiece (image based on (Kennedy 2014, p. 29))

To perform this benchmark, the simulation work flow applied for the the $\vec{\mathbf{T}}_{\mathbf{0}}$ and $\vec{\mathbf{H}}$ is used:

- create the geometry to be used
- mesh the geometry
- build the input deck file
- perform the tests
- analyze the outputs
- perform iterations

7.3.1 Create the geometry to be used

In this case, since a different mesh type has to be realized depending on the area near the center piece, the **cgx** package of **Calculix** is used which is much more versatile than Hypermesh to generate the necessary geometry and to mesh it.

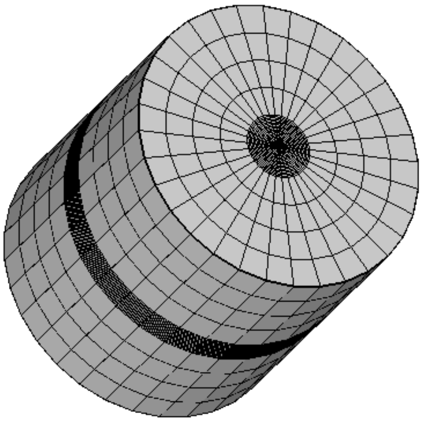
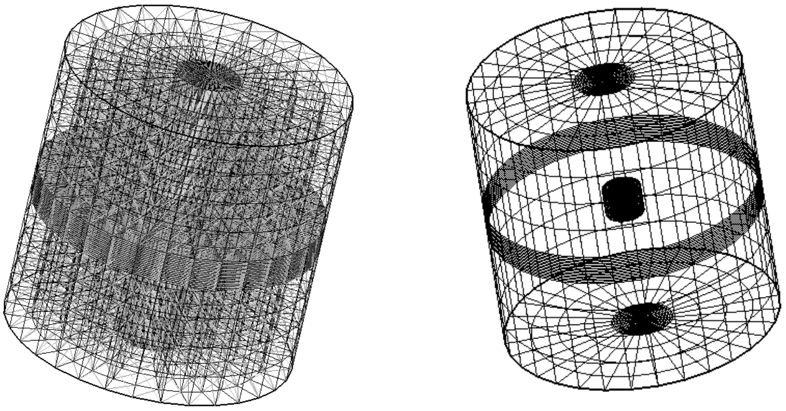


Image 7.5: FE Model

The geometry generated is a cylinder.



(a) Internal Mesh View *(b)* Internal Mesh View with Piece

Image 7.6: Air Cylinder mesh

7.4 Results

The analytic solution according to Biot-Savart is used in the Z axis of the coil-

$$B_Z = \frac{\mu_0 N_c I_c}{2l_c} \left[\frac{x l_c}{\sqrt{(x l_c)^2 + r_{cs}^2}} + \frac{(1-x) l_c}{\sqrt{((1-x) l_c)^2 + r_{cs}^2}} \right] \tag{7.11}$$

The magnetic field calculated by **ccx** in the center of the air-core coil along the z axis shows convergence with the analytic solution by Biot-Savart for solenoidal coil.

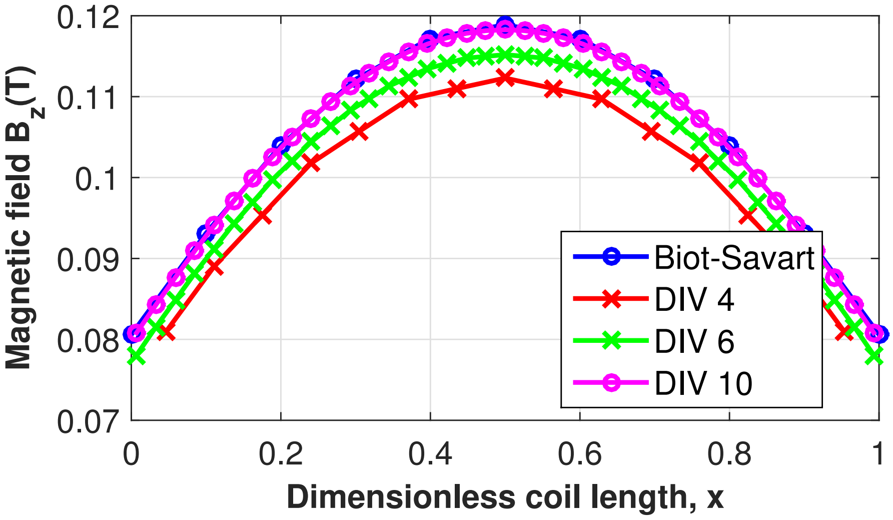


Image 7.7: Air core convergence

However, the meshes used in the last graph should be changed to be able to perform a simulation with the workpiece. There were problems of stability with the solver. The problem is that these new meshes, that work well with the workpiece, do not do so in the air-core case as can be seen in the following image.

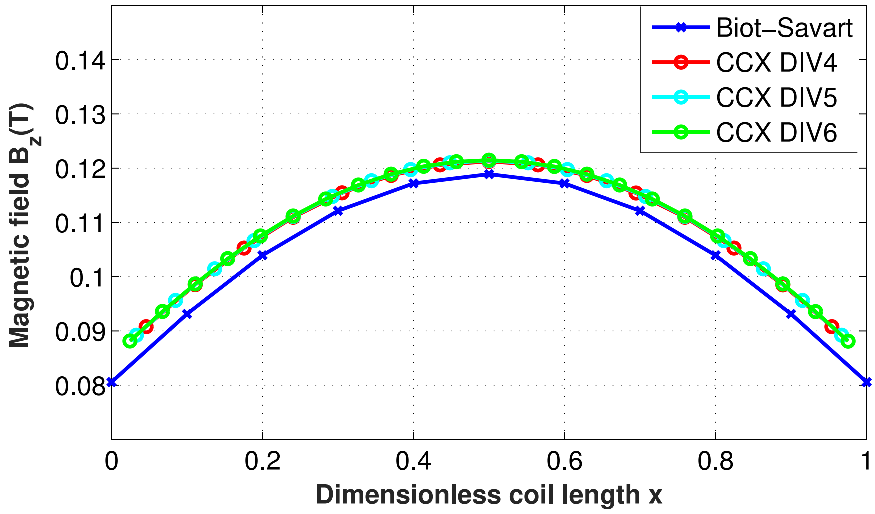


Image 7.8: Air core divergence

At least, thanks to the new meshes, the stability problems are overcome and it can be seen that the physical phenomena is well simulated. In the next graph, it can be appreciated that the introduction of the workpiece increases the magnetic flux density in the air-gap.

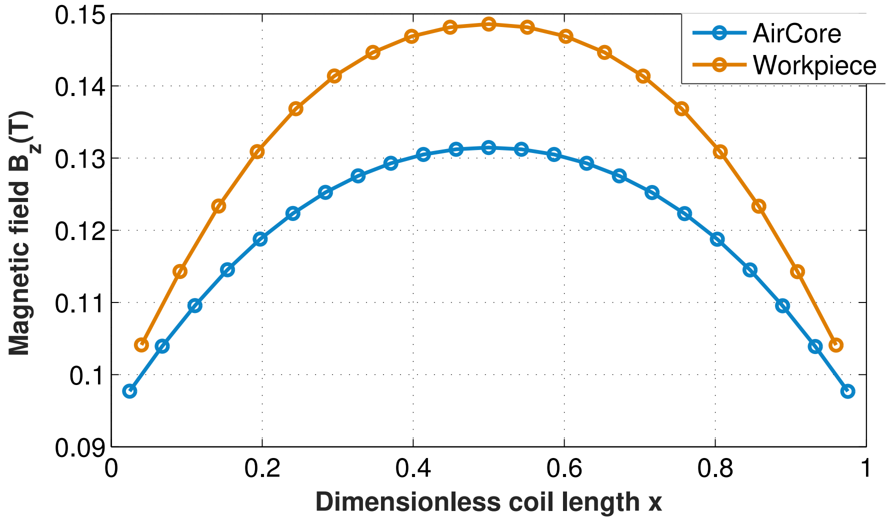


Image 7.9: Air-gap Work Piece

To obtain the heating rates, **CalculiX** does not offer this result by default, and it was required to add some lines of code to a routine and recompile the code. The results obtained were the following. They differ of all the results found in Dr.Kennedy's thesis.

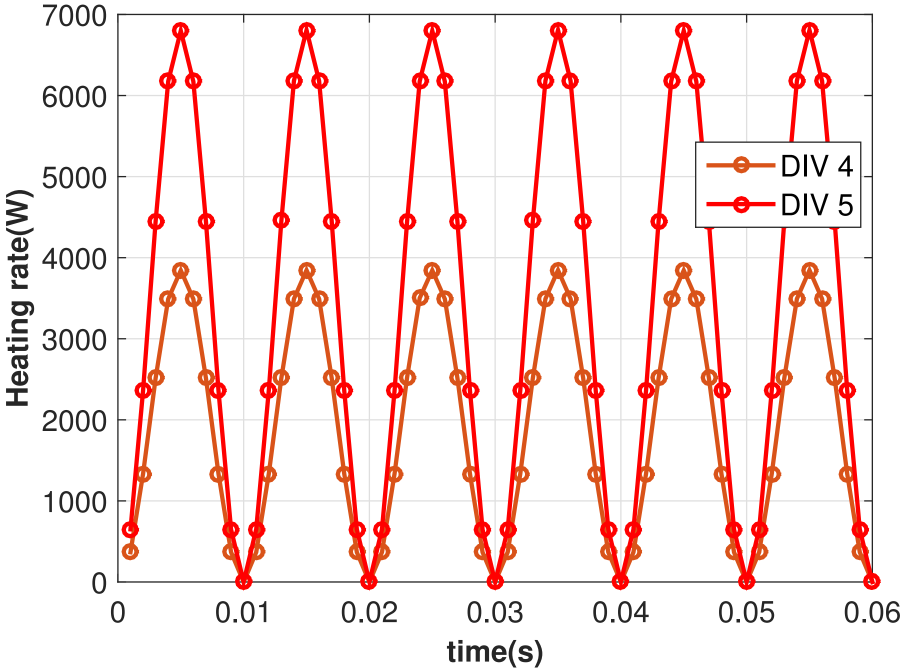


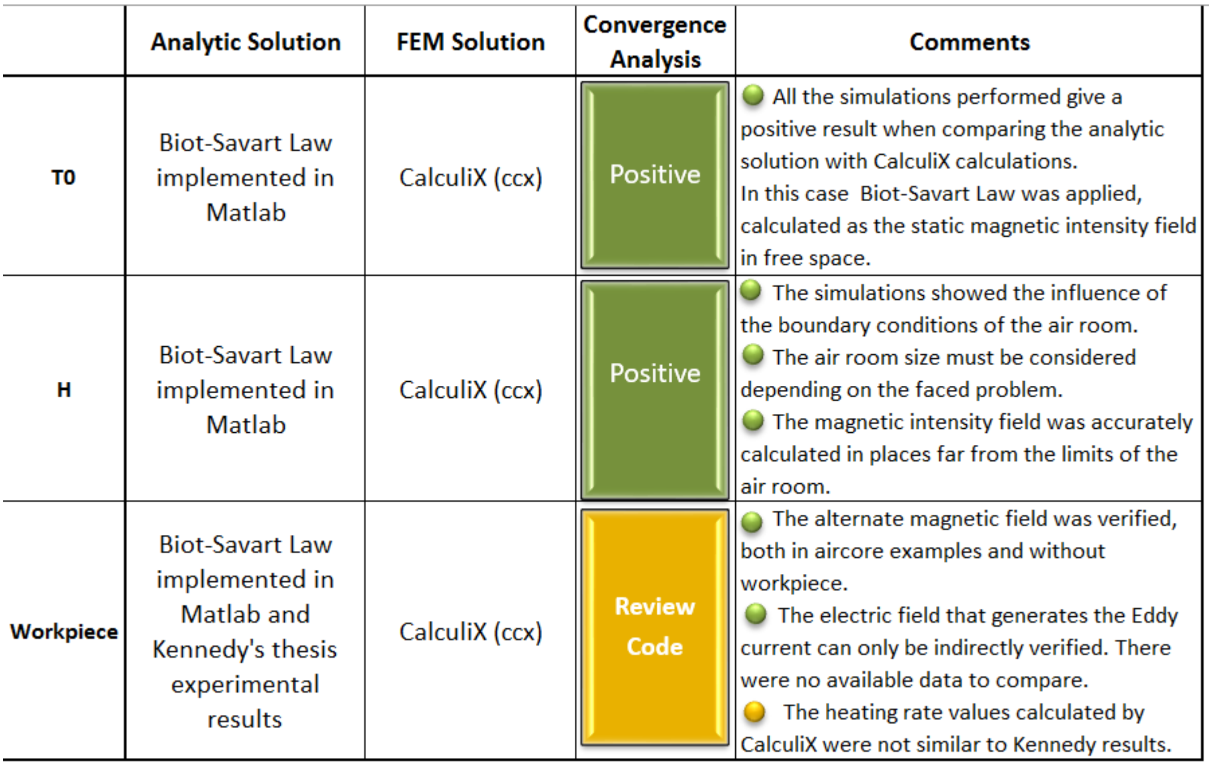
Image 7.10: Heating rate

INTENTIONALLY BLANK PAGE

8 Conclusions and Outlook

Conclusions After all work done, it could be said that **CalculiX** is performing well, only few things have to be review or implemented related to the heating rate calculations.

In figure 8.1 there is a summary of this.



	Analytic Solution	FEM Solution	Convergence Analysis	Comments
T0	Biot-Savart Law implemented in Matlab	CalculiX (ccx)	Positive	🟢 All the simulations performed give a positive result when comparing the analytic solution with CalculiX calculations. In this case Biot-Savart Law was applied, calculated as the static magnetic intensity field in free space.
H	Biot-Savart Law implemented in Matlab	CalculiX (ccx)	Positive	🟢 The simulations showed the influence of the boundary conditions of the air room. 🟢 The air room size must be considered depending on the faced problem. 🟢 The magnetic intensity field was accurately calculated in places far from the limits of the air room.
Workpiece	Biot-Savart Law implemented in Matlab and Kennedy's thesis experimental results	CalculiX (ccx)	Review Code	🟢 The alternate magnetic field was verified, both in aircore examples and without workpiece. 🟢 The electric field that generates the Eddy current can only be indirectly verified. There were no available data to compare. 🟡 The heating rate values calculated by CalculiX were not similar to Kennedy results.

Image 8.1: Summary table

Outlook and future work To get a robust **CalculiX** in this area, some new activities could be planned:

- Review actual **CalculiX** code for the thermal calculation.
- Test new functionalities in **CalculiX**.
- Generate a validation set of data.

- Set up regression test.
- Automate testing processes.
- Implement a quality plan.
- Integrate in **CalculiX** a full trace log functionality to help on future tests.

References

ALTAIR 2015

Altair: *Practical Aspects of Finite Element Simulation: Study Guide*. 3rd Edition. 2015. URL: www.altairuniversity.com.

CHAVES 2007

CHAVES, E.: *Mecánica del Medio Continuo. Conceptos Básicos*. 3Âª Edicion. Centro Internacional de Métodos Numéricos en Ingeniería - CIMNE - Barcelona. 2007. ISBN: 978-84-96736-38-2.

DAVIES 1990

Davies, J.: *Conduction and induction heating*. 1990. ISBN: 978-0-86341-174-8.

DHONDT 2004

Dhondt, G.: *The finite element method for three-dimensional thermomechanical applications*. Chichester: Wiley. 2004. ISBN: 0-470-85752-8. URL: http://www.loc.gov/catdir/bios/wiley047/2004047964.html.

DHONDT 2016a

Dhondt, G.: Calculix: A Free Software Three-Dimensional Structural Finite Element Program. URL: http://www.calculix.de/.

DHONDT 2016b

Dhondt, G.: CalcuLiX CrunchiX User's Manual. URL: http://www.dhondt.de/ccx_2.11.pdf.

DOXYGEN 2016

Doxygen: Generate documentation from source code. Internet download. URL: http://www.stack.nl/~dimitri/doxygen/.

GEIGER 2016

Geiger, B.: Validierung der Berechnung elektromagnetischer Felder mit der FEM-Software "CalculiX". Bachelort Thesis. Germany: MTU Aero engines AG & Hochschule Karlsruhe. 2016.

GRAPHVIZ 2016
Graphviz: Graph Visualization. Internet download. Internet. URL: http://www.graphviz.org/.

HYPERMESH 2016
HyperMesh. Internet download. URL: http://www.altairhyperworks.com/product/HyperMesh.

KENNEDY 2012
Kennedy, M. W.: IMPROVED SHORT COIL CORRECTION FACTOR FOR INDUCTION HEATING OF BILLETS. Licentiate Thesis in Materials Science and Engineering Supplement I. Norway, Sweden: Dept. of Material Science and Engineering, Norwegian University of Science and Technology (NTNU) and 2Dept. of Materials Science and Engineering, Royal Institute of Technology (KTH). 2012. URL: https://www.researchgate.net/publication/256121282_Improved_Short_Coil_Correction_Factor_for_Induction_Heating_of_Billets.

KENNEDY 2013a
Kennedy, M. W.: ANALYTICAL AND FEM MODELING OF ALUMINUM BILLET INDUCTION HEATING WITH EXPERIMENTAL VERIFICATION. Licentiate Thesis in Materials Science and Engineering Supplement II. Norway, Sweden: Dept. of Material Science and Engineering, Norwegian University of Science and Technology (NTNU) and 2Dept. of Materials Science and Engineering, Royal Institute of Technology (KTH). 2013. URL: https://www.researchgate.net/publication/256121171_Analytical_and_FEM_Modeling_of_Aluminum_Billet_Induction_Heating_with_Experimental_Verification.

KENNEDY 2013b
Kennedy, M. W.: Magnetic Fields and Induced Magnetic Fields and Induced Power in the Induction Heating of Aluminium Billets. Licentiate Thesis in Materials Science and Engineering. Sweden: KTH Royal Institute of Technology. 2013. URL: https://www.researchgate.net/publication/256121245_Magnetic_Fields_and_Induced_Power_in_the_Induction_Heating_of_Aluminium_Billets.

KENNEDY 2014
Kennedy, M. W.: Empirical Verification of a Short Coil Correction Factor. Licentiate Thesis in Materials Science and Engineering Supplement III. Sweden: Article in IEEE Transactions on Industrial Electronics Â· May 2014. 2014. URL: https://www.researchgate.net/publication/260541900_Empirical_Verification_of_a_Short-Coil_Correction_Factor.

KUCZMANN 2009
Kuczmann, M.: Potential formulations in magnetics applying the finite element method: Lecture notes. Diss. Hungary: Laboratory of Electromagnetic Fields,"Széchenyi István",University Gyor. 2009. URL: `http://maxwell.sze.hu/docs/C4.pdf`.

LEVI-CIVITA 2005
Levi-Civita, T.: *The Absolute Differential Calculus (Calculus of Tensors)*. Dover Phonix Editions. 2005. ISBN: 978-0486446370.

MAXWELL 1862
Maxwell, J. C.: On physical lines of force. The London, Edinburgh, and Dublin Philosophical Magazine and Journal of Science. (1862).

NAYFEH & BRUSSEL 2015
Nayfeh, M. H.; M. K. Brussel: *Electricity and magnetism*. Dover edition. Mineola New York: Dover Publications Inc. 2015. ISBN: 978-0-486-78971-2.

NOPPER 2011
Nopper, B.: Numerische Simulation zur Optimierung der Induktiven Bauteilerwärmung während der Fertigung unter Berücksichtigung der Fertigungeandbedingungen. Masterarbeit. MTU Aero engines AG & Hochschule Karlsruhe. 2011.

PROF. DR.-ING. W. A. WALL 2015
Prof. Dr.-Ing. W. A. Wall: *Finite Elemente*. Munich: Fachschaft Maschinenbau. 2015.

PURCELL et al. 1994
Purcell, E. M.; C. Kittel; M. Pujal Carrera: *Berkeley physics course*. 2. ed. Barcelona: Reverté. 1994. ISBN: 978-84-291-4319-5.

TIPLER 2010
Tipler Paul Allen; Mosca, G.: *Fisica para la ciencia y la tecnologa*. 6a ed. Barcelona: Reverte. 2010. ISBN: 978-8429144307.

WITTIG 2017
Wittig, K.: CalculiX Graphics UserÂ´s Manual. Germany. URL: `http://www.dhondt.de/`.

ZANGWILL 2013
Zangwill, A.: *Modern electrodynamics*. 2013. ISBN: 978-0-521-89697-9.

ZUREKS 2009
Zureks: *Magnetic permeability graph*. open source. 2009. URL: `https://commons.wikimedia.org/wiki/File:Permeability_by_Zureks.svg`.

INTENTIONALLY BLANK PAGE

Appendix

A.1 Vectorial calculus

The mathematical theory behind the electromagnetic fields was developed and unified by J. C. Maxwell through the use of the vectorial calculus. In the following section, the main tools of vectorial calculus needed to understand the electromagnetic field theory within the context of this thesis are presented.

Divergence of a vectorial function The divergence is a local property of a vectorial field, in fact it is a scalar magnitude that can be defined in each point of the field. Its definition can be developed from a surface integral over a closed surface. Conceptually, it is the mathematical way to express if a particular entity is a source or a sink of a vectorial field. Being $\vec{\mathbf{F}}$ a general vectorial function, its divergence is defined in each spatial point as follows (PURCELL et al. 1994, P. 55):

$$\nabla \cdot \vec{\mathbf{F}} = \lim_{V_i \longrightarrow 0} \frac{1}{V_i} \int_{S_i} \vec{\mathbf{F}} \cdot d\vec{\mathbf{S}}_{\mathbf{i}} \tag{A.1}$$

Where V_i is a volume which includes the point, and S_i is the surface that delimits such volume. As already mentioned, the divergence so defined is a scalar magnitude.

Mathematically speaking, $\nabla \cdot \vec{\mathbf{F}}$ can be expressed in the following way: divergence of $\vec{\mathbf{F}}$ is the outflow of V_i per unit volume (the net flux can be described as outflow or inflow), in the limit of an infinitesimal V_i.

