TECHNISCHE UNIVERSITÄT
ILMENAU

Numerical Modeling of an Inductively Coupled Plasma (ICP)

Master Thesis

von

Christian Busse

Fakultät für Maschinenbau

Fachgebiet Fertigungstechnik
Prof. Dr.-Ing. habil J. P. Bergmann

Betreuer TU Ilmenau	Betreuer KNRTU-KAI
Dr. Jörg Hildebrand	PhD Ilya Tsivilskiy

Ilmenau, 18. November 2019

DOI: 10.22032/dbt.40314
URN: urn:nbn:de:gbv:ilm1-2019200520

Kurzfassung

Die stetig voranschreitende Entwicklung und der zunehmende Einsatz von additiven Fertigungstechnologien in der Industrie führt zu einer verstärkten Nachfrage nach hochwertigen Metall- und Keramikpulverwerkstoffen, welche strenge Qualitätsanforderungen wie chemische Reinheit, Packungsdichte und Fließfähigkeit erfüllen müssen. Die Wärmebehandlung von Metallpulverpartikeln, die sogenannte Sphäroidisierung, mit induktiv gekoppeltem Plasma (ICP) wird bereits erfolgreich eingesetzt, um die Pulverqualität signifikant zu verbessern. Um die experimentelle Forschung mit dem selbst entwickelten ICP-Brenner an der Kasaner Nationalen Technischen Forschungs-Universität (KNRTU-KAI) zu unterstützen, wird ein numerisches Simulationsmodell mit der kostenfreien Open-Source Software-Plattform OpenFOAM entwickelt. Hierzu wird ein zweidimensionaler achsensymmetrischer Modellierungsansatz für die Simulation des induktiv gekoppelten Plasmabrenners verfolgt. Um die beim ICP auftretenden hochfrequenten elektromagnetischen Felder ($f \approx 3\,\mathrm{MHz}$) effizient zu berechnen, wird dazu ein Löser auf Basis der Vektorpotentialformulierung der Maxwell-Gleichungen und der Frequenzbereich-Methode implementiert. Hierbei wird die in *foam-extend*, einem OpenFOAM-Entwicklungszweig, zur Verfügung stehende Blockmatrizen-Bibliothek (`fvBlockMatrix.H`) genutzt, um die Gleichungen gekoppelt zu lösen. Der Elektromagnetfeld-Löser wird anhand analytischer Lösungen validiert und anschließend mit dem *rhoPimpleFoam* Code, einem Standard-Strömungslöser von OpenFOAM, zusammengeführt. Daraus wird ein Löser kompiliert, der sowohl für die stationäre als auch transiente Berechnung des Plasmas verwendet wird. Die Simulationsergebnisse mit dem eigens entwickelten Plasma-Löser wurden mit Literaturdaten für den Plasmabrenner TEKNA-PL50 verglichen und zeigen gute Übereinstimmung.

Abstract

The growing industrial use of additive manufacturing technologies leads to an increased demand for high-quality metal and ceramic powder materials, which must meet strict quality requirements such as chemical purity, packing density and flowability. The heat treatment of powder particles, the so-called spheroidization, using inductively coupled plasma (ICP) technology is being used successfully to improve powder quality significantly. In order to support experimental research with the in-house built ICP torch at the Kazan National Research Technical University named after A.N. Tupolev - KAI (KNRTU-KAI), a numerical simulation model is developed using the free and open-source software platform OpenFOAM. Therefore, a two-dimensional axisymmetric modeling approach is chosen to simulate the inductively coupled plasma torch. To efficiently calculate the high-frequency electromagnetic fields ($f \approx 3\,\mathrm{MHz}$) generated by the ICP, a solver based on the vector potential formulation of the Maxwell equations and the frequency domain method is implemented using the block coupled matrix library (`fvBlockMatrix.H`) provided by *foam-extend*, which is a forked version of OpenFOAM. The electromagnetic field solver is validated using analytical solutions and then merged with the *rhoPimpleFoam* code, a standard flow solver in OpenFOAM, to compile a plasma solver for steady-state and transient calculations. The simulation results obtained by means of the custom-developed plasma solver have been compared with literature data for the plasma torch TEKNA-PL50 and show good agreement.

Acknowledgments

This study abroad would have been impossible without the financial support from the Erasmus+ program and the Technical University of Ilmenau.

I would like to thank my supervisors Ilya Tsivilskiy and Dr Jörg Hildebrand for their continuous interest, support and guidance through this work.

Also thanks to my friends and colleagues at the German-Russian Institute of Advanced Technologies (GRIAT), who have helped me with everyday life questions and made me enjoy my stay in Kazan.

Contents

Abbreviations and Symbols

AM	Additive Manufacturing
FDM	Finite-Difference-Method
FEM	Finite-Element-Method
FVM	Finite-Volume-Method
ICP	Inductively Coupled Plasma
KNRTU-KAI	Kazan National Research Technical University named after A.N. Tupolev - KAI
LTE	Local Thermodynamic Equilibrium
OpenFOAM	Open Source Field Operation and Manipulation
PISO	Pressure Implicit with Splitting of Operator
RF	Radio Frequency
SIMPLE	Semi-Implicit Method for Pressure-Linked Equations
TU Ilmenau	Technische Universität Ilmenau
$\mathbf{A}$	Magnetic vector potential $(\mathrm{V\,s\,m^{-1}})$
$\widetilde{\mathbf{A}}$	Complex magnetic vector potential $(\mathrm{V\,s\,m^{-1}})$
$\widetilde{\mathbf{A}}_{\mathrm{I}}$	Imaginary part of the magnetic vector potential $(\mathrm{V\,s\,m^{-1}})$
$\widetilde{\mathbf{A}}_{\mathrm{I,x}}$	Imaginary part of the magnetic vector potential, x-component $(\mathrm{V\,s\,m^{-1}})$
$\widetilde{\mathbf{A}}_{\mathrm{I,y}}$	Imaginary part of the magnetic vector potential, y-component $(\mathrm{V\,s\,m^{-1}})$
$\widetilde{\mathbf{A}}_{\mathrm{I,z}}$	Imaginary part of the magnetic vector potential, z-component $(\mathrm{V\,s\,m^{-1}})$
$\widetilde{\mathbf{A}}_{\mathrm{R}}$	Real part of the magnetic vector potential $(\mathrm{V\,s\,m^{-1}})$
$\widetilde{\mathbf{A}}_{\mathrm{R,x}}$	Real part of the magnetic vector potential, x-component $(\mathrm{V\,s\,m^{-1}})$
$\widetilde{\mathbf{A}}_{\mathrm{R,y}}$	Real part of the magnetic vector potential, y-component $(\mathrm{V\,s\,m^{-1}})$
$\widetilde{\mathbf{A}}_{\mathrm{R,z}}$	Real part of the magnetic vector potential, z-component $(\mathrm{V\,s\,m^{-1}})$
B_{mag}	Magnitude of the magnetic flux density $(\mathrm{V\,s\,m^{-2}})$
$\mathbf{B}$	Magnetic flux density $(\mathrm{V\,s\,m^{-2}})$
$\widetilde{B}_{\mathrm{y}}$	Complex magnetic flux density y-component $(\mathrm{V\,s\,m^{-2}})$
$\widetilde{\mathbf{B}}$	Fourier transform of the magnetic flux density $(\mathrm{V\,s\,m^{-2}})$
$\widetilde{\mathbf{B}}^{*}$	Conjugate of the complex magnetic flux vector $(\mathrm{V\,s\,m^{-2}})$
$\widetilde{\mathbf{B}}_{\mathrm{I}}$	Imaginary part of the complex magnetic flux density $(\mathrm{V\,s\,m^{-2}})$
$\widetilde{\mathbf{B}}_{\mathrm{R}}$	Real part of the complex magnetic flux density $(\mathrm{V\,s\,m^{-2}})$
c_{p}	Specfic heat capacitance $(\mathrm{J\,kg^{-1}\,K^{-1}})$
d_{c}	Coil wire diameter (m)
D_{A}	Vector potential matrix off-diagonal coefficient $(\mathrm{rad\,m^{-2}})$

$\mathbf{D}$	Electric displacement field ($\mathrm{C\,m^{-2}}$)
$\frac{\partial \mathbf{D}}{\partial t}$	Displacement current density ($\mathrm{C\,m^{-2}\,s^{-1}}$)
$\mathbf{e}_\mathrm{x}$	Unit vector
$\mathbf{e}_\mathrm{y}$	Unit vector
$\mathbf{e}_\mathrm{z}$	Unit vector
$\mathbf{E}$	Electric field strength ($\mathrm{V\,m^{-1}}$)
$\mathbf{E}_\mathrm{ind}$	Induced electric field ($\mathrm{V\,m^{-1}}$)
$\widetilde{\mathbf{E}}$	Complex electric field ($\mathrm{V\,m^{-1}}$)
$\mathbf{f}_\mathrm{L}$	Lorentz force density ($\mathrm{N\,m^{-3}}$)
$\overline{\mathbf{f}}_\mathrm{L}$	Time-averaged orentz force density ($\mathrm{N\,m^{-3}}$)
h	Enthalpy ($\mathrm{J\,kg^{-1}}$)
H_mag	Magnitude of the magnetic field strength ($\mathrm{A\,m^{-1}}$)
$\mathbf{H}$	Magnetic field strength ($\mathrm{A\,m^{-1}}$)
I_coil	Coil current (A)
j	Imaginary Unit
J_y	Coil current density, y-component ($\mathrm{A\,m^{-2}}$)
$\widetilde{J}_\mathrm{z}$	Complex coil current density, z-component ($\mathrm{A\,m^{-2}}$)
$\mathbf{J}$	Current density ($\mathrm{A\,m^{-2}}$)
$\widetilde{\mathbf{J}}$	Complex induced current density ($\mathrm{A\,m^{-2}}$)
$\widetilde{\mathbf{J}}^*$	Conjugate of the complex current density ($\mathrm{A\,m^{-2}}$)
$\widetilde{\mathbf{J}}_\mathrm{I}$	Imaginary part of the current density ($\mathrm{A\,m^{-2}}$)
$\widetilde{\mathbf{J}}_\mathrm{R}$	Real part of the current density ($\mathrm{A\,m^{-2}}$)
$\mathbf{J}_\mathrm{coil}$	Coil current density ($\mathrm{A\,m^{-2}}$)
$\widehat{\mathbf{J}}_\mathrm{coil}$	Current density amplitude ($\mathrm{A\,m^{-2}}$)
$\widetilde{\mathbf{J}}_\mathrm{coil}$	Complex coil current density ($\mathrm{A\,m^{-2}}$)
$\widetilde{\mathbf{J}}_\mathrm{coil,R}$	Real part of the complex coil current density ($\mathrm{A\,m^{-2}}$)
$\widetilde{\mathbf{J}}_\mathrm{coil,x}$	Complex coil current density ($\mathrm{A\,m^{-2}}$)
$\widetilde{\mathbf{J}}_\mathrm{coil,y}$	Complex coil current density ($\mathrm{A\,m^{-2}}$)
$\widetilde{\mathbf{J}}_\mathrm{coil,z}$	Complex coil current density ($\mathrm{A\,m^{-2}}$)
$\mathbf{J}_\mathrm{ind}$	Induced current density ($\mathrm{A\,m^{-2}}$)
$\widetilde{\mathbf{J}}_\mathrm{ind}$	Complex induced current density ($\mathrm{A\,m^{-2}}$)
k	Thermal conductivity ($\mathrm{W\,m^{-1}\,K^{-1}}$)
L_EM	Length of the electromagnetic computational domain (m)
N	Number of cells
$\mathbf{n}$	Normal vector
p	Pressure (Pa)
P_diss	Dissipated plasma power (W)
q_J	Volumetric Joule heat ($\mathrm{W\,m^{-3}}$)
$\overline{q}_\mathrm{J}$	Time-averaged volumetric Joule heat ($\mathrm{W\,m^{-3}}$)
q_R	Radiative heat loss ($\mathrm{W\,m^{-3}}$)
$\mathbf{r}$	Position vector (m)
R_c	Coil radius (m)
R_EM	Radial distance of the electromagnetic computational domain (m)
S_EM	Square size of the electromagnetic computational domain (m)
T	Temperature (K)
$\mathbf{u}$	Fluid velocity vector ($\mathrm{m\,s^{-1}}$)
$\mathbf{u}_\mathrm{in}$	Inlet velocity ($\mathrm{m\,s^{-1}}$)
x_c	Radial coordinate, starting at the torch wall in skin (m)

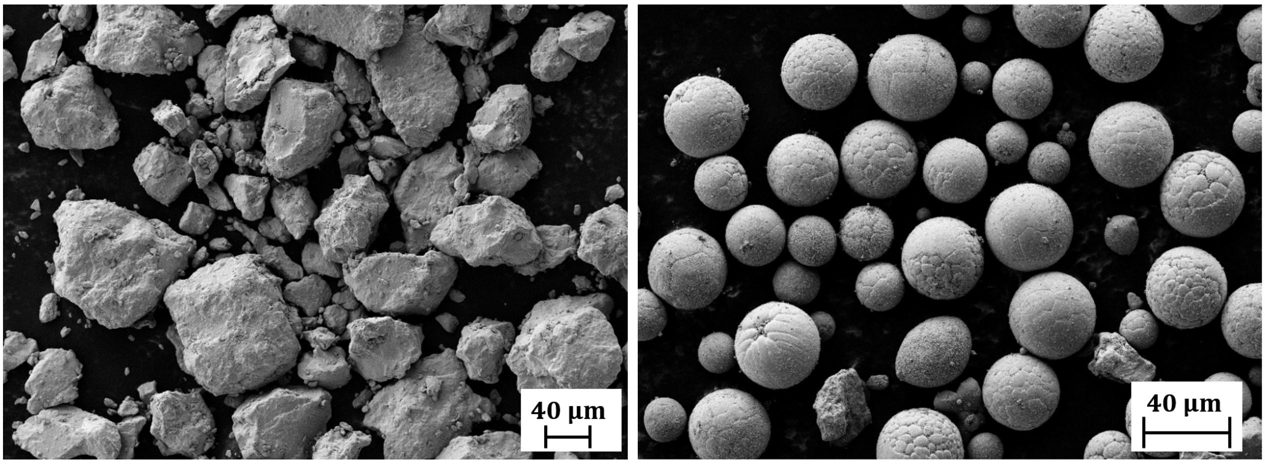
$\widehat{\mathbf{y}}$	Solution vector
z_{c}	Axial coordinate, starting at the central coil turn (m)
α_{eff}	Effective laminar thermal diffusivity ($\mathrm{kg\,m^{-1}\,s^{-1}}$)
δ	Skin depth (m)
$\Delta\rho$	Error mass density approximation (%)
Δx	Cell size in x-direction (m)
Δz	Cell size in z-direction (m)
γ	Auxiliary constant
ϕ	Flux ($\mathrm{kg\,m^{-2}\,s^{-1}}$)
ρ_{v}	Volume charge density ($\mathrm{C\,m^{-3}}$)
ρ	Density ($\mathrm{kg\,m^{-3}}$)
σ	Electrical conductivity ($\mathrm{S\,m^{-1}}$)
$\boldsymbol{\tau}$	Stress tensor ($\mathrm{N\,m^{-2}}$)
μ	Dynamic viscosity ($\mathrm{Pa\,s}$)
μ_0	Permability of vacuum ($\mathrm{V\,s\,A^{-1}\,m^{-1}}$)
μ_{mag}	Magnetic permability ($\mathrm{V\,s\,A^{-1}\,m^{-1}}$)
ω	Frequency of the coil current ($\mathrm{rad\,s^{-1}}$)

CHAPTER 1

Introduction

1.1. Motivation: Plasma Spheroidization

The emergence of Additive Manufacturing (AM) technologies in the past decade has increased the demand for high-quality metal and ceramic powder materials having strict quality requirements such as chemical purity, packing density and flowability [Die+16], [Shu16]. In powder and laser-based manufacturing processes, for example, selective laser sintering (SLS), the powder quality influences the physical properties of the generated product such as tensile strength, brittleness, impact resistance, heat tolerance and resistance to corrosion [Fou19]. Heat treatment using Inductively Coupled Plasma (ICP) is a promising technology that is already being used successfully to improve powder quality significantly. The result of the heat treatment can be seen from the images in Figure 1.1, which were obtained by using a scanning electron microscope (SEM). As the resulting particle shape is spherical, this process is also called *spheroidization*. Spherical powder particles have a higher flowability and lead to higher packing density as well as an improved surface finish of the built parts [Bre19]. Another great advantage of the heat treatment with the inductively coupled plasma against other plasma types is that a high purity can be achieved due to the absence of electrodes. However, the transfer of plasma spheroidization to large-scale industrial production is still a major concern of research. In reaction to the trend in additive manufacturing an experimental ICP torch was set up at the Department of Laser Technologies at Kazan National Research Technical University named after A.N. Tupolev - KAI (KNRTU-KAI). A major challenge here is that many types of metal and ceramic powders need to be investigated. Therefore, the goal is to use numerical simulation as a tool in order to find optimal process parameters and reduce the number of experiments with the in-house built ICP torch.



(a) Raw powder before plasma treatment

(b) Powder after plasma treatment (spheroidization)

Figure 1.1.: Images of tungsten carbide (WC) powder taken by scanning electron microscope at the Department of Laser Technologies (KNRTU-KAI). Used with permission of Konstantin Nagulin from [Tsi18].

1.2. State of the Art Review

1.2.1. Inductively Coupled Plasma (ICP)

In the past thirty years, considerable improvements have been made in the development of inductively coupled plasma torches. Besides their traditional application in spectrochemical analysis, ICPs are currently used in material processing for powder spheroidization and the synthesis of nanopowders [MPY18, p. 2570]. The concept of an ICP under atmospheric pressure conditions was first described as *Electrodeless discharges in high-frequency electromagnetic fields* by Babat [Bab47] in 1947. A continuous operation of the ICP was, however, first demonstrated by Reed in 1961 [Ree61]. Although, the basic concept behind the ICP torch has been known for more than sixty years, it was only used in laboratories for a long time. Tekna Plasma Systems Inc., incorporated in 1990, was the first company, which realized the industrial scale production of spheroidized powder material and nano-powder synthesis by means of ICP technology [Bou04]. Since the completion of an experimental setup in the middle of 2017, research has been carried out at the Department of Laser Technologies at KNRTU-KAI. First successful spheroidization results, as shown in the Figure 1.1b, were presented in 2018 [NNK18]. In the following, the design of the experimental ICP torch (KNRTU-KAI) is analyzed and the process parameters involved in powder spheroidization are identified.

The experimental setup is shown in Figure 1.2 and consists mainly of two parts. The first one is the upper part of the setup, where the actual ICP torch is located, also shown as a technical scheme in the same figure. It comprises a cylindrical enclosure made of quartz, surrounded by a water-cooled induction copper coil, which is supplied by a Radio Frequency (RF) power generator providing a total input power range of 5 kW to 24 kW at an operating frequency of 5.28 MHz. The quartz cylinder used in this experimental setup has a diameter of 9 cm. ICP torches used in material processing are typically larger and consume more power and gas than ICPs used for elemental analysis [Vor+18].

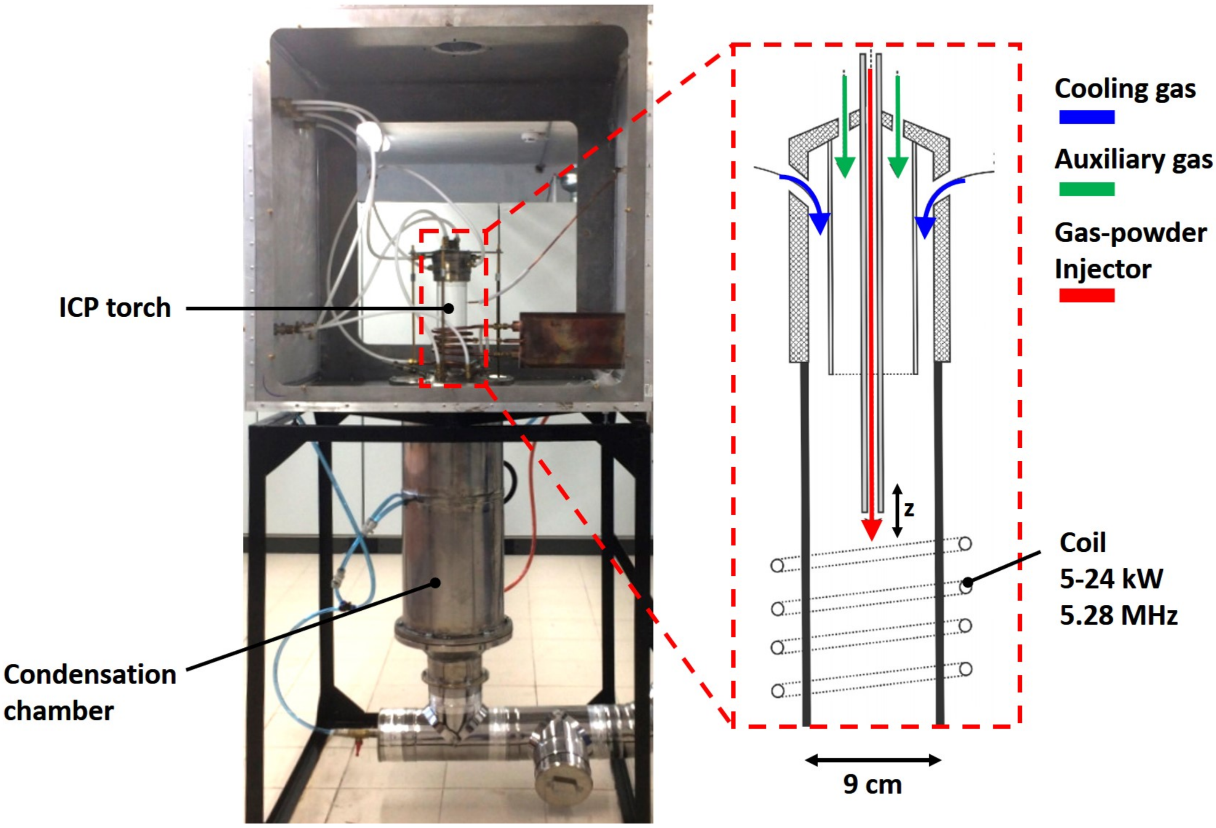


Figure 1.2.: Experimental setup of the Inductively Coupled Plasma at the Department of Laser Technologies of KNRTU-KAI. Used with permission of Ilya Tsivilskiy from [Tsi18].

The plasma gas, for example, argon, is introduced into the quartz cylinder through three concentric tubes and flows from top to bottom. Powder is injected by a carrier gas through the central tube, which is made of stainless steel and has integrated water-cooling, into the plasma zone. The axial position of the central tube can be adjusted in order to reduce the disturbance of vortices on the particle flow. The evolution of vortices caused by Lorentz-forces is described in more detail by Voronov et al. [Vor+18]. The auxiliary gas inlet denotes the working gas, which is producing and sustaining the plasma. The cooling gas, also called sheath gas, gives additional stabilization to the plasma and prevents the destruction of the quartz cylinder enclosure due to the high heat load from the plasma. However, additional cooling is required at power levels above 5 kW to protect the quartz cylinder walls, which can be achieved, for example, by compressed air streaming on the outer surface of the quartz cylinder.

The spheroidization process of the particles starts by feeding the powder into the burning plasma. During this process, the particles are melted in-flight, followed by cooling under controlled conditions in the condensation chamber, which is the lower part of the setup shown in Figure 1.2. For high process efficiency, it is necessary to control the energy input in such a way that the powder is melted but not evaporated [Dzu09, p. 4]. In addition, the spheroidization is influenced by various parameters, which are summarized in the following [Vor+18], [Dzu09, p. 4]:

- torch size,

- plasma power (coil current), current frequency, coil geometry,
- gas flow rates, gas type,
- powder feed rate, particle size,
- material properties of the powder, which determine the energy required to melt a particle of a specific diameter, are: specific heat capacitance, thermal conductivity, melting point, enthalpy of fusion.

Not all of these parameters can be changed due to the present design of the experimental setup. Torch and coil geometry, as well as the frequency of the RF generator, are fixed conditions. An exception here is the height of the central inlet tube, which can be adjusted to change the flight time of the particles. The relevant input and output quantities of the spheroidization process are shown in Figure 1.3a. Information about the plasma temperature is needed to monitor and control the process, whereas the degree of spheroidization is used to evaluate the spheroidization results and is also a measure for the overall process efficiency [Dzu09, p. 65,67]. The degree of spheroidization can be obtained after all powder has been processed by

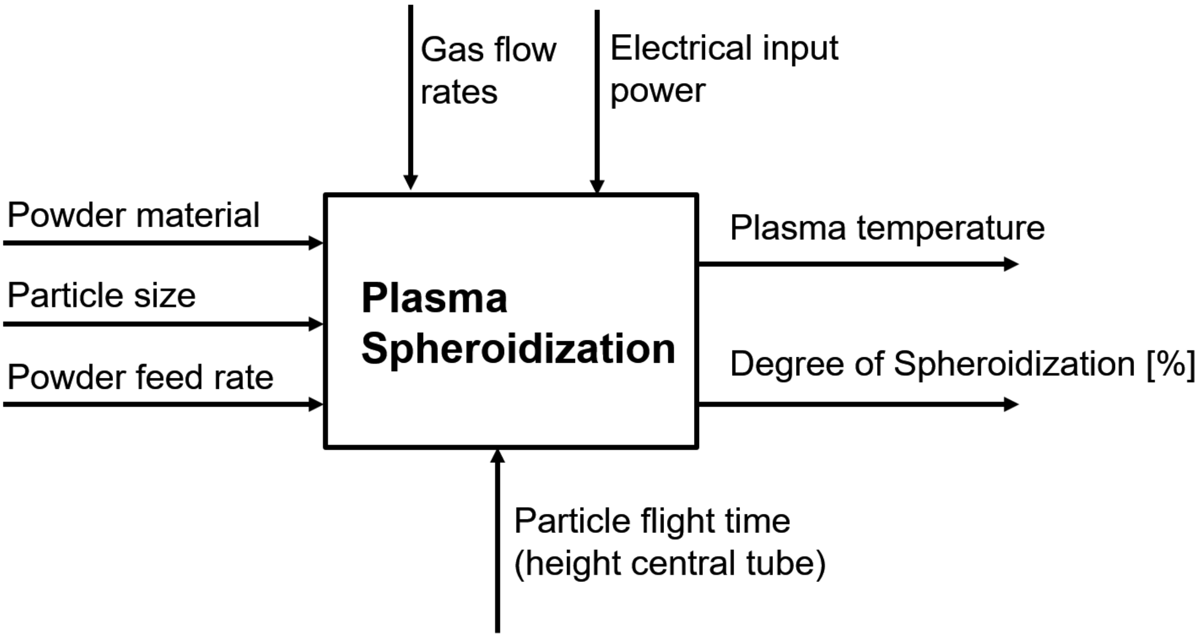
$$\eta_{\mathrm{sph}} = \frac{\text{number of spheroidized particles}}{\text{total number of particles}} \cdot 100\%.$$

Both the plasma temperature as well as the degree of spheroidization are influenced by the input quantities given by the powder characteristics and operating conditions of the ICP torch. Table 1.1 shows the typical range, in which the operating and input parameters of the experimental setup may vary. The electrical input power provided by the RF generator, also known as plate-power, is adjusted by the anode current and anode voltage. The powder feed rates refer to experiments in which a degree of spheroidization of about 95% was achieved with the experimental ICP torch. Note, that the third flow inlet, which was intended for cooling, is currently no longer used, as it was found that it leads to instabilities [Vor+18].

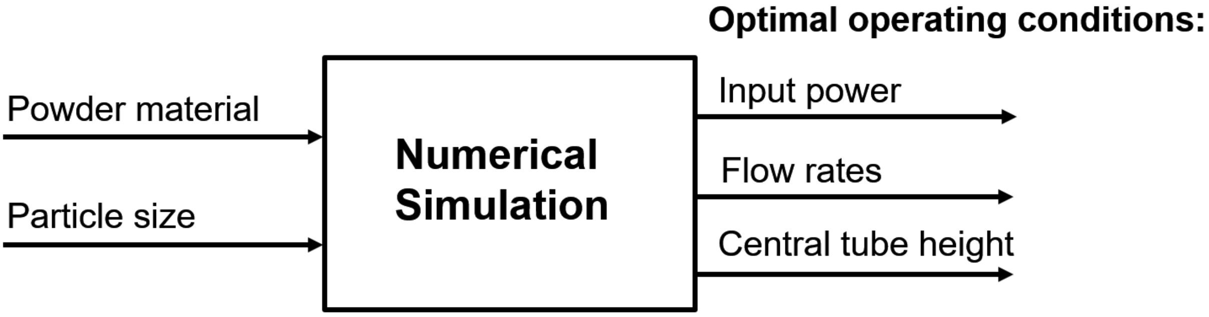
The purpose of numerical simulations is illustrated in Figure 1.3b. The goal is to find optimal ICP operating conditions for a given powder so that the degree of spheroidization is maximized. Numerical investigations are needed in order to reduce the number of experiments to be conducted with the real ICP torch.

Table 1.1.: Process parameters of the experimental setup (KNRTU-KAI)

Parameter		Value	Unit
Electrical input	Anode current (DC)	0 to 5	A
	Anode voltage	3 to 9	kV
	Frequency	5.28	MHz
Gas flow rates	Central inlet	0 to 10	$\mathrm{L\,min^{-1}}$
	Intermediate inlet	40	$\mathrm{L\,min^{-1}}$
	Outer inlet	0	$\mathrm{L\,min^{-1}}$
	Outer cooling stream	240	$\mathrm{L\,min^{-1}}$
Powder characteristics	Materials	Aluminum, Nickel, Titanium, Tungsten,...	
	Particle diameter	20 µm to 80 µm	
	Feed rate	$5\,\mathrm{g\,min^{-1}}$ to $10\,\mathrm{g\,min^{-1}}$	



(a) Plasma spheroidization process parameters



(b) Optimization of the spheroidization process by using numerical simulations

Figure 1.3.: Block diagrams illustrating the input/output parameters of the spheroidization process and a derived numerical model for the purpose of optimization

1.2.2. ICP modeling

Although the use of 3D models has increased in recent years [Ber+05; Guo+19], the faster two-dimensional models still play an important role as large sets of varying parameters need to be studied in order to optimize the spheroidization process and design of ICP torches.

The first implementations of ICP models were realized by Xue et al. [XPB01] by customizing the commercial Fluent code (today: ANSYS Fluent) to treat the electromagnetic field coupling. In particular, new user-defined scalars (UDS) for the real and imaginary part of the vector potential were added and were then solved along with the fluid equations. This approach has the drawback that Fluent requires all equations to be solved in the same computational domain. As a consequence the fluid equations must be also solved outside the torch region, which might result in numerical instabilities or slow convergence of the solution. An improved approach to treat the electromagnetic field was proposed by Bernardi et al. [Ber+03]. They implemented a custom electromagnetic solver based on the finite difference method (FDM) in C programming language. The external user-defined Function (UDF) interface of Fluent was then used to link the custom electromagnetic solver to the built-in Fluent code.

Until today ANSYS Fluent is still one of the most preferred frameworks, whereas the use of the simulation software COMSOL Multiphysics has increased just recently, see [BB15], [Vor+18]. Moreover, COMSOL released a plasma module, which provides interfaces to model different types of plasmas [COM19c]. However, a combination of three COMSOL products is required to fully model the ICP torch: COMSOL Multiphysics, AC/DC Module and Plasma Module [COM19a]. The ICP torch is one example that shows that nowadays there is no one code that can be used for all problems, especially, in the field of plasma physics. It usually always requires some modification of the code based on the assumptions and approximations of the model. This leads to a greater need for code with high customization capabilities.

1.2.3. OpenFOAM

OpenFOAM is a free and open-source CFD Toolbox written in pure C++, which since its release in December 2004 has attracted a large user base across most areas of engineering and science, from both commercial and academic organizations [Hrv16]. The framework already provides various applications to solve fluid dynamics problems, however, it is lacking suitable libraries that can treat electromagnetic dynamics involved in the modeling of induction plasmas. Therefore, a major challenge of this work is to develop a custom electromagnetic solver based on the magnetic vector potential formulation of Maxwell's equations. In general the compilation of custom applications (solvers) is relatively easy in OpenFOAM, as the implementation is based on an equation syntax that corresponds to the mathematical notation.

For example the magnetic vector potential equation

$$\mathbf{\Delta A} = -\mu_0 \mathbf{J}_{\text{coil}} + \mu_0 \sigma \frac{\partial \mathbf{A}}{\partial t}, \tag{1.1}$$

is represented by the code:

```
// update auxiliary variable
DA = muMag*sigma;
// solve the vector potential equation
solve
(
fvm::laplacian(A) == - muMag*Jcoil + fvm::ddt(DA,A)
);
```

Listing 1.1: Implementation of the electromagnetic vector potential equation in OpenFOAM

Since the ICP torch operates under high frequencies, the discretization of the time derivative term in Equation (1.1) causes an excessive computational cost. However, this can be avoided by transforming the vector potential equation into the frequency domain, which in return leads to a strongly coupled system of equations to be solved. As a consequence, all implementations in this work were done in *foam-extend 4.0* [Sou19b], which is a forked version of OpenFOAM and is especially dedicated to the integration of community contributions. Moreover, it provides a very remarkable library (`fvBlockMatrix.H`) that allows to implicitly solve block coupled matrix systems such as the magnetic vector potential matrix and thus significantly improve the numerical stability of the solution. In related studies, a rather sophisticated approach of solving the vector potential equations by means of this block coupled solver library was already presented by Beckstein [BGV17; Bec18]. In the case of the inductively coupled plasma, however, it was found that it is not necessary to adopt his method completely and instead a simplified approach is presented: The magnetic vector potential is solved in this work on a single electromagnetic mesh whereas Beckstein uses a multi-mesh technique in which the magnetic vector potential within the induction region is treated in a separate mesh with boundary values calculated according to Biot-Savart's law. In addition, he solves the electric scalar potential, which is not required according to the applied plasma model assumptions and therefore the electric scalar potential was omitted in the solver presented in this work.

Note, that there is another general difference between the implementations in Fluent, COMSOL, and OpenFOAM: The Fluent-based technique presented by Bernardi et al [Ber+03] solves the fluid equations based on the Finite-Volume-Method (FVM), while the electromagnetic equations are solved by means of the Finite-Difference-Method (FDM), which was implemented and linked to the Fluent code via User-Defined-Functions. The COMSOL code solves fluid as well as electromagnetic equations using only the Finite-Element-Method (FEM). In OpenFOAM all equations are solved by means of FVM.

1.3. Summary: Introduction

1.3.1. Aim of this work

The heat treatment of powder particles using inductively coupled plasma, also known as plasma spheroidization, is a promising technology to meet the requirements of high-quality powders in additive manufacturing. In order to realize production on an industrial scale, this process requires further investigation and optimization. One of these optimization objectives is to maximize the degree of spheroidization for different types of metal and ceramic powders. As the experimental identification of the optimal process parameters is very time-consuming, the numerical simulation shall serve as a tool to reduce the number of experiments. While numerical models of the ICP torch have been investigated in the past using commercial codes, no studies have been found which examined the use of the free and open-source platform OpenFOAM. Often the interfaces of commercial general-purpose CFD tools are not sufficiently flexible to handle complex physics, whereas OpenFOAM provides full access to the source code and thus offers the potential for the development of customized application-specific models.

