POLITECNICO DI TORINO

Master of Science in Aerospace Engineering

Department of Mechanical and Aerospace Engineer

Master Thesis

Electromagnetic Wiping System for Hot-Dip Coating

Politecnico di Torino

von KARMAN INSTITUTE FOR FLUID DYNAMICS

Supervisors:

Politecnico di Torino

Prof. Sandra Pieraccini

von Karman Institute for Fluid Dynamics

Ing. Fabio Pino

Prof. Miguel Alfonso Mendez

Candidate:

Andrea Palmarini

October 2021

Abstract

Galvanizing is the process of applying a zinc coating to iron and steel to prevent corrosion. Hot-dip galvanizing, electrogalvanizing, and zinc spraying are three of the most used methods for adding zinc to iron and steel. The continuous sheet hot-dip technique is used to coat the majority of galvanized products

Jet wiping is a process of hot-dip galvanization and is used to regulate the thickness of a liquid coat on a moving substrate by wiping the free surface. Nevertheless, some undulation phenomena occur over the zinc film due to the jet wiping action; among different actuators used to cope with this issue, the electromagnetic (EM) control system seems a viable solution. Hence, a reduced-order model capable of accounting for magnetohydrodynamic effects has been elaborated at von Karman Institute.

The project presented in this thesis has been part of the validation campaign of the numerical results of the electromagnetic model mentioned above and it has been undertaken in Elmer. This freeware software has been developed by the Finnish IT centre for science (CSC) and it is based on the finite element method (FEM). Elmer has offered an appealing opportunity being a multiphysics software, user-friendly thanks to its Graphical User Interface (GUI) version and easy to learn as CFD tool.

The numerical results obtained by the simulations in Elmer have demonstrated the accuracy of the EM model both with a uniform and Gaussian magnetic field.

Acknowledgments

Before proceeding with my work, I would like to dedicate a few lines to all those close to me in this journey of personal and professional growth. First of all, I would like to thank the supervisors of this thesis, Professor Sandra Pieraccini, Professor Miguel Alfonso Mendez and Engineer Fabio Pino, who provided me with invaluable guidance and corrections throughout writing my thesis teaching me to have a more open critical mind. I would like to thank my whole family, who has supported me in every moment since I have started university. They have always taught me to believe in my capabilities and not give up in front of difficulties. Eventually, I want to express my gratitude to the other STP students of von Karman Institute to have created a cosy and friendly working environment.

Contents

List of symbols and units

Dimensional Quantities

$\mathbf{U}$	Velocity	$[m/s]$
U_p	Strip velocity	$[m/s]$
σ_m	Electrical conductivity	$[S/m]$
μ_l	Dynamic viscosity for zinc	$[Pa \cdot s]$
μ_g	Dynamic viscosity for air	$[Pa \cdot s]$
$\mu_{l,g}$	Dynamic Viscosity for zinc and air	$[Pa \cdot s]$
$\mathbf{B}$	Magnetic induced flux density	$[T]$
$\mathbf{B}^i$	Magnetic induction flux density	$[T]$
$\mathbf{B}^e$	Magnetic external flux density	$[T]$
$\mathbf{H}$	Magnetic field	$[H/m]$
$\mathbf{H}^i$	Induced magnetic field	$[H/m]$
$\mathbf{M}$	Magnetization	$[A/m]$
$\mathbf{J}$	Current density	$[A/m^2]$
$\mathbf{J}_s$	Surface current density on the interface	$[A/m^2]$
$\mathbf{I}$	Induced current	$[A]$
$\mathbf{E}$	Electrical field	$[V/m]$
$\mathbf{E}_r$	Electrical field in a relative frame	$[V/m]$
$\mathbf{D}$	Electric displacement field	$[C/m^2]$
$\mathbf{F_L}$	Lorentz force	$[N]$
ρ_l	Density for the zinc	$[kg/m^3]$
ρ_g	Density for the air	$[kg/m^3]$
$\rho_{l,g}$	Density for zinc and air	$[kg/m^3]$
ρ_c	Volume density charge	$[C/m^3]$
ρ_s	Surface density charge	$[C/m^2]$
p_g	Gas pressure	$[Pa]$
p_l	Liquid pressure	$[Pa]$
h	Liquid film thickness	$[m]$
ϵ_0	Electrical permittivity of free space	$[C^2/(N \cdot m^2)]$
ϵ	Electrical permittivity of a material	$[C^2/(N \cdot m^2)]$
Q	Test charge	$[C]$
q	Source charge	$[C]$
$\mathbf{F}_{mag}$	Magnetic force	$[N]$
μ_0	Magnetic permeability of free space	$[N/A^2]$
μ	Magnetic permeability of a material	$[N/A^2]$
V	Volume fluid	$[m^3]$
τ_g	Shear stress	$[N/m^2]$
$\mathbf{E}_{l,g}$	Linearized strain rate tensor for a liquid/gas	$[1/s]$

$\mathbf{f}_s$	Electrostatic force	$[N]$
$\mathbf{f}_r$	Electrostatic force in a relative frame	$[N]$
$\mathbf{f}_{ext}$	External force	$[N]$
Δx	Grid dimension	$[m]$
Δt	Time step	$[s]$
T	Total Computational time	$[s]$
σ	Surface tension	$[N/m]$
$[h]$	Reference quantity for the film liquid thickness	$[m]$
$[y]$	Reference quantity of the streamwise coordinate	$[m]$
$[y]$	Reference quantity of the wall-normal coordinate	$[m]$
$[v]$	Reference quantity of the strip velocity	$[m/s]$
$[B]$	Reference quantity of the magnetic field	$[T]$
A	Free surface length	$[m^2]$
V_0	Initial volume	$[m^3]$
$\mathbf{r}$	Coordinates vector	$[m]$
V_0	Characteristic velocity	$[m/s]$
L	Characteristic length	$[m]$

Dimensionless Quantities

R_m	Magnetic Reynolds
H_a	Hartmann number
C_a	Capillary number
$\hat{U}_p$	Withdrawal speed
Ω	Computation domain
ϕ	Implicit function for the interface
x	Generic point of the domain
x_c	Point over the interface
ψ_j	Shape function
$\hat{\phi}$	Test solution
χ_m	Magnetic susceptibility
$\mathbf{I}$	Identity tensor
$\hat{n}$	Unit normal vector
λ	Relaxation factor
γ	Magnetic field standard deviation
ϵ	Film parameter
k	Curvature of the interface

List of abbreviations

FEM	Finite Element Method
ODE	Ordinary Differential Equations
CFD	Computational Fluid Dynamics
MHD	MagnetoHydroDynamics
EMF	Electromotive Force
BDF	Backward Differences Formula
EM	Electromagnetic
GUI	Graphical User Interface
CSC	Finnish IT center for science
CFL	Courant Number
BEM	Boundary Element Method
LAPCK	The Linear Algebra Package
UMFPACK	The Unsymmetric Multifrontal Sparse LU Factorization Package
TFQMR	Transpose-Free Quasi-Minimal Residual
GMRES	Generalized Minimal Residual
GCR	Generalized Conjugate Residual
CG	Conjugate Gradient
CGS	Conjugate Gradient Squared
BiCGStab	Biconjugate Gradient Stabilized
GMG	Geometric Multigrid
AMG	Algebraic Multigrid
SIF	Solver Input File
MPI	Message Passing Interface

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Overview

Hot-dip galvanization is one of the most widespread and consolidated industrial processes, it consists of galvanizing a steel strip to protect it from oxidation. By the way, to reduce production cost an electromagnetic model has been designed [8]. This solution has been described thoroughly in the following chapters and it has been tested in Elmer in the light of the study [15],[16] and [19].
Thereby, this chapter presents a general description of hot-dip galvanization, by following the definition proposed in [5].

1.2 Hot-Dip Galvanization

Galvanizing is the process of applying a zinc coating to iron and steel to prevent corrosion. Hot-dip galvanizing, electrogalvanizing, and zinc spraying are three of the most used methods for adding zinc to iron and steel. The continuous sheet hot-dip technique is used to coat the majority of galvanized products. To produce a metallurgically bonded zinc or zinc-iron alloy coating, a moving steel continuous sheet is immersed in a bath of molten zinc at a temperature close to 465°C for 2 to 15 seconds. Other coatings, such as zinc-aluminium alloys, are made using the same hot-dip immersion method.
Steel rusts (oxidizes/corrodes) in nearly any climate if left exposed. Protecting steel with a thin layer of zinc is a cost-efficient technique to protect it from corrosion. Zinc coatings protect the underlying steel by providing a physical barrier as well as cathodic protection. The zinc corrosion rates in air are 10 to 100 times slower than steel, so galvanized surfaces do not need to be painted [5].

1.3 Jet Wiping

Once the steel strip exited the zinc bath, the excess zinc must be removed. To do this, jet wiping is applied; it consists of gas knives employing a low-pressure air as a medium as illustrated in figure (1.1)

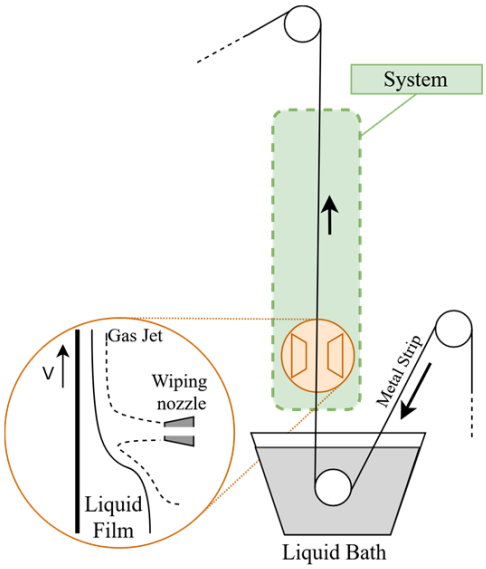


Figure 1.1: Jet wiping on the steel strip exiting the liquid bath

Although pressure is the most important variable, the management of other factors necessitates the use of a lot of equipment to keep the wiping process under control. The unnecessary liquid zinc is "cut" away by the air exiting the knife orifices, which returns to the zinc bath, leaving only the thickness required to satisfy the standard. Gas wiping is the coating control process used on all galvanizing lines, which run at rates ranging from $0.15 \quad m/s$ to over $2.5 \quad m/s$.
Nevertheless, after the jet wiping, a strip surface that is too smooth, high strip and bath temperatures, excessive strip vibrations, and fracture of the oxide layer on the liquid zinc coating surface can determine undulation phenomena on the zinc coating [24], as shown in figure (1.2).

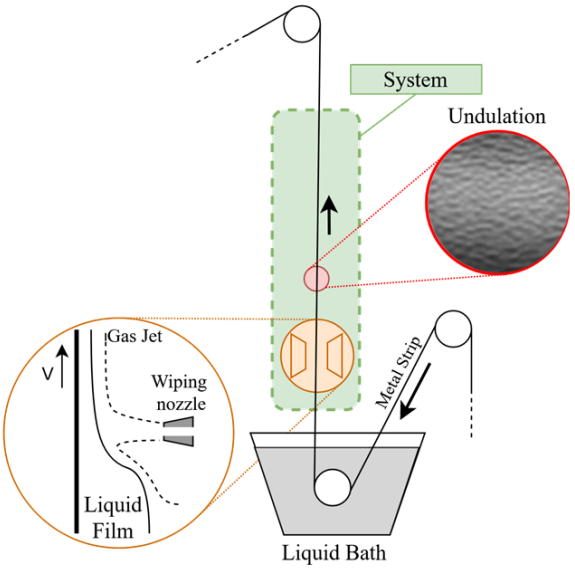


Figure 1.2: Coating degeneration on the strip after the jet wiping

To cope with the undulation phenomenon, the electromagnetic (EB) brake principle has been elaborated as described in [9]. The physics is the same of that reported in [7], nevertheless, in [7] this solution constitute a complementary to the jet wiping for hot-dip galvanizing lines, since gas knives are not sufficient to wipe the excess of zinc, due to limitation of gas pressure increase for preserving a good homogeneity of the coating. Moreover, while in [7] an electromagnet's application is described, in this work a couple of permanent magnets are involved but the physical properties and working are the same, as shown in figure (1.3)

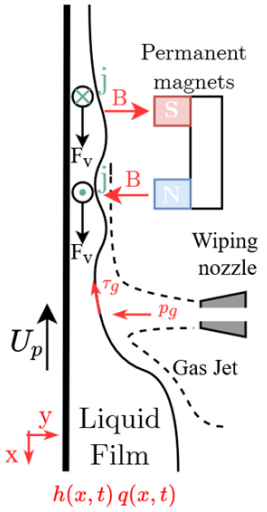


Figure 1.3: Application of permanent magnet as a brake for hot-dip galvanizing lines

The main working principle is described below. The strip coming with a velocity $U = (0, U_p)$ from

the jet wiping region encounters a magnetic field produced by a couple of permanent magnets. This field is considered steady and uniform as starting hypothesis. When the zinc coating of the strip meets the external magnetic field $\mathbf{B}^e$, the latter one combined with the velocity of the strip U_p determines an induced current density $\mathbf{J}$
Indeed, from the Maxwell-Faraday equation

$$\mathbf{J} = \sigma_m(\mathbf{U} \times \mathbf{B}^e)$$

Where σ_m is the electric conductivity. According to Ampere's rule, this induced current must produce an induced magnetic field $\mathbf{B}^i$.

$$\nabla \times \mathbf{B}^i = \mu_0 \mathbf{J}$$

Where μ_0 is the magnetic permeability of free space. Eventually, this current coupled with induced and the external magnetic fields generates the Lorentz force $\mathbf{F}_L$ acting downwards with respect to the strip motion.

$$\mathbf{F}_L = \mathbf{J} \times (\mathbf{B}^e + \mathbf{B}^i) = \mathbf{J} \times \mathbf{B} \tag{1.1}$$

Thus, this force pushes the film thickness down as shown in figure (1.3) and it copes with the undulations arising on the film.

Chapter 2

Magnetohydrodynamics

The equations mentioned in the previous chapter are constitutive for the Magnetohydrodynamics (MHD). MHD is a formal term that refers to the mutual interaction of fluid movement and magnetic fields. Electrically conducting and non-magnetic fluids are required, restricting liquid metals, hot ionized gases (plasmas), and strong electrolytes [6].

2.1 Navier-Stokes Equations

Heat transfer and viscous fluid flow in solid and liquid materials are regulated by the heat and Navier-Stokes equations, which can be derived from the fundamental concepts of mass, momentum, and energy conservation. Fluids can be Newtonian or non-Newtonian in nature [20]. The main characteristic of non-Newtonian fluids is a nonlinear dependency of shear stress τ_g on shear rate $\partial u_i/\partial x_i$ [10]

$$\tau_g = \mu_l \frac{\partial u_i}{\partial x_i}$$
$$i,j = \{x,y,z\} \quad \{u_x,u_y,u_z\} = \{u,v,w\}$$

Where μ_l is the dynamic viscosity. If a fluid is Newtonian the ratio τ_g/μ_l is constant as opposed to non-Newtonian ones. Consider the forces acting on an incompressible Newtonian fluid with density ρ_l and moving with a velocity $\mathbf{U}$. The mass and moment conservation equations are

$$\nabla \cdot \mathbf{U} = 0 \tag{2.1a}$$

$$\rho_l \left(\frac{\partial \mathbf{U}}{\partial t} + (\mathbf{U} \cdot \nabla)\mathbf{U} \right) - \nabla \cdot (2\mu_l \mathbf{E}_l) + \nabla p_l = \rho_l \mathbf{f}_{ext} \tag{2.1b}$$

Where $\mathbf{E}_l$ is the linearized strain rate tensor of the fluid.

$$E_{li,j} = \frac{1}{2}\left(\frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i} \right) i,j = \{x,y,z\} \quad \{u_x,u_y,u_z\} = \{u,v,w\}$$

The acting force on the fluid are

- Surface forces expressed by ∇p_l and $\nabla \cdot (2\mu_l \mathbf{E}_l)$
- Generic volume force expressed by $\mathbf{f}_{ext}$

This last contribute and the velocity $\mathbf{U}$ allow the interaction of fluid movement and magnetic fields. Actually, the simulations realized in Elmer concern a steady evolution, thereby the time contribute of the equation (2.1b) should be removed. Nevertheless, for a mayor clarity, the equation (2.1b) has been reported in its complete form.

2.2 MHD Equations

The mutual interaction of a magnetic field, **B**, and a velocity field, **U**, is caused in part by Faraday and Ampère's equations and in part by the Lorentz force (1.1) experienced by a current-carrying body. These two equations are gathered in Maxwell's equations which constitute the basis of the Electromagnetism theory

$$\nabla \cdot \mathbf{E} = \frac{\rho_c}{\epsilon_0} \quad Gauss's\ Law \tag{2.2a}$$

$$\nabla \cdot \mathbf{B} = 0 \tag{2.2b}$$

$$\nabla \times \mathbf{E} = -\frac{\partial \mathbf{B}}{\partial t} \quad Faraday's\ Law \tag{2.2c}$$

$$\nabla \times \mathbf{B} = \mu_0 \left(\mathbf{J} + \epsilon_0 \frac{\partial \mathbf{E}}{\partial t} \right) \quad Ampere's\ Law \tag{2.2d}$$

Where ρ_c is the external charge distribution and ϵ_0 the permittivity of free space. A more detailed derivation of this set of equations can be found in [2]. Gauss's law expresses the continuity of the electric field, whereas the equation (2.2b) confirms the solenoidal property of the magnetic field. This means that all magnetic field lines are close to one another, implying that there are no magnetic monopoles.
After the description of Maxwell and Navier-Stokes's equations, MHD equations for an incompressible fluid immediately follow. Firstly, there are the continuity equations for the charge conservation and the fluid density

$$\nabla \cdot \mathbf{J} = \frac{\partial \rho_c}{\partial t} = 0 \tag{2.3}$$

$$\nabla \cdot (\rho_l \mathbf{U}) = \frac{\partial \rho_l}{\partial t} = 0 \tag{2.4}$$

Where ρ_l is the density of the fluid and $\mathbf{U}$ is its velocity field. The external charge distribution is not considered

$$\rho_c = 0$$

Maxwell's equations are now

$$\nabla \cdot \mathbf{B} = 0 \tag{2.5a}$$

$$\nabla \times \mathbf{E} = -\frac{\partial \mathbf{B}}{\partial t} \quad Faraday's\ Law \tag{2.5b}$$

$$\nabla \times \mathbf{B} = \mu \mathbf{J} \quad Ampere's\ Law \tag{2.5c}$$

One can note that Gauss's law is not regarded due to the absence of charge distribution in the system; indeed, $\mathbf{E}$ originates from changes in the magnetic field. Considering again the equation (1.1), the

Lorentz force $\mathbf{F}_L = \mathbf{J} \times \mathbf{B}$ is the volume force term appearing in the momentum conservation equation of Navier-Stokes equations (2.1b).

$$\rho_l \left(\frac{\partial \mathbf{U}}{\partial t} + (\mathbf{U} \cdot \nabla)\mathbf{U} \right) - \nabla \cdot (2\mu_l \mathbf{E}_l) + \nabla p_l = \rho_l (\mathbf{J} \times \mathbf{B}) \tag{2.6}$$

It is worth noting that the interaction between magnetic induction equation and Navier-Stokes equations is determined by the fluid velocity $\mathbf{U}$ and Lorentz force.