The divergence of a vectorial function expressed in Cartesian coordinates looks as follows(PURCELL et al. 1994, P. 59):

$$\nabla \cdot \vec{\mathbf{F}} = \frac{\partial F_x}{\partial x} + \frac{\partial F_y}{\partial y} + \frac{\partial F_z}{\partial z} \tag{A.2}$$

Curl of vectorial function If the divergence is explained from a surface integral over a closed surface, analogously the curl can be explained from a line integral along a closed curve (cyclic integral). Being $\vec{\mathbf{F}}$ a general vectorial function, its curl is a vector defined in a spatial point of the field as follows(PURCELL et al. 1994, P. 65):

$$(\nabla \times \vec{\mathbf{F}}) \cdot \vec{\mathbf{n}} = \lim_{a_i \longrightarrow 0} \frac{\Gamma_i}{a_i} = \lim_{a_i \longrightarrow 0} \frac{\int_{C_i} \vec{\mathbf{F}} \cdot d\vec{\mathbf{l}}}{a_i} \tag{A.3}$$

Where a_i is an area which includes the point where the curl is calculated, and Γ_i is the result of the line integral along a closed curve or cycle, known as C_i, which delimits the area a_i. The curl is defined as a vector, thus what is represented in the last formula is how to calculate a component of this vector. The limit depicted in the formula is a scalar magnitude, which is associated to the point in the vectorial field $\vec{\mathbf{F}}$, where its curl is calculated. The vector $\vec{\mathbf{n}}$ is used for a dot product, in order to define one of the components of the curl, the vector curl of $\vec{\mathbf{F}}$. If $\vec{\mathbf{n}}$ is chosen as each one of the Cartesian directions, the formula above delivers three different numbers, which correspond to the three components of the vector curl of $\vec{\mathbf{F}}$.

The direction of the vector curl of $\vec{\mathbf{F}}$ in each point of the space, is normal to the plane which contains the point and where the value of the cycle integral is maximum. Its module is the value of this integral in the limit of an infinitesimal a_i per surface unit, in the plane around the point where the curl is calculated.

Laplace's equation It is a equation in partial derivatives of second order of elliptic type (PURCELL et al. 1994, P. 64).

$$\nabla^2 \varphi = 0 \tag{A.4}$$

Poisson's equation Laplace's equation is actually a particular case of Poisson's equation. The Poisson's equation is defined equal to a function (PURCELL et al. 1994, P. 64).

$$\nabla^2 \varphi = f \tag{A.5}$$

Vector calculus identities In the following chapters some common vectorial identities will be applied to develop the calculations (CHAVES 2007) (NAYFEH & BRUSSEL 2015).

$$\nabla \cdot (\varphi \vec{\mathbf{F}}) = \varphi \nabla \cdot \vec{\mathbf{F}} + (\nabla \varphi)\vec{\mathbf{F}} \tag{A.6}$$

$$\nabla \cdot (\vec{\mathbf{F}} \times \vec{\mathbf{G}}) = (\nabla \times \vec{\mathbf{F}})\vec{\mathbf{G}} - \vec{\mathbf{F}}(\nabla \times \vec{\mathbf{G}}) \tag{A.7}$$

$$\nabla \times (\varphi \vec{\mathbf{F}}) = \varphi \nabla \times \vec{\mathbf{F}} + (\vec{\nabla} \varphi \times \vec{\mathbf{F}}) \tag{A.8}$$

$$(\vec{\mathbf{F}} \times \vec{\mathbf{G}}) \cdot \vec{\mathbf{n}} = \vec{\mathbf{F}} \cdot (\vec{\mathbf{G}} \times \vec{\mathbf{n}}) \tag{A.9}$$

A.1.0.1 Second derivative rules

φ is a scalar function and $\vec{\mathbf{F}}$ is a vectorial function (CHAVES 2007) (NAYFEH & BRUSSEL 2015).

$$\nabla \cdot (\nabla \times \vec{\mathbf{F}}) = 0 \tag{A.10}$$

$$\nabla \times (\vec{\nabla} \varphi) = 0 \tag{A.11}$$

$$\nabla \times (\nabla \times \vec{\mathbf{F}}) = \nabla(\nabla \cdot \vec{\mathbf{F}}) - \nabla^2 \vec{\mathbf{F}} \tag{A.12}$$

Vectorial relations

A.1.0.2 Gauss theorem

The Gauss theorem allows to transform between a surface integral and volume integral (CHAVES 2007).

$$\int_S \vec{\mathbf{F}} \cdot d\vec{\mathbf{S}} = \int_V (\nabla \cdot \vec{\mathbf{F}}) dV \tag{A.13}$$

A.1.0.3 Stoekes theorem

The Stokes theorem allows to transform between a line integral and a surface integral (PURCELL et al. 1994, P. 70).

$$\int_{curve} \vec{\mathbf{F}} \cdot \vec{\mathbf{dl}} = \int_S (\nabla \times \vec{\mathbf{F}}) \cdot d\vec{\mathbf{S}} \tag{A.14}$$

A.1.0.4 Gradient

(Nayfeh & Brussel 2015)

$$\varphi_2 - \varphi_1 = \int\limits_{curve} \vec{\nabla}\varphi \cdot \vec{\mathbf{dl}} \tag{A.15}$$

A.2 Tensorial calculus

The mathematical tools of vectorial calculus just covered are needed to understand the Maxwell's equations. To solve them numerically the FEM is applied. Following the standard FE theory, the Maxwell's equations in its differential form (the potential formulation form), known within the context of the FE as strong form, are developed into an integral form, known in the FE theory as the weak form. The final solutions are obtained from a set of matrix equations systems, which are derived from this weak form. In order to get the equations systems from the weak form, because of the vectorial nature of the variables existing in the integrals, tensorial calculus is needed to perform such mathematical development.

The tensorial calculus deals with tensors, which represent physical quantities that must appear the same to all observes, i.e. independent of the selected coordinate systems (CHAVES 2007). The concept is similar to the one of the vectors, the components of a vector are coordinate system dependent, as they are the components of a tensor, whereas the vector itself is independent of the coordinate system, as it is a tensor. Indeed, as it will be now explained, a vector is a type of tensor.

Tensors can be classified according to what it is known as the rank of a tensor. The rank of a tensor is defined as the number of index needed to specify without ambiguity a component of the tensor. The classification is as follows:

Scalar (tensor of rank 0): quantity defined as a magnitude without direction. It has only one component.

Vector (tensor of rank 1): It has both magnitude and direction and three components in 3D (two components in in 2D).

Dyad (tensor of rank 2): It has magnitude and two directions. It possess nine components, represented as a 3x3 matrix.

Triad (tensor of rank 3): It has magnitude and three directions. It possess twenty-seven components. It can not be represented as a matrix.

Between the components of a tensor and its rank a rule can be established in 3D space: a tensor of rank **n** has 3^n components.

Regarding the tensors of rank 1, the vectors, it should be added that position vectors are not tensors themselves (position vectors are not coordinate independent), while the difference between any two position vectors is truly a tensor of rank 1.

Einstein convention or summation convention Under thee traditional orthonormal rectangular coordinate system, a generic vector can be represented as follows (CHAVES 2007):

$$\vec{\mathbf{F}} = F_x\vec{\mathbf{i}} + F_y\vec{\mathbf{j}} + F_z\vec{\mathbf{k}} \tag{A.16}$$

By changing the notation of the unit vectors of the orthonormal base, it is obtained:

$$\vec{\mathbf{F}} = F_1\vec{\mathbf{e_1}} + F_2\vec{\mathbf{e_2}} + F_3\vec{\mathbf{e_3}} \tag{A.17}$$

Then it is possible to express the mathematical representation of the vector through a sum:

$$\vec{\mathbf{F}} = F_1\vec{\mathbf{e_1}} + F_2\vec{\mathbf{e_2}} + F_3\vec{\mathbf{e_3}} = \sum_{i=1}^{3} F_i\vec{\mathbf{e_i}} \tag{A.18}$$

Another way used to express the last formula in tensorial calculus is by applying the Einstein convention or summation convention. According to such convention, the repetition of an index (as i in the last formula) implies its summation:

$$\vec{\mathbf{F}} = F_1\vec{\mathbf{e_1}} + F_2\vec{\mathbf{e_2}} + F_3\vec{\mathbf{e_3}} = F_i\vec{\mathbf{e_i}} \quad (i = 1, 2, 3) \tag{A.19}$$

The summation convention was introduced by Albert Einstein in 1916, and it constitutes the starting point of the indicial notation, which will be used in this thesis when applying tensorial calculus.

Indicial notation By indicial notation F_i represents not a single value, but i values. The sub-indexes of this notation are of two kinds (CHAVES 2007):

- Free subindex: they only appear once in a term of an expression. The number of free sub-indexes indicate the rank of the tensor.
- Dummy subindex: they are repeated more than once in a expression indicating summation.

Here it should be remarked that by indicial notation, a subindex in a term of an expression can appear only two times. In case it appears three or more times, the expression results incorrect.

The importance of the indicial notation can be appreciated later on when developing the integrals of the weak form. In tensorial calculus, to manipulate algebraical expressions without using the indicial notation, can be really difficult, because of the great amount of terms that might take part.

Kronecker delta Known as substitution operator will be used in this thesis too, it is defined as follows (CHAVES 2007):

$$\delta_{ij} = \begin{cases} 1 & if \quad i = j \\ 0, & if \quad i \neq j \end{cases} \tag{A.20}$$

Accordingly, if the sub-indexes are equal, a 1 is obtained, otherwise a 0. The definition of Kronecker delta can be used to represent the scalar product of two orthonormal bases, i.e. the scalar product between each of the unit vectors which constitute the base:

$$\vec{\mathbf{e_i}} \cdot \vec{\mathbf{e_j}} = \begin{pmatrix} \vec{\mathbf{e_1}} \cdot \vec{\mathbf{e_1}} & \vec{\mathbf{e_1}} \cdot \vec{\mathbf{e_2}} & \vec{\mathbf{e_1}} \cdot \vec{\mathbf{e_3}} \\ \vec{\mathbf{e_2}} \cdot \vec{\mathbf{e_1}} & \vec{\mathbf{e_2}} \cdot \vec{\mathbf{e_2}} & \vec{\mathbf{e_2}} \cdot \vec{\mathbf{e_3}} \\ \vec{\mathbf{e_3}} \cdot \vec{\mathbf{e_1}} & \vec{\mathbf{e_3}} \cdot \vec{\mathbf{e_2}} & \vec{\mathbf{e_3}} \cdot \vec{\mathbf{e_3}} \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} = \delta_{ij} \tag{A.21}$$

The reason why Kronecker delta is called substitution operator as well, it is due to the following property (remenber $\vec{\mathbf{F}}$ is a generic vector):

$$\delta_{ij} F_i = \delta_{1j} V_1 + \delta_{2j} V_2 + \delta_{3j} V_3 \tag{A.22}$$

Notice that in the last formula the Einstein convention is being used in indicial notation(repetition of a subindex indicates sum). Besides, in this last formula, the subindex i is the dummy one and the subindex j is the free one, therefore, substituting the values of Kronecker delta:

$$\left.\begin{aligned} \delta_{ij} F_i &= \delta_{11} V_1 + \delta_{21} V_2 + \delta_{31} V_3 = F_1 \quad (j = 1) \\ \delta_{ij} F_i &= \delta_{12} V_1 + \delta_{22} V_2 + \delta_{32} V_3 = F_2 \quad (j = 2) \\ \delta_{ij} F_i &= \delta_{13} V_1 + \delta_{23} V_2 + \delta_{33} V_3 = F_3 \quad (j = 3) \end{aligned}\right\} \rightarrow \delta_{ij} F_i = F_j \tag{A.23}$$

It can be seen that in the presence of the Kronecker delta (substitution operator), the repeated subindex is replaced. In this case, the repeated subindex i is replaced by the subindex j.

Permutation symbol; Levi-Civita The permutation symbol represents the components of the pseudo-tensor known as Levi-Civita. Its definition is as follows (CHAVES 2007):

$$\epsilon_{ijk} = \begin{cases} +1 & if \quad (i,j,k) \in (1,2,3),(2,3,1),(3,1,2) \\ 0, & \text{all other cases} \quad (i=j) \quad or \quad (j=k) \quad or \quad (i=k) \\ -1, & if \quad (i,j,k) \in (3,2,1),(1,3,2),(2,1,3) \end{cases} \tag{A.24}$$

Another way to express the Levi-Civita tensor is through its sub-indexes:

$$\epsilon_{ijk} = \frac{1}{2}(i-j)(j-k)(k-i) \tag{A.25}$$

In addition to that, there is a graphic approach to the definition of the Levi-Civita tensor, which facilitates its interpretation. The following image illustrates such approach:

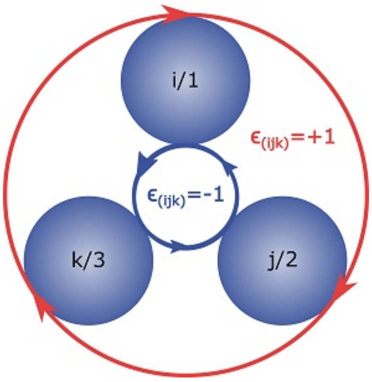


Image A.1: Graphic rule for Levi-Civita

If the sub-indexes of ϵ_{ijk} are ordered clockwise, then its value is 1, but if they are ordered counter-clockwise, its value is -1.

$$\epsilon_{ijk} = \epsilon_{jki} = \epsilon_{kij} \tag{A.26}$$

$$\epsilon_{ijk} = -\epsilon_{ikj} = -\epsilon_{jik} = -\epsilon_{kji} \tag{A.27}$$

The relation between the Levi-Civita tensor and Kronecker delta is important for the tensorial calculus:

$$\epsilon_{ijk} = \begin{vmatrix} \delta_{1i} & \delta_{1j} & \delta_{1k} \\ \delta_{2i} & \delta_{2j} & \delta_{2k} \\ \delta_{3i} & \delta_{3j} & \delta_{3k} \end{vmatrix} = \begin{vmatrix} \delta_{1i} & \delta_{2i} & \delta_{3i} \\ \delta_{1j} & \delta_{2j} & \delta_{3j} \\ \delta_{1k} & \delta_{2k} & \delta_{3k} \end{vmatrix} \tag{A.28}$$

That relation allows to express the product of two Levi-Civita tensors as follows:

$$\epsilon_{ijk}\epsilon_{pqr} = \begin{vmatrix} \delta_{1i} & \delta_{2i} & \delta_{3i} \\ \delta_{1j} & \delta_{2j} & \delta_{3j} \\ \delta_{1k} & \delta_{2k} & \delta_{3k} \end{vmatrix} \begin{vmatrix} \delta_{1p} & \delta_{1q} & \delta_{1r} \\ \delta_{2p} & \delta_{2q} & \delta_{2r} \\ \delta_{3p} & \delta_{3q} & \delta_{3r} \end{vmatrix} \rightarrow \epsilon_{ijk}\epsilon_{pqr} = \begin{vmatrix} \delta_{ip} & \delta_{iq} & \delta_{ir} \\ \delta_{jp} & \delta_{jq} & \delta_{jr} \\ \delta_{kp} & \delta_{kp} & \delta_{kr} \end{vmatrix} \tag{A.29}$$

In order to understand how each term of the last determinant has been obtained, it is worth to have a look at the operation made to get δ_{ip}. Bearing in mind a determinants rule that apply for two equal square matrices: $det(A)det(B) = det(AB)$; then δ_{ip} can be obtained as follows:

$$\delta_{1i}\delta_{1p} + \delta_{2i}\delta_{2p} + \delta_{3i}\delta_{3p} = \delta_{mi}\delta_{mp} = \delta_{ip} \tag{A.30}$$

In the expression above, first was applied the Einstein convention (A.19), and then the substitution property of the Kronecker delta (A.23). The rest of the terms of the final determinant of expression (A.29) are obtained analogously.

A particular case of the expression (A.29), where r=k, delivers a useful equation widely used in tensorial calculus:

$$\epsilon_{ijk}\epsilon_{pqk} = \delta_{ip}\delta_{jq} - \delta_{iq}\delta_{jp} \quad (i, j, k, p, q = 1, 2, 3) \tag{A.31}$$

A.3 Type of analisys: Governing equation evolution

A.3.0.1 First Step: use standard shape functions

Governing equation (3.47)

1-Term: $(\nabla \times \delta\vec{\mathbf{A}})$ $\Rightarrow$ As $(\nabla \times \delta\vec{\mathbf{A}})$ is the rotational of $\delta\vec{\mathbf{A}}$, doing basic development for a 3x3 (CHAVES 2007)

$$\begin{aligned}
(\nabla \times \delta\vec{\mathbf{A}}) &= \begin{vmatrix} \vec{\mathbf{I}} & \vec{\mathbf{J}} & \vec{\mathbf{K}} \\ \dfrac{\partial}{\partial I} & \dfrac{\partial}{\partial J} & \dfrac{\partial}{\partial K} \\ \delta\vec{\mathbf{A}}_{\mathbf{I}} & \delta\vec{\mathbf{A}}_{\mathbf{J}} & \delta\vec{\mathbf{A}}_{\mathbf{K}} \end{vmatrix} \\
&= (\frac{\partial\delta\vec{\mathbf{A}}_{\mathbf{K}}}{\partial J} - \frac{\partial\delta\vec{\mathbf{A}}_{\mathbf{J}}}{\partial K})\vec{\mathbf{I}} + (\frac{\partial\delta\vec{\mathbf{A}}_{\mathbf{I}}}{\partial K} - \frac{\partial\delta\vec{\mathbf{A}}_{\mathbf{K}}}{\partial I})\vec{\mathbf{J}} + (\frac{\partial\delta\vec{\mathbf{A}}_{\mathbf{J}}}{\partial I} - \frac{\partial\delta\vec{\mathbf{A}}_{\mathbf{I}}}{\partial J})\vec{\mathbf{K}} \qquad (A.32) \\
&= (\delta\vec{\mathbf{A}}_{K,J} - \delta\vec{\mathbf{A}}_{J,K})\vec{\mathbf{I}} + (\delta\vec{\mathbf{A}}_{I,K} - \delta\vec{\mathbf{A}}_{K,I})\vec{\mathbf{J}} + (\delta\vec{\mathbf{A}}_{J,I} - \delta\vec{\mathbf{A}}_{I,J})\vec{\mathbf{K}} \\
&= \epsilon_{IJK}\delta\vec{\mathbf{A}}_{J,K} = \epsilon_{IJK}(\delta\vec{\mathbf{A}}_{J})_{,K} \\
&= \epsilon_{IJK}\sum_i \varphi_{i,J}(\delta\vec{\mathbf{A}}_i)_{,K}
\end{aligned}$$

Having applied:

- Nomenclature in derivatives: " , " and Levi Civita's permutations of tensors
- Looking at 3x3 case

$$\epsilon_{ijk} = \begin{cases} +1 & (i,j,k) \in \begin{cases} (1,2,3) \\ (2,3,1) \\ (3,1,2) \end{cases} \\ -1 & (i,j,k) \in \begin{cases} (1,3,2) \\ (3,2,1) \\ (2,1,3) \end{cases} \\ 0 & others \end{cases} \qquad (A.33)$$

- and to

$$\begin{cases} c_1 = \epsilon_{123}\delta\vec{\mathbf{A}}_{2,3} + \epsilon_{132}\delta\vec{\mathbf{A}}_{3,2} \\ c_2 = \epsilon_{231}\delta\vec{\mathbf{A}}_{3,1} + \epsilon_{213}\delta\vec{\mathbf{A}}_{1,3} \\ c_3 = \epsilon_{312}\delta\vec{\mathbf{A}}_{1,2} + \epsilon_{321}\delta\vec{\mathbf{A}}_{3,1} \end{cases} \qquad (A.34)$$

$$\begin{cases} c_1 = \epsilon_{1jk}\delta\vec{\mathbf{A}}_{j,k} \\ c_2 = \epsilon_{2jk}\delta\vec{\mathbf{A}}_{j,k} \\ c_3 = \epsilon_{3jk}\delta\vec{\mathbf{A}}_{j,k} \end{cases} \qquad (A.35)$$

- then

$$c_i = \epsilon_{ijk}\delta\vec{\mathbf{A}}_{j,k} \qquad (A.36)$$

- Nomenclature: $\delta\vec{\mathbf{A}}_{J;K} \longleftrightarrow (\delta\vec{\mathbf{A}}_J)_{,K}$
- Discretization of the solution using shape functions:
  - $\vec{\mathbf{A}} = \sum_i \varphi_i(\xi,\eta,\zeta) \cdot \vec{\mathbf{A}}_i$
  - $\vec{\mathbf{A}}_K = \sum_i \varphi_i(\xi,\eta,\zeta) \cdot \vec{\mathbf{A}}_{iK}$
  - $\vec{\mathbf{A}}_{K,L} = \sum_i \varphi_{i,L}(\xi,\eta,\zeta) \cdot \vec{\mathbf{A}}_{iK}$

2-Term: $(\nabla \times \vec{\mathbf{A}})$ $\Rightarrow$ Applying the same steps as previous part of the term 1 of equation. Doing basic development for a 3x3, then

$$(\nabla \times \vec{\mathbf{A}}) = \begin{vmatrix} \vec{\mathbf{I}} & \vec{\mathbf{J}} & \vec{\mathbf{K}} \\ \dfrac{\partial}{\partial I} & \dfrac{\partial}{\partial J} & \dfrac{\partial}{\partial K} \\ \vec{\mathbf{A}}_{\mathbf{I}} & \vec{\mathbf{A}}_{\mathbf{J}} & \vec{\mathbf{A}}_{\mathbf{K}} \end{vmatrix} \tag{A.37}$$

So,

$$\nabla \times \vec{\mathbf{A}} = \epsilon_{ILM} \sum_j \varphi_{i,L} (\vec{\mathbf{A}}_j)_{,M} \tag{A.38}$$

3-Term: $(\nabla \times \delta\vec{\mathbf{A}}) \cdot (\nabla \times \vec{\mathbf{A}})$ $\Rightarrow$ Expressing the permutation symbol of Levi Civita in function of the Delta of Kroneker (CHAVES 2007)

$$\delta_{ij} = \begin{cases} 1 & if i = j \\ 0 & if i \neq = j \end{cases} \tag{A.39}$$

$$\epsilon_{ijk} = \epsilon_{nlm}\delta_{ni}\delta_{lj}\delta_{mk} \tag{A.40}$$

Once developed, $\epsilon_{ijk}\epsilon_{nlm}$ can be expressed as the product of two determinants .