The aim of this work is to investigate whether the OpenFOAM framework is suited for the implementation of a numerical model for the inductively coupled plasma torch. Therefore, the main objectives are as follows: (1) to develop a solver based on the vector potential formulation of Maxwell's equations that treats the high-frequency electromagnetic fields of the induction coil, (2) to couple the custom electromagnetic solver with a suitable standard flow solver in OpenFOAM, (3) to implement the nonlinear properties of argon gas, which are available in the form of tabulated data, (4) to validate the full model for the inductively coupled plasma torch by comparing the simulation results with literature data.

1.3.2. Limitations of this work

This work is focused on a two-dimensional axisymmetric modeling approach for ICP torches described in by Xue et al. [XPB01] and Bernardi et al. [Ber+03]. The model does not cover the ignition transient of the plasma and is rather designed to describe the burning state of the plasma. The modeling and implementation of powder particles is beyond the scope of this work and is therefore not considered.

1.3.3. Structure of this work

This work is structured as follows: In chapter 2, the governing equations and the implementation of the inductively coupled plasma model are described. In chapter 3, the numerical results obtained with the electromagnetic solver are validated using known analytical solutions. Finally, the simulation results of the full plasma model are presented and compared with the literature results. In chapter 4, the achievements of this work are briefly summarized and recommendations for future work are given.

CHAPTER 2

Modeling

In this chapter, a two-dimensional axisymmetric modeling approach for an Inductively Coupled Plasma (ICP) torch is presented. This model is well known and studied by several authors such as Xue et al [XPB01] and Bernardi et al [Ber+03]. A fundamental assumption of this model is that the plasma is assumed to be in Local Thermodynamic Equilibrium (LTE); in this state it is also referred to as *thermal plasma*. Under LTE, the plasma can be treated as a conductive fluid, which is heated by an electromagnetic field generated by the induction coil. Therefore, the gas flow can be generally described by the Navier-Stokes equations. The numerical computation of the electromagnetic field, however, requires special treatment due to the high operating frequency of the induction coil. To avoid excessive long computation times, a solution method in the frequency domain is employed, which will be discussed in detail in this chapter. Finally, the main features of the custom-developed plasma solver are presented.

2.1. Assumptions

The assumptions for the two-dimensional axisymmetric plasma torch model are summarized in the following, including the general assumptions adopted from [XPB01] and [Ber+03]:

- The plasma is optically thin and in local thermodynamic equilibrium (LTE); assuming LTE state means that locally the electron temperature is equal to the heavy particle temperature, which allows defining only one temperature T;
- The induction plasma torch is operating under constant, atmospheric pressure condition,
- Viscous dissipation in the energy equation is neglected,
- The influence of gravitational force on the gas flow is neglected,

- Quasi-stationary field approximation:$\frac{\partial \mathbf{D}}{\partial t} \ll \mathbf{J} \rightarrow \boldsymbol{\nabla} \times \mathbf{H} = \mathbf{J}$, where $\mathbf{D}$ is the electric displacement field and $\mathbf{J}$ is the current density,
- Quasi-neutral plasma $\rightarrow \boldsymbol{\nabla} \cdot \mathbf{D} = 0$,
- The magnetic permeability is constant throughout all regions (quartz cylinder, argon, coil) and equals the permeability of vacuum $\mu_{\text{mag}} = \mu_0$,
- The coil is modeled by three parallel current ring loops, assuming that the current density is uniformly distributed across the cross-section of the loop. Because we are not interested in the magnetic field and current density distribution inside the coil, the electrical conductivity of the coil is set to zero for this modeling approach,
- The ignition of the plasma, which is realized for example by sparks generated by a Tesla coil, is not considered in the model. Instead, the initial field values are precomputed using the technique described in section 2.4.2 in order to start the simulation,
- The plasma gas is assumed to be pure argon, for which tabulated physical properties are given in the range of the ICP's operating temperature.

2.2. Model Structure

In the following section, the physical principles involved in the inductively coupled plasma phenomenon are briefly explained and presented in a structured overview. A summary of the governing equations can be found at the end of this chapter.

Once ignited, the argon plasma draws power from an induction coil, which is driven by an alternating current. The resulting high-frequency alternating magnetic field of the coil induces eddy currents in the electrically conductive argon gas flowing inside a quartz cylinder. Similar to conductive wires and metals the gas heats up due to Joule heating. Figure 2.1 shows a simplified scheme of the inductively coupled plasma model. Since OpenFOAM is based on the Finite-Volume-Method, the computational domain has to be a geometrical wedge shape in order to simulate two-dimensional axisymmetric cases. The following three regions can be distinguished within the computational domain: inner torch region filled with argon, air region outside the plasma torch, and a coil represented here as a simple block.

The mathematical model describing the electromagnetic field (EM-field), which will be presented in detail in the following section 2.3, uses the following input and output quantities: the coil current density $\mathbf{J}_{\text{coil}}$ in $\mathrm{A\,m^{-2}}$ and the frequency ω in $\mathrm{rad\,s^{-1}}$, which are specified by the geometry and operating conditions of the coil; the Joule heat density q_{J} in $\mathrm{W\,m^{-3}}$ and the Lorentz force density $\mathbf{f}_{\mathrm{L}}$ in $\mathrm{N\,m^{-3}}$, which describe the interaction with the argon fluid. The fluid dynamics are generally described by the Navier-Stokes equations (continuity and momentum equations) and the energy equation. Assuming a laminar compressible gas flow and including the Lorentz force and the Joule heat into the momentum and energy equations, respectively, leads to the following fluid dynamic model:

Continuity Equation:

$$\frac{\partial \rho}{\partial t} + \boldsymbol{\nabla} \cdot (\rho \mathbf{u}) = 0, \tag{2.1}$$

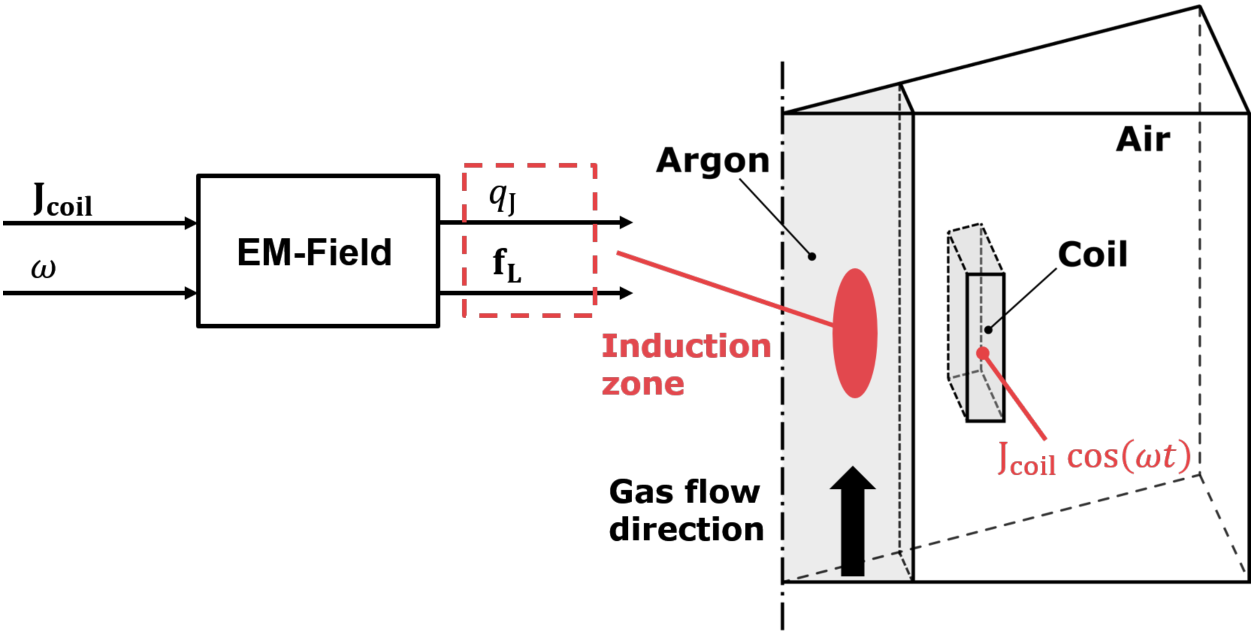


Figure 2.1.: Electromagnetic domain of the plasma model with the coil current density $\mathbf{J}_\mathrm{coil}$ and frequency ω as input quantities, and the output quantities given by the Joule heat q_J and Lorentz force $\mathbf{f}_\mathrm{L}$

Momentum Equation:

$$\frac{\partial}{\partial t}(\rho\mathbf{u}) + \rho(\mathbf{u}\cdot\boldsymbol{\nabla})\mathbf{u} = -\boldsymbol{\nabla}p + \boldsymbol{\nabla}\cdot\boldsymbol{\tau} + \mathbf{f}_\mathrm{L}, \tag{2.2}$$

Energy (Enthalpy) Equation:

$$\frac{\partial}{\partial t}(\rho h) + \boldsymbol{\nabla}\cdot(\rho\mathbf{u}h) - \frac{\partial p}{\partial t} = \boldsymbol{\nabla}\cdot(k\boldsymbol{\nabla}T) - q_\mathrm{R} + q_\mathrm{J}, \tag{2.3}$$

where h is the enthalpy of the plasma; $\mathbf{u}$, p and T are the gas velocity, pressure and temperature, respectively; q_J and $\mathbf{f}_\mathrm{L}$ are the dissipated power density (Joule heat) and the Lorentz force density, respectively. The stress tensor $\boldsymbol{\tau}$ is given by

$$\boldsymbol{\tau} = \left[\mu\Big(\boldsymbol{\nabla}\mathbf{u} + (\boldsymbol{\nabla}\mathbf{u})^\mathrm{T}\Big) - \frac{2}{3}\mu(\boldsymbol{\nabla}\cdot\mathbf{u})\mathbf{I}\right], \tag{2.4}$$

where $\mathbf{I}$ is the identity matrix and ρ, k, μ, c_p are the mass density, thermal conductivity, dynamic viscosity and specific heat capacitance at constant pressure, respectively.

To compute the radiative heat loss q_R, given in $\mathrm{W\,m^{-3}}$, the following radiation law was adopted from [Ber+03]

$$q_\mathrm{R}(T) = \begin{cases} 0, & \text{if} \quad T \leq 9500\,\mathrm{K}, \\ 5600\cdot(T-9500) + 181\cdot(T-9500)^2 & \text{else.} \end{cases} \tag{2.5}$$

The whole system shown in Figure 2.2 illustrates that when solving the fluid dynamic equa-

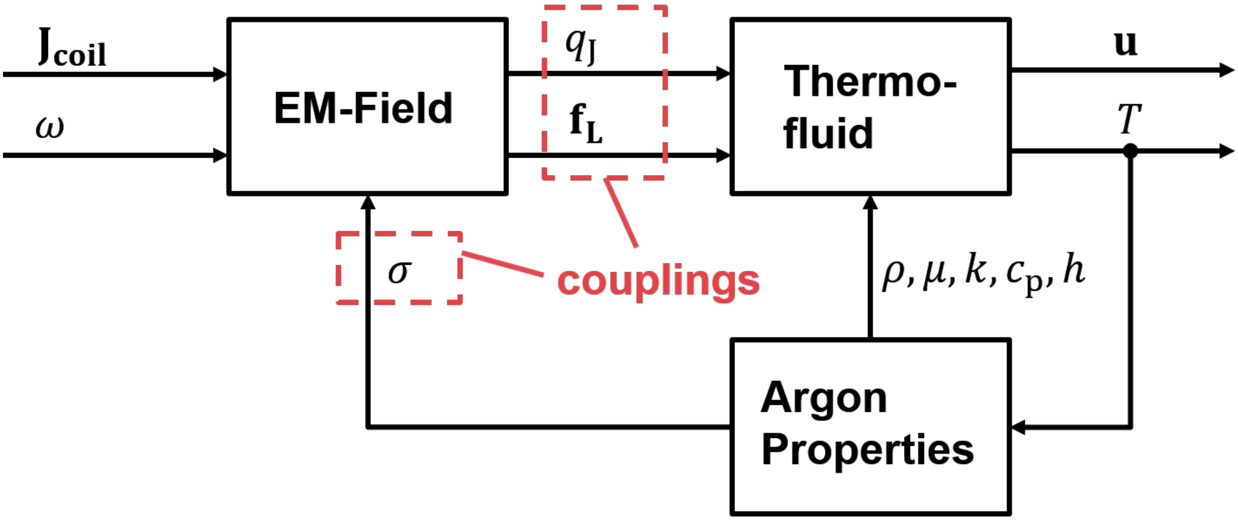


Figure 2.2.: Model structure of the inductively coupled plasma

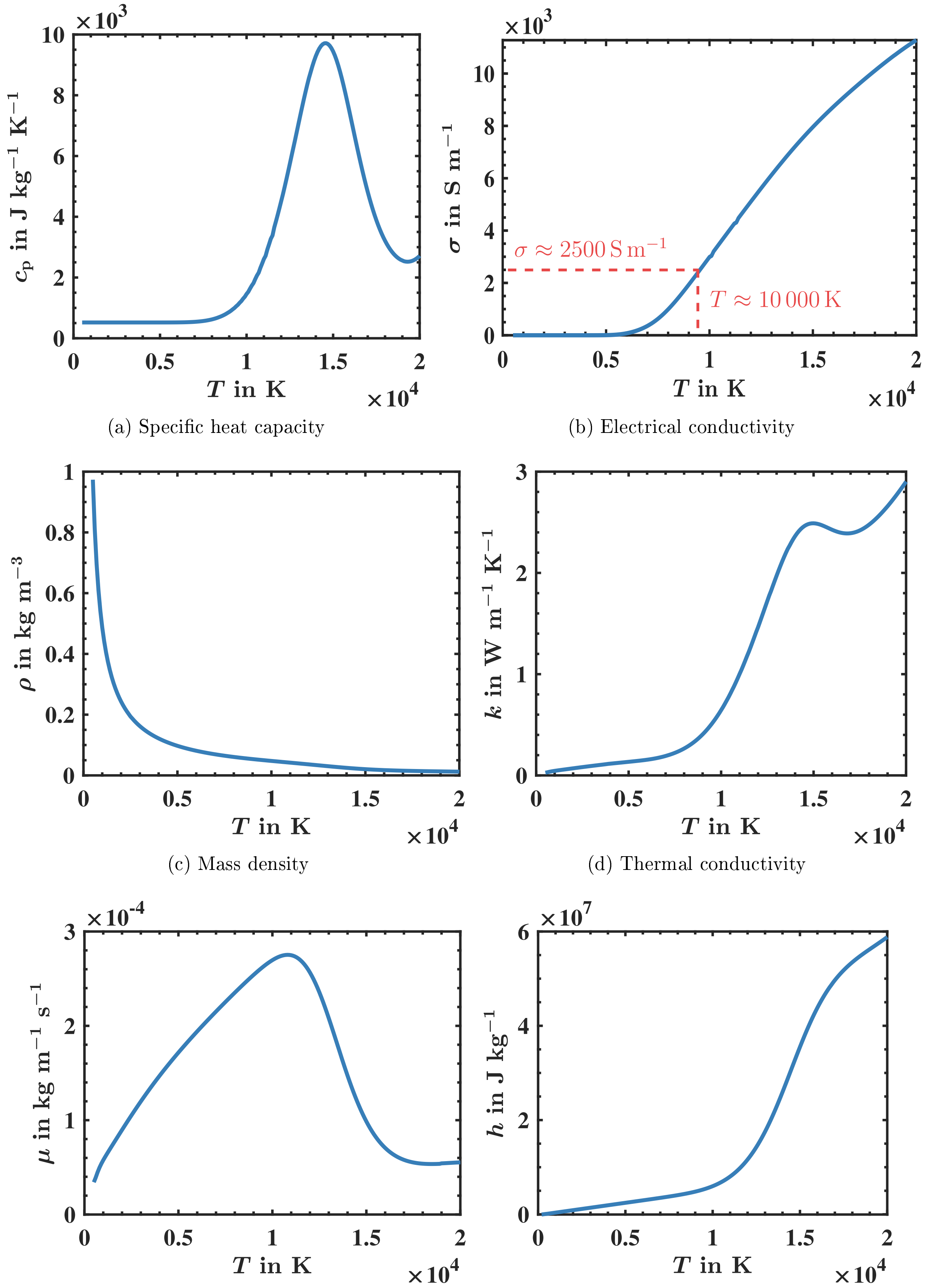
tions in order to obtain the temperature and velocity fields of the plasma, the bidirectional couplings between the electromagnetic and fluid domain must be taken into account, where a feedback loop is created by the electrical conductivity $\sigma(T)$, which is a nonlinear function of the temperature as shown in Figure 2.3b. For low temperatures, the electrical conductivity of argon is close to zero, whereas at plasma operating temperature, which is about 10 000 K, the electrical conductivity has a value of about $2500\,\mathrm{S\,m^{-1}}$.

Additional material properties of argon (mass density $\rho(T)$, dynamic viscosity $\mu(T)$, thermal conductivity $k(T)$, specific heat capacitance at constant pressure $c_{\mathrm{p}}(T)$, the specific enthalpy $h(T)$ in $\mathrm{J\,kg^{-1}}$, electrical conductivity $\sigma(T)$), which are given by nonlinear functions of the temperature T are also shown in Figure 2.3. The background and how tabulated data of argon properties can be obtained by means of kinetic gas theory are described in [Cre+13]. In this work the material data is taken from [Cre+13] and merged with data found in [Cho15a]. Note, that it is not clearly described in the studies of Xue et al. [XPB01] and Bernardi et al. [Ber+03], where the argon data was taken from. However, the data from [Cre+13] and [Cho15a] were also compared against argon data provided by the COMSOL material library [COM19b]. Overall, only minor differences are found between the three data sets.

Mass density approximation. As an alternative to using tabulated data, the mass density ρ can also be approximated by the ideal-gas law

$$\rho = \frac{p}{RT}, \tag{2.6}$$

where p and R denote the pressure and the specific gas constant, respectively. As noted in Figure 2.4 the error $\Delta\rho$ between the tabulated density $\rho_{\text{tabulated}}$ and the density ρ approximated by the ideal-gas law is less than 2% for the temperature range up to 10 000 K. To simplify the implementation, the ideal-gas Equation (2.6) was used in this work to calculate the mass density.



(a) Specific heat capacity

(b) Electrical conductivity

(c) Mass density

(d) Thermal conductivity

(e) Dynamic viscosity

(f) Enthalpy

Figure 2.3.: Argon Properties: Enthalpy taken from [Cho15a] and others from [Cre+13]

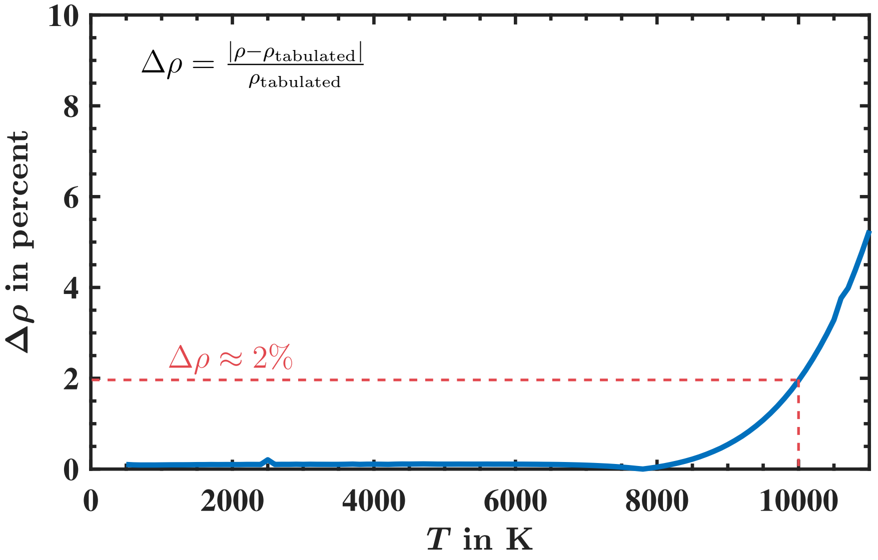


Figure 2.4.: Error of using the ideal-gas law approximation for the mass density

2.3. Electromagnetic Solver

In this section, the Maxwell equations, which are describing the electromagnetic field generated by the coil, are formulated by means of the magnetic vector potential. This has the advantage that instead of solving the coupled Maxwell equations, only a single partial differential equation needs to be solved. Furthermore, the frequency domain method, which is commonly used in ICP modeling [XPB01] and other magnetohydrodynamic problems [BGV17] to efficiently compute high frequency electromagnetic fields, will be presented in detail.

2.3.1. Magnetic vector potential formulation

With the general assumptions mentioned in the previous section 2.1 the governing Maxwell equations are formulated as

$$\nabla \cdot \mathbf{D} = 0, \tag{2.7}$$

$$\nabla \cdot \mathbf{B} = 0, \tag{2.8}$$

$$\nabla \times \mathbf{E} = -\frac{\partial \mathbf{B}}{\partial t}, \tag{2.9}$$

$$\nabla \times \mathbf{H} = \mathbf{J}, \tag{2.10}$$

where $\mathbf{D}$ and $\mathbf{E}$ are the electric displacement field and electric field, respectively; $\mathbf{J}$ is the current density; $\mathbf{B}$ and $\mathbf{H}$ are the magnetic flux density and magnetic field strength, respectively. This form of the Maxwell equations implies quasi-static field approximation (neglection of the displacement field time-derivative in Equation (2.10)), and quasi-neutrality of the plasma (neglection of volumetric charge in Equation (2.7)). Generally, in magnetohydrodynamics an additional term $\mathbf{u} \times \mathbf{B}$ is considered in the Maxwell equations, which describes the influence of the conductive fluid flow on the imposed magnetic field. However, it can be shown that the term $\mathbf{u} \times \mathbf{B}$ can be neglected for small magnetic Reynolds numbers [Dav01, p. 118].

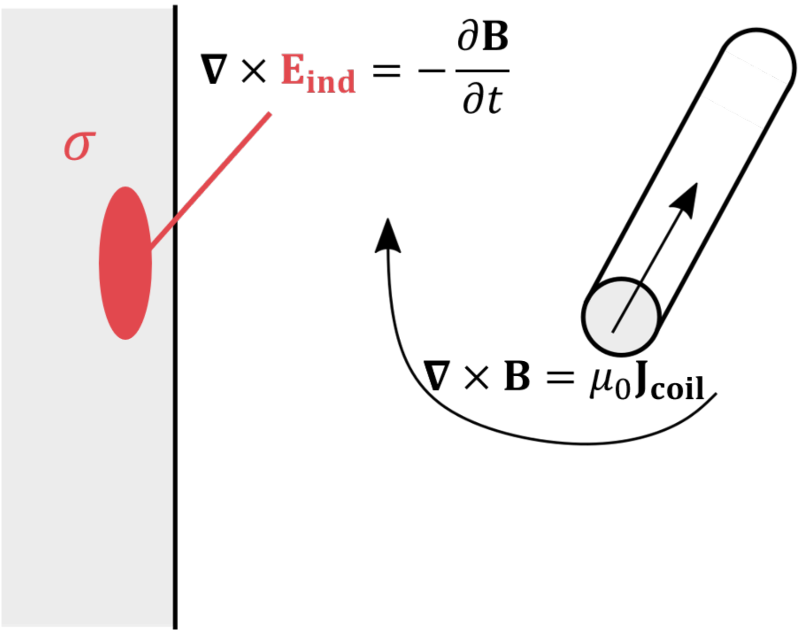

$\nabla \times \mathbf{E}_{\mathrm{ind}} = -\frac{\partial \mathbf{B}}{\partial t}$
σ
$\nabla \times \mathbf{B} = \mu_0 \mathbf{J}_{\mathrm{coil}}$

Figure 2.5.: Application of the Maxwell equations to describe basic electromagnetic phenomena

The magnetic Reynolds number is given by

$$Re_{\mathrm{m}} = \mu_0 \sigma u L, \tag{2.11}$$

with u and L being the flow velocity and the length scale, respectively; σ and μ_0 denote the electrical conductivity of the fluid and the magnetic permeability in vacuum, respectively. Assuming plasma torch conditions with, for example, $u_{\mathrm{max}} \sim 10\,\mathrm{m\,s^{-1}}$, $L \sim 0.05\,\mathrm{m}$, $\mu_0 \sim 1.26 \times 10^{-6}\,\mathrm{V\,s\,A^{-1}\,m^{-1}}$, $\sigma \sim 2500\,\mathrm{S\,m^{-1}}$ the magnetic Reynolds number yields $Re_{\mathrm{m}} \approx 0.0016 \ll 1$. Thus, the disturbance of the magnetic field by the conductive fluid flow is negligible.

The simple scheme in Figure 2.5 illustrates how the Maxwell equations are applied to describe the electromagnetic field of an inductively coupled plasma. Ampere's law (2.10), which is rewritten here using the material law $\mathbf{B} = \mu_0 \mathbf{H}$ in the form

$$\nabla \times \mathbf{B} = \mu_0 \mathbf{J}, \tag{2.12}$$

describes the magnetic field $\mathbf{B}$ surrounding a current-carrying piece of wire, whereas the induction effects due to time-varying magnetic fields in a conductive region σ are described by Faraday's law (2.9). The goal of this problem is to calculate the $\mathbf{B}$ and $\mathbf{E}$ field for a given coil current $\mathbf{J}_{\mathrm{coil}}$. In order to decouple and simplify the solution of the Maxwell equations, the magnetic vector potential $\mathbf{A}$ is introduced, which is defined by the Helmholtz theorem as

$$\mathbf{B} = \nabla \times \mathbf{A}. \tag{2.13}$$

By inserting Equation (2.13) into (2.12) the following expression is obtained

$$\boldsymbol{\nabla} \times (\boldsymbol{\nabla} \times \mathbf{A}) = \mu_0 \mathbf{J}, \tag{2.14}$$

which can be rewritten using the mathematical rule $\boldsymbol{\nabla} \times (\boldsymbol{\nabla} \times \mathbf{A}) = \boldsymbol{\nabla}(\boldsymbol{\nabla} \cdot \mathbf{A}) - \boldsymbol{\Delta}\mathbf{A}$ as

$$\boldsymbol{\nabla}(\boldsymbol{\nabla} \cdot \mathbf{A}) - \boldsymbol{\Delta}\mathbf{A} = \mu_0 \mathbf{J}. \tag{2.15}$$

In general, the vector potential is not uniquely defined and an arbitrary gradient term $\boldsymbol{\nabla}\psi$ can be added to the vector potential without changing the solution of the magnetic field. In other words, it means that there is a degree of freedom when choosing $\mathbf{A}$, which is also known as the gauge invariance. A common choice of gauge is the so-called Coulomb gauge $\boldsymbol{\nabla} \cdot \mathbf{A} = 0$, which leads when applied to Equation (2.15) to

$$\boldsymbol{\Delta}\mathbf{A} = -\mu_0 \mathbf{J}. \tag{2.16}$$

Further, we can distinguish for the current density

$$\mathbf{J} = \mathbf{J}_{\text{coil}} + \mathbf{J}_{\text{ind}} \tag{2.17}$$

between a source term $\mathbf{J}_{\text{coil}}$ and an induced current $\mathbf{J}_{\text{ind}}$. In the present model the coil current density $\mathbf{J}_{\text{coil}}$ has a constant value, while the induced current $\mathbf{J}_{\text{ind}}$ is given by Ohm's law

$$\mathbf{J}_{\text{ind}} = \sigma \mathbf{E}_{\text{ind}}, \tag{2.18}$$

where $\mathbf{E}_{\text{ind}}$ is the induced electric field, which can be expressed as a function of the vector potential by substituting the magnetic flux density (2.13) in Faraday's law (2.9)

$$\begin{aligned} \boldsymbol{\nabla} \times \mathbf{E}_{\text{ind}} &= -\frac{\partial \mathbf{B}}{\partial t} \\ &= -\frac{\partial(\boldsymbol{\nabla} \times \mathbf{A})}{\partial t} \\ &= \boldsymbol{\nabla} \times -\frac{\partial \mathbf{A}}{\partial t}. \end{aligned} \tag{2.19}$$

Comparing the arguments of the curl operator on both sides of Equation (2.19) yields

$$\mathbf{E}_{\text{ind}} = -\frac{\partial \mathbf{A}}{\partial t}, \tag{2.20}$$

which substituted into Equation (2.18) leads to

$$\mathbf{J}_{\text{ind}} = -\sigma \frac{\partial \mathbf{A}}{\partial t}. \tag{2.21}$$

Recursive substitution of equations (2.17) and (2.21) into Equation (2.16) finally leads to the governing partial differential equation of the magnetic vector potential

$$\boldsymbol{\Delta}\mathbf{A} = -\mu_0 \mathbf{J}_{\text{coil}} + \mu_0 \sigma \frac{\partial \mathbf{A}}{\partial t}. \tag{2.22}$$

2.3.2. Frequency domain method

Given that the induction coil is driven by a sinusoidal current with high operating frequency $\mathbf{J}_{\text{coil}} = \widehat{\mathbf{J}}_{\text{coil}} \cos(\omega t)$, solving the vector potential Equation (2.22) numerically will lead to an unreasonably high computational effort. For example, resolving a harmonic function by 50 samples at a frequency of 5 MHz requires a simulation time step of $1/(50 \cdot f) = 4\,\text{ns}$. However, this can be avoided by using the frequency domain method. Also, it allows using higher simulation time steps, since the timescale of the fluid dynamics is much higher (milliseconds) than the timescale of the electromagnetic field (nanoseconds).

Usually, the frequency domain method is based on applying the Fourier transform to the vector potential equation (2.22), which yields

$$\mathbf{\Delta}\widetilde{\mathbf{A}} = -\mu_0 \widetilde{\mathbf{J}}_{\text{coil}} + j\omega\mu_0\sigma\widetilde{\mathbf{A}} \tag{2.23}$$

where $\widetilde{\mathbf{A}}$ and $\widetilde{\mathbf{J}}_{\text{coil}}$ are the complex amplitude of the magnetic vector potential and coil current density, respectively; j and ω are the imaginary unit and the harmonic frequency, respectively. However, the computation of complex numbers cannot be handled by OpenFOAM, as it is a rather rare feature in CFD software. Therefore, the following mathematical trick is applied to bring the Equation (2.23) into an implementable form: By using the identity[1]

$$\widetilde{\mathbf{A}} = \widetilde{\mathbf{A}}_{\text{R}} + j\widetilde{\mathbf{A}}_{\text{I}} \tag{2.24}$$

to rewrite Equation (2.23) as

$$\begin{aligned} \mathbf{\Delta}(\widetilde{\mathbf{A}}_{\text{R}} + j\widetilde{\mathbf{A}}_{\text{I}}) &= -\mu_0\widetilde{\mathbf{J}}_{\text{coil}} + j\omega\mu_0\sigma(\widetilde{\mathbf{A}}_{\text{R}} + j\widetilde{\mathbf{A}}_{\text{I}}) \\ \mathbf{\Delta}\widetilde{\mathbf{A}}_{\text{R}} + j\mathbf{\Delta}\widetilde{\mathbf{A}}_{\text{I}} &= -\mu_0\widetilde{\mathbf{J}}_{\text{coil}} + j\omega\mu_0\sigma\widetilde{\mathbf{A}}_{\text{R}} - \omega\mu_0\sigma\widetilde{\mathbf{A}}_{\text{I}}, \end{aligned} \tag{2.25}$$

and by separating the real and imaginary parts of Equation (2.25) the following two equations are obtained:

$$\begin{aligned} \text{Re:}\quad \mathbf{\Delta}\widetilde{\mathbf{A}}_{\text{R}} &= -\mu_0\widetilde{\mathbf{J}}_{\text{coil}} - \omega\mu_0\sigma\widetilde{\mathbf{A}}_{\text{I}}, \\ \text{Im:}\quad \mathbf{\Delta}\widetilde{\mathbf{A}}_{\text{I}} &= \omega\mu_0\sigma\widetilde{\mathbf{A}}_{\text{R}}, \end{aligned} \tag{2.26}$$

where $\widetilde{\mathbf{A}}_{\text{R}}$ and $\widetilde{\mathbf{A}}_{\text{I}}$ are the real and imaginary parts of the complex vector potential, respectively. Equations (2.26) can be also rewritten in matrix form as

$$\begin{bmatrix} \frac{1}{\mu_0}\mathbf{\Delta} & \sigma\omega \\ -\sigma\omega & \frac{1}{\mu_0}\mathbf{\Delta} \end{bmatrix} \begin{bmatrix} \widetilde{\mathbf{A}}_{\text{R}} \\ \widetilde{\mathbf{A}}_{\text{I}} \end{bmatrix} = \begin{bmatrix} -\widetilde{\mathbf{J}}_{\text{coil}} \\ 0 \end{bmatrix}. \tag{2.27}$$

Here, the off-diagonal term $\sigma\omega$ indicates the coupling between the equations of the real and imaginary parts. As the plasma torch operates at high frequencies, strong coupling of the system (2.27) is expected. The electromagnetic solver, *emFoam*, developed in this work utilizes a block coupled matrix solver to ensure the stability and convergence of the

[1] Since a source current of the form $\mathbf{J}_{\text{coil}} = \widehat{\mathbf{J}}_{\text{coil}} \cos(\omega t)$ is given, the complex current has only a real part $\widetilde{\mathbf{J}}_{\text{coil}} = \text{Re}\left\{\widetilde{\mathbf{J}}_{\text{coil}}\right\}$

numerical solution of the vector potential system (2.27). Details of the implementation can be found in the appendix D.1.

Magnetic field solution. After the solution of the real and imaginary parts of the vector potential is calculated, the magnetic field $\mathbf{B}$ can be obtained by applying Equation (2.13) as

$$\begin{aligned} \widetilde{\mathbf{B}}_{\mathrm{R}} &= \boldsymbol{\nabla} \times \widetilde{\mathbf{A}}_{\mathrm{R}}, \\ \widetilde{\mathbf{B}}_{\mathrm{I}} &= \boldsymbol{\nabla} \times \widetilde{\mathbf{A}}_{\mathrm{I}}, \end{aligned} \tag{2.28}$$

and using the Euclidean norm for complex numbers to yield

$$B_{\mathrm{mag}} := \left|\widetilde{\mathbf{B}}\right| = \sqrt{\left|\widetilde{\mathbf{B}}_{\mathrm{R}}\right|^2 + \left|\widetilde{\mathbf{B}}_{\mathrm{I}}\right|^2}, \tag{2.29}$$

where B_{mag} is the magnitude of the flux density $\mathbf{B}$. Subsequently, the magnitude of the magnetic field strength H_{mag} can be obtained by

$$H_{\mathrm{mag}} = \frac{B_{\mathrm{mag}}}{\mu_0}. \tag{2.30}$$

Joule heat density. The dissipated plasma power per unit volume, in the following referred to as Joule heat density, describes the heat, which is produced inside the argon plasma due to eddy currents. In general, the Joule heat density is defined as

$$q_{\mathrm{J}} = \mathbf{E} \cdot \mathbf{J}. \tag{2.31}$$

However, when time-varying fields $\mathbf{J}$ and $\mathbf{E}$ are considered the dissipated power is not constant but a function of time (harmonic oscillation). The electrical power, which is dissipated into the plasma within a period, is given by the time-averaged Joule heat density, which can be obtained by means of the frequency domain method by

$$\overline{q}_{\mathrm{J}} = \frac{1}{2} \mathrm{Re}\left\{\widetilde{\mathbf{E}} \cdot \widetilde{\mathbf{J}}^*\right\}, \tag{2.32}$$

where $\widetilde{\mathbf{E}}$ and $\widetilde{\mathbf{J}}^*$ are the complex electric field and the complex conjugated current density, respectively.

