Faculty for mechanical engineering
MathCCES
Prof. Dr. Manuel Torrilhon

Bachelorthesis

Arc Simulations in OpenFOAM

Jannick Wolters
Matriculation number: 302517

April 2014

Internal reviewer: Prof. Dr. Manuel Torrilhon
External reviewer: Prof. Dr. Vincent Dousset

I hereby affirm that I composed this work independently and used no other than the specified sources and tools and that I marked all quotes as such.

Hiermit versichere ich, diese Arbeit selbständig verfasst und keine anderen als die angegebenen Quellen und Hilfsmittel benutzt, sowie Zitate kenntlich gemacht zu haben.

Aachen, April 13, 2014

Jannick Wolters

Contents

Chapter 1

Abstract

The study of electromagnetic arc phenomena has always been a challenging multi-physics problem. As electric arcs are burning with temperatures up to 20000°K and higher, the gas in the arc region turns into plasma, in which atoms are now fully ionized and electrons can move freely. This alters the properties of the material significantly.
The high temperatures also cause physical effects like erosion and ablation of other materials present in the domain.
Getting a complete impression of the acting physical forces by performing measurements in the laboratory, is extremely difficult, even if the considered geometries are simple. The pure variety of phenomena and the rapidly changing nature of the problem make it a long and expensive process to either investigate problems or to optimize the geometry due to certain aspects. When it comes to more complex designs, like nowadays state of the art high current circuit breakers, analysing and optimizing the design parameters is a challenging task, that can badly be handled with pure empirical measurements.
Numerical simulations can resolve arc phenomena on small time scales and are therefore able to give good optimization approaches and deeper insight into the problems.
The computational fluid dynamics (CFD) tool OpenFOAM offers some promising features and the standard configuration has many needed modeling abilities already available. Nevertheless, it is not able to simulate electric arc phenomena by default.
This work will deal with extending OpenFOAMs abilities and examining if it is a suitable alternative to other available CFD-tools.

Chapter 2

Introduction

The goal of this work is to examine the CFD-tool OpenFOAM with regards to its capability of solving electric arc phenomena.
The considered problem is a burning arc between two electrodes, which is surrounded by a nozzle made of polytetrafluoroethylene (PTFE). As the arc ignites, the gas in the arc region quickly heats up to about 20000°K. This causes several different effects:
First of all the gas expands rapidly, causing a shock pointing outwards from the arcs center. As the whole gas inside the nozzle can only exit through the ends, high velocities up to supersonic speed are obtained in these regions.
Secondly, the arcs high temperatures cause all surrounding materials in close range to the arc to vaporize and thereby additional materials are added to the system. In this work these ablation effects will only be considered at the nozzle. Erosion of cathode and anode will be neglected.
Due to the high velocities and occuring shocks a density based solver has to be implemented. Such a solver is already present in OpenFOAM, but it has to be modified with respect to the governing electromagnetic forces. Therefore, the magnetohydrodynamic equations will be examined and with several simplifying assumptions a weakly coupled set of equations will be derived for the physical problem described above.
Another missing package in the OpenFOAM framework is the ability to model ablation effects. Such a package will therefore be created and added to the library.
Early investigations have shown, that non of the thermophysical configurations in OpenFOAM is suitable for the expected large temperature and pressure ranges. To deal with this problem an extension will be implemented, allowing fast interpolation on property tables.
Finally the obtained results will be discussed and the positive and negative sides of OpenFOAM will be highlighted.

Chapter 3

OpenFOAM

3.1 History

OpenFOAM stands for "Open Source Field Operation and Manipulation". It was first released under the GNU general public license in 2004 by OpenCFD Ltd. Its commercial predecessor FOAM was originally developed at Imperial College London in the 1980s.
The motivation behind the project was to create a highly flexible and expandable software package that could easily be reused by other scientists.
Nowadays OpenFOAM is broadly used in industry and research. It additionally has a big online community of private scientists, that improve the functionality day by day.
Since the very first release, OpenFOAM has steadily been improved and just recently was updated to the latest version 2.3.0 in the beginning of 2014. Due to unfulfilled software requirements it was not possible to install OpenFOAM v2.3.0 on the cluster used for this work. Because of that, the previous version 2.2.1 has been used.

3.2 Features

As previously mentioned, OpenFOAM is a completely open source and license free computational library. It provides different sorts of software packages, that can be assembled into solvers to form a powerful, problem specific simulation tool. Here a short list of default abilities:

- solving PDEs through finite volume approach
- turbulence modeling (RAS, LES)

- radiative heat transfer (P1, DOM, viewFactor)
- multi species and multi domain solving
- combustion and chemical reactions
- dynamic meshes
- parallelization by Open MPI

All these abilities come preconfigured in several default solvers for different kinds of physical problems. The range goes from simple solvers for potential flows to compressible, chemical reacting, combustion solvers.
OpenFOAM is written in C++ and follows an object oriented approach combined with an extensive use of templatization. The code is designed to be as intuitive as possible for the user, while at the same time aiming for high efficiency in the subroutines. Equations can therefore almost be transfered into OpenFOAM in their exact mathematical expression.
To give an example, the compressible Euler momentum equation

$$\frac{\partial}{\partial t}(\rho\vec{U}) + \nabla \cdot (\phi\vec{U}) = -\nabla p \tag{3.1}$$

is represented in OpenFOAM by

Listing 3.1: OpenFOAM syntax

```
solve
(
        fvm::ddt(rho, U)
      + fvm::div(phi, U)
      ==
      - fvc::grad(p)
)
```

fvm denotes, that the velocity $\vec{U}$ has to be calculated implicitly from e.g. the divergence term. *fvc* refers to explicitly given values like the gradient of the pressure p in this example. Implicit/explicit in this context means that the corresponding variables are put on the left/right side of the linear system $Ax = b$ which represents the equation after a discretization scheme has been applied.
Due to this simple top level programming syntax, the user is not required to have an in-depth knowledge of C++.
This experience quickly changes if one takes a deeper look into the source code. The highly user friendly top level code structure and the "plug-and-play" ability of most modules is achieved by heavy usage of templates and inheritance. This techniques makes the code difficult to read, because executing functions are spread across many different files.

3.3 General case setup

Every OpenFOAM case has a similar base structure. Differences are only motivated by solver specific requirements. In general a case consists of three directories:

- **0**: solver variables with boundary conditions and initial fields
- **constant**: module properties, mesh definition
- **system**: solver setup

The most important settings will be explained in this section as they are important later in this work.

3.3.1 Initial fields and boundary conditions

All solver variables have to be defined with a complete initial field and boundary conditions at solver initialization. Every variable requires its own definition file, stored in the folder *0*. Each of these files has to match a simple layout to be processed correctly.
Four blocks can be determined for each file:
First of all the header, which is important for OpenFOAMs object registry. It provides information about the fields name and about the type of field which is represented in the file (scalar or vector).

Listing 3.2: File header

```
1 FoamFile
2 {
3     version     2.0;
4     format      ascii;
5     class       volVectorField;
6     location    "0";
7     object      U;
8 }
```

Next up the fields dimension in SI units has to be given. The SI units (International System of Units) are a modern metric unit system, based on the meter-kilogram-second (MKS) system. It consists of seven units, which can be seen in listing 3.3. Providing OpenFOAM these units is necessary, because it will perform unit checks in all field operations. This means it will prevent the user from adding e.g. energy and velocity fields as they are not logically compatible.

Listing 3.3: SI Units

```
1 //                  kg m   s K kgmol A cd
2 dimensions        [ 0 1 -1 0    0    0   0];
```

The internal initial field values are given by

Listing 3.4: Initial field values

```
1 internalField    uniform (0 0 0);
```

The *uniform* keyword declares the following value to be set for the whole domain. As this example refers to the velocity $\vec{U}$, all vector components have to be given for the x-, y- and z-directions. For scalars only a single value is required.
Boundary conditions have to be declared linked to the boundary names defined by the mesh. The example from the velocity file shows how the conditions are applied correctly.

Listing 3.5: Definition of boundary conditions

```
1  boundaryField
2  {
3      inlet
4      {
5          type     fixedValue;
6          value    uniform (1 0 0);
7      }
8
9      outlet
10     {
11         type     zeroGradient;
12     }
13
14     ...
15 }
```

The boundary with the name *inlet* has a condition of type *fixedValue*. This sets the boundary faces to values corresponding to the keyword *value* in the following line. Here again values can be given as uniform or as a list, containing entries for all faces of the boundary.
OpenFOAM offers many different kinds of boundary conditions for various purposes. A list can be obtained from the official homepage[1].

[1] http://openfoam.org/docs/user/boundaries.php (April 2014)

3.3.2 Constant settings

The directory *constant* includes several solver specific configuration files. The most important and most used file among the standard solvers, is the thermodynamic configuration file named *thermophysicalProperties*.
This file determines fluid properties like the state of equation, mixtures and transport properties. As an example, the file would read for air:

Listing 3.6: Thermophysical configuration file

```
1  FoamFile
2  {
3      version     2.0;
4      format      ascii;
5      class       dictionary;
6      location    "constant";
7      object      thermophysicalProperties;
8  }
9
10 thermoType
11 {
12 // thermodynamic package based on density
13     type            heRhoThermo;
14 // single component fluid
15     mixture         pureMixture;
16 // constant transport properties
17     transport       const;
18 // constant heat capacity
19     thermo          hConst;
20 // ideal/perfect gas law
21     equationOfState perfectGas;
22 // only option - needed for mixture
23     specie          specie;
24 // energy calculated by solver is enthalpy
25     energy          sensibleEnthalpy;
26 }
27
28 mixture
29 {
30     specie
31     {
32         nMoles          1;
33         molWeight       28.96;
34     }
35     thermodynamics
36     {
37         Cp              1004.5;
38         Hf              2.544e+06;
39     }
```

```
40      transport
41      {
42          mu              1.8e-05;
43          Pr              0.7;
44      }
45  }
```

Again, as in all OpenFOAM files, the file starts with the object registry header.
Next the thermodynamic package is configured in the section *thermoType*. In this example the package is based on the density ρ, with constant transport properties and a constant heat capacity c_p. Furthermore, the ideal gas law is used as the equation of state. Except from defining these properties as constant, polynomial based approaches are also available.
The settings for each selected module are defined below at the bottom of the file. If a multi-species setup is chosen, each component has to be defined separately.
All available modules can be changed (e.g. constant transport properties can be replaced by a module based on Sutherland's law) as long as the setup physically makes sense. This plug and play ability is a good example for OpenFOAMs extensibility.