And how $|AB| = |A||B|$, then

$$\epsilon_{ijk}\epsilon_{nlm} = \begin{vmatrix} \delta_{in} & \delta_{il} & \delta_{im} \\ \delta_{jn} & \delta_{jl} & \delta_{jm} \\ \delta_{kn} & \delta_{kl} & \delta_{km} \end{vmatrix} \tag{A.41}$$

Assuming that i = n

$$\epsilon_{ijk}\epsilon_{ilm} = \delta_{jl}\delta_{km} - \delta_{jm}\delta_{kl} witki, j, k, l, m = 1, 2, 3 \tag{A.42}$$

so that

$$
\begin{aligned}
&(\nabla \times \delta\vec{\mathbf{A}}) \cdot (\nabla \times \vec{\mathbf{A}}) = \epsilon_{IJK} \sum_i \varphi_{i,J} (\delta\vec{\mathbf{A}}_i)_{,K} \epsilon_{ILM} \sum_j \varphi_{i,L} (\vec{\mathbf{A}}_j)_{,M} \\
&= \epsilon_{IJK}\epsilon_{ILM} (\sum_i \sum_j \varphi_{i,J} (\delta\vec{\mathbf{A}}_i)_{,K} \varphi_{i,L} (\vec{\mathbf{A}}_j)_{,M}) \\
&= (\delta_{JL}\delta_{KM} - \delta_{JM}\delta_{KL}) (\sum_i \sum_j \varphi_{i,J} (\delta\vec{\mathbf{A}}_i)_{,K} \varphi_{i,L} (\vec{\mathbf{A}}_j)_{,M}) \\
&= \delta_{JL}\delta_{KM} (\sum_i \sum_j \varphi_{i,J} (\delta\vec{\mathbf{A}}_i)_{,K} \varphi_{i,L} (\vec{\mathbf{A}}_j)_{,M}) - \delta_{JM}\delta_{KL} (\sum_i \sum_j \varphi_{i,J} (\delta\vec{\mathbf{A}}_i)_{,K} \varphi_{i,L} (\vec{\mathbf{A}}_j)_{,M}) \\
&= \sum_i \sum_j (\varphi_{i,L} \delta_{KM} (\delta\vec{\mathbf{A}}_i)_{,K} \varphi_{j,L} (\vec{\mathbf{A}}_j)_{,M} - \varphi_{i,M} (\delta\vec{\mathbf{A}}_i)_{,K} \varphi_{j,K} (\vec{\mathbf{A}}_j)_{,M})
\end{aligned}
\tag{A.43}
$$

Putting all together:

$$
\begin{aligned}
&\int_{\Omega_2} \frac{1}{\mu} (\nabla \times \delta\vec{\mathbf{A}}) \cdot (\nabla \times \vec{\mathbf{A}}) dV = \sum_e \int_{V_{Oe2}} \frac{1}{\mu} (\nabla \times \delta\vec{\mathbf{A}}) \cdot (\nabla \times \vec{\mathbf{A}}) dV_e \\
&= \sum_e \sum_i \sum_j \int_{V_{Oe2}} \frac{1}{\mu} [\varphi_{i,L} \delta_{KM} (\delta\vec{\mathbf{A}}_i)_{,K} \varphi_{j,L} (\vec{\mathbf{A}}_j)_{,M} dV_e - \int_{V_{Oe2}} \varphi_{i,M} (\delta\vec{\mathbf{A}}_i)_{,K} \varphi_{j,K} (\vec{\mathbf{A}}_j)_{,M} dV_e \\
&= \sum_e \sum_i \sum_j \{ \int_{V_{Oe2}} [\frac{1}{\mu} [\varphi_{i,L} \delta_{KM} \varphi_{j,L} dV_e] \delta\vec{\mathbf{A}}_{iK} \vec{\mathbf{A}}_{jM} - [\int_{V_{Oe2}} \frac{1}{\mu} \varphi_{i,M} \varphi_{j,K} dV_e] \delta\vec{\mathbf{A}}_{iK} \vec{\mathbf{A}}_{jM} \}
\end{aligned}
\tag{A.44}
$$

4-Term: $(\boldsymbol{\nabla} \cdot \delta\vec{\mathbf{A}})$ ⇒As $(\nabla \cdot \delta\vec{\mathbf{A}})$ is the divergence of $\delta\vec{\mathbf{A}}$, doing basic development for a 3x3, then

$$\nabla \cdot \delta\vec{\mathbf{A}} = \frac{\partial \delta\vec{\mathbf{A}}_{\mathbf{I}}}{\partial I} + \frac{\partial \delta\vec{\mathbf{A}}_{\mathbf{J}}}{\partial J} + \frac{\partial \delta\vec{\mathbf{A}}_{\mathbf{K}}}{\partial K} = \delta\vec{\mathbf{A}}_{\mathbf{I,I}} + \delta\vec{\mathbf{A}}_{\mathbf{J,J}} + \delta\vec{\mathbf{A}}_{\mathbf{K,K}} \tag{A.45}$$

The concept of tensor: $\nabla \cdot \vec{\mathbf{f}} = \dfrac{\partial \vec{\mathbf{f}}_{\mathbf{k}}}{\partial \vec{\mathbf{f}}_{\mathbf{k}}}$

$$\nabla \cdot \delta\vec{\mathbf{A}} = \frac{\partial \delta\vec{\mathbf{A}}_{\mathbf{K}}}{\partial K} = \delta\vec{\mathbf{A}}_{K,K} = \sum_i \varphi_{i,K} (\delta\vec{\mathbf{A}}_{iK})_{,K} \tag{A.46}$$

5-Term: $(\boldsymbol{\nabla} \cdot \vec{\mathbf{A}})$ ⇒Applying the same steps as previous part of the term 2 of equation

$$\nabla \cdot \vec{\mathbf{A}} = \sum_j \varphi_{j,M} (\vec{\mathbf{A}}_{iM})_{,M} \tag{A.47}$$

Putting previous 1, 2, 3, 4 and 5 terms together:

$$\begin{aligned}&\int_{\Omega_2} \frac{1}{\mu}(\nabla\cdot\delta\vec{\mathbf{A}})\cdot(\nabla\cdot\vec{\mathbf{A}})dV = \sum_e \int_{V_{Oe2}} \sum_i\sum_j \frac{1}{\mu}\varphi_{i,K}(\delta\vec{\mathbf{A}}_{iK})_{,K}\varphi_{j,M}(\vec{\mathbf{A}}_{iM})_{,M}dV_e \\ &= \sum_e\sum_i\sum_j[\int_{V_{Oe2}} \frac{1}{\mu}\varphi_{i,K}\varphi_{j,M}dV_e]\delta\vec{\mathbf{A}}_{iK}\vec{\mathbf{A}}_{iM}\end{aligned} \tag{A.48}$$

6-Term: $\boldsymbol{\delta\vec{\mathbf{A}}\cdot\sigma\frac{\partial\vec{\mathbf{A}}}{\partial t}} \quad \Rightarrow$

$$\begin{aligned}&\delta\vec{\mathbf{A}}\cdot\frac{\partial\vec{\mathbf{A}}}{\partial t} = \delta\vec{\mathbf{A}}_K\cdot(\frac{\partial\vec{\mathbf{A}}}{\partial t})_K \\ &\sum_i\varphi_i(\delta\vec{\mathbf{A}}_i)_K\cdot\sum_j\varphi_j(\frac{\partial\vec{\mathbf{A}}_j}{\partial t})_K = \sum_i\sum_j\varphi_i\varphi_j(\delta\vec{\mathbf{A}}_i)_K\cdot(\frac{\partial\vec{\mathbf{A}}}{\partial t})_K = \\ &\sum_i\sum_j\varphi_i\varphi_j(\delta\vec{\mathbf{A}}_i)_K\cdot(\frac{\partial\vec{\mathbf{A}}_j}{\partial t})_M\delta_{KM}\end{aligned} \tag{A.49}$$

7-Term: $\boldsymbol{\delta\vec{\mathbf{A}}\cdot\nabla(\frac{\partial\vec{\mathbf{v}}}{\partial t})} \quad \Rightarrow$

$$\delta\vec{\mathbf{A}}\cdot\nabla(\frac{\partial\vec{\mathbf{v}}}{\partial t}) = \sum_i\varphi_i(\delta\vec{\mathbf{A}}_i)_K\cdot\sum_j\varphi_{j,K}\cdot(\frac{\partial\vec{\mathbf{v}}}{\partial t}) \tag{A.50}$$

8-Term: $\boldsymbol{(\vec{\mathbf{P}}\nabla\times\delta\vec{\mathbf{A}})\cdot n_2} \quad \Rightarrow$ As before

$$\nabla\times\delta\vec{\mathbf{A}} = \epsilon_{KLM}\delta\vec{\mathbf{A}}_{J,M} = \epsilon_{KLM}(\sum_j\varphi_{j,L}(\delta\vec{\mathbf{A}}_j)_M \tag{A.51}$$

9-Term: $\boldsymbol{\vec{\mathbf{P}}\cdot n_2} \quad \Rightarrow$

$$\begin{aligned}&\vec{\mathbf{P}} = \sum_i\vec{\mathbf{P}}_i \\ &\vec{\mathbf{P}}\cdot n_1 = \sum_i\vec{\mathbf{P}}_i\varphi_i(n_2)_{,K}\end{aligned} \tag{A.52}$$

Putting previous 6, 7, 8 and 9 terms together

$$\begin{aligned}
&\int\limits_{\Gamma_{12}} \vec{\mathbf{P}}(\nabla \times \delta\vec{\mathbf{A}}) \cdot n_2 dS = \\
&= \sum_e \int\limits_{A_{0e12}} \sum_i \sum_j P_i \varphi_i n_{2,K} \epsilon_{KLM} \varphi_{j,L} (\delta\vec{\mathbf{A}}_j)_M dA_e \\
&= \sum_e \sum_i \sum_j [\int\limits_{A_{0e12}} \varphi_i \varphi_{j,L} n_{2,K} \epsilon_{KLM} dA_e] \vec{\mathbf{P}}_i \delta\vec{\mathbf{A}}_{jM}
\end{aligned} \tag{A.53}$$

10-Term: $\delta\vec{\mathbf{A}} \cdot (T_0 \times n_2)$ ⇒ Based on

$$\vec{\mathbf{A}} \times \vec{\mathbf{B}} = \vec{\mathbf{A}}_j \vec{\mathbf{B}}_i \vec{\mathbf{e}}_k \epsilon_{ijk} = \epsilon_{ijk} \vec{\mathbf{A}}_j \vec{\mathbf{B}}_i \tag{A.54}$$

$$\begin{aligned}
&\int\limits_{\Gamma_{12}} \delta\vec{\mathbf{A}} \cdot (\vec{\mathbf{T}}_{\mathbf{0}} \times n_2) dS \\
&= \sum_e \int\limits_{A_{0e12}} \sum_j \varphi_j \delta\vec{\mathbf{A}}_{j,M} \epsilon_{KLM} \vec{\mathbf{T}}_{0L} n_{2,K} dA_e \\
&= \sum_e \sum_j [\int\limits_{A_{0e12}} \epsilon_{KLM} \varphi_j \vec{\mathbf{T}}_{0L} n_{2K} dA_e] \delta\vec{\mathbf{A}}_{jM}
\end{aligned} \tag{A.55}$$

Final Governing equation (3.47) (DHONDT 2016b)

$$
\begin{aligned}
&\sum_e \sum_{i=1}^{N} \sum_{j=1}^{N} \{ \left[\int_{V_{Oe2}} \frac{1}{\mu} \varphi_{i,L} \delta_{KM} \varphi_{j,L} dV_e \right] \delta \vec{\mathbf{A}}_{iK} \vec{\mathbf{A}}_{jM} - \\
&\left[\int_{V_{Oe2}} \frac{1}{\mu} \varphi_{i,M} \varphi_{j,K} dV_e \right] \delta \vec{\mathbf{A}}_{iK} \vec{\mathbf{A}}_{jM} + \\
&\left[\int_{V_{Oe2}} \frac{1}{\mu} \varphi_{i,K} \varphi_{j,M} dV_e \right] \delta \vec{\mathbf{A}}_{iK} \vec{\mathbf{A}}_{iM} + \\
&\left[\int_{V_{Oe2}} \varphi_i \sigma \varphi_j dV_e \delta_{K,M} \right] \delta \vec{\mathbf{A}}_{iK} \frac{D \vec{\mathbf{A}}_{jM}}{Dt} + \\
&\left[\int_{V_{Oe2}} \varphi_i \sigma \varphi_{j,K} dV_e \right] \delta \vec{\mathbf{A}}_{iK} \frac{D v_j}{Dt} + \\
&\left[\int_{A_{0e12}} \varphi_i \varphi_{j,L} n_{2,K} \epsilon_{KLM} dA_e \right] \vec{\mathbf{P}}_i \delta \vec{\mathbf{A}}_{jM} \} = \\
&\sum_e \sum_{j=1}^{N} \left[\int_{A_{0e12}} \epsilon_{KLM} \varphi_j \vec{\mathbf{T}}_{0L} n_{2K} dA_e \right] \delta \vec{\mathbf{A}}_{jM}
\end{aligned}
\tag{A.56}
$$

Governing equation (3.48)

11-Term: $\boldsymbol{\nabla \delta \vec{V} \cdot (\frac{\partial \vec{A}}{\partial t})} \quad \Rightarrow$

$$
\begin{aligned}
&\nabla \delta \vec{\mathbf{V}} \cdot (\frac{\partial \vec{\mathbf{A}}}{\partial t}) = \sum_j \varphi_{j,K} \cdot \delta \vec{\mathbf{V}}_j \cdot \sum_i \varphi_i \cdot (\frac{\partial \vec{\mathbf{A}}_i}{\partial t})_K = \\
&\sum_i \sum_j \varphi_i \varphi_{j,K} \cdot \delta \vec{\mathbf{V}}_j \cdot (\frac{\partial \vec{\mathbf{A}}_i}{\partial t})_K
\end{aligned}
\tag{A.57}
$$

12-Term: $\boldsymbol{\nabla \delta \vec{V} \cdot (\nabla \frac{\partial \vec{V}}{\partial t})} \quad \Rightarrow$

$$
\begin{aligned}
&\nabla \delta \vec{\mathbf{V}} \cdot (\nabla \frac{\partial \vec{\mathbf{V}}}{\partial t}) = \sum_i \varphi_{i,K} \delta \vec{\mathbf{V}}_i \cdot \sum_j \varphi_{j,K} \cdot (\frac{\partial \vec{\mathbf{V}}_j}{\partial t})_K = \\
&\sum_i \sum_j \varphi_{i,K} \varphi_{j,K} \cdot \delta \vec{\mathbf{V}}_i \cdot (\frac{\partial \vec{\mathbf{V}}_j}{\partial t})
\end{aligned}
\tag{A.58}
$$

Final Governing equation (3.48) (DHONDT 2016b)

$$\sum_e \sum_{i=1}^{N} \sum_{j=1}^{N} \{ \left[\int\limits_{V_{Oe2}} \varphi_{i,K} \sigma \varphi_i dV_e \right] \delta v_j \frac{D\vec{\mathbf{A}}_{iK}}{Dt} + \\ \left[\int\limits_{V_{Oe2}} \varphi_{i,K} \sigma \varphi_{j,K} dV_e \right] \delta v_i \frac{Dv_j}{Dt} \} = 0 \tag{A.59}$$

Governing equation (3.46)

13-Term: $(\nabla \delta \vec{\mathbf{P}}) \cdot \nabla \vec{\mathbf{P}}$ ⇒As previous steps

$$\nabla \delta \vec{\mathbf{P}} \cdot \nabla \vec{\mathbf{P}} = \sum_i \varphi_{i,K} \delta \vec{\mathbf{P}}_i \sum_j \varphi_{j,K} \vec{\mathbf{P}}_j \tag{A.60}$$

So,

$$\int\limits_{\Omega_1} \mu \nabla \delta \vec{\mathbf{P}} \cdot \nabla \vec{\mathbf{P}} d\Omega = \\ \sum_e \int\limits_{V_{Oe1}} \mu \sum_i \varphi_{i,K} \delta \vec{\mathbf{P}}_i \sum_j \varphi_{j,K} \vec{\mathbf{P}}_j dV_e = \\ \sum_e \sum_i \sum_j \left[\int\limits_{V_{Oe1}} \varphi_{i,K} \mu \varphi_{j,K} dV_e \right] \delta \vec{\mathbf{P}}_i \vec{\mathbf{P}}_j \tag{A.61}$$

14-Term: $(\delta \vec{\mathbf{P}})(\nabla \times \vec{\mathbf{A}})$ ⇒With

$$(\nabla \times \vec{\mathbf{A}}) = \epsilon_{KLM} \vec{\mathbf{A}}_{J,M} = \epsilon_{KLM} (\sum_j \varphi_{j,L} (\vec{\mathbf{A}}_j)_{,M}) \tag{A.62}$$

$$\delta \vec{\mathbf{P}} \cdot n_2 = \sum_i \delta \vec{\mathbf{P}} \varphi_i (n_2)_K \tag{A.63}$$

$$\begin{aligned}
&\int\limits_{\Gamma_{12}} (\delta\vec{\mathbf{P}})(\nabla \times \vec{\mathbf{A}}) \cdot n_2 dS = \\
&\sum_e \int\limits_{A_{Ve12}} \sum_i \delta\vec{\mathbf{P}}_i \varphi_i (n_2)_K \epsilon_{KLM} (\sum_j \varphi_{j,L} (\vec{\mathbf{A}}_j)_{,M}) dA_e = \\
&\sum_e \sum_i \sum_j \left[\int\limits_{A_{Ve12}} \varphi_i n_{2K} \epsilon_{KLM} \varphi_{j,L} dA_e \right] \delta\vec{\mathbf{P}}_i \vec{\mathbf{A}}_{jM}
\end{aligned} \tag{A.64}$$

15-Term: $(\nabla \delta\vec{\mathbf{P}}) \cdot \vec{\mathbf{T_0}}$ ⇒With

$$\nabla \delta\vec{\mathbf{P}} \cdot \vec{\mathbf{T_0}} = \sum_i \delta\vec{\mathbf{P}}_i \varphi_{i,K} \vec{\mathbf{T}}_{0K} \tag{A.65}$$

Then

$$\begin{aligned}
&\int\limits_{\Omega_1} \mu \nabla \delta\vec{\mathbf{P}} \cdot \vec{\mathbf{T_0}} dV = \\
&\sum_e \int\limits_{V_{0e1}} \mu \sum_i \delta\vec{\mathbf{P}}_i \varphi_{i,K} \vec{\mathbf{T_{0K}}} dV_e = \\
&\sum_e \sum_i \left[\int\limits_{V_{0e1}} \mu \varphi_{i,K} \vec{\mathbf{T_{0K}}} dV_e \right] \delta\vec{\mathbf{P}}_i
\end{aligned} \tag{A.66}$$

Final Governing equation (3.46) (DHONDT 2016b)

$$\begin{aligned}
&\sum_e \sum_{i=1}^{N} \sum_{j=1}^{N} \{ \left[\int\limits_{V_{Oe2}} \varphi_{i,K} \mu \varphi_{j,K} dV_e \right] \delta\vec{\mathbf{P}}_i \vec{\mathbf{P}}_j + \\
&\left[\int\limits_{A_{0e12}} \varphi_i \epsilon_{KLM} \varphi_{j,L} n_{2K} dA_e \right] \delta\vec{\mathbf{P}}_i \vec{\mathbf{A}}_{jM} = \\
&-\sum_e \sum_{i=1}^{N} \left[\int\limits_{V_{Oe2}} \varphi_{i,K} \mu \vec{\mathbf{T}}_{0K} dV_e \right] \delta\vec{\mathbf{P}}_i
\end{aligned} \tag{A.67}$$

A.3.0.2 Second Step: evolve to matrices

(DHONDT 2016b)

$$[K_{AA}]_{e(iK)(jM)} = \int\limits_{(V_{0e})_{\Omega_2}} [\frac{1}{\mu}\varphi_{i,L}\delta_{KM}\varphi_{j,L} - \varphi_{i,M}\varphi_{i,K} + \varphi_{i,K}\varphi_{j,M}]dV_e \tag{A.68}$$

$$[K_{AP}]_{e(jM)(i)} = \int\limits_{(A_{0e})_{\Gamma_{12}}} \varphi_i\varphi_{j,L}n_{1,K}\epsilon_{KLM}dA_e \tag{A.69}$$

$$[K_{PA}]_{e(i)(jM)} = \int\limits_{(A_{0e})_{\Gamma_{12}}} \varphi_i\epsilon_{KLM}\varphi_{j,L}n_{1K}dA_e \tag{A.70}$$

$$[K_{PP}]_{e(i)(j)} = -\int\limits_{(V_{0e})_{\Omega_1}} \varphi_{i,K}\mu\varphi_{j,K}dV_e \tag{A.71}$$

$$[M_{AA}]_{e(iK)(jM)} = \int\limits_{(V_{0e})_{\Omega_2}} \varphi_i\sigma\varphi_j\delta_{K,M}dV_e \tag{A.72}$$

$$[M_{Av}]_{e(iK)(j)} = \int\limits_{(V_{0e})_{\Omega_2}} \varphi_i\sigma\varphi_{j,K}\delta_{K,M}dV_e \tag{A.73}$$

$$[M_{vA}]_{e(j)(iM)} = \int\limits_{(V_{0e})_{\Omega_2}} \varphi_{j,K}\sigma\varphi_i\delta_{KM}dV_e \tag{A.74}$$

$$[M_{vv}]_{e(i)(j)} = \int\limits_{(V_{0e})_{\Omega_2}} \varphi_{i,K}\sigma\varphi_{i,K}\delta_{KM}dV_e \tag{A.75}$$

$$[F_A]_{e(jM)} = -\int\limits_{(A_{0e})_{\Gamma_{12}}} \epsilon_{KLM}\varphi_j\vec{\mathbf{T}}_{0L}n_{1K}dA_e \tag{A.76}$$

$$[F_p]_{e(i)} = -\int\limits_{(V_{0e})_{\Omega_1}} \varphi_{i,K}\mu\vec{\mathbf{T}}_{0K}dV_e \tag{A.77}$$

B.1 CalculiX

CalculiX is one of the most important open source software package for numerical 3D structural calculation using the finite element method (FEM) (DHONDT 2016b), (**Wittig.2016**) .