Using Ohm's law (2.18) this can be expressed as

$$\overline{q}_{\mathrm{J}} = \frac{1}{2\sigma} \mathrm{Re}\left\{\widetilde{\mathbf{J}} \cdot \widetilde{\mathbf{J}}^*\right\}. \tag{2.33}$$

The dot product can be further rewritten as

$$\begin{aligned} \widetilde{\mathbf{J}} \cdot \widetilde{\mathbf{J}}^* &= \left(\widetilde{\mathbf{J}}_{\mathrm{R}} + j\widetilde{\mathbf{J}}_{\mathrm{I}}\right) \cdot \left(\widetilde{\mathbf{J}}_{\mathrm{R}} - j\widetilde{\mathbf{J}}_{\mathrm{I}}\right) \\ &= \widetilde{\mathbf{J}}_{\mathrm{R}}{}^2 + \widetilde{\mathbf{J}}_{\mathrm{I}}{}^2 = \left|\widetilde{\mathbf{J}}_{\mathrm{ind}}\right|^2 \end{aligned} \tag{2.34}$$

where $\widetilde{\mathbf{J}}_{\mathrm{ind}}$ is the magnitude of the induced current. With the expression (2.34) the Joule heat density is now written as

$$\overline{q}_{\mathrm{J}} = \frac{1}{2\sigma}\left|\widetilde{\mathbf{J}}_{\mathrm{ind}}\right|^2 = \frac{1}{2\sigma}\left(\left|\widetilde{\mathbf{J}}_{\mathrm{R}}\right|^2 + \left|\widetilde{\mathbf{J}}_{\mathrm{I}}\right|^2\right). \tag{2.35}$$

Further, the complex induced current density is given by the Fourier transform of equation (2.21)

$$\widetilde{\mathbf{J}}_{\mathrm{ind}} = -\sigma j\omega\widetilde{\mathbf{A}} \tag{2.36}$$

and the real and imaginary parts of the induced current density are obtained by substituting the vector potential by (2.24)

$$\begin{aligned} \widetilde{\mathbf{J}}_{\mathrm{R}} &= \sigma\omega\widetilde{\mathbf{A}}_{\mathrm{I}}, \\ \widetilde{\mathbf{J}}_{\mathrm{I}} &= -\sigma\omega\widetilde{\mathbf{A}}_{\mathrm{R}}. \end{aligned} \tag{2.37}$$

Finally, by substituting equations (2.37) into (2.35) the time-averaged Joule heat density as a function of the vector potential is obtained

$$\overline{q}_{\mathrm{J}} = \frac{\sigma\omega^2}{2}\left(\left|\widetilde{\mathbf{A}}_{\mathrm{R}}\right|^2 + \left|\widetilde{\mathbf{A}}_{\mathrm{I}}\right|^2\right). \tag{2.38}$$

Lorentz force density. The general definition of the Lorentz force is given by

$$\mathbf{f}_{\mathrm{L}} = \mathbf{J} \times \mathbf{B}. \tag{2.39}$$

In order to obtain an expression based on the vector potential similar to the Joule heat, the time-averaged Lorentz force in the frequency domain is needed here, which is given by

$$\overline{\mathbf{f}}_{\mathrm{L}} = \frac{1}{2}\,\mathrm{Re}\left\{\widetilde{\mathbf{J}} \times \widetilde{\mathbf{B}}^*\right\}, \tag{2.40}$$

where $\widetilde{\mathbf{B}}^*$ is the complex conjugate of the magnetic field $\widetilde{\mathbf{B}}$. The Equation (2.40) can be further rewritten as

$$\begin{aligned} \overline{\mathbf{f}}_{\mathrm{L}} &= \frac{1}{2}\,\mathrm{Re}\left\{\widetilde{\mathbf{J}} \times \widetilde{\mathbf{B}}^*\right\} \\ &= \frac{1}{2}\,\mathrm{Re}\left\{\left(\widetilde{\mathbf{J}}_{\mathrm{R}} + j\widetilde{\mathbf{J}}_{\mathrm{I}}\right) \times \left(\widetilde{\mathbf{B}}_{\mathrm{R}} - j\widetilde{\mathbf{B}}_{\mathrm{I}}\right)\right\} \\ &= \frac{1}{2}\,\mathrm{Re}\left\{\left[\left(\widetilde{\mathbf{J}}_{\mathrm{R}} \times \widetilde{\mathbf{B}}_{\mathrm{R}}\right) + \left(\widetilde{\mathbf{J}}_{\mathrm{I}} \times \widetilde{\mathbf{B}}_{\mathrm{I}}\right)\right] + j\left[\left(\widetilde{\mathbf{J}}_{\mathrm{I}} \times \widetilde{\mathbf{B}}_{\mathrm{R}}\right) - \left(\widetilde{\mathbf{J}}_{\mathrm{R}} \times \widetilde{\mathbf{B}}_{\mathrm{I}}\right)\right]\right\} \\ &= \frac{1}{2}\left[\left(\widetilde{\mathbf{J}}_{\mathrm{R}} \times \widetilde{\mathbf{B}}_{\mathrm{R}}\right) + \left(\widetilde{\mathbf{J}}_{\mathrm{I}} \times \widetilde{\mathbf{B}}_{\mathrm{I}}\right)\right]. \end{aligned} \tag{2.41}$$

Finally, substitution of equations (2.28) and (2.37) into (2.41) leads to the following form of the Lorentz force

$$\overline{\mathbf{f}}_{\mathrm{L}} = \frac{\sigma\omega}{2}\left[\widetilde{\mathbf{A}}_{\mathrm{I}} \times \left(\boldsymbol{\nabla} \times \widetilde{\mathbf{A}}_{\mathrm{R}}\right) - \widetilde{\mathbf{A}}_{\mathrm{R}} \times \left(\boldsymbol{\nabla} \times \widetilde{\mathbf{A}}_{\mathrm{I}}\right)\right]. \tag{2.42}$$

2.4. Implementation

In order to perform a numerical simulation of the plasma model, the partial differential equations and the computational domain are discretized in OpenFOAM by means of the Finite-Volume-Method (FVM). The resulting systems of algebraic equations are then typically solved iteratively by linear methods. A comprehensive description of the finite volume method as well as the available discretization schemes and linear solvers in OpenFOAM can be found in [Wol19]. The following section focuses on the main features of the electromagnetic and plasma solvers developed in the course of this work.

2.4.1. Algorithm description

While the electromagnetic part of the solver was completely developed from scratch, the fluid part is based on the *rhoPimpleFoam* code, which is a *transient solver for laminar or turbulent flow of compressible fluids* provided by default in OpenFOAM. It utilizes the PIMPLE algorithm [CFD18a], which is a combination of the Pressure Implicit with Splitting of Operator (PISO) algorithm and the Semi-Implicit Method for Pressure-Linked Equations (SIMPLE). These solution techniques are widely used in CFD codes and allow to solve the coupled Navier-Stokes Equations in a segregated manner. The custom developed electromagnetic solver and the PIMPLE-based fluid solver are combined inside the main *time.loop()* of the plasma solver, which is listed in Algorithm 1.

Algorithm 1: Plasma solver

```
while time.loop() do
    look up electrical conductivity of argon $\sigma(T)$;
    solve vector potential equations (2.27) for $\widetilde{\mathbf{A}}_\mathrm{R}$ and $\widetilde{\mathbf{A}}_\mathrm{I}$; retrieve $\overline{q}_\mathrm{J}$ and $\overline{\mathbf{f}}_\mathrm{L}$;
    map $\overline{q}_\mathrm{J}$ and $\overline{\mathbf{f}}_\mathrm{L}$ fields to the fluid mesh;
    solve continuity equation (2.1); //applies only to plasmaTransientFoam;
    while pimple.loop() do
        solve momentum equation (2.2);
        solve energy equation (2.3);
        look up argon properties $\rho(T)$, $k(T)$, $\mu(T)$, $c_\mathrm{p}(T)$;
        solve equation of state (D.6) to retrieve new temperature $T$;
        compute radiation loss $q_\mathrm{R}$, equation (2.5);
        while piso.loop() do
            solve pressure correction equation;
```

Note, that a multi-mesh technique has been employed in order to achieve that the fluid equations are only solved in the inner plasma torch area, whereas the computational domain for the electromagnetic equations extends far outside the torch. The interaction between the domains, illustrated in Figure 2.6, is implemented in the following manner: At each iteration, the electrical conductivity $\sigma(T)$ is calculated based on the current temperature values and stored in the respective cells of the overlapping electromagnetic mesh. The magnetic vector potential is calculated for updating the values of the Lorentz force and of the Joule heat, which are then mapped to the fluid mesh, where the Navier-Stokes and energy equations are solved. In order to map and interpolate field values of the Lorentz

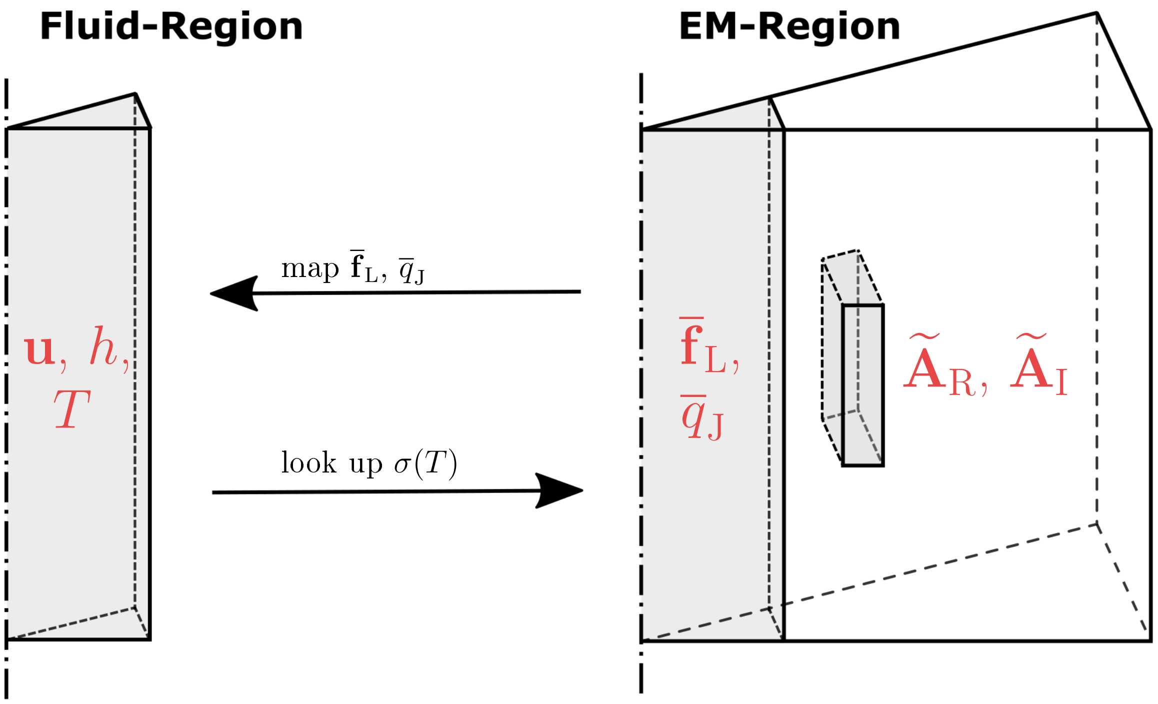

Fluid-Region
EM-Region
map $\overline{\mathbf{f}}_{\mathrm{L}}$, $\overline{q}_{\mathrm{J}}$
look up $\sigma(T)$
$\mathbf{u}$, h, T
$\overline{\mathbf{f}}_{\mathrm{L}}$, $\overline{q}_{\mathrm{J}}$
$\widetilde{\mathbf{A}}_{\mathrm{R}}$, $\widetilde{\mathbf{A}}_{\mathrm{I}}$

Figure 2.6.: Multi-mesh technique: fluid mesh (left), electromagnetic mesh (right)

force and Joule heat from the electromagnetic mesh to the fluid mesh, the `meshToMesh.H` library can be used in OpenFOAM. However, it was found that errors occur when using the `meshToMesh.H` library to map and interpolate field values from a smaller to a larger mesh, as it is needed in the case for the electrical conductivity $\sigma(T)$. To solve this problem the following workaround was found: By using a fluid mesh that is identical with the overlapping region of the electromagnetic mesh, a simple look-up routine for the electrical conductivity $\sigma(T)$ can be implemented, which also avoids the interpolation of field values. See also Appendix D.3 for details about the implementation of the two-mesh technique.

In order to resolve the non-linearity and pressure-velocity coupling of the Navier-Stokes equations, a segregated solution method is employed by the standard *rhoPimpleFoam* code, which performs several iterations to solve the momentum and pressure correction equation within the *pimple.loop()*. Usually, the energy equation is solved within the *pimple.loop()* as well to obtain the enthalpy, which is then used to calculate the temperature by solving the equation of state

$$\Delta h = \int c_{\mathrm{p}}(T)\, dT. \tag{2.43}$$

Details on how the Equation (D.6) is treated by the standard thermodynamic library in OpenFOAM, can be found in the appendix D.2.

Furthermore, a number of major modifications have been made to the default thermodynamic library, in order to model the nonlinear properties of partly ionized argon. Therefore the code available at [Cho15b; Cho15a] was adapted to implement a custom thermodynamic library with look-up table functions for the argon properties using a piece-wise linear interpolation between tabulated data points. Finally, the calculation of radiation loss is included inside the *pimple.loop()* as well, as it is a nonlinear function of the temperature and it is coupled to the energy equation.

The plasma solver can run in steady-state and transient mode. The difference between the

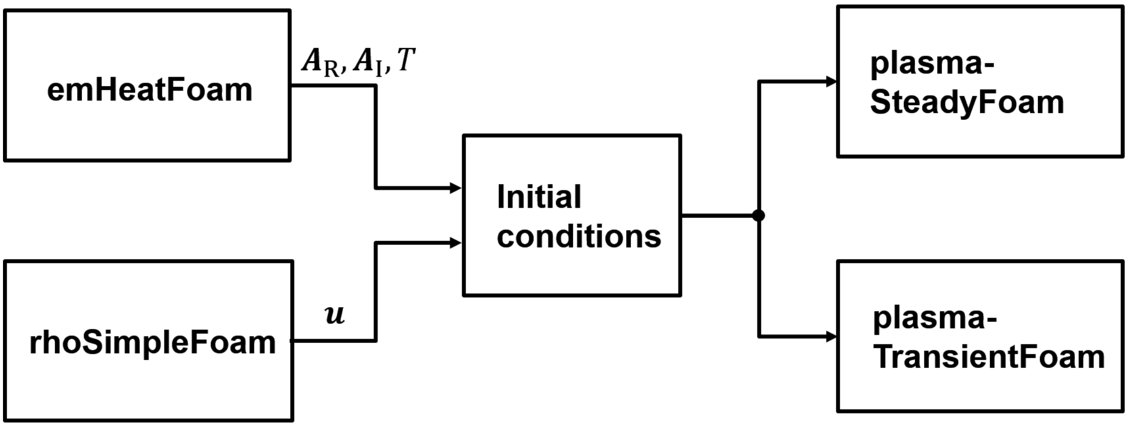


Figure 2.7.: Initialization of the plasma solver

two variants is that in the transient version of the plasma solver, Algorithm 1, the continuity equation is solved, whereas in the steady-state version this part of the code is omitted. Note, that in the case of the steady-state solver, the *time.loop()* counts the number of iterations of the whole system being solved.

2.4.2. Plasma initialization technique

The ignition of the plasma is not covered by the model, since, we are mainly interested in the heated plasma flow under the steady-state operating conditions. As a result, the following method, as illustrated in Figure 2.7, is used to initialize the plasma solver: The initial velocity field $\mathbf{u}$ is taken from the steady-state solution obtained by the standard *rhoSimpleFoam* solver, whereas the custom-developed *emHeatFoam* solver has been especially compiled for the purpose of computing the initial values of the temperature and magnetic vector potential fields. The fluid flow is not considered by the *emHeatFoam* solver. Instead, it is based on the *emFoam* code, which is extended to treat induction-based heating processes by solving the temperature equation

$$\frac{\partial}{\partial t}(\rho c_{\mathrm{p}} T) = \boldsymbol{\nabla} \cdot (k \boldsymbol{\nabla} T) - q_{\mathrm{R}} + \overline{q}_{\mathrm{J}}, \tag{2.44}$$

where the radiation loss law and tabulated data for argon properties are adopted from the previous sections. Note, that the *emHeatFoam* solver is not an actual steady-state solver but rather a pseudo-transient solver, which means that the Euler time-derivative scheme is used and the under-relaxation method [CFD18a] is applied to stabilize the solution of the temperature equation. As with this solver the temperature continuously increases, the solver is stopped after the desired temperature distribution is reached. In order to speed up the heating process, the electrical conductivity is held to a value of $\sigma = 1\,\mathrm{S\,m^{-1}}$ during the simulation until the temperature reaches 5000 K. Above this temperature a look-up table function is applied to model the nonlinear electrical conductivity as shown in Figure 2.3b. In addition, a zero gradient condition $\frac{\partial T}{\partial \mathbf{n}} = 0$ is used for the temperature at all boundaries of the computational domain.

2.5. Summary

2.5.1. Governing equations

Electromagnetic domain

Magnetic vector potential:

$$\begin{bmatrix} \frac{1}{\mu_0}\boldsymbol{\Delta} & \sigma\omega \\ -\sigma\omega & \frac{1}{\mu_0}\boldsymbol{\Delta} \end{bmatrix} \begin{bmatrix} \widetilde{\mathbf{A}}_{\mathrm{R}} \\ \widetilde{\mathbf{A}}_{\mathrm{I}} \end{bmatrix} = \begin{bmatrix} -\widetilde{\mathbf{J}}_{\mathrm{coil}} \\ 0 \end{bmatrix}, \tag{2.45}$$

Couplings

Lorentz force:

$$\overline{\mathbf{f}}_{\mathrm{L}} = \frac{\sigma\omega}{2}\left[\widetilde{\mathbf{A}}_{\mathrm{I}} \times \left(\boldsymbol{\nabla} \times \widetilde{\mathbf{A}}_{\mathrm{R}}\right) - \widetilde{\mathbf{A}}_{\mathrm{R}} \times \left(\boldsymbol{\nabla} \times \widetilde{\mathbf{A}}_{\mathrm{I}}\right)\right], \tag{2.46}$$

Joule heat:

$$\overline{q}_{\mathrm{J}} = \frac{\sigma\omega^2}{2}\left(\left|\widetilde{\mathbf{A}}_{\mathrm{R}}\right|^2 + \left|\widetilde{\mathbf{A}}_{\mathrm{I}}\right|^2\right), \tag{2.47}$$

Electrical conductivity:

$$\sigma = \sigma(T), \tag{2.48}$$

Fluid domain

Continuity equation:

$$\frac{\partial\rho}{\partial t} + \boldsymbol{\nabla} \cdot (\rho\mathbf{u}) = 0, \tag{2.49}$$

Momentum equation:

$$\frac{\partial}{\partial t}(\rho\mathbf{u}) + \rho(\mathbf{u} \cdot \boldsymbol{\nabla})\mathbf{u} = -\boldsymbol{\nabla} p + \boldsymbol{\nabla} \cdot \boldsymbol{\tau} + \overline{\mathbf{f}}_{\mathrm{L}}, \tag{2.50}$$

Energy equation:

$$\frac{\partial}{\partial t}(\rho h) + \boldsymbol{\nabla} \cdot (\rho\mathbf{u}h) - \frac{\partial p}{\partial t} = \boldsymbol{\nabla} \cdot (k\boldsymbol{\nabla} T) + -q_{\mathrm{R}} + \overline{q}_{\mathrm{J}}, \tag{2.51}$$

Plasma radiation loss:

$$q_{\mathrm{R}}(T) = \begin{cases} 0, & \text{if} \quad T \leq 9500\,\mathrm{K}, \\ 5600 \cdot (T - 9500) + 181 \cdot (T - 9500)^2 & \text{else.} \end{cases} \tag{2.52}$$

2.5.2. OpenFOAM solvers

- **emFoam**: Custom-developed electromagnetic field solver based on vector potential formulation. Input: $\mathbf{J}_{\mathrm{coil}}$, ω, σ. Output: $\widetilde{\mathbf{A}}_{\mathrm{R}}$, $\widetilde{\mathbf{A}}_{\mathrm{I}} \rightarrow H_{\mathrm{mag}}$, $\overline{\mathbf{f}}_{\mathrm{L}}$, $\overline{q}_{\mathrm{J}}$.
- **emHeatFoam:** Extension of *emFoam*; includes the temperature equation to solve induction heating. Used to precompute initial fields: $\widetilde{\mathbf{A}}_{\mathrm{R}}$, $\widetilde{\mathbf{A}}_{\mathrm{I}}$, T.
- **rhoSimpleFoam:** Standard OpenFOAM solver to calculate the steady-state of the argon gas flow without electromagnetic fields. Used to precompute the velocity field $\mathbf{u}$.
- **plasmaSteadyFoam:** Custom-developed plasma solver. Computes the plasma steady-state. Requires precomputed initial conditions from *emHeatFoam* and *rhoSimpleFoam.*
- **plasmaTransientFoam:** Used for the transient flow calculation of the burning plasma. Precomputed initial conditions from *emHeatFoam* and *rhoSimpleFoam* are required as well.

All implementations have been done in `foam-extend 4.0` [Sou19a], which is a forked version of OpenFOAM.

CHAPTER 3

Results and Discussion

This chapter presents the simulation results obtained with the custom plasma solver developed in OpenFOAM. In order to validate the two-dimensional, axisymmetric ICP model described in the previous chapter, the Tekna-PL50 plasma torch will be used as a reference. Therefore, the electromagnetic solver *emFoam*, which has been written completely new in OpenFOAM, will be tested and validated by means of analytical methods first. A second purpose of running only the electromagnetic solver with the coil geometry of the plasma torch is to determine the appropriate mesh size and cell density, which then will be used in the simulation of the complete plasma model. The computational results will be validated by comparison against data from similar studies of Tekna-PL50, which can be found in the literature. Finally, the study is completed with an error analysis and evaluation of the simulation performance.

3.1. Plasma Torch Tekna-PL50

Figure 3.1 shows the scheme of the Tekna-PL50 plasma torch, which consists of a quartz cylinder with three gas inlets (outer cooling stream, intermediate plasma-forming stream and the central one used to inject powder into the plasma zone) and an induction coil surrounding the cylinder. The dimensions and operating conditions of the plasma torch, which were given in [XPB01] and [Ber+03] are summarized in Table 3.1. Note, that the coil current I, which is an input parameter of the computational model, is not uniquely given in the literature. Two different values of $I = 150\,\mathrm{A}$, and $I = 161\,\mathrm{A}$ were mentioned by Bernardi et al [Ber+03] and Xue et al [XPB01], respectively. Therefore, it may be necessary to adjust the coil current in the model so that a dissipated plasma power of about $5\,\mathrm{kW}$ is achieved. Furthermore, the computational model requires values for the width R_{EM} and height L_{EM} of the electromagnetic domain to be specified. These values were chosen independently from the literature data and will be explained in the course of this chapter.

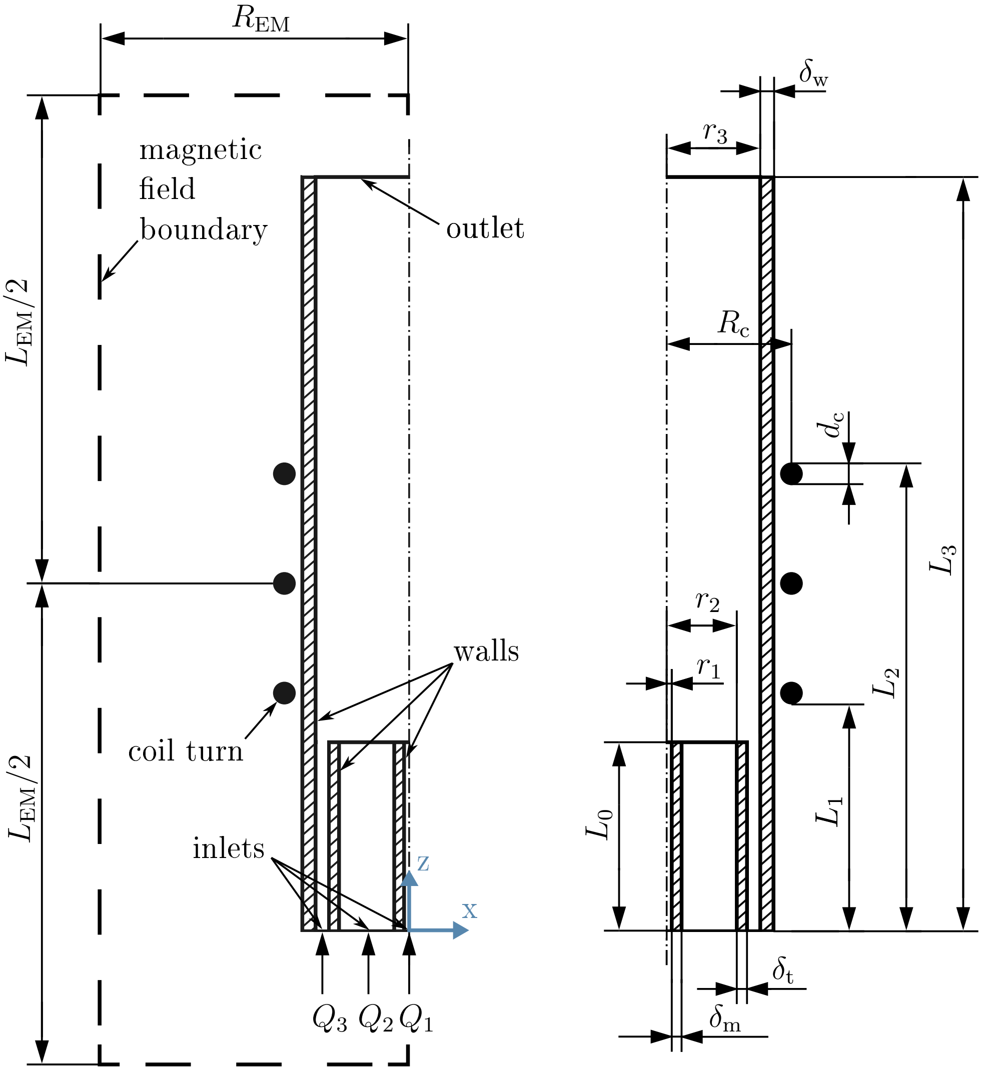


Figure 3.1.: Plasma torch scheme (Tekna-PL50)

Table 3.1.: Plasma torch characteristics (Tekna-PL50)

Parameter	Value	Unit	Parameter	Value	Unit
r_1	1.7	mm	Q_1	1	slpm[1]
r_2	18.8	mm	Q_2	3	slpm
r_3	25	mm	Q_3	31	slpm
δ_m	2	mm	P_{diss}	5	kW
δ_t	2.2	mm	f	3	MHz
δ_w	3.5	mm	I	[150, 161]	A
L_0	50	mm	d_c	6	mm
L_1	60	mm	R_c	33	mm
L_2	124	mm	R_{EM}	200	mm
L_3	200	mm	L_{EM}	300	mm

[1] standard liter per minute

3.2. Validation: Electromagnetic Solver

In order to validate the implementation of the electromagnetic solver, two test cases shown in Figure 3.2 were investigated. First, the coil in the air only is considered, meaning that the electrical conductivity is set to zero throughout the whole computation domain. In the second test case, in contrary, a constant electrical conductivity $\sigma = 2500\,\mathrm{S\,m^{-1}}$ is applied to the fluid region. It can be noted that the behavior of the magnetic field changes significantly if an electrically conductive fluid is treated. This difference can be explained by the induction phenomenon, the so-called *skin-effect*. The skin effect occurs due to the fact that eddy currents are induced in the conductor, which in turn generate magnetic fields that interact with the superimposed magnetic field. In the following, the accuracy of the numerical solution by means of the *emFoam* code is evaluated for both test cases by comparing the results against analytical solutions.

Coil in the air ($\sigma = 0$): The computational mesh was created by means of OpenFOAM's *blockMesh* utility, see also notes in the Appendix C. Since OpenFOAM is based on the finite volume method, a wedge geometry, as illustrated in Figure 3.2, is used to calculate two-dimensional axisymmetric models. Note, that the Cartesian coordinate system is aligned to the symmetry axes of the wedge so that the x-axis can be considered as the radial axis and the y-axis as the tangential axis, similar to a cylindrical coordinate system. Thus, the coil current density is specified only by a tangential y-component, which is given by

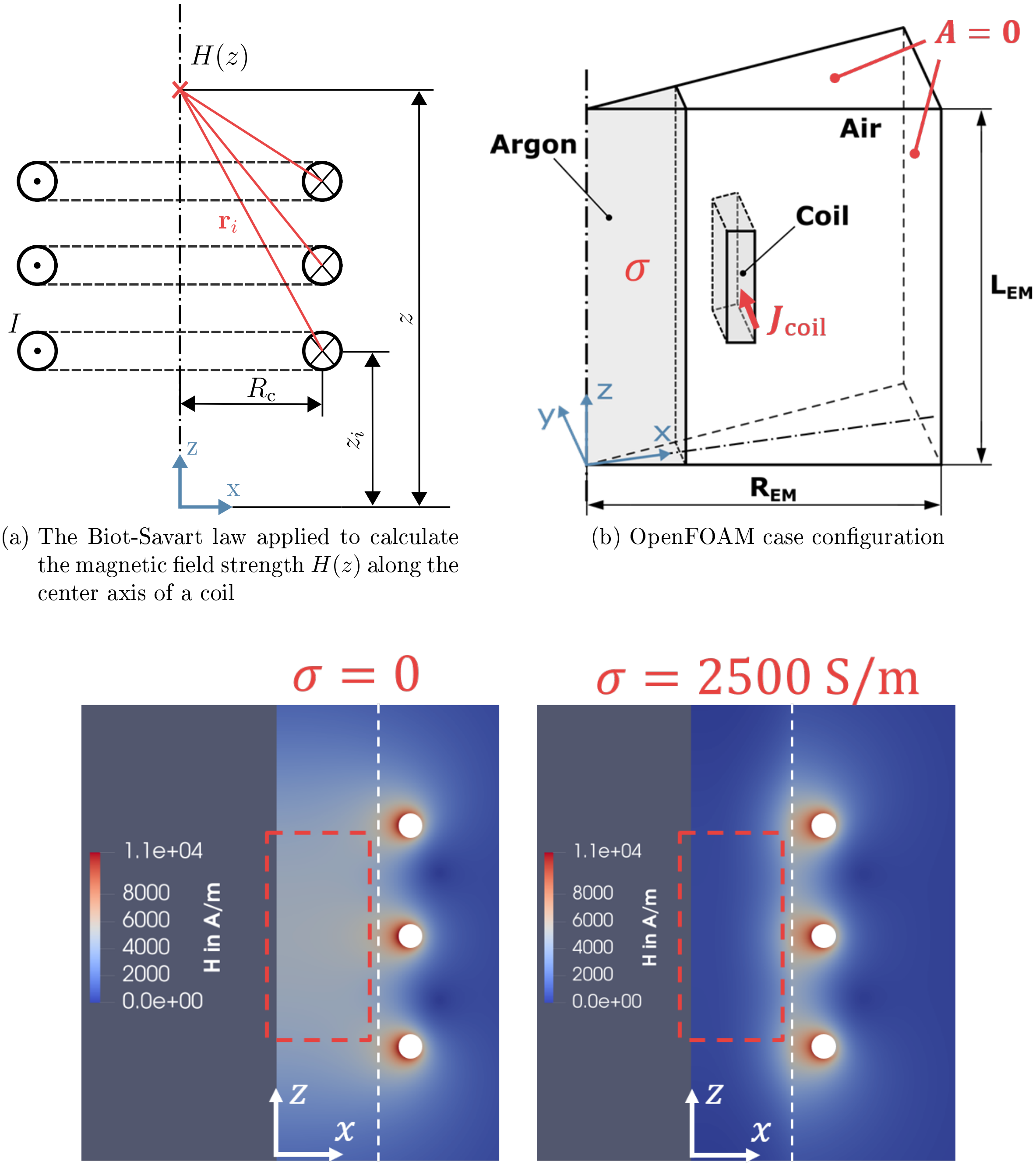
$$J_\mathrm{y} = \frac{I}{\frac{\pi}{4} {d_\mathrm{c}}^2}, \tag{3.1}$$

where I is the coil current and d_c is the coil wire diameter. Further, the far-field approximation $\widetilde{\mathbf{A}}_\mathrm{R} = \mathbf{0}$, $\widetilde{\mathbf{A}}_\mathrm{I} = \mathbf{0}$ is used as a boundary condition for the vector potential. This means that the magnetic field decreases to zero at infinitely great distances from the current source. However, in a computational model this can be only approximated. Therefore, the choice of the domain size ($R_\mathrm{EM} \times L_\mathrm{EM}$) plays an important role, as the position of the vector potential boundaries will affect the solution of the magnetic field. The influence of domain size and mesh density is discussed in more detail in section 3.4, whereas in this section only the results for the following parameters are presented: A width of $R_\mathrm{EM} = 200\,\mathrm{mm}$ and a height of $L_\mathrm{EM} = 300\,\mathrm{mm}$ with the domain discretized into 400×600 cells yielding an uniform hexahedral mesh with a cell size of $0.5\,\mathrm{mm}$. It was found that these parameters are a good trade-off between accuracy and low computation time and therefore they are being used in later simulations of the plasma solver as well.

For the purpose of validation, an analytical solution of the total magnetic field strength $H(z)$ along the center axis of the coil, as illustrated in Figure 3.2a, can be obtained by using Biot-Savart's law and the principle of superposition by

$$H(z) = \sum_{i=1}^{3} \frac{I}{2} \frac{{R_\mathrm{c}}^2}{|\mathbf{r}_i|^3} = \sum_{i=1}^{3} \frac{I}{2} \frac{{R_\mathrm{c}}^2}{\sqrt{{R_\mathrm{c}}^2 + (z - z_i)^2}^3}\,, \tag{3.2}$$

where z_i is the axial position of the i-th coil turn and R_c is the radial position of all coil turns. While the values for the coil geometry R_c, d_c are given in Table 3.1, all the other parameters of this test case are summarized in Table 3.2.