2.2.1 Magnetic Induction Equation

For the derivation of the magnetic induction equation is worth following [2] and [6]. Assume that there is an electric field and a magnetic field in the laboratory frame. The force per unit charge on a charge q at rest in that frame defines the electric field, $\mathbf{E}$. When the charge moves, the force owing to the electric field remains

$$\mathbf{f}_s = q\mathbf{E}$$

Nevertheless, an extra force $q\mathbf{u} \times \mathbf{B}$ emerges, which is utilized to define $\mathbf{B}$. If a frame of reference in which the charge is at rest (but moving with velocity $\mathbf{U}$ relative to the laboratory frame) is adopted, the force on the charge can only be attributed to an electric field $\mathbf{E}_r$. The subscript r stands for "relative to a moving frame" as shown in figure (2.1)

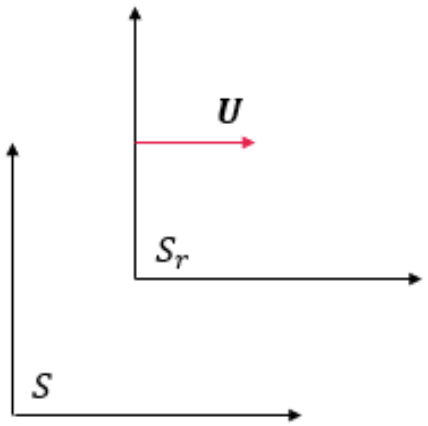


Figure 2.1: Observers in relative motion. Relative frame S_r moves with velocity $\mathbf{U}$ relative to the laboratory frame S

According to Newton's second law the forces in the two frames are

$$\mathbf{f} = q(\mathbf{E} + \mathbf{U} \times \mathbf{B}) \quad Laboratory \quad Frame \tag{2.7a}$$

$$\mathbf{f}_r = q\mathbf{E}_r \quad Relative \quad Frame \tag{2.7b}$$

The force described in equation (2.7a) is the complete expression of a force acting on a charge and it is called again Lorenz force; indeed, it is the same force described in (1.1) but with the contemporary

presence of an electric field [14]. The Newtonian relativity, which is all that is required for MHD [6], allows matching the two forces

$$\mathbf{f} = \mathbf{f}_r \tag{2.8}$$

From (2.8) it is possible to write the electric fields in the two frames

$$\mathbf{E}_r = \mathbf{E} + \mathbf{U} \times \mathbf{B} \tag{2.9}$$

The magnetic fields $\mathbf{B}$ and $\mathbf{B}_r$ are equal (relativistic corrections apart)
In a conductor Ohm's law states that the current density $\mathbf{J}$ is proportional to the force experienced by the free charges for a conductor

$$\mathbf{J} = \sigma_m \mathbf{E}$$

Where σ_m is the electrical conductivity. Now, assume a conducting fluid and consider an electric field measured in a frame moving with the velocity of the fluid $\mathbf{U}$. Ohm's law applies for a conducting fluid too and can be recast as follows

$$\mathbf{J} = \sigma_m \mathbf{E}_r = \sigma_m (\mathbf{E} + \mathbf{U} \times \mathbf{B}) \tag{2.10}$$

The key point is the substitution of Ohm's law (2.10) in Faraday's law (2.2c)

$$\begin{aligned} \frac{\partial \mathbf{B}}{\partial t} &= -\nabla \times \mathbf{E} \quad Faraday's\ law \\ &= -\nabla \times \left[\frac{\mathbf{J}}{\sigma_m} - \mathbf{U} \times \mathbf{B} \right] \\ &= \nabla \times \mathbf{U} \times \mathbf{B} - \nabla \times \nabla \times \left(\frac{\mathbf{B}}{\sigma_m \mu} \right) \end{aligned} \tag{2.11}$$

Conjuring the vector identity for a curl of a curl

$$\nabla \times \nabla \times \mathbf{F} = \nabla(\nabla \cdot \mathbf{F}) - \nabla^2 \mathbf{F}$$

and recalling the solenoidal property for the magnetic field (2.2b) yield

$$\frac{\partial \mathbf{B}}{\partial t} = \nabla \times \mathbf{U} \times \mathbf{B} - \cancel{\nabla \left(\nabla \cdot \left(\frac{\mathbf{B}}{\sigma_m \mu} \right) \right)} - \nabla^2 \left(\frac{\mathbf{B}}{\sigma_m \mu} \right)$$

The final form follows

$$\frac{\partial \mathbf{B}}{\partial t} = \nabla \times \mathbf{U} \times \mathbf{B} - \frac{1}{\sigma_m \mu} \nabla^2 \mathbf{B} \tag{2.12}$$

Where μ is the magnetic permeability. If $\mathbf{U} = 0$

$$\frac{\partial \mathbf{B}}{\partial t} = \frac{1}{\sigma_m \mu} \nabla^2 \mathbf{B} \tag{2.13}$$

Equation (2.13) shares the canonical form of a diffusion equation.

2.2.2 Boundary Conditions in MHD

Given an interface separating two media 1 and 2 shown in figure (2.2), it is always possible to write the boundary conditions below with all the field quantities measured in any arbitrary reference frame [11]

$$(\mathbf{E}_2 - \mathbf{E}_1) \times \hat{n} = 0 \quad (\mathbf{H}_2 - \mathbf{H}_1) \times \hat{n} = \mathbf{J}_s \tag{2.14a}$$

$$(\mathbf{D}_2 - \mathbf{D}_1) \cdot \hat{n} = \rho_s \quad (\mathbf{B}_2 - \mathbf{B}_1) \cdot \hat{n} = 0 \tag{2.14b}$$

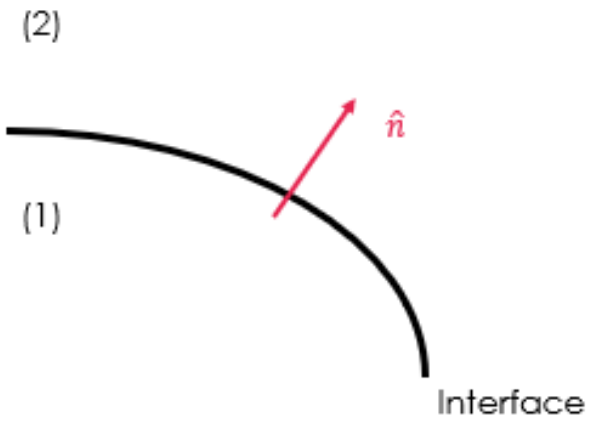


Figure 2.2: Interface between media 1 and 2. $\hat{n}$ is the unit normal vector

Where $\mathbf{J}_s$ is the surface current on the interface and ρ_s is the surface charge density on the interface. Moreover, $\mathbf{D}$ is the electric displacement field and $\mathbf{H}$ is the magnetic field

$$\mathbf{D} = \epsilon \mathbf{E} \quad \mathbf{H} = \frac{\mathbf{B}}{\mu}$$

Where ϵ is the electrical permittivity of a material and μ is the magnetic permeability of a material. Equation (2.14) states that If there is a surface current, the tangential component of $\mathbf{H}$ jumps, and the normal component of $\mathbf{D}$ jumps if there is a surface charge; the tangential component of $\mathbf{E}$ and the normal component of $\mathbf{B}$ are continuous across the interface.

Chapter 3

Steady Model With Magnetic Field

The electromagnetic model, presented in [19] and reported in the second part of this chapter, represents the theoretic base of this manuscript. It originates from the integral model of the jet wiping [15] which has been described below.

3.1 Integral Modeling of the Jet Wiping Process

Figure (3.1) shows the configuration considered in this 2D problem. A gas jet impinges on a liquid film falling along a vertical plate which moves upwards.

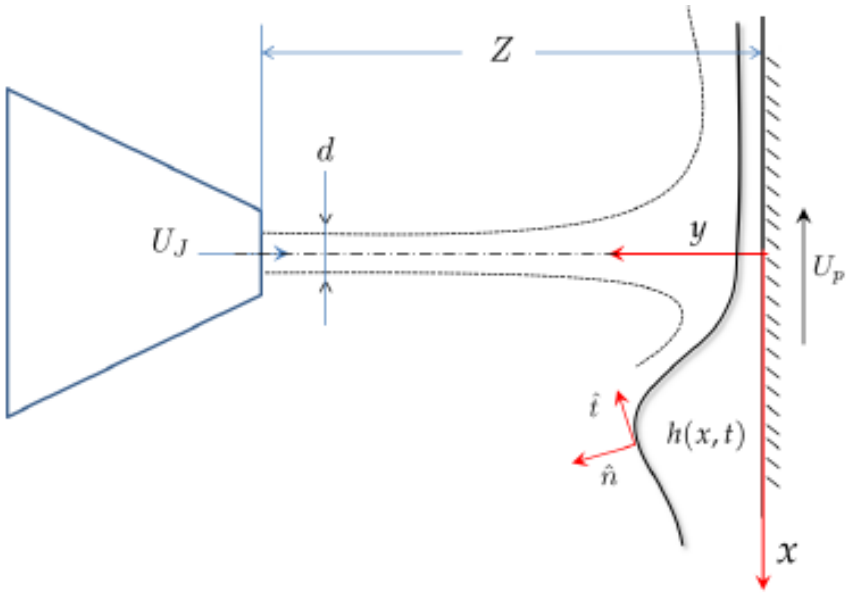


Figure 3.1: Sketch of the jet wiping configuration: a liquid film bounded by an interface $h = f(x,t)$ is dragged along a plate moving at a speed U_p, and it is impinged by a 2D gas jet released at a speed U_j , from a slot of opening d located at a distance Z. [15]

The Navier-Stokes equations regulate liquid film flow under isothermal circumstances. The continuity and momentum equations along x and y for a 2D incompressible flow with constant

characteristics (density ρ_l, dynamic viscosity μ_l, and surface tension σ) are:

$$\partial_x u + \partial_y v = 0 \tag{3.1a}$$

$$\rho_l(\partial_t u + u\partial_x u + v\partial_y u) = -\partial_x p_l + \mu_l(\partial_{xx} u + \partial_{yy} u) + \rho_l g \tag{3.1b}$$

$$\rho_l(\partial_t v + u\partial_x v + v\partial_y v) = -\partial_y p_l + \mu_l(\partial_{xx} v + \partial_{yy} v) \tag{3.1c}$$

The subscripts l and g differentiate characteristics of the liquid and gas sides, respectively.
The wall at $y = 0$ and a moving interface at h define the film flow $h(x,t)$. The no-slip and non-permeability conditions are the wall's kinematic boundary conditions:

$$\mathbf{U} = (u, v) = (-U_p, 0) \quad for \quad y = 0 \tag{3.2}$$

The kinematic and dynamic boundary conditions at the interface are both present. As observed by an Eulerian observer, the kinematic condition assures that the contact between the two fluids stays continuous, namely the material derivative of $(y - h)$ equals to zero:

$$\frac{D(y-h)}{Dt} = \frac{\partial(y-h)}{\partial t} + \mathbf{U}\nabla(y-h) = 0$$

This condition reads:

$$v = \partial_t h + u\partial_x h \quad for \quad y = h(x,t) \tag{3.3}$$

The dynamic condition ensures that the force balance at the interface is maintained, which ultimately determines the form of the interface. The normal $\hat{n}$ and tangential $\hat{t}$ directions, which are specified by the unit vectors, are used to formulate this force balance.

$$\hat{n} = \frac{(-\partial_x h, 1)}{\sqrt{1 + (\partial_x h)^2}} \quad \hat{t} = \frac{(1, \partial_x h)}{\sqrt{1 + (\partial_x h)^2}} \tag{3.4}$$

At the interface $y = h(x,t)$, the force balance along these directions is

$$Along \quad \hat{n}: \quad \hat{n}\mathbf{T}_l\hat{n} - \hat{n}\mathbf{T}_g\hat{n} = \sigma\nabla\hat{n} \tag{3.5}$$

$$Along \quad \hat{n}: \quad \hat{t}\mathbf{T}_l\hat{t} - \hat{n}\mathbf{T}_g\hat{t} = 0 \tag{3.6}$$

where $\nabla\hat{n}$ is the curvature of the interface, which for a 2D geometry reads:

$$\nabla\hat{n} = \frac{\partial_{xx} h}{[1 + (\partial_{xx} h)^2]^{3/2}} \tag{3.7}$$

and

$$\mathbf{T}_{l,g} = -p_{l,g}\mathbf{I} + 2\mu_l\mathbf{E}_{l,g}$$

is the stress tensor in the two fluids. Here, **I** is the identity tensor, **E** is the strain of rate tensor containing the symmetric (deviatoric) part of the velocity gradient. For the liquid film, in the 2D configuration analyzed, it reads:

$$\mathbf{E}_l = \frac{1}{2}\left[\nabla\mathbf{U} + \nabla\mathbf{U}^T\right] = \frac{1}{2}\begin{bmatrix} 2\partial_x u & \partial_x v + \partial_y u \\ \partial_y u + \partial_x v & 2\partial_y v \end{bmatrix} \tag{3.8}$$

Observing that $\hat{n}(-p\mathbf{I})\hat{n} = -p$ and $\hat{n}(-p\mathbf{I})\hat{t} = 0$ the projection of (3.5) and (3.6) onto the unitary vectors in equation (3.4) yields two scalar equations:

$$p_g - p_l + \hat{n}(2\mu_l\mathbf{E}_l)\hat{n} - \hat{n}(2\mu_l\mathbf{E}_g)\hat{n} = \sigma_l\nabla\hat{n} \tag{3.9a}$$

$$\hat{n}(2\mu_l\mathbf{E}_l)\hat{t} - \hat{n}(2\mu_l\mathbf{E}_g)\hat{t} = 0 \tag{3.9b}$$

Assume that gas pressure p_g and the shear stress τ_g (the so-called wiping actuators) are known.The tensor $\mathbf{E}_g$ is therefore replaced by

$$p_g(x,t) = p_g - \hat{n}(2\mu_l\mathbf{E}_g) \tag{3.10a}$$

$$\tau_g(x,t) = \hat{n}(2\mu_l\mathbf{E}_g) \tag{3.10b}$$

For the liquid flow, introducing (3.4) and equations (3.10) in equation (3.9) yields

$$p_l - p_g(x) + \frac{2\mu_l}{1+(\partial_x h)^2}[(\partial_y u + \partial_x v)\partial_x h - \partial_y v - \partial_y v - \partial_x u(\partial_x h)^2] + \frac{\sigma_l\partial_{xx}h}{[1+(\partial_x h)^2]^{3/2}} = 0 \tag{3.11}$$

in the normal direction $\hat{n}$, and

$$(1-(\partial_x h)^2)(\partial_x v + \partial_y u) + 2\partial_x h(\partial_y v - \partial_x u) = \frac{1}{\mu_l}\tau_g(x) \tag{3.12}$$

in the tangential direction $\hat{t}$. The full set of equations for the jet wiping process is given by equation (3.1) with the boundary conditions in equations (3.2), (3.3), (3.11) and (3.12).

3.2 The Simplified Steady State Solutions

For the derivation of the classical steady solution for the jet wiping problem, consider only the far field conditions, namely the regions where the jet influence is negligible and the liquid film is flat.

3.2.1 Far Field Conditions

Examine a piece of the liquid film that's distant from the impinging point. At $x \to \pm\infty$, strictly speaking. To get the steady solution, the interface is assumed to be steady ($\partial_t = 0$) and flat ($\partial_x \sim 0$), with no stream-wise pressure gradient ($\partial_x p_l = 0$). The continuity equation and kinematic boundary conditions imply that the flow is mono-dimensional ($v \sim 0$) under these assumptions, and the momentum equations in (3.1) simplify to:

$$0 = \nu_l\partial_{yy}u + g \tag{3.13}$$

The cross stream momentum equation (3.1c) an the dynamic boundary conditions are now negligible. equation (3.13) can be integrated twice to obtain the cross-stream velocity profile:

$$u(y) = \int\left(\int \frac{g}{\nu_l}\,dy\right)dy = -\frac{1}{2}\frac{g}{\nu_l}y^2 + c_1 y + c_2 \tag{3.14}$$

From the boundary conditions: $u(0) = -U_p$ and $\partial_y u(h) = 0$, the two integration constants are $c_1 = gh/\nu_l$ and $c_2 = -Up$. The velocity profile becomes:

$$u(y) = -\frac{1}{2}\frac{g}{\nu_l}y^2 + \frac{g}{\nu_l}hy - U_p \tag{3.15}$$

Therefore, the flow rate per unit width can written

$$q = \int_0^h u(y) = \frac{1}{3}\frac{g}{\nu_l}h^3 - hU_p \tag{3.16}$$

Setting $U_s = 0$ restores the classic Nusselt film solution, whereas setting $U_s \neq 0$ adds a plug flow contribution. The film thickness h determines the relative significance of the descending section ($q > 0$) and the withdrawn region ($q < 0$). Each dimensional variable is written as the product of its dimensionless representation (marked with a hat $\hat{\bullet}$) plus a reference quantity (shown within square brackets $[\bullet]$) to scale both equations. It is as follows for equation (3.16):

$$\hat{q}[q] = \frac{1}{3}\frac{g}{\nu_l}[h]^3\hat{h}^3 - [h]\hat{h}U_p \tag{3.17}$$

Equation (3.17) is divided by $[q]$ yielding:

$$\hat{q} = \frac{1}{3}\left(\frac{g[h]^3}{\nu_l[q]}\right)\hat{h}^3 - \left(\frac{U_s[h]}{[q]}\right) \tag{3.18}$$

The reference quantities can now be chosen from two options. The most obvious option is to set $[q] = [u][h] = U_s[h]$, which means using the plug flow rate as a reference flow rate. The second dimensionless quantity inside the curly brackets in (3.18) becomes unitary, and the same may be done for the first by adjusting the thickness scale $[h]$. As a result, this method is based on the following reference quantities:

$$[h] = \sqrt{\frac{\nu_l U_p}{g}} \quad [u] = U_p \quad [q] = [h][u] = \sqrt{\frac{\nu_l U_p^3}{g}} \tag{3.19}$$

Eventually, the equations (3.15) and (3.16) become:

$$\hat{u}(\hat{y}) = -\frac{1}{2}\hat{y}^2 + \hat{h}\hat{y} - 1 \quad \hat{q} = \frac{1}{3}\hat{h}^3 - \hat{h} \tag{3.20}$$

It is worth mentioning a second scale option. This is done by using the dropping portion of the flow rate (first term in equation (3.16) as a guide:

$$[q] = [u][h] = 3u_N[h] = \frac{g[h]^3}{\nu_l} \tag{3.21}$$

Where $u_N = g[h]^2/(3\nu_l)$ is the Nusselt average velocity and $[h] = h_N$ is the Nusselt film thickness. Now, the equation (3.20) can be rewritten:

$$\hat{u}(\hat{y}) = -\frac{1}{2}\hat{y}^2 + \hat{h}\hat{y} - \hat{U}_p \quad \hat{q} = \frac{1}{3}\hat{h}^3 - \hat{U}_p\hat{h} \tag{3.22}$$

where $\hat{U}_p = U_p/u_N$ is the dimensionless withdrawal speed.

Figure (3.2) shows the admissible range of interest for equation (3.22). The flow rate equation is shown on the left in figure (3.2a). The amount of liquid film descending is more than the amount absorbed by the wall when $\hat{h} > \hat{h}_c = \sqrt{3}$, and the situation becomes non physical. Figure (3.2b) shows the velocity profiles at $h_1 = 0.5$, $h_2 = 1.5$, and $h_3 = 2$ (marked with 1, 2, and 3 correspondingly) and compares them to those at $\hat{h} = 1$ and $\hat{h}_c = \sqrt{3}$.
In the range $0 < \hat{h} < \hat{h}_c$, it is clear that the decreasing flow rate ($\hat{q} < 0$) achieves a maximum of $\hat{q}_M = -2/3$ for $\hat{h} = 1$, and that for every other flow rate, two potential film thicknesses exist: a 'thin' one for $\hat{h} < 1$, and a 'thick' one for $\hat{h} > 1$. These solutions correspond to the final coat thickness and the run-back flow thickness in the wiping process, which can only occur for $\hat{h} > \sqrt{2}$. If the liquid interface is assumed to stay flat, every point on the curve $0 < \hat{h} < \hat{h}_c$ is a feasible configuration in the event of no jet wiping, i.e. in the well-known 2D drag out problem.

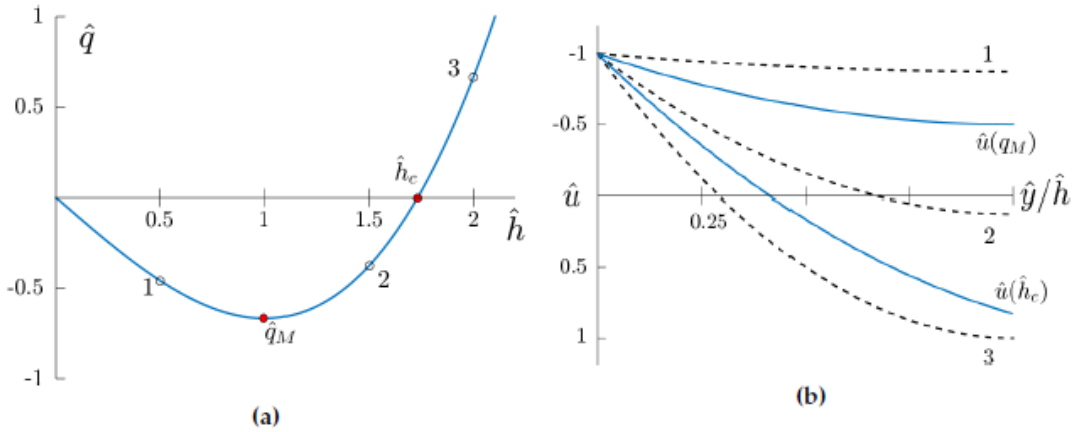


Figure 3.2: Figure a): Liquid flow rate per unit width versus film thickness in dimensionless form (3.22). Figure b): dimensionless velocity profiles (3.20) per unit thickness for three example points, labeled as 1, 2, 3 in Figure a). [15]

Contrary to the Landau theory, which predicts a film thickness of $\hat{h} = \sqrt{3}C_a$, with $C_a = U_p\mu_l/\sigma$ as the capillary number, none of the flux-based methods account for surface tension or the dynamics of the liquid meniscus produced at the bath.