It has been developed by Dr. Guido Dhondt and Dr. Klaus Wittig in their raw spare time. They work for MTU Aero Engines in Munich (Germany) and MTU granted the publication. First approach was generated for Linux/Unix and after that, the package has been migrated to Windows. Actual available Version is 2.11.

The software package, several samples and an exhaustive documentation could be found at `http://calculix.de/` and at `http://www.dhondt.de/`.

It has two well defined parts that could be used separately:

- **CalculiX CrunchiX Solver (CCX):**
  - **ccx** has been developed by Dr. Guido Dhondt
  - **ccx** needs and input file to work with ($ccx - iDeck\_Input\_File$). The Input Deck file format must has been generated with a similar input format to ABAQUS, and because of that, it is possible to use a commercial pre-processor like Hypermesh or the other part of the **CalculiX** package, the **cgx** package.
  - **ccx** generates four output files with the results founds. In particular, the Results output file (.frd) can be processed by the other part of the **CalculiX** SW, by **cgx**.
- **CalculiX GraphiX (cgx):**
  - **cgx** has been developed by Dr. Klaus Wittig
  - **cgx** has pre- and post-processor capabilities and it is compatible with different formats, for instance, with abaqus, ansys, nastran, etc...
  - On one side, generates and displays the FE, and for the other, handles the **ccx** outputs. (Result .frd file format definition is documented in the **cgx** Users Manual)

B.1.1 CalculiX Code Analysis (Electromagnetism part)

The standard **CalculiX** Solver (version 11) package (**Dhondt.2016a**) is composed by 798 modules:

C (.c)	h (.h)	Fortran (.f)	Perl (.pl)
97	7	693	1
$12,16\%$	$0,88\%$	$86,84\%$	$0,12\%$

Table B.1: Modules type split

The starting point of the application, **main function**, is on ***ccx_2.11.c*** module.

Everything related to the electromagnetism part starts from ***electromagnetisc.c*** module.

B.1.2 CalculiX Execution analysis

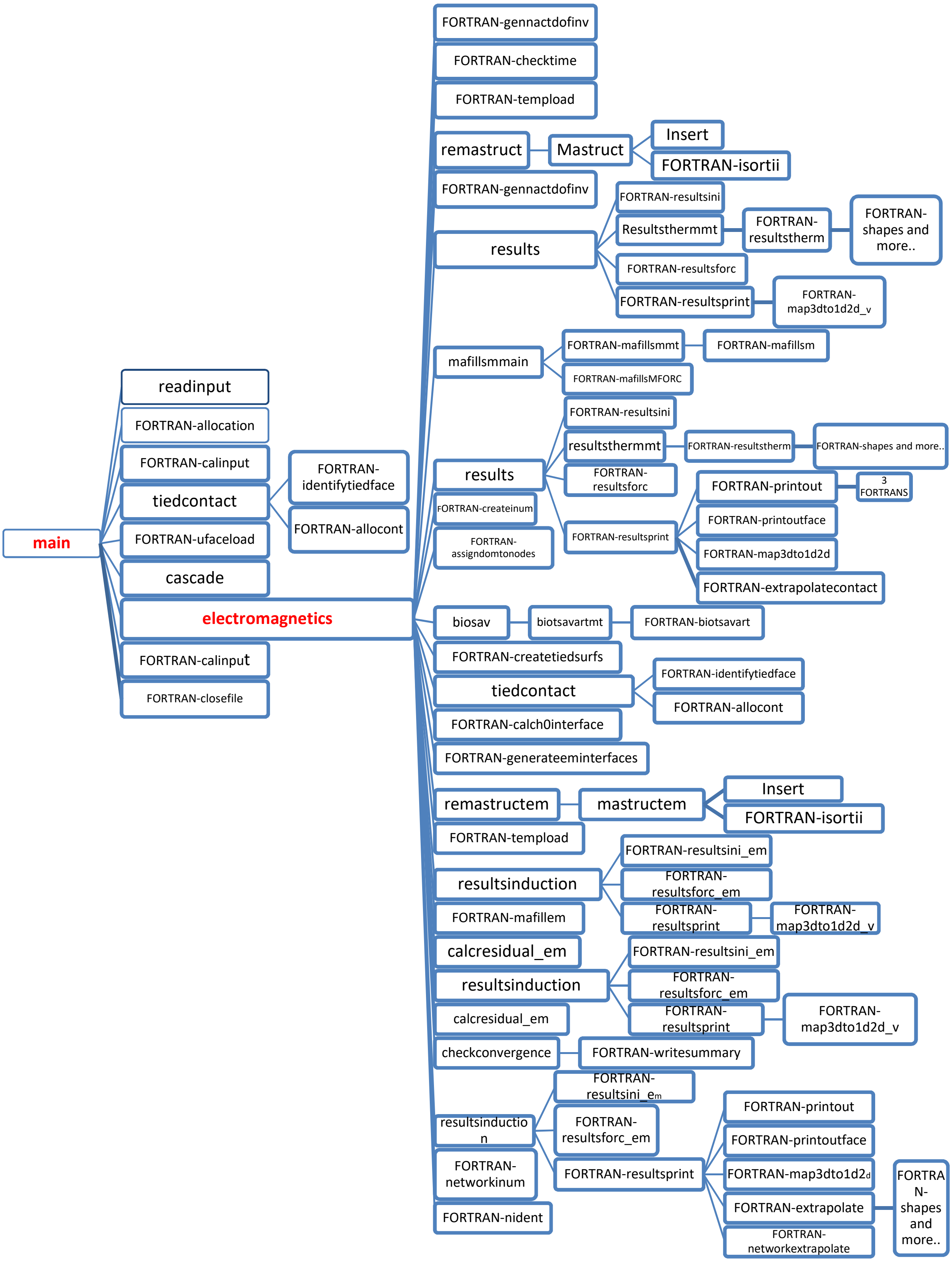
For the Electromagnetism part of the **CalculiX** Solver package (**Dhondt.2016a**), a simple Trace Log functionality has been implemented.

This functionality helps to follow the behavior of **CalculiX** Solver execution at specific points.The execution information, for the electromagnetic part, has been generated using that trace log functionality.

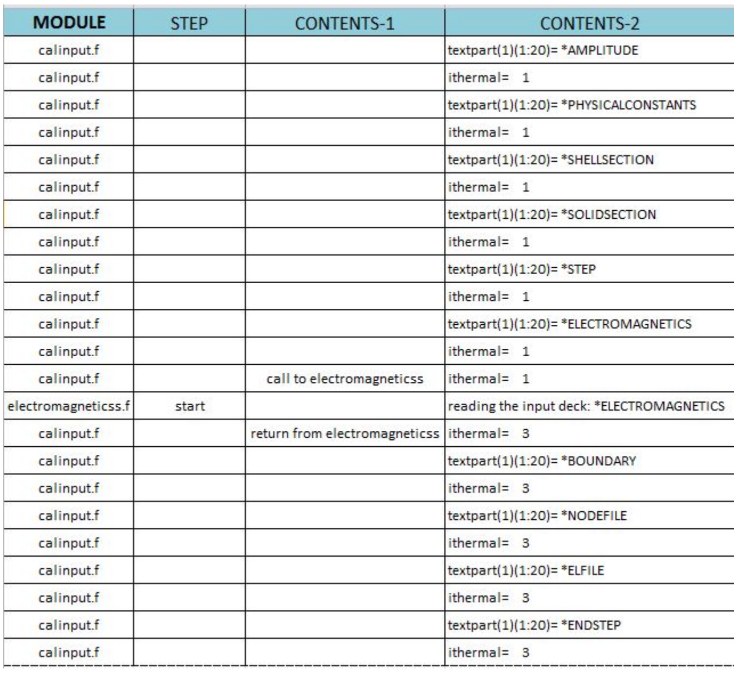
Attached next the output of the trace function log. It is necessary to take into account that tracking times goes up to milliseconds, with 6 digits in C modules and 3 in the Fortran ones, therefore, in the case of Fortran, padding with zeros is used, so, it implies that, in some cases, this can cause a very small offset in the order of the entries of the generated trace file, when applying sort filters at visualizing time. In addition, it must be borne in mind, that the use of multi-threading at runtime, when parallelizing tasks, can also influence it.

Based on that information, except for errors or omissions, different views have been generated, for example

- **Graph of modules execution path** (*itermal*=no temperature involved in the calculation & *nmethod*=static): A graphical view of the hierarchical dependencies at the time of execution, this graph has been generated manually from the table pointed out above


main
readinput
FORTRAN-allocation
FORTRAN-calinput
tiedcontact
FORTRAN-identifytiedface
FORTRAN-allocont
FORTRAN-ufaceload
cascade
electromagnetics
FORTRAN-calinput
FORTRAN-closefile
FORTRAN-gennactdofinv
FORTRAN-checktime
FORTRAN-tempload
remastruct
Mastruct
Insert
FORTRAN-isortii
FORTRAN-gennactdofinv
results
FORTRAN-resultsini
Resultsthermmt
FORTRAN-resultstherm
FORTRAN-shapes and more..
FORTRAN-resultsforc
FORTRAN-resultsprint
FORTRAN-map3dto1d2d_v
mafillsmmain
FORTRAN-mafillsmmt
FORTRAN-mafillsm
FORTRAN-mafillsMFORC
results
FORTRAN-resultsini
resultsthermmt
FORTRAN-resultstherm
FORTRAN-shapes and more..
FORTRAN-resultsforc
FORTRAN-resultsprint
FORTRAN-printout
3 FORTRANS
FORTRAN-printoutface
FORTRAN-map3dto1d2d
FORTRAN-extrapolatecontact
FORTRAN-createinum
FORTRAN-assigndomtonodes
biosav
biotsavartmt
FORTRAN-biotsavart
FORTRAN-createtiedsurfs
tiedcontact
FORTRAN-identifytiedface
FORTRAN-allocont
FORTRAN-calch0interface
FORTRAN-generateeminterfaces
remastructem
mastructem
Insert
FORTRAN-isortii
FORTRAN-tempload
resultsinduction
FORTRAN-resultsini_em
FORTRAN-resultsforc_em
FORTRAN-resultsprint
FORTRAN-map3dto1d2d_v
FORTRAN-mafillem
calcresidual_em
resultsinduction
FORTRAN-resultsini_em
FORTRAN-resultsforc_em
FORTRAN-resultsprint
FORTRAN-map3dto1d2d_v
calcresidual_em
checkconvergence
FORTRAN-writesummary
resultsinduction
FORTRAN-resultsini_em
FORTRAN-resultsforc_em
FORTRAN-resultsprint
FORTRAN-printout
FORTRAN-printoutface
FORTRAN-map3dto1d2d
FORTRAN-extrapolate
FORTRAN-shapes and more..
FORTRAN-networkextrapolate
FORTRAN-networkinum
FORTRAN-nident

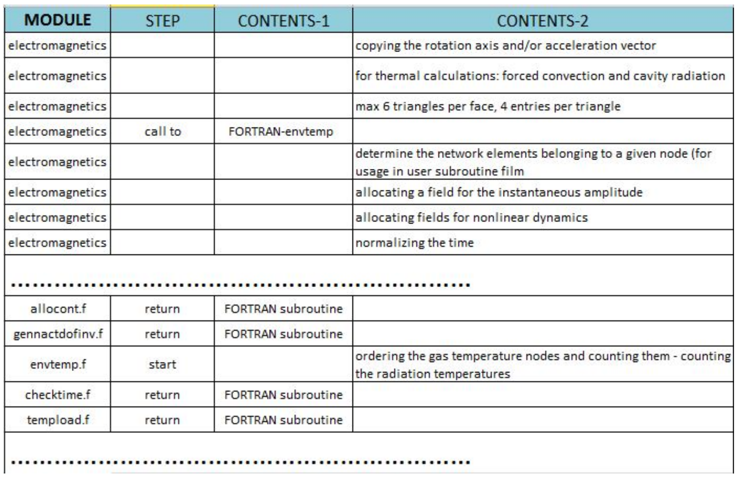
- **Module path execution exercise**(*itermal*=uncoupled thermal-mechanical analysis & *nmethod*=dynamic): After running **ccx** using an input deck like the ones used for T0/H verifications and another one like the ones used for the Work Piece verification, a comparison can be made between trace log output files. In this comparison, it has verified the existence of specific components of **CalculiX** that are only executed for the case of the Work Piece and not for the case of T0 / H. For example, ***calinput.f*** module is managing, among other things, the information provided by input deck file, like temperature, electromagnetic, amplitude, etc...



MODULE	STEP	CONTENTS-1	CONTENTS-2
calinput.f			textpart(1)(1:20)= *AMPLITUDE
calinput.f			ithermal= 1
calinput.f			textpart(1)(1:20)= *PHYSICALCONSTANTS
calinput.f			ithermal= 1
calinput.f			textpart(1)(1:20)= *SHELLSECTION
calinput.f			ithermal= 1
calinput.f			textpart(1)(1:20)= *SOLIDSECTION
calinput.f			ithermal= 1
calinput.f			textpart(1)(1:20)= *STEP
calinput.f			ithermal= 1
calinput.f			textpart(1)(1:20)= *ELECTROMAGNETICS
calinput.f			ithermal= 1
calinput.f		call to electromagneticss	ithermal= 1
electromagneticss.f	start		reading the input deck: *ELECTROMAGNETICS
calinput.f		return from electromagneticss	ithermal= 3
calinput.f			textpart(1)(1:20)= *BOUNDARY
calinput.f			ithermal= 3
calinput.f			textpart(1)(1:20)= *NODEFILE
calinput.f			ithermal= 3
calinput.f			textpart(1)(1:20)= *ELFILE
calinput.f			ithermal= 3
calinput.f			textpart(1)(1:20)= *ENDSTEP
calinput.f			ithermal= 3

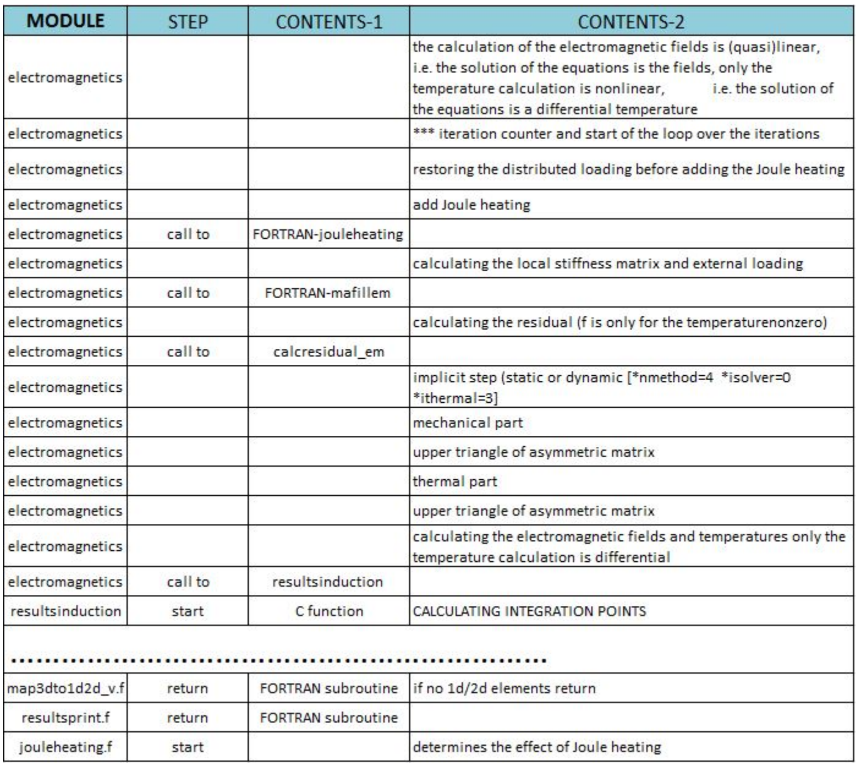
Image B.1: TraceLog related to one part of calinput module

or ***envtemp.f*** is performing thermal task



MODULE	STEP	CONTENTS-1	CONTENTS-2
electromagnetics			copying the rotation axis and/or acceleration vector
electromagnetics			for thermal calculations: forced convection and cavity radiation
electromagnetics			max 6 triangles per face, 4 entries per triangle
electromagnetics	call to	FORTRAN-envtemp	
electromagnetics			determine the network elements belonging to a given node (for usage in user subroutine film
electromagnetics			allocating a field for the instantaneous amplitude
electromagnetics			allocating fields for nonlinear dynamics
electromagnetics			normalizing the time
....			
allocont.f	return	FORTRAN subroutine	
gennactdofinv.f	return	FORTRAN subroutine	
envtemp.f	start		ordering the gas temperature nodes and counting them - counting the radiation temperatures
checktime.f	return	FORTRAN subroutine	
tempload.f	return	FORTRAN subroutine	
....			

Image B.2: TraceLog related to one part of envtemp module or ***jouleheating.f*** module



MODULE	STEP	CONTENTS-1	CONTENTS-2
electromagnetics			the calculation of the electromagnetic fields is (quasi)linear, i.e. the solution of the equations is the fields, only the temperature calculation is nonlinear, i.e. the solution of the equations is a differential temperature
electromagnetics			*** iteration counter and start of the loop over the iterations
electromagnetics			restoring the distributed loading before adding the Joule heating
electromagnetics			add Joule heating
electromagnetics	call to	FORTRAN-jouleheating	
electromagnetics			calculating the local stiffness matrix and external loading
electromagnetics	call to	FORTRAN-mafillem	
electromagnetics			calculating the residual (f is only for the temperaturenonzero)
electromagnetics	call to	calcresidual_em	
electromagnetics			implicit step (static or dynamic [*nmethod=4 *isolver=0 *ithermal=3]
electromagnetics			mechanical part
electromagnetics			upper triangle of asymmetric matrix
electromagnetics			thermal part
electromagnetics			upper triangle of asymmetric matrix
electromagnetics			calculating the electromagnetic fields and temperatures only the temperature calculation is differential
electromagnetics	call to	resultsinduction	
resultsinduction	start	C function	CALCULATING INTEGRATION POINTS
....			
map3dto1d2d_v.f	return	FORTRAN subroutine	if no 1d/2d elements return
resultsprint.f	return	FORTRAN subroutine	
jouleheating.f	start		determines the effect of Joule heating

Image B.3: TraceLog related to one part of jouleheating module

- **Execution representation from Finite Element point of view** (itermal=no temperature involved in the calculation & nmethod=static): A summary table containing the main steps performed within each component at runtime. This information has been generated by tracing the different program comments of each module involved, together with the necessary manual adjustments in order to show a more clear presentation.