If the solver also handles turbulent flows, there has to be an additional file in the *constant* folder named *turbulenceProperties*. In this file one of the three turbulence options has to be set (RAS, LES, laminar). An additional file might be required dependent on the selected model.

The computational mesh is stored in the subfolder *polyMesh*. If OpenFOAMs built in mesh generator *blockMesh* shall be used, the file *blockMeshDict* has to be given in that subfolder.

3.3.3 Schemes and solution algorithms

The *systems* folder always contains three files:

- *controlDict*: simulation time, timestep size, output control
- *fvSolution*: algorithms setup for solved variables
- *fvSchemes*: numerical schemes and solver specific settings

An excerpt from the *fvSolution* file could for example look like:

Listing 3.7: Solver setup

```
solvers
{
    U
    {
        solver          PBiCG;
        preconditioner  DILU;
        tolerance       1e-06;
        relTol          0.1;
    }
}
```

The vectorfield U in this setup would be solved by a preconditioned biconjugate gradient solver (PBiCG) for asymmetric matrices. The preconditioning will be done by a diagonal-based incomplete LU (lower upper) method (DILU). Maximal residuals for the PBiCG decomposition is defined by the *tolerance* keyword, while the relative tolerance is prescribed by *relTol*.

For the discretization all used schemes are defined in the *fvSchemes* file. The file is divided in different sections, where each section is used for specific discretization schemes related to the keywords used in the equation implementation.

The continuity equation

$$\frac{\partial}{\partial t}\rho + \nabla \cdot (\rho \vec{v}) = 0$$

expressed to OpenFOAM syntax

Listing 3.8: Coninuity equation in OpenFoam syntax

```
1 fvm::ddt(rho) + fvc::div(phi) == 0
```

would for example need the following *fvSchemes* file:

Listing 3.9: Discretization schemes

```
1 ddtSchemes
2 {
3     ddt(rho)      Euler;
4 }
5
6 divSchemes
7 {
8     div(phi)      Gauss linear;
9 }
```

A list of available discretization schemes is given on the OpenFOAM homepage[2].

[2]http://openfoam.org/docs/user/fvSchemes.php (April 2014)

Chapter 4

Physical Problem

4.1 Problem description

The physical problem considered in this work is a arc burning in an initially insulating gas between two electrodes. The arc itself is surrounded by a nozzle made of polytetrafluoroethylene (PTFE), a synthetic polymer (chem. $(C_2F_4)_n$), which is most known under the brand name $Teflon^{®}$. Non-participating walls surround the electrodes and are connected to the nozzle. The gas itself is considered to be 100% gaseous PTFE.
The current flowing through the arc is inducing an azimuthal magnetic field, which in turn drives centripetal Lorentz forces. These forces squeeze the arc, by inducing a negative radial velocity to the gas.
As the arc is burning, temperatures inside reach values of the order of 10000°K, while the far field remains at ambient temperature. The large variations in temperature go along with large variations in pressure and subsequently also of other material properties like density, thermal conductivity, electric conductivity and viscosity.
With such high temperatures inside the arc, heat transfer is mostly driven by radiations. These radiations are partially absorbed by the surrounding gas. Radiation hitting the PTFE tube, causes its surface to evaporate and to release cool material into the surrounding gas.
All the previously mentioned aspects have to be taken into account in the physical model of the simulations. However some aspects are neglected.
At first it is considered, that local thermal equilibrium is satisfied and therefore the arc can be considered as a single "fluid" without distinguishing electrons from ions. Additionally all chemical reactions, occurring during the arcing process, are ignored. The chemical composition of the domain is considered as invariant. At last, erosion of the arcing contacts is neglected,

so that no metal vapor is released into the domain.
A suitable basic set of equations describing the problem are the so called *magnetohydrodynamic (MHD) equations*. These equations are a combination of the Navier-Stokes equations for fluid dynamics and Maxwell's equations for electromagnetism.

4.2 Navier-Stokes equations

To model the fluid physics, the compressible, transient and viscous Navier-Stokes equation have been taken into account. This set of equations consists of three conservation equations for mass, momentum and energy. Because of the influence of the electric and magnetic fields on the fluid, additional source/coupling terms have been added to the equations. Ablation and radiation sources/sinks have also been added to the standard Navier-Stokes equations.
Mass conservation:

$$\frac{\partial \rho}{\partial t} + \nabla \cdot (\rho \vec{U}) = \underbrace{S_{abl,m}}_{\text{ablation source}} \tag{4.1}$$

Momentum conservation:

$$\frac{\partial}{\partial t}(\rho \vec{U}) + \nabla \cdot (\rho \vec{U}\vec{U}) - \nabla \cdot (\mu \nabla \vec{U}) = -\nabla p + \underbrace{\vec{j} \times \vec{B}}_{\text{Lorentz force}} \tag{4.2}$$

Energy conservation:

$$\frac{\partial}{\partial t}(\rho e) + \nabla \cdot (\rho \vec{U} e) - \nabla \cdot (\alpha \nabla e) = \underbrace{\vec{j} \cdot \vec{E}}_{\text{Joule heating}} + \underbrace{q_{rad}}_{\text{radiation loss}} + S_{abl,e} \tag{4.3}$$

where ρ is the mass density, $\vec{U}$ the velocity vector, μ the dynamic viscosity, p the pressure, e the systems internal energy and α the thermal diffusivity.
The previously mentioned coupling terms between the fluid and the electromagnetic phenomena are the Lorentz force in the momentum equation and the Joule heating in the energy conservation as seen in equations 4.2 and 4.3. The Lorentz force will have a squeezing effect on the arc, as it adds a velocity component pointing towards the radial arc center. Because the fluid is resistive, applying a current will result in power losses, that are converted into heat. This heat source has to be added to the energy conservation and corresponds to the Joule heating term.
As the ablation phenomena is adding mass to the system, an according source

term $S_{abl,m}$ has been added to the mass conservation (4.1). Because the ablated mass also has a specific temperature, the related source term $S_{abl,e}$ had to be added to the energy equation. Further details on this topic will be given in section 4.7.
The source term q_{rad} represents the energy loss due to heat radiation.

4.3 Turbulence modeling

To model turbulent flow a Reynolds-averaged model has been chosen.
In this kind of turbulence approach the time depending variables Φ are decomposed into a fluctuating and a time averaged component such that:

$$\Phi(x,t) = \bar{\Phi}(x) + \Phi'(x,t) \tag{4.4}$$

where $\bar{\Phi'}(x,t) = 0$. This leads to a modified set of conservation equations known as the Reynolds-averaged Navier-Stokes (RANS) equations. Due to the reformulation, a closure problem arises, because of a newly added nonlinear term, the Reynold's stress tensor. This term now requires additional modeling, where turbulence models come into play.
In this work the $k-\omega-SST$ model has been used. This model is based on the standard $k-\omega$ model, but provides a special shear stress transport formulation.
In general two additional coupled transport equations have to be solved for the turbulent kinetic energy k and the turbulent frequency ω

$$\frac{\partial k}{\partial t} + U_j \frac{\partial k}{\partial x_j} = P_k - \beta^* k\omega + \frac{\partial}{\partial x_j}\left[(\nu + \sigma_k \nu_T)\frac{\partial k}{\partial x_j}\right] \tag{4.5}$$

$$\frac{\partial \omega}{\partial t} + U_j \frac{\partial \omega}{\partial x_j} = \alpha S^2 - \beta\omega^2 + \frac{\partial}{\partial x_j}\left[(\nu + \sigma_\omega \nu_T)\frac{\partial \omega}{\partial x_j}\right] + 2(1-F_1)\sigma_{\omega 2}\frac{1}{\omega}\frac{\partial k}{\partial x_i}\frac{\partial \omega}{\partial x_i} \tag{4.6}$$

with the wall distance controlfunction

$$F_1 = \tanh\left\{\left\{\min\left[\max\left(\frac{\sqrt{k}}{\beta^*\omega y}, \frac{500\nu}{y^2\omega}\right), \frac{4\sigma_{\omega 2}k}{CD_{k\omega}y^2}\right]\right\}^4\right\} \tag{4.7}$$

and the kinematic eddy viscosity ν_T:

$$\nu_T = \frac{a_1 k}{\max(a_1\omega, SF_2)} \tag{4.8}$$

whereas the second controlfunction F_2 is defined as

$$F_2 = \tanh\left[\left[\max\left(\frac{2\sqrt{k}}{\beta^*\omega y}, \frac{500\nu}{y^2\omega}\right)\right]^2\right] \tag{4.9}$$

Additional closure coefficients have to be provided to fully describe the equations. These remaining constants are based on empirical research. Values according to the paper [Rey+13] have been chosen for this work.
Overall the shear stress transport formulation provides the typically good $k-\omega$ accuracy in close wall distance, but also has $k-\epsilon-$like behavior in the freely flowing fluid. These properties and its numerical robustness made the $k-\omega-SST$ model a good choice for this work.

4.4 Maxwell equations

4.4.1 Set of equations

Because of the burning arc between the electrodes is caused by a transport of charge in form of electrons, electromagnetic forces are interact with the fluid. The high current imposes an electric field $\vec{E}$, which itself gives rise to an induced magnetic field $\vec{B}$.
To model these occurring electromagnetic phenomena, the Maxwell-equations are used.
Gauss's law for electrics:

$$\nabla \cdot \vec{E} = \frac{\rho_{charge}}{\epsilon_0} \tag{4.10}$$

Gauss's law for magnetism:

$$\nabla \cdot \vec{B} = 0 \tag{4.11}$$

Faraday's law of induction:

$$\nabla \times \vec{E} = -\frac{\partial \vec{B}}{\partial t} \tag{4.12}$$

Amperè's law:

$$\nabla \times \vec{B} = \mu_0 \vec{j} + \mu_0 \epsilon_0 \frac{\partial \vec{E}}{\partial t} \tag{4.13}$$

where μ_0 is the magnetic permeability, a physical constant defined as:

$$\mu_0 = 4\pi \times 10^{-7} H \cdot m^{-1} = 4\pi \times 10^{-7} N \cdot A^{-2}$$

ϵ_0 the permittivity of free space:

$$\epsilon_0 = 8.8541878176 \cdots \times 10^{-12} F \cdot m^{-1}$$

and ρ_{charge} the total charge density, which obeys to the charge conservation equation

$$\frac{\partial \rho_{charge}}{\partial t} + \nabla \cdot \vec{j} = 0 \tag{4.14}$$

Furthermore, a relationship for the current density $\vec{j}$ has to be given to fully close the system. Ohm's law provides this relationship as follows:

$$\vec{j} = \sigma \vec{E} + \sigma(\vec{U} \times \vec{B}) - \sigma(\frac{1}{n_e}\vec{j} \times \vec{B}) \tag{4.15}$$

where n_e is the electron density and σ the fluids electrical conductivity.