3.3 Electromagnetic Model

For the derivation of the electromagnetic scheme that will be used in Elmer, the starting point is the the simplified Navier-Stokes set of equations for the steady 1D Jet wiping model

$$\partial_x u = 0 \tag{3.23a}$$

$$0 = -\frac{1}{\rho}\partial_x p_l + \nu_l(\partial_{xx} u) + g - \sigma_m B^2 u \tag{3.23b}$$

$$0 = -\partial_y p_l \tag{3.23c}$$

Where σ_m is the liquid zinc's electrical conductivity or specific conductance, and B denotes the magnetic field produced by the permanent magnets. The non-slip condition on the plate determines

the boundary conditions:

$$u(0) = -U_p \quad for \quad y = 0 \tag{3.24}$$

and the dynamic conditions at the interface:

$$p_l - p_g(x) + \sigma \partial_{xx} h = 0 \quad \partial_y u = \frac{1}{\mu_l} \tau_g(x) \tag{3.25}$$

Introducing (3.25) into (3.23b) and multiplying all the terms by ρ yield:

$$0 = -\partial_x p_g(x) + \sigma \partial_{xxx} h + \mu_l \partial_{yy} u + \rho g - \sigma_m B^2 u \tag{3.26}$$

Now, introduce non-dimensional quantities in (3.26)

$$0 = -\frac{[p]}{[x]} \partial_{\hat{x}} \hat{p}_g + \frac{\sigma [h]}{[x]^3} \partial_{\hat{x}\hat{x}\hat{x}} \hat{h} + \mu_l \frac{U_p}{[h]^2} \partial_{\hat{y}\hat{y}} \hat{u} + \rho g - \sigma_m [B]^2 U_p \hat{B}^2 \hat{u} \tag{3.27}$$

Define $[B]$ as the maximum values of the magnetic field:

$$[B] = sup(B(x,t))$$

Introduce the ratio between the film thickness and the characteristic length in the streamwise direction

$$\epsilon = \frac{[h]}{[x]}$$

and divide all the terms by (ρg):

$$0 = -\frac{[p]}{[x]\rho g} \partial_{\hat{x}} \hat{p}_g - \frac{\sigma \cancel{[h]} \epsilon^3}{\rho g \cancel{[h]^2}} \partial_{\hat{x}\hat{x}\hat{x}} \hat{h} + \frac{\sigma U_p}{g[h]^2} \partial_{\hat{y}\hat{y}} \hat{u} + 1 - \frac{\sigma_m [B]^2 U_p}{\rho g} \hat{B}^2 \hat{u} \tag{3.28}$$

By considering

$$[p] = \rho g [x] \quad [\tau_g] = \frac{\mu_l U_p}{[h]} \quad [h] = \sqrt{\frac{\nu_l U_p}{g}}$$

the dimensionless form of (3.28) can be written as follows

$$0 = -\partial_{\hat{x}} \hat{p}_g + \frac{\epsilon^3}{C_a} \partial_{\hat{x}\hat{x}\hat{x}} \hat{h} + \partial_{\hat{y}\hat{y}} \hat{u} + 1 - H_a^2 \hat{B}^2 \hat{u} \tag{3.29}$$

From the previous equation the capillary number is defined as $C_a = \mu_l U_p / \sigma$ and weights the importance of surface tension to viscosity; whereas, the Hartmann number $H_a = (\sigma_m [B]^2 U_p / \rho g)^{1/2}$ compares the importance of electromagnetic force to the viscous force. The boundary conditions (3.24) and (3.25) keep the same form:

$$\hat{u}(0) = -1 \quad for \quad y = 0 \quad \partial_{\hat{y}} \hat{u}(\hat{y}) = \hat{\tau}_g \quad for \quad \hat{y} = \hat{h} \tag{3.30}$$

Regrouping the terms of (3.29):

$$0 = A_1 + \partial_{\hat{y}\hat{y}} \hat{u} + 1 - H_a^2 \hat{B}^2 \hat{u} \tag{3.31}$$

where

$$A_1 = -\partial_{\hat{x}} \hat{p}_g + \frac{\epsilon^3}{C_a} \partial_{\hat{x}\hat{x}\hat{x}} \hat{h} \tag{3.32}$$

The homogeneous solution plus the particular one of (3.32) is given by:

$$\hat{u}(\hat{y}) = k_1 e^{(H_a \hat{B} \hat{y})} + k_2 e^{(-H_a \hat{B} \hat{y})} + \frac{A_1 + 1}{H_a^2 \hat{B}^2} \tag{3.33}$$

To determine the constants k_1 and k_2, it needs to add the boundary conditions from (3.30). After some mathematical steps, the final equation is written below

$$\hat{u}(\hat{y}) = \left\{ \frac{\hat{\tau}_g / \left(H_a \hat{B} \right) + \sinh\left(\hat{h} H_a \hat{B} \right) \left(1 + \frac{A_1 + 1}{H_a^2 \hat{B}^2} \right)}{\cosh\left(\hat{h} H_a \hat{B} \right)} \right\} \sinh\left(\hat{y} H_a \hat{B} \right) - \cosh\left(\hat{y} H_a \hat{B} \right) \left(1 + \frac{A_1 + 1}{H_a^2 \hat{B}^2} \right) + \frac{A_1 + 1}{H_a^2 \hat{B}^2} \tag{3.34}$$

The non-dimensional flow rate per unit width is obtained integrating the velocity profile in the wall-normal direction: $\hat{q} = \int_0^{\hat{h}} \hat{u}(\hat{y}')\, d\hat{y}'$

$$\hat{q} = \frac{\hat{\tau}_g}{H_a^2 \hat{B}^2} + \frac{A_1 + 1}{H_a^2 \hat{B}^2} \hat{h} - \left\{ \frac{\hat{\tau}_g / (H_a \hat{B}) + \sinh(\hat{h} H_a \hat{B}) \left(1 + \frac{A_1 + 1}{H_a^2 \hat{B}^2} \right)}{\cosh(\hat{h} H_a \hat{B})} \right\} \frac{1}{H_a \hat{B}} \tag{3.35}$$

Chapter 4

Elmer Environment

The testing campaign has been undertaken in Elmer which is a freeware software based on the finite elements method. Before of describing the test cases realized on this software, it is worth presenting Elmer's functionalities, solvers, pre-and post-processing tools. For the drafting of this chapter and further reading, the Elmer models collection [20] and solvers manual [21] can be consulted.

4.1 Overview

Elmer is a multiphysics finite element software package. It is, in essence, an open source software program for solving partial differential equations. Elmer is a powerful method since it can work with a large range of different equations that can be combined generically, and it allows users to change existing solution procedures and thus create new solvers for equations of their own choosing [23] Elmer main models present in the software and used in the simulations are:

- Electromagnetism: electrostatics, magnetostatics, induction;
- Fluid flow: the Navier-Stokes, Stokes and Reynolds equations;
- Level set method: Eulerian free boundary problems;

Elmer, like most CFD software packages, is made up of three key components: a pre-processor, a solver, and a post-processor. There are individual executables that can be used together or separately:

- **ElmerGUI**: graphical user interface (GUI) for Elmer;
- **ElmerGrid**: provides functionalities for the generation of simple meshes and conversion of accepted file formats to the native format;
- **ElmerSolver**: the main part of Elmer, the solver;
- **ElmerPost**: simple GUI post-processor.

4.2 Models/Solvers

The following solvers, whose full description can be found in [20], have been used for the implementation of this project:

- **FlowSolve** solves the Navier-Stokes equations;
- **FreeSurfaceSolver** allows the specification of a boundary as a free surface, which can then be solved in combination with the Navier-Stokes equations (FlowSolve).
- **LevelSet** fixes the interface as a zero level-set function of a higher dimensional variable. It is an Eulerian descriptive technique;
- **MagneticSolve** solves the magnetic induction equation. The latter one describes interaction between an applied or an induced electric field and a fluid (gas or liquid)

In order to solve linear and non-linear systems, Elmer includes several solution methods. These are explained briefly according to [21]. The list below sums up the main ones:

- All basic element shapes in 1D, 2D and 3D with the Lagrange shape functions of degree $k \leq 2$;
- Direct linear system solvers;
- Multigrid solvers for some basic equations;
- ILU preconditioning of linear systems;
- Time integration schemes for the first and second order equations;

4.2.1 Methods for Linear Systems

For the linear systems, there are two main categories: direct and iterative methods. It should be remembered that the direct solvers' performance is highly dependent on the sparse matrix's bandwidth. As a result, these routines often collapse miserably in 3D. For the direct methods, there are:

- The Linear Algebra Package (LAPACK) collection of subroutines;
- The Unsymmetric Multifrontal Sparse LU Factorization Package (UMFPACK) set of routines;

In the iterative methods, there are two categories: the preconditioned Krylov methods and the multilevel methods. The first ones can be used to solve both real and complex systems, and they are:

- Transpose-Free Quasi-Minimal Residual (TFQMR);
- Generalized Minimal Residual (GMRES);

- Generalized Conjugate Residual (GCR);
- Conjugate Gradient (CG);
- Conjugate Gradient Squared (CGS);
- Biconjugate Gradient Stabilized (BiCGStab);
- BiCGStab(l).

Concerning the multilevel methods, they let solve large linear systems. Two different multilevel-method approaches are available in ElmerSolver:

- Geometric Multigrid (GMG)
- Algebraic Multigrid (AMG)

4.2.2 Methods for Non-Linear Systems

The non linearity problems can be met, for instance, when the Navier-Stokes equations are involved. Regardless, the nonlinear equations are linearized by methods that depend on the used solver. For example, for the Navier-Stokes ones, the used techniques are:

- Picard linearization
- Newton linearization

4.2.3 Time Discretization Strategies

Elmer discretizes the first-order time derivates by two methods:

- the Crank-Nicolson method
- the Backward Differences Formulae (BDF) of several orders

In the case of the first order BDF scheme, the adaptive time-stepping strategy may also be used. Concerning the second-order time, it can be discretized by either using the Bossak method or reformulating the second-order equations as equivalent systems of first-order equations.

4.3 Interfaces

There are two available versions of Elmer: NoGUI and GUI. Hence for ElmerNoGUI the user has to call the software by the command line by calling the solver executable: ElmerSolver. The pre- and post- processing executables, respectively, ElmerGrid and ElmerPost, can also be called from the command line. By contrary, the ElmerGUI, used in this work, has a graphical user friendly interface.

4.3.1 Graphical User Interface

ElmerGUI can accept many mesh formats and it can be used to edit an imported one. The figure (4.1) represents the interface of the software.

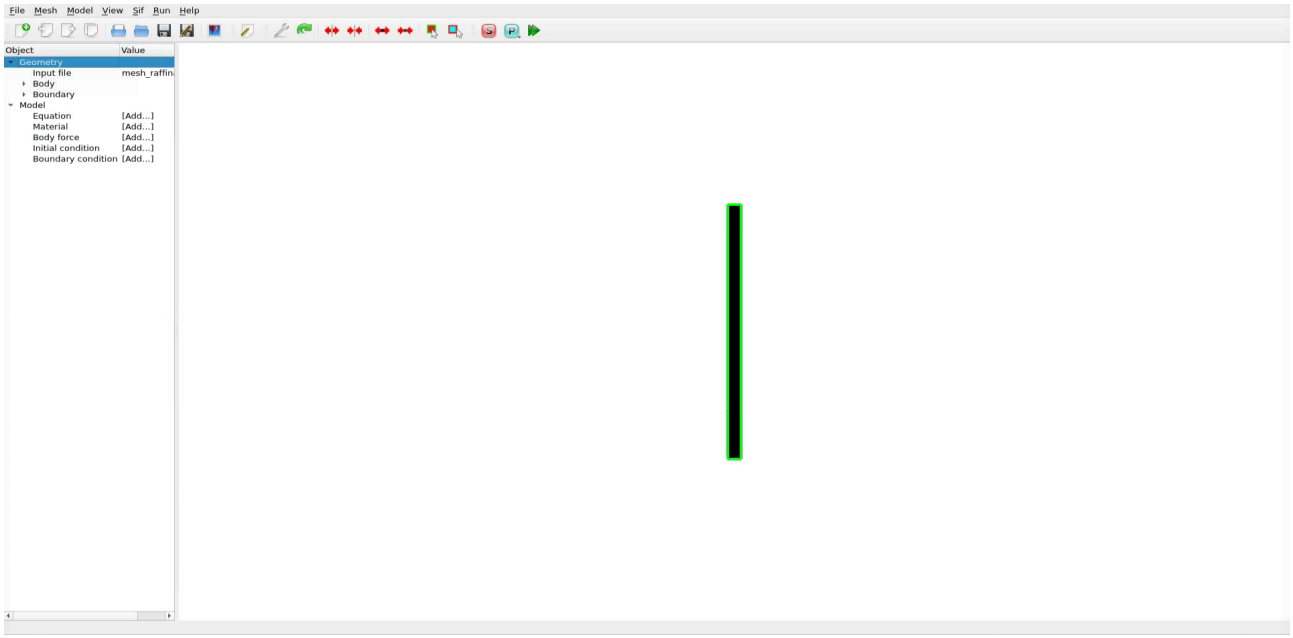


Figure 4.1: ElmerGUI interface with a loaded mesh

There a lot of menu elements in this window, this short list describes the main ones:

- **File** allows the user to load a saved project or to start a new one by loading a mesh file. The GUI's definitions and the save buttons are also located in this menu;
- **Help** is the Help menu;
- **Mesh** allocates the mesh configuration buttons;
- **Model** allows the user to stipulate the model definitions. The Setup, Equation, Material, Body force, Initial condition and Boundary condition sub-menus are located here. Defining the parameters located inside each one of these sub-menus defines the model to be simulated;
- **Run** is used to start the solver or the post-processor (ElmerPost);
- **Sif** allows the generation of the Solver Input File (case.sif) based on the Model defined properties. The user can also manually edit the Sif;
- **View** allows the user to set view preferences;

4.3.2 Solver Input File and ElmerGrid

The Solver Input File can be edited manually in order to add functionalities absent in the ElmerGUI, such as solvers or methods for linear and not linear problems. A typical configuration of a Sif file is:

- Header
- Simulation
- Constants
- Body
- Material
- Body Force
- Equation
- Solver
- Boundary Condition
- Initial Condition

Once the case.sif is written, the user can press the "Start Simulation" to run it. In case of NoGUI version, the Sif file can be called by Command line as:

```
1 ElmerSolver Test.sif
```

Issuing this command runs the solver and it saves the results in the chosen directory. ElmerGrid may be used to create the mesh, or it can be used to convert one of Elmer's mesh acceptable input files to a native Elmer mesh file. The information about ElmerGrid are in the respective manual [22]. ElmerGrid is in charge of pre-processing and includes a rudimentary mesh generator and mesh manipulation tool. It can read meshes created by other applications and edit and convert them to a format that ElmerSolver understands. ElmerGrid supports many mesh formats. The user can find them in the section "Configure" in the menu Mesh. These formats can be converted using ElmerGrid to Elmer's native mesh format: .mesh.* . Since Gmsh has been used as software to mesh the geometries of the test cases, the output format has been .msh. When the simulation has concluded, the post-processing can be launched.

4.4 Post-Processing

The are two types of Elmer file produced as output for the post-processing: .ep and .vtu. The first extension is for ElmerPost, that is enough for visualizing the results, nevertheless the developers suggest using ParaView for a rapid and user friendly visualization.

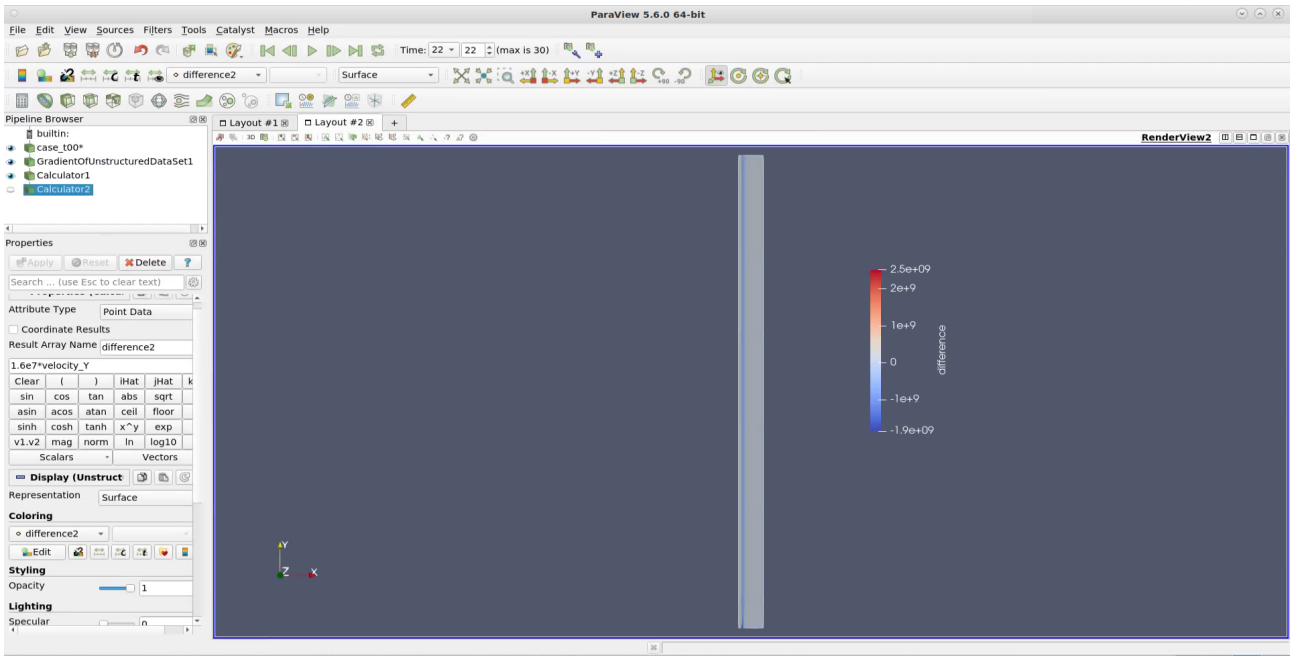


Figure 4.2: Typical ParaView interface

Chapter 5

Level Set Method

This chapter describes the mathematical model used for the simulation of the interface movement in Elmer. Being the problem multiphase, there is a contact surface between two fluids: zinc and air. Nevertheless, this interface can show issues caused by the inaccurate mesh refinement or particles generation and their subsequent separation from the interface.

5.1 Definition of the Interface

As suggested in [13], the starting point is to assume ϕ as an implicit function that describes the domain interface, that is a point or a collection of them delimiting a certain region of space from another.

$$\phi : \Omega \to \mathbb{R} \quad \Omega \subset \mathbb{R}^m \tag{5.1}$$

Where Ω is the computational domain. The time dependent position of interface is defined as:

$$\Gamma(t) = \{x \in \Omega : \phi(\mathbf{x}, t) = 0\} \tag{5.2}$$

Equation (5.2) describes the locus of the points where this function ϕ goes to zero. For instance a 2D implicit function is $\phi(\mathbf{x}) = x^2 + y^2 - 1$, where the interface $\phi(\mathbf{x}) = 0$ is the unit circle defined by $\partial\Omega = \{\mathbf{x}|\mathbf{x}| = 1\}$. The interior region is the unit open disk $\Omega^- = \{\mathbf{x}|\mathbf{x}| < 1\}$ and the exterior one is $\Omega^+ = \{\mathbf{x}|\mathbf{x}| > 1\}$. In few words, this function specifies an interface between two fluids as in figure (5.1).

5.2 Level-Set Function

After introducing the implicit function to describe an interface, a distance function can be defined. There is an additional requirement for this kind of function:

$$|\nabla\phi| = 1 \tag{5.3}$$

This indicates that the function's growth equals the minimum distance between the point and the interface when measured at a certain moment. The distance function is defined as:

$$d(\mathbf{x}) = \min(|\mathbf{x} - \mathbf{x_c}|) \tag{5.4}$$

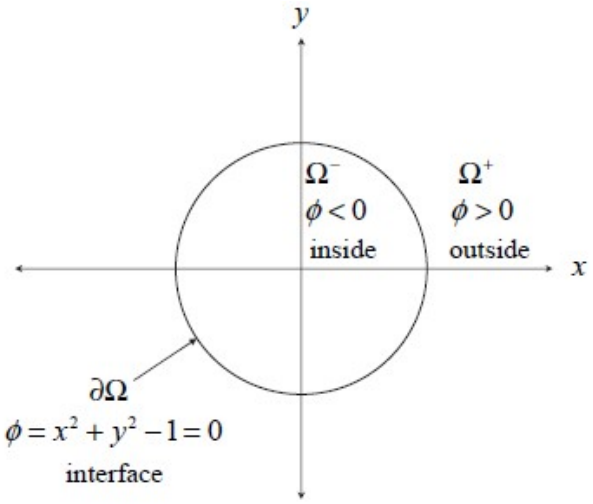


Figure 5.1: Implicit representation of the curve $x^2 + y^2 = 1$.[17]

Where $\mathbf{x_c}$ are the points belonging to the interface (where the interface function goes to zero) and $\mathbf{x}$ is a generic point of the domain Ω. Obviously, $d(\mathbf{x}) = 0$ indicates points on $\partial\Omega$.

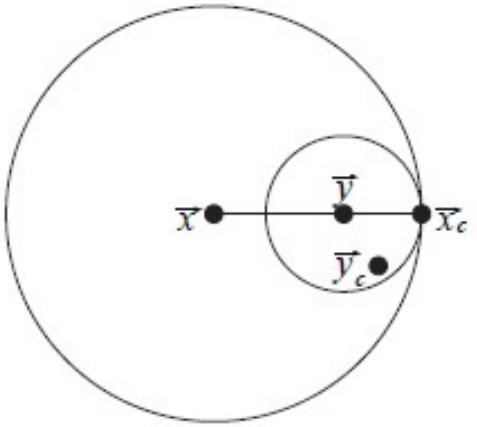


Figure 5.2: $\mathbf{x_c}$ is the closest interface point to $\mathbf{x}$ and $\mathbf{y}$.[17]

Hence, by using the distance function, the definition of a signed distance function results.

$$|\phi(\mathbf{x})| = d(\mathbf{x}) \tag{5.5}$$

For all the points of the domain $\mathbf{x}$. The signed distance function is defined as

$$\begin{cases} \phi(\mathbf{x}) = -d(\mathbf{x}) \quad over \quad \Omega^- \\ \phi(\mathbf{x}) = 0 \quad over \quad \partial\Omega \\ \phi(\mathbf{x}) = d(\mathbf{x}) \quad over \quad \Omega^+ \end{cases} \tag{5.6}$$

Elmer's level-set equation (5.6) is computed using a signed distance that takes the norm of the distance concerning a defined interface location and multiplies it for the signed distance.

$$\phi_0(\mathbf{x}) = -sgn(\mathbf{x} - \mathbf{x}_c)||\mathbf{x} - \mathbf{x}_c||_2 \tag{5.7}$$

From (5.7) the positive values are inside and negative ones are outside; all of the properties of the distance functions, including the one represented in equation (5.3), are shared by this new collection of functions. This feature, however, only applies to locations that are equidistant from one point of the interface. That property no longer holds when there are two or more points of zero isocontours with the same distance from a domain point, particularly when the derivatives of the signed distance function are estimated using a discretised approach; this becomes an issue.
Furthermore, the level-set function ϕ corresponds to class C^0 in the worst-case scenario. Although the signed distance function is always continuous and differentiable, its initial derivatives may have discontinuities in their values, making it non-differentiable.
The level-set methods just add dynamics to signed distance functions; as a result, an implicit interface transforms into a moving object that evolves over time.