Execution representation from Finite Element point of view: CalculiX Solver execution trace log info sent to the standard output

TIME	MODULE	STEP	CONTENTS-1	CONTENTS-2	CONTENTS-3
133142920456	main	start	C function		
133142922469	main	call to	readinput		
133142961755	main	call to	FORTRAN-allocation		
133143120155	main	call to	FORTRAN-calinput	reading the input file	
133143326607	main	call to	tiedcontact		
133143326632	tiedcontact	start	C function		
133143326641	tiedcontact	call to	FORTRAN-identifytiedface		
133143326705	tiedcontact	call to	FORTRAN-allocont		
133143326735	tiedcontact	return	C function		
133143327677	main	call to	FORTRAN-ufaceload		
133143327700	main	call to	cascade		
133143327927	main			nmethod type: nmethod= 8	
133143327939	main	call to	electromagnetics		
133143327967	electromagnetics	start	C function		
133143327976	electromagnetics			avoid that elements with negative ipkon are taken into account in extrapolate.f	
133143327986	electromagnetics			invert nactdof (The active degrees of freedom are stored in a two-dimensional field nactdof.)	
133143328090	electromagnetics	call to	FORTRAN-gennactdofinv		
133143328988	electromagnetics			allocating a field for the stiffness matrix	
133143329017	electromagnetics			allocating force fields	
133143329035	electromagnetics			allocating fields for the actual external loading	
133143329047	electromagnetics			copying the rotation axis and/or acceleration vector	
133143329056	electromagnetics			allocating a field for the instantaneous amplitude	
133143329056	electromagnetics			allocating fields for nonlinear dynamics	
133143329065	electromagnetics			normalizing the time	
133143329065	electromagnetics	call to	FORTRAN-checktime		
133143329105	electromagnetics			calculating an initial flux norm	
133143329114	electromagnetics			calculating the initial quasi-static magnetic intensity due to the coil current	
133143329114	electromagnetics			calculate the current density in the coils: in this section nload, nforc, nbody and nam are set to zero. The electrical potential is supposed to be given (in the form of a temperature), the current is calculated (in the form of heat flux) by thermal analogy	
133143329145	electromagnetics			the coil current is assumed to be applied at once, i.e. asstep loading. The calculation, however, is a quasi-static calculation	
133143329145	electromagnetics	call to	FORTRAN-tempload		
133143329242	electromagnetics			deactivating all elements except the shells	
133143329242	electromagnetics	call to	remastruct		
133143329253	remastruct	start	C function	WHEN THE TOPOLOGY OF THE MPCs HAS CHANGED	
133143329253	remastruct			reconstructs the nonzero locations in the stiffness and mass matrix after a change in MPC's	
133143329253	remastruct			decascading the MPC's	
133143329262	remastruct			determining the matrix structure	
133143329286	remastruct	call to	mastruct		
133143329286	remastruct			reallocating fields the size of which depends on neq[1] or *nzs	
133143329306	mastruct	start	C function	DETERMINE THE MATRIX STRUCTURE FOR STRUCTURES NOT EXHIBITING CYCLIC SIMMETRY	
133143329306	mastruct			determines the structure of the thermo-mechanical matrices, (i.e. the location of the nonzeros	
133143329324	mastruct			determining nactdof (only at start of step or if MPCs changed (The active degrees of freedom are stored in a two-dimensional field nactdof)	
133143329332	mastruct	for loop		initialisation of nactdof	
133143329332	mastruct	inside loop		determining the mechanical active degrees of freedom due to elements	
133143330113	mastruct	for loop		determining the thermal active degrees of freedom due to elements	

133143330478	mastruct	for/do loop		determining the active degrees of freedom due to mpc's	
133143330524	mastruct			numbering the active degrees of freedom	
133143330805	mastruct	for loop		determining the position of each nonzero matrix element in the SUBdiagonal matrix (excluding diagonal): x-elements on the left of the vertical line	
133143330805	mastruct	inside loop		mast1(ipointer(i)) = first nonzero row in column i next(ipointer(i)) points to further nonzero elements in column i	
133143331244	mastruct	inside loop		mechanical entries	
133143331253	mastruct	for loop		thermal entries	
133143331262	mastruct	inside loop		check whether one of the DOF belongs to a SPC or MPC	
133143337111	mastruct	inside loop		MPC id1 / MPC id1	
133143337141	mastruct	inside loop		MPC id1 /MPC id2	
133143337150	mastruct	inside loop		idof1: genuine DOF - idof2: nominal DOF of the SPC/MPC	
133143337158	mastruct	inside loop		regular DOF / MPC	
133143344029	mastruct			determination of the following fields. irow: row numbers, column per column icol(i)=# SUBdiagonal nonzero's in column i jq(i)= location in field irow of the first SUBdiagonal nonzero in column i	
133143344029	mastruct			subdiagonal elements of the regular stiffness/mass matrices	
133143345195	mastruct			sorting the row numbers within each column	
133143349948	mastruct			removing duplicate entries	
133143350465	mastruct			determining irow, jq and icol for the boundary stiffness matrix (only for frequencyand modal dynamic calculations)	
133143350559	mastruct	return	C function		calls done to insert FORTRAN-isortii
133143350867	remastruct			for static calculations fini has to be set to f at the start of the calculation, in dynamic calculations this is not needed, since the initial accelerations has already been calculated	
133143350935	remastruct	return	C function		
133143351063	electromagnetics			invert nactdof (The active degrees of freedom are stored in a two-dimensional field nactdof.)	
133143351063	electromagnetics	call to	FORTRAN-gennactdofinv		
133143352086	electromagnetics	call to	results		
133143352114	results	start	C function	CALCULATING THE INTERNAL FORCES	
133143352174	results	call to	FORTRAN-resultsini		
133143353676	results	call to	resultsthermmt	pthread_create -> resultsthermmt and wait	
133143353775	resultsthermmt	start	C function		
133143354931	resultsthermmt	call to	FORTRAN-resultstherm	inside: void *resultsthermmt(ITG *i)	
133143370608	resultsthermmt	return	C function		
133143372077	results	call to	FORTRAN-resultsforc		
133143372857	results	call to	FORTRAN-resultsprint		
133143376615	results	return	C function		
133143376836	electromagnetics	call to	mafillsmmain		
133143376849	mafillsmmain	start	C function	CALCULATING THE STIFFNESS AND/OR MASS MATRIX (symmetric part)	
133143377037	mafillsmmain	call to	FORTRAN-mafillsmmt	pthread_create -> mafillsmmt	
133143377118	mafillsmmt	call to	FORTRAN-mafillsm		
133143400818	mafillsmmt	return	C function		
133143402209	mafillsmmain	call to	FORTRAN-mafillsmforc		
133143402348	mafillsmmain	return	C function		
133143650500	electromagnetics	call to	results		
133143650541	results	start	C function	CALCULATING THE INTERNAL FORCES	
133143650589	results	call to	FORTRAN-resultsini		
133143651977	results	call to	resultsthermmt	pthread_create -> resultsthermmt and wait	
133143652062	resultsthermmt	start	C function		
133143652079	resultsthermmt	call to	FORTRAN-resultstherm	inside: void *resultsthermmt(ITG *i)	
133143658674	resultsthermmt	return	C function		
133143659924	results	call to	FORTRAN-resultsforc		
133143659993	results	call to	FORTRAN-resultsprint		
133143686357	results	return	C function		

133143686392	electromagnetics			reactivating the non-shell elements (for mesh output purposes) deactivating the initial temperature for the non-shell nodes	
133143687821	electromagnetics			deactivating the output of temperatures	
133143687854	electromagnetics			createinum is called in order to store the nodes and elements of the complete structure, not only of the coil	
133143687854	electromagnetics	call to	FORTRAN-createinum		
133143849525	electromagnetics			reactivating the temperature output, if previously deactivated	
133143849586	electromagnetics			calculating the magnetic intensity caused by the current	
133143849586	electromagnetics	call to	FORTRAN-assigndomtonodes		
133143849813	electromagnetics	call to	biosav		
133143849838	biosav	start	C function	CALCULATES THE MAGNETIC INTENSITY	
133143849847	biosav			calculates the magnetic intensity due to currents in the phi- domain of an electromagnetic calculation- Variables for multithreading procedure	
133143849855	biosav			explicit user declaration prevails	
133143849864	biosav			automatic detection of available number of processors	
133143849909	biosav			local declaration prevails, if strictly positive	
133143849919	biosav			else global declaration, if any, applies	
133143849927	biosav			determining the nodal bounds in each thread	
133143849936	biosav			n1 is the number of nodes in the phi(magnetostatic)-domain in an electromagnetic calculation	
133143850011	biosav			dividing the range from 1 to the number of phi-nodes	
133143850019	biosav			translating the bounds of the ranges to real node numbers	
133143850032	biosav			create threads and wait	
133143850041	biosav	call to	biotsavartmt	pthread_create -> biotsavartmt	
133143850095	biotsavartmt	start	C function	subroutine for multithreading of biotsavart	
133143850112	biotsavartmt	call to	FORTRAN-biotsavart		
133143326000	identifytiedface.f	return	FORTRAN subroutine		
133143326000	allocont.f	return	FORTRAN subroutine		
133143328000	gennactdofinv.f	return	FORTRAN subroutine		
133143329000	checktime.f	return	FORTRAN subroutine		
133143329000	tempload.f	return	FORTRAN subroutine		
133143351000	gennactdofinv.f	return	FORTRAN subroutine		
133143353000	resultsini.f	return	FORTRAN subroutine		
133143370000	resultstherm.f	return	FORTRAN subroutine	include "gauss.f"	calls to FORTRAN-shapexx FORTRAN-lintemp_th FORTRAN-networkforc FORTRAN-springforc_n2f_th FORTRAN-materialdata_th FORTRAN-thermmodel FORTRAN-beamintscheme FORTRAN-advecforc
133143372000	resultsforc.f	return	FORTRAN subroutine		
133143372000	resultsprint.f	call to	FORTRAN-map3dto1d2d_v		
133143372000	map3dto1d2d_v.f	start		INTERPOLATES BASIC DEGREE OF FREEDOM NODAL VALUES (displacements,temperatures) TO 1d/2d NODAL LOCATION	
133143372000	map3dto1d2d_v.f			removing any results in 1d/2d nodes	
133143372000	map3dto1d2d_v.f			inactivating the 3d expansion nodes of 1d/2d elements	
133143374000	map3dto1d2d_v.f	do loop		interpolation of 3d results on 1d/2d nodes	
133143374000	map3dto1d2d_v.f	inside loop		check whether linear or quadratic element	
133143374000	map3dto1d2d_v.f	inside loop		making the mean across the thickness	
133143374000	map3dto1d2d_v.f	inside loop		middle nodes: weights 1/2,1/2	
133143376000	map3dto1d2d_v.f	inside loop		taking the mean of nodal contributions coming from different elements having the node in common	
133143376000	map3dto1d2d_v.f	return	FORTRAN subroutine		
133143376000	resultsprint.f	return	FORTRAN subroutine		

133143402000	mafillsmforc.f	return	FORTRAN subroutine		
133143651000	resultsini.f	return	FORTRAN subroutine		
133143658000	resultstherm.f	return	FORTRAN subroutine	include "gauss.f"	calls to FORTRAN-shapexx FORTRAN-lintemp_th FORTRAN-networkforc FORTRAN-springforc_n2f_th FORTRAN-materialdata_th FORTRAN-thermmodel FORTRAN-beamintscheme FORTRAN-advecforc
133143659000	resultsforc.f	return	FORTRAN subroutine		
133143660000	resultsprint.f	call to	FORTRAN-printout		
133143660000	printout.f	return	FORTRAN subroutine		calls to FORTRAN-map3dto1d2d FORTRAN-printoutnode FORTRAN-printoutelem
133143660000	resultsprint.f	call to	FORTRAN-printoutface		
133143660000	resultsprint.f	call to	FORTRAN-map3dto1d2d		
133143660000	map3dto1d2d.f	start		INTERPOLATES 3d FIELD NODAL VALUES TO 1d/2d NODAL LOCATIONS. The number of internal state variables is limited to 999 (cfr. array field)	
133143660000	map3dto1d2d.f	do loop		removing any results in 1d/2d nodes	
133143660000	map3dto1d2d.f	inside loop		inactivating the 3d expansion nodes of 1d/2d elements in case forces are mapped this field is used to ensure that the forces in the 3d-nodes are mapped only once onto the 2d-nodes	
133143662000	map3dto1d2d.f	inside loop		if no 1d/2d elements return	
133143662000	map3dto1d2d.f	inside loop		interpolation of 3d results on 1d/2d nodes	
133143662000	map3dto1d2d.f	inside loop		check whether linear or quadratic element	
133143662000	map3dto1d2d.f	inside loop		taking the mean across the thickness	
133143662000	map3dto1d2d.f	inside loop		middle nodes: weights 1/2,1/2	
133143677000	map3dto1d2d.f			taking the mean of nodal contributions coming from different elements having the node in commonrestoring inum for the 3d-nodes to zero.	
133143677000	map3dto1d2d.f	inside loop		inactivating the 3d expansion nodes of 1d/2d elements in case forces are mapped this field is used to ensure that the forces in the 3d-nodes are mapped only once onto the 2d-nodes	
133143679000	map3dto1d2d.f	inside loop		beam section forces in the middle nodes	
133143681000	map3dto1d2d.f	return	FORTRAN subroutine	include ""gauss.f"	
133143681000	resultsprint.f	call to	FORTRAN-extrapolatecontact	7th call	
133143681000	extrapolate.f	start		EXTRAPOLATES FIELD VALUES AT INTEGRATION POINTS TO THE NODES the C3D20RB element has 50 integration points, however, the first 8 integration points coincide with those of a C3D20R element. In this routine the C3D20RBR and C3D20RBC elements are treated as an ordinary C3D20R element	
133143681000	extrapolate.f	inside loop		storage in global coordinates. Determining the field values in the vertex nodes, for C3D20R and C3D8: trilinear extrapolation (= use of theC3D8 linear interpolation functions) for C3D8R: constant field value in each element for C3D8R: constant field value in each element	
133143681000	extrapolate.f	inside loop		determining the field values in the midside nodes .	
133143685000	extrapolate.f			taking the mean of nodal contributions coming from different elements having the node in common.	
133143685000	extrapolate.f			for 1d and 2d elements only: finding the solution in the original nodes.	

133143685000	extrapolate.f	return	FORTRAN subroutine	include "gauss.f"	calls to FORTRAN-shapexx FORTRAN-beamintscheme FORTRAN-transformatrix FORTRAN-map3dto1d2d
133143686000	resultsprint.f	return	FORTRAN subroutine		
133143687000	createinum.f	start		determines inum in case no extrapolation is requestedin the input deck (e.g. only nodal variables are requested)	
133143687000	createinum.f	inside loop		counting the number of elements a node belongs to	
133143688000	createinum.f	return	FORTRAN subroutine		
133143849000	assigndomtonodes.f	start		ASSINGS THE DOMAIN A NODE BELONGS TO, TO THIS NODE (for electromagnetic calculations, only nodes not belonging to shells)	
133143849000	assigndomtonodes.f	do loop		loads nope variable, (number of nodes in spring element [sum of master and slave nodes])	
133143849000	assigndomtonodes.f	return	FORTRAN subroutine		
133143850000	biotsavart.f	start		CALCULATS DE MAGNETIC INTENSITY due to currents in the phi-domain of an electromagnetic calculation	
133143850000	biotsavart.f	do loop		currents are supposed to be modeled by shell elements	
133143850000	biotsavart.f	inside loop		applying gauss3d1 or..d2 or..d3 or..d8 or..3d values	
133143850000	biotsavart.f	inside loop		shape functions	
133148454176	biotsavartmt	return	C function		
133148454242	biosav	return	C function		
133148454446	electromagnetics			deactivating the shell elements	
133148454511	electromagnetics			*** creating connecting MPC's between the domains	
133148454511	electromagnetics			creating contact ties between the domains	
133148454587	electromagnetics	call to	FORTRAN-createtiedsurfs		
133148456341	electromagnetics			tied contact constraints: generate appropriate MPC's	
133148456341	electromagnetics	call to	tiedcontact		
133148456365	tiedcontact	start	C function		
133148456374	tiedcontact	call to	FORTRAN-identifytiedface		
133148456423	tiedcontact	call to	FORTRAN-allocont		
133148456452	tiedcontact	return	C function		
133148456461	electromagnetics			mapping h0 from the phi domain onto the border of the A and A-V domains	
133148456461	electromagnetics	call to	FORTRAN-calch0interface		
133148456521	electromagnetics			*** creating the A.n MPC	
133148456521	electromagnetics			identifying the interfaces between the A and A-V domains and the phi-domain	
133148456521	electromagnetics	call to	FORTRAN-generateeminterfaces		
133148456623	electromagnetics			determining the new matrix structure	
133148456623	electromagnetics	call to	remastructem		
133148456635	remastructem	start	C function		
133148456875	remastructem	call to	mastructem		
133148456887	mastructem	start	C function		
133148488847	mastructem	return	C function		calls done to insert FORTRAN-isortii
133148488994	remastructem	return	C function		
133148489005	electromagnetics			*** starting the loop over the increments	
133148489005	electromagnetics			saving the distributed loads (volume heating will be added because of Joule heating)	
133148489015	electromagnetics			previous increment converged: update the initial values vold is copied into vini	
133148490632	electromagnetics			check for max. # of increments	
133148490673	electromagnetics			determining the actual loads at the end of the new increment	
133148490673	electromagnetics			restoring the distributed loading before adding the Joule heating	
133148490683	electromagnetics			determining the actual loading	
133148491997	electromagnetics	call to	FORTRAN-temploadd		
133148492169	electromagnetics			prediction of the next solution (only for temperature)	
133148493607	electromagnetics			first iteration in first increment: heat tangent	

133148493607	electromagnetics	call to	resultsinduction		
133148493638	resultsinduction	start	C function	CALCULATING INTEGRATION POINTS	
133148493646	resultsinduction			calculating integration point values (strains, stresses, heat fluxes, material tangent matrices and nodal forces). Storing the nodal and integration point results in the .dat file iout=-2: v is assumed to be known and is used to calculate strains, stresses..., no result output corresponds to iout=-1with in addition the calculation of the internal energy density iout=-1: v is assumed to be known and is used to calculate strains, stresses..., no result output - is used to take changes in SPC's and MPC's at the start of a new increment or iteration into account iout=0: v is calculated from the system solution and strains, stresses.. are calculated, no resultoutput iout=1: v is calculated from the system solution and strains, stresses.. are calculated, requested results output iout=2: v is assumed to be known and is used to calculate strains, stresses..., requested results output	
133148493702	resultsinduction			nodewise storage of the primary variables determination which derived variables have to be calculated	
133148493711	resultsinduction	call to	FORTRAN-resultsini_em		
133148495132	resultsinduction			electromagnetic calculation is linear: should not be taken into account in the convergence check (only thermal part is taken into account)	
133148495149	resultsinduction			calculating the stresses and material tangent at the integration points, calculating the internal forces	
133148495161	resultsinduction			calculating the magnetic field	
133148495229	resultsemmt	start	C function		
133148504665	resultsemmt	return	C function		
133148504723	resultsinduction			calculating the thermal flux and material tangent at the integration points, calculating the internal point flux	
133148504740	resultsinduction			calculating the thermal internal forces	
133148504749	resultsinduction	call to	FORTRAN-resultsforc_em		
133148504820	resultsinduction			storing results in the .dat file extrapolation of integration point values to the nodes interpolation of 3d results for 1d/2d elements	
133148504829	resultsinduction	call to	FORTRAN-resultsprint		
133148504964	resultsinduction	return	C function		
133148504975	electromagnetics			the calculation of the electromagnetic fields is (quasi)linear, i.e. the solution of the equations is the fields, only the temperature calculation is nonlinear, i.e. the solution of the equations is a differential temperature	
133148506357	electromagnetics			*** iteration counter and start of the loop over the iterations	
133148506357	electromagnetics			estoring the distributed loading before adding the Joule heating	
133148506449	electromagnetics			calculating the local stiffness matrix and external loading	
133148506449	electromagnetics	call to	FORTRAN-mafillem		
133148535601	electromagnetics			calculating the residual (f is only for the temperaturenonzero)	
133148535601	electromagnetics	call to	calcresidual_em		
133148535641	electromagnetics			mplicit step (static or dynamic	
133148595033	electromagnetics			ucalculating the electromagnetic fields and temperatures only the temperature calculation is differential	
133148596574	electromagnetics	call to	resultsinduction		
133148596609	resultsinduction	start	C function	CALCULATING INTEGRATION POINTS	

133148596618	resultsinduction			calculating integration point values (strains, stresses, heat fluxes, material tangent matrices and nodal forces). Storing the nodal and integration point results in the .dat file iout=-2: v is assumed to be known and is used to calculate strains, stresses..., no result output corresponds to iout=-1with in addition the calculation of the internal energy density iout=-1: v is assumed to be known and is used to calculate strains, stresses..., no result output - is used to take changes in SPC's and MPC's at the start of a new increment or iteration into account iout=0: v is calculated from the system solution and strains, stresses.. are calculated, no resultoutput iout=1: v is calculated from the system solution and strains, stresses.. are calculated, requested results output iout=2: v is assumed to be known and is used to calculate strains, stresses..., requested results output	
133148596672	resultsinduction			nodewise storage of the primary variables determination which derived variables have to be calculated	
133148596681	resultsinduction	call to	FORTRAN-resultsini_em		
133148598377	resultsinduction			electromagnetic calculation is linear: should not be taken into account in the convergence check (only thermal part is taken into account)	
133148598396	resultsinduction			calculating the stresses and material tangent at the integration points, calculating the internal forces	
133148598405	resultsinduction			calculating the magnetic field	
133148598465	resultsemmt	start	C function		
133148607587	resultsemmt	return	C function		
133148607662	resultsinduction			calculating the thermal flux and material tangent at the integration points, calculating the internal point flux	
133148607680	resultsinduction			calculating the thermal internal forces	
133148607690	resultsinduction	call to	FORTRAN-resultsforc_em		
133148607766	resultsinduction			storing results in the .dat file extrapolation of integration point values to the nodes interpolation of 3d results for 1d/2d elements	
133148607775	resultsinduction	call to	FORTRAN-resultsprint		
133148607944	resultsinduction	return	C function		
133148608999	electromagnetics			calculating the residual	
133148608999	electromagnetics	call to	calcresidual_em		
133148609050	electromagnetics			calculating the maximum residual (only thermal part)	
133148609059	electromagnetics			calculating the maximum residual (only thermal part)	
133148609066	electromagnetics			athermal electromagnetic calculations are linear: set iit=2 to force convergence	
133148609074	electromagnetics	call to	checkconvergence		
133148609085	checkconvergence	start	C function	THE CONVERGENCE CRITERIA IS RATHER SIMILAR TO THE ONE USED IN ABAQUS	
133148609094	checkconvergence	call to	FORTRAN-writesummary		
133148611555	checkconvergence	return	C function		
133148611585	electromagnetics			*** end of the iteration loop icutb=0 means that the iterations in the increment converged, icutb!=0 indicates that the increment has to be reiterated with another increment size (dtheta)	
133148613105	electromagnetics			alculating the displacements and the stresses and storing the results in frd format	
133148613105	electromagnetics	call to	resultsinduction		
133148613139	resultsinduction	start	C function	CALCULATING INTEGRATION POINTS	