4.4.2 Potential formulation

By using vector identities, the Maxwell equations can be rewritten in a potential form to reduce complexity and the total amount of equations that have to be solved.
Gauss's law for magnetism (eq. 4.11) together with the Helmholtz's theorem, which states that every vector field $\vec{B}$ satisfying $\nabla \cdot \vec{B} = 0$ can be written as

$$\nabla \cdot \vec{B} = \nabla \cdot (\nabla \times \vec{A}) = 0$$

which yields

$$B = \nabla \times \vec{A} \tag{4.16}$$

where $\vec{A}$ is called a vector potential.
Plugging this into Faraday's law (eq. 4.12) gives further:

$$\nabla \times \vec{E} = -\frac{\partial}{\partial t}(\nabla \times \vec{A}) = -\nabla \times (\frac{\partial \vec{A}}{\partial t})$$

$$\nabla \times (\vec{E} + \frac{\partial \vec{A}}{\partial t}) = 0 \tag{4.17}$$

Using Helmholtz theorem a second time, regarding that any vector field $\vec{F}$, which satisfies $\nabla \times \vec{F} = 0$, can be written as

$$\vec{F} = \nabla \phi$$

for some scalar field ϕ, equation 4.17 yields

$$\vec{E} + \frac{\partial \vec{A}}{\partial t} = -\nabla\phi$$

or

$$\vec{E} = -\nabla\phi - \frac{\partial \vec{A}}{\partial t}$$

The electric and magnetic fields can now be described only in terms of their corresponding potentials:

$$B = \nabla \times \vec{A} \tag{4.18}$$

$$\vec{E} = -\nabla\phi - \frac{\partial \vec{A}}{\partial t} \tag{4.19}$$

4.4.3 Assumptions

Electro neutrality

Due to the nature of the considered problem, simplifications can be made to the set of Maxwell equations.
First of all, inside of the arc the Debye wavelength in the plasma is much smaller than the mean free path of an electron as described by [ST09]. An indefinitely small volume element (e.g. mesh cell) can thus be considered to be completely electrically neutral. Therefore, the absolute charge in such an element is zero. Taking this into account Gauss's law for electrics (eq 4.10) can be simplified to:

$$\nabla \cdot \vec{E} = 0 \tag{4.20}$$

and the charge conservation (eq. 4.14) reduces to

$$\nabla \cdot \vec{j} = 0 \tag{4.21}$$

Ampere's law

The second assumption that is made, is the validness of quasi-static electromagnetic phenomena. This assumption states, that the convection current can be neglected in comparison to the conduction current in Ampere's law (eq. 4.13). Thereby the temporal derivative of the electric field is negligibly small

$$\nabla \times \vec{B} = \mu_0 \vec{j} + \cancel{\mu_0 \epsilon_0 \frac{\partial \vec{E}}{\partial t}}^{0} \tag{4.22}$$

Ohm's law

Taking a closer look at Ohm's law (eq. 4.15) again, shows that additional terms can be neglected due to the problems nature.

$$\vec{j} = \underbrace{\sigma\vec{E}}_{\vec{j}_{\text{drift}}} + \underbrace{\sigma(\vec{U}\times\vec{B})}_{\vec{j}_{\text{ind}}} - \underbrace{\sigma(\frac{1}{n_e}\vec{j}\times\vec{B})}_{\vec{j}_{\text{Hall}}}$$

To evaluate which terms play a major role in Ohm's law, different characteristic parameters have been investigated.
At first the influence of the induced Hall current $\vec{j}_{\text{Hall}}$ has been examined. For an ionized gas (plasma) the Hall effect can be estimated by the Hall parameter according to [ST09] as

$$\beta_{\text{Hall}} = \frac{e\vec{B}}{m_e\nu}$$

where e is the elementary charge, m_e the electron mass and ν the electron to heavy-particle collision frequency defined by

$$\nu = \frac{n_e e^2}{m_e\sigma_{\text{arc}}}$$

with σ_{arc} as the characteristic electrical conductivity and therefore

$$\beta_{\text{Hall}} = \frac{e\vec{B}}{m_e\nu} = \frac{\vec{B}\sigma_{\text{arc}}}{n_e e} \tag{4.23}$$

With $\sigma_{\text{arc}} \approx 10^4[\frac{S}{m}]$ and an estimated value of $\vec{B} \approx 5\cdot 10^{-2}[T]$ the Hall parameter yields

$$\beta_{\text{Hall}} \approx 0.0312$$

Because $\beta_{\text{Hall}} < 1$ the induced Hall current can be neglected in this problem and thereby Ohm's law (eq. 4.15) reduces to:

$$\vec{j} = \sigma\vec{E} + \sigma(\vec{U}\times\vec{B}) = \sigma(\vec{E} + \vec{U}\times\vec{B}) \tag{4.24}$$

To determine if the induction current can be neglected as well, a dimensional analysis of the ratio between the electric field $\vec{E}$ and the term including the magnetic field, $\vec{U}\times\vec{B}$, is done. The order of magnitude for the electric field is linked to the prescribed electric potential $\phi = 24000[\frac{S}{m}]$ and the distance $l = 0.09[m]$ between the arcing contacts. Furthermore, the magnetic field has an estimated value of $\vec{B} \approx 0.05[T]$. As previously mentioned, the fluid

velocity is expected to be supersonic. It can therefore be approximated with the ideal gas formula for the speed of sound

$$c = \sqrt{\gamma R_s T} \tag{4.25}$$

With temperatures around $T \approx 20000[K]$, the specific gas constant $R_s = 83.028[\frac{J}{kg \cdot K}]$ and the heat capacity ratio $\gamma = 1.33$, the speed of sound calculates to:

$$c = 1486.12[\frac{m}{s}]$$

As for the Mach number holds $Ma = \frac{\vec{U}}{c} > 1$, the velocity is roughly estimated with $U_{approx} \approx 2000[\frac{m}{s}]$.
The ratio therefore holds

$$\frac{|\vec{U} \times \vec{B}|}{|\vec{E}|} = \frac{U_{approx} \cdot \vec{B}}{l \cdot \phi} \approx 0.046 \tag{4.26}$$

The influence of the of the induction current is therefore negligibly small. Finally Ohm's law can be simplified to

$$\vec{j} = \sigma \vec{E} \tag{4.27}$$

Furthermore, the influence of the magnetic fields temporal derivative has to be determined. By comparing the time-derivative to the conduction current, the so called screening parameter is obtained as agian described by [ST09]

$$\frac{|\frac{\partial \vec{B}}{\partial t}|}{|\nabla \times (\frac{1}{\mu_0 \sigma} \nabla \times \vec{B})|} \approx \frac{\mu_0 \sigma_{arc} \frac{\vec{B}}{t}}{\frac{1}{l}\frac{1}{l}\vec{B}} = \frac{\mu_0 \sigma_{arc} l^2}{t} \approx 0.2262 \tag{4.28}$$

where t is the ratio of the characteristic length l and the characteristic speed $\vec{U}_{approx}$. As the calculated ratio is smaller than unity, the time-variation in the magnetic field can be neglected as well. This conclusion yields for Faraday's law

$$\nabla \times \vec{E} = 0 \tag{4.29}$$

and for the previously derived potential formulation

$$\vec{E} = -\nabla \phi - \frac{\partial \cancelto{0}{\vec{A}}}{\partial t} = -\nabla \phi \tag{4.30}$$

4.4.4 Simplified set of equations

To finally derive a suitable potential formulation, here a short overview of the so far simplified of Maxwell equations:

$$\nabla \cdot \vec{E} = 0 \tag{4.31}$$

$$\nabla \cdot \vec{B} = 0 \tag{4.32}$$

$$\nabla \times \vec{E} = -\frac{\partial \vec{B}}{\partial t} \tag{4.33}$$

$$\nabla \times \vec{B} = \mu_0 \vec{j} \tag{4.34}$$

and additional

$$\vec{j} = \sigma \vec{E} \tag{4.35}$$

$$\nabla \cdot \vec{j} = 0 \tag{4.36}$$

$$\vec{B} = \nabla \times \vec{A} \tag{4.37}$$

$$\vec{E} = -\nabla \phi \tag{4.38}$$

Combining equations 4.35 and 4.38 yields for the current density

$$\vec{j} = \sigma \vec{E} = -\sigma \nabla \phi \tag{4.39}$$

Which together with equation 4.36 gives

$$\nabla \cdot \vec{j} = \nabla \cdot (\sigma \nabla \phi) = 0 \tag{4.40}$$

The simplified Ampere's law (eq. 4.34) with equations 4.37 and 4.39 gives

$$\nabla \times \nabla \times \vec{A} = -\mu_0 \sigma \nabla \phi$$

where $\nabla \times \nabla \times \vec{A}$ can be reformulated via the vector identity $\nabla \times \nabla \times \bullet = \nabla(\nabla \cdot \bullet) - \Delta \bullet$ to

$$\nabla(\nabla \cdot \vec{A}) - \Delta \vec{A} = -\mu_0 \sigma \nabla \phi \tag{4.41}$$

As the Helmholtz theorem again states that a vector is only uniquely defined if and only if its curl and its divergence are specified. To get a unique solution $\nabla \cdot \vec{A}$ has to be specified, which is done by choosing the divergence such that the most simplest solution for the equation above is obtained. This leads to the Coulomb gauge condition

$$\nabla \cdot \vec{A} = 0 \tag{4.42}$$

Finally a simplified set of equations has been derived, suitable specifically for our application. The number of equation has been reduced to two, such that now the only equations, which have to be solved are a Laplace equation for the electric potential ϕ and a Poisson equation for the magnetic vector potential. The complete set reads

$$\nabla \cdot (\sigma \nabla \phi) = 0 \tag{4.43}$$

$$\Delta \vec{A} = \mu_0 \sigma \nabla \phi \tag{4.44}$$

4.5 Physical properties

Due to the large expected temperature and pressure ranges and even different states of matter (gas, plasma) thermophysical properties have to be given as a function of e.g. pressure and temperature$f = f(p, T)$.
The in OpenFOAM implemented polynomial approach only offers a six coefficient approximation, which is not enough to capture the changing conditions over the whole range of temperatures.
To deal with this insufficiency, the thermodynamic package has been expanded with a bi-cubic table interpolation. A C++ based table manager has been implemented and integrated to OpenFOAM selection of thermophysical setups.
As the electrical conductivity σ is not part of OpenFOAMs transport properties by default, it has been added to the packages.