5.3 Advection of The Level-Set Function

Assume that the velocity $\mathbf{U}(\mathbf{x})$ is known for every point $\mathbf{x}$ of the implicit surface $\phi(\mathbf{x})$; all the points on the surface move with this velocity. The simplest way to do this is to solve the ordinary differential equation (ODE):

$$\frac{d\mathbf{x}}{dt} = \mathbf{U}(\mathbf{x}) \tag{5.8}$$

for every point $\mathbf{x}$ on the front, that is for all $\mathbf{x}$ with $\phi(\mathbf{x}) = 0$. For sure, this is the Lagrangian approach and it means discretising the front, formed by an infinite number of points. Such discretisation can be lead by using segments in 2D or triangles in 3D (front tracking method). However, substantial boundary element distortion is likely to occur, resulting in rapid degradation of the results. This problem may be bypassed by implementing a periodic adjustment of the interface's discretisation to keep it smooth and regular; however choosing a new discretisation is not easy, making it difficult to implement.
To avoid problems with instabilities, deformation of surface elements, and complicated surgical procedures for topological repair of interfaces [17], the implicit function ϕ is used to represent the interface and evolve the same one in time through an equation. This approach is called: Eulerian perspective. Namely, it is the method of developing the entire signed distance function field through an equation, and the law employed is the so-called transport equation. Hence, the level-set equation is the Eulerian advection equation of the interface ϕ, which, integrated in time, gives the surface motion.

$$\phi_t + \mathbf{U} \cdot \nabla\phi = 0 \tag{5.9}$$

This approach for the interface evolution is called Eulerian since the interface is captured by the implicit function ϕ instead of being tracked by single interface elements as the Lagrangian formulation does. The velocity field $\mathbf{U}$ is defined on all over domain containing grid nodes because it is complicated to have a velocity just on the interface.
However, for the numerical interest, it is sufficient to define the velocity in a band containing the interface ($\phi(\mathbf{x}) = 0$). The thickness of this band is ϵ. It should be significantly bigger than Δx of the grid to get a good approximation of the velocity near the interface and minimise the variation in the velocity field.

As already mentioned, the velocity $\mathbf{U}$ of the equation (5.9) is that one of the domain,i.e, from the Navier-Stokes equations (2.1). In studied problem ϕ of the level set indicates an interface separating two different incompressible fluids. The sign of ϕ is used to identify which gas or liquid occupied which region, i.e., to determine the local equation of state.

However, since the advection equation (5.9) is not physical, problems with the volume and mass conservation may occur. The correction, brought by Elmer [20], has no physical basis, but it may be argued that a consistently small update of the level set function has a minor effect in overall results.

$$d\phi = \frac{V_0 - V}{A}$$

where V_0 is the initial volume, V is the volume at time step t and A is the free surface length at time step t.

5.4 Fluid Properties

The information provided by the level-Set function (5.7) is useful to define the fluid properties like density and viscosity in the domain. They depend on the location of the interface $\Gamma(t)$ and the curvature k of the latter one leaning on the interface function ϕ
For instance, the density ρ and dynamic viscosity μ functions in the domain are

$$\begin{aligned} \rho_{l,g}(\mathbf{x}) &= \rho_g + (\rho_l - \rho_g)\theta(\phi(\mathbf{x})) \\ \mu_{l,g}(\mathbf{x}) &= \mu_g + (\mu_l - \mu_g)\theta(\phi(\mathbf{x})) \end{aligned} \tag{5.10}$$

$\theta(\phi(\mathbf{x}))$ is a smooth equation such as a hyperbolic tangent

$$\theta(\phi(\mathbf{x})) = \frac{1}{2}\left[1 + \tanh\left(\pi\frac{\phi(\mathbf{x})}{E}\right)\right] \tag{5.11}$$

where E is a smooth parameter consistent with grid dimension Δx.

5.5 Surface Tension Force

It is worth mentioning some tricks about the computation of the surface tension force in Elmer as mentioned in [20]. In the Eulerian approach to the free surface problems, the surface tension force must be smeared out to a volume force within a narrow band from the interface. The transformation is achieved by using a regularised delta function:

$$\int_\Gamma \sigma k \; d\Gamma = \int_\Omega \sigma k \delta(\phi) \nabla\phi \; d\Omega \tag{5.12}$$

Where σ is the surface tension coefficient and k is the curvature of the interface given by

$$k = \nabla \cdot \frac{\nabla\phi}{|\nabla\phi|}$$

Nevertheless, in the FEM approach, the surface force cannot be estimated directly since it involves three derivatives of the level set function. Thereby, an additional equation for the curvature k must be solved

$$k - c_k \nabla^2 k = \nabla \cdot \nabla \tilde{\phi} \tag{5.13}$$

Here, c_k is an ad hoc diffusion coefficient that may be used to smooth the resulting curvature field k so that it could avoid the born of sharp corners. $\nabla \tilde{\phi}$ is the normal at the interface from which the surfaces fluxes are evaluated. Hence, Elmer first computes k and then multiplies it for the gradient of ϕ to avoid the computation of the third derivative and limiting it to a second one.
Eventually, surface tension can be used as a volume force in the flow equations once the level-set function and associated curvature have been determined. However, in the model realized in this project, the absence of the surface tension has been assumed.

5.6 Reinitialization

The reinitialization of the level-set is lead periodically [4] in order to avoid the steepening and flattening effects. One can stop the calculation at any point in time and reset the other isocontours so that ϕ is again initialised to a signed distance function. To achieve that, a straightforward routine is restricting the calculations of the interface motion and reinitialising it to a small band of points near $\phi = 0$ isocontour. Thereby, only the isocontour $\phi = 0$ needs to stay well behaved; conversely, in a standard numerical method, it is supposed that the solution will stay behaved until the final solution is computed. The reinitialization techniques attempt to improve mass or volume conservation because the level set methods, as mentioned above, tend to lose mass in underresolved flow regions.
2D reinitialization in Elmer may be readily accomplished using a geometric method [20]. To create the zero level-set, first, it needs to go through all of the elements and locate the line segments that make up the zero level-set. After that, a brute-force search is used to find the shortest distance between all of the nodes. If there are N nodes and M line segments, the search procedure is $N \times M$, which is a reasonable complexity for small situations but might be computationally expensive in larger ones.
It is reasonable to suppose that the line segments will follow the flow, generating an on-the-fly Lagrangian mesh in the process. As a result, when the velocity field is supplied, it is also feasible to advect the line segments because for every node $\mathbf{r} = \mathbf{r} + \mathbf{U}dt$. The shortest distance is calculated after the advection. The sign of the distance is inherited from the original level set function when there is no advection. When the level-set is also convected, however, the sign must be inferred from the geometric data. In the present approach, each line segment has a flag that indicates where the fluid of interest is placed on the element. The right sign is then given based on the directional information.

Chapter 6

Steady Model Without Magnetic Field

In the light of level-set explication, a first test case has been realized. This is the steady model without a magnetic field for the zinc liquid film and it validates what has been explained in the chapter 3 where the integral model [15] has been reported.

6.1 Domain

The original domain, taken from [16], is rectangular, as shown in figure (6.1); the two main lengths are

$$L_y = 12300h \quad L_x = 7.5h$$

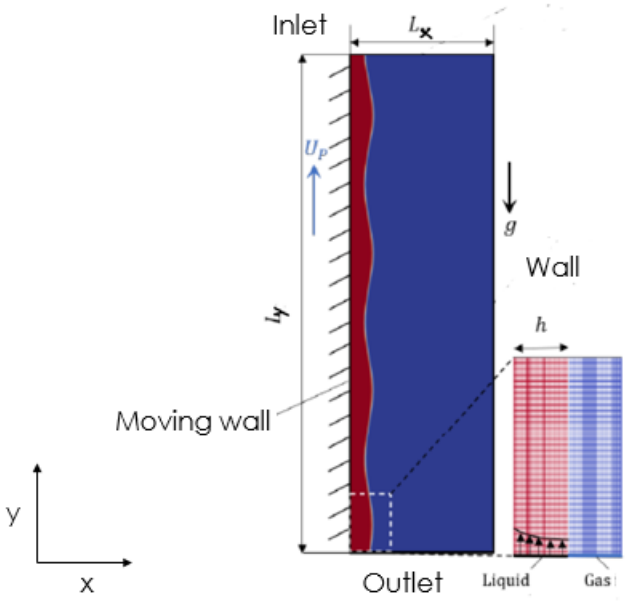


Figure 6.1: Schematic of the flow configuration used for validation purposes in Elmer: flow domain and zoom on the near-wall mesh in [16]

Nevertheless, these magnitudes used in [16] are excessively disproportionate for the meshing and the computation in Elmer for any value of h; thereby, by keeping the concept of streamwise direction

larger than the wall-normal's one: $L_x << L_y$, it has been advantageous to opt for dimensions more feasible in Elmer and independent of h.

$$L_y = 0.02\ m \quad L_x = 0.001\ m$$

6.2 Meshing

Gmsh, a freeware software, has been used for the meshing of the interest region. As the first step, the geometric characteristics in the section "Geometry" have been created, as in figure (6.2)

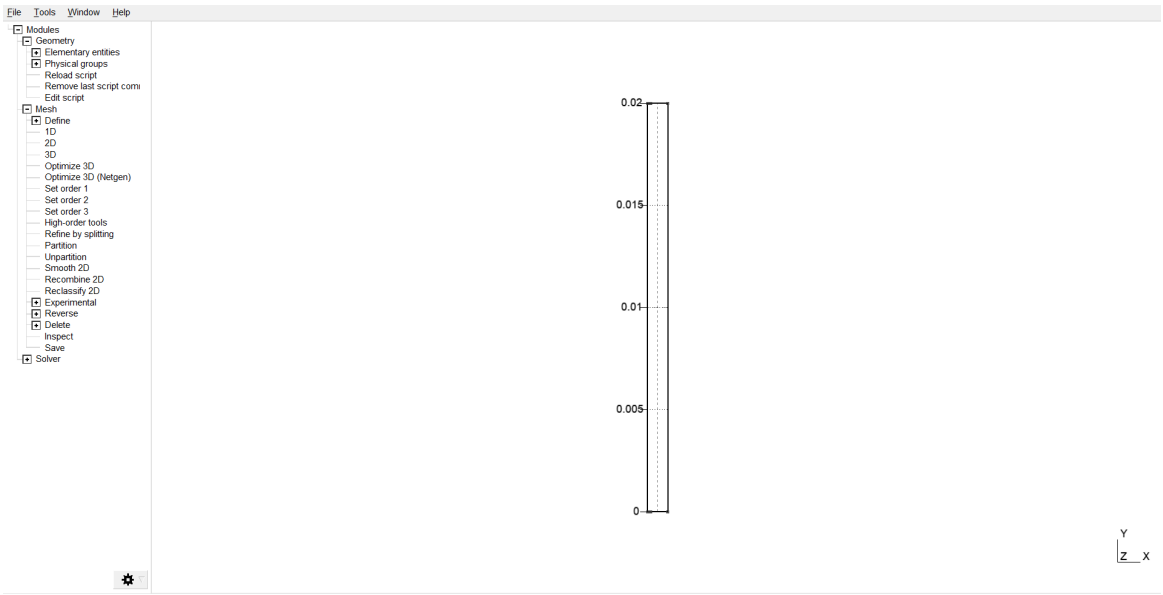


Figure 6.2: Geometry of the domain realized in Gmsh

Once the region's surface has been built, the boundaries have been specified. These are easily set through the section "Physical groups"; in this case, they are four.
The meshing is specified in "Mesh"→"Transfinite"→"Surface" where it is possible to specify the limits of the mesh and eventually, the number of elements on each side (six sides in total) can be chosen by the box named "Curve" in the same decision tree. The final mesh is of structured type; the interest area, i.e., the liquid zinc part of the domain, has been refined instead of the air zone preserved coarser as in figure (6.3).

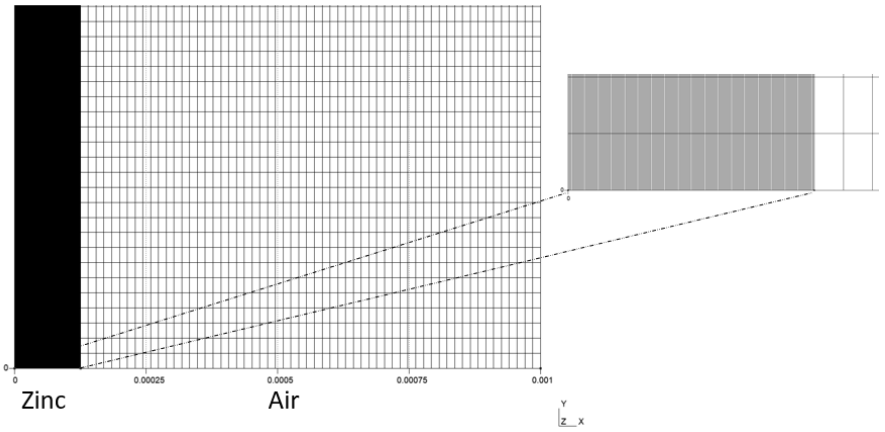


Figure 6.3: Different meshing in the domain: zinc area presents a finer mesh compared to air region

Actually, the position of the interface zinc-air is not exactly at h according to [16]. This is due to the fact that the level-set is involved for the liquid film. This method does not allow to specify precisely the interface, hence the need for a smoothing region which has been placed at a distance of $2h$. Known various $\hat{h}$ from [15], it is easy to compute h for each one thanks to the dimensionless representation presented in chapter 3

$$\begin{gathered} h = \hat{h}[h] = \hat{h}\sqrt{\frac{\nu_l U_p}{g}} \\ U_p = 0.5\ m/s \quad g = 9.81\ m/s^2 \quad \nu_l = 4.461 \cdot 10^{-7}\ m^2/s \end{gathered} \tag{6.1}$$

Table (6.1) illustrates the position of the interface in the mesh for different values of $\hat{h}$

$\hat{h}$	$h = \hat{h}[h]$	$2h$
0.5	$7.5 \cdot 10^{-5}\ m$	$1.5 \cdot 10^{-4}\ m$
1	$1.5 \cdot 10^{-4}$ m	$3 \cdot 10^{-4}\ m$
1.5	$2.25 \cdot 10^{-4}\ m$	$4.5 \cdot 10^{-4}\ m$

Table 6.1: Computation of the interface position in the mesh

Concerning the number of elements used for the meshing, it has been useful to start with a certain amount of elements and, afterwards, an independent mesh study has been realized to individuate the right quantity that did not affect any result.

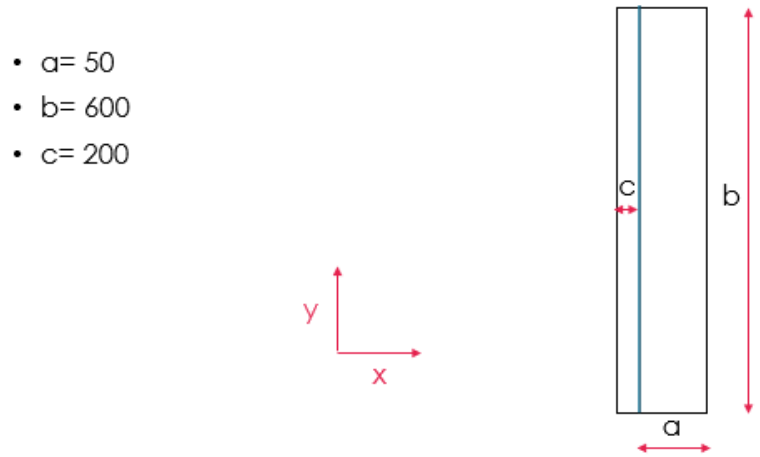


Figure 6.4: Schematic of the distribution of the elements in the mesh

6.3 Test Case Description

Once the mesh has been realized, it has been upload by the command "Open" in Elmer to start the next definition of the simulation parameters.

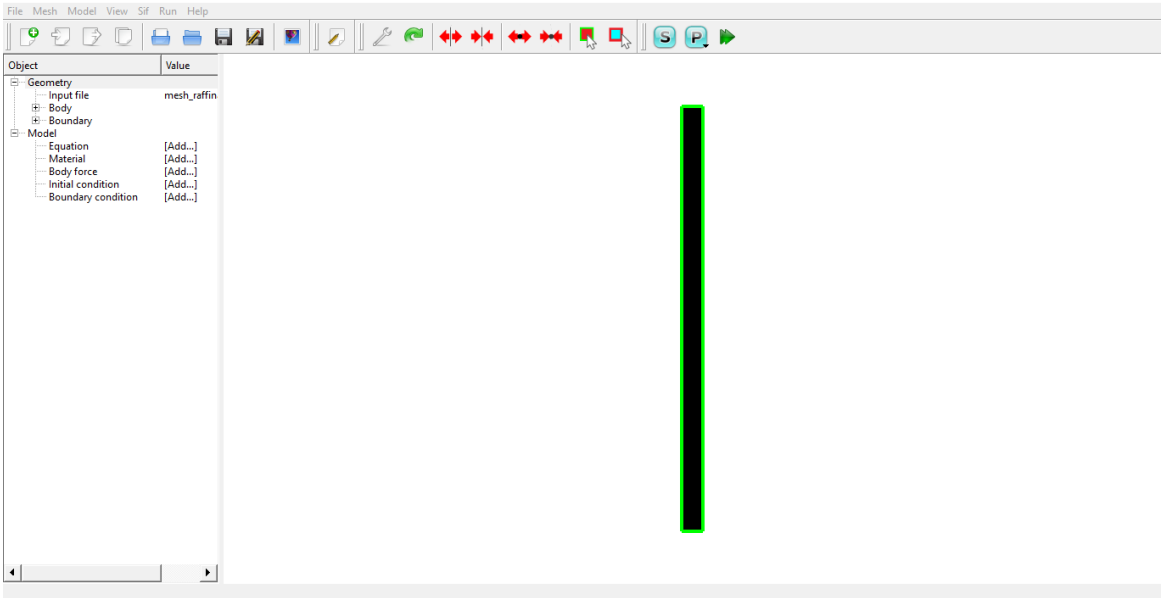


Figure 6.5: Opening of the mesh in Elmer

To facilitate the comprehension of the file.sif in Elmer, its every section has been reported. Some elements are default and have not been changed to ensure convergence; many suggestions have come from the Elmer community. The full file has been inserted in Appendix 11.

6.3.1 Simulation Set-Up

Concerning the time length of the simulation, the time step size is $0.01\ s$ and the number of time steps is 10000. Thus, the total computational time has been

$$T = 0.01 \times 10000 = 100\ s$$

If a simple transient simulation had been kept, divergence issues would have occurred; thereby, an adaptive time step has been applied to avoid numerical problems with the level set.
As [21] explains, the adaptive time-stepping is achieved by first solving the system with a trial time step, then comparing the results with two-time steps that are half the duration of the trial time step. The usage of the trial time step is acceptable if the difference between the findings is determined to be modest enough. Otherwise, a new trial time step is defined by splitting the previous trial time step into two equal-length steps and then repeating the operation. Any times step has to respect the CFL condition to avoid numerical instabilities.

$$CFL = \frac{U\Delta t}{\Delta x} < 1 \tag{6.2}$$

The CFL has been fixed to assure the stability conditions was respected. Moreover, based on the characteristic valued of the spatial mesh and the characteristic velocity of the system, it has been possible to define a time step Δt and thanks to the adaptive time-stepping technique, the latter one could change.

$$CFL = 0.5 \quad U = U_p = 0.5\ m/s \quad \Delta x = 6.4 \cdot 10^{-7}\ m$$

According to the Elmer manual [20], the CFL condition can be applied just for the level set solver since it is not provided for other solvers. Increasing the actual CFL value has not presented substantial results, while its decrease has led to divergence.
The discretization method for the time derivatives is backward differences formula (BDF) and it is second order. In general, a BDF is used to solve an initial value problem like

$$\frac{dy}{dt} = f(t, y) \quad y(t_0) = y_0$$

By defining with n the order of the BDF and p the time step size $t_n = t_0 + np$, the general BFD formula is

$$\sum_{k=0}^{s} a_k y_{n+k} = p\beta f(t_{n+s}, y_{n+s}) \tag{6.3}$$

BDF is implicit and needs the resolution of the non-linear equation for each step because f is evaluated for the unknown y_{n+s}, a and β are chosen to assure the reach of the maximum grade s. Thus, the BDF of second order used in Elmer is:

$$y_{n+2} - \frac{4}{3}y_{n+1} + \frac{1}{3}y_n = \frac{2}{3}f(t_{n+2}, y_{n+2}) \tag{6.4}$$

The example above has been taken from [1]. The non-linear equations, as declared in chapter 4, are solved in Elmer by using Picard Iteration or Newton iteration.