133148613147	resultsinduction			calculating integration point values (strains, stresses, heat fluxes, material tangent matrices and nodal forces). Storing the nodal and integration point results in the .dat file iout=-2: v is assumed to be known and is used to calculate strains, stresses..., no result output corresponds to iout=-1with in addition the calculation of the internal energy density iout=-1: v is assumed to be known and is used to calculate strains, stresses..., no result output - is used to take changes in SPC's and MPC's at the start of a new increment or iteration into account iout=0: v is calculated from the system solution and strains, stresses.. are calculated, no resultoutput iout=1: v is calculated from the system solution and strains, stresses.. are calculated, requested results output iout=2: v is assumed to be known and is used to calculate strains, stresses..., requested results output	
133148613203	resultsinduction			nodewise storage of the primary variables determination which derived variables have to be calculated	
133148613212	resultsinduction	call to	FORTRAN-resultsini_em		
133148613289	resultsinduction			electromagnetic calculation is linear: should not be taken into account in the convergence check (only thermal part is taken into account)	
133148613299	resultsinduction			calculating the stresses and material tangent at the integration points, calculating the internal forces	
133148613307	resultsinduction			calculating the magnetic field	
133148613378	resultsemmt	start	C function		
133148622570	resultsemmt	return	C function		
133148622642	resultsinduction			calculating the thermal flux and material tangent at the integration points, calculating the internal point flux	
133148622660	resultsinduction			calculating the thermal internal forces	
133148622669	resultsinduction	call to	FORTRAN-resultsforc_em		
133148622681	resultsinduction			storing results in the .dat file extrapolation of integration point values to the nodes interpolation of 3d results for 1d/2d elements	
133148622688	resultsinduction	call to	FORTRAN-resultsprint		
133148625716	resultsinduction	return	C function		
133148626557	electromagnetics	call to	FORTRAN-networkinum		
133148632550	electromagnetics			restoring the distributed loading	
133148632567	electromagnetics			abs(*nmethod)==1 setting the velocity to zero at the end of a quasistatic or stationary step	
133148632841	electromagnetics	call to	FORTRAN-nident	*nboun= 200 times	
133148632841	electromagnetics			thermal boundary conditions are updated only if the step was thermal or thermomechanical	
133148632841	electromagnetics			mechanical boundary conditions are updated only if the step was not thermal or the node is a network node	
133148633318	electromagnetics			nmethod change nmethod==1>8 nmethod==4>9 nmethod==2>10	
133148633491	main	step		resetting the amplitude to none except for time=total time amplitudes	
133148665821	main	call to	FORTRAN-calinput	reading the input file	
133143850000	biotsavart.f	inside loop		coordinates of the gauss point, where pgauss(k)=pgauss(k)+shp(4,l)*xl(k,l)	
133143850000	biotsavart.f	inside loop		distance from node to gauss point, where r(k)=con(k)-pgauss(k)	

133143850000	biotsavart.f	inside loop		step1: calculating h0, where h0(1,j)=h0(1,j)+c2*(qfx(2,kk,i)*r(3)-qfx(3,kk,i)*r(2)) h0(2,j)=h0(2,j)+c2*(qfx(3,kk,i)*r(1)-qfx(1,kk,i)*r(3)) h0(3,j)=h0(3,j)+c2*(qfx(1,kk,i)*r(2)-qfx(2,kk,i)*r(1)) with c2=weight*xsj/((r(1)*r(1)+r(2)*r(2)+r(3)*r(3))**(1.5d0))	
133143853000	biotsavart.f	inside loop		step2: calculating h0, where h0(k,j)=h0(k,j)*c1 c1=1.d0/(16.d0*datan(1.d0))	
133148454000	biotsavart.f	return	FORTRAN subroutine	include "gauss.f"	
133148456000	createtiedsurfs.f	return	FORTRAN subroutine		
133148456000	identifytiedface.f	return	FORTRAN subroutine		
133148456000	allocont.f	return	FORTRAN subroutine		
133148456000	calch0interface.f	start		CALCULATIN h0 ON THE INTERFACE BETWEEN A OR A-V DOMAINS and the phi-domain.At the start of the routine h0 is available inthe complete phi-domain. Using the MPCs developed in tiedcontact to tie the phi-values at the border of the A or A-Vdomains to those in the phi-domain, the h0 values are calculated in a similar way	
133148456000	calch0interface.f	do loop		looking for MPCs tying phi between the A or A-V domains and the phi-domain	
133148456000	calch0interface.f	return	FORTRAN subroutine		
133148456000	generateeminterfaces.f	return	FORTRAN subroutine		
133148495000	resultsini_em.f	return	FORTRAN subroutine		
133148504000	resultsprint.f	call to	FORTRAN-map3dto1d2d_v		
133148504000	map3dto1d2d_v.f	start		INTERPOLATES BASIC DEGREE OF FREEDOM NODAL VALUES (displacements,temperatures) TO 1d/2d NODAL LOCATION	
133148504000	map3dto1d2d_v.f			removing any results in 1d/2d nodes	
133148504000	map3dto1d2d_v.f	return	FORTRAN subroutine	if no 1d/2d elements return	
133148504000	resultsprint.f	return	FORTRAN subroutine		
133148598000	resultsini_em.f	return	FORTRAN subroutine		
133148607000	resultsprint.f	call to	FORTRAN-map3dto1d2d_v		
133148607000	map3dto1d2d_v.f	start		INTERPOLATES BASIC DEGREE OF FREEDOM NODAL VALUES (displacements,temperatures) TO 1d/2d NODAL LOCATION	
133148607000	map3dto1d2d_v.f			removing any results in 1d/2d nodes	
133148607000	map3dto1d2d_v.f	return	FORTRAN subroutine	if no 1d/2d elements return	
133148607000	resultsprint.f	return	FORTRAN subroutine		
133148609000	writesummary.f	return	FORTRAN subroutine		
133148613000	resultsini_em.f	return	FORTRAN subroutine		
133148622000	resultsprint.f	call to	FORTRAN-printout		
133148622000	printout.f	return	FORTRAN subroutine		calls to FORTRAN-map3dto1d2d FORTRAN-printoutnode FORTRAN-printoutelem
133148622000	resultsprint.f	call to	FORTRAN-printoutface		
133148622000	resultsprint.f	call to	FORTRAN-map3dto1d2d		
133148622000	map3dto1d2d.f	start		INTERPOLATES 3d FIELD NODAL VALUES TO 1d/2d NODAL LOCATIONS. The number of internal state variables is limited to 999 (cfr. array field)	
133148622000	map3dto1d2d.f	do loop		removing any results in 1d/2d nodes	
133148622000	map3dto1d2d.f	inside loop		if no 1d/2d elements return	
133148622000	resultsprint.f	call to	FORTRAN-extrapolate		
133148622000	extrapolate.f	start		EXTRAPOLATES FIELD VALUES AT INTEGRATION POINTS TO THE NODES the C3D20RB element has 50 integration points, however, the first 8 integration points coincide with those of a C3D20R element. In this routine the C3D20RBR and C3D20RBC elements are treated as an ordinary C3D20R element	

133148624000	extrapolate.f	inside loop		storage in global coordinates. Determining the field values in the vertex nodes, for C3D20R and C3D8: trilinear extrapolation (= use of theC3D8 linear interpolation functions) for C3D8R: constant field value in each element for C3D8R: constant field value in each element	
133148624000	extrapolate.f	inside loop		determining the field values in the midside nodes .	
133148625000	extrapolate.f			taking the mean of nodal contributions coming from different elements having the node in common.	
133148625000	extrapolate.f			for 1d and 2d elements only: finding the solution in the original nodes.	
133148625000	extrapolate.f	return	FORTRAN subroutine	include "gauss.f"	calls to FORTRAN-shapexx FORTRAN-beamintscheme FORTRAN-transformatrix FORTRAN-map3dto1d2d
133148625000	resultsprint.f	call to	FORTRAN-networkextrapolate		
133148625000	resultsprint.f	return	FORTRAN subroutine		
133148626000	networkinum.f	return	FORTRAN subroutine		
133148665909	main	call to	FORTRAN-closefile		
133148668012	main	end	C function		

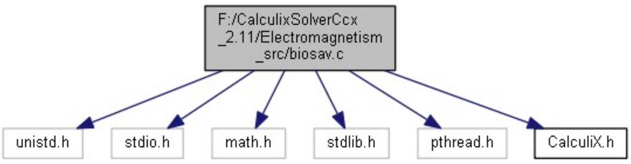
B.1.3 CalculiX Static analysis

Doxygen (DOXYGEN 2016) has been used to generate a Source Cross-Reference documentation of **CalculiX** Solver package. Doxygen is an open source tool that uses that handles C and Fortran code and uses specific functionality from the Graphviz open source package (GRAPHVIZ 2016) to generate advanced graphs and diagrams. Just for few module, cross-references and general information are included here:

biosav.c

```
#include <unistd.h>
#include <stdio.h>
#include <math.h>
#include <stdlib.h>
#include <pthread.h>
#include "CalculiX.h"
```

Include dependency graph for biosav.c:



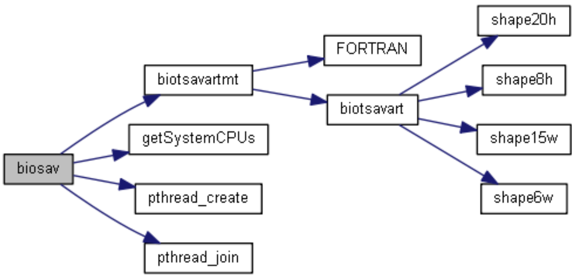
Functions

- void **biosav** (**ITG** *ipkon, **ITG** *kon, char *lakon, **ITG** *ne, double *co, double *qfx, double *h0, **ITG** *mi, **ITG** *inomat, **ITG** *nk)
- void * **biotsavartmt** (**ITG** *i)

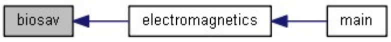
References biotsavartmt(), co1, getSystemCPUs(), h01, ipkon1, ITG, ITGFORMAT, kon1, lakon1, mi1, ne1, nkapar, nkepar, NNEW, num_cpus, pthread_create(), pthread_join(), qfx1, and SFREE.

Referenced by electromagnetics().

Here is the call graph for this function:



Here is the caller graph for this function:

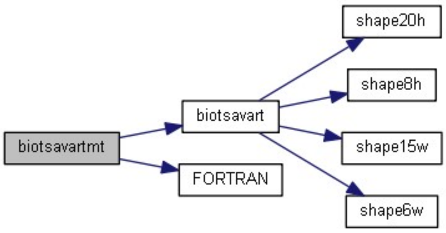


void* biotsavartmt (ITG * *i*)

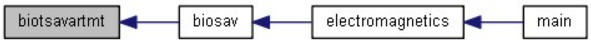
References biotsavart(), co1, FORTRAN(), h01, ipkon1, ITG, kon1, lakon1, mi1, ne1, nkapar, nkepar, and qfx1.

Referenced by biosav().

Here is the call graph for this function:



Here is the caller graph for this function:



biotsavart.f

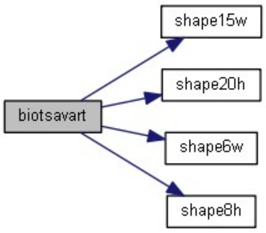
Functions/Subroutines

- subroutine **biotsavart** (ipkon, kon, lakon, ne, co, qfx, h0, mi, nka, nkb)

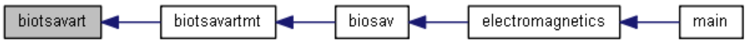
  References shape15w(), shape20h(), shape6w(), and shape8h().

  Referenced by biotsavartmt().

Here is the call graph for this function:



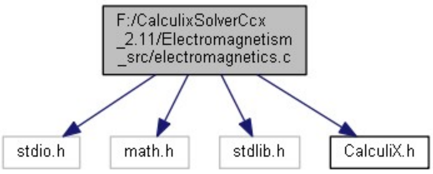
Here is the caller graph for this function:



electromagnetics.c

```
#include <stdio.h>
#include <math.h>
#include <stdlib.h>
#include "CalculiX.h"
```

Include dependency graph for electromagnetics.c:



Functions

- void **electromagnetics** (double **cop, **ITG** *nk, **ITG** **konp, **ITG** **ipkonp, char **lakonp, **ITG** *ne, **ITG** *nodeboun, **ITG** *ndirboun, double *xboun, **ITG** *nboun, **ITG** **ipompcp, **ITG** **nodempcp, double **coefmpcp, char **labmpcp, **ITG** *nmpc, **ITG** *nodeforc, **ITG** *ndirforc, double *xforc, **ITG** *nforc, **ITG** **nelemloadp, char **sideloadp, double *xload, **ITG** *nload, **ITG** *nactdof, **ITG** **icolp, **ITG** *jq, **ITG** **irowp, **ITG** *neq, **ITG** *nzl, **ITG** *nmethod, **ITG** **ikmpcp, **ITG** **ilmpcp, **ITG** *ikboun, **ITG** *ilboun, double *elcon, **ITG** *nelcon, double *rhcon, **ITG** *nrhcon, double *alcon, **ITG** *nalcon, double *alzero, **ITG** **ielmatp, **ITG** **ielorienp, **ITG** *norien, double *orab, **ITG** *ntmat_, double *t0, double *t1, double *t1old, **ITG** *ithermal, double *prestr, **ITG** *iprestr, double **voldp, **ITG** *iperturb, double *sti, **ITG** *nzs, **ITG** *kode, char *filab, **ITG** *idrct, **ITG** *jmax, **ITG** *jout, double *timepar, double *eme, double *xbounold, double *xforcold, double *xloadold, double *veold, double *accold, char *amname, double *amta, **ITG** *namta, **ITG** *nam, **ITG** *iamforc, **ITG** **iamloadp, **ITG** *iamt1, double *alpha, **ITG** *iexpl, **ITG** *iamboun, double *plicon, **ITG** *nplicon, double *plkcon, **ITG** *nplkcon, double **xstatep, **ITG** *npmat_, **ITG** *istep, double *ttime, char *matname, double *qaold, **ITG** *mi, **ITG** *isolver, **ITG** *ncmat_, **ITG** *nstate_, **ITG** *iumat, double *cs, **ITG** *mcs, **ITG** *nkon, double **enerp, **ITG** *mpcinfo, char *output, double *shcon, **ITG** *nshcon, double *cocon, **ITG** *ncocon, double *physcon, **ITG** *nflow, double *ctrl, char **setp, **ITG** *nset, **ITG** **istartsetp, **ITG** **iendsetp, **ITG** **ialsetp, **ITG** *nprint, char *prlab, char *prset, **ITG** *nener, **ITG** *ikforc, **ITG** *ilforc, double *trab, **ITG** *inotr, **ITG** *ntrans, double **fmpcp, char *cbody, **ITG** *ibody, double *xbody, **ITG** *nbody, double *xbodyold, **ITG** *ielprop, double *prop, **ITG** *ntie, char **tiesetp, **ITG** *itpamp, **ITG** *iviewfile, char *jobnamec, double **tietolp, **ITG** *nslavs, double *thicke, **ITG** *ics, **ITG** *nalset, **ITG** *nmpc_, **ITG** *nmat, char *typeboun, **ITG** *iaxial, **ITG** *nload_, **ITG** *nprop)

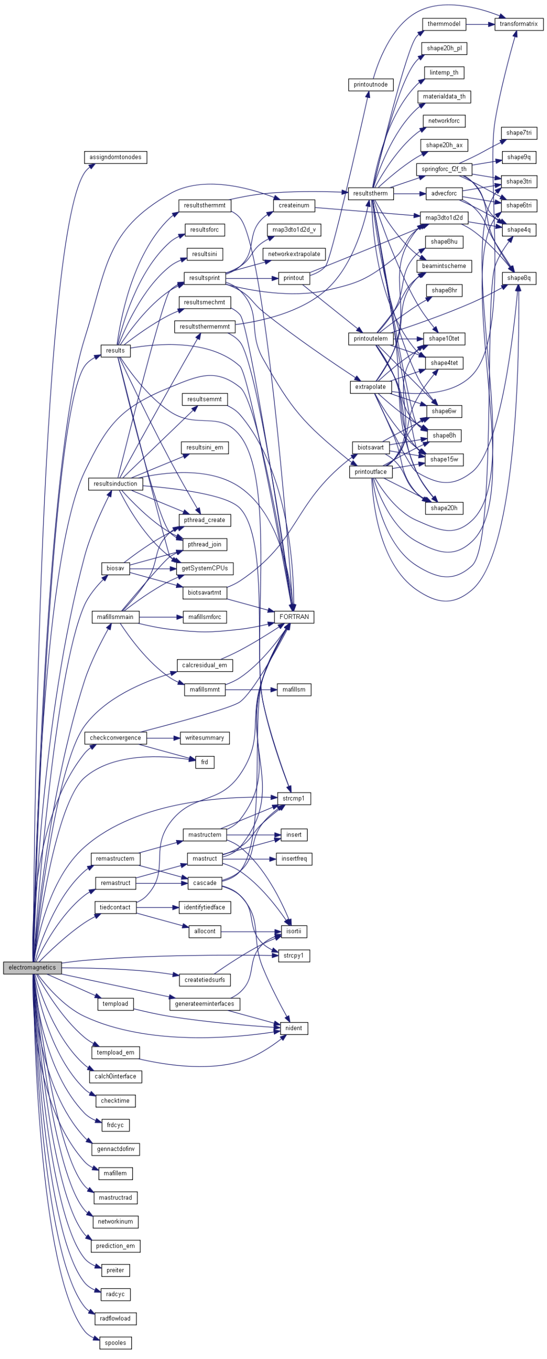
  References assigndomtonodes(), biosav(), calch0interface(), calcresidual_em(), checkconvergence(), checktime(), createinum(), createtiedsurfs(), DMEMSET, FORTRAN(), frd(), frdcyc(), generateeminterfaces(), gennactdofinv(), ITG, ITGFORMAT, mafillem(), mafillsmmain(), mastructrad(), networkinum(), nident(), NNEW, prediction_em(), preiter(), radcyc(), radflowload(), remastruct(), remastructem(), RENEW, results(), resultsinduction(), SFREE, spooles(), strcmp1(), strcpy1(), tempload(), tempload_em(), and tiedcontact().

  Referenced by main().

Here is the caller graph for this function:

electromagnetics
main

Here is the call graph for this function:

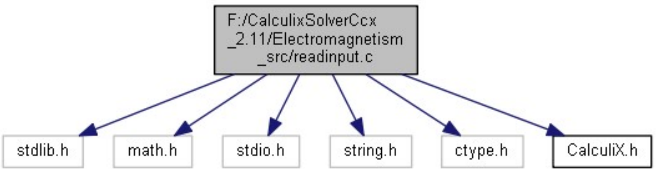


transformatrix
thermmodel
shape20h_pl
lintemp_th
materialdata_th
printoutnode
networkforc
shape7tri
shape20h_ax
shape9q
springforc_f2f_th
shape3tri
assigndomtonodes
resultstherm
advecforc
resultsthermmt
createinum
shape6tri
map3dto1d2d
shape4q
resultsforc
map3dto1d2d_v
shape8hu
resultsini
networkextrapolate
beamintscheme
resultsprint
printout
shape8q
shape8hr
resultsmechmt
resultsthermemmt
printoutelem
shape10tet
results
shape4tet
extrapolate
resultsemmt
shape6w
shape8h
biotsavart
resultsini_em
shape15w
printoutface
resultsinduction
shape20h
pthread_create
pthread_join
biosav
getSystemCPUs
biotsavartmt
mafillsmmain
mafillsmforc
FORTRAN
calcresidual_em
mafillsmmt
mafillsm
checkconvergence
writesummary
frd
strcmp1
mastructem
insert
remastructem
mastruct
insertfreq
remastruct
cascade
tiedcontact
identifytiedface
allocont
isortii
electromagnetics
strcpy1
createtiedsurfs
tempload
generateeminterfaces
nident
tempload_em
calch0interface
checktime
frdcyc
gennactdofinv
mafillem
mastructrad
networkinum
prediction_em
preiter
radcyc
radflowload
spooles

readinput.c

```
#include <stdlib.h>
#include <math.h>
#include <stdio.h>
#include <string.h>
#include <ctype.h>
#include "CalculiX.h"
```

Include dependency graph for readinput.c:



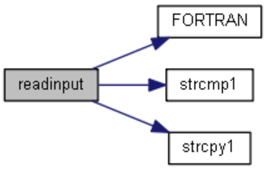
Functions

- void **readinput** (char *jobnamec, char **inpcp, **ITG** *nline, **ITG** *nset, **ITG** *ipoinp, **ITG** **inpp, **ITG** **ipoinpcp, **ITG** *ithermal)

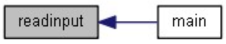
  References FORTRAN(), ITG, NNEW, RENEW, strcmp1(), and strcpy1().

  Referenced by main().

Here is the call graph for this function:



Here is the caller graph for this function:



C.1 CalculiX Solver Files

Input ⇒ **CalculiX** Solver uses an Input Desk file for input. The structure of a **CalculiX** Input Deck consists in set of keywords. Each keyword type could be complemented with a set of parameters, those parameters must be separated by commas and on the same line of the keyword they belong to (DHONDT 2016b).

An Input Deck file, used by **CalculiX** for the electromagnetism part, includes a model definition section describing the geometry and boundary conditions of the problem and one or more blocks defining the loads (DHONDT 2016b). That file is an ABAQUS Input Deck file that could be generated by HyperMesh Software or **cgx** package.

The Input Deck file structure is shown in figure C.1:

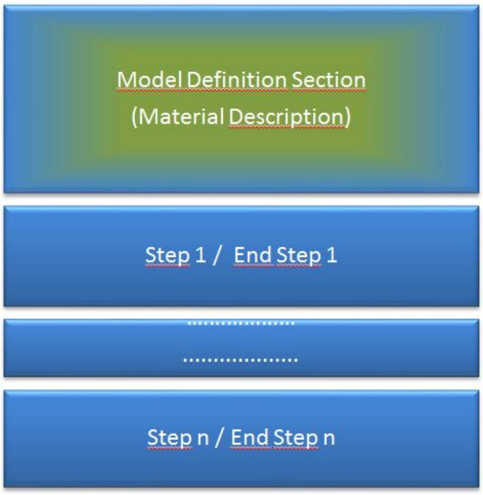


Image C.1: Input Deck Structure (image based on (DHONDT 2016b))

Input Deck file contents could be in upper or lower case, data in the same line must be separated my commas, at least two asterisk (**) are use for comment lines entries and one asterisk (*) is use for keyword card entry, blanks are not significant.