4.5.1 Table data

Different tables have been provided for properties like the dynamic viscosity ν or the thermal conductivity λ. The properties themselves are given as a

function of pressure p and temperature T with ranges

$$p \in [2Pa, 6 \cdot 10^7 Pa]$$
$$T \in [6K, 50000K]$$

The tables in principle are formatted as it can be seen in table 4.1.
In total, each of the tables stores the same 300 grid points for the pressure

	p
T	f(p,T)

Table 4.1: Property table layout

and the temperature. This results in a total of $300^2 = 90000$ property values.

4.5.2 Bi-cubic interpolation

The bi-cubic interpolation method is needed to obtain property values, that are between two table values. As the provided tables are rather sparse compared to their range, simply taking to closest value would cause erroneous solutions, as the properties would not be valid with respect to the actual temperature and pressure.
The bi-cubic interpolation for two dimensional arrays is evaluated by several one dimensional interpolations. A cubic interpolation was chosen, because it uses additional information about the derivative at the interval endpoints and thus provides a smooth spline.
The one dimensional interpolation approximates the function $f(x)$ by a third degree polynomial and its derivative:

$$f(x) = ax^3 + bx^2 + cx + d \tag{4.45}$$

$$f'(x) = 3ax^2 + 2bx + c \tag{4.46}$$

Assuming an interval of $x \in [0, 1]$ is given, the function and its derivative become

$$\begin{aligned} f(0) &= d \\ f(1) &= a + b + c + d \\ f'(0) &= c \\ f'(1) &= 3a + 2b + c \end{aligned}$$

at the endpoints.
This linear system of equations can be rewritten to

$$\begin{aligned} a &= 2f(0) - 2f(1) + f'(0) + f'(1) \\ b &= -3f(0) + 3f(1) - 2f'(0) - f'(1) \\ c &= f'(0) \\ d &= f(0) \end{aligned}$$

and the complete cubic interpolation formula can be obtained.
As no informations about the derivatives is given, the Catmull-Rom suggestion [CR74] was used for the splines. Assuming the point values p_{0-3} are given at respectively $x = -1$, $x = 0$, $x = 1$, and $x = 2$, a central linear interpolation can be applied

$$\begin{aligned} f(0) &= p_1 \\ f(1) &= p_2 \\ f'(0) &= \frac{p_2 - p_0}{2} \\ f'(1) &= \frac{p_3 - p_1}{2} \end{aligned}$$

Plugging these definitions into the cubic interpolation formula yields:

$$\begin{aligned} a &= -\frac{1}{2}p_0 + \frac{3}{2}p_1 - \frac{3}{2}p_2 + \frac{1}{2}p_3 \\ b &= p_0 - \frac{5}{2}p_0 + 2p_2 - \frac{1}{2}p_3 \\ c &= -\frac{1}{2}p_0 + \frac{1}{2}p_2 \\ d &= p_1 \end{aligned}$$

Finally the complete formula reads:

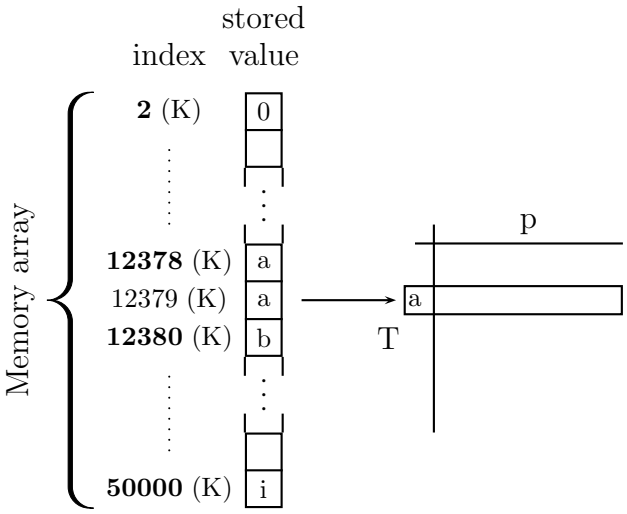
$$\begin{aligned} f(p_0,p_1,p_2,p_3,x) = (&-\frac{1}{2}p_0 + \frac{3}{2}p_1 - \frac{3}{2}p_2 + \frac{1}{2}p_3)x^3 \\ &+(p_0 - \frac{5}{2}p_0 + 2p_2 - \frac{1}{2}p_3)x^2 \\ &+(-\frac{1}{2}p_0 + \frac{1}{2}p_2)x \\ &+p_1 \end{aligned} \tag{4.47}$$

For the bi-cubic interpolation, four one dimensional interpolations are executed in x-direction at first. With these resulting four points, one last interpolation can then be evaluated in the y-direction.
As the first four interpolations require four grid points each, a total of 16 grid points and five cubic interpolations are required to perform a single bi-cubic interpolation.
To enable the interpolations at the table boundaries a two layer thick ghost-layer has been added to table, once it is stored to the table manager.

4.5.3 Fast table access

As all properties have to be updated at least once every simulated time step, the amount of time spend in the table interpolation is crucial to the performance of the solver.
A study of the running solver has shown, that most of the computational time is spend not in the interpolation itself, but rather in the routine searching for the correct table indices, which define the entry point for the interpolation. Usually classical divide and conquer algorithms, e.g. binary search, perform searches in sorted arrays in $\mathcal{O}(log(n))$. Though faster techniques are existing, those mostly go hand in hand with a larger overhead that negates their performance advantage on small arrays like the ones is this case (size about 300 entries).

To be as fast as possible, a memory intense storage solution was implemented. For upcoming explanations the temperature and pressure grid points will be referred to by Ψ, because the technique is equal for both ranges.
At first the complete table is stored in a matrix without any further modification. An array of the size of Ψ's range is created, with the effect that the array indices are uniquely linked to a grid point. The array is then filled with the corresponding indices of the original property table. As there are now probably more array entries than table grid points, all the array cells between table point A and table point B store the index of table point A. This is done because the bi-cubic interpolation needs the next lower table



Note: table provided data is displayed bold

Figure 4.1: Sketch of linking array mechanic

entry to calculate the desired interpolation point. A supporting graphical explanation is given in figure 4.1.
With the help of this additional linker between interpolation point and corresponding index, a table index can be found in constant time ($\mathcal{O}(1)$).

4.6 Radiation

To calculate the heat radiation inside the domain the discrete ordinates (DO) method has been used. This method solves the radiative transfer equation (RTE) for a finite number of solid angles. Because each additional solid angle has a unique vector direction $\vec{s}$ and therefore its own RTE that has to be solved, the used number of angles is a trade-off between accuracy and speed. The RTE reads for radiation intensity I_λ

$$\frac{dI}{d\vec{s}} = \rho k_\lambda (B_\lambda(T) - I) \tag{4.48}$$

where λ is the wavelength, k_λ the absorption cross section and $B_\lambda(T)$ the Planck function

$$B_\lambda(T) = \frac{2hc^2}{\lambda^5} \frac{1}{e^{\frac{hc}{\lambda k_B T}} - 1} \tag{4.49}$$

with k_B the Boltzmann constant, h the Planck constant and c the speed of light. The RTE equation above is valid for a non-scattering medium in LTE.

The DO method is already available in OpenFOAM, but is not working properly on axisymmetric meshes as it causes numerical errors at the symmetry axis. These errors were only caused by calculated intensity rays, that were "crossing" the axis. If the DOs origin is resetted onto the axis, all rays being shot below no longer contribute to the radiation. By using this property, the erroneous rays can be neglected and thus the DO method works again for axisymmetric meshes.
Investigations have shown, that the errors are most likely caused by the *wedge* boundary condition, which is required for the creation of an axisymmetric mesh in OpenFOAM. Therefore the error probably lies deeper in the source code and might not be completely solved by the described workaround.
Nevertheless the results in comparison with the also available P1 method showed a qualitative match, which is why the DO method has been used in this work after all.

4.7 Ablation

As already mentioned in section 4.2, the ablation phenomena at the nozzle is modeled by two source terms: a mass source and an energy source.
Based on the paper of Iordanidis and Franck [IF08], the mass source S_{abl} is defined as

$$S_{abl} = \dot{m} = \frac{Q_{inc}}{dE_{\mathrm{PTFE}(T_w,p)}} \tag{4.50}$$

where Q_{inc} is the incident heat radiation, calculated by the discrete ordinate model described in the section above and $dE_{\mathrm{PTFE}(T_w,p)}$ is the heat that is needed to vaporize and heat up the PTFE to the effective wall temperature T_w. The pressure p hereby is the local pressure at the nozzle boundary. Any kind of heat losses caused by radiation penetration are omitted in this model due to their minor impact.
The energy source term is furthermore calculated by

$$\dot{e} = \dot{m} \cdot E_{\mathrm{PTFE}}(T_w, p) \tag{4.51}$$

Though this model is a quite simplified approach and can not describe the ablation process at the nozzles boundary layer, it gives good results on the macroscopic scale.
PTFE vaporizes around $T_{vap} \approx 1000[K]$, where most of the ablation is caused by hard radiation of high UV frequencies. In reality this causes the existence of several kinetic non-equilibrium layers close to the boundary of the nozzle.