6.3.2 Body Forces

According to [16] the only body force is the gravitational one whose value and direction have been written in the section "Body Force" of the decision tree diagram. As one can see in figure (6.1), the direction is along y downwards.

6.3.3 Initial Conditions

In the initial condition section, the free surface for the level set has been defined, namely, the implicit function ϕ depending on the wall-normal coordinate x. Once it has been included in Elmer, the level set computes the function ϕ_0 and transports it in time using (5.9) whose velocity is from the Navier-Stokes equation. The interface is located at h computed in (6.1) and is defined as a surface variable in Elmer, called tx. In the code (11), tx is the wall-normal coordinate x, h is $\hat{h}$ and $h1$ is $[h]$. Regarding the velocity field, the only non-zero velocity is along the streamwise direction y; nevertheless, since the equation (3.15) is defined in a different reference scheme, as visible in figure (6.1), it has been adapted according to that one used in this simulation. To get it, the boundary conditions and the momentum equation (3.13) have been rewritten

$$v(0) = U_p \quad \partial_x v(h) = 0 \tag{6.5}$$

$$0 = \nu_l \partial_{xx} v - g \tag{6.6}$$

Now, by integrating twice equation (6.6), the cross-stream velocity profile can be obtained

$$v(x) = \int \left(\int \frac{g}{\nu_l} \, dx \right) dx = \frac{1}{2} \frac{g}{\nu_l} x^2 + c_1 x + c_2 \tag{6.7}$$

and by using the new boundary conditions (6.5) the coefficients c_1 and c_2 can be determined

$$c_1 = -\frac{g}{\nu_l} h \quad c_2 = U_p$$

Thus, the cross-stream velocity profile is

$$v(x) = \frac{1}{2} \frac{g}{\nu_l} x^2 - \frac{g}{\nu_l} h x + U_p \tag{6.8}$$

6.3.4 Material

The surface variable computed above is used to define the fluid properties: density and dynamic viscosity. To ensure a smooth evolution of these variables, the hyperbolic tangent (5.11) has been inserted in equation (5.10) according to the level set free surface and the whole equation has been provided to Elmer. E, appearing in (5.11), is the smooth coefficient and should be larger than Δx grid dimension.

$$E = 1 \cdot 10^{-5}$$

$$\Delta x = 6.4 \cdot 10^{-7}$$

Figure (6.6) displays the density function's trend for three different values of E. Actually, different E parameters have been tested, particularly for $E > 10^{-4}$ some divergence problems have occurred with the increasing of h, nevertheless only one has guaranteed a good smoothing and the convergence of the simulation.

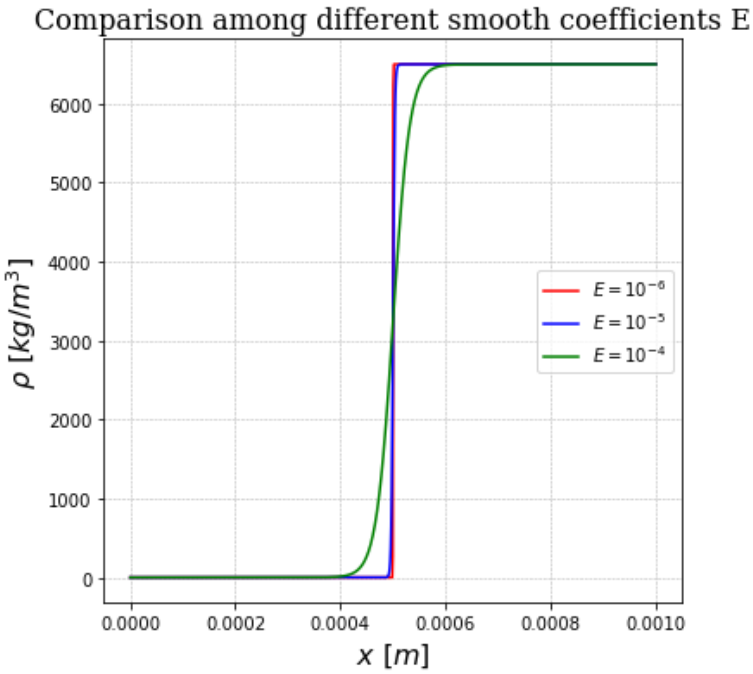


Figure 6.6: Influence of the parameter E. ρ refers to the density function $\rho_{l,g}$

Table (6.2) resumes the parameters used in Elmer

$Elmer$	$Variable$	$Value$
$rhoPlus$	ρ_l	$6500\ kg/m^3$
$rhoMin$	ρ_g	$1\ kg/m^3$
$muPlus$	μ_l	$0.0029\ Pa \cdot s$
$muMin$	μ_g	$1 \cdot 10^{-5}\ Pa \cdot s$

Table 6.2: Names of the parameters employed in Elmer

Moreover, the advection speed term has been imposed as level-set velocity; thereby, it has been matched with the velocity of the initial conditions for Navier-Stokes equations. The last parameter concerning the level-set is called in Elmer bandwidth bw and indicates the value at which the viscosity is smeared out, that is, no longer considered.

6.3.5 Solvers

Four solvers have been used

- two for level-set: free surface and reinitialization.

- one for Navier-Stokes
- one for post-processing

The full equations system, solved by Elmer, is presented, as follows

$$\begin{cases} \nabla \cdot \mathbf{U} = 0 \quad Mass \quad conservation \quad equation \\ \rho_{l,g}\left(\frac{\partial \mathbf{U}}{\partial t} + (\mathbf{U}\cdot\nabla)\mathbf{U}\right) - \nabla(2\mu_{l,g}\mathbf{E}_{l,g}) + \nabla p_{l,g} = \rho_{l,g}\mathbf{f}_{ext} \quad Momentum \quad equation \\ \frac{\partial \phi}{\partial t} + \mathbf{U}\cdot\nabla\phi = 0 \quad Level-set \quad equation \end{cases}$$

Figure (6.7) shows the execution order of the solvers in Elmer

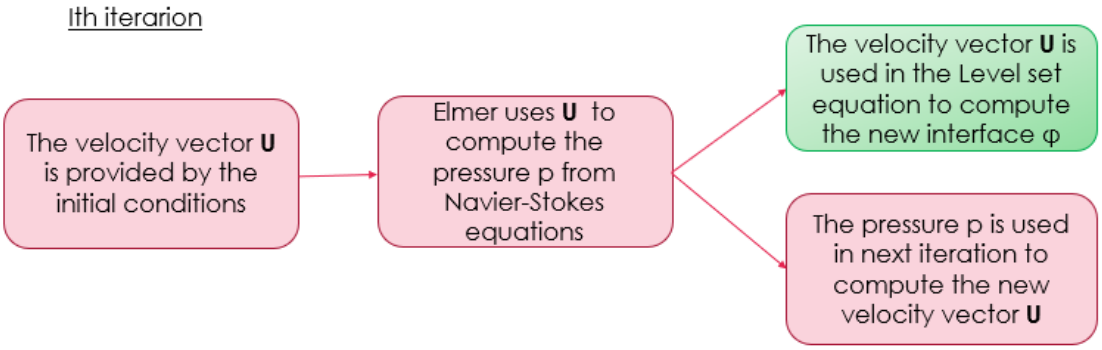


Figure 6.7: Order of equations resolution in Elmer

For level set and Navier-Stokes, umfpack has been used as a direct method for linear system resolution; indeed, both equations are linear. However, the convection term $(\mathbf{U}\cdot\nabla)\mathbf{U}$ of the momentum conservation equation (2.1b) is nonlinear and must be linearized for the computing of the solution. There are two linearization methods of the convection term in Elmer: Picard (first-order) and Newton (second-order).

$$(\mathbf{U_{n+1}}\cdot\nabla)\mathbf{U_{n+1}} \approx (\mathbf{U_n}\cdot\nabla)\mathbf{U_{n+1}} \quad Picard\ iteration$$
$$(\mathbf{U_{n+1}}\cdot\nabla)\mathbf{U_{n+1}} \approx (\mathbf{U_n}\cdot\nabla)\mathbf{U_{n+1}} + (\mathbf{U_{n+1}}\cdot\nabla)\mathbf{U_n} - (\mathbf{U_n}\cdot\nabla)\mathbf{U_n} \quad Newton\ iteration$$

Where $\mathbf{U_n}$ is the velocity vector from the previous iteration, Newton method has been chosen because Picard's one is slower in achieving the solution with computational times extremely huge. Nevertheless, although Newton is a second order method, a good initial guess for velocity and pressure fields is required.
Eventually, the reinitialization for the level-set, described in the chapter 5, can be fitted; indeed, it is possible to decide the interval how often it is done, " Reinitialize Interval" in Elmer, and how often the zero level-set $\phi(\mathbf{x}) = 0$ is extracted ("Extract Interval").

6.3.6 Boundary Conditions

As depicted in figure (6.1), the interested domain has four boundaries; a boundary condition on each side according to the numerical case in [16] has been imposed. The main issue was to assure the mass flow conservation without any leak of the zinc film; this was achieved through a specific combination of the boundary conditions for Navier-Stokes since there are no specific ones for the level set solver. Particularly, this scope has been reached by adjusting the boundary conditions only on velocity terms. Homogeneous Dirichlet and Neumann boundary conditions either only on pressure or on the whole pressure and velocity have not been produced the expected results determining leaks in the liquid film.

Wall

Homogeneous Dirichlet boundary conditions have been applied for the two velocity components

$$\mathbf{U} \cdot \hat{n} = 0$$

it means writing in Elmer

$$u = 0$$
$$v = 0$$

Moving Wall

The velocity for the wall in the streamwise direction has been specified, whereas a homogeneous Dirichlet condition has been applied for wall-normal.

$$u = 0$$
$$v = U_p$$

Inlet and Outlet

A homogeneous Dirichlet condition has defined the wall-normal velocity

$$u = 0$$

Indeed, for the streamwise velocity, no numerical quantity has been written. This means a homogeneous Neumann boundary condition implicitly or also called natural condition.

$$\frac{\partial v}{\partial x} \cdot \hat{n} = 0$$

Figure (6.8) sums up the initial and boundary conditions applied for the current test case.

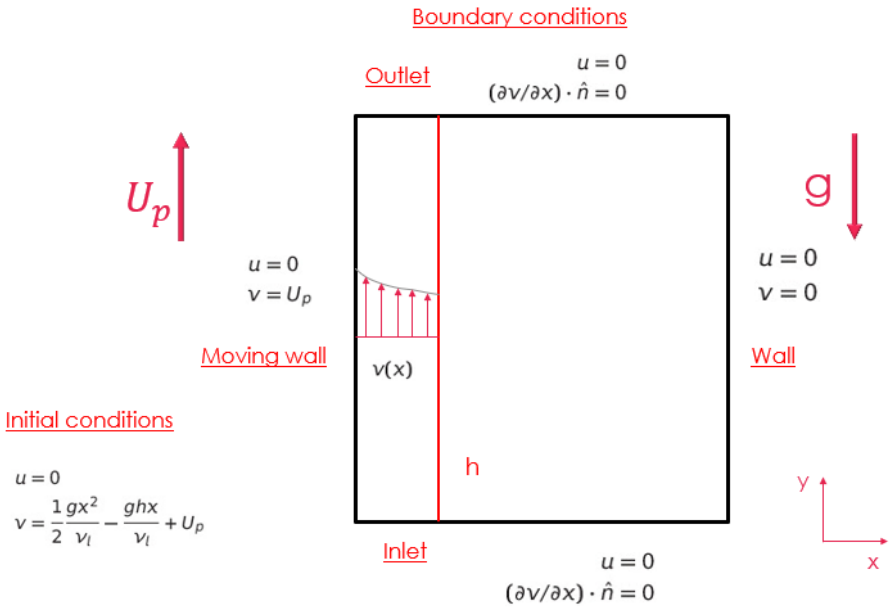


Figure 6.8: Schematic of the initial and boundary conditions applied in the model

6.4 Post-processing

The velocity of the moving wall is $U_p = 0.5\ m/s$, since a higher speed has led to unacceptable results compared to those expected in the reference paper [16] and has brought divergence issues. Once the simulation with the case.sif, defined above, has been launched, Elmer could replicate the results found in [15]. Figure (6.9) displays the comparison between Elmer's computation and the results given by the velocity profile computed in [15], which have been processed in Matlab by providing equation (6.8).

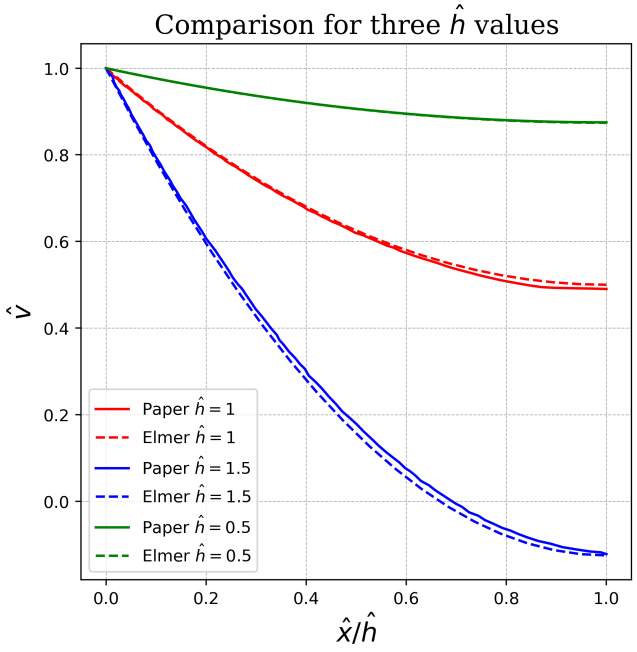
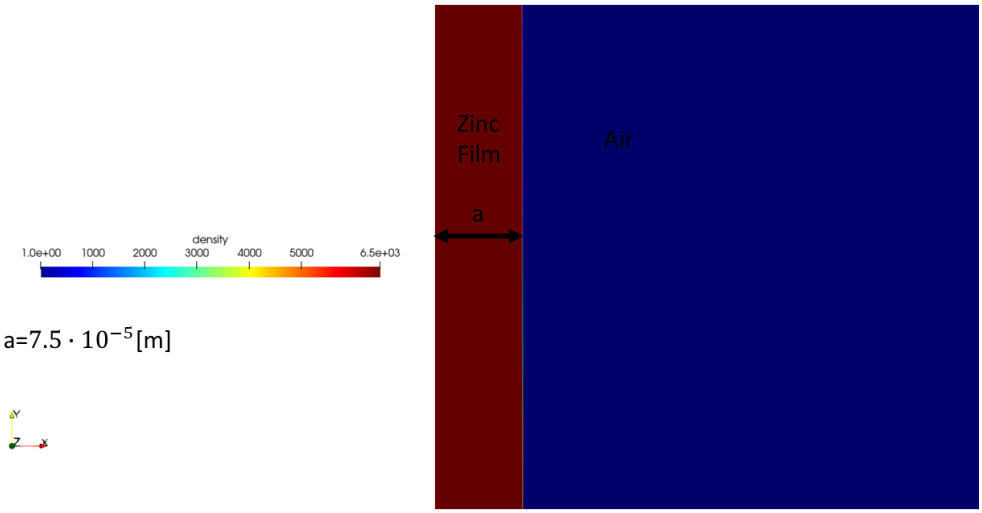
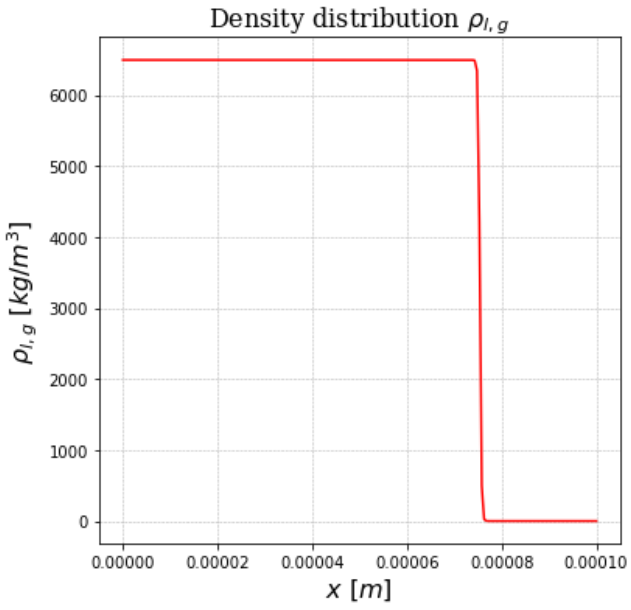


Figure 6.9: Dimensionless velocity profiles per unit thickness for three $\hat{h}$ values, labelled as 1.5, 1, 0.5 compared with ones from [15]

Regarding figure (6.9), one can note the different orienting axes due to the distinct reference system and the exact superposition between Elmer's results and those from the paper already mentioned. Matlab's code is in Appendix 11.
As already said before, a smooth change for the fluid properties is needed in Elmer, because it is impossible to have a steep variation of these characteristics between the inside and the outside of the liquid film. Thereby, a hyperbolic tangent function (5.9) for the evolution of the density and dynamic viscosity over the interface has been used as shown in figure (6.10)



(a)



(b)

Figure 6.10: Figure (a) the liquid and gas distribution in terms of density where zinc film has a density ρ_l and air has a density ρ_g; figure (b) shows the density function $\rho_{l,g}$ described over the interface

6.5 Independent Mesh Study

An independent mesh study has been carried-out to define the best mesh in terms of the lowest Euclidean norm L_e^2 and to save computational time.

$$L_e^2 = \frac{\sum_{i=1}^{N}[\phi_E(x_i) - \phi_M(x_i)]^2}{[\phi_E(x_i)]^2}$$

Where N is the number of nodes x_i that constitute the velocity vectors in the streamwise component $v(x)$, ϕ_E is the same velocity vector computed in Elmer and ϕ_M is that one computed in Matlab according to the model case in [15]. Table (6.3) resumes the results of this study, where N denotes the elements along $a + c$ and M those in b as shown in figure (6.4); since the air region is not interesting from a computational point of view, the number of elements for the discretization has been kept: $c = 50$, whereas the amount in the liquid zinc region has been increased. Moreover, it has not been possible to raise $N > 350$ and $M > 700$ due to memory problems in Elmer for this domain.

$N \times M$	600	650	700
250	$3.99 \cdot 10^{-7}$	$2.67 \cdot 10^{-7}$	$2.17 \cdot 10^{-7}$
300	$2.64 \cdot 10^{-7}$	$2.83 \cdot 10^{-7}$	$3.03 \cdot 10^{-7}$
350	$3.96 \cdot 10^{-7}$	$4.71 \cdot 10^{-7}$	$4.58 \cdot 10^{-7}$

Table 6.3: Computation of the Euclidean norm L_e^2 for different meshes

One can identify the best combination of elements in 250×700

Chapter 7

The Hartmann Problem

Before explaining the effects of a magnetic field on the liquid film, it is worth describing a typical MHD problem. The Hartmann problem example has been taken from [11] and is at the base of the theory behind the electromagnetic control for the jet wiping process. The interaction between the velocity field of an electrically conductive fluid and an applied external magnetic field generates an electric current. In turn, the interplay between the current and the magnetic field produces the Lorentz force which, acting downwards, pushes the fluid down.
This example is essential to understand how implementing an MHD problem in Elmer. In particular, the chapter shows the accuracy of Elmer's computation by the comparison between Elmer's resolution and that one presented [11] for the same Hartmann problem. The effects on the velocity field, described above, have not been considered.
To simplify the writing of the equation in terms of length, the notation $(\bullet)_{,i}$ with $i = (x, y, z)$ has been used to indicate the partial derivative of the variable $(\bullet)$ with respect to direction i.

7.1 Problem Description

There is an electrically conductive, viscous and incompressible fluid between parallel plates with an applied transverse steady and uniform magnetic field $\mathbf{B}^e = [0, B_0]$. The flow has been assumed to be fully developed and in presence of a constant pressure gradient along the x direction ($\partial p_l/\partial x = con$). The channel extent in the z direction is much greater than that one in the y direction so that no variations occur along z. σ_m is the electrical conductivity, μ is the magnetic permeability and μ_l is the dynamic viscosity for the fluid. The problem described above is taken from [11].

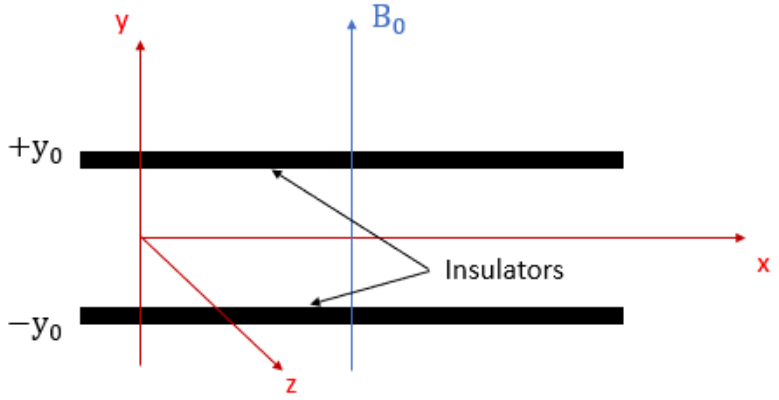


Figure 7.1: The physical domain of the Hartmann problem.