(a) The Biot-Savart law applied to calculate the magnetic field strength $H(z)$ along the center axis of a coil

(b) OpenFOAM case configuration

(c) Evaluation of the influence of electrical conductive material on the magnetic field H of a coil. Results obtained with the custom *emFoam* solver; size of the computational domain: $R_{\text{EM}} = 200\,\text{mm}$ and $L_{\text{EM}} = 300\,\text{mm}$, coil current: $I = 161\,\text{A}$, position of the inner torch wall (white dotted line): $x = r_3 = 25\,\text{mm}$

Figure 3.2.: Test cases for the validation of the electromagnetic solver *emFoam*

Table 3.2.: Parameters for the *emFoam* test case

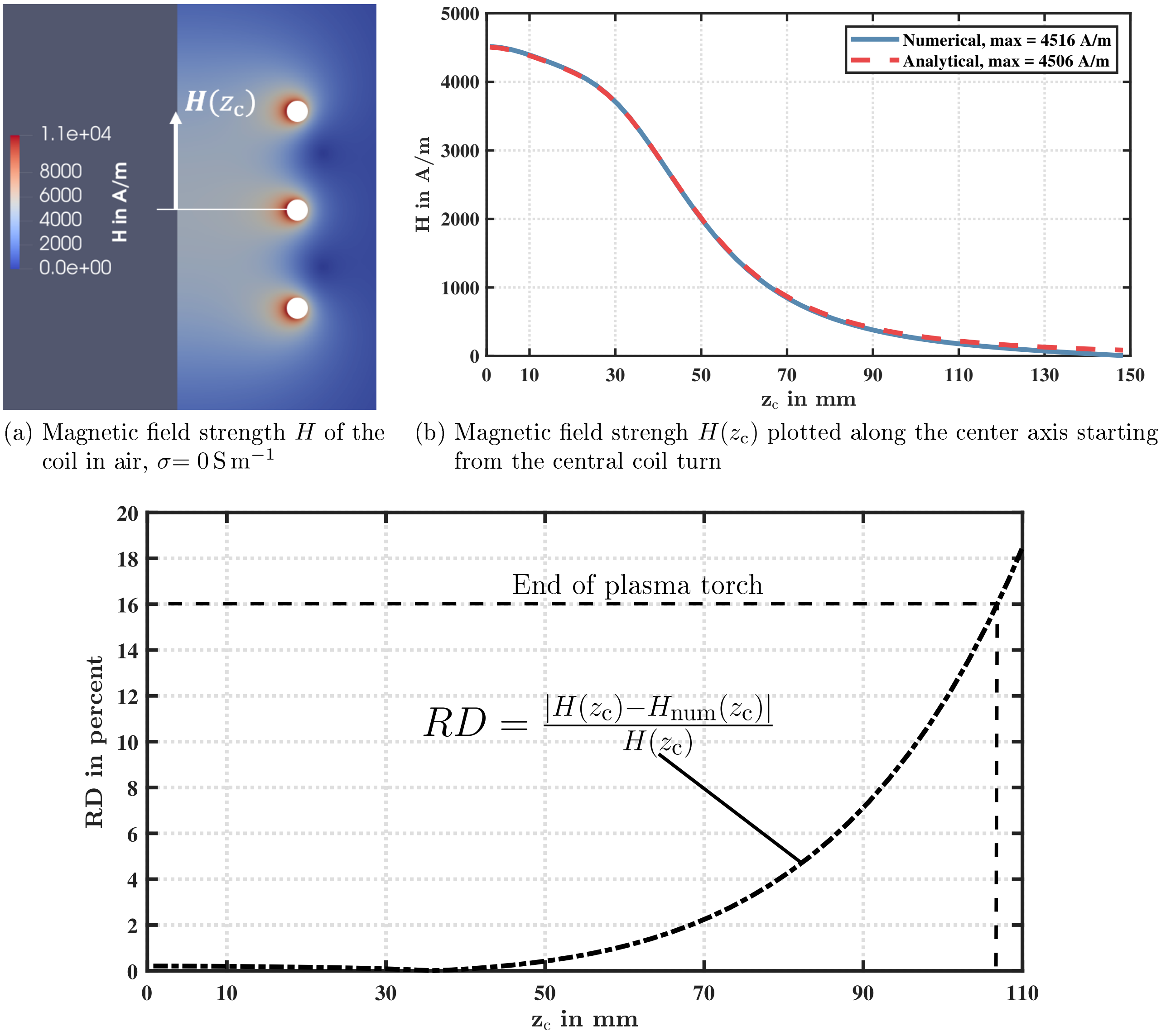
Parameter	Value
Vector potential	$\widetilde{\mathbf{A}}_{\mathrm{R}} = \mathbf{0}$, $\widetilde{\mathbf{A}}_{\mathrm{I}} = \mathbf{0}$
Coil current density	$\mathbf{J}_{\mathrm{coil}} = (0\; J_{\mathrm{y}}\; 0)$
J_{y}	$5\,694\,210\,\mathrm{A\,m^{-2}}$
I	$161\,\mathrm{A}$
z_1	$121\,\mathrm{mm}$
z_2	$150\,\mathrm{mm}$
z_3	$179\,\mathrm{mm}$
ω	$3.3 \times 10^{7}\,\mathrm{s^{-1}}$
μ_0	$1.26 \times 10^{-6}\,\mathrm{V\,s\,A^{-1}\,m^{-1}}$
σ	$[0\ 2500]\ \mathrm{S\,m^{-1}}$
cells(R_{EM})	400
cells(L_{EM})	600

Figure 3.3a shows the numerical result of the magnetic field strength obtained by means of the *emFoam* code, which is compared against the analytical solution of the magnetic field along the center axis as shown in Figure 3.3b. The derivation of how the magnitude of the magnetic field strength H_{mag} is calculated numerically was described in the previous chapter, see Equation (2.30). For simplification, the magnitude H_{mag} will be referred to as the magnetic field strength H in the following. Also note, that in Figure 3.3 the magnetic field along the z-axis starting from the position of the central coil turn at $z = 92\,\mathrm{mm}$ is considered. The transformed coordinates z_{c} are therefore given by $z_{\mathrm{c}} = z + 92\,\mathrm{mm}$.

The increasing difference between numerical and analytical solution as shown in Figure 3.3c is an inherent approximation error caused by the far-field boundary conditions of the magnetic vector potential $\widetilde{\mathbf{A}}_{\mathrm{R}} = \widetilde{\mathbf{A}}_{\mathrm{I}} = 0$. The maximum relative difference of about 16% at the end of plasma torch, however, can be seen negligible, because if conductive material is present inside the torch, the magnetic field strength near the plasma central axis is very low due to the skin effect and thus contributes little to the induction heating process. At this point it can be concluded that the numerical solution meets the expectations and is in good agreement with the analytical solution. Note, that the accuracy may be even further improved by expanding the electromagnetic domain size.

Induction coil ($\sigma = 2500$): In the second test case, a constant electrical conductivity $\sigma = 2500\,\mathrm{S\,m^{-1}}$ was applied to the inner torch region in order to examine whether the induction phenomenon is being correctly treated by the solver code. Therefore, the numerical solution of the magnetic field strength along the radial oriented x-axis as noted in Figure 3.4b is compared to the analytical solution. Figure 3.4a shows the radial profile of the magnetic field, where the plot starts at the torch wall ($x_{\mathrm{c}} = 0$) and ends at the center of the torch ($x_{\mathrm{c}} = 25\,\mathrm{mm}$). The analytical solution is approximated by an exponential function with the initial value adopted from the surface value of the numerical solution. The main parameter for the evaluation here is the skin depth, which is given by

$$\delta = \sqrt{\frac{2}{\omega \mu_0 \sigma}}. \tag{3.3}$$



(a) Magnetic field strength H of the coil in air, $\sigma = 0\,\mathrm{S\,m^{-1}}$

(b) Magnetic field strengh $H(z_c)$ plotted along the center axis starting from the central coil turn

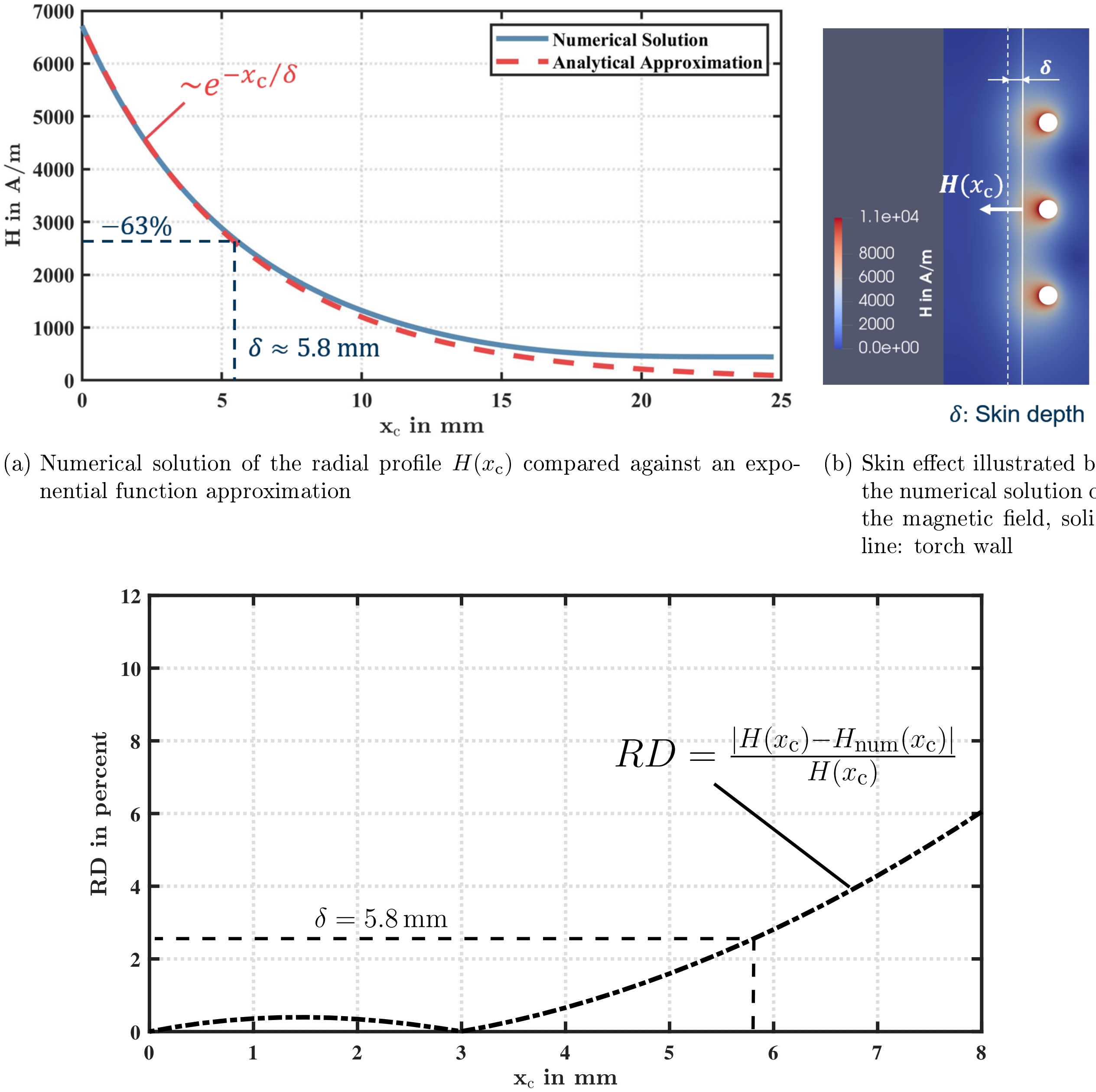
(c) Relative difference (RD) between the analytical and numerical solution

Figure 3.3.: Axial profile of the magnetic field strength of a coil in air; comparison of analytical and numerical solutions with coil current $I = 161\,\mathrm{A}$

The skin depth δ describes the depth where the magnetic field strength has decreased about 63% from its surface value

$$H(x_c = \delta) \approx 0.37 \cdot H(x_c = 0). \tag{3.4}$$

As shown in Figure 3.4c, the relative difference between numerical and analytical solution is less than 3% in the range of the skin depth $x_c \leq \delta \approx 5.8\,\mathrm{mm}$. The increasing difference towards the center, as noted in Figure 3.4a, is due to the fact that the exponential function is not the exact analytical solution in the case of cylindrical geometry. In order to get a more accurate solution, one would have to exploit the Bessel functions. The skin depth has another important consequence with respect to the inductively coupled plasma torch: 86% of the plasma power is dissipated within the area of the skin depth.



(a) Numerical solution of the radial profile $H(x_\mathrm{c})$ compared against an exponential function approximation

(b) Skin effect illustrated by the numerical solution of the magnetic field, solid line: torch wall

(c) Relative difference (RD) between the analytical and numerical solution radial profile of the magnetic field strength $H(x_\mathrm{c})$ as shown in Figure 3.4b

Figure 3.4.: Magnetic field strength H evaluated for a constant electrical conductivity of the inner torch region, $\sigma = 2500\,\mathrm{S\,m^{-1}}$, coil current $I = 161\,\mathrm{A}$

3.3. Validation: Plasma Solver

3.3.1. Mesh configuration

While the fluid equations are solved only in the inner region of the plasma torch with dimensions of $25\,\text{mm} \times 200\,\text{mm}$ ($r_3 \times L_3$), see Figure 3.1, the computational domain for the electromagnetic equations needs to be extended beyond the torch region as well. Therefore a multi-mesh technique was implemented into the plasma solver, as explained previously in section 2.4, in order to separately compute the fluid and electromagnetic equations on different meshes. The computational mesh for the plasma simulation was found based on the results of the mesh-dependency analysis conducted with the electromagnetic solver *emFoam*, which will be presented in the section 3.4.1. As a result the computational domain of $R_{\text{EM}} = 200\,\text{mm}$ and $L_{\text{EM}} = 300\,\text{mm}$ discretized into 400×600 cells was chosen. However, this configuration was slightly adjusted to match the real torch geometry shown in Figure 3.1. To maintain an average cell size of $\Delta x = \Delta z \approx 0.5\,\text{mm}$ a mesh configuration as shown in Figure 3.5 was created.

In similar studies an extended electromagnetic domain width of $R_{\text{EM}} = 125\,\text{mm}$ was proposed by Xue et al [XPB01], which was adopted also by Bernardi et al [Ber+03], while the height of the electromagnetic field region was extended only by Bernardi to $L_{\text{EM}} = 277\,\text{mm}$. In the study by Xue et al [XPB01] the region outside the torch was discretized by an unstructured mesh consisting of 7489 triangular cells, while the fluid region and quartz tubes were discretized into 54×80 (total 4320) quadrilateral cells. In the work of Bernardi et al the torch region consists of a uniform, structured mesh with 57×200 (total 11400) quadrilateral elements. In contrary, a non-uniform, structured grid with 81×71 (total 5751) nodes has been employed for the solution of the vector potential equations in the outer torch region, where each turn of the induction coil was resolved with 13 nodes.

In comparison, in this work the fluid and electromagnetic regions were discretized into a total of 19200 cells and 206500 cells, respectively; the coil current density patch as shown in Figure 3.5 consists of 114 cells. While the lateral cell size Δx inside the fluid region is comparable with those from the literature, the number of cells in axial direction is almost twice in this work. This was done to ensure good mesh properties by maintaining the aspect ratio of all cells close to one.

3.3.2. Initial and boundary conditions

The initial and boundary conditions used to obtain the plasma simulation results, which are presented in the following sections, are summarized in Table 3.3. In contrary to the validation of the electromagnetic solver in the previous section, the coil current here is set to $I = 150\,\text{A}$. In addition, atmospheric pressure at the outlet and constant wall temperature of 300 K apply. The latter implies that the wall temperature is cooled down and kept at a constant level, which in reality might not be true. For the purpose of validation, however, these values are adopted according to the model assumptions given in the literature. The axial inlet velocities components are calculated based on the torch operating conditions given in Table 3.1 as

$$u\,[\text{m}\,\text{s}^{-1}] = \frac{Q\,[\text{slpm}]}{A\,[\text{m}^2]} \cdot \frac{10^{-3}}{60}, \tag{3.5}$$

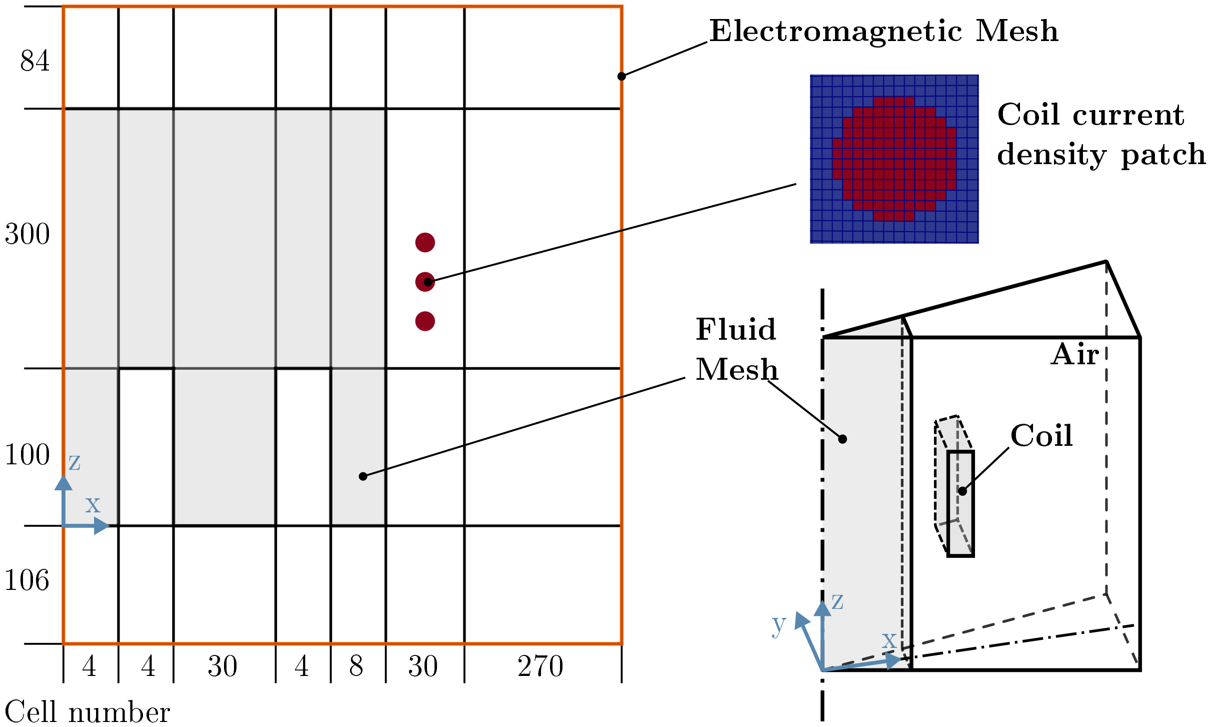


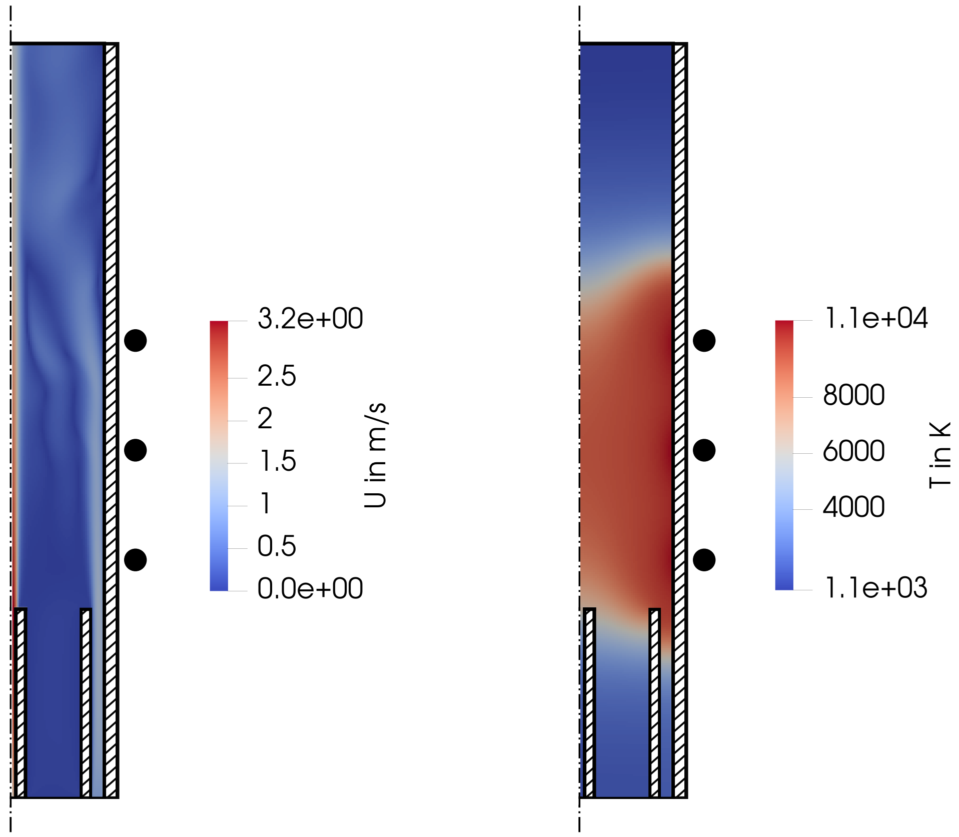
Figure 3.5.: Grid scheme of the fluid and electromagnetic mesh regions created by means of OpenFOAM's *blockMesh* utility; total number of cells: 19200 (fluid mesh), 206500 (electromagnetic mesh)

where Q is the flow rate given in unit standard liters per minute (slpm), yet assuming that such environmental conditions apply so that one slpm is equivalent to one unit liter per minute (lpm); A denotes the cross-section of the respective inlet, which can be obtained from the scheme in Figure 3.1.

Table 3.3.: Initial and boundary conditions for the plasma solvers. Similar conditions apply to the emHeatFoam and rhoSimpleFoam solvers except for the initial field values, which are $T = 300\,\mathrm{K}$ and $\mathbf{u} = \mathbf{0}$.

Field variable	Initial value	Boundary Conditions
$\mathbf{J}_{\mathrm{coil}}$	$J_{\mathrm{y}} = 5\,305\,165\,\mathrm{A\,m^{-2}}$ $(I = 150\,\mathrm{A})$	$\mathbf{J}_{\mathrm{coil}} = \mathbf{0}$
$\widetilde{\mathbf{A}}_{\mathrm{R}}$, $\widetilde{\mathbf{A}}_{\mathrm{I}}$	precomputed with emHeatFoam	All boundaries: fixed value $\widetilde{\mathbf{A}}_{\mathrm{R}} = \mathbf{0}$, $\widetilde{\mathbf{A}}_{\mathrm{I}} = \mathbf{0}$
$\mathbf{u}$	precomputed with rhoSimpleFoam	Inlets: fixed value $u_1 = 1.84\,\mathrm{m\,s^{-1}}$, $u_2 = 0.047\,\mathrm{m\,s^{-1}}$, $u_3 = 0.894\,\mathrm{m\,s^{-1}}$, Walls: no slip, Outlet: zero gradient $\frac{\partial \mathbf{u}}{\partial \mathbf{n}} = 0$
T	precomputed with emHeatFoam	Inlets and walls: fixed value $T = T_{\mathrm{room}} = 300\,\mathrm{K}$, Outlet: zero gradient $\frac{\partial T}{\partial \mathbf{n}} = 0$
p	$p = p_{\mathrm{atm}} = 1013\,\mathrm{hPa}$	Inlets and walls: zero gradient $\frac{\partial p}{\partial \mathbf{n}} = 0$, Outlet: fixed value $p = p_{\mathrm{atm}} = 1013\,\mathrm{hPa}$

As described in the previous chapter in section 2.4, the initial field values of the plasma state were merged from the solutions of the *rhoSimpleFoam* solver, which was used to precompute the velocity field $\mathbf{u}$ of the pure gas flow, and the custom *emHeatFoam* solver, computing the temperature field T and magnetic vector potential $\widetilde{\mathbf{A}}_{\mathrm{R}}$, $\widetilde{\mathbf{A}}_{\mathrm{I}}$. Figure 3.6 shows the solutions for the velocity and temperature field obtained by *rhoSimpleFoam* (2000 solver iterations) and *emHeatFoam* (350 solver iterations), respectively. Note, that the maximum values of the precomputed temperature field are relatively high. It was found that initializing the plasma solver by using a temperature distribution with smooth gradients and an average temperature above 6000 K in the plasma region is the best practice to achieve a stable solution.



(a) Velocity field computed with *rhoSimpleFoam* (2000 iterations)

(b) Temperature field computed with *emHeatFoam* (350 iterations)

Figure 3.6.: Separately precomputed velocity and temperature field used as initial conditions for the plasma solver

3.3.3. Steady-state solver

In this section the plasma steady-state results obtained by means of the custom OpenFOAM code are compared against results, which were obtained by means of the commercial FLUENT code (today: ANSYS Fluent) by the following authors:

- Figure 3.7 and 3.8b: Bernardi et al [Ber+03; Ber+05],
- Figure 3.8a: Ilya Tsivilskiy (these simulation results are not published yet, but the numerical model has been described in [Nag+17]),
- Figure 3.9 and 3.10: Xue et al [XPB01].

Basically, all authors used identical model assumptions, however, it cannot be ruled out that some model and simulation parameters may differ, as will be explained later.

Description of the data. The plasma temperature fields (raw numerical field values) are compared in Figure 3.8, whereas Figure 3.7a shows a temperature contour plot with the lines indicating the levels of constant temperature as annotated on the respective line. The streamlines as shown in Figure 3.7b (right) were calculated by means of with MATLAB's `streamline()` function based on the velocity field data from OpenFOAM. These results can be qualitatively compared to the stream function contours (left part of the Figure 3.7b) obtained by Bernardi et al. Note, that in order to better compare the results to the reference data from literature the plots in Figure 3.7 are stretched in width and are therefore not true to scale. In contrary, in the plots in all the other Figures the downscaled torch geometry is presented maintaining the true aspect ratio.

In Table 3.4 the numerical values are noted to create the contour plots of the real and imaginary parts of the magnetic vector potential, shown in Figure 3.9, as well as the contours of temperature and Joule heat density fields shown in Figure 3.10a and 3.10b, respectively. The same contour levels are used as in the reference data by Xue et al [XPB01]. Note, that for the OpenFOAM results, shown in the right half of each plot, the contour levels are also indicated by the ticks in the colorbar. In Figure 3.10c the same streamlines are shown again in comparison with stream function contours by Xue et al.

Table 3.4.: Contour levels used in figures 3.9 and 3.10 for comparing results with Xue et al [XPB01]

Field	Unit	Minimum	Incremental value	Maximum
T	K	781	481	9440.2
$\overline{q}_\mathrm{J}$	$\mathrm{W\,m^{-3}}$	6.507e6	6.5e6	1.366e8
$\widetilde{\mathbf{A}}_\mathrm{R,y}$	$\mathrm{V\,s\,m^{-1}}$	-5.87e-07	3e-06	1.07e-04
$\widetilde{\mathbf{A}}_\mathrm{I,y}$	$\mathrm{V\,s\,m^{-1}}$	-1.37e-05	7.22e-07	-7.2e-07

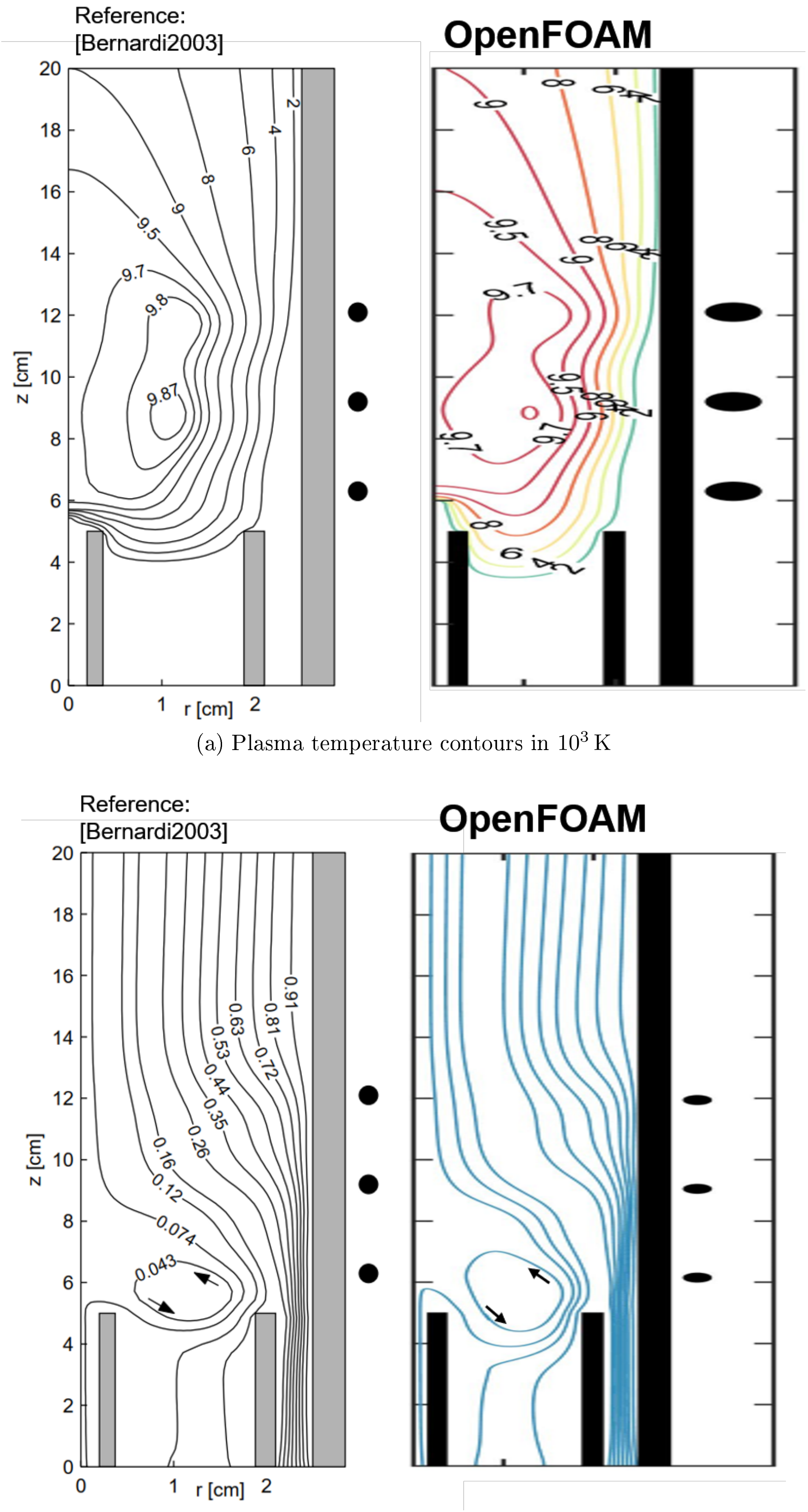
Data evaluation and comments. The presented OpenFOAM results show overall good agreement to the reference data. Some minor differences are observed for the temperature distribution near the inlets, especially the central one (Figure 3.7a, 3.10a). This might be due to different treatment of the flow velocity boundary conditions, as in the literature only the flow rates in slpm are given. In contrary, the temperature distribution above the first inlet matches with the results by Ilya Tsivilskiy in Figure 3.8a, which have been obtained using identical inlet boundary conditions as in OpenFOAM. Also, it may be assumed that the flow inside the central tube has not been simulated by Xue et al, see the white fill in figures 3.10.

Based on the data of Figure 3.8b, the maximum temperature that has been calculated in OpenFOAM is about 120 K lower than the one reported by Bernardi et al. This equals a relative difference about 1% and is thus within an acceptable range. In the literature two different values for the coil current $I = 150\,\mathrm{A}$ and $I = 161\,\mathrm{A}$ were proposed by Bernardi et al [Ber+03] and Xue et al [XPB01], respectively. The latter leads to dissipated power of $P_{\mathrm{diss}} = 6\,\mathrm{kW}$ in the OpenFOAM simulations, whereas $I = 150\,\mathrm{A}$ yields a dissipated power of $P_{\mathrm{diss}} = 5.1\,\mathrm{kW}$, which deviates from the expected power output by 2%.

There may be more possible explanations due to different settings of linear solvers, discretization schemes, mesh configuration or data for the argon properties. Note, that the OpenFOAM solutions presented here have been obtained using the mass density approximation, see section 2.2, which already can contribute an error of up to 2%. However, based on the results presented here the implementation of the plasma solver can be seen validated, which is the main purpose of this work.

3.3.4. Transient solver

In future works, powder particles will be added to the model in order to simulate the spheroidization process. Therefore, the transient solution of the plasma by means of the *plasmaTransientFoam* code is considered now. Basically, the same initial and boundary conditions as for the steay-state simulation with *plasmaSteadyFoam* are used. The solution of temperature field is calculated with a simulation time step of $dt = 0.0005\,\mathrm{s}$. Figure 3.11 shows the results for the time steps $t = [0,\ 0.01,\ 0.05,\ 0.1,\ 0.2,\ 1]$ s. Finally, at $t = 1\,\mathrm{s}$ it can be seen that the solutions of the transient *plasmaTransientFoam* and steady-state *plasmaSteadyFoam* solver are consistent.