Output ⇒ CalculiX Solver generates four output files (DHONDT 2016b) :

- *xxxxxxx.frd* contains the information related for each layer when is expanded independently and, the displacements and stresses are interpolated/extrapolated accordingly. The *.frd* file format is described in detail in chapter 10 of the **CalculiX** USER 'S MANUAL for **CalculiX** GraphiX (**Wittig.2016**).
- *xxxxxx.cvg*: contains a summary of convergence information
- *xxxxxx.sta*: contains a summary of job information

- *xxxxxx.dat*: it is generated if, for example, NODE PRINT and EL PRINT are defined in the Input Deck file

C.1.1 Input Deck file (.inp)

An Input Deck file used for the Work Piece verification part is included here. In this case, this *.inp* file has been generated by the **cgx** package.

```
*INCLUDE,INPUT=Air.nam
*INCLUDE,INPUT=Coil.nam
*INCLUDE,INPUT=plus.nam
*INCLUDE,INPUT=minus.nam
*INCLUDE,INPUT=Part.nam
*INCLUDE,INPUT=n1.nam
***********************************
***********************************
*MATERIAL,NAME=COPPER
*DENSITY
8.96E-9
** In Kg/mm^3
*CONDUCTIVITY
385.2
** In W/(mm*K)
*SPECIFIC HEAT
387E6
** In J/(Kg*K)
*ELECTRICAL CONDUCTIVITY
46400.8
** 80% IACS (here depicted in S/mm)
*********************************
*MATERIAL,NAME=ALUMINIUM
*DENSITY
2.67E-9
** In Kg/mm^3
*CONDUCTIVITY
151
** In W/(mm*K)
*SPECIFIC HEAT
880E6
```

```
** In J/(Kg*K)
*ELECTRICAL CONDUCTIVITY
28072.484
** 48.4% IACS (here depicted in S/mm)
*MAGNETIC PERMEABILITY
1.256637e-3,2
*********************************
*MATERIAL,NAME=AIR
*DENSITY
1.185E-12
** In Kg/mm^3
*CONDUCTIVITY
2.4E-2
** In W/(mm*K)
*SPECIFIC HEAT
1012E6
** In J/(Kg*K)
*MAGNETIC PERMEABILITY
1.256637e-3,1
*************************************
** In H/mm (not correct)
*PHYSICAL CONSTANTS,ABSOLUTE ZERO=0,STEFAN BOLTZMANN=5.669E-8

*SHELL SECTION,ELSET=ECoil,MATERIAL=COPPER
1
** In mm
*SOLID SECTION,ELSET=EAir,MATERIAL=AIR
*SOLID SECTION,ELSET=EPart,MATERIAL=ALUMINIUM

*INITIAL CONDITIONS,TYPE=TEMPERATURE
NAir,300.
NPart,300.
NCoil,300.

****************************************
*STEP,INC=1000
*ELECTROMAGNETICS,direct
1.e-3,6.e-2
```

```
*BOUNDARY
Nn1,8,8
** AMPLITUDE for shift
*AMPLITUDE, NAME=ZSHIFT
*INCLUDE,INPUT=VoltageAC.txt
*BOUNDARY, AMPLITUDE=ZSHIFT
Nplus,11,11,8.054295e-1
Nminus,11,11,0
*NODE FILE
POT,U,NT
*EL FILE
ECD,EMFE,EMFB
*END STEP
```

where *all.msh, Air.nam, Coil.nam. plus.nam, minus.nam, Part.nam and n1.nam* files contains needed definitions for nodes and elements:

- *NODE, NSET=Nall
- *ELEMENT, TYPE=S4, ELSET=Eall
- *ELEMENT, TYPE=C3D8, ELSET=Eall
- *ELEMENT, TYPE=S4, ELSET=Eall
- *NSET,NSET=NCoil
- *ELSET,ELSET=ECoil
- *NSET,NSET=NAir
- *ELSET,ELSET=EAir
- *NSET,NSET=Nminus
- *NSET,NSET=Nn1
- *NSET,NSET=NPart
- *ELSET,ELSET=EPart
- *NSET,NSET=Nplus
- *NSET,NSET=Nplus

and VoltageAC.txt contains voltage amplitude values, like:

```
        ......
        ......
        0.0148,−0.99803
        0.0152,−0.99803
        0.0156,−0.98229
        0.016,−0.95106
        0.0164,−0.90483
        0.0168,−0.84433
        0.0172,−0.77051
        0.0176,−0.68455
        0.018,−0.58779
        0.0184,−0.48175
        0.0188,−0.36812
        ......
        ......
```

C.1.2 Results file (.frd)

An Results file generated by the **ccx** package for the Work Piece verification part is included here. In this case, this *.frd* file has 11.885.165 lines, so only few entry types and small parts of each one are shown:

```
    1C
    1UUSER
    1UDATE              09.april.2017
    1UTIME              21:29:42
    1UHOST
    1UPGM               CalculiX
    1UVERSION           Version 2.12
    1UCOMPILETIME       Fr 24. Mär 20:05:01 CET 2017
    1UDIR
    1UDBN
    1UMAT      1COPPER
    1UMAT      2ALUMINIUM
    1UMAT      3AIR
    2C                                 60603
                                                  1
 −1          273  0.00000E+00  1.12000E+02−6.50000E+01
 −1          274  2.18501E+01  1.09848E+02−6.50000E+01
 −1         275  2.18501E+01  1.09848E+02−5.68750E+01
 −1         276  0.00000E+00  1.12000E+02−5.68750E+01
 −1         277  0.00000E+00  1.07344E+02−6.50000E+01
. . . . . . .
. . . . . . .
 −3
     3C                              41472
                                                  1
−1            1      1      0      1
−2       45244      45258      45272      45286      45246      45260
         45274      45288
−1            2      1      0      1
−2       45286      45272      45300      45314      45288      45274
         45302      45316
−1            3      1      0      1
−2       45314      45300      45328      45342      45316      45302
         45330      45344
. . . . . . .
. . . . . . .
 −3
     1PSTEP                              1            0            1
   100CL   101  0.00000E+00          60603                       0     1
                      1
 −4  ELPOT          1      1
 −5  V              1      1      0      0
 −1         273  0.00000E+00
 −1         274  0.00000E+00
 −1         275  0.00000E+00
 −1         276  0.00000E+00
. . . . . . .
. . . . . . .
 −3
     1PSTEP                              2            0            1
   100CL   101  0.00000E+00          60603                       0     1
                      1
 −4  CURR           4      1
 −5   j1            1      2      1      0
 −5   j2            1      2      2      0
 −5   j3            1      2      3      0
 −5  ALL          1    2    0    0    1ALL
 −1         273 0.00000E+00 0.00000E+00 0.00000E+00
 −1         274 0.00000E+00 0.00000E+00 0.00000E+00
 −1         275 0.00000E+00 0.00000E+00 0.00000E+00
 −1         276 0.00000E+00 0.00000E+00 0.00000E+00
 −1         277 0.00000E+00 0.00000E+00 0.00000E+00
 −1         278 0.00000E+00 0.00000E+00 0.00000E+00
.......
.......
 −3
    1PSTEP                          3           1           1
  100CL  102 1.00000E−03        40891                     1    2
                 1
 −4  DISP         4    1
 −5  D1           1    2    1    0
 −5  D2           1    2    2    0
 −5  D3           1    2    3    0
 −5  ALL          1    2    0    0    1ALL
 −1         273 0.00000E+00 0.00000E+00 0.00000E+00
 −1         274 0.00000E+00 0.00000E+00 0.00000E+00
 −1         275 0.00000E+00 0.00000E+00 0.00000E+00
 −1         276 0.00000E+00 0.00000E+00 0.00000E+00
 −1         277 0.00000E+00 0.00000E+00 0.00000E+00
.......
.......
 −3
    1PSTEP                          4           1           1
  100CL  102 1.00000E−03        70971                     1    2
                 1
 −4  NDTEMP       1    1
 −5  T            1    1    0    0
 −1           1 2.48398E−01
 −1           2 2.48398E−01
 −1          11 2.48398E−01
.......
.......
 −3
    1PSTEP                          5           1           1
  100CL   102  1.00000E-03          40891                            1     2
                            1
-4   EMFE          4     1
-5   E1            1     2     1     0
-5   E2            1     2     2     0
-5   E3            1     2     3     0
-5   ALL           1     2     0     0     1ALL
-1          273  0.00000E+00  0.00000E+00  0.00000E+00
-1          274  0.00000E+00  0.00000E+00  0.00000E+00
-1          275  0.00000E+00  0.00000E+00  0.00000E+00
-1          276  0.00000E+00  0.00000E+00  0.00000E+00
.......
.......
-3
    1PSTEP                               6             1             1
  100CL   102  1.00000E-03          40891                            1     2
                            1
-4   EMFB          4     1
-5   B1            1     2     1     0
-5   B2            1     2     2     0
-5   B3            1     2     3     0
-5   ALL           1     2     0     0     1ALL
-1          273-6.57677E-06-2.96254E-03-8.72811E-04
-1          274-5.82257E-04-2.90316E-03-8.71201E-04
-1          275-7.39927E-04-3.68251E-03-1.50308E-03
-1          276-9.70539E-06-3.75792E-03-1.50546E-03
.......
.......
.......
.......
.......
.......
.......
.......
-3
9999
```

Matlab has been used to develop a lot of functions to support the performed analysis and verification. Few of these functions are included here.

C.1.3 Matlab Module: Master_12

```matlab
close all;
clear all;

%% Calculation of the deviation between CalculiX and analytical values
%Call to Reader.m routine to read the desired .frd file
[FileName,Path] = uigetfile('*.frd','Select the .frd file','/ya/ya129/ya12952/y/Work/Simulations');
PathName=fullfile(Path,FileName);
[Bdata,A_3C]=Reader(PathName);

%In Block 1 are stored the coordinates of the nodes in cartesian system
Bdata(1).Data=cell2mat(Bdata(1).Data);
%In Block 5 are stored the values of T0 (vector form xyz) for each node in
%the air
Bdata(5).Data=cell2mat(Bdata(5).Data);
Bdata(6).Data=cell2mat(Bdata(6).Data);
Bdata(4).Data=cell2mat(Bdata(4).Data);
Bdata(3).Data=cell2mat(Bdata(3).Data);
%Block 2 is difficult to change to number format

%Translation to the center of the cube (when needed)

[Nair_nummer,Nair_coord,Nair_H0_calculix,Nair_H_calculix]=GetAirNodes(Bdata(1).Data,Bdata(5).Data,Bdata(6).Data);

Nair_coord_cylindrical=Cylindrical(Nair_nummer,Nair_coord(:,2),Nair_coord(:,3),Nair_coord(:,4));

[I_analytic,I_FE,Nair_H0_analytic,Nair_H0_analyticFE,Subtraction,SubtractionFE,Quotient,QuotientFE,Err_abs,Err_absFE,Err_rel,Err_relFE,Nair_H_calculix_module,H_rel]=Check(Nair_nummer,Nair_coord_cylindrical(:,3),Nair_coord_cylindrical(:,4),Nair_H0_calculix,Nair_H_calculix);

```

```

Nair_SumUp=[Nair_coord Nair_H0_analytic(:,2) Nair_H0_analyticFE(:,2) arrayfun(@(x,y,z) module(x,y,z),Nair_H0_calculix(:,2),Nair_H0_calculix(:,3),Nair_H0_calculix(:,4)) Err_abs(:,2) Err_absFE(:,2) Err_rel(:,2) Err_relFE(:,2) Nair_H_calculix_module(:,2) (arrayfun(@(x,y,z) module(x,y,z),Nair_H0_calculix(:,2),Nair_H0_calculix(:,3),Nair_H0_calculix(:,4))-Nair_H_calculix_module(:,2)) H_rel(:,2) Nair_coord_cylindrical(:,2:4)];
Nair_SumUp_Table=array2table(Nair_SumUp,'VariableNames',{'NodeNumber','x','y','z','H0_analytic','H0_analyticFE','H0_Calculix','Absolute','AbsoluteFE','Relative','RelativeFE','H_module','P_gradient','H_relative','THETA','RHO','Z'});

%% Write results to text file
WriteErrors(Nair_SumUp,I_analytic,I_FE);

%% Scatter Plots
[Logical_RHO,Logical_Z]=ScatterPlot2D_Nodes(Nair_SumUp_Table);
[RHO_UpperNodes_plotx,RHO_UpperNodes_ploty]=Upper_Boundary(Nair_SumUp_Table);
ScatterPlot3D(Nair_SumUp_Table);
%% Contourslice

ContourSlice(Nair_SumUp_Table);

%% Deviation displayed by each cylindric coordinate
CoordinatePlots(Nair_coord_cylindrica
```

C.1.4 Matlab Module: WriteErrors

```
function WriteErrors(Nair_SumUp,I_analytic,I_FE)
fid=fopen('H_Data.txt','a+');

[FileNameLOG,Path] = uigetfile('*.LOG','Select the .LOG file','/ya/ya129/ya12952/y/Work/Simulations');
PathName=fullfile(Path,FileNameLOG);
fid_LOG=fopen(PathName,'r');

tline_LOG=fgetl(fid_LOG);
while ischar(tline_LOG)
tline_LOG=fgetl(fid_LOG);
if strfind(tline_LOG,'total memory used')
len=length(tline_LOG);
Memory_used=tline_LOG(19:len);
end
if strfind(tline_LOG,'memory allocation')
len=length(tline_LOG);
Memory_allocation=tline_LOG(19:len);
end
if strfind(tline_LOG,'total wall time')
len=length(tline_LOG);
Wall_time=tline_LOG(17:len);
end
if strfind(tline_LOG,'total CPU time')
len=length(tline_LOG);
CPU_time=tline_LOG(16:len);
end
if strfind(tline_LOG,'global efficiency')
len=length(tline_LOG);
Global_efficiency=tline_LOG(19:len);
end
end

fclose(fid_LOG);

if str2double(Memory_used((strfind(Memory_used,':')+1):(strfind(Memory_used,'GBytes')-1)))==0
Wall_time='0';
```

```
CPU_time='0';
Global_efficiency='100 %';
end

tline=fgetl(fid);
if ischar(tline)==0
fprintf(fid,'%295s\n','EType  Size_cm  EType  Diameter  Drag
    MeanAbsolute  MeanAbsoluteFE  MeanRelative  MeanRelativeFE
    MaxAbsolute  MaxAbsoluteFE  MaxRelative  MaxRelativeFE
    H_H0_relative  H_H0_variance  H_H0_StanDevi  IAnalytic  I_FE
            TotalMemoryUsed          MemoryAllocation
    TotalWallTime      TotalCPUTime  GlobalEfficiency');
end

prompt={'Element type','Size'};
dlg_title='Air';
num_lines=1;
%Default=C3D8_S4
AirName=inputdlg(prompt,dlg_title,num_lines);
prompt={'Element type','Diameter Divisions','Drag Divisions'};
dlg_title='Coil';
CoilName=inputdlg(prompt,dlg_title,num_lines);

Data={AirName{1,1},AirName{2,1},CoilName{1,1},CoilName{2,1},
    CoilName{3,1},mean(Nair_SumUp(:,8)),mean(Nair_SumUp(:,9)),
    mean(Nair_SumUp(:,10)),mean(Nair_SumUp(:,11)),max(Nair_SumUp
    (:,8)),max(Nair_SumUp(:,9)),max(Nair_SumUp(:,10)),max(
    Nair_SumUp(:,11)),mean(Nair_SumUp(:,14)),var(Nair_SumUp
    (:,14)),std(Nair_SumUp(:,14))};

fprintf(fid,'%5s %8s %6s %9s %5s % 1.10f % 1.12f % 1.10f % 1.12
    f % 1.9f % 1.11f % 1.9f % 1.11f % 1.11f % 1.11f % 1.11f %
    5.5f % 5.5f %18s %22s %14s %15s %17s\n',Data{1,:},I_analytic
    ,I_FE,Memory_used,Memory_allocation,Wall_time,CPU_time,
    Global_efficiency);

fclose(fid);
end
```

C.1.5 Matlab Module: Reader

```
% close all
% clear all

function [Bdata,A_3C]=Reader(FileName)

fid=fopen(FileName,'r');

%Preallocation of memory for the cell arrays contained in the structure
maxcols=8;

% Data of each block contained in Bdata (dynamic structure array)
Bdata=struct('Header',[],'Data',[]);

% Block Number
Nblock=0;
n=0;
ON=0;

tline=fgetl(fid);

while ischar(tline)
if strfind(tline,'2C')
Nblock=Nblock+1;
Nall_Bdata=str2double(tline(10:40));
Bdata(Nblock).Data=cell(Nall_Bdata,4);
Bdata(Nblock).Header=tline;
end

if strfind(tline,'100CL')
Nblock=Nblock+1;
Nall_entity=str2double(tline(26:50));
Bdata(Nblock).Header=cell(2,1);
Bdata(Nblock).Header{1}=tline;
tline=fgetl(fid);
Bdata(Nblock).Header{2}=tline;
Bdata(Nblock).Data=cell(Nall_entity,4);
```

```
%ON=1;
end

if strfind(tline,'3C')
Nblock=Nblock+1;
Eall_TotalNumber=str2double(tline(10:40));
Eall_Bdata=Eall_TotalNumber*2;
Bdata(Nblock).Data=cell(Eall_Bdata,maxcols);
Bdata(Nblock).Header=tline;
A_3C=zeros(Eall_Bdata,maxcols);

ON=1;
end

% Reading operation of one line
tline=fgetl(fid);

if strcmp(tline,' -3')==1
ON=0;
n=0;
end

if ON==1

if strfind(tline,'-1')
val={str2double(tline(4:13)),str2double(tline(14:18)),
    str2double(tline(19:23)),str2double(tline(24:28))};
n=n+1;
Bdata(Nblock).Data(n,1:4)=val;
A_3C(n,1)=str2double(tline(4:13));
A_3C(n,2)=str2double(tline(14:18));
A_3C(n,3)=str2double(tline(19:23));
A_3C(n,4)=str2double(tline(24:28));
end

if strfind(tline,'-2')
l=length(tline);
col=(l-3)/10;
val=cell(1,col);
```

```

n=n+1;
index=4;
for a=1:col
val{1,a}=str2double(tline(index:(index+9)));
A_3C(n,a)=str2double(tline(index:(index+9)));
index=index+10;
end
index=0;

Bdata(Nblock).Data(n,1:col)=val;

end

elseif strfind(tline,'-1')
l=length(tline);
column=(l-13)/12;
value=cell(1,column+1);
value{1,1}=str2double(tline(4:13));

n=n+1;
index=14;
for a=2:(column+1)
value{1,a}=str2double(tline(index:(index+11)));
index=index+12;
end
index=0;

Bdata(Nblock).Data(n,1:(column+1))=value;

end

end

fclose(fid);

end
```

C.1.6 Matlab Module: BiotSavart

```
function H0=BiotSavart(RHO,Z,I)
%Funtion to calculate analytically T0 according to Biot-Savart Law.

x=RHO;
%Length of the wire
length=1000;
z2=length/2-Z;
z1=(-length/2)-Z;
%Proof to know if the measurements are OK (This is only true if the point considered is within the scope of the wire):

H0=(I*(z2./sqrt(z2.^2+x.^2)-z1./sqrt(z1.^2+x.^2)))./(4*pi*x);

end
```

C.1.7 Matlab Module: GetAirNodes

```
function [Nair_nummer,Nair_coord,Nair_H0_calculix,Nair_H_calculix]=GetAirNodes(Nall,Nair_H0,Nair_H)
%In Nall are stored the coordinates of all the nodes.
%In Nair are stored the H0 value of the air nodes (vector with 3 %components).

%Find out which nodes correspond to those located in the air, where T0 is calculated.To do that,
%"intersect" function from Matlab is used. This funtion shows the elements in common between 2 arrays.

[Nair_nummer,I_coord,I_H0]=intersect(Nall(:,1),Nair_H0(:,1));

%In I_coord are stored the index array number of then nodes of those corresponding to the
%air nodes.

Nair_coord=Nall(I_coord,:);
Nair_H0_calculix=Nair_H0(I_H0,:);
Nair_H_calculix=Nair_H(I_H0,:);

Logical=Nair_coord(:,2)<1e-12 & Nair_coord(:,2)>(-1e-12) & Nair_coord(:,3)<1e-12 & Nair_coord(:,3)>(-1e-12);
Nair_coord(Logical,:)=[];
Nair_nummer(Logical,:)=[];
Nair_H0_calculix(Logical,:)=[];
Nair_H_calculix(Logical,:)=[];
end
```

C.1.8 Matlab Module: CoordinatePlots

```
function CoordinatePlots(Nair_coord_cylindrical,Err_rel,
    Err_relFE,H_rel)
%Deviation displayed by each cylindric coordinate

%% Z graph

%Eliminate the nodes in the center of the cube (nan problems)

Zvector = -550:100:550;

Nodes_Zlevel=cell(1,length(Zvector)-1);
Index_Nodes_Zlevel=cell(1,length(Zvector)-1);

for j = 1:length(Zvector)-1
Logical = Zvector(j)<=Nair_coord_cylindrical(:,4) &
    Nair_coord_cylindrical(:,4)<Zvector(j+1);
Nodes_Zlevel{1,j}=Nair_coord_cylindrical(Logical,1);
[Index_Nodes_Zlevel{1,j},Z_column]=find(Logical==1);
end

Zdeviation=cell(1,length(Zvector)-1);
ZdeviationFE=cell(1,length(Zvector)-1);
ZdeviationH=cell(1,length(Zvector)-1);
Zmean_rel=zeros(1,length(Zvector)-1);
Zmean_relFE=zeros(1,length(Zvector)-1);
Zmean_H_rel=zeros(1,length(Zvector)-1);

for i=1:length(Zvector)-1

Zdeviation{1,i}=Err_rel(Index_Nodes_Zlevel{1,i},2);
Zmean_rel(1,i)=mean(Zdeviation{1,i});
ZdeviationFE{1,i}=Err_relFE(Index_Nodes_Zlevel{1,i},2);
Zmean_relFE(1,i)=mean(ZdeviationFE{1,i});
ZdeviationH{1,i}=H_rel(Index_Nodes_Zlevel{1,i},2);
Zmean_H_rel(1,i)=mean(ZdeviationH{1,i});
end
figure;
plot(Zvector(1:(length(Zvector)-1)),Zmean_rel,'-ro','LineWidth'
,2);
figure;
plot(Zvector(1:(length(Zvector)-1)),Zmean_relFE,'-ro','LineWidth',2);
figure;
plot(Zvector(1:(length(Zvector)-1)),Zmean_H_rel,'-ro','LineWidth',2);

%% RHO graph
RHOvector = 0:60:720;

Nodes_RHOlevel=cell(1,length(RHOvector)-1);
Index_Nodes_RHOlevel=cell(1,length(RHOvector)-1);

for j = 1:length(RHOvector)-1
Logical = RHOvector(j)<=Nair_coord_cylindrical(:,3) & Nair_coord_cylindrical(:,3)<RHOvector(j+1);
Nodes_RHOlevel{1,j}=Nair_coord_cylindrical(Logical,1);
[Index_Nodes_RHOlevel{1,j},RHO_column]=find(Logical==1);
end

RHOdeviation=cell(1,length(RHOvector)-1);
RHOdeviationFE=cell(1,length(RHOvector)-1);
RHOdeviationH=cell(1,length(RHOvector)-1);
RHOmean_rel=zeros(1,length(RHOvector)-1);
RHOmean_relFE=zeros(1,length(RHOvector)-1);
RHOmean_H_rel=zeros(1,length(RHOvector)-1);

for i=1:length(RHOvector)-1
RHOdeviation{1,i}=Err_rel(Index_Nodes_RHOlevel{1,i},2);
RHOdeviationFE{1,i}=Err_relFE(Index_Nodes_RHOlevel{1,i},2);
RHOdeviationH{1,i}=H_rel(Index_Nodes_RHOlevel{1,i},2);
RHOmean_rel(1,i)=mean(RHOdeviation{1,i});
RHOmean_relFE(1,i)=mean(RHOdeviationFE{1,i});
RHOmean_H_rel(1,i)=mean(RHOdeviationH{1,i});
end
figure;
plot(RHOvector(1:(length(RHOvector)-1)),RHOmean_rel,'-bo','LineWidth',2);
```