7.1.1 Magnetic Induction Equation

Elmer solves the magnetic induction equation which describes the interaction of a conductive liquid or gas with applied or generated magnetic fields. Hereunder, a second way to ensue this equation, with respect to that one reported in chapter 2, has been described.
Start from the Ohm's law (2.10) $\mathbf{J} = \sigma_m(\mathbf{E} + \mathbf{U} \times \mathbf{B})$ and replace the right member of the Ohm's law with the Ampere-Maxwell's one: $\nabla \times \mathbf{H} = \mathbf{J}$

$$\nabla \times \mathbf{H} = \sigma_m(\mathbf{E} + \mathbf{U} \times \mathbf{B}) \tag{7.1}$$

Afterwards the curl operator is applied to (7.1):

$$\nabla \times (\nabla \times \mathbf{H}) = \sigma_m \nabla \times (\mathbf{E} + \mathbf{U} \times \mathbf{B}) \tag{7.2}$$

Now, $\mathbf{H}$ is replaced with the constitutive equation: $\mathbf{B} = \mu\mathbf{H}$ and the identity vector operation can be used:

$$\frac{1}{\mu}\nabla \times \nabla \times \mathbf{B} = \frac{1}{\mu}\nabla(\nabla \cdot \mathbf{B}) - \frac{1}{\mu}\nabla^2\mathbf{B} \tag{7.3}$$

By knowing that the induced magnetic field is a solenoidal vector field $\nabla \cdot \mathbf{B} = 0$, one can delete the first term on the equation's left hand.

$$\frac{\cancel{\nabla(\nabla \cdot \mathbf{B})}}{\sigma_m\mu} - \frac{\nabla^2\mathbf{B}}{\sigma_m\mu} = \nabla \times \mathbf{E} + \nabla \times (\mathbf{U} \times \mathbf{B}) \tag{7.4}$$

Eventually, by employing the Faraday's law: $\nabla \times \mathbf{E} = -\frac{\partial \mathbf{B}}{\partial t}$, the final form of the magnetic induction equation can be written:

$$\boxed{\frac{\partial \mathbf{B}}{\partial t} - \frac{1}{\sigma_m\mu}\nabla \times \nabla \times \mathbf{B} - \nabla \times (\mathbf{U} \times \mathbf{B}) = 0} \tag{7.5}$$

The magnetic flux density induction equation is always coupled to the Navier-Stokes equation by its velocity term ($\mathbf{U}$) of the fluid. In turn, the magnetic field produces the Lorentz force which appears as volume force in the Navier-Stokes momentum equation (2.1b).

$$\mathbf{F_L} = \mathbf{J} \times \mathbf{B} \tag{7.6}$$

7.2 Elmer Method

Taking advantage of the linearity of equation (7.5), in Elmer each term of the magnetic induced flux **B** in the magnetic induction equation is read as

$$\mathbf{B} = \mathbf{B}^i + \mathbf{B}^e$$

where the first element is the induced magnetic component in the fluid and the second one is the external magnetic field (produced by an external magnet).

$$\frac{\partial(\mathbf{B}^i + \mathbf{B}^e)}{\partial t} - \frac{1}{\sigma_m \mu}\nabla \times \nabla \times (\mathbf{B}^i + \mathbf{B}^e) - \nabla \times [\mathbf{U} \times (\mathbf{B}^i + \mathbf{B}^e)] = 0 \tag{7.7}$$

The external magnetic field may be defined in the material section. In this case $\mathbf{B}^e$ is applied along y as in figure (7.1). By considering figure (7.2) and the physical properties listed in table (7.1), the external magnetic field can be computed by the definition of Hartmann number in [11]

$$H_a = \frac{b}{2}\mathbf{B}^e\sqrt{\frac{\sigma_m}{\mu_l}} \rightarrow \mathbf{B}^e = \frac{H_a}{\frac{b}{2}\sqrt{\frac{\sigma_m}{\mu_l}}} = 0.0134\ T$$

H_a	15
$b/2$	$0.015\ m$
σ_m	$1.6 \cdot 10^7\ S/m$
μ_l	$0.0029\ Pa \cdot s$
μ	$1.2 \cdot 10^{-6}\ N/A^2$
ρ_l	$6500\ Kg/m^3$

Table 7.1: Physical values adopted in the simulation

Moreover, Elmer assumes that the sources of the external field are outside the flow region, i.e.:

$$\nabla \times \mathbf{B}^e = 0$$

Thereby, in this problem the equation (7.5) can be expanded as:

$$\begin{cases} B^i_{x,t} + \frac{B^i_{x,xx}+B^i_{x,yy}}{\mu\sigma_m} - \partial_y(uB_y - \cancel{vB_x}) = 0 \\ B^i_{y,t} + \frac{B^i_{y,xx}+B^i_{y,yy}}{\mu\sigma_m} - \partial_x(\cancel{vB_x} - uB_y) = 0 \end{cases} \tag{7.8}$$

Where the term v has disappeared since the flow develops only in the x direction.

$$\begin{cases} B^i_{x,t} + \frac{B^i_{x,xx}+B^i_{x,yy}}{\mu\sigma_m} - \partial_y[u(B^i_y + B^e_y)] = 0 \\ B^i_{x,t} + \frac{B^i_{x,xx}+B^i_{x,yy}}{\mu\sigma_m} + \partial_x[u(B^i_y + B^e_y)] = 0 \end{cases} \tag{7.9}$$

After the expansion of the derivatives and by considering asymptotic conditions, the equation (7.9) can be recast as

$$\begin{cases} \frac{B^i_{x,xx}+B^i_{x,yy}}{\sigma_m\mu} - u_{,y}B^i_y - uB^i_{y,y} - u_{,y}B^e_y = 0 \\ \frac{B^i_{y,xx}+B^i_{y,yy}}{\sigma_m\mu} + uB^i_{y,x} = 0 \end{cases} \tag{7.10}$$

Hence, from (7.10), Elmer computes the induced magnetic flux density $\mathbf{B}^i$, then the induced magnetic field $\mathbf{H}^i$ can be obtained by the relation

$$\mathbf{H}^i = \frac{\mathbf{B}^i}{\mu}$$

Besides, from the Ampere's law the induced current $\mathbf{J}$ results:

$$\mathbf{J} = \nabla \times \mathbf{H}^i = \frac{1}{\mu} \begin{pmatrix} B^i_{z,y} - B^i_{y,z} \\ B^i_{x,z} - B^i_{z,x} \\ B^i_{y,x} - B^i_{x,y} \end{pmatrix} \tag{7.11}$$

Nevertheless, from the previous hypothesis, the current density is the only obtained component J_z since there is no magnetic field contribution in z

$$J_z = \frac{1}{\mu} \begin{pmatrix} \cancel{B^i_{z,y}} - \cancel{B^i_{y,z}} \\ \cancel{B^i_{x,z}} - \cancel{B^i_{z,x}} \\ B^i_{y,x} - B^i_{x,y} \end{pmatrix} = \frac{1}{\mu}(B^i_{y,x} - B^i_{x,y}) \tag{7.12}$$

Eventually, by knowing the current density, Elmer computes the Lorentz force (7.6)

$$F_{L_x} = -J_z[(B^i_{y,x} - B^i_{x,y}) + B_0] \tag{7.13}$$

This magnitude appears as a volume force term in the Navier-Stokes flow momentum equation (2.1b).

7.2.1 Geometry and Meshing of the Domain

According to the example described in [18], the domain, illustrated in figure (7.2), has been assembled in Gmsh

- a=0,18 m
- b=0,03 m

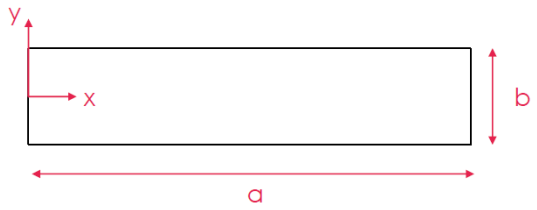


Figure 7.2: Geometry adopted for the Hartmann Problem

Concerning the mesh, it is unstructured and with a number of elements

$$a = 180 \quad b = 30$$

According to the geometry depicted in figure (7.2). Figure (7.3) displays a portion of the unstructured mesh.

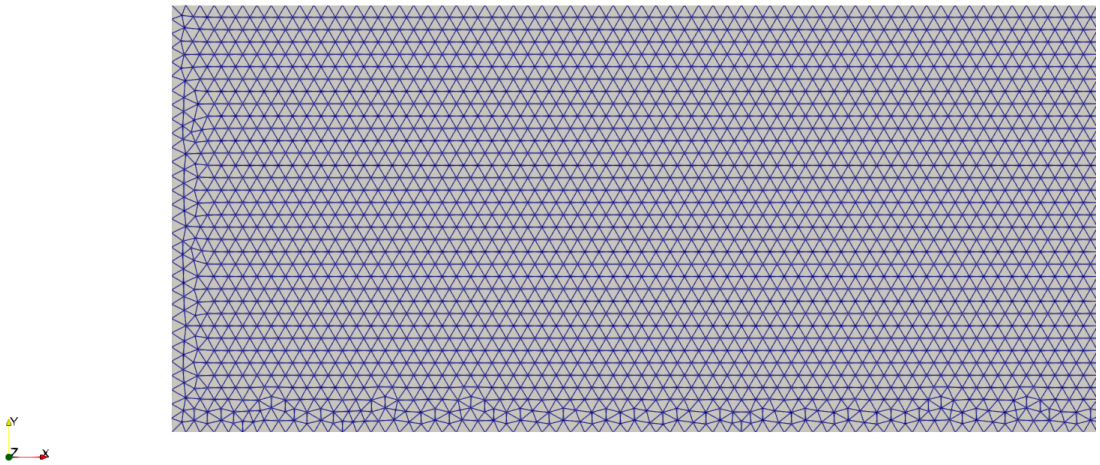


Figure 7.3: A portion of the unstructured mesh realized in Gmsh and plotted in ParaView

7.2.2 Initial Conditions and Boundary Conditions

As done for the geometry, the numerical data of the variables have been taken from [18], the imposed velocity is

$$\mathbf{U} = (u, v) = (0.0015, 0)\ m/s$$

In the section "Initial condition", it has been possible to define the values of the dependent variables in the whole domain at $t = 0\ s$.

$$u = 0$$

$$v = 0$$

The equations system resolved by Elmer is presented, as follows

$$\begin{cases} \nabla \cdot \mathbf{U} = 0 \quad Mass \quad conservation \quad equation \\ \rho_{l,g} \left(\frac{\partial \mathbf{U}}{\partial t} + (\mathbf{U} \cdot \nabla)\mathbf{U} \right) - \nabla(2\mu_{l,g}\mathbf{E}_{l,g}) + \nabla p_{l,g} = \rho_{l,g}\mathbf{F}_L \quad Momentum \quad equation \\ \frac{\partial \mathbf{B}}{\partial t} - \frac{1}{\sigma_m \mu}\nabla \times \nabla \times \mathbf{B} - \nabla \times (\mathbf{U} \times \mathbf{B}) = 0 \quad Magnetic \quad induction \quad equation \end{cases}$$

Regarding the boundary conditions in Elmer, each side has a specific boundary value

Inlet

Dirichlet conditions for the velocity components u and v have been chosen

$$u = 0.0015\ m/s$$

$$v = 0$$

Walls

A homogeneous Dirichlet condition for $\mathbf{U}$ has been elected by setting no slip wall as reported below

$$u = 0$$

$$v = 0$$

Moreover, the induced magnetic field $\mathbf{B}^i$ has a zero value at the border (homogeneous Dirichlet) in the two directions.

$$B_x^i = 0$$
$$B_y^i = 0$$

This means, according to what said in 2.2.2, that the induced magnetic flux normal component $(\mathbf{B}^i \cdot \hat{n})$ does not change between the two means (zinc/air).

Outlet

The value for the velocity component along y has been specified

$$v = 0$$

This implies a homogeneous Neumann condition for the component along x

$$\frac{\partial u}{\partial y} \cdot \hat{n} = 0$$

When there is no boundary value for the induced magnetic field $\mathbf{B}^i$, a homogeneous Neumann condition is assumed automatically by Elmer

$$\nabla \mathbf{B}^i \cdot n = 0$$

This means, according to what said in 2.2.2, that the induced magnetic field tangent component $(\mathbf{H}^i \times \hat{n})$ is equal to less than a jump due to a possible surface current $(\mathbf{J}_s)$ determined by magnetization **M**. Since the condition is homogeneous, no current is present on the border [3]. Figure (7.4) resumes the used initial and boundary conditions. The full script of the simulation is available in the appendix 11

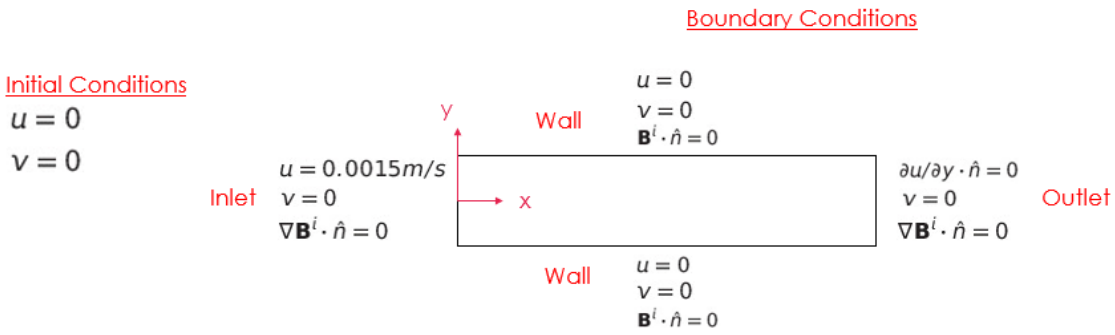


Figure 7.4: Initial and Boundary Conditions applied in Elmer

Actually, the induced component $\mathbf{B}^i$ is much smaller than B_0 and can be neglected, as has been stated and demonstrated below. Nevertheless, Elmer considers both components.

7.3 Book Method

It is worth writing the equation (7.5) in its dimensionless form. To get it, one has to introduce the dimensionless variables

$$\hat{\mathbf{B}} = \frac{\mathbf{B}}{B_0} \quad \hat{t} = t\frac{U_0}{L} \quad \hat{\mathbf{r}} = \frac{\mathbf{r}}{L}$$

Where $\mathbf{r}$ is the coordinates vector, L is the characteristic length and U_0 is the characteristic velocity. Thus, equation (7.5) can be recast as

$$\frac{\partial \hat{\mathbf{B}}}{\partial \hat{t}} = \frac{1}{R_m}\hat{\nabla} \times (\hat{\nabla} \times \hat{\mathbf{B}}) + \hat{\nabla} \times (\hat{\mathbf{U}} \times \hat{\mathbf{B}})$$

which indicates the significance of the magnetic Reynolds number R_m

$$R_m = U_0 L \sigma_m \mu$$

Where the first term on the right side is a diffusion term and the second a convection. The main hypothesis, lead in [11], is to assume that the magnetic Reynolds (R_m) is much lower than one, since the problem is on a laboratory scale.

$$R_m << 1$$

Thus, being the external magnetic field uniform and steady and thanks to the hypothesis mentioned above, it is easy to demonstrate that the magnetic induction field is negligible compared to the external one thanks to the definition of R_m as the ratio between the magnetic convection and the magnetic diffusion

$$R_m = \frac{\hat{\nabla} \times (\hat{\mathbf{U}} \times \hat{\mathbf{B}}^i)}{\hat{\nabla} \times (\hat{\nabla} \times \hat{\mathbf{B}}^i)} << 1$$

By knowing $\mathbf{B}^e = (0, B_0)$ and the boundary conditions,

$$u = 0 \quad y = \pm y_0$$

from the momentum conservation equation in x

$$\frac{\partial p_l}{\partial x} - \mu_l \frac{d^2 u}{dy^2} + \sigma_m u B_0^2 = 0 \tag{7.14}$$

the velocity in the x direction can be computed

$$u = \frac{y_0^2}{H_a^2}\frac{1}{\mu_l}\frac{\partial p}{\partial x}\left(\frac{\cosh(H_a y/y_0)}{\cosh H_a} - 1\right)$$

Then, the current can be computed by matching the equation above in the Ohm's law:

$$\mathbf{J} = \sigma_m(\mathbf{E} + \mathbf{U} \times \mathbf{B})$$

where the applied electric field **E** has not been considered

$$J_z = \sigma_m B_0 u = \sigma_m B_0 \frac{y_0^2}{H_a^2}\frac{1}{\mu_l}\frac{\partial p}{\partial x}\left(\frac{\cosh(H_a y/y_0)}{\cosh H_a} - 1\right) \tag{7.15}$$

Afterwards, from the integration along y of equation (7.15), the induced magnetic field $\mathbf{H}^i$ results

$$H_x^i = \frac{y_0}{H_a}\frac{\partial p_l}{\partial x}\sqrt{\frac{\sigma_m}{\mu_l}}\left(\frac{y_0\sinh(H_a y/y_0)}{H_a\cosh(H_a)} - y\right) \tag{7.16}$$

It is clear that the induced magnetic field has just one component. Eventually, the Lorentz force has been calculated as the superposition of the component determined by $\mathbf{B}^e$ and that one by $\mathbf{B}^i$ thanks to the linearity of the magnetic induction equation

$$F_{L_x} = F^e_{L_x} + F^i_{L_x} = J_z(\mathbf{B}^e + \mathbf{B}^i) = \sigma_m u B_0\left[B_0 + \mu\frac{y_0}{H_a}\frac{\partial p_l}{\partial x}\sqrt{\frac{\sigma_m}{\mu_l}}\left(\frac{y_0\sinh(H_a y/y_0)}{H_a\cosh(H_a)} - y\right)\right] \tag{7.17}$$

7.4 Matching Between Elmer and the Book

As result of the simulation, the accordance between the induced magnetic field component $\mathbf{B}^i$ (7.12) computed in Elmer and that one from equation (7.16) according to [11] has been displayed in figure (7.5)

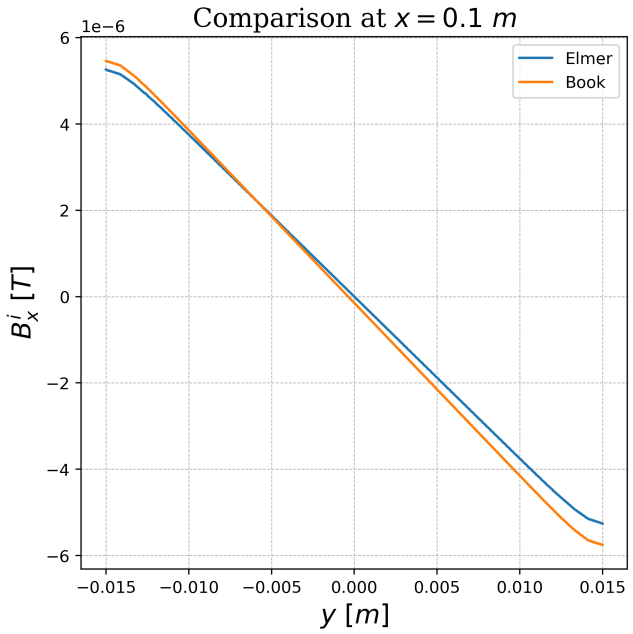


Figure 7.5: The accordance of the induced magnetic field $\mathbf{B}^i$ between Elmer and the book results

Figure (7.6) displays the comparison between the two components of the Lorentz force of the equation (7.17), which considers $\mathbf{B}^e$ and $\mathbf{B}^i$.

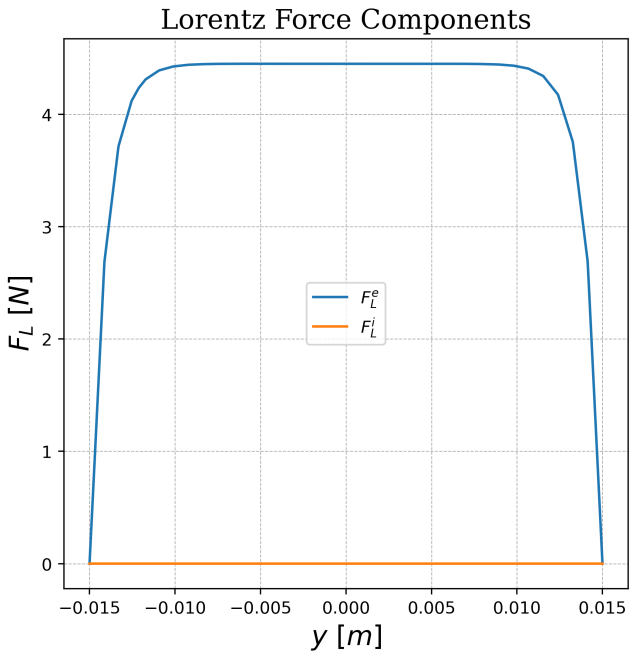


Figure 7.6: Difference between the two components of the Lorentz force: inner Lorentz force ($\mathbf{F}_L^i$) and outer Lorentz force ($\mathbf{F}_L^e$)

It is clear that the hypothesis of neglecting the induced magnetic field component has been confirmed. Eventually, figure (7.7) points out that the Lorentz Force calculated by Elmer (7.13) and that one from equation (7.17) in [11] are coincident. This implies that the assumption, mentioned above, is worth for Elmer as well.