(a) Plasma temperature contours in 10^3 K

(b) Streamlines

Figure 3.7.: OpenFOAM simulation results for Tekna-PL50 compared against results by Bernardi et al. (2003). Reprinted by permission from Springer Nature: Springer-Verlag, from Bernardi et al. [Ber+03]

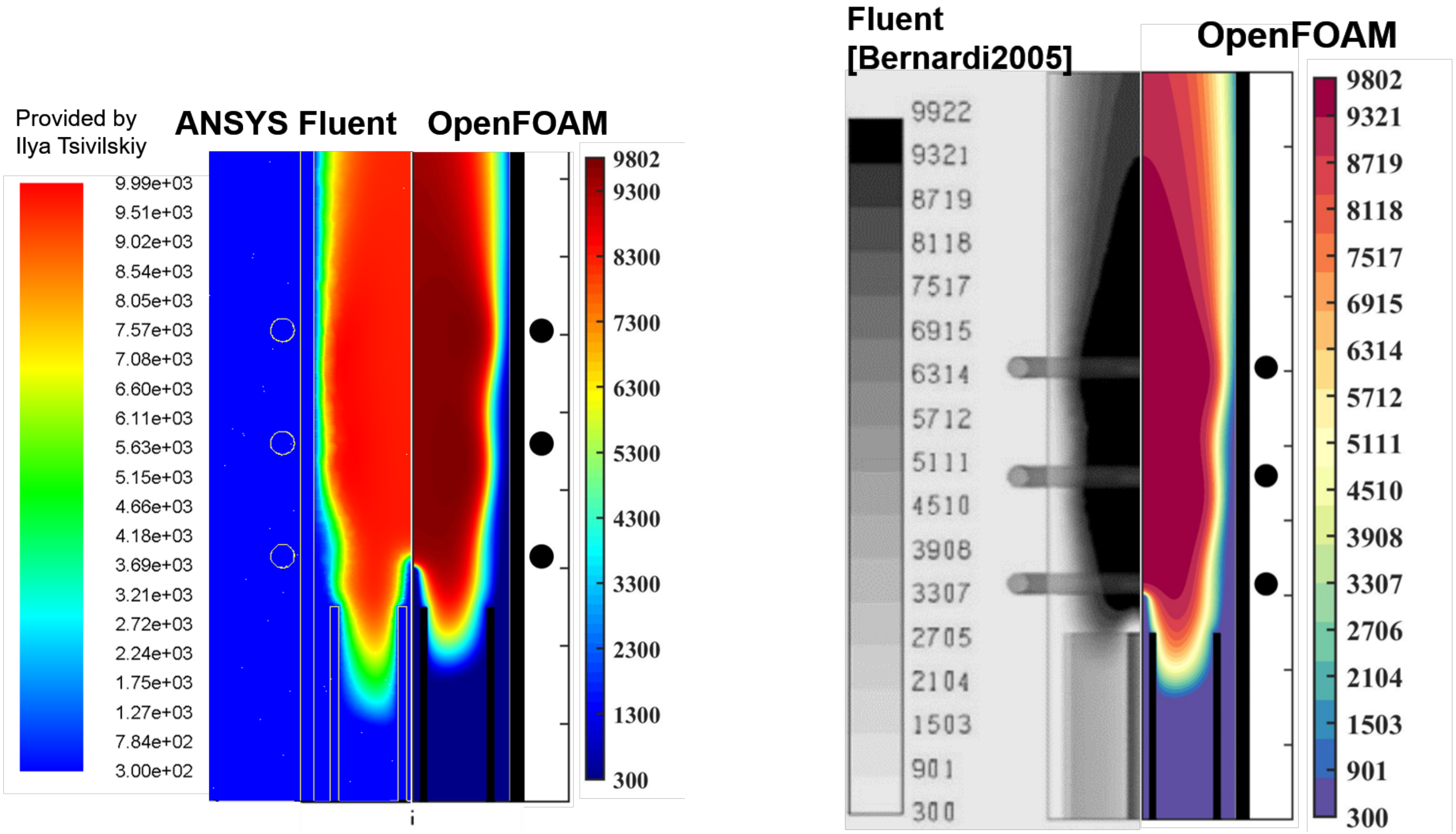


Figure 3.8.: Temperature field in Kelvin calculated with the custom OpenFOAM code in comparison against results obtained by Ilya Tsivilskiy (left) and Bernardi et al. [Ber+05] (right) with the commercial Fluent code (today ANSYS Fluent)

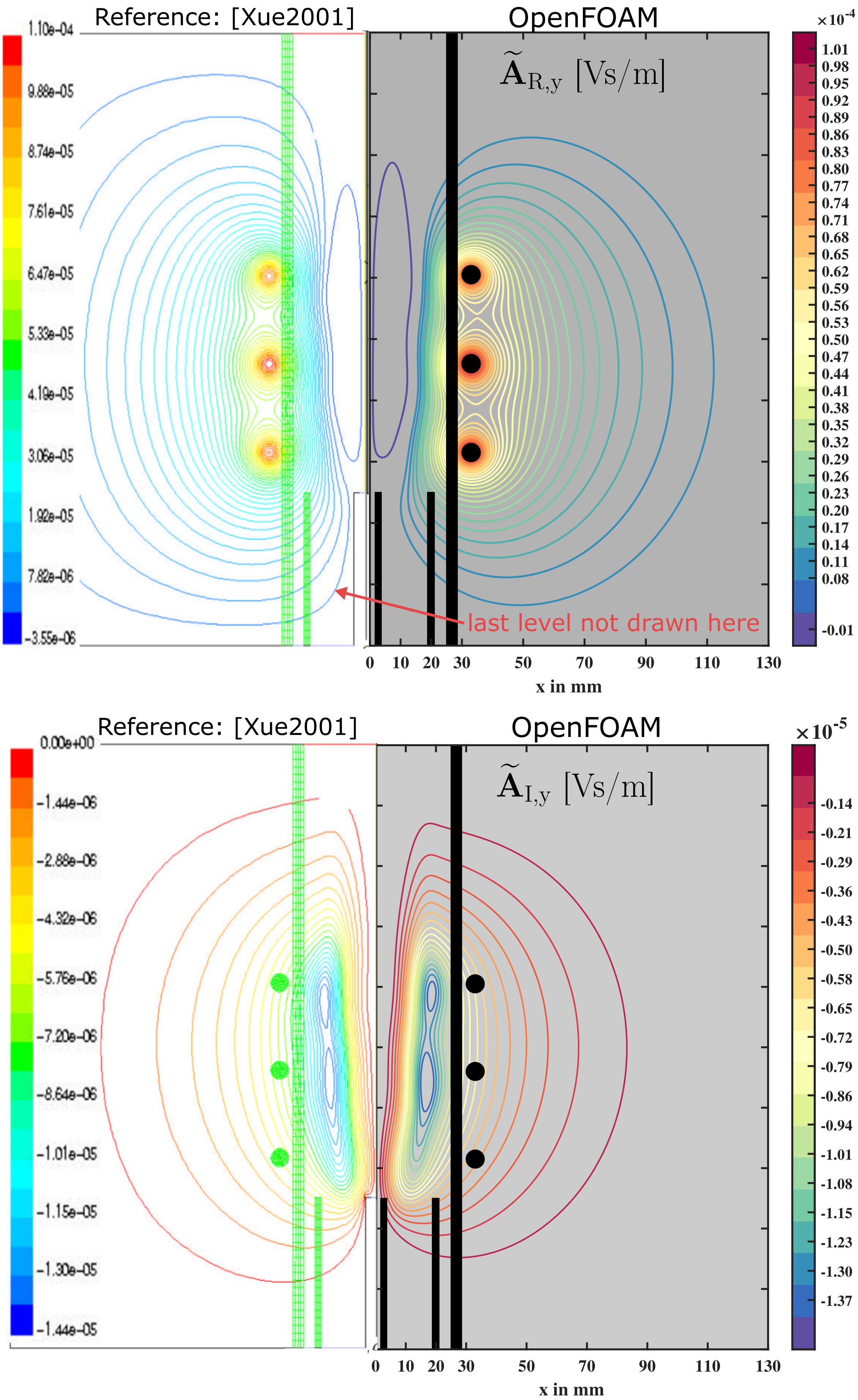
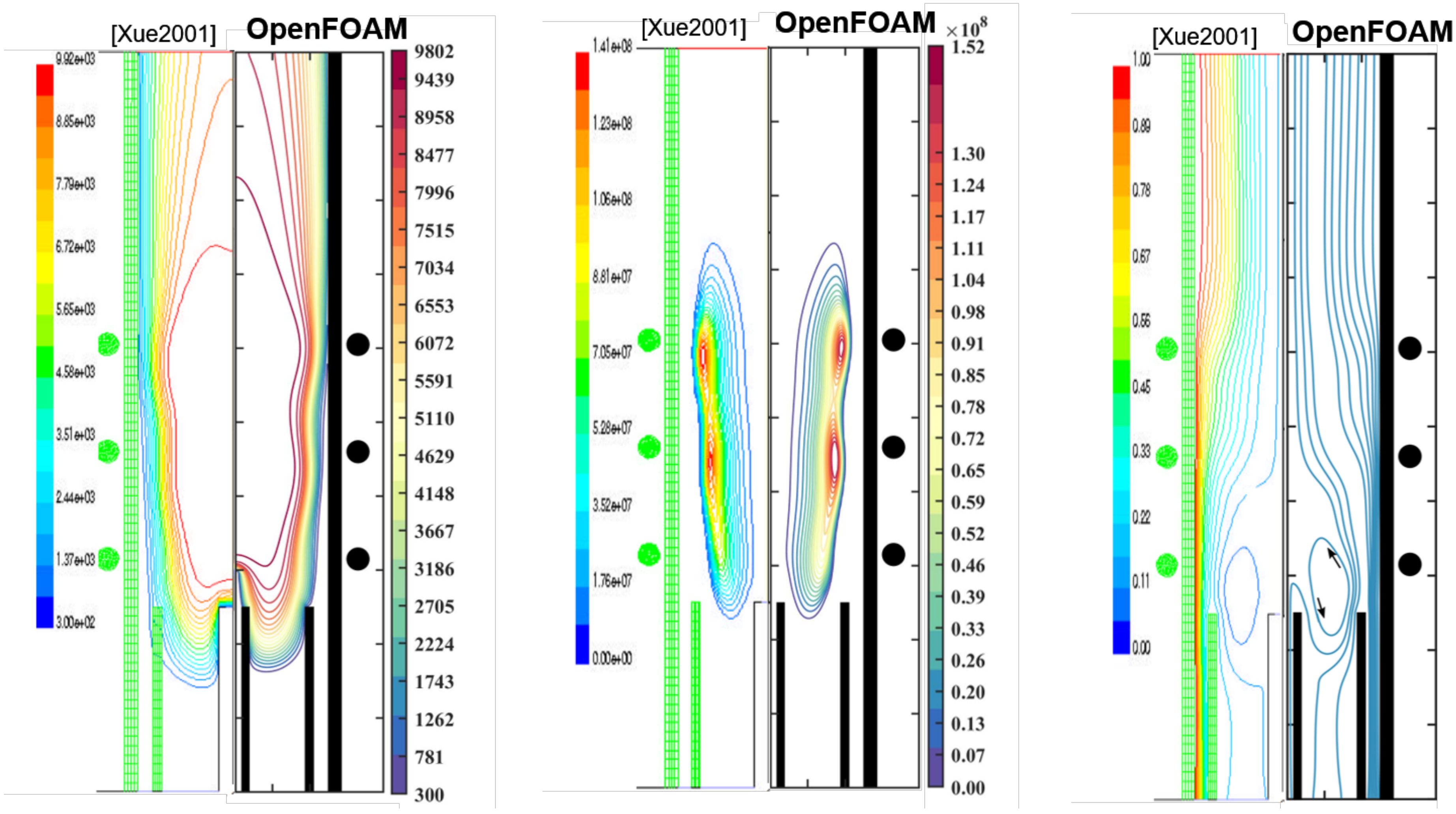


Figure 3.9.: Contours for the real and imaginary parts of the vector potential calculated with OpenFOAM compared against results by Xue et al. (2001). Used with permission of IOP Publishing Ltd, from Xue et al. [XPB01].



(a) Plasma temperature in K

(b) Joule heat density in W m^{-3}

(c) Streamlines

Figure 3.10.: OpenFOAM simulation results for the Tekna-PL50 plasma torch compared against results by Xue et al. (2001). Used with permission of IOP Publishing Ltd, from Xue et al. [XPB01] permission conveyed through Copyright Clearance Center, Inc.

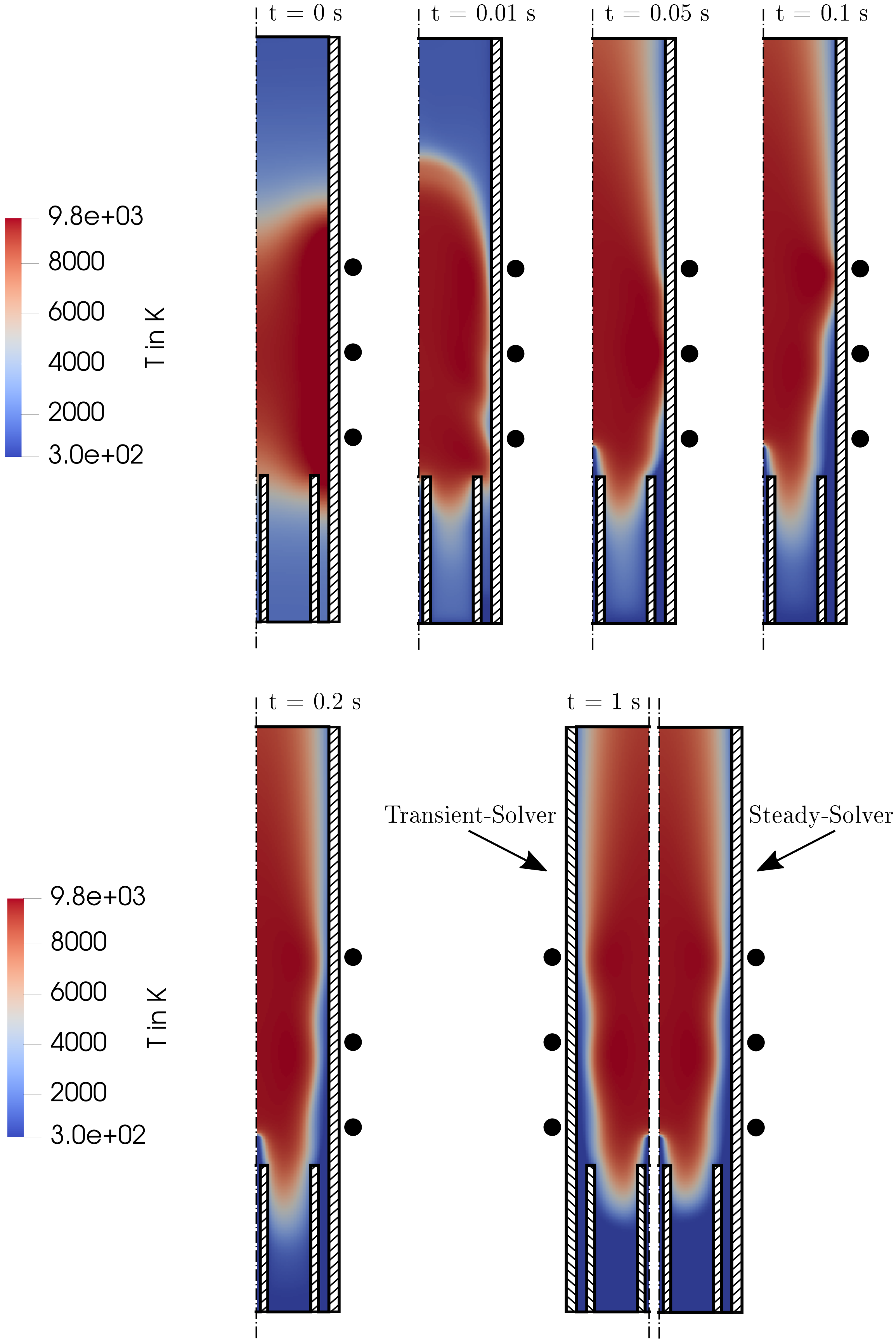


Figure 3.11.: Transient solution of the temperature field obtained with *plasmaTransientFoam*; at $t = 1\,\mathrm{s}$ comparison with steady-state solution obtained with *plasmaSteadyFoam*

3.4. Error Analysis

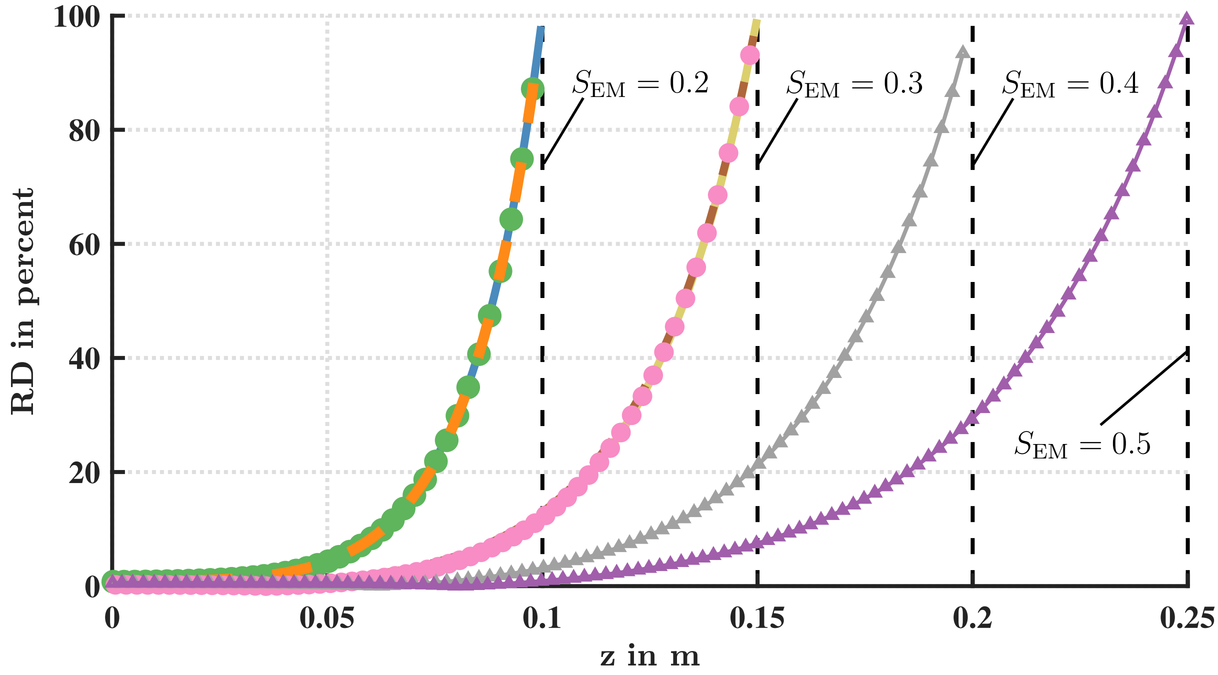
The error analysis is an essential part of numerical simulation studies in order to evaluate the quality and accuracy of the results. In general, there are two types of numerical errors that can be distinguished [Sch13, p. 101]: The first one is the discretization error, which is a result of the spatial discretization of the computational domain and the equation discretization. In CFD simulations second-order accuracy can be achieved if linear difference schemes are used. In other words, on a uniform structured mesh the numerical error decreases quadratically with decreasing cell size [Sch13, p. 69]. A mesh-dependency analysis using the electromagnetic solver *emFoam* is conducted in the first part of this section to investigate the influence of the cell size and to examine the influence of the magnetic field boundaries. Another numerical error is made due to fact that iterative methods are used to solve the linear systems of equations. However, the so-called iteration error can be greatly minimized by a sufficient number of iteration steps, which is also referred to as solution convergence and will be discussed in the second part of this section.

3.4.1. Mesh-dependency analysis

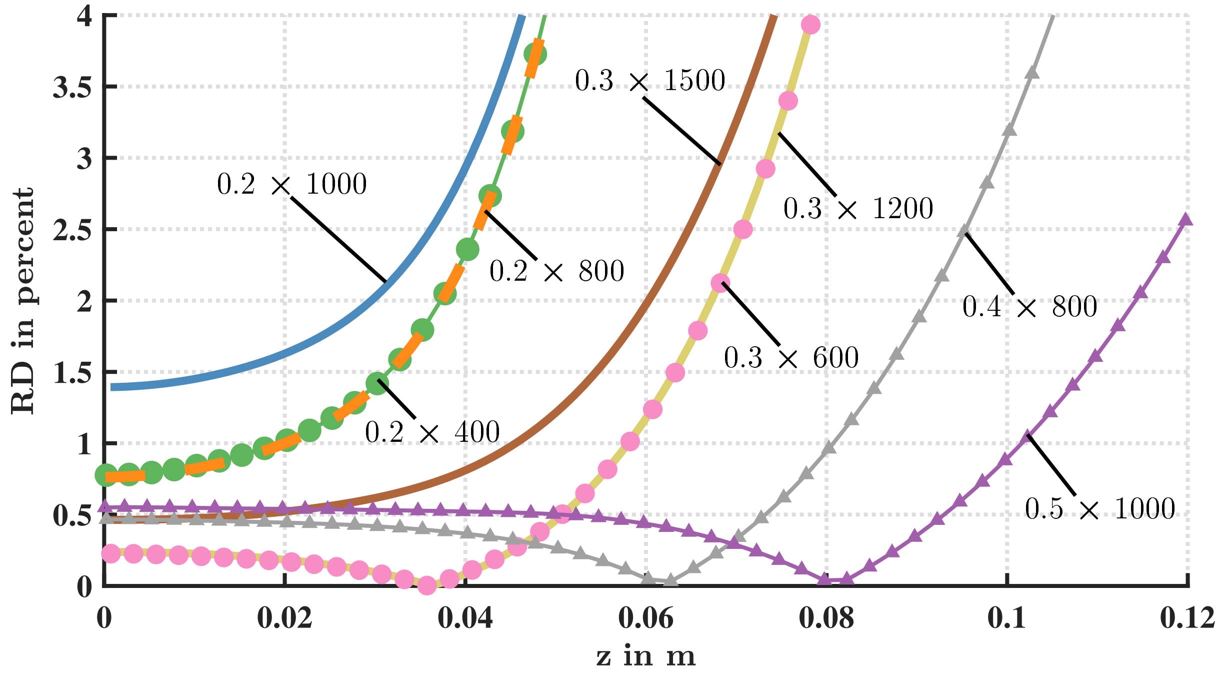
Coil in air ($\sigma = 0$)

The numerical solution of the magnetic field strength along the center axis $H(z)$ is computed for different mesh configurations and compared against the analytical solution similar to the validation procedure previously described in section 3.2. Figure 3.12 shows the relative difference RD, similarly calculated as in Figure 3.3b, obtained for a square-shaped computational domain $R_{\mathrm{EM}} = L_{\mathrm{EM}} = S_{\mathrm{EM}}$ with the domain size S_{EM} varying from $S_{\mathrm{EM}} = 0.2\,\mathrm{m}$ to $S_{\mathrm{EM}} = 0.5\,\mathrm{m}$ in steps of $0.1\,\mathrm{m}$, and different cell sizes [0.2 0.25 0.5] mm yielding the mesh configurations summarized in the first block of the Table 3.5. From Figure 3.12a it can be noted that the cell size has little influence on the solution, whereas the position of the boundaries by changing the domain size S_{EM} determines when the magnetic field strength is forced to zero. This inconsistency is unavoidable when the zero vector potential boundary condition is used. However, at a distance of about $50\,\mathrm{mm}$ away from the boundary the relative difference in most cases is already less than 20%.

From the zoomed view in Figure 3.12b only minor differences can be noted depending on the chosen cell size and domain size configuration. Surprisingly, increasing the number of cells on a fixed domain size does not lead to higher accuracy, for example, comparing 0.2×1000 configuration with 0.2×800 and 0.2×400 or comparing 0.3×1500 with 0.3×1200 and 0.3×600. Based on this finding the computational cost can be greatly reduced without loosing accuracy; for example, solving the vector potential system on the 0.3×1500 mesh takes about 18 minutes while the 0.3×600 mesh was calculated after approximately 1.5 minutes. The workstation specifications and further information regarding simulation performance are presented in section 3.5. Based on the configuration of 0.3×600, the influence of reducing the domain width R_{EM} while maintaining a fixed cell size of $\Delta x = 0.5\,\mathrm{mm}$ was studied. From the results shown in Figure 3.13 it can be concluded that $R_{\mathrm{EM}} = 0.1\,\mathrm{m}$ leads to a significant error, while an error less than 1% within the coil region (ending at $z = 0.032\,\mathrm{m}$) can be achieved by using a domain width of $R_{\mathrm{EM}} = 0.15\,\mathrm{m}$ and larger. Also, very little difference is noticed between $R_{\mathrm{EM}} = 0.2\,\mathrm{m}$ and larger sizes.



(a) Simulation results for different computational domain sizes $R_{\mathrm{EM}} = L_{\mathrm{EM}} = S_{\mathrm{EM}}$. The deviation of 100% is due to the zero vector potential boundary conditions.



(b) Zoomed view of the Figure above; $S_{\mathrm{EM}} \times N$, where S_{EM} is the square size of the electromagnetic computational domain $R_{\mathrm{EM}} = L_{\mathrm{EM}} = S_{\mathrm{EM}}$; N is the number of cells per length $N_{\mathrm{x}} = N_{\mathrm{z}} = N$

Figure 3.12.: OpenFOAM simulation results for different sizes of the electromagnetic computational domain and different mesh densities, showing the relative difference *RD* between numerical and analytical solution of $H(z)$ for $\sigma = 0$, see also Figure 3.3

Table 3.5.: Overview of the analyzed mesh configurations. Only uniform structured meshes with hexagonal cell type were studied. Therefore, Δx denotes the square cell size.

Mesh size study (Figure 3.12)	S_{EM} in m		$N_{\mathrm{x}} = N_{\mathrm{z}} = N$		Δx in mm
	0.2		400		0.50
	0.2		800		0.25
	0.2		1000		0.20
	0.3		600		0.50
	0.3		1200		0.25
	0.3		1500		0.20
	0.4		800		0.50
	0.5		1000		0.50
Width study (Figure 3.14)	R_{EM} in m	L_{EM} in m	$N_{\mathrm{x}} = N(R_{\mathrm{EM}})$	$N_{\mathrm{z}} = N(L_{\mathrm{EM}})$	Δx in mm
	0.1	0.3	200	600	0.50
	0.15	0.3	300	600	0.50
	0.2	0.3	400	600	0.50
	0.25	0.3	500	600	0.50
	0.3	0.3	600	600	0.50
Cell size study (Figure 3.14)	S_{EM} in m		$N_{\mathrm{x}} = N_{\mathrm{z}} = N$		Δx in mm
	0.3		200		1.50
	0.3		300		1.00
	0.3		400		0.75
	0.3		500		0.60
	0.3		600		0.50
	0.3		700		0.43
	0.3		800		0.38
	0.3		900		0.33
	0.3		1000		0.30
	0.3		1100		0.27
	0.3		1200		0.25
	0.3		1300		0.23
	0.3		1400		0.21
	0.3		1500		0.20
	0.3		1600		0.19

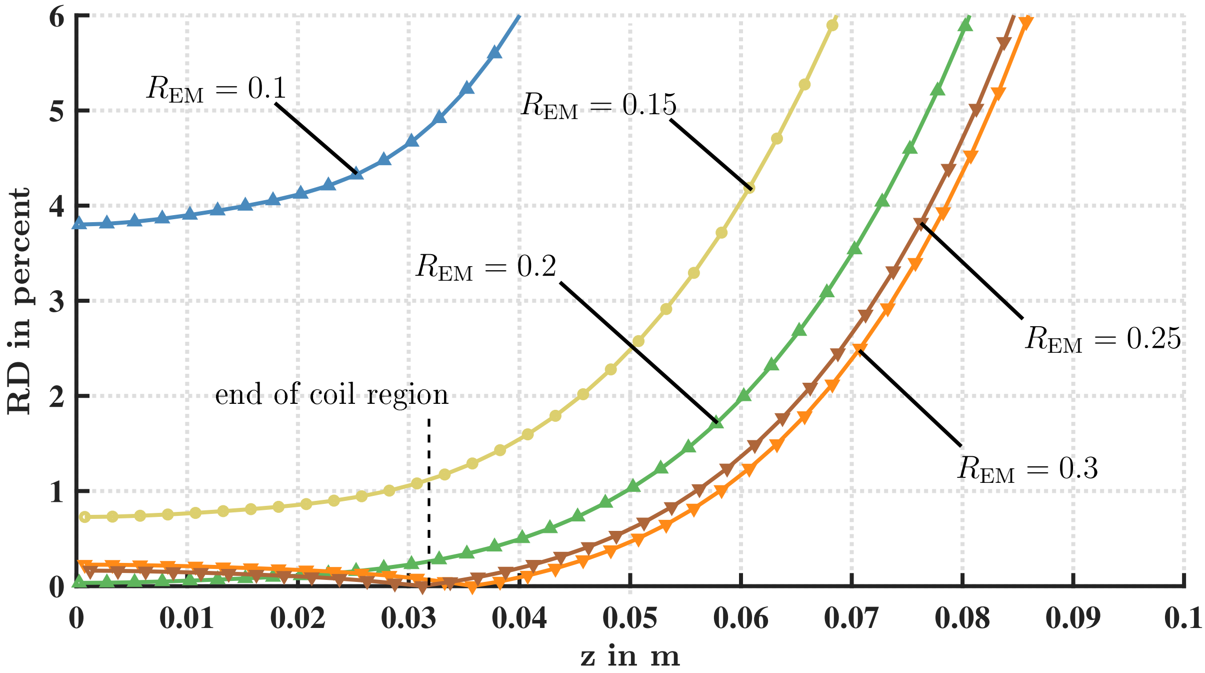
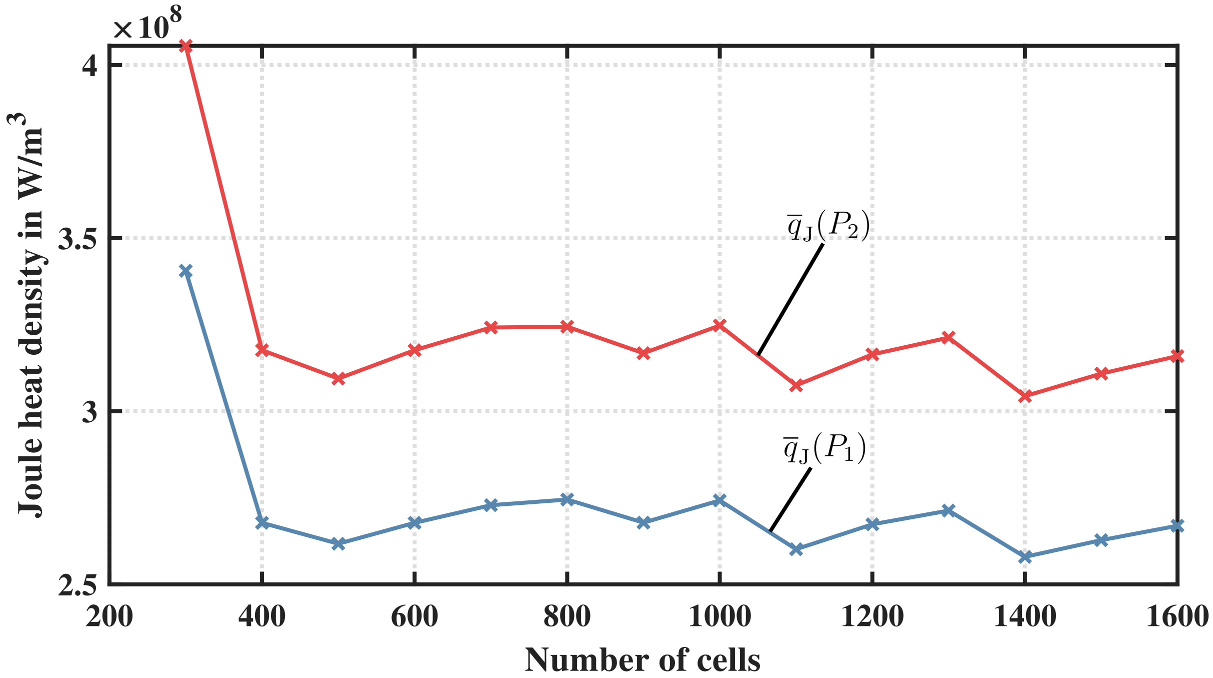


Figure 3.13.: Variation of the computational domain width, while maintaining constant cell size $\Delta x = 0.5\,\mathrm{mm}$, so that $R_{\mathrm{EM}} = 0.1$ is resolved with 100 cells, $R_{\mathrm{EM}} = 0.15$ with 150 cells and so on.

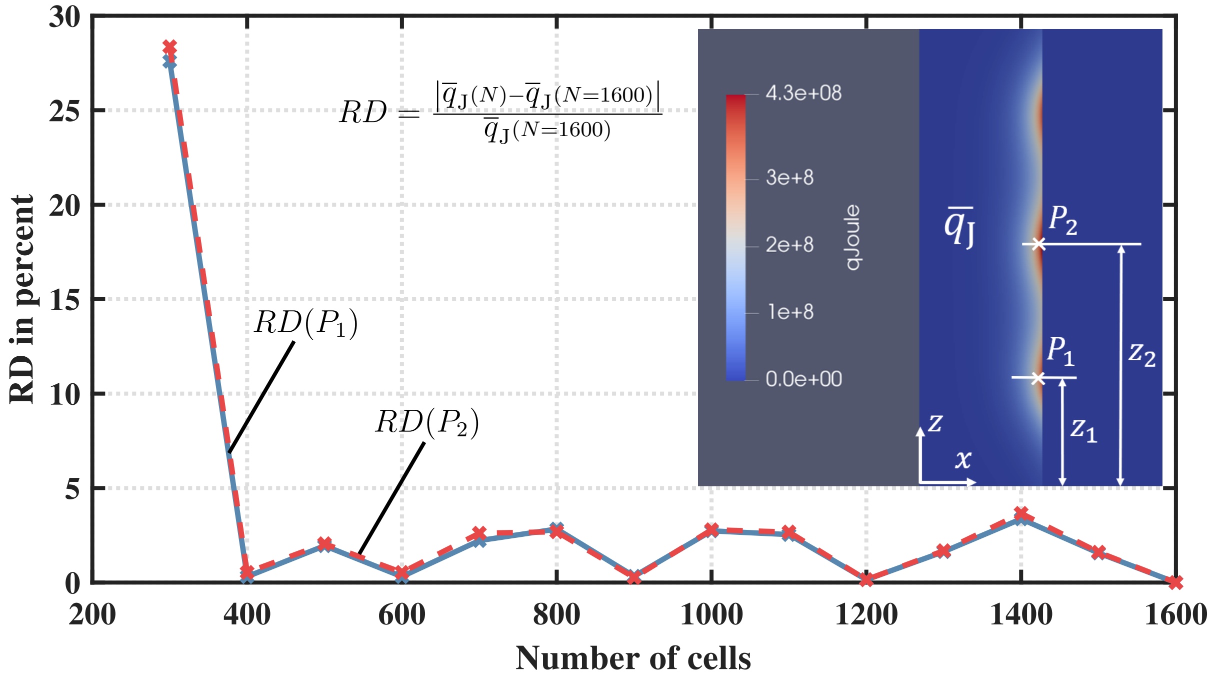
Induction coil ($\sigma = 2500$)

Since the behavior of the magnetic field changes completely when an electrically conductive material is involved, an additional mesh dependency analysis is required in which the induction effects are taken into account. Based on the previous results, a domain size of $R_{\mathrm{EM}} = 0.3\,\mathrm{m}$ and $L_{\mathrm{EM}} = 0.3\,\mathrm{m}$ is chosen for the study shown in Figure 3.14, which is similar to the second test case presented in section 3.2. In contrary, here the Joule heat density is evaluated in two specific points $P_1(x = 0.024, z = z_1)$ and $P_2(x = 0.024, z = z_2)$ as shown by the scheme included in Figure 3.14b, where z_1 and z_2 denote the center positions of the first and second (middle) coil turns, respectively. The two sample points are chosen 1 mm apart from the torch wall located within the skin depth. This region is particularly interesting, as 86% of the plasma power is dissipated within the skin depth and thus accurate values for the Joule heat density are required here. Figure 3.14a shows the Joule heat density at P_1 and P_2 evaluated for meshes with different total number of cells N varying from 300 to 1600 in steps of 100 cells. It can be noted that the calculated values for $\overline{q}_{\mathrm{J}}(P_1)$ and $\overline{q}_{\mathrm{J}}(P_2)$ stagnate at almost constant level except for $N = 300$. This becomes more clearer when looking at Figure 3.14b, which shows the relative difference with respect to the solution with the finest mesh $N = 1600$. Here, except for $N = 300$ the difference between calculated values for $\overline{q}_{\mathrm{J}}$ is less than 5% regardless of the cell density.