These layers interact differently with the incident radiation and can even shield the nozzle from it. According to [IF08], this effect can be considered by assuming an effective ablation temperature of $T_w = 3500[K]$.
To make the model configuration available from the case files, the file *ablationProperties* in the folder constant is parsed module initialization. The following layout is required:

Listing 4.1: Example of a ablationProperties file

```
1 ablation          on;
2 ablationModel     singleCompound;
3
4 singleCompoundCoeffs
5 {
6     patch         nozzle; //ablation patch name
7     Tw            3500;   //effective ablation temp.
8     dHvap         1e6;    //evaporation enthalpy
9 }
```

The model *singleCompound* is currently the only available model. It only considers ablation effects where the same material is ablated, as it is aleady present in the domain.

Chapter 5

Numerics

This section will deal with the numerical methods used for the discretization of the physical problem described in chapter 4.

5.1 Finite volume method

The MHD equations are discretized using the finite volume method. By using this method the domain is divided into several contiguous control volumes. The control volumes themselves are polyhedral cells with an arbitrary number of faces, each again with an arbitrary number of vertices. While the number of neighboring cells can vary, each face only connects two cells at a time.
The method makes use of Gauss's divergence theorem, by converting the divergence terms of the conservation laws to surface integrals. These terms can then be evaluated as fluxes over the surface boundaries. According to the mass conservation, the sum of fluxes entering the finite volume has to be equal to the sum of fluxes leaving adjacent volumes. Thus finite volume methods are conservative methods by construction.

5.2 Weak coupling

As the electromagnetic effects are assumed to be quasi-static, a weakly coupled approach was chosen. This results in a segregated solver setup. The solver main loop can be seen in figure 5.1. At first the Navier-Stokes equations are solved. Afterwards all fluid properties are updated, based on the results of the new calculated time step. With these new values for σ the two potential equations are solved and the electric and magnetic fields are calculated by equations 4.43 and 4.44.

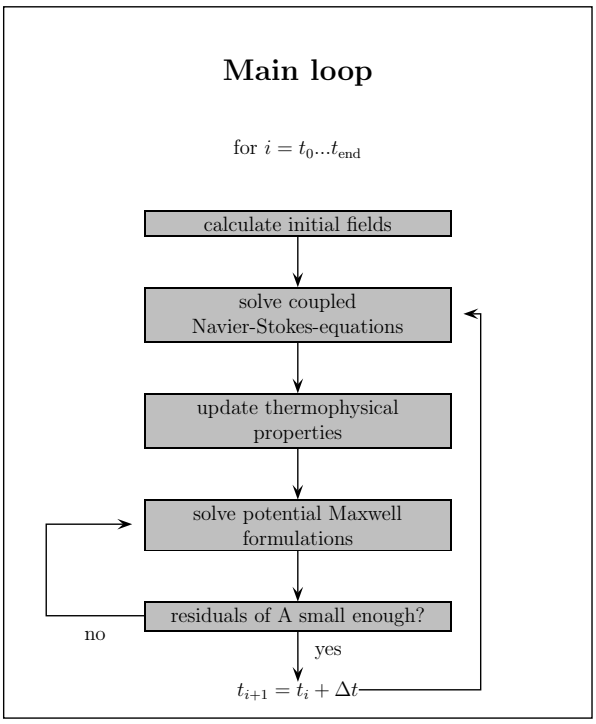


Figure 5.1: Solver loop

5.3 Kurganov-Tadmor scheme

When it comes to solving the Navier-Stokes equations, a default OpenFOAM solver, called *rhoCentralFoam*, has been modified to solve the desired equations.
As the the fluid is compressible, properties are not only transported by the flow ϕ, but rather by the propagation of waves.
rhoCentralFoam therefore uses the Kurganv-Tadmor (KT) scheme [KT00]. This scheme is a second-order central differencing scheme. Its formulation is a modification of the Nessyahu-Tadmore (NT) scheme [NT90].
It offers great advantages over common upwind schemes, as being Riemann solver free and having a smaller numerical viscosity than the older NT scheme. Unlike most other central schemes, it can also be formulated in a semi-discrete way, which has proven to have better resolution at discontinuities for parabolic convection-diffusion equations. As the Navier-Stokes equations for newtonian fluids are exactly this kind of equations, *rhoCentralFoam* uses this semi-discrete formulation.

Applying the finite volume method on the convective terms $\nabla \cdot (\vec{U}\Psi)$ of

the Navier-Stokes equations (4.1) - (4.3) yields by integration over a control volume V and additional linearization:

$$\int_V \nabla \cdot (\vec{U}\Psi) dV = \Sigma_f [S_f \cdot \vec{U}_f] \Psi_f \approx \Sigma_f \phi_f \Psi_f \tag{5.1}$$

Where Ψ represents the transport variable of the corresponding conservation equation and S_f the faces area vector. The faces area vector which always pointing outwards with respect to the cell center. $S_f \cdot \vec{U}_f$ is more commonly defined as ϕ_f, the volume flux through a face. Index $\bullet_f$ denotes face values in this context.
As the flux is now given on the faces, rather than the cell centers, some kind of interpolation has to be applied. The suggested interpolation by KT splits the flux in two directions corresponding to be in- or outwards pointing related to the faces owner cell. To be more specific the fluxes f are either pointing in direction of S_f (f_+) or in the opposing direction (f_-).
The resulting discretized numerical flux for compressible flow reads:

$$\Sigma_f \phi_f \Psi_f = \Sigma_f (\alpha \phi_{f_+} \Psi_{f_+} + (1-\alpha) \phi_{f_-} \Psi_{f_-} + \omega_f (\Psi_{f_-} - \Psi_{f_+})) \tag{5.2}$$

As the KT scheme is a strictly central scheme $\alpha = \frac{1}{2}$ is defined to achieve a symmetric weighting. In general other weighting functions, like the one used in the Kurganov-Noelle-Petrova scheme [KNP01], are possible.
The first two terms on the right hand side of equation 5.2 represent the flux evaluations in f_+ and f_- directions. The last term is required in case the convection term is part of the total derivative. It adds an additional diffusion term depended on ω_f, which for the KT scheme is defined by:

$$\omega_f = \alpha \cdot max(\phi_{f_+}, \phi_{f_-}) \tag{5.3}$$

The volumetric fluxes $\Psi_{f_\pm}$ are defined by

$$\Psi_{f_\pm} = max(c_{f_+}|S_f| \pm \phi_{f_+}, c_{f_-}|S_f| \pm \phi_{f_-}, 0) \tag{5.4}$$

This relationship is derived from the maximum absolute value of the Jacobian of the flux. The flux is therefore associated with the local propagation speed. Apart from these CFL related properties, no further characteristics are used in the KT scheme.
The $c_{f_\pm}$ in eq. 5.4 is the speed of sound, which for ideal gas is defined by the relation

$$c_{f_\pm} = \sqrt{\gamma R T_{f_{pm}}}$$

with γ as the ratio of specific heats, R the specific gas constant and $T_{f_\pm}$ the temperatures of the faces neighbor cells.

The KT scheme is of second order in space, but according to Godunovs theorem [Wes09] a first order scheme must be used to avoid unphysical oscillations at discontinuities. To switch from high order schemes to a first order scheme, a flux limiter is used for flux interpolation.
In this paper the van Leer limiter [VL74] is used. It is a symmetric and total variation diminishing (TVD) limiter. This means the limiter function

$$\Phi_{\text{van Leer}}(r) = \frac{r + |r|}{1 + |r|} \tag{5.5}$$

has the following symmetry property

$$\frac{\Phi(r)}{r} = \Phi(\frac{1}{r})$$

which states that limiting operations for forward and backward gradients operate equally. Using a TVD limiter adds a monotonicity preserving property at the discontinuities, preventing the creation of oscillations.

Like all currently available OpenFOAM solvers, the KT scheme has been implemented in a segregated manner to solve the Navier-Stokes equations. The equations are therefore solved one by one and not as a whole system. This technique generally provides solutions faster at the costs of being not as accurate, because the coupling between the equations is limited.

5.4 Electromagnetic loop

As the potential equations were derived under the premise of quasi-static electromagnetic phenomena, a pseudo time stepping was implemented. A first solution is thereby calculated from an initial guess. The obtained results are then used as new initial values for the next solver iteration.
This procedure is continued until a certain convergence criterion is satisfied. A sketch of the process can be found in figure 5.1.
Because the potential equations 4.43 and 4.44 have been derived from the Maxwell equations (4.10 - 4.13), the Maxwell equations can be used to evaluate the solutions for ϕ_{elec} and $\vec{A}$.
Gauss law for magnetics (4.11)

$$\nabla \cdot \vec{B} = 0$$

is clearly a good option for this kind of convergence criterion, as it can directly be computed from $\vec{A}$. The vector potential $\vec{A}$, again, is just dependent on ϕ_{elec} and the electrical conductivity σ. This makes Gauss law for magnetics a good representative for the overall quality of the solution.

Chapter 6

Validation

6.1 Case description

To validate the electromagnetic solver part, an infinite electric rod was examined. The rod itself consisting of a high conductive material, is surrounded by a second insulating substance. Non of the materials needs further specifications as this test case only computes the electromagnetic fields and thus, despite from the electrical conductivities, neither fluid nor solid properties have an influence on the solution. Figure 6.1 shows the methodic rod setup.

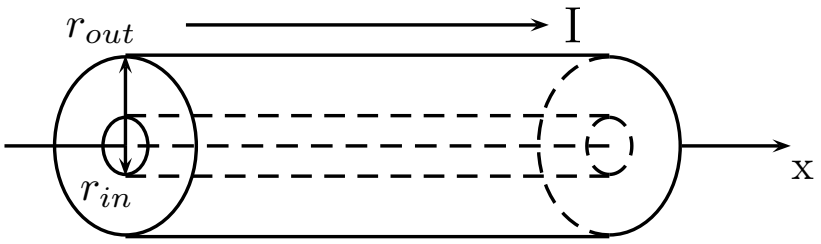


Figure 6.1: Sketch of electric rod

A constant current density corresponding to a current of $I = 600A$ is imposed on the left side of the high conducting part. For the electrical conductivity $\sigma_{in} \gg \sigma_{out} \Leftrightarrow 2700\frac{A}{Vm} \gg$ 1e-05 $\frac{A}{Vm}$ has been set. These values correspond to argon gas at 10500 Kelvin and ambient temperature (300 Kelvin). As the energy conservation is not evaluated in this test case, $\sigma_{in,out}$ is constant in (pseudo-)time. All other parameters are listed in table 6.1.