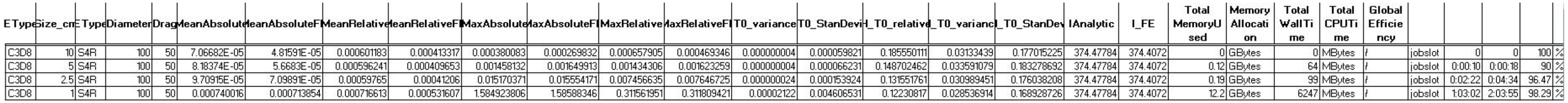
```matlab
figure;
plot(RHOvector(1:(length(RHOvector)-1)),RHOmean_relFE,'-bo','LineWidth',2);
figure;
plot(RHOvector(1:(length(RHOvector)-1)),RHOmean_H_rel,'-bo','LineWidth',2);

%% THETA graph
Nair_coord_cylindrical(:,2)=Nair_coord_cylindrical(:,2)*180/pi;
THETAvector = -180:32.727:180;

Nodes_THETAlevel=cell(1,length(THETAvector)-1);
Index_Nodes_THETAlevel=cell(1,length(THETAvector)-1);

for j = 1:length(THETAvector)-1
Logical = THETAvector(j)<=Nair_coord_cylindrical(:,2) & Nair_coord_cylindrical(:,2)<THETAvector(j+1);
Nodes_THETAlevel{1,j}=Nair_coord_cylindrical(Logical,1);
[Index_Nodes_THETAlevel{1,j},THETA_column]=find(Logical==1);
end

THETAdeviation=cell(1,length(THETAvector)-1);
THETAdeviationFE=cell(1,length(THETAvector)-1);
THETAdeviationH=cell(1,length(THETAvector)-1);
THETAmean_rel=zeros(1,length(THETAvector)-1);
THETAmean_relFE=zeros(1,length(THETAvector)-1);
THETAmean_H_rel=zeros(1,length(THETAvector)-1);

for i=1:length(Zvector)-1
THETAdeviation{1,i}=Err_rel(Index_Nodes_THETAlevel{1,i},2);
THETAdeviationFE{1,i}=Err_relFE(Index_Nodes_THETAlevel{1,i},2);
THETAdeviationH{1,i}=H_rel(Index_Nodes_THETAlevel{1,i},2);
THETAmean_rel(1,i)=mean(THETAdeviation{1,i});
THETAmean_relFE(1,i)=mean(THETAdeviationFE{1,i});
THETAmean_H_rel(1,i)=mean(THETAdeviationH{1,i});
end

figure;
plot(THETAvector(1:(length(THETAvector)-1)),THETAmean_rel,'-go'
    ,'LineWidth',2);
figure;
plot(THETAvector(1:(length(THETAvector)-1)),THETAmean_relFE,'-go','LineWidth',2);
figure;
plot(THETAvector(1:(length(THETAvector)-1)),THETAmean_H_rel,'-go','LineWidth',2);

end
```

D.1 H and T0 support tables

Within the functions developed with Matlab, some have been implemented for the specific management of the data obtained during the simulation sessions.

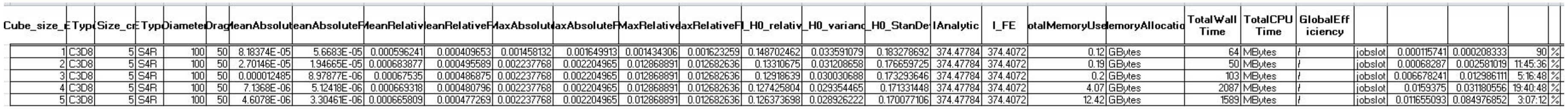
It is very important to collect this information in order to be able to manage it and use it later, since it allows, if necessary, to work in a system alternative to that of the tests and development.

The tables that have been generated are:



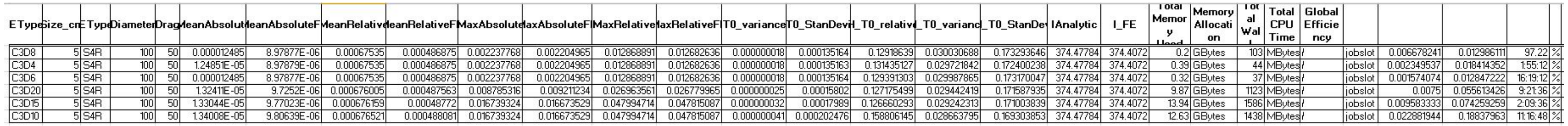
ETyp	Size_cm	EType	Diameter	Drag	MeanAbsolut	MeanAbsoluteF	MeanRelative	MeanRelativeF	MaxAbsolute	MaxAbsoluteF	MaxRelative	MaxRelativeF	T0_variance	T0_StanDevi	H_T0_relative	_T0_varianc	_T0_StanDev	IAnalytic	I_FE	Total MemoryUsed	Memory Allocation	Total WallTime	Total CPUTime	Global Efficiency					
C3D8	10	S4R	100	50	7.06682E-05	4.81591E-05	0.000601183	0.000413317	0.000380083	0.000269832	0.000657905	0.000469346	0.000000004	0.000059821	0.185550111	0.03133439	0.177015225	374.47784	374.4072	0	GBytes	0	MBytes	/	jobslot	0	0	100	%
C3D8	5	S4R	100	50	8.18374E-05	5.6683E-05	0.000596241	0.000409653	0.001458132	0.001649913	0.001434306	0.001623259	0.000000004	0.000066231	0.148702462	0.033591079	0.183278692	374.47784	374.4072	0.12	GBytes	64	MBytes	/	jobslot	0:00:10	0:00:18	90	%
C3D8	2.5	S4R	100	50	9.70915E-05	7.09891E-05	0.00059765	0.00041206	0.015170371	0.015554171	0.007456635	0.007646725	0.000000024	0.000153924	0.131551761	0.030989451	0.176038208	374.47784	374.4072	0.19	GBytes	99	MBytes	/	jobslot	0:02:22	0:04:34	96.47	%
C3D8	1	S4R	100	50	0.000740016	0.000713854	0.000716613	0.000531607	1.584923806	1.58588346	0.311561951	0.311809421	0.00002122	0.004606531	0.12230817	0.028536914	0.168928726	374.47784	374.4072	12.2	GBytes	6247	MBytes	/	jobslot	1:03:02	2:03:55	98.29	%

Image D.1: File: H_T0_Data_DifferentMeshes_1m.txt



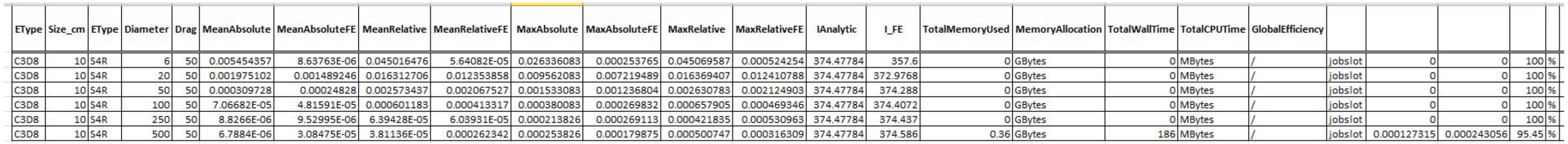
Cube_size_E	ETyp	Size_cr	ETyp	Diamete	Drag	MeanAbsolut	MeanAbsoluteF	MeanRelativ	MeanRelativeF	MaxAbsolut	MaxAbsoluteF	MaxRelative	MaxRelativeF	_H0_relativ	_H0_varianc	_H0_StanDe	IAnalytic	I_FE	TotalMemoryUse	MemoryAllocatio	TotalWall Time	TotalCPU Time	GlobalEfficiency					
1	C3D8	5	S4R	100	50	8.18374E-05	5.6683E-05	0.000596241	0.000409653	0.001458132	0.001649913	0.001434306	0.001623259	0.148702462	0.033591079	0.183278692	374.47784	374.4072	0.12	GBytes	64	MBytes	/	jobslot	0.000115741	0.000208333	90	%
2	C3D8	5	S4R	100	50	2.70146E-05	1.94665E-05	0.000683877	0.000495589	0.002237768	0.002204965	0.012868891	0.012682636	0.13310675	0.031208658	0.176659725	374.47784	374.4072	0.19	GBytes	50	MBytes	/	jobslot	0.00068287	0.002581019	11:45:36	%
3	C3D8	5	S4R	100	50	0.000012485	8.97877E-06	0.00067535	0.000486875	0.002237768	0.002204965	0.012868891	0.012682636	0.12918639	0.030030688	0.173293646	374.47784	374.4072	0.2	GBytes	103	MBytes	/	jobslot	0.006678241	0.012986111	5:16:48	%
4	C3D8	5	S4R	100	50	7.1368E-06	5.12418E-06	0.000669318	0.000480796	0.002237768	0.002204965	0.012868891	0.012682636	0.127425804	0.029354465	0.171331448	374.47784	374.4072	4.07	GBytes	2087	MBytes	/	jobslot	0.0159375	0.031180556	19:40:48	%
5	C3D8	5	S4R	100	50	4.6078E-06	3.30461E-06	0.000665809	0.000477269	0.002237768	0.002204965	0.012868891	0.012682636	0.126373698	0.028926222	0.170077106	374.47784	374.4072	12.42	GBytes	1589	MBytes	/	jobslot	0.011655093	0.084976852	3:07:12	%

Image D.2: File: H_T0_Data_DifferentSizes.txt



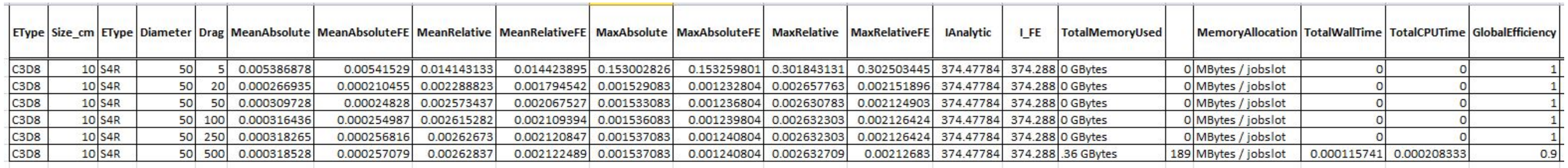
ETyp	Size_cm	EType	Diameter	Drag	MeanAbsolut	MeanAbsoluteF	MeanRelative	MeanRelativeF	MaxAbsolute	MaxAbsoluteF	MaxRelative	MaxRelativeF	T0_variance	T0_StanDevi	H_T0_relative	_T0_varianc	_T0_StanDev	IAnalytic	I_FE	Total Memory Used	Memory Allocation	Total Wal	Total CPU Time	Global Efficiency					
C3D8	5	S4R	100	50	0.000012485	8.97877E-06	0.00067535	0.000486875	0.002237768	0.002204965	0.012868891	0.012682636	0.000000018	0.000135164	0.12918639	0.030030688	0.173293646	374.47784	374.4072	0.2	GBytes	103	MBytes	/	jobslot	0.006678241	0.012986111	97.22	%
C3D4	5	S4R	100	50	1.24851E-05	8.97879E-06	0.00067535	0.000486875	0.002237768	0.002204965	0.012868891	0.012682636	0.000000018	0.000135163	0.131435127	0.029721842	0.172400238	374.47784	374.4072	0.39	GBytes	44	MBytes	/	jobslot	0.002349537	0.018414352	1:55:12	%
C3D6	5	S4R	100	50	0.000012485	8.97877E-06	0.00067535	0.000486875	0.002237768	0.002204965	0.012868891	0.012682636	0.000000018	0.000135164	0.129391303	0.029987865	0.173170047	374.47784	374.4072	0.32	GBytes	37	MBytes	/	jobslot	0.001574074	0.012847222	16:19:12	%
C3D20	5	S4R	100	50	1.32411E-05	9.7252E-06	0.000676005	0.000487563	0.008785316	0.009211234	0.026963561	0.026779965	0.000000025	0.00015802	0.127175499	0.029442419	0.171587935	374.47784	374.4072	9.87	GBytes	1123	MBytes	/	jobslot	0.0075	0.055613426	9:21:36	%
C3D15	5	S4R	100	50	1.33044E-05	9.77023E-06	0.000676159	0.00048772	0.016739324	0.016673529	0.047994714	0.047815087	0.000000032	0.00017989	0.126660293	0.029242313	0.171003839	374.47784	374.4072	13.94	GBytes	1586	MBytes	/	jobslot	0.009583333	0.074259259	2:09:36	%
C3D10	5	S4R	100	50	1.34008E-05	9.80639E-06	0.000676521	0.000488081	0.016739324	0.016673529	0.047994714	0.047815087	0.000000041	0.000202476	0.158806145	0.028663795	0.169303853	374.47784	374.4072	12.63	GBytes	1438	MBytes	/	jobslot	0.022881944	0.18837963	11:16:48	%

Image D.3: File: H_T0_ElementType_Data_ok_3m.txt



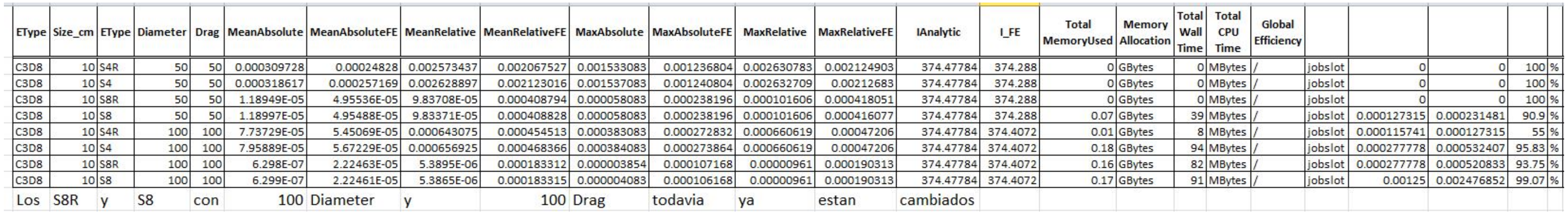
EType	Size_cm	EType	Diameter	Drag	MeanAbsolute	MeanAbsoluteFE	MeanRelative	MeanRelativeFE	MaxAbsolute	MaxAbsoluteFE	MaxRelative	MaxRelativeFE	IAnalytic	I_FE	TotalMemoryUsed	MemoryAllocation	TotalWallTime	TotalCPUTime	GlobalEfficiency					
C3D8	10	S4R	6	50	0.005454357	8.63763E-06	0.045016476	5.64082E-05	0.026336083	0.000253765	0.045069587	0.000524254	374.47784	357.6	0	GBytes	0	MBytes	/	jobslot	0	0	100	%
C3D8	10	S4R	20	50	0.001975102	0.001489246	0.016312706	0.012353858	0.009562083	0.007219489	0.016369407	0.012410788	374.47784	372.9768	0	GBytes	0	MBytes	/	jobslot	0	0	100	%
C3D8	10	S4R	50	50	0.000309728	0.00024828	0.002573437	0.002067527	0.001533083	0.001236804	0.002630783	0.002124903	374.47784	374.288	0	GBytes	0	MBytes	/	jobslot	0	0	100	%
C3D8	10	S4R	100	50	7.06682E-05	4.81591E-05	0.000601183	0.000413317	0.000380083	0.000269832	0.000657905	0.000469346	374.47784	374.4072	0	GBytes	0	MBytes	/	jobslot	0	0	100	%
C3D8	10	S4R	250	50	8.8266E-06	9.52995E-06	6.39428E-05	6.03931E-05	0.000213826	0.000269113	0.000421835	0.000530963	374.47784	374.437	0	GBytes	0	MBytes	/	jobslot	0	0	100	%
C3D8	10	S4R	500	50	6.7884E-06	3.08475E-05	3.81136E-05	0.000262342	0.000253826	0.000179875	0.000500747	0.000316309	374.47784	374.586	0.36	GBytes	186	MBytes	/	jobslot	0.000127315	0.000243056	95.45	%

Image D.4: File: H0_Data_Diameter.txt



EType	Size_cm	EType	Diameter	Drag	MeanAbsolute	MeanAbsoluteFE	MeanRelative	MeanRelativeFE	MaxAbsolute	MaxAbsoluteFE	MaxRelative	MaxRelativeFE	IAnalytic	I_FE	TotalMemoryUsed		MemoryAllocation	TotalWallTime	TotalCPUTime	GlobalEfficiency
C3D8	10	S4R	50	5	0.005386878	0.00541529	0.014143133	0.014423895	0.153002826	0.153259801	0.301843131	0.302503445	374.47784	374.288	0 GBytes	0	MBytes / jobslot	0	0	1
C3D8	10	S4R	50	20	0.000266935	0.000210455	0.002288823	0.001794542	0.001529083	0.001232804	0.002657763	0.002151896	374.47784	374.288	0 GBytes	0	MBytes / jobslot	0	0	1
C3D8	10	S4R	50	50	0.000309728	0.00024828	0.002573437	0.002067527	0.001533083	0.001236804	0.002630783	0.002124903	374.47784	374.288	0 GBytes	0	MBytes / jobslot	0	0	1
C3D8	10	S4R	50	100	0.000316436	0.000254987	0.002615282	0.002109394	0.001536083	0.001239804	0.002632303	0.002126424	374.47784	374.288	0 GBytes	0	MBytes / jobslot	0	0	1
C3D8	10	S4R	50	250	0.000318265	0.000256816	0.00262673	0.002120847	0.001537083	0.001240804	0.002632303	0.002126424	374.47784	374.288	0 GBytes	0	MBytes / jobslot	0	0	1
C3D8	10	S4R	50	500	0.000318528	0.000257079	0.00262837	0.002122489	0.001537083	0.001240804	0.002632709	0.00212683	374.47784	374.288	.36 GBytes	189	MBytes / jobslot	0.000115741	0.000208333	0.9

Image D.5: File: H0_Data_Drag.txt



EType	Size_cm	EType	Diameter	Drag	MeanAbsolute	MeanAbsoluteFE	MeanRelative	MeanRelativeFE	MaxAbsolute	MaxAbsoluteFE	MaxRelative	MaxRelativeFE	IAnalytic	I_FE	Total MemoryUsed	Memory Allocation	Total Wall Time	Total CPU Time	Global Efficiency					
C3D8	10	S4R	50	50	0.000309728	0.00024828	0.002573437	0.002067527	0.001533083	0.001236804	0.002630783	0.002124903	374.47784	374.288	0	GBytes	0	MBytes	/	jobslot	0	0	100	%
C3D8	10	S4	50	50	0.000318617	0.000257169	0.002628897	0.002123016	0.001537083	0.001240804	0.002632709	0.00212683	374.47784	374.288	0	GBytes	0	MBytes	/	jobslot	0	0	100	%
C3D8	10	S8R	50	50	1.18949E-05	4.95536E-05	9.83708E-05	0.000408794	0.000058083	0.000238196	0.000101606	0.000418051	374.47784	374.288	0	GBytes	0	MBytes	/	jobslot	0	0	100	%
C3D8	10	S8	50	50	1.18997E-05	4.95488E-05	9.83371E-05	0.000408828	0.000058083	0.000238196	0.000101606	0.000416077	374.47784	374.288	0.07	GBytes	39	MBytes	/	jobslot	0.000127315	0.000231481	90.9	%
C3D8	10	S4R	100	100	7.73729E-05	5.45069E-05	0.000643075	0.000454513	0.000383083	0.000272832	0.000660619	0.00047206	374.47784	374.4072	0.01	GBytes	8	MBytes	/	jobslot	0.000115741	0.000127315	55	%
C3D8	10	S4	100	100	7.95889E-05	5.67229E-05	0.000656925	0.000468366	0.000384083	0.000273864	0.000660619	0.00047206	374.47784	374.4072	0.18	GBytes	94	MBytes	/	jobslot	0.000277778	0.000532407	95.83	%
C3D8	10	S8R	100	100	6.298E-07	2.22463E-05	5.3895E-06	0.000183312	0.000003854	0.000107168	0.00000961	0.000190313	374.47784	374.4072	0.16	GBytes	82	MBytes	/	jobslot	0.000277778	0.000520833	93.75	%
C3D8	10	S8	100	100	6.299E-07	2.22461E-05	5.3865E-06	0.000183315	0.000004083	0.000106168	0.00000961	0.000190313	374.47784	374.4072	0.17	GBytes	91	MBytes	/	jobslot	0.00125	0.002476852	99.07	%
Los	S8R	y	S8	con	100	Diameter	y	100	Drag	todavia	ya	estan	cambiados											

Image D.6: File: H0_Data_EtypeMod.txt