Conclusion. The results above show that the computational domain size should be chosen carefully depending on the region of interest and electromagnetic phenomena involved. The data also shows that a further reduction of the cell size below $\Delta x = 0.5\,\mathrm{mm}$ has no significant benefits in terms of accuracy. Finally, an electromagnetic mesh configuration of $R_{\mathrm{EM}} = 0.2\,\mathrm{m}$ at 400 cells and $L_{\mathrm{EM}} = 0.3\,\mathrm{m}$ at 600 cells is chosen for this work.



(a) The Joule heat density $\overline{q}_\mathrm{J}$ calculated for different cell numbers of the mesh and evaluated in two specific points $P_1(x = 0.024, z = z_1)$ and $P_2(x = 0.024, z = z_2)$, see scheme in the figure below



(b) Relative difference RD calculated for the results shown in the figure above with respect to the Joule heat density value of the finest mesh ($N = 1600$) of the corresponding point; N - number of cells

Figure 3.14.: Induction test case ($\sigma = 2500\,\mathrm{S\,m^{-1}}$) used to study the variation of the Joule heat density calculated with different cell numbers of the mesh; electromagnetic domain size used for these simulations $R_\mathrm{EM} = 0.3\,\mathrm{m}$ and $L_\mathrm{EM} = 0.3\,\mathrm{m}$

3.4.2. Convergence analysis

Typically, the convergence of a solution is determined by evaluating the residuals of the solved variables. Also, the residuals are a measure for the error of a solution. In order to achieve high numerical accuracy, the residuals should be very small. Considering a linear system of the form

$$\mathbf{\Psi}\mathbf{y} = \mathbf{b}, \tag{3.6}$$

with $\mathbf{\Psi}$ and $\mathbf{b}$ being a matrix and a vector with known coefficients, respectively, and $\mathbf{y}$ being the solution vector, then the residual can be defined as

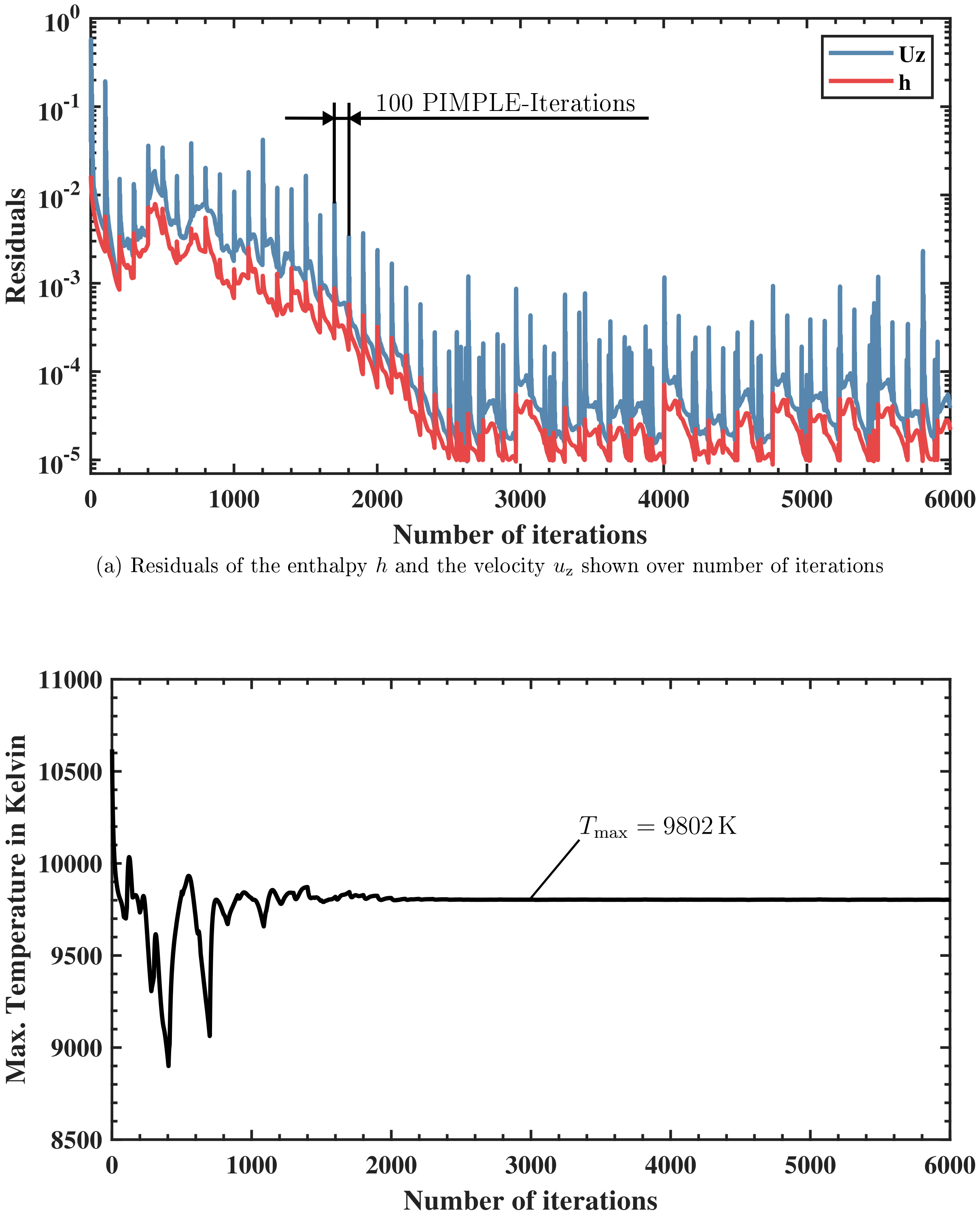
$$R^k = \left|\mathbf{\Psi}\widehat{\mathbf{y}}^k - \mathbf{b}\right|, \tag{3.7}$$

where R^k is the residual for the solution $\widehat{\mathbf{y}}^k$ after the k-th iteration. In addition, the solution residuals are usually normalized, however, the normalization procedure may be different for each solver [Ope19a]. In OpenFOAM the residuals are typically printed to the console during the execution of a solver in the form of

```
BiCGStab:  Solving for h, Initial residual = 1.924028e-05,
Final residual = 2.463816e-06, No Iterations 1
```

In the example shown above the linear system is solved using Biconjugate gradients stabilized (BiCGStab) method. The linear solver has finished after one iteration with a final residual of $2.463\,816 \times 10^{-6}$. In the case of the nonlinear and coupled plasma system the initial residual is rather of interest, as it is a measure of stability of the solution with respect to previous iteration of the total system solved. In the following, the steady-state solution is analyzed, which was obtained by running 100 *time.loop()* iterations with the *plasmaSteadyFoam* solver. In addition, the *pimple.loop()* was configured so that it runs until either the residual of the enthalpy h falls below 1×10^{-5} or the maximal number of 100 pimple iterations is reached. As a result, the number of iterations of the fluid system within the *pimple.loop()* being solved could reach a maximum of 100 000 total iterations.

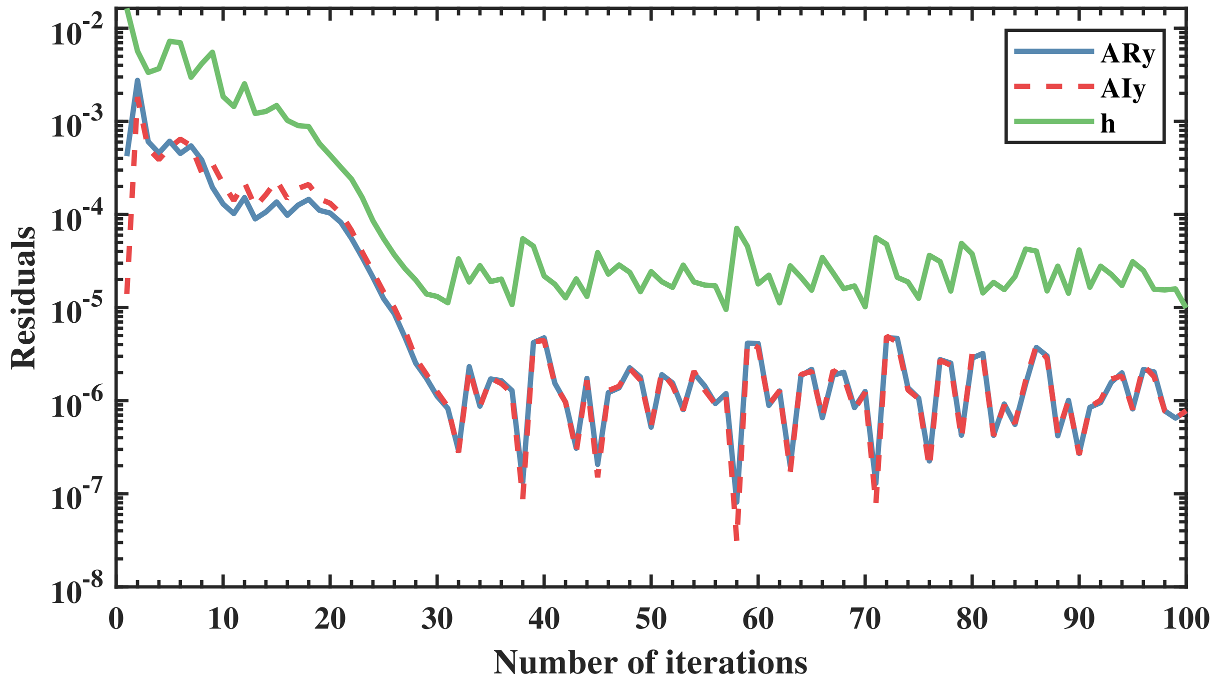
Figure 3.15a shows the residuals of the enthalpy h and the z-component of the fluid velocity u_z. In this case, the 100 iterations of the *time.loop()* correspond to a total of about 6000 iterations including the counts of the *pimple.loop()*. The residuals of u_z decreased from about 1×10^{-2} to a value oscillating between 1×10^{-3} and 2×10^{-5}, whereas the enthalpy residual shows less oscillation reaching a value of about 1×10^{-4} and 1×10^{-5}. When performing a steady-state analysis, it is also useful to monitor the behavior of quantities of interest. Therefore, the maximum plasma temperature changing over iterations as shown in Figure 3.15b is also considered for the convergence analysis. Here it can be clearly noted that after about 3000 iterations the maximum temperature converged to a value of 9802 K. Figure 3.16a shows the residuals of the vector potential and the enthalpy a function of the main loop iterations. In addition, the dissipated power, which is obtained from the volume integral of the Joule heat density $\overline{q}_\mathrm{J}$, is shown in Figure 3.16b. Based on this data it can be concluded that the simulation converged after about 30 main loop iterations with the dissipated plasma power converging to a value of $P_\mathrm{diss} \approx 5.1\,\mathrm{kW}$. The residuals of the enthalpy are oscillating between 1×10^{-5} and 5×10^{-5}, whereas the real and imaginary parts of the magnetic vector potential residuals maintain values below 1×10^{-5}. The oscillations may depend on many factors such as the selection of linear solvers, tolerance settings, relaxation factors. Also, it should be pointed out that the plasma model is a multi-physical and highly coupled system, which makes it very difficult to achieve stable residuals.



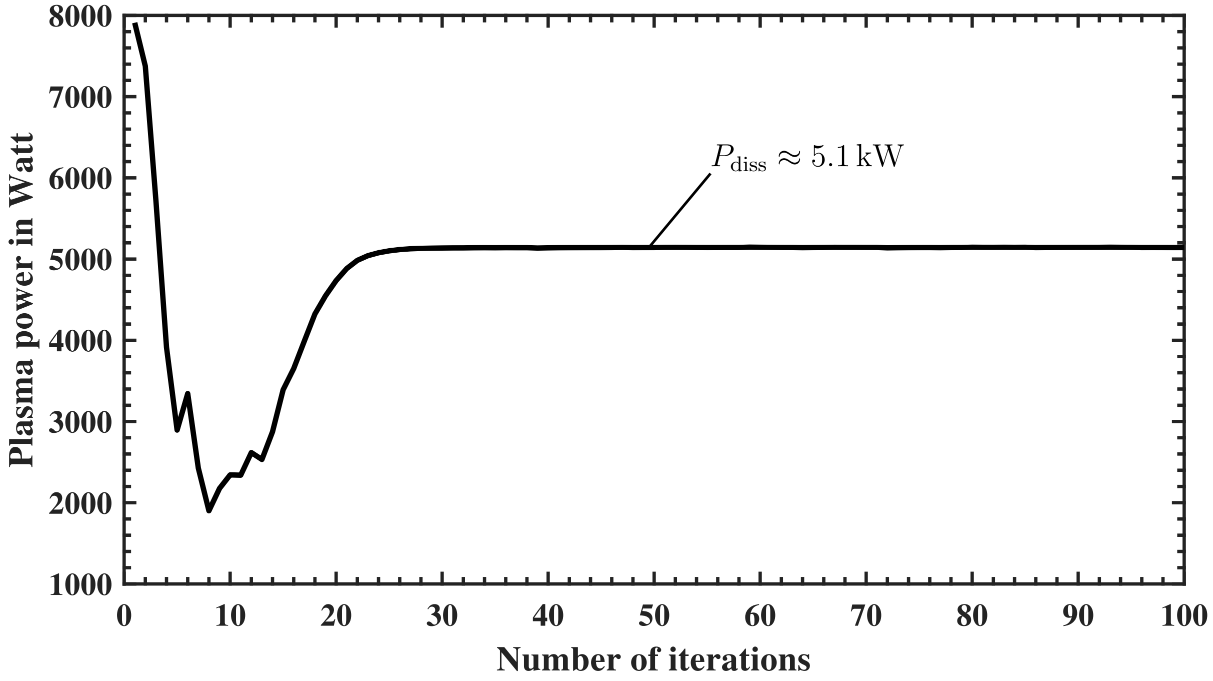
(a) Residuals of the enthalpy h and the velocity u_z shown over number of iterations

(b) Maximum plasma temperature shown over the number of iterations

Figure 3.15.: Convergence analysis based on data shown over the total number of iterations including *pimple.loop()*



(a) Residuals of the enthalpy (h) and the real and imaginary part of the magnetic vector potential ($\widetilde{\mathbf{A}}_{\mathrm{R,y}}$ and $\widetilde{\mathbf{A}}_{\mathrm{I,y}}$) shown over number of iterations



(b) Dissipated plasma power obtained by $P_{\mathrm{diss}} = 72 \cdot \int \overline{q}_{\mathrm{J}} dV_{\mathrm{wedge}}$ for the wedge-shaped computational domain with an angle of 5°

Figure 3.16.: Convergence analysis based on data shown over number of *time.loop()* iterations

3.5. Workstation Specifications

To conclude the plasma solver evaluation, the used soft- and hardware, as well as performance data is presented in the following. The data is summarized in Table 3.6. All implementations and simulations in this work have been conducted with foam-extend-4.0, however, it was found that post-processing with `sampleDict` utility works better on the OpenFOAM version v1812. The OpenFOAM framework is running on a virtual Linux system, where the actual hardware resources that could be utilized for computations consists of four logical cores with 3.2 GHz CPU speed each and 8 GB RAM. The total computation time for the simulation of the plasma steady-state including the calculation of the initial conditions is about two hours, whereas the transient solution for 1 s simulation time at a time step of 0.0005 s takes about 14 hours. Note, that all calculations are performed on a single core and that these are preliminary results. The potential has not been fully exploited yet and better solver performance may be achieved by tuning the solver and case settings.

Table 3.6.: Workstation specifications and solver performance

Workstation specifications		
Host system		Windows 7 Pro 64-bit, Intel i5 3.2 GHz processors, 4 physical cores, 8 logical cores, 12 GB RAM, 500 GB HDD
Virtual machine		Linux Mint 19.1 64-bit, 4 logical cores and 8 GB RAM allocated by the host system
OpenFOAM version		foam-extend-4.0, v1812 (post-processing only)
Mesh		**Total cell number**
Fluid mesh		19200 cells
Electromagnetic mesh		206500 cells
Solver		**Computation time**
rhoSimpleFoam	iterations = 2000	5.2 min
emHeatFoam	iterations = 350	70.9 min
plasmaSteadyFoam	iterations = 40	56 min
plasmaTransientFoam	end time = 1s, time step = 0.0005s	13.3 hours

In comparison, the simulations by means of the customized ANSYS Fluent framework, Figure 3.8a, were performed with the following specifications

- CPU: i7 , 6 cores parallel,
- unstructured mesh, triangular cells,
- fluid region: 24453 cells,
- whole computational domain: 82579 cells,

- transient simulation: $t_{\mathrm{sim}} = 3\,\mathrm{s}$, $\Delta t = 0.0005\,\mathrm{s}$
- computation time: 6 hours.

The present customized ANSYS code can only run in transient mode, whereas the OpenFOAM code has the ability to perform a steady-state simulation, which took about 1 hour in case of the Tekna plasma torch. Note, that precomputation of initial conditions is only necessary again if the geometry of the torch or the mesh configuration is changed. For example, simulations with varying operating parameters such as coil current, frequencies, and inlet velocities can be performed without a new precomputation of the initial conditions.

3.6. Experimental ICP (KNRTU-KAI)

In this last section, the simulation results for the experimental ICP at the KNRTU-KAI are presented and concepts for experimental validation are briefly discussed.

Figure 3.17 and Table 3.7 show the geometry and operating parameters used for the simulation of the experimental ICP torch, which in comparison to the Tekna-PL50 torch has 4 coil turns and a central tube, which is extended into the plasma zone. The KNRTU-KAI plasma torch is about twice the size of the Tekna-PL50 and operates therefore at a higher frequency and power. The third inlet is inactive as it was found that due to the design of torch the outer gas stream causes instabilities to the plasma. The steady-state solution of the temperature field is shown in Figure 3.18. In the simulation a dissipated plasma power of $P_{\mathrm{diss}} = 13\,\mathrm{kW}$ and a maximum temperature of 9940 K were obtained for a coil current of $I_{\mathrm{coil}} = 100\,\mathrm{A}$.

Unfortunately, neither the model output (temperature) nor the input (coil current) is currently available as experimental data, however, measurements from both quantities are needed in order to validate and improve the model predictions. Figure 3.19a shows a typical scheme of an electrical circuit of the radio-frequency (RF) power generator. The coil current I_{coil} is determined by the anode voltage U_{A}, and anode current I_{A}, which are manually set during the experiment and available as measurement signals. A linear correlation between anode current, anode voltage and coil current can be noted from Figure 3.17, which shows a series of measurements of the electrical quantities for the Tekna-PL50 torch. As a result, linear circuit theory can be used to derive a function $I_{\mathrm{coil}} = f(U_{\mathrm{A}}, I_{\mathrm{A}})$, which allows the estimation of the coil current based on measurements of U_{A} and I_{A}. The derivation of transfer functions for the electrical circuit and the estimation of the plasma inductance L_{plasma} and resistance R_{plasma} were also described in earlier studies [Dew+10],[TC13],[TC15]. A further study with more focus on the electrical power supply circuit modeling is therefore suggested.

The temperature and velocity distributions within the plasma torch can be measured using the enthalpy probe technique, and when combined with mass spectroscopy the concentration field can be also obtained [Jia+10]. Experiments using this method are described in [RSB95], [Dzu04]. It was noted by Dzur [Dzu04, p.55] that the temperature measurement inside the plasma torch fluctuates by about 6% yielding an error of ±500 K. Another drawback of the enthalpy probe technique is that measurements can be only carried out point-wise. Also, this method may not be suited to monitor the temperature during the process of powder spheroidization.

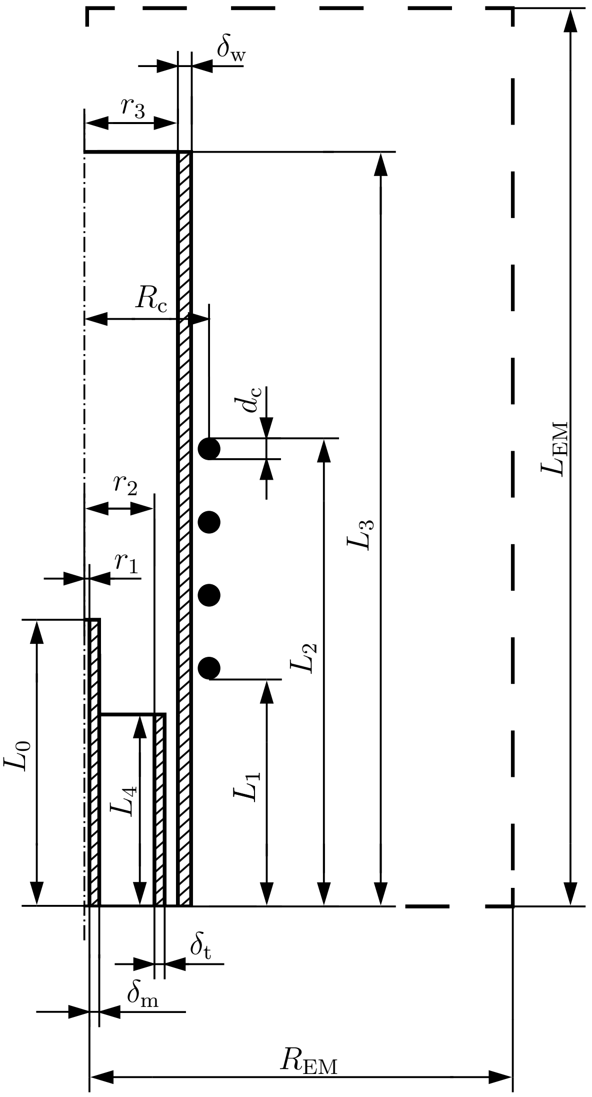


Figure 3.17.: Sketch of the experimental ICP torch (KNRTU-KAI); dimensions of the sketch are not true to scale

Table 3.7.: Plasma torch characteristics (KNRTU-KAI)

Parameter	Value	Unit	Parameter	Value	Unit
r_1	3.7	mm	u_1	1.15	$\mathrm{m\,s^{-1}}$
r_2	35	mm	u_2	0.07	$\mathrm{m\,s^{-1}}$
r_3	42.5	mm	u_3	0	$\mathrm{m\,s^{-1}}$
δ_{m}	1.3	mm	P_{diss}	13	kW
δ_{t}	2.5	mm	f	5.28	MHz
δ_{w}	2.5	mm	I_{coil}	100	A
L_0	251	mm	$\mathbf{J}_{\mathrm{coil}}$	1989437	$\mathrm{A\,m^{-2}}$
L_1	221	mm	d_{c}	8	mm
L_2	289	mm	R_{c}	58	mm
L_3	368	mm	R_{EM}	300	mm
L_4	90	mm	L_{EM}	510	mm

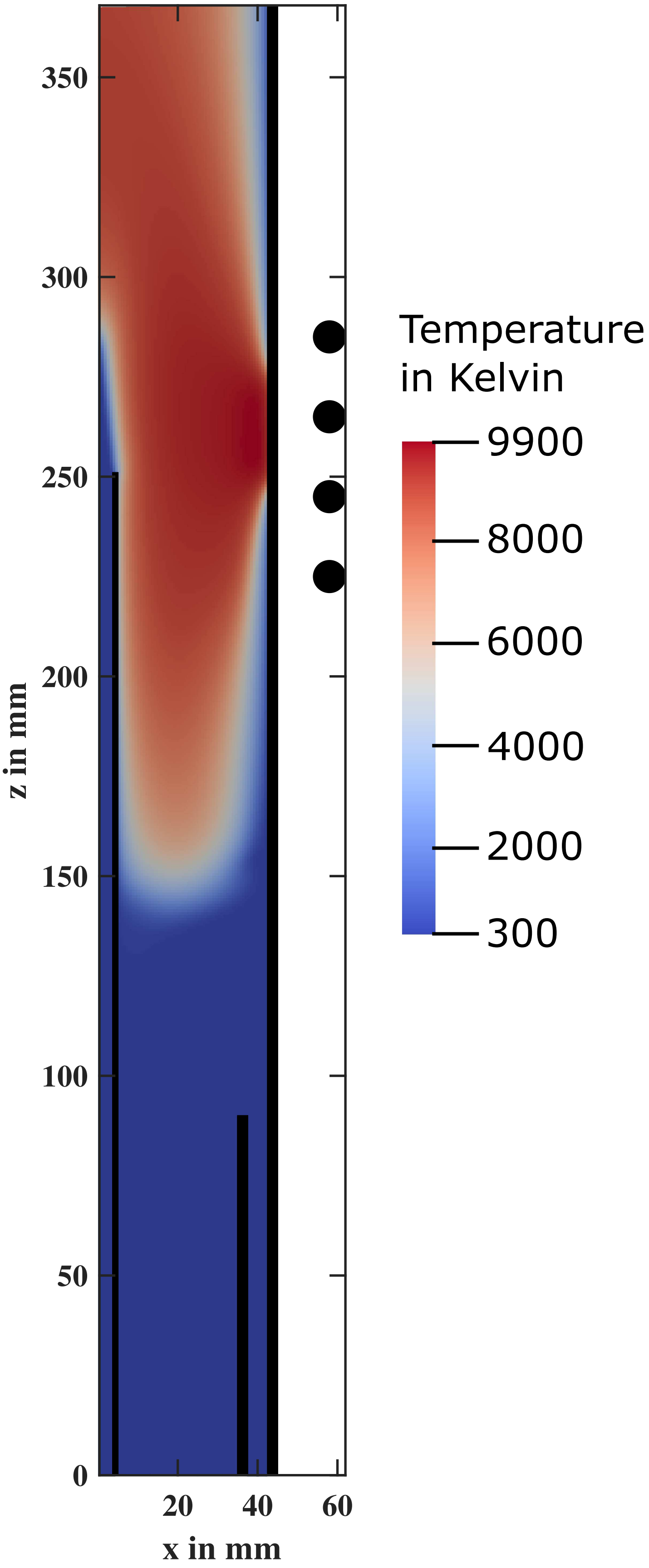
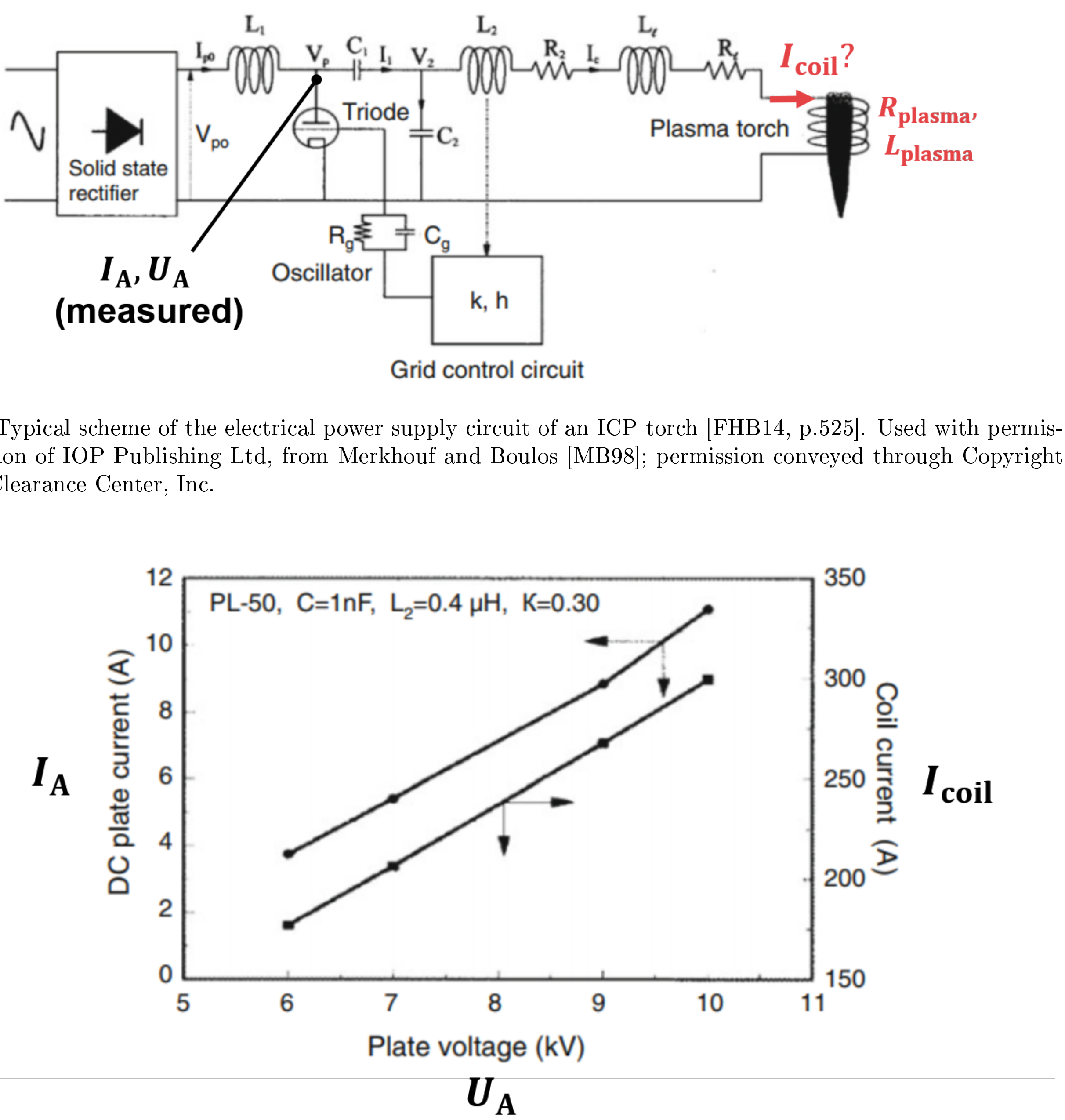


Figure 3.18.: Temperature field of the experimental ICP torch (KNRTU-KAI) obtained with *plasmaSteadyFoam*



(a) Typical scheme of the electrical power supply circuit of an ICP torch [FHB14, p.525]. Used with permission of IOP Publishing Ltd, from Merkhouf and Boulos [MB98]; permission conveyed through Copyright Clearance Center, Inc.

(b) Measured data from the generator circuit of the Tekna-PL50 plasma torch, where the anode current I_{A} and coil current I_{coil} are shown as a function of the anode voltage U_{A} [FHB14, p.525]. Used with permission of IOP Publishing Ltd, from Merkhouf and Boulos [MB98]; permission conveyed through Copyright Clearance Center, Inc.

Figure 3.19.: Coil current estimation can be achieved based on the measured anode voltage and anode current by deriving a transfer function $I_{\text{coil}} = f(U_{\text{A}}, I_{\text{A}})$ using electric circuit theory

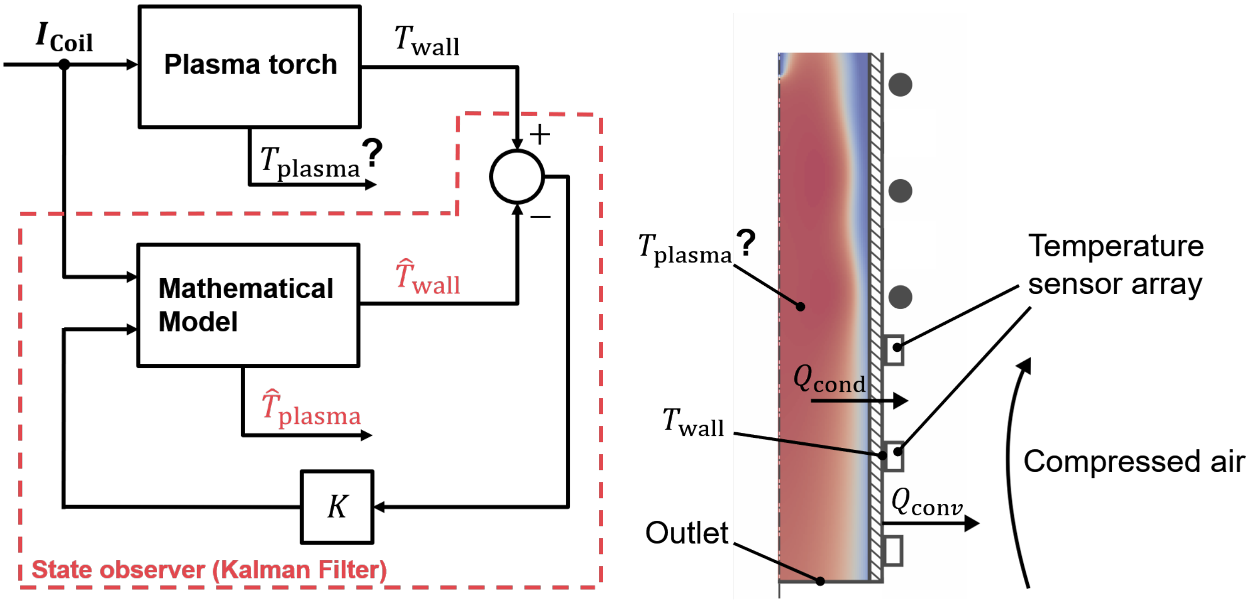


Figure 3.20.: Concept for the estimation of the internal plasma temperature based on measurements of the torch wall temperature

The group of Hieftje et al. described an experimental setup [Hua+92] for a spectroanalytical ICP, where a spatial temperature map was measured by means of Thomson and Rayleigh scattering. While Thomson scattering can be used to determine the electron density and electron temperature, Rayleigh scattering allows the measurement of the density and temperature of atoms and ions (heavy particles) [Nij19]. In [Yan+00] Hieftje et al. impressively demonstrated the agreement between the measured and simulated temperature maps, which were calculated using a two-temperature plasma model for the spectroanalytical ICP. However, for the purpose of in-situ process control for spheroidization, a concept is needed, which allows the real-time estimation of the plasma temperature and can be realized without major modifications of the experimental setup.

Therefore, the following novel concept illustrated in Figure 3.20 is proposed. The idea is to estimate the internal temperature T_{plasma} by using a state observer based on measurements of the external wall temperature T_{wall}. For example, a common implementation of a state observer is the Kalman-Filter, which is an optimal estimation algorithm. Assuming that an accurate estimate or measurement of the coil current is available as an input signal, the mathematical model is used to estimate the internal plasma temperature $\widehat{T}_{\text{plasma}}$ and the external wall temperature $\widehat{T}_{\text{wall}}$. The Kalman-Filter will try to eliminate the difference between the measured and the estimated wall temperature. If both temperatures are equal, then the model will converge to the real system, so that the estimated internal plasma temperature will converge to its true value. In order to calculate the wall temperature, which at the same time is measured by a sensor array, the conductive and convective heat transfer at the quartz wall needs to be considered in the ICP model. However, the main question here is whether the temperature sensors, for example, thermocouples, can be positioned in the environment of the strong electromagnetic fields of the coil. To prevent the destruction of the sensors, it may be possible to find measurement points near the outlets or inlets of the torch, where the sensors are positioned at a sufficient distance, with shielding if necessary, from the coil. Further research should be undertaken to investigate the feasibility of this concept.