I [A]	r_{in} [m]	r_{out} [m]	σ_{in} [A/(Vm)]	σ_{out} [A/(Vm)]	L [m]
600	0.001	0.1	2700	1e-05	0.01

Table 6.1: Test case setup

6.2 Analytical solution

Analytical solutions to the problem are given separately for the inside and the outside of the rod. For $r \leq r_{in}$ the solution in cylindrical coordinates for the vector potential $\vec{A}$ and the magnetic field $\vec{B}$ are given by:

$$\vec{A}_{x,in} = \vec{A}_0 - \frac{\mu_0 \vec{j}_x r^2}{4} \tag{6.1}$$

$$\vec{B}_{\theta,in} = \frac{\mu_0 \vec{j}_x r}{2} \tag{6.2}$$

where for the outer domain $r > r_{in}$ holds

$$\vec{A}_{x,out} = \vec{A}_0 - \frac{\mu_0 \vec{j}_x r_0^2}{2}[\frac{1}{2} + ln(\frac{r}{r_0})] \tag{6.3}$$

$$\vec{B}_{\theta,out} = \frac{\mu_0 \vec{j}_x r_0^2}{2r} \tag{6.4}$$

The current density $\vec{j}_x$ is directly given in this example because the current I is prescribed:

$$\vec{j}_x = \frac{I}{A} = \frac{I}{\pi r_{in}^2} \quad [\frac{A}{m^2}] \tag{6.5}$$

$\vec{A}_0$ is the maximal value of $\vec{A}$, which emerges at $r = 0$. As this reference value is not given, numerical results for $\vec{A}_0$ have been used for later comparisons of radial plots.

6.3 OpenFOAM setup

Because of the rods radial symmetry an axisymmetric mesh has been created for the simulations. The mesh creation was done by OpenFOAMs internal mesher *blockMesh*. To define an 2D-axisymmetric mesh the wedge is recommended to have an angle of 5° and additionally the front and back planes must have boundary conditions of type *wedge*. Furthermore, the boundary condition on the axis has to be of type *symmetry*.

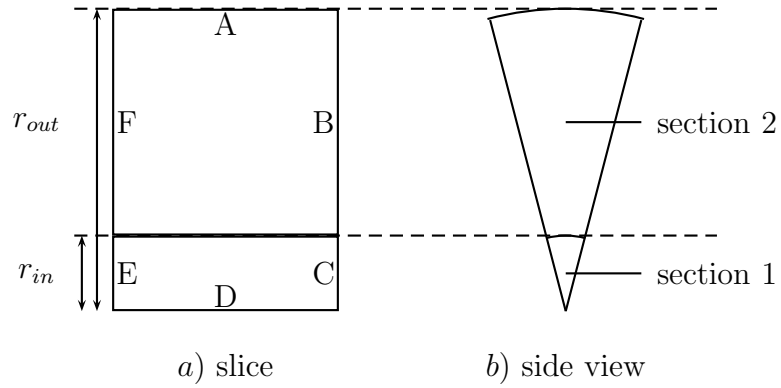


Figure 6.2: Axisymmetric sketch

	A	B	C	D	E	F
$\vec{A}$	$\frac{\partial \vec{A}}{\partial \vec{n}} = 0$	$\frac{\partial \vec{A}}{\partial \vec{n}} = 0$	$\frac{\partial \vec{A}}{\partial \vec{n}} = 0$	sym.	$\frac{\partial \vec{A}}{\partial \vec{n}} = 0$	$\frac{\partial \vec{A}}{\partial \vec{n}} = 0$
ϕ	$\frac{\partial \phi}{\partial \vec{n}} = 0$	$\frac{\partial \phi}{\partial \vec{n}} = 0$	$\phi = 0$	sym.	$\phi = 707$	$\frac{\partial \phi}{\partial \vec{n}} = 0$

Table 6.2: Boundary conditions

A sketch of the computational domain can be seen in figure 6.2. Boundary conditions have been applied according to previous descriptions and can be seen in table 6.2. $\vec{n}$ thereby denotes the outward pointing normal vector. The different notations for the boundary conditions are linked to figure 6.2. The used mesh consists strictly of rectangular cells. It is made of 21250 cells equally spaced in x-direction and with increasing cell length in radial direction because results further away from the rod won't differ a lot and are generally expected to be less significant.

6.4 Results

In this section the computed results will now be compared with the analytical solution. The computation itself is performed within three pseudo-timesteps and takes $t_{calc} = 0.52s$. Obtained results are plotted against the corresponding analytical solutions in figures 6.3 and 6.4. Because the radius of the domain ($r_{out} = 0.1\,m$) is large compared to the rods radius ($r_{in} = 0.001\,m$), where the actual significant results are obtained, the big plots show the data only in range of $0 < r < 0.0012$. The qualitative results for the complete domain can additionally be taken from the nested graphs. The overall results show a good match between the analytical and numerical solution.

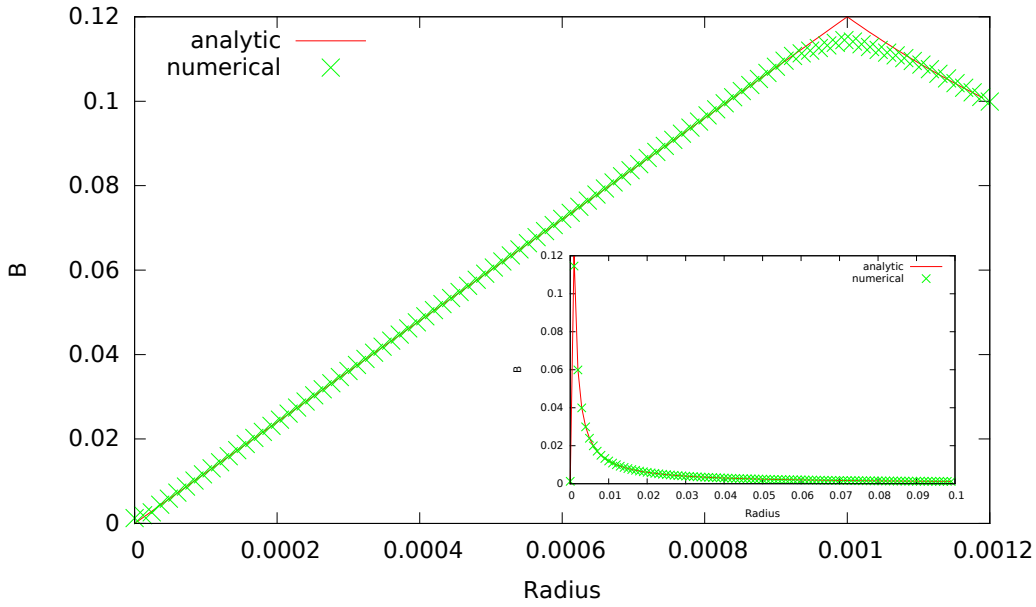


Figure 6.3: Magnetic field $\vec{B}$

The combination of the analytic solutions for the magnetic field $\vec{B}$ results in a sharp bend at $r = r_{in} = 0.001[m]$. Numerically the level of sharpness can't be reproduced. The solution is smeared at the bend, which is caused by numerical dissipation added by the solution scheme. A special scheme, able to handle the jump in σ, could improve this solution.
Another small divergence can be found close to the axis of symmetry. The numerical solution does not reach zero completely, like the analytical does. This is most likely caused by the iterative solver method used for the Poisson equation, as such methods would never give exactly zero as a result.

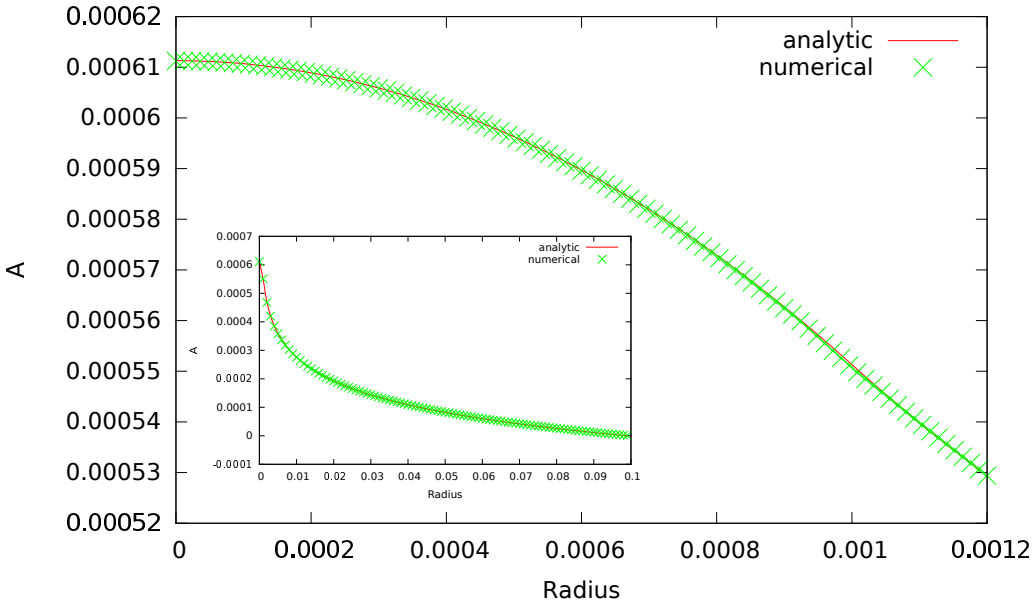


Figure 6.4: Vektor potential $\vec{A}$

Chapter 7

Results

7.1 Case setup

As already described several times, the case the has been solver created for, concerns a burning arc between two electrodes. In the center of the domain the arc is surrounded by a PTFE nozzle, which will vaporize during the simulation process.
A two dimensional slice of the domain can be seen in figure 7.1. The overall domain is $l_{domain} = 0.13m$ long and has a radius of $r_{domain} = 0.034m$. In the section of the nozzle the radius shrinks to $r_{nozzle} = 0.005m$. The distance from the electrodes to the nozzles is $\Delta l_{EN} = 0.02m$, where the distance between both electrodes is $l_{arc} = 0.09$. These parameters and a few others are also given in table 7.1.