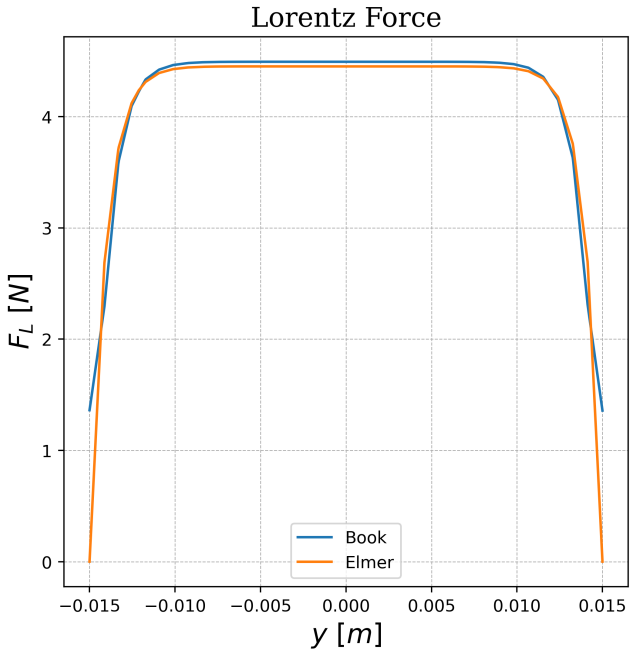
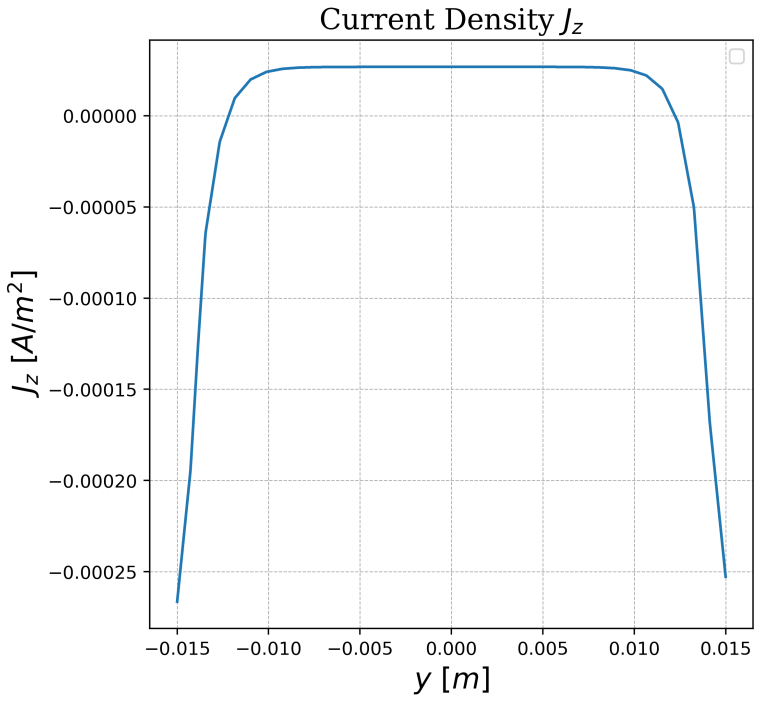


Figure 7.7: The accordance between the Lorentz force computed by Elmer and the book [11]

Nevertheless, it is also evident a small difference at the edges of the domain, indeed Elmer shows a no-zero result in contrast to the equation (7.17). This is given by a no-zero induced current density J_z on the borders as visible in figure (7.8)



The p

Figure 7.8: Plot of the current density J_z computed by Elmer

Actually, if the two components of the equation (7.12) which constitute J_z are plotted, it is clear to discover that the component $B^i_{y,x}$ is much smaller than $B^i_{x,y}$, as shown in figure (7.9)

$$J_z = \frac{1}{\mu}(\cancel{B^i_{y,x}} - B^i_{x,y}) = -\frac{B^i_{x,y}}{\mu}$$

Hence, only $B^i_{x,y}$ determines the sign and the existence of the induced current density.

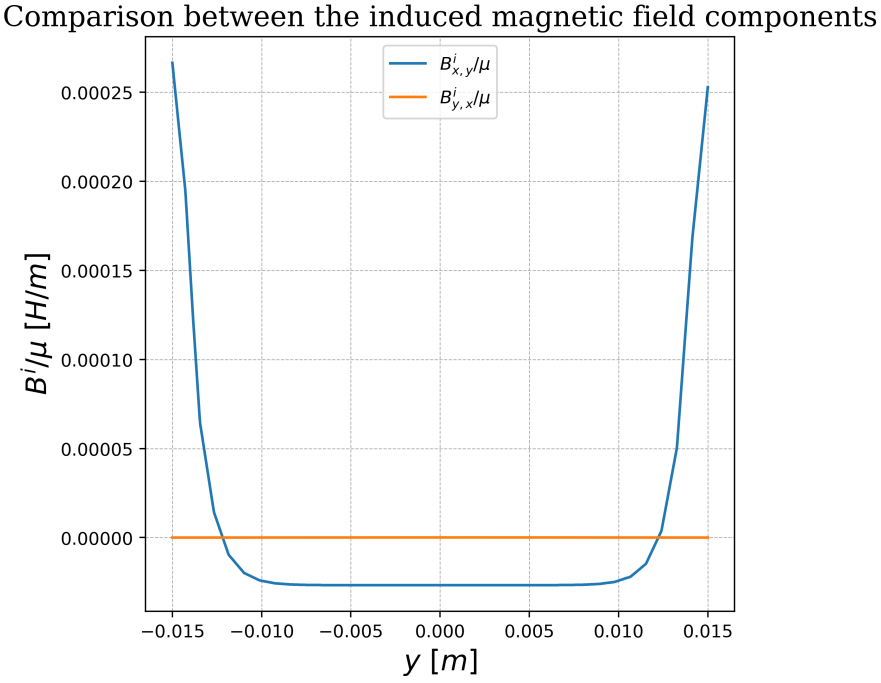


Figure 7.9: Difference between $B^i_{y,x}$ and $B^i_{x,y}$

Chapter 8

Steady Liquid Film Model: Expected Velocity Profile in the Presence of a Magnetic Field as Initial Condition

In the light of the Hartmann problem's example described in the previous chapter, hereunder, the objective is to validate the velocity profile computed in [19] in presence of an uniform magnetic field. After the jet wiping region, the liquid film meets an uniform magnetic field produced by permanent magnets. This affects the cross-stream velocity profile which is reduced due to the Lorentz force acting downwards as described in chapter 1.

8.1 Domain

The domain is the same as in chapter 6, but with the presence of a uniform magnetic field as depicted in figure (8.1)

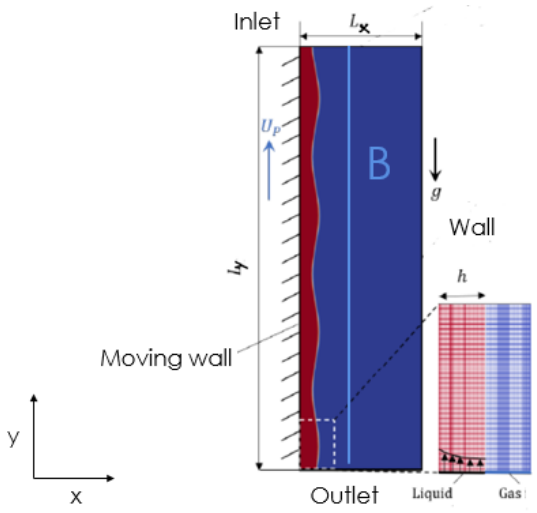


Figure 8.1: Schematic of the flow configuration used for validation purposes in Elmer: flow domain and zoom on the near-wall mesh in. B indicates the uniform magnetic field [16]

Where

$$L_y = 0.02\ m \quad L_x = 0.001\ m$$

8.2 Meshing

As said before in chapter 6, to capture a smoother area for the computation, the interface has been collocated in the mesh at $2h = 1.51 \cdot 10^{-4}\ m$ in the x direction, as in figure (8.2)

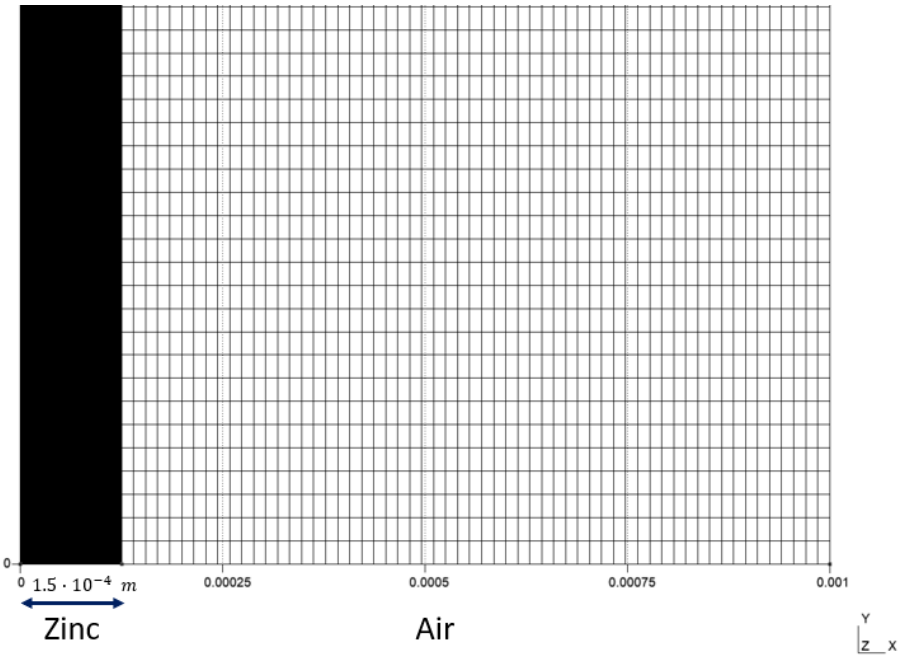


Figure 8.2: Position of the interface for the meshing in Gmsh

In the light of the previous independent mesh study, the optimum distribution of the mesh elements has been employed in this simulation, as shown in figure (8.3)

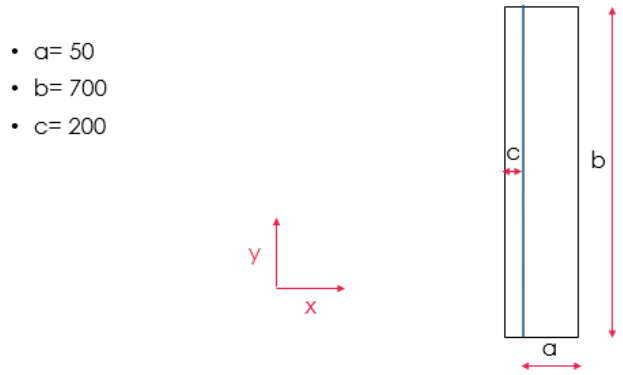


Figure 8.3: Distribution of the mesh elements in the domain

8.3 Test Case Description

The simulation input file has many parts in common with the previous case in chapter 6. As done so far, the specific distinct parts of the test case have been pointed out; the reader may look at the full script in Appendix 11.

8.3.1 Simulation Set-Up

The same tricks for the evolution in time of the case without the magnetic field have been adopted in this example. Indeed, an adaptive time step has been set with a starting value of $0.01\ s$; then it changes by keeping the Courant number until it reaches a maximum of 10000 iterations. Thus, the total computational time is

$$T = 0.01 \times 10000 = 100\ s$$

The CFL condition is again given by

$$CFL = \frac{\Delta t U}{\Delta x} = 0.5$$
$$U_p = 0.5 \quad m/s \quad \Delta x = 7.4 \cdot 10^{-7}\ m$$

8.3.2 Body Forces

According to [16] the body force is gravitational and its value and direction can be specified in the section "Body Force" of the decision tree diagram in Elmer. As one can see in figure (8.1), the force direction is along y downwards. In the same section, Lorentz force must be specified since it appears as volume force in the Navier-Stokes momentum conservation equation.

8.3.3 Initial Conditions

The level-set free surface has been defined as in the case without magnetic field 6.3.3, while the initial velocity profile has been taken from [19].
Since the reference system in [19] is different from the one in Elmer, the equations (3.31) and (3.33) have been held, but the boundary conditions have been adapted according to the Elmer reference system.

$$\begin{gathered} v(0) = -U_p \quad for \quad x = 0 \quad kinematic \quad condition \\ p_l - p_g(x) + \sigma_m \partial_{xx} h = 0 \quad and \quad \partial_y u = \frac{1}{\mu_l}\tau_g(y) = 0 \quad dynamic \quad conditions \end{gathered} \tag{8.1}$$

The shear-stress term τ_g has been neglected. The steps for the computation of the velocity profile are the same ones described in [19]; thus, by adding (8.1), $v(x)$ results, as follows

$$\hat{v}(\hat{x}) = \left\{ \frac{\sinh\left(\hat{h}H_a\hat{B}\right)\left(1+\frac{1}{H_a^2\hat{B}^2}\right)}{\cosh\left(\hat{h}H_a\hat{B}\right)} \right\} \sinh\left(\hat{x}H_a\hat{B}\right) - \cosh\left(\hat{x}H_a\hat{B}\right)\left(1+\frac{1}{H_a^2\hat{B}^2}\right) + \frac{1}{H_a^2\hat{B}^2} \tag{8.2}$$

Where $H_a = (\sigma_m [B]^2 U_p / g\rho)^{1/2}$ is the Hartmann number. Since the equation (8.2) is dimensionless, it has been dimensionalized for the computation in Elmer. To get it, it needs to recall that any dimensional variable is given by the product between its reference quantity ($[\bullet]$) and the dimensionless form ($\hat{\bullet}$) described in 3.2.1.

$$\hat{v} = \frac{v}{[v]} \quad \hat{x} = \frac{x}{[x]} \quad \hat{B} = \frac{B}{[B]} \quad \hat{h} = \frac{h}{[h]} \tag{8.3}$$

Where

$$[v] = U_p \quad [h] = \sqrt{\frac{\nu_l U_p}{g}} \quad [x] = [h] \quad [B] = B \tag{8.4}$$

Using (8.3) allows recasting the equation (8.2) in its dimensional form.

$$v(x) = -U_p \Bigg[\tanh\left(\frac{h}{[h]} H_a \frac{B}{[B]}\right) \left(1 + \frac{1}{H_a^2 (B/[B])^2}\right) \sinh\left(\frac{x}{[x]} H_a \frac{B}{[B]}\right) + \\ - \cosh\left(\frac{x}{[x]} H_a \frac{B}{[B]}\right) \left(1 + \frac{1}{H_a^2 (B/[B])^2}\right) + \frac{1}{H_a^2 (B/[B])^2} \Bigg]$$

The ratio $B/[B] = 1$ has been simplified, being the magnetic field uniform.

$$v(x) = -U_p \left[\tanh\left(\frac{h}{[h]} H_a\right) \left(1 + \frac{1}{H_a^2}\right) \sinh\left(\frac{x}{[h]} H_a\right) - \cosh\left(\frac{x}{[h]} H_a\right) \left(1 + \frac{1}{H_a^2}\right) + \frac{1}{H_a^2} \right] \tag{8.5}$$

Where h is the film thickness

$$h = \hat{h}[h] = \hat{h}\sqrt{\frac{\nu_l U_p}{g}} = 7.51 \cdot 10^{-5}\ m$$

$$U_p = 0.5\ m/s \quad g = 9.81\ m/s^2 \quad \nu_l = \frac{\mu_l}{\rho_l} = 4.461 \cdot 10^{-7}\ m^2/s \quad \hat{h} = 0.5$$

Equation (8.5) is the final form implemented in Elmer.

8.3.4 Material

In addition to what has already been defined in 6.3.4, the presence of an applied magnetic field has been added in the "Material" section in Elmer by calling it B_0 in wall-normal direction. This software computes the induced magnetic field in the material caused by the external source according to the equation (7.7). The properties of the zinc film and air are listed in table (8.1)

μ	$1.2 \cdot 10^{-6}\ N/A^2$
σ_m	$1.6 \cdot 10^{7}\ S/m$
μ_l	$0.0029\ Pa \cdot s$
μ_g	$1 \cdot 10^{-5}\ Pa \cdot s$
ρ_l	$6500\ kg/m^3$
ρ_g	$1\ kg/m^3$

Table 8.1: Physical values adopted in the simulation

8.3.5 Solvers

Five solvers have been used.

- 2 for level-set equation: computation of the interface and its reinitialization
- 1 for Navier-Stokes equations
- 1 for magnetic induction equation
- 1 for post-processing

The full equations system, solved by Elmer, is presented, as follows

$$\begin{cases} \nabla \cdot \mathbf{U} = 0 \quad Mass \quad conservation \quad equation \\ \rho_{l,g}\left(\frac{\partial \mathbf{U}}{\partial t} + (\mathbf{U}\cdot\nabla)\mathbf{U}\right) - \nabla(2\mu_{l,g}\mathbf{E}_{l,g}) + \nabla p_{l,g} = \rho_{l,g}\mathbf{f}_{ext} \quad Momentum \quad equation \\ \frac{\partial \phi}{\partial t} + \mathbf{U}\cdot\nabla\phi = 0 \quad Level-set \quad equation \\ \frac{\partial \mathbf{B}}{\partial t} - \frac{1}{\sigma_m \mu}\nabla\times\nabla\times\mathbf{B} - \nabla\times(\mathbf{U}\times\mathbf{B}) = 0 \quad Magnetic \quad induction \quad equation \end{cases}$$

As already declared before in 6.3.5, the umfpack has been utilized as a direct method for linear system resolution, due to the linearity of the equations, while the Newton iteration method has been applied for the non-linear convection term in Navier-Stokes. Figure (8.4) illustrates the steps followed by Elmer for the computation of the solution

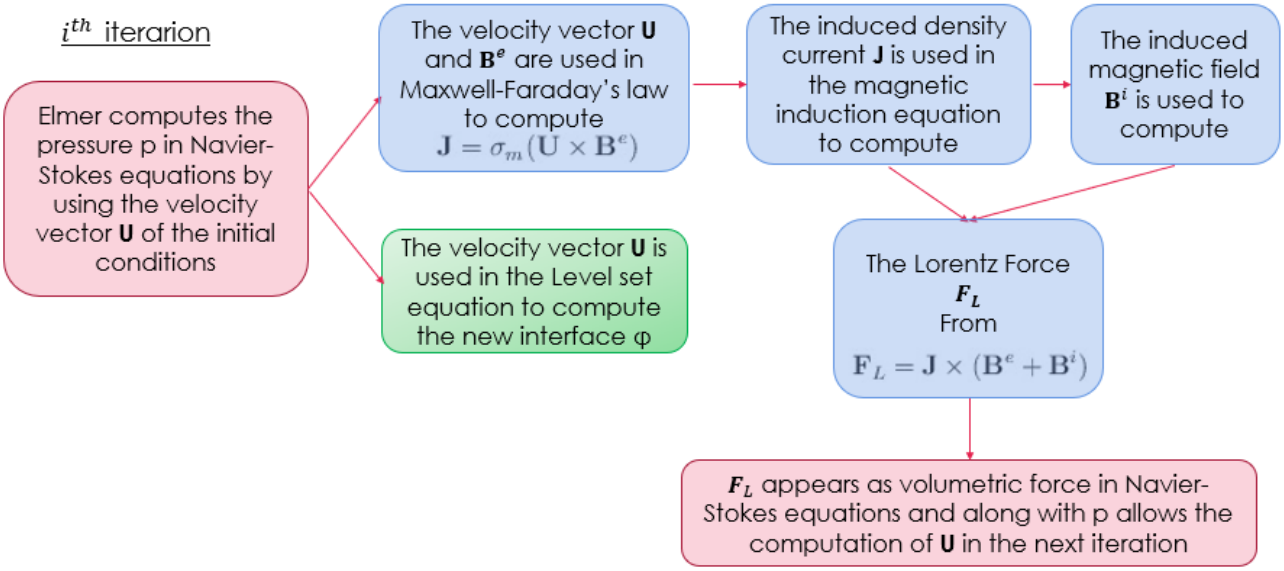


Figure 8.4: Order of the resolution of the equation in Elmer

To stabilize the simulation, the relaxation factor's decrease of the non-linear system is needed. This reduction implies very small changes between the computation of two subsequent solutions. This factor λ is needed to improve the convergence of the nonlinear system; however, λ smaller than 0.3 slows it down. The relaxed variable replacing the second-order term is defined as

$$\mathbf{U}'_n = \lambda\mathbf{n} + (1-\lambda)\mathbf{U}_{n-1} \tag{8.6}$$

This means that the Newton method with (8.6) is now

$$(\mathbf{U_{n+1}} \cdot \nabla)\mathbf{U_{n+1}} \approx (\mathbf{U_n} \cdot \nabla)\mathbf{U_{n+1}} + (\mathbf{U_{n+1}} \cdot \nabla)\mathbf{U_n} - \nabla \mathbf{U}_n \tag{8.7}$$

In this simulation the relaxation factor has been chosen as follows

$$\lambda = 0.1$$

8.3.6 Boundary Conditions

The interested domain as depicted in figure (8.1) has four boundary conditions according to the numerical case in [16]. The boundary conditions for this test case concern a new property for the magnetic induction equation.

Wall

Homogeneous Dirichlet boundary conditions have been applied for the two velocity components. It means to write in Elmer

$$u = 0$$
$$v = 0$$

Conversely, nothing has been specified for the induction equation; it results a homogeneous Neumann condition for the induced magnetic field component $\mathbf{B}^i$

$$\nabla \mathbf{B}^i \cdot \hat{n} = 0$$

Moving Wall

The wall velocity in the streamwise direction has been defined.

$$v = U_p$$

Whereas for the wall-normal component, a homogeneous Dirichlet condition has been applied.

$$u = 0$$

By concerning the induction equation a homogeneous Dirichlet condition referred to the induced magnetic field component $\mathbf{B}^i$ has been set

$$B_x^i = 0$$
$$B_y^i = 0$$

This means, according to what said in 2.2.2, that the induced magnetic flux normal component ($\mathbf{B}^i \cdot \hat{n}$) does not change between the two means (zinc/air).

Inlet and Outlet

The wall-normal velocity has a value equal to zero.

$$u = 0$$

Conversely, in the streamwise velocity, no numerical quantity has been written; it involves a homogeneous Neumann boundary condition implicitly.

$$\frac{\partial v}{\partial x} \cdot \hat{n} = 0$$

The same condition has been associated with the induced magnetic field in the two directions.

$$\nabla \mathbf{B}^i \cdot \hat{n} = 0$$

This means, according to what said in 2.2.2, that the induced magnetic field tangent component ($\mathbf{H}^i \times \hat{n}$) is equal to less than a jump due to a possible surface current ($\mathbf{J}_s$) determined by magnetization **M**. Since the condition is homogeneous, no current is present on the border [3]

Figure (8.5) recaps the initial and boundary conditions used for this simulation

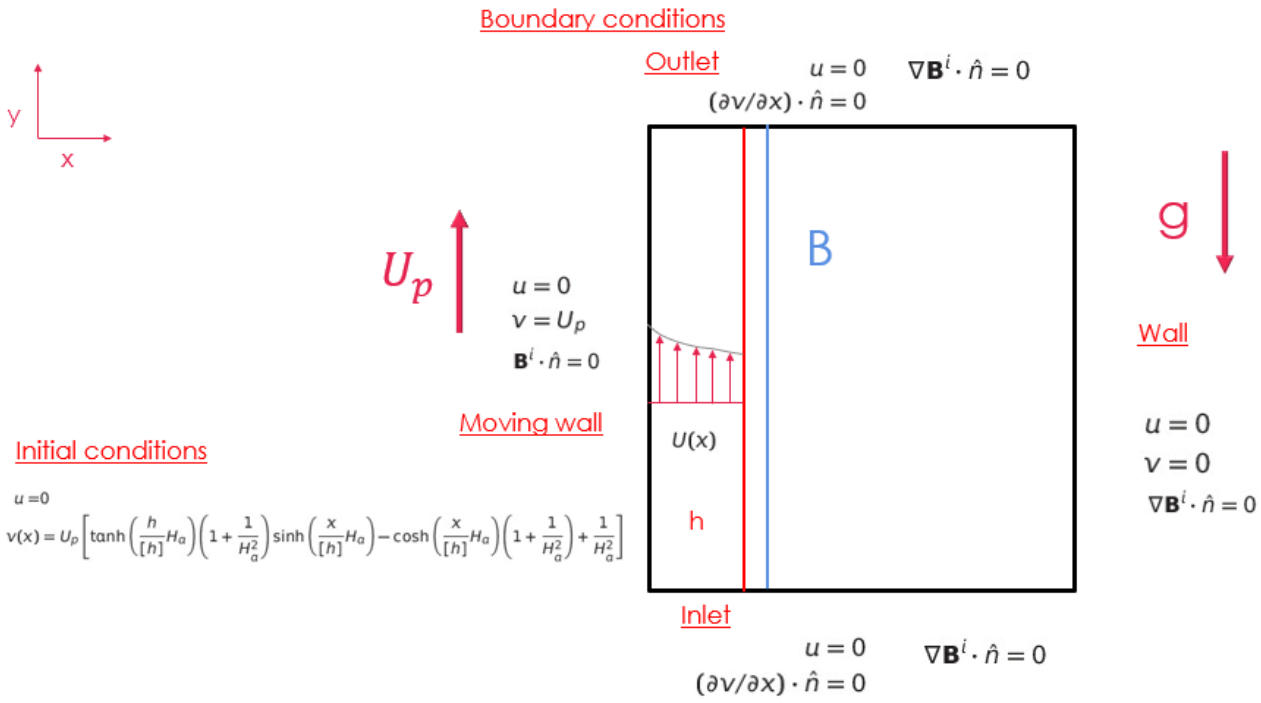


Figure 8.5: Full set of initial and boundary conditions used for the test case

8.4 Post-processing

The velocity of the moving wall is

$$U_p = 0.5 \quad m/s$$

A higher speed has led to unacceptable results and far from those expected in [16] due to divergence issues. This last problem also has occurred when the external magnetic field is high $\mathbf{B}^e > 0.5\ T$; thereby, the simulation has been run applying a small one.

$$\mathbf{B}^e = (B_0, 0) = 0.5\ T$$

By knowing $B = [B] = B_0$, the Hartmann number, appearing in the equation (8.5), can be easy computed

$$H_a = [B]\sqrt{\frac{\sigma_m U_p}{g\rho_l}} = 5.6$$

Hence, after every variable and parameter have been defined, Elmer has computed the cross-stream velocity profile (8.5). Figure (8.6) shows the comparison between Elmer's cross velocity profile and the one computed in Matlab (Appendix 11) by providing the velocity profile (8.5) taken from [19]. Data have been collected in the liquid film at position $y = 0.01\ m$ of the interest domain and for a dimensionless value $\hat{h} = 0.5$

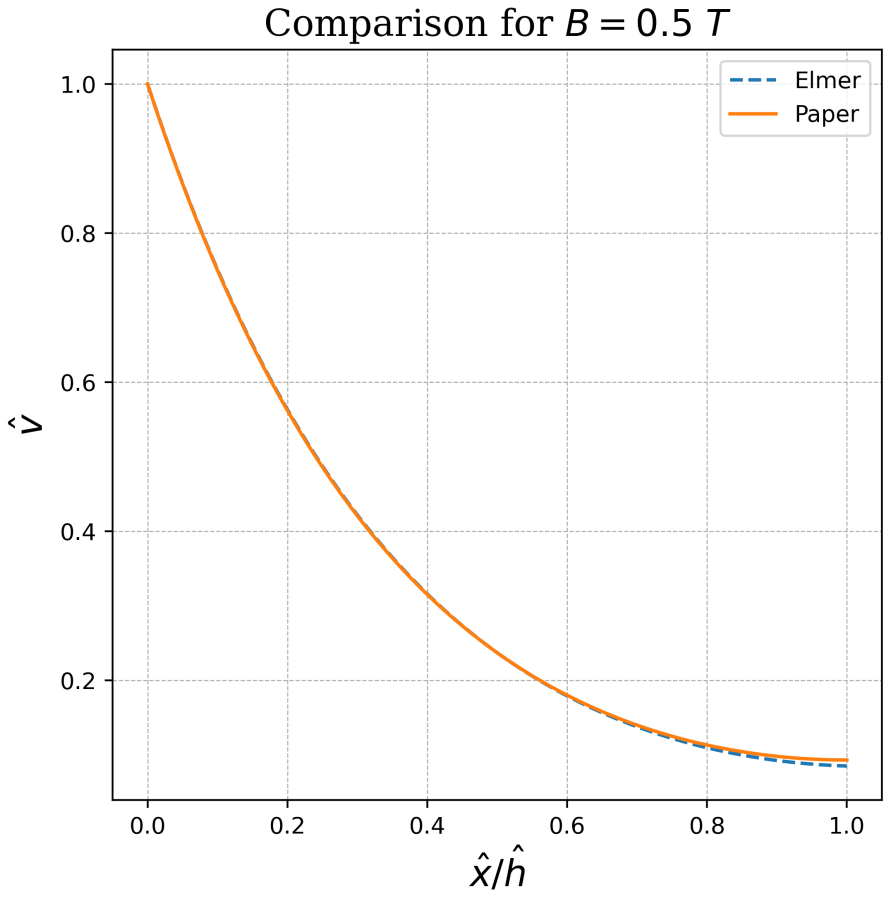


Figure 8.6: Comparison between Elmer's result and the velocity profile computed in [19] marked as "paper" in the legend.The variables are in their dimensionless form

One can note the coincidence of both computations, Figure (8.7) displays the reduction of the velocity profile compared to the one in chapter 6 without a magnetic field

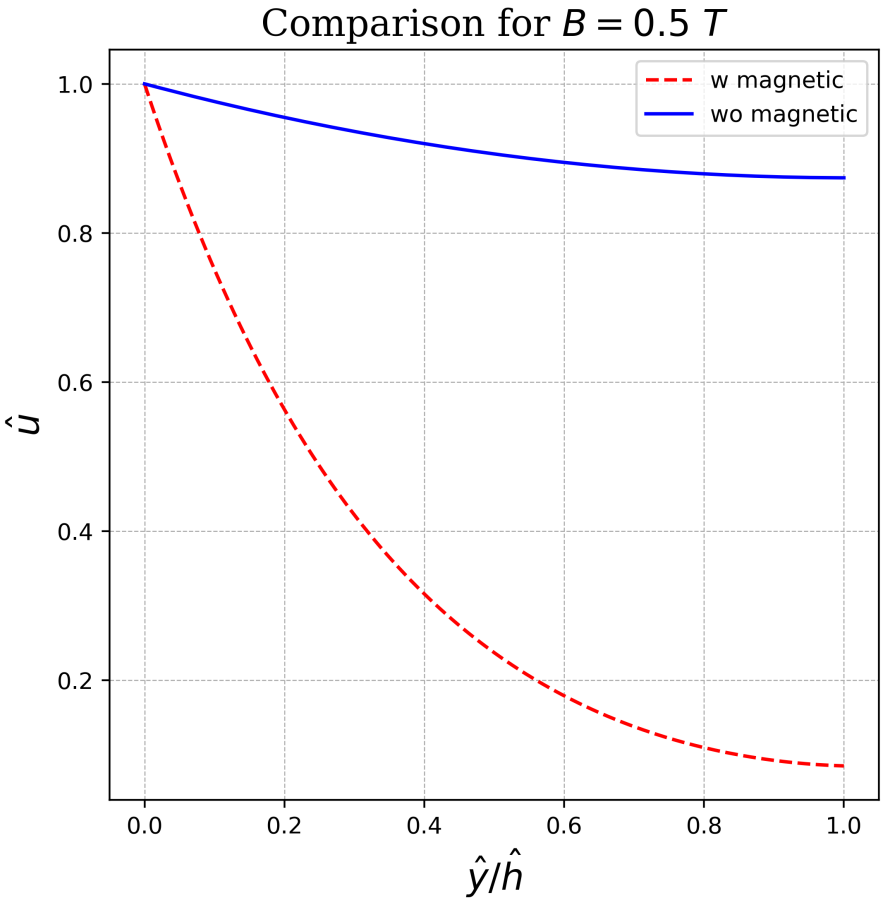


Figure 8.7: Comparison between the velocity profiles in presence (red) and in absence (blue) of the magnetic field

This difference is caused by the Lorenz force acting downwards over the liquid film. Indeed, this force assumes negative values in the zinc film as visible in figure (8.8).

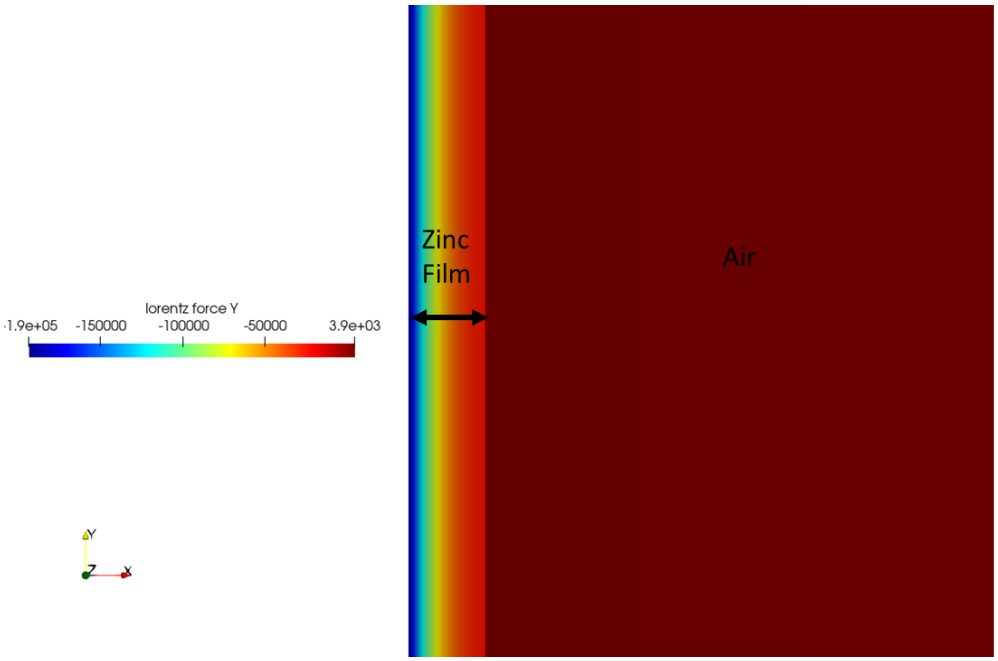


Figure 8.8: Lorentz Force acting on the liquid and air regions along the streamwise direction

This is explicable thanks to the relation between the Lorentz force $\mathbf{F}_L$ and the current density $\mathbf{J}$ as delineated in the equation (7.6). The induced current $\mathbf{J}$ comes from the Ohm's law (2.10) and determines an induced magnetic field, as shown in Ampere's law

$$\mathbf{J} = \nabla \times \mathbf{H} = \begin{pmatrix} \cancel{H^i_{z,y}} - \cancel{H^i_{y,z}} \\ \cancel{H^i_{x,z}} - \cancel{H^i_{z,x}} \\ H^i_{y,x} - H^i_{x,y} \end{pmatrix} = \frac{1}{\mu} \begin{pmatrix} \cancel{B^i_{z,y}} - \cancel{B^i_{y,z}} \\ \cancel{B^i_{x,z}} - \cancel{B^i_{z,x}} \\ B^i_{y,x} - B^i_{x,y} \end{pmatrix} \quad Ampere's \quad law \tag{8.8}$$

Where $\mathbf{H}^i = \mathbf{B}^i/\mu$, as explained in chapter 7. One can note that there is not a magnetic field component along z direction, since the problem is 2D. Besides, The sign of J_z is negative as evident in figure (8.9)

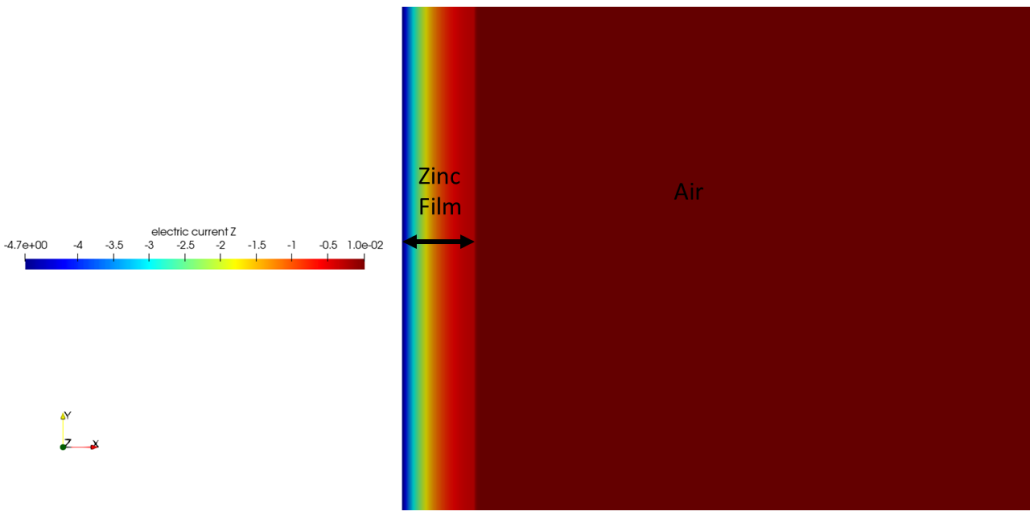


Figure 8.9: Induced current originated in the domain

.

Inserting the equation (8.8) in (7.6) yields

$$\mathbf{F}_L = \mathbf{J} \times (\mathbf{B}^i + \mathbf{B}^e) = \begin{pmatrix} -J_z(B_y^i + B_0) \\ +J_z B_x^i \\ 0 \end{pmatrix} = \frac{1}{\mu} \begin{pmatrix} -(B_{y,x}^i - B_{x,y}^i)(B_y^i + B_0) \\ +(B_{y,x}^i - B_{x,y}^i) B_x^i \\ 0 \end{pmatrix} \tag{8.9}$$

Hence, the Lorenz force has two components from (8.9), that one along x is represented in figure (8.10). Moreover, the superposition of the two components $\mathbf{B}^i$ and $\mathbf{B}^e$ is permitted, since the problem is linear.

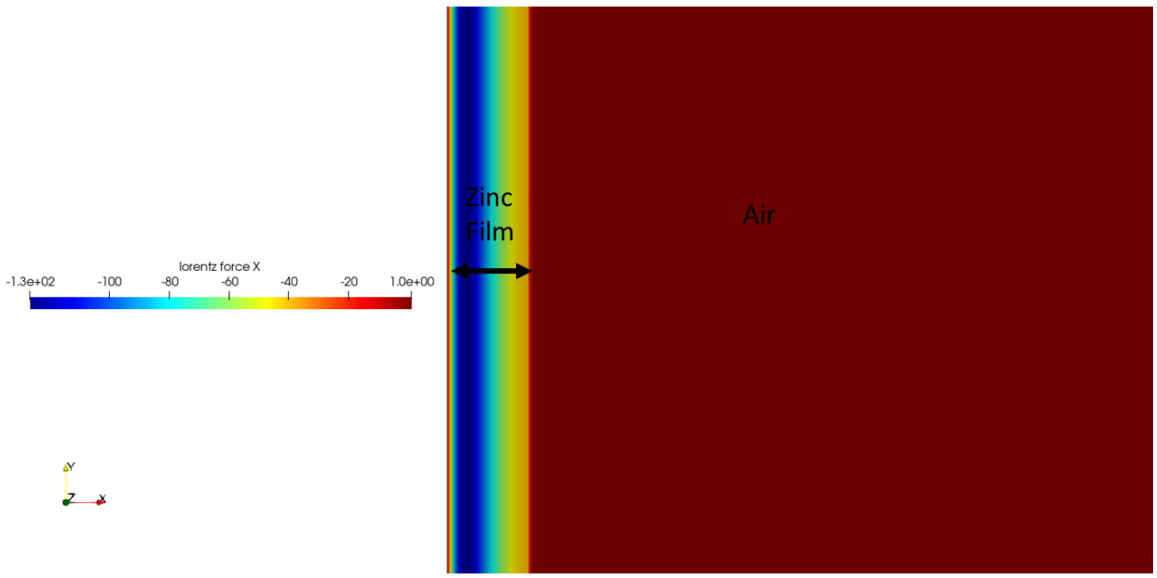
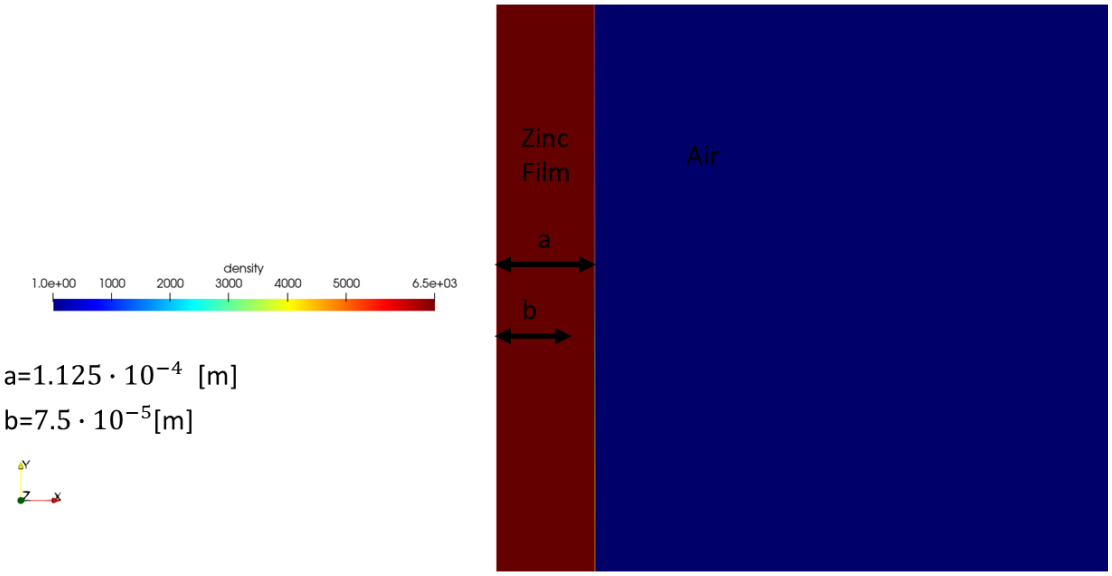