CHAPTER 4

Conclusion

4.1. Summary

The goal of this work was to study the free and open-source CFD framework OpenFOAM and develop a customized code to simulate the inductively coupled plasma torch. Therefore, a standard two-dimensional axisymmetric modeling approach has been used. A major challenge was the implementation of the electromagnetic solver and the couplings between the fluid and electromagnetic domains, but this has been successfully resolved due to the flexibility and customizability of OpenFOAM. The implementation of the electromagnetic solver has been validated by analytical methods and the plasma simulation results show good agreement in comparison with literature data.

In the following, the achievements of this work are summarized:

- The present work contributes a new functionality to solve electromagnetic problems in the OpenFOAM framework. It is based on known theory about the vector potential formulation of Maxwell's equations together with the frequency domain method in order to efficiently simulate cases that involve high-frequency electromagnetic fields.
- A plasma modeling approach based on fundamental physical principles has been implemented providing a useful tool to investigate the behavior of plasma torches under different operating conditions. Due to the ability to compute steady-state solutions fast (less than 1 hour on a single-core processor), the present *plasmaSteadyFoam* code can already be used to run parameter studies, for example, with varying coil current, frequency or inlet velocities.
- The potential of the OpenFOAM framework has been demonstrated by exploiting features such as: the block coupled matrix solver library to implement a fully coupled solver for the magnetic vector potential matrix; a multi-mesh technique to separately compute the equations of the electromagnetic and fluid domain; the implementation of a customized thermodynamic model, which includes the tabulated physical properties

of argon. Since the OpenFOAM code is based on the Finite-Volume-Method, the solvers implemented in this work are not limited to two-dimensional problems and therefore 3D-cases can be studied as well. Furthermore, the provided methods may be applied to simulate other processes such as tungsten inert gas (TIG) welding, induction heating, and other magnetohydrodynamic problems.

4.2. Outlook

Further modeling work will have to be conducted in order to numerically investigate the spheroidization of powders and to determine optimal operating conditions for the ICP torch. The transient version of the plasma solver *plasmaTransientFoam* provides the basis for further studies and therefore the following next steps are proposed for future work:

- A major task is to improve the estimation of the internal plasma temperature. Experimental data obtained either by optical methods or using dedicated measuring points outside the plasma torch can help to improve the model predictions by comparing the estimated and measured temperature at the reference points. Therefore, the present model needs to be modified to match the experimental setup at KAI. One task here could be to replace the constant temperature boundary conditions at the quartz walls by a custom heat flux condition, which considers the convective cooling of the inner tube and outer tube walls.
- The present model requires the coil current as an input, however, it is not available as a measurement signal. Therefore, another task could be to derive a model of the power supply circuit in order to obtain an estimate of the coil current based on measurements of the anode voltage and anode current.
- Further work is needed to determine the influence of 3D and turbulence effects on the results. The present code is already capable of performing these types of simulation.
- The final step is to integrate a suited powder particle model and study the spheroidization process at varying operating parameters such as input power, flow rates and the central tube height.

Overall, this work shows that the OpenFOAM framework is a real alternative to commercial software providing a flexible tool for multi-physical and application-specific modeling. Finally, this work would have been impossible without the open-source nature of this platform and the numerous contributions of the research community. Therefore, it can only be encouraged to continue to contribute to this framework in order to further strengthen its capabilities, so that everyone can benefit from it.

List of Figures

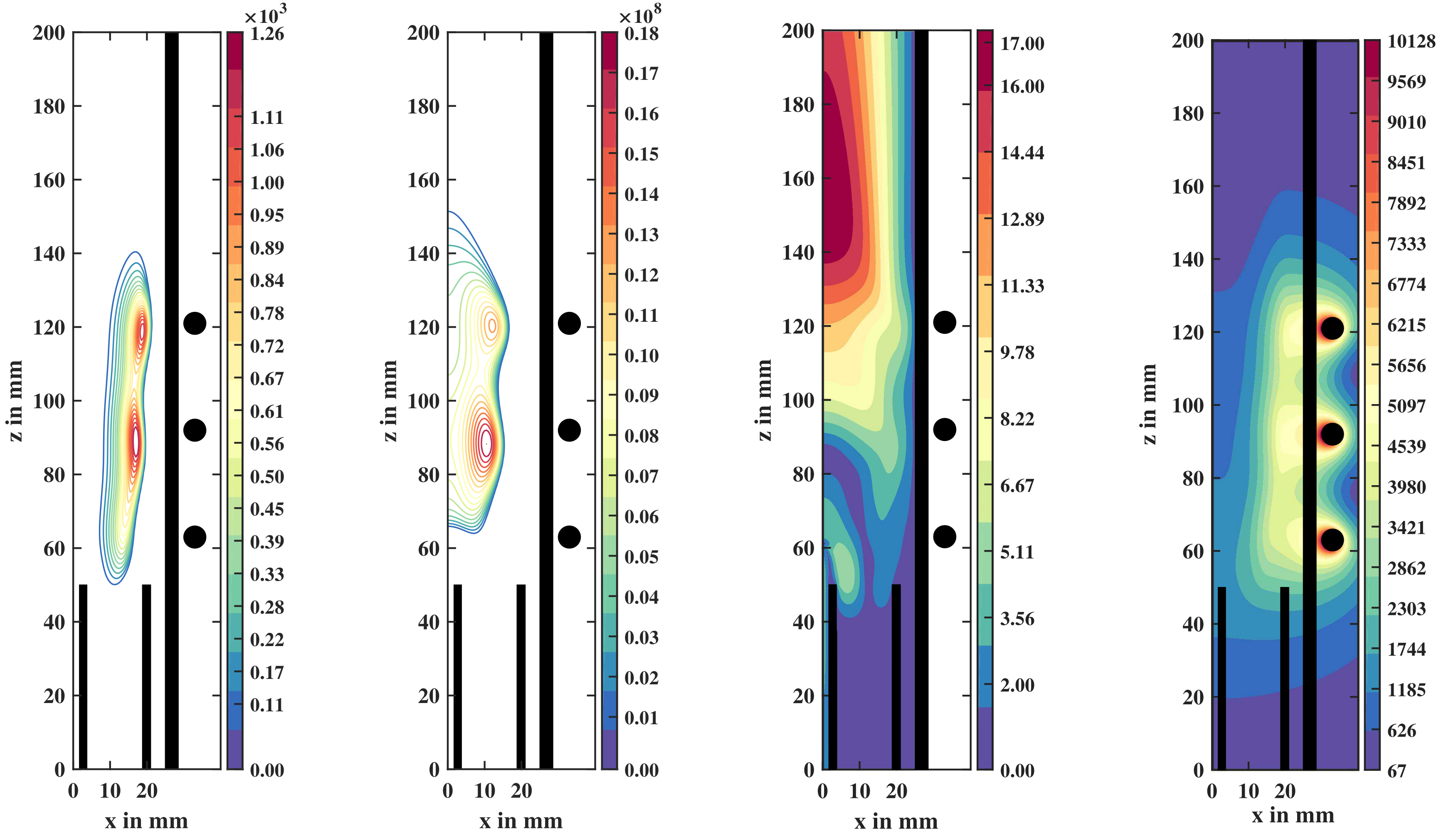
List of Tables

APPENDIX A

Supplementary Simulation Output for TEKNA-PL50

Not all physical quantities, which can be calculated by the plasma solver, have been discussed in this work. For reference or comparison in future studies the following additional results calculated for the plasma torch Tekna-PL50 are presented:

- Contours of the Lorentz force density $\overline{\mathbf{f}}_{\mathrm{L}}$, Figure A.1a,
- Contours of the radiation loss q_{R}, Figure A.1b,
- Contour map of the gas velocity magnitude $|\mathbf{u}|$, Figure A.1c,
- Contour map of the magnetic field strength magnitude H_{mag}, Figure A.1d,
- Distribution of the argon properties: electrical conductivity σ, thermal conductivity k, mass density ρ, specific heat capacity c_{p}, and dynamic viscoity μ; Figure A.2.



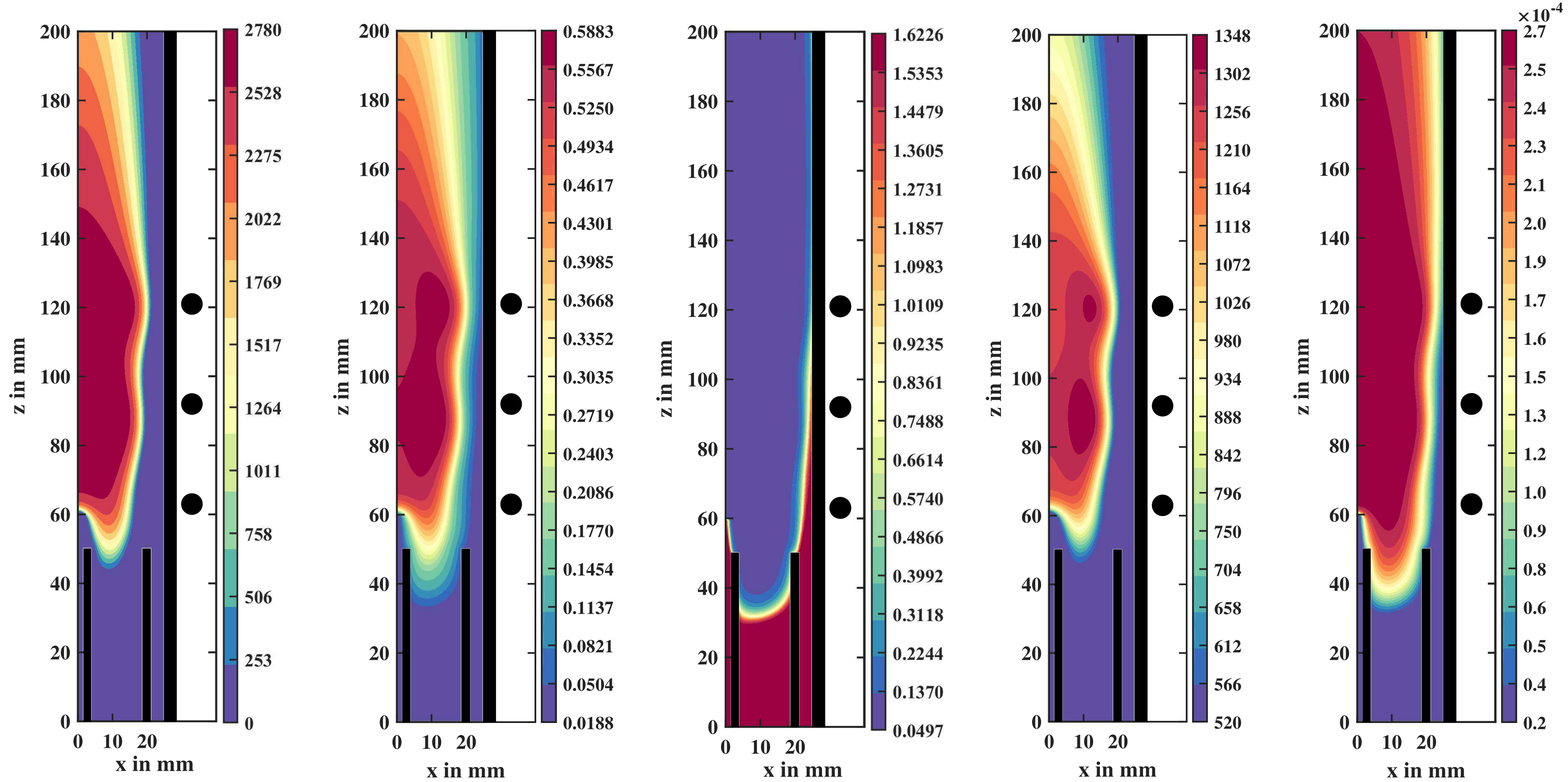
(a) Lorentz-force density in $\mathrm{N\,m^{-3}}$

(b) Radiation loss density in $\mathrm{W\,m^{-3}}$

(c) Gas velocity magnitude in $\mathrm{m\,s^{-1}}$

(d) Magnitude of the magnetic field strength in $\mathrm{A\,m^{-1}}$

Figure A.1.: Simulation results obtained with the custom-developed *plasmaSteadyFoam* solver for the Tekna-PL50 plasma torch



(a) Electrical conductivity in $\mathrm{S\,m^{-1}}$

(b) Thermal conductivity in $\mathrm{W\,m^{-1}\,K^{-1}}$

(c) Mass density in $\mathrm{kg\,m^{-3}}$

(d) Specific heat capacity in $\mathrm{J\,kg^{-1}\,K^{-1}}$

(e) Dynamic viscosity in $\mathrm{kg\,m^{-1}\,s^{-1}}$

Figure A.2.: Argon properties calculated with the *plasmaSteadyFoam* solver for the Tekna-PL50 plasma torch

APPENDIX B

Test Case: Half-Space Model

In section 3.2 it was mentioned that an exact analytical solution of the magnetic field induced into a cylindrical geometry is difficult to obtain. In contrary, the so-called *half-space problem* is a simple model that allows to calculate exact analytical solutions of the magnetic field and the resulting quantities such as Lorentz force and Joule heat density. This test case is well-known from electrodynamic theory; a detailed description can be found, for example, in the text book *An Introduction to Magnetohydrodynamics* [Dav01, p.391]. Therefore, a variant of an electromagnetic solver is presented in this section, which in contrary to *emFoam* is not based on the magnetic vector potential. Instead only the magnetic flux density **B** is solved based on the specified boundary conditions. The implemented solver, named *BDiffFoam*, is then applied to investigate the half-space problem.

B.1. Governing equation

Consider a time-varying magnetic field $\mathbf{B}(t) = B_0 \cos(\omega t)\mathbf{e}_\mathrm{y}$ applied at the surface of a conductive region the so-called half-space, as shown in Figure B.1. To derive the governing differential equation we start by applying the curl operator on both sides of Ampere's law

$$\boldsymbol{\nabla} \times \mathbf{H} = \mathbf{J}, \tag{B.1}$$

to obtain the expression

$$\boldsymbol{\nabla} \times (\boldsymbol{\nabla} \times \mathbf{H}) = \boldsymbol{\nabla} \times \mathbf{J}. \tag{B.2}$$

Assuming $\mu_0 = \mathrm{const}$ and substitution with $\mathbf{J} = \sigma \mathbf{E}$ and $\mathbf{B} = \mu_0 \mathbf{H}$ yields

$$\frac{1}{\mu_0} \boldsymbol{\nabla} \times (\boldsymbol{\nabla} \times \mathbf{B}) = \sigma \cdot \boldsymbol{\nabla} \times \mathbf{E}, \tag{B.3}$$

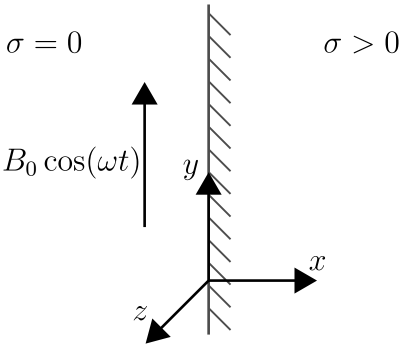


Figure B.1.: Half-space problem

Further, Equation (B.3) can be rewritten using Faraday's law $\boldsymbol{\nabla} \times \mathbf{E} = -\frac{\partial \mathbf{B}}{\partial t}$ and the mathematical rule $\boldsymbol{\nabla} \times (\boldsymbol{\nabla} \times \mathbf{B}) = \boldsymbol{\nabla}(\boldsymbol{\nabla} \cdot \mathbf{B}) - \boldsymbol{\Delta}\mathbf{B}$ as

$$\boldsymbol{\nabla}(\boldsymbol{\nabla} \cdot \mathbf{B}) - \boldsymbol{\Delta}\mathbf{B} = -\mu_0 \sigma \frac{\partial \mathbf{B}}{\partial t}. \tag{B.4}$$

Applying Gauss' law for magnetic fields $\boldsymbol{\nabla} \cdot \mathbf{B} = 0$, finally yields

$$\boldsymbol{\Delta}\mathbf{B} = \mu_0 \sigma \frac{\partial \mathbf{B}}{\partial t}. \tag{B.5}$$

Since we are only interested in the time-averaged solutions of the Lorentz force and the Joule heat inside the conductive region, the Fourier transform of Equation (B.5) is considered in the following, which is given by

$$\boldsymbol{\Delta}\widetilde{\mathbf{B}} = j\omega\mu_0\sigma\widetilde{\mathbf{B}}, \tag{B.6}$$

B.2. Analytical solution

To obtain the analytical solution for the magnetic field, the vector differential equation (B.6) is further simplified to one equation, since the outer magnetic field is only applied in y-direction. Using substitution $\gamma^2 = j\omega\mu_0\sigma$ Equation (B.6) is rewritten in one-dimensional form as

$$\boldsymbol{\Delta}\widetilde{B}_{\mathrm{y}} - \gamma^2 \widetilde{B}_{\mathrm{y}} = 0, \tag{B.7}$$

which represents a typical parabolic partial differential equation with the corresponding characteristic equation

$$\lambda^2 - \gamma^2 = 0. \tag{B.8}$$

The general solution of Equation (B.8) is given by

$$\widetilde{B}_{\mathrm{y}} = C_1 e^{\gamma x} + C_2 e^{-\gamma x}, \tag{B.9}$$

where the constants C_1 and C_2 are determined by evaluation of the boundary conditions

$$\begin{aligned} \widetilde{B}_{\mathrm{y}}(x \rightarrow \infty) = 0 \rightarrow C_1 = 0, \\ \widetilde{B}_{\mathrm{y}}(x = 0) = B_0 \rightarrow C_2 = B_0. \end{aligned} \tag{B.10}$$

Thus, we obtain the final solution

$$\widetilde{B}_{\mathrm{y}} = B_0 e^{-\gamma x}. \tag{B.11}$$

By back-substitution of γ and using the identity

$$\sqrt{j} = \sqrt{\frac{(1-j)^2}{2}} = \frac{1-j}{\sqrt{2}} \tag{B.12}$$

Equation (B.11) can be rewritten as

$$\begin{aligned} \widetilde{B}_{\mathrm{y}} &= B_0\, e^{-\gamma x} \\ &= B_0\, e^{-\sqrt{j\omega\mu_0\sigma}x} \\ &= B_0\, e^{-(1-j)\,x/\delta} \\ &= \underbrace{B_0\, e^{-x/\delta}}_{:=B_{\mathrm{mag}}} e^{jx/\delta}, \end{aligned} \tag{B.13}$$

with the skin depth δ given by

$$\delta = \sqrt{\frac{2}{\omega\mu_0\sigma}}. \tag{B.14}$$

It can be noted from the analytical solution (B.13) that the magnitude of the magnetic field is a decaying exponential function

$$B_{\mathrm{mag,analytical}} = B_0\, e^{-x/\delta}, \tag{B.15}$$

which is also typical for diffusion processes. In the following we will compare the analytical solution of the magnitude of the magnetic flux density (B.15) against the numerical solution obtained by OpenFOAM.

B.3. Numerical solution

Similarly as in Section 2.3.2, handling of complex numbers is avoided by using the following identity

$$\widetilde{\mathbf{B}} = \widetilde{\mathbf{B}}_\mathrm{R} + j\widetilde{\mathbf{B}}_\mathrm{I}, \tag{B.16}$$

which is substituted into Equation (B.6) to obtain

$$\boldsymbol{\Delta}\widetilde{\mathbf{B}}_\mathrm{R} + j\boldsymbol{\Delta}\widetilde{\mathbf{B}}_\mathrm{I} = j\omega\mu_0\sigma\widetilde{\mathbf{B}}_\mathrm{R} - \omega\mu_0\sigma\widetilde{\mathbf{B}}_\mathrm{I}, \tag{B.17}$$

followed by separation of the real and imaginary parts in order to obtain

$$\begin{aligned}\boldsymbol{\Delta}\widetilde{\mathbf{B}}_\mathrm{R} &= -\omega\mu_0\sigma\widetilde{\mathbf{B}}_\mathrm{I},\\ \boldsymbol{\Delta}\widetilde{\mathbf{B}}_\mathrm{I} &= \omega\mu_0\sigma\widetilde{\mathbf{B}}_\mathrm{R}.\end{aligned} \tag{B.18}$$

The implementation of a fully coupled solver by means of the `fvBlockMatrix.C` library in foam-extend is based on the matrix representation of (B.18), which is written as

$$\begin{bmatrix} \frac{1}{\mu_0}\boldsymbol{\Delta} & \sigma\omega \\ -\sigma\omega & \frac{1}{\mu_0}\boldsymbol{\Delta} \end{bmatrix} \begin{bmatrix} \widetilde{\mathbf{B}}_\mathrm{R} \\ \widetilde{\mathbf{B}}_\mathrm{I} \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix}. \tag{B.19}$$

After the system (B.19) is solved, the magnitude of the magnetic flux density is finally obtained by

$$B_\text{mag,numerical} := \left|\widetilde{\mathbf{B}}\right| = \sqrt{\left|\widetilde{\mathbf{B}}_\mathrm{R}\right|^2 + \left|\widetilde{\mathbf{B}}_\mathrm{I}\right|^2}. \tag{B.20}$$

B.4. Comparison and validation

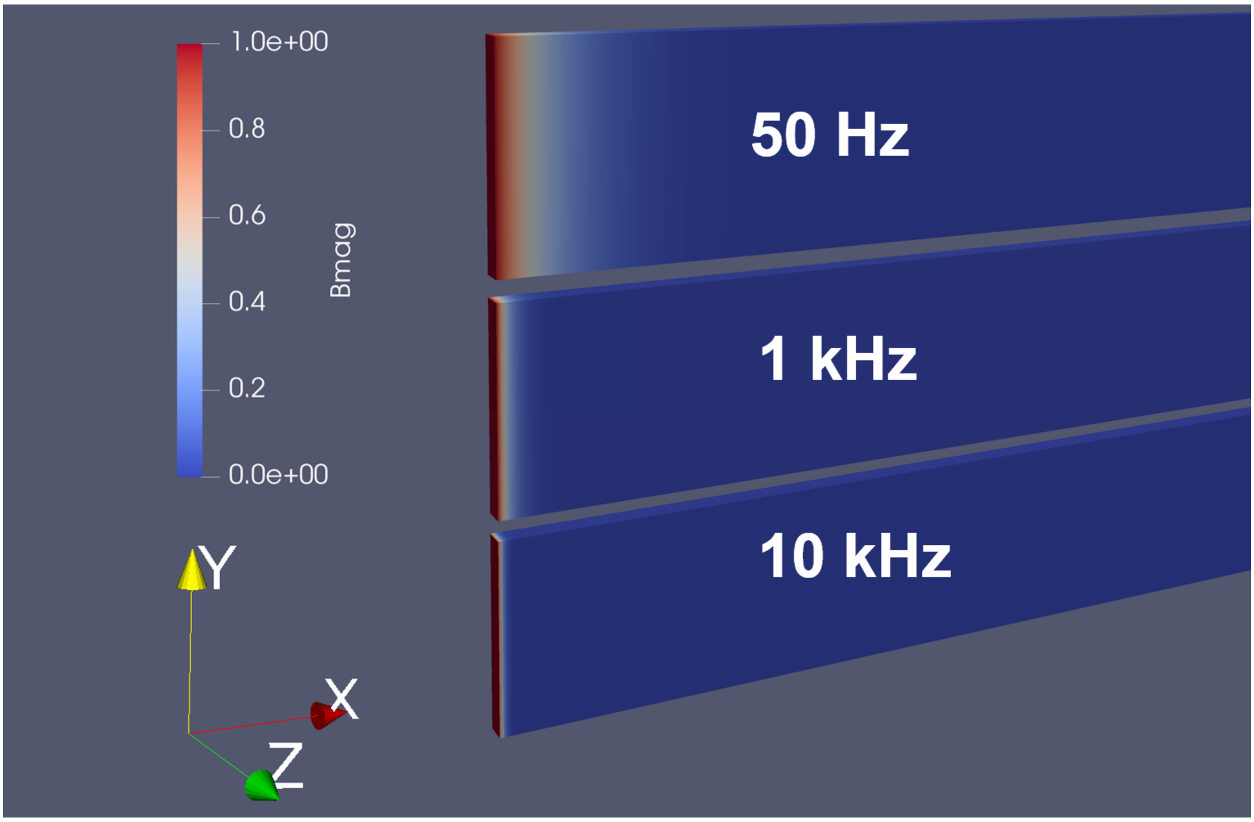
The test case has been simulated with $B_0 = 1\,\mathrm{T}$, $\sigma = 1 \times 10^7\,\mathrm{S\,m^{-1}}$ at the frequencies $f = [50\,\mathrm{Hz}, 1\,\mathrm{kHz}, 10\,\mathrm{kHz}]$; results are shown in Figure B.2. It can be seen from Figure B.2b that the numerical and analytical solution of the magnetic flux density are in agreement. In addition, it is also possible to compare further quantities such as the Joule heat and Lorentz force, for which the following analytical solutions can be derived, see also [Dav01, p.391-392],

Joule heat density:

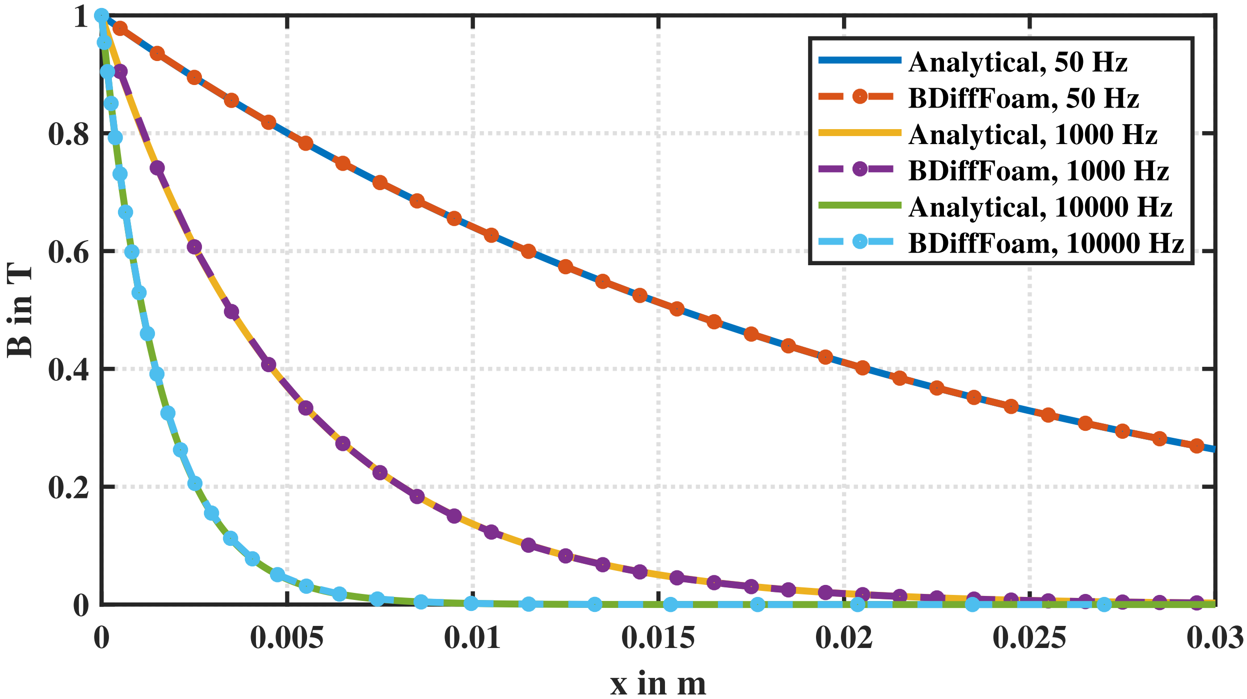
$$\overline{q}_\mathrm{J}(x) = \frac{B_0^2\omega}{2\mu_0}e^{-2x/\delta}, \tag{B.21}$$

Joule heat per unit surface area:

$$\int\limits_0^\infty \overline{q}_\mathrm{J} dx = \frac{B_0^2}{4\mu_0}\delta\omega, \tag{B.22}$$



(a) Post-processing of the OpenFOAM solution in paraview: Showing the half-space solution of the magnetic flux density for different frequencies



(b) Numerical solution compared against the analytical solution obtained with Equation (B.15)

Figure B.2.: Magnetic flux density B_{mag} computed with the *BDiffFoam* solver at frequencies $f = [50\,\mathrm{Hz}, 1\,\mathrm{kHz}, 10\,\mathrm{kHz}]$

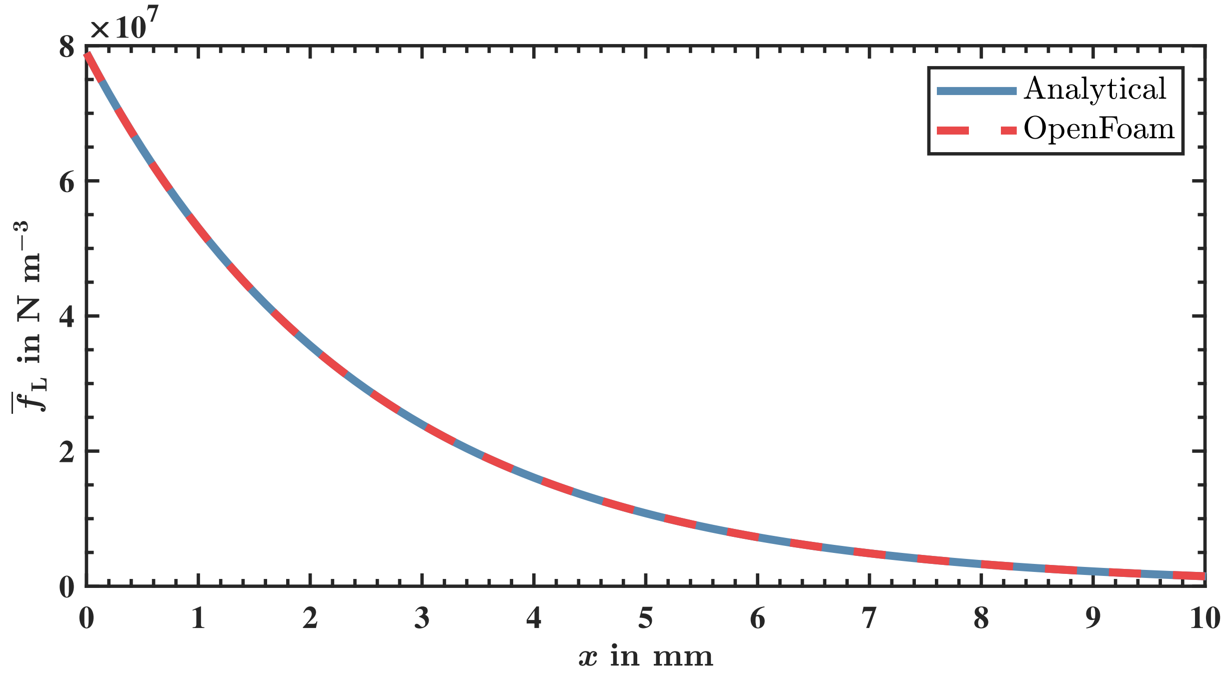


Figure B.3.: Lorentz force density plotted along the x-Axis; comparison of the numerical and analytical solution at frequency $f = 1\,\mathrm{kHz}$

Lorentz force density:

$$\bar{\mathbf{f}}_\mathrm{L}(x) = \frac{B_0^2}{2\mu_0\delta} e^{-2x/\delta} \mathbf{e}_\mathrm{x}, \tag{B.23}$$

Lorentz force per unit surface area:

$$\int_0^\infty \bar{\mathbf{f}}_\mathrm{L} dx = \frac{B_0^2}{4\mu_0} \mathbf{e}_\mathrm{x}. \tag{B.24}$$

For example, the analytical solution of the Lorentz force density given by Equation (B.23), can be compared against the numerical solution, which is obtained by the *BDiffFoam* code by

$$\bar{\mathbf{f}}_\mathrm{L} = \frac{1}{2}\left[\left(\widetilde{\mathbf{J}}_\mathrm{R} \times \widetilde{\mathbf{B}}_\mathrm{R}\right) + \left(\widetilde{\mathbf{J}}_\mathrm{I} \times \widetilde{\mathbf{B}}_\mathrm{I}\right)\right], \tag{B.25}$$

where the real and imaginary parts of the induced current density are obtained by

$$\begin{aligned} \widetilde{\mathbf{J}}_\mathrm{R} &= \frac{1}{\mu_0} \boldsymbol{\nabla} \times \widetilde{\mathbf{B}}_\mathrm{R}, \\ \widetilde{\mathbf{J}}_\mathrm{I} &= \frac{1}{\mu_0} \boldsymbol{\nabla} \times \widetilde{\mathbf{B}}_\mathrm{I}. \end{aligned} \tag{B.26}$$

The agreement of numerical and analytical solutions for $f = 1\,\mathrm{kHz}$ is shown in Figure B.3. Finally, the integration of the Lorentz force along the x-Axis yields 198 413 N m^{-2} (analytical) and 198 395 N m^{-2} (OpenFOAM), which is a relative difference of about 0.009%.

APPENDIX C

Notes on OpenFOAM

OpenFOAM is based on C++ programming language and comes without graphical user interface. Simulation cases are constructed in the form of script files, which are then executed from the command line. For post-processing the open-source visulization tool ParaView is commonly used.

Case setup. The structure of a typical case is shown in Figure C.1. A basic simulation case consists mainly of three subdirectories: the `0`, `constant`, and `system` folder. In the `system` directory the main simulation parameters such as solver settings, simulation time, and discretization schemes are stored in the three files `controlDict`, `fvSchemes`, and `fvSolutions`. The `constant` directory contains the mesh information, which is stored in the `polyMesh` subdirectory. The generation of meshes will be briefly described in the following section. In addition, the `constant` directory can be used to store files, which contain information about the transport properties or other needed physical constants, for example, the operating frequency of the ICP torch, and the magnetic permeability in vacuum have been defined here. Finally, the initial and boundary conditions for the field variables are specified in the `0` directory. When executing the solver, time directories are created automatically to store the solution of the field variables with respect to the time step and write interval specifications made in the `controlDict` file.

Mesh generation. There are different ways to create a mesh in OpenFOAM, the following list some of the possibilities, see also [CFD18b]:

- blockMesh: The blockMesh utility is the most generic way to generate simple meshes consisting of hexahedral cells.
- snappyHexMesh: This utility is suited for generating complex meshes of hexahedral and split-hexahedral cells automatically from triangulated surface geometries (e.g., .STL files) [Ope19b]
- external mesh tools: e.g., the open-source mesh generator Gmsh [GR09]; also, it is possible to import ANSYS Fluent meshes to OpenFOAM.