The boundary conditions that have been applied according to table 7.2. The cathode has been divided into two separate sections. One section is the part of the cathode surface where the arc is supposed to be burning (denoted by cathode$_{arc}$). The other section cathode$_{top}$ describes the remaining top

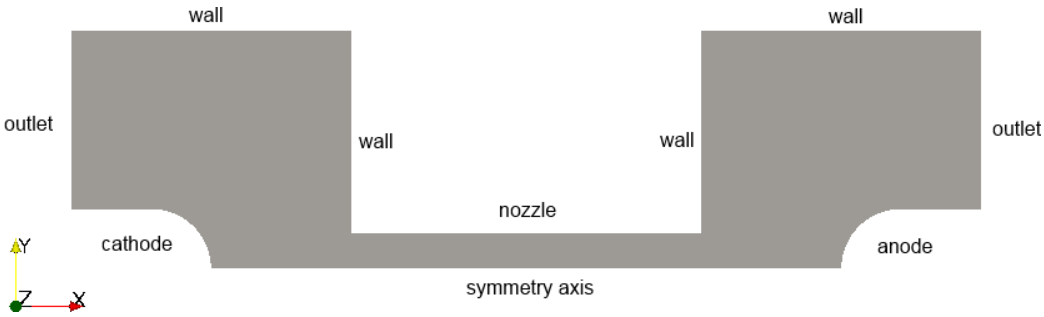


Figure 7.1: Sketch of the domain

r_{nozzle}	r_{domain}	$r_{electrode}$	l_{arc}	l_{domain}	h_{outlet}	Δl_{EN}
0.005	0.034	0.0085	0.09	0.13	0.0255	0.02

Table 7.1: Length scales of the domain in [m]

	cathode_{arc}	cathode_{top}	anode	nozzle	outlets	walls
ϕ	$\nabla\phi = -2e4^{(*)}$	$\nabla\phi$=0	0	$\nabla\phi$=0	$\nabla\phi$=0	$\nabla\phi$=0
A	∇A=0	∇A=0	∇A=0	∇A=0	0	∇A=0
$\vec{v}$	0	0	0	0	$\nabla\vec{v}$=0	0
p	∇p=0	∇p=0	∇p=0	∇p=0	p_a	∇p=0
T	∇T=0	∇T=0	$\nabla T = 0$	∇T=$0^{(*)}$	T_a	∇T=0

Table 7.2: Boundary conditions

part of the cathode. This division was done because the separate boundary conditions tend to give better results for the electromagnetic equations than a uniformly induced current over the whole surface.
The boundary condition for the electric potential ϕ at the cathode_{arc} is denoted with a $(*)$, because the gradient is not fixed at all times. To slowly create the arc, the gradient is ramped up over $t = 0.005s$. This procedure helped the arc ignition as the gas gets slowly heated and has more time to expand. Additionally the jump at the created discontinuity is not as big as without the ramping procedure.
For the temperature at the nozzle, the zero-gradient boundary condition is also denoted with a star, as the maximal temperature at this boundary can never be higher than the effective ablation temperature T_w. Therefore a zero-gradient boundary condition, but it is manually limited to values $T \leq T_w$.

The computational mesh consists of $\sim$ 78000 cells, which all hexahedral except from the cells at the axis of symmetry. Boundary layers have been added to all solid surfaces, to resolve turbulence and ablation phenomena correctly. An excerpt from mesh around the cathode can be seen in figure 7.2. This exact mesh has been used for previous simulations with a similar setup in the commercial simulation software called FLUENT. As these simulations gave correct results, the same mesh has been used in this case.

7.2 Arc ignition

To handle the arc ignition, a column with a high electrical conductivity σ_{init} is prescribed in the section where the arc is supposed to be burning. The rest of the domain is initialized with the corresponding values to ambient

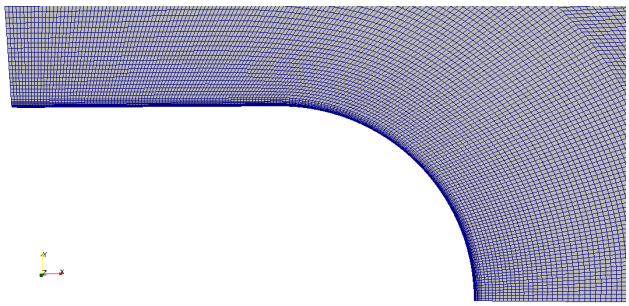


Figure 7.2: Mesh at cathode

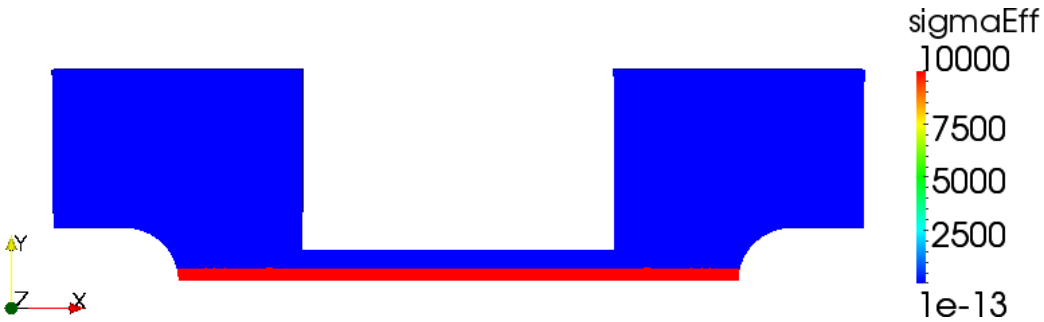


Figure 7.3: Prescribed σ_{init}

pressure and temperature.

As σ generally is computed by the thermophysical package, a boolean is stored in each cell acting as a switch. When the relation $\sigma_{thermo} > \sigma_{init}$ holds for the first time, this boolean is set *true* and from that point on, the σ in that specific cell will be calculated only by the thermophysical package. The prescribed $\sigma_{init} = 10000\ [\frac{S}{m}]$ corresponds to temperatures below the expected results, so the prescribed column will vanish in time.

7.3 Simulation results

The simulation was run in parallel using 6 cores on a network cluster.
Even though a first order implicit solver for the time derivative was used, the time step had to be restricted to CFL numbers below 0.8 to prevent the solver from crashing.
Normally implicit methods are not restricted by the CFL condition, but the segregated implementation of the KT scheme makes it require smaller time steps as it is much more error prone. Due to the small cell size Δx and the CFL restriction, timesteps have been of the order 10^{-8}, once the arc is burn-

ing. Nevertheless most of the simulations crashed around the $t = 0.00043s$ mark. The presented results is this chapter are therefore given at this time. The simulation took a total computing time of 7h, 30min and 6 seconds.

The fatal error is related to the computation of the electrical field. The error source will be explained after the examination of the other obtained fields.
The velocity fields are given in figure 7.4. As expected the simulation results in high velocities at the nozzles outlets with a Mach number of $Ma \approx 0.8 < 1$. This an expected result, as the induced current density at the cathode has not yet reached its maximal value.
The turbulent flow over the electrodes also explains the alternating signs

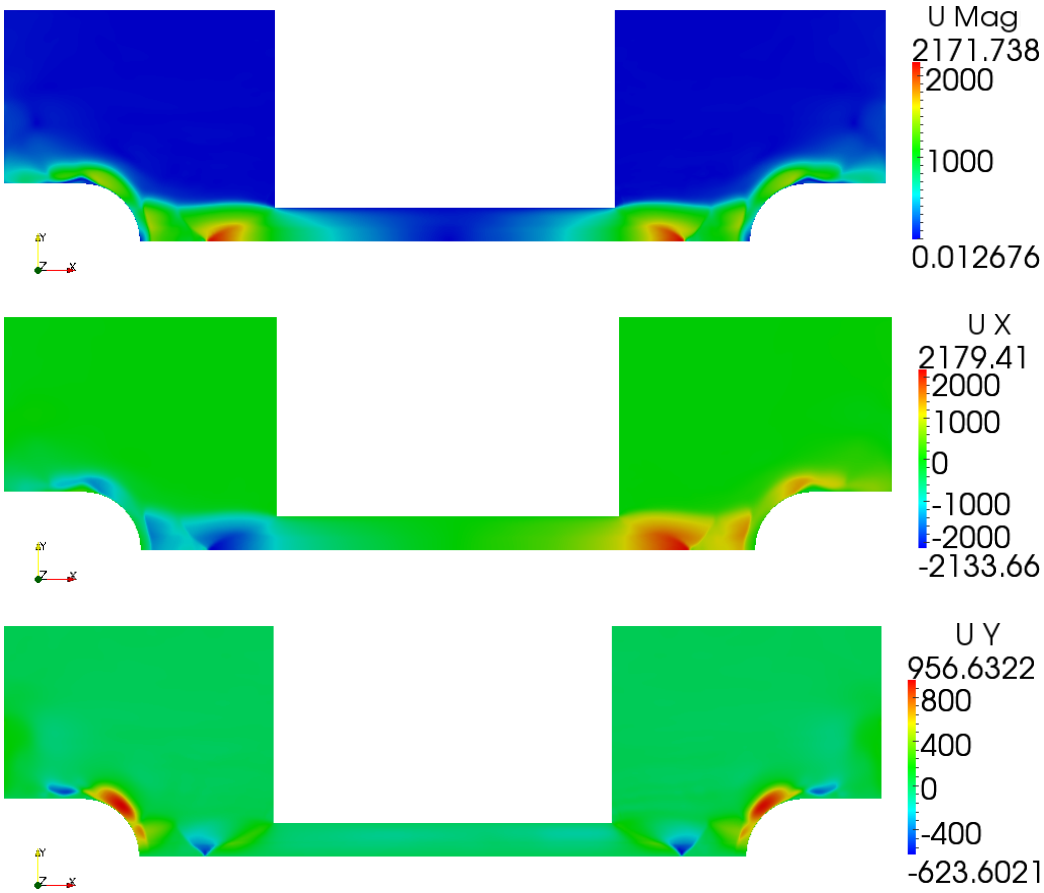


Figure 7.4: Results for $\vec{U}$ at $t = 0.00043s$

close to the surfaces. Ablation caused velocities can not be seen in any of the fields in figure 7.4 as the ablation effects are not big enough at the currently governing temperatures.