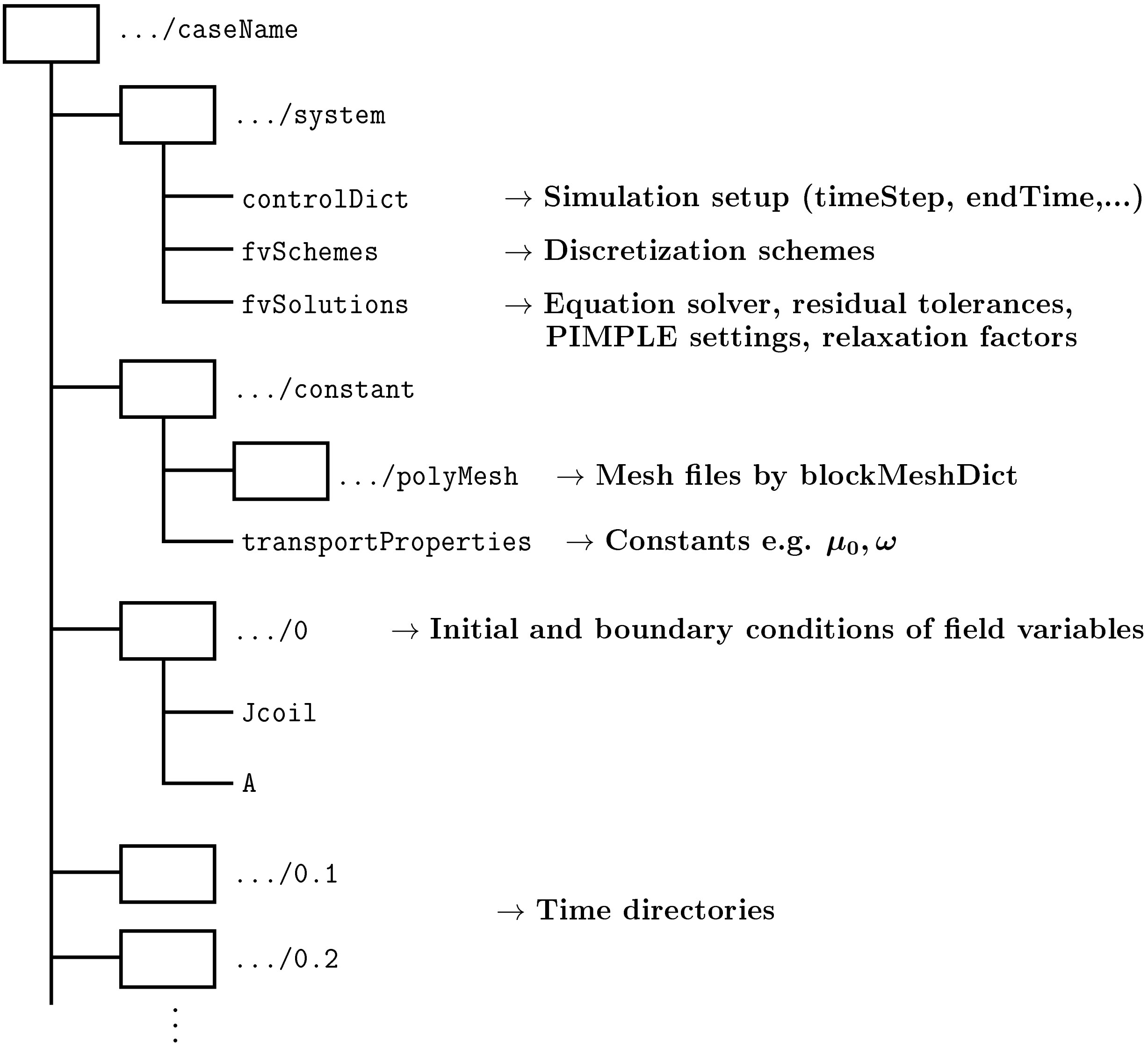
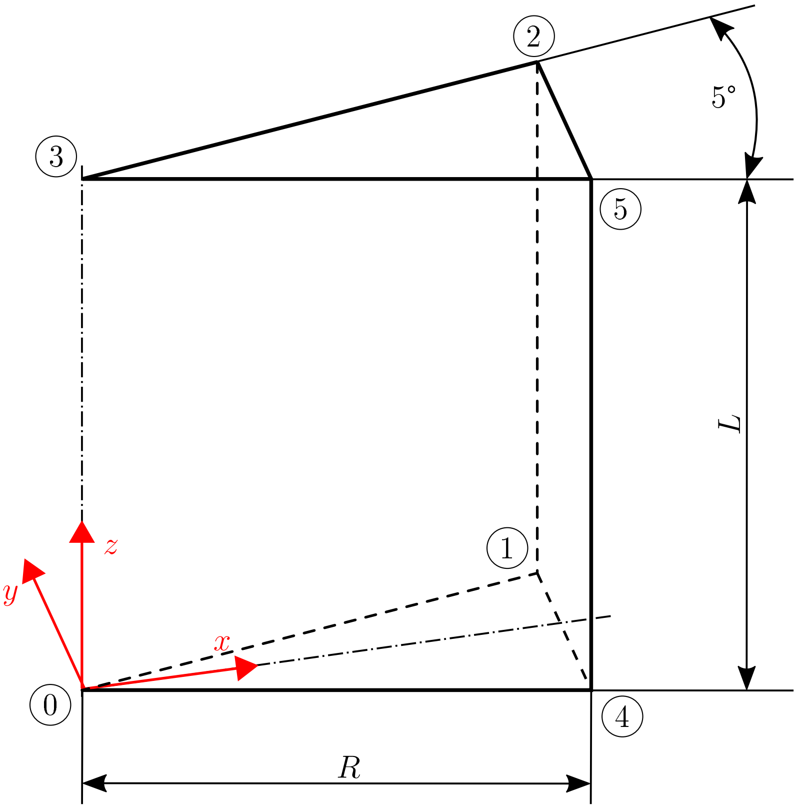
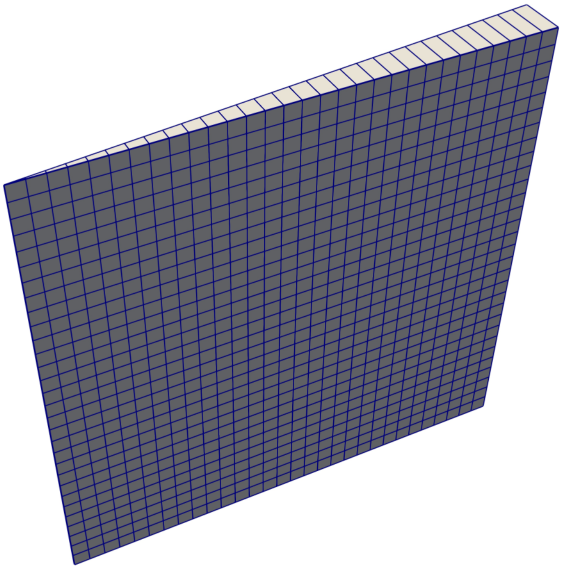

.../caseName
.../system
controlDict
→ Simulation setup (timeStep, endTime,...)
fvSchemes
→ Discretization schemes
fvSolutions
→ Equation solver, residual tolerances, PIMPLE settings, relaxation factors
.../constant
.../polyMesh
→ Mesh files by blockMeshDict
transportProperties
→ Constants e.g. μ_0, ω
.../0
→ Initial and boundary conditions of field variables
Jcoil
A
.../0.1
→ Time directories
.../0.2

Figure C.1.: Basic structure of a simulation case in OpenFOAM

In this work the blockMeshDict utility was employed to generate the mesh for the axisymmetric model of the ICP torch. Since OpenFOAM is based on the finite-volume method, two-dimensional axisymmetric cases can be simulated by using a mesh with wedge-shaped geometry as shown in Figure C.2. The details of generating wedge-shaped geometries using blockMesh are described in the OpenFOAM Userguide [Ope19c, U-47], and [Ope19d]. In this work the complete mesh for the ICP torch is rather sophisticated and consists of multiple wedge blocks, where for each block the coordinates of vertices need to be specified similarly to the schematic in Figure C.2a. Therefore, a Python-script has been written to automatically compute the coordinates of vertices based on the input dimensions of the torch $r_1, r_2, L_1, L_2, ...$ according to the presented schemes, e.g., Figure 3.17.



(a) Schematic showing how to specify the vertices of a wedge block according to the blockMesh utility



(b) View of a generated wedge mesh in Paraview

Figure C.2.: Mesh generation using blockMeshDict

APPENDIX D

Notes on the Code

The following sections provides some additional explanations with code snippets that might be useful for the future OpenFOAM user.

D.1. Implementation: Electromagnetic solver

The implementation of the electromagnetic solver is based on the vector potential matrix Equation (2.27), which in expanded form is expressed as

$$\begin{bmatrix} \boldsymbol{\Delta} & 0 & 0 & D_{\mathrm{A}} & 0 & 0 \\ 0 & \boldsymbol{\Delta} & 0 & 0 & D_{\mathrm{A}} & 0 \\ 0 & 0 & \boldsymbol{\Delta} & 0 & 0 & D_{\mathrm{A}} \\ -D_{\mathrm{A}} & 0 & 0 & \boldsymbol{\Delta} & 0 & 0 \\ 0 & -D_{\mathrm{A}} & 0 & 0 & \boldsymbol{\Delta} & 0 \\ 0 & 0 & -D_{\mathrm{A}} & 0 & 0 & \boldsymbol{\Delta} \end{bmatrix} \begin{bmatrix} \widetilde{\mathbf{A}}_{\mathrm{R,x}} \\ \widetilde{\mathbf{A}}_{\mathrm{R,y}} \\ \widetilde{\mathbf{A}}_{\mathrm{R,z}} \\ \widetilde{\mathbf{A}}_{\mathrm{I,x}} \\ \widetilde{\mathbf{A}}_{\mathrm{I,y}} \\ \widetilde{\mathbf{A}}_{\mathrm{I,z}} \end{bmatrix} = \begin{bmatrix} -\mu_0 \widetilde{\mathbf{J}}_{\mathrm{coil,x}} \\ -\mu_0 \widetilde{\mathbf{J}}_{\mathrm{coil,y}} \\ -\mu_0 \widetilde{\mathbf{J}}_{\mathrm{coil,z}} \\ 0 \\ 0 \\ 0 \end{bmatrix}, \tag{D.1}$$

with the auxiliary coefficient D_{A} given by

$$D_{\mathrm{A}} = \mu_0 \sigma \omega. \tag{D.2}$$

In order to build and solve the matrix Equation (D.1) in a fully coupled manner, the library `fvBlockMatrix.h` provided in `foam-extend 4.0` was used. The corresponding code snippet is shown in D.1.

```
// --- Electromagnetic field solver
// Update DA
DA = muMag*sigma*w;

// Initialize the Up block system
```

```
fvBlockMatrix<vector6> AEqn(A);

fvVectorMatrix AREqn
(
fvm::laplacian(AR) == - muMag*Jcoil
);

fvVectorMatrix AIEqn
(
fvm::laplacian(AI)
);

// insert fvVectorMatrix equations into the fvBlockMatrix
AEqn.insertEquation(0, AREqn);
AEqn.insertEquation(3, AIEqn);

// Add off-diagonal coupling terms
AEqn.insertEquationCoupling(0, 3, DA);
AEqn.insertEquationCoupling(1, 4, DA);
AEqn.insertEquationCoupling(2, 5, DA);
AEqn.insertEquationCoupling(3, 0, -DA);
AEqn.insertEquationCoupling(4, 1, -DA);
AEqn.insertEquationCoupling(5, 2, -DA);

// Solve the block matrix
maxResidual = cmptMax(AEqn.solve().initialResidual());

// Retrieve vector potential solution
AEqn.retrieveSolution(0, AR.internalField());
AEqn.retrieveSolution(3, AI.internalField());
Amag = sqrt((AR & AR) + (AI & AI)); // & - dot product

// Retrieve B, H, Jind fields
BR = fvc::curl(AR);
BI = fvc::curl(AI);
Hmag = Bmag/muMag;
JR = sigma*w*AI;
JI = -sigma*w*AR;
Jind = sqrt((JR & JR) + (JI & JI)); // & - dot product

// Compute the Joule heat and Lorentz force
qJ = sigma/2.0 * sqr(w) * sqr(Amag); //sqr(x) = x^2: square operator
fL = 0.5 * ((JR ^ BR) + (JI ^ BI)); // a^b - vector product
```

Listing D.1: Electromagnetic solver, code snippet from the source file: emFoam.C

D.2. Note: Enthalpy equation and temperature calculation

In OpenFOAM the enthalpy equation (2.3) is implemented in a slightly different form. The following code snippet is taken from the source file `hEqn.H` of the plasma solver:

```
fvScalarMatrix hEqn
(
        fvm::ddt(rho, h)
        + fvm::div(phi, h)
        - fvm::laplacian(turbulence->alphaEff(), h)
        ==
        DpDt
        + qJ
        - qR
);
```

which corresponds to the equation

$$\frac{\partial \rho h}{\partial t} + \boldsymbol{\nabla} \cdot (\phi h) + \Delta(\alpha_{\text{eff}} h) = \frac{\partial p}{\partial t} + \overline{q}_{\text{J}} - q_{\text{R}}, \tag{D.3}$$

where ϕ is the flux given by

$$\phi = \rho \mathbf{u} \tag{D.4}$$

and α_{eff} is the laminar thermal diffusivity for enthalpy in $\text{kg}\,\text{m}^{-1}\,\text{s}^{-1}$ given by

$$\alpha_{\text{eff}} = \frac{k}{c_{\text{p}}}. \tag{D.5}$$

Temperature calculation. The default implementation of OpenFOAM's thermophysical library calculates the temperature by solving the equation of state

$$\Delta h = \int c_{\text{p}}(T)\, dT, \tag{D.6}$$

in a iterative manner using the Newton-Raphson method. The corresponding source file can be found in the directory

`.../src/thermophysicalModels/specie/thermo/specieThermo/specieThermoI.H`

In discretized form the Equation of State (D.6) can be written as

$$h(T_{\text{est}}) - h = c_{\text{p}}(T_{\text{est}})[T_{\text{new}} - T_{\text{est}}], \tag{D.7}$$

which can be rearranged to obtain the equation to be solved

$$T_{\text{new}} = T_{\text{est}} + \frac{h(T_{\text{est}}) - h}{c_{\text{p}}(T_{\text{est}})}, \tag{D.8}$$

where h is the enthalpy obtained by solving the enthalpy equation; T_{new} is the new temperature value for the next iteration; T_{est} is the current estimate of the temperature; the values of $c_{\text{p}}(T_{\text{est}})$ and $h(T_{\text{est}})$ are updated via look-up table interpolation in every iteration. The iterative solution is finished when the residual $|T_{\text{new}} - T_{\text{est}}|$ is below the tolerance of $T_0 \cdot 1 \times 10^8$, where T_0 is the initial estimate of the temperature.

D.3. Two-mesh technique

In order to handle multiple meshes, a couple of modifications to the solver source code are needed. Basically, this can be applied to any application in OpenFOAM. In the following the key steps in order to utilize a second mesh, as it was done in this work, are described.

In the main .C-file of the solver, the following code is inserted, e.g., after the line where `"createTime.H"` is included:

Code:

```
// create a new mesh object for the electromagnetic domain
fvMesh emMesh
(
        Foam::IOobject
        (
                word("emRegion"),
                runTime.timeName(),
                runTime,
                IOobject::MUST_READ
        )
);

// keep default variable "mesh" for the fluid domain to avoid
// compilation errors with the default fluid solvers
fvMesh mesh
(
        Foam::IOobject
        (
                word("fluidRegion"),
                runTime.timeName(),
                runTime,
                IOobject::MUST_READ
        )
);
```

As a result, the basic case structure needs to be extended with subdirectories for the fluid and the electromagnetic domain, containing the respective field variables, mesh data, and solutions settings, as follows:

```
../caseName/
        |
        |-- 0/
        | |-- emRegion/ --> initial conditions AR, AI, Jcoil, sigma,...
        | |-- fluidRegion/ --> initial conditions T, U, p,...
        |
        |-- const/
        | |-- emRegion/polyMesh/ --> emMesh
        | |-- fluidRegion/polyMesh/ --> mesh (fluidMesh)
        |
        |-- system/
        | |-- emRegion/ --> fvSolution, fvSchemes
        | |-- fluidRegion/ --> fvSolution, fvSchemes
```

Accordingly, the field variables need to be assigned to the corresponding meshes in the `createFields.H` file of the solver. For example, a new field variable in the electromagnetic mesh is assigned by:

Code:

```
volScalarField A
(
        IOobject
        (
                "A",
                runTime.timeName(),
                emMesh,
                IOobject::MUST_READ,
                IOobject::AUTO_WRITE
        ),
        emMesh
);
```

Whereas the standard fluid fields (temperature, velocity, pressure) may remain unchanged:

Code:

```
volScalarField T
(
        IOobject
        (
                "T",
                runTime.timeName(),
                mesh,
                IOobject::MUST_READ,
                IOobject::AUTO_WRITE
        ),
        mesh
);
```

However, if a field variable is used in both the fluid and the electromagnetic mesh (due to physical coupling), then it is recommended to assign a variable on each mesh. For example, the Joule heat (qJ) should be declared as

Code:

```
volScalarField qJfluid
(
        IOobject
        (
                "qJ",
                runTime.timeName(),
                mesh,
                IOobject::MUST_READ,
                IOobject::AUTO_WRITE
        ),
        mesh
);

volScalarField qJem
(
        IOobject
        (
                "qJ",
                runTime.timeName(),
                emMesh,
                IOobject::MUST_READ,
                IOobject::AUTO_WRITE
        ),
        emMesh
);
```

This will also make it clear from the code, to which mesh the variable is assigned to. In order to map the field values between the meshes at run time, the `meshToMesh.H` library can be employed.

Therefore, the following code is added to the main .C-file, also after `"createTime.H"`:

Code:

```
// read the mapFieldsDict located in /case/system
IOdictionary mapFieldsDict
(
        IOobject
        (
                "mapFieldsDict",
                runTime.system(),
                runTime,
                IOobject::MUST_READ,
                IOobject::NO_WRITE,
                false
        )
);

HashTable<word> patchMap;
wordList cuttingPatches;
mapFieldsDict.lookup("patchMap") >> patchMap;
mapFieldsDict.lookup("cuttingPatches") >>  cuttingPatches;
```

In addition, an empty `mapFieldsDict` file needs to be created and put into the `system` directory of the case:

Code:

```
FoamFile
{
        version     2.0;
        format      ascii;
        class       dictionary;
        object      mapFieldsDict;
}
// * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * //

// List of pairs of source/target patches for mapping
patchMap
(
);

// List of target patches cutting the source domain (these need to be
// handled specially e.g. interpolated from internal values)
cuttingPatches
(
);
```

Finally, to perform the mapping and interpolation of the field variable at runtime, the following code is executed:

Code:

```
// specify source and target mesh for the mapping
const fvMesh& meshSource = emMesh; //meshToMeshInterp.fromMesh();
const fvMesh& meshTarget = mesh; //meshToMeshInterp.toMesh();

// Create the interpolation scheme
meshToMesh meshToMeshInterp
(
        meshSource,
        meshTarget,
        patchMap,
        cuttingPatches
);

Info<< "Target mesh size: " << meshTarget.nCells() << endl;

Info<< nl
<< "Mapping fields for time " << meshSource.time().timeName()
<< nl << endl;

// Interpolate field
meshToMeshInterp.interpolate
(
        qJfluid, //target field
        qJem, //source field
        meshToMesh::INTERPOLATE
);
```

Note, that during this work it was found, that an error occured, when mapping temperature field values from the smaller fluid mesh to the bigger electromagnetic mesh, which led to an undesired extrapolation of the field values on the electromagnetic mesh. Therefore, the following simple routine is used as a workaround in the plasma solver to update the electrical conductivity field values and to avoid the mapping of the temperature field values:

Code:

```
forAll(T, idx)
{
        const scalar& cellT = T[idx];
        sigma[idx] = sigmaLookUp(cellT);
}
```

As a consequence, the fluid mesh and the electromagnetic need to be identical in the overlapping region in order to avoid incorrect addressing of cells.

Bibliography

[Bab47] George I. Babat. „Electrodeless discharges and some allied problems“. In: *Journal of the Institution of Electrical Engineers - Part III: Radio and Communication Engineering* 94.27 (1947), pp. 27–37. DOI: 10.1049/ji-3-2.1947.0005 (cit. on p. 2).

[BB15] Angel Ochoa Brezmes and Cornelia Breitkopf. „Fast and reliable simulations of argon inductively coupled plasma using COMSOL“. In: *Vacuum* 116 (2015), pp. 65–72. DOI: 10.1016/j.vacuum.2015.03.002 (cit. on p. 6).

[Bec18] Pascal Beckstein. „Methodenentwicklung zur Simulation von Strömungen mit freier Oberfläche unter dem Einfluss elektromagnetischer Wechselfelder“. Dissertation. Dresden: Technische Universität Dresden, 2018. URL: https://nbn-resolving.org/urn:nbn:de:bsz:14-qucosa-232474 (cit. on p. 7).

[Ber+03] D. Bernardi et al. „Comparison of different techniques for the FLUENT©-based treatment of the electromagnetic field in inductively coupled plasma torches“. In: *The European Physical Journal D - Atomic, Molecular and Optical Physics* 27 (2003), pp. 55–72. DOI: 10.1140/epjd/e2003-00227-1 (cit. on pp. 6–9, 11 sq., 25, 32, 35–37).

[Ber+05] D. Bernardi et al. „Three-dimensional modeling of inductively coupled plasma torches“. In: *Pure and Applied Chemistry* 77.2 (2005), pp. 359–372. DOI: 10.1351/pac200577020359 (cit. on pp. 6, 35, 38).

[BGV17] Pascal Beckstein, Vladimir Galindo, and Vuko Vukčević. „Efficient solution of 3D electromagnetic eddy-current problems within the finite volume framework of OpenFOAM“. In: *Journal of Computational Physics* 344 (2017), pp. 623–646. DOI: 10.1016/j.jcp.2017.05.005 (cit. on pp. 7, 14).

[Bou04] Maher Boulos. „Plasma power can make better powders“. In: *Metal Powder Report* 59.5 (2004), pp. 16–21. DOI: 10.1016/S0026-0657(04)00153-5 (cit. on p. 2).

[Bre19] E. W.I. Brendan Lewis. *The Importance of Powder Characterization in Metal Additive Manufacturing.* 2.10.2019. URL: https://tinyurl.com/y4u2rf6h (visited on 10/02/2019) (cit. on p. 1).

[CFD18a] CFD Direct Ltd. *OpenFOAM v6 User Guide: 4.5 Solution and algorithm control.* 2018. URL: https://cfd.direct/openfoam/user-guide/v6-fvsolution/ (visited on 11/16/2019) (cit. on pp. 20, 22).

[CFD18b] CFD Direct Ltd. *OpenFOAM v6 User Guide: 5 Mesh generation and conversion.* 2018. URL: `https://cfd.direct/openfoam/user-guide/v6-mesh/` (visited on 11/13/2019) (cit. on p. 70).

[Cho15a] Isabelle Choquet. *Source files: Argon Properties Look-up table.* 2015. URL: `http://www.tfd.chalmers.se/~hani/kurser/OS_CFD_2015/IsabelleChoquet/files.tgz` (visited on 09/15/2019) (cit. on pp. 12 sq., 21).

[Cho15b] Isabelle Choquet. „ThermophysicalModels library in OpenFOAM-2.3.x (or 2.4.x). How to implement a new thermophysical model". In: (2015). URL: `http://www.tfd.chalmers.se/~hani/kurser/OS_CFD_2015/IsabelleChoquet/thermophysicalModels-OF-2.3-or-2.4.x.pdf` (visited on 09/15/2019) (cit. on p. 21).

[COM19a] COMSOL INC. *Inductively Coupled Plasma (ICP) torch.* 2019. URL: `https://www.comsol.de/model/inductively-coupled-plasma-icp-torch-18125` (visited on 08/05/2019) (cit. on p. 6).

[COM19b] COMSOL INC. *Materialbibliothek für Multiphysiksimulationen - Hinzufügen von Materialeigenschaften.* 2019. URL: `https://www.comsol.de/material-library` (visited on 11/16/2019) (cit. on p. 12).

[COM19c] COMSOL INC. *Plasma Software - Model Low-Temperature Plasma Sources and Systems.* 2019. URL: `https://www.comsol.eu/plasma-module` (visited on 08/05/2019) (cit. on p. 6).

[Cre+13] Y. Cressault et al. „Thermal plasma properties for Ar–Cu, Ar–Fe and Ar–Al mixtures used in welding plasmas processes: II. Transport coefficients at atmospheric pressure". In: *Journal of Physics D: Applied Physics* 46.41 (2013), p. 415207. DOI: `10.1088/0022-3727/46/41/415207` (cit. on pp. 12 sq.).

[Dav01] P. A. Davidson. *An introduction to magnetohydrodynamics.* eng. Vol. 25. Cambridge texts in applied mathematics. Cambridge: Cambridge University Press, 2001. 1452 pp. DOI: `10.1017/CBO9780511626333` (cit. on pp. 14, 64, 67).

[Dew+10] Rakesh Kumar Dewangan et al. „State-space modeling of the radio frequency inductively-coupled plasma generator". In: *Plasma Sources Science and Technology* 208 (2010), p. 012056. DOI: `10.1088/1742-6596/208/1/012056` (cit. on p. 51).

[Die+16] S. Dietrich et al. „A New Approach for a Flexible Powder Production for Additive Manufacturing". In: *Procedia Manufacturing* 6 (2016), pp. 88–95. DOI: `10.1016/j.promfg.2016.11.012` (cit. on p. 1).

[Dzu04] Birger Dzur. „Ein Beitrag zur Anwendung des induktiv gekoppelten Hochfrequenz-Plasmas zum atmosphärischen Plasmaspritzen oxidkeramischer Werkstoffe". Fakultät für Elektrotechnik und Informationstechnik. Dissertation. Ilmenau: Technische Universität Ilmenau, 2004. URL: `https://nbn-resolving.org/urn:nbn:de:gbv:ilm1-2002000234` (cit. on p. 51).

[Dzu09] Birger Dzur. *Pulverbehandlung mit dem ICP.* Praktikumsanleitung. Technische Universität Ilmenau. 2009. URL: `https://www.tu-ilmenau.de/fileadmin/media/wt/Lehre/Praktikum/Werkstoffwissenschaft_WSW/5._Fachsemester/_PulverICP__Pulverbehandlung_mit_dem_ICP.pdf` (visited on 11/08/2019) (cit. on pp. 3 sq.).

[FHB14] Pierre L. Fauchais, Joachim V.R. Heberlein, and Maher I. Boulos. *Thermal Spray Fundamentals*. Boston, MA: Springer US, 2014. DOI: 10.1007/978-0-387-68991-3 (cit. on p. 54).

[Fou19] Foundry Trade Journal. *The importance of powder quality in additive manufacturing*. 2.10.2019. URL: https://tinyurl.com/y3lkz7s4 (visited on 10/02/2019) (cit. on p. 1).

[GR09] Christophe Geuzaine and Jean-François Remacle. „Gmsh: A 3-D finite element mesh generator with built-in pre- and post-processing facilities“. In: *International Journal for Numerical Methods in Engineering* 79.11 (2009), pp. 1309–1331. DOI: 10.1002/nme.2579 (cit. on p. 70).

[Guo+19] Xiaoman Guo et al. „Time-resolved particle image velocimetry and 3D simulations of single particles in the new conical ICP torch“. In: *Journal of Analytical Atomic Spectrometry* 34.3 (2019), pp. 469–479. DOI: 10.1039/c8ja00407b (cit. on p. 6).

[Hrv16] Hrvoje Jasak. *Numerics Improvements in OpenFOAM with Examples of Industrial CFD*. Moscow, Dec. 1, 2016. URL: https://www.ispras.ru/conf/2016/pdf/Jasak.pdf (visited on 09/13/2019) (cit. on p. 6).

[Hua+92] Mao Huang et al. „Isocontour maps of electron temperature, electron number density and gas kinetic temperature in the Ar inductively coupled plasma obtained by laser-light Thomson and Rayleigh scattering“. In: *Spectrochimica Acta Part B: Atomic Spectroscopy* 47.6 (1992), pp. 765–785. DOI: 10.1016/0584-8547(92)80072-O (cit. on p. 55).

[Jia+10] Guo Jiayin et al. „Development of Nanopowder Synthesis Using Induction Plasma“. In: *Journal of Physics D: Applied Physics* 12.2 (2010), pp. 188–199. DOI: 10.1088/1009-0630/12/2/12 (cit. on p. 51).

[MB98] A. Merkhouf and M. I. Boulos. „Integrated model for the radio frequency induction plasma torch and power supply system“. In: *Journal of Physics D: Applied Physics* 7 (1998), pp. 599–606. DOI: 10.1088/0963-0252/7/4/017 (cit. on p. 54).

[MPY18] Javad Mostaghimi, Larry Pershin, and Subramaniam Yugeswaran. „Heat Transfer in DC and RF Plasma Torches“. In: *Handbook of Thermal Science and Engineering*. Vol. 30. Cham: Springer International Publishing, 2018, pp. 2523–2597. DOI: 10.1007/978-3-319-26695-4_27 (cit. on p. 2).

[Nag+17] K.Yu. Nagulin et al. „Transient three-dimensional dynamics of argon plasma within the vacuum interface of the inductively coupled plasma mass spectrometer system“. In: *Spectrochimica Acta Part B: Atomic Spectroscopy* 135 (2017), pp. 63–72. DOI: 10.1016/j.sab.2017.07.006 (cit. on p. 35).

[Nij19] Sander Nijdam. *Thomson, Rayleigh and Raman scattering*. English. Eindhoven University of Technology. 2019. URL: https://www.tue.nl/en/our-university/departments/applied-physics/research/research-groups/elementary-processes-in-gas-discharges-epg/facilities/active-spectroscopy/thomson-rayleigh-and-raman-scattering/ (visited on 11/11/2019) (cit. on p. 55).

[NNK18] K. Nagulin, R. Nazarov, and Kudimov O., Salikhov R., Gilmutdinov A. „Plasmochemical processing of powder materials for heat-protective coatings". In: *9th Nordic Conference on Plasma Spectrochemistry.* (Loen, Norway). 2018, p. 92. URL: https://orbit.dtu.dk/ws/files/149814111/NordicPlasma_Programme_2018_Final.pdf (cit. on p. 2).

[Ope19a] OpenCFD Ltd. *OpenFOAM: User Guide: Residuals.* 10.10.2019. URL: https://www.openfoam.com/documentation/guides/latest/doc/guide-solvers-residuals.html (visited on 10/28/2019) (cit. on p. 47).

[Ope19b] OpenCFD Ltd. *Mesh generation with the snappyHexMesh utility.* 2019. URL: https://www.openfoam.com/documentation/user-guide/snappyHexMesh.php (visited on 11/13/2019) (cit. on p. 70).

[Ope19c] OpenFOAM. *User Guide. The Open Source CFD Toolbox.* 2019. URL: https://sourceforge.net/projects/openfoamplus/files/v1812/UserGuide.pdf (visited on 04/06/2019) (cit. on p. 71).

[Ope19d] OpenFOAMWiki Disclaimers. *Main ContribExamples/AxiSymmetric - OpenFOAMWiki.* 2019. URL: https://openfoamwiki.net/index.php/Main_ContribExamples/AxiSymmetric (visited on 11/13/2019) (cit. on p. 71).

[Ree61] Thomas B. Reed. „Induction–Coupled Plasma Torch". In: *Journal of Applied Physics* 32.5 (1961), pp. 821–824. DOI: 10.1063/1.1736112 (cit. on p. 2).

[RSB95] Mohamed Rahmane, Gervais Soucy, and Maher I. Boulos. „Diffusion phenomena of a cold gas in a thermal plasma stream". In: *Plasma Chemistry and Plasma Processing* 16.S1 (1995), S169–S189. DOI: 10.1007/BF01512634 (cit. on p. 51).

[Sch13] Rüdiger Schwarze. *CFD-Modellierung. Grundlagen und Anwendungen bei Strömungsprozessen.* ger. Berlin and Heidelberg: Springer, 2013. 11 pp. DOI: 10.1007/978-3-642-24378-3 (cit. on p. 42).

[Shu16] Shukla, Vipin and Vadlamannati, Sriram. *Additive Manufacturing Using Metal Powder – An Insight.* Technology Information Forecasting & Assessment Council. 2016. URL: https://www.researchgate.net/publication/305317402_Additive_Manufacturing_Using_Metal_Powder_-_An_Insight (visited on 11/12/2019) (cit. on p. 1).

[Sou19a] Sourceforge. *foam-extend 4.0 Download.* 2019. URL: https://sourceforge.net/projects/foam-extend/ (visited on 10/13/2019) (cit. on p. 24).

[Sou19b] Sourceforge. *foam-extend / foam-extend-4.0 / [268bb0] /ReleaseNotes.* 2019. URL: https://sourceforge.net/p/foam-extend/foam-extend-4.0/ci/master/tree/ReleaseNotes.txt (visited on 09/04/2019) (cit. on p. 7).

[TC13] Emre Turkoz and Murat Celik. „2D Fluid Model for Axisymmetric RF Ion Thruster Cylindrical Discharge Chamber". In: *Joint Propulsion Conferences. 49th AIAA/ASME/SAE/ASEE Joint Propulsion Conference.* 49th AIAA/ASME/SAE/ASEE Joint Propulsion Conference. (San Jose, CA). 2013, p. 2111. DOI: 10.2514/6.2013-4110 (cit. on p. 51).

[TC15] Emre Turkoz and Murat Celik. „AETHER: A simulation platform for inductively coupled plasma". In: *Journal of Computational Physics* 286 (2015), pp. 87–102. DOI: 10.1016/j.jcp.2015.01.027 (cit. on p. 51).

[Tsi18] Ilya Tsivilskiy. *Additive Manufacturing Intro: Overview. Lecture Notes, GRIAT summer school: Wings for Life.* Kazan National Research Technical University named after A.N.Tupolev - KAI, Aug. 16, 2018 (cit. on pp. 2 sq.).

[Vor+18] M. Voronov et al. „Force-based analysis of vortexes in atmospheric pressure ICPs“. en. In: *Plasma Sources Science and Technology* 27.12 (2018), p. 125005. DOI: 10.1088/1361-6595/aaef1c (cit. on pp. 2–4, 6).

[Wol19] Wolf Dynamics. *Finite Volume Method: A Crash introduction.* 2019. URL: http://www.wolfdynamics.com/wiki/fvm_crash_intro.pdf (visited on 10/20/2019) (cit. on p. 20).

[XPB01] Siwen Xue, Pierre Proulx, and Maher I. Boulos. „Extended-field electromagnetic model for inductively coupled plasma“. In: *Journal of Physics D: Applied Physics* 34.12 (2001), pp. 1897–1906. DOI: 10.1088/0022-3727/34/12/321 (cit. on pp. 6, 8 sq., 12, 14, 25, 32, 35 sq., 39 sq.).

[Yan+00] Pengyuan Yang et al. „Comparison of simulated and experimental fundamental ICP parameters“. In: *Spectrochimica Acta Part B: Atomic Spectroscopy* 55.12 (2000), pp. 1833–1845. DOI: 10.1016/S0584-8547(00)00274-3 (cit. on p. 55).