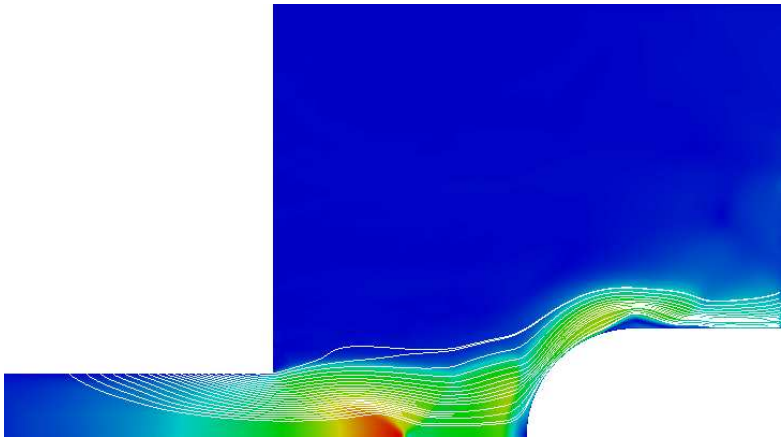


Figure 7.5: Streamlines at $t = 0.00043s$

However the streamlines in figure 7.5 show that a flux is going from the nozzle to the outlets and thus ablation effects are present.
The pressure field in figure 7.6 supports this statement, as its values are significantly higher at the nozzles boundary, which is caused by the injected mass.

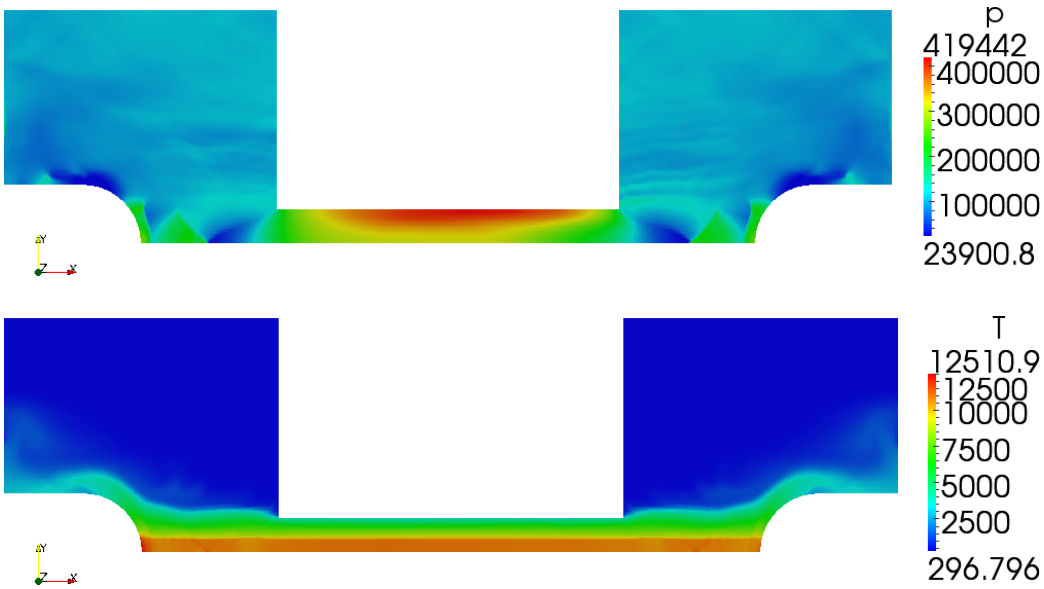


Figure 7.6: Results for the pressure p and temperature T at $t = 0.00043s$

The temperature also behaves as expected. It shows the burning electric arc in the hot region right at the axis of symmetry. The rest of the domain

is either heated by the flows transported heat or by heat radiation.

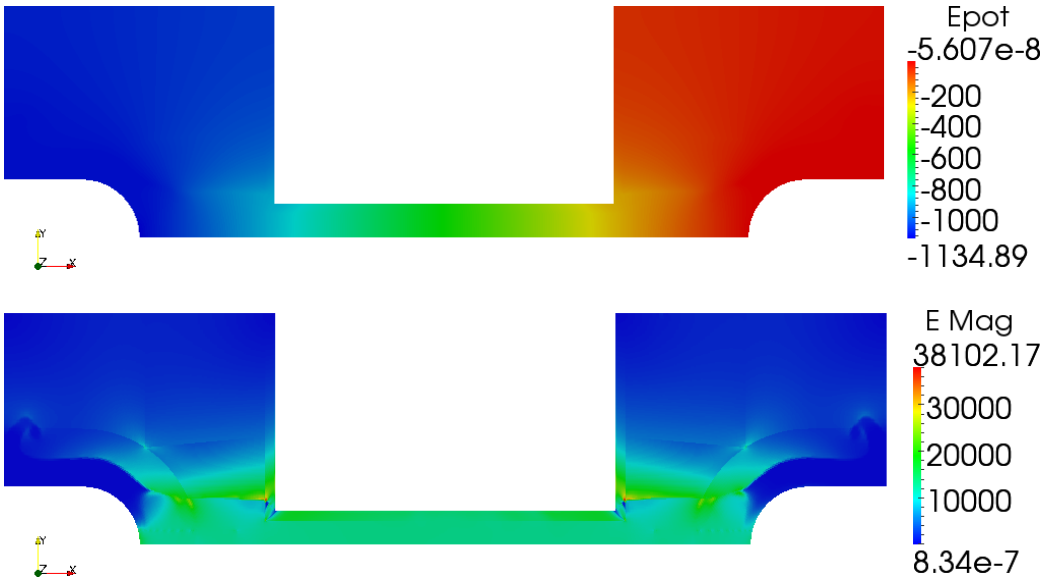


Figure 7.7: Results for the electric field $\vec{E}$ and the electric potential ϕ (Epot) at $t = 0.00043s$

The fatal error occurred in the calculation of the electric field $\vec{E}$.
As it can be seen in figure 7.7 the solution for the electric potential ϕ appears still smooth and has no artifacts. The electric field on the other hand, which is computed by equation 4.38 as $\vec{E} = -\nabla\phi$ is not smooth at all. The solutions appears to be extremely mesh dependent as several mesh attributes can directly be recognized (boundary layer, mesh bending around the electrodes, etc...). Several gradient schemes have been tested, but did not show any effect worth mentioning. Even special skewness correcting schemes failed.
With increasing changes in the electrical conductivity the solution becomes more and more unstable. These changes are caused by the joule heating and the ablation of hot PTFE into the domain. The unstable solution for the electric field causes irregularities in the field, especially close to the boundaries. At some point these irregularities blow up and all terms linked to the electric field yield faulty results.

Chapter 8

Conclusion

OpenFOAMs abilities towards the modeling of electric arcs have been examined in this work.
To model the physical problem, the MHD equations have been modified using a potential formulation for the Maxwell equations. This formulation has been derived uniquely for this problems characteristics.
A default solver for compressible, viscous and transonic flow has then been extended to solve for these additional equations.
As OpenFOAM was not able to provide a proper method to evaluate thermophysical properties for the large range of expected temperature and pressure values, a fast bi-cubic table interpolation has been implemented as a new module available in the thermophysical setup.
Following investigations have shown that the DOM radiation model was not working on axisymmetric meshes due to numerical errors regarding certain boundary conditions. The radiation model has then been analyzed and a solution/workaround was implemented.
Finally the a required ablation module has been added to OpenFOAM, providing simple top level access and easy expandability.
Sadly, in the end it was not able to achieve comparable results to other available CFD-tools. It was not able to prevent errors in the calculation of the electrical field on more complex meshes with a changing electrical conductivity.

For future work, developing a method to prevent these problems would be a necessary step to start with. Additionally further investigations on the discrete ordinates method for axisymmetric meshes, have to be done. As the problem appears to be deeply nested into the source code, this can be a difficult task.
OpenFOAM has just recently been updated with basic abilities to solve cou-

pled equations. Developing a fully coupled solver would therefore be a challenging, but interesting task. It also has the potential to improve the solvers stability and thus allowing larger time steps than segregated approaches do. All in all OpenFOAM offers a good foundation of abilities. The advantage of being completely open source gives it a unique position among the other CFD-tools on the market.

Bibliography

[CR74] Edwin Catmull and Raphael Rom. "A class of local interpolating splines". In: *Computer aided geometric design* 74 (1974), pp. 317–326.

[IF08] A A Iordanidis and C M Franck. "Self-consistent radiation-based simulation of electric arcs: II. Application to gas circuit breakers". In: *Journal of Physics D: Applied Physics* 41.13 (2008), p. 135206.

[KNP01] Alexander Kurganov, Sebastian Noelle, and Guergana Petrova. "Semidiscrete central-upwind schemes for hyperbolic conservation laws and Hamilton–Jacobi equations". In: *SIAM Journal on Scientific Computing* 23.3 (2001), pp. 707–740.

[KT00] Alexander Kurganov and Eitan Tadmor. "New high-resolution central schemes for nonlinear conservation laws and convection–diffusion equations". In: *Journal of Computational Physics* 160.1 (2000), pp. 241–282.

[NT90] Haim Nessyahu and Eitan Tadmor. "Non-oscillatory central differencing for hyperbolic conservation laws". In: *Journal of computational physics* 87.2 (1990), pp. 408–463.

[Rey+13] Dasia A Reyes et al. "Computations of High-Lift Wing Configuration on Unstructured Grids Using k-ω Models". In: *Journal of Aircraft* 50.6 (2013), pp. 1682–1695.

[ST09] Margarita Sass-Tisovskaya. *Plasma arc welding simulation with OpenFOAM*. Chalmers University of Technology, 2009.

[VL74] Bram Van Leer. "Towards the ultimate conservative finite difference scheme. II. Monotonicity and conservation combined in a second order scheme". In: *Journal of Computational Physics* 14 (1974), pp. 361–376.

[Wes09] Pieter Wesseling. *Principles of computational fluid dynamics*. Vol. 29. Springer Science & Business, 2009, pp. 339–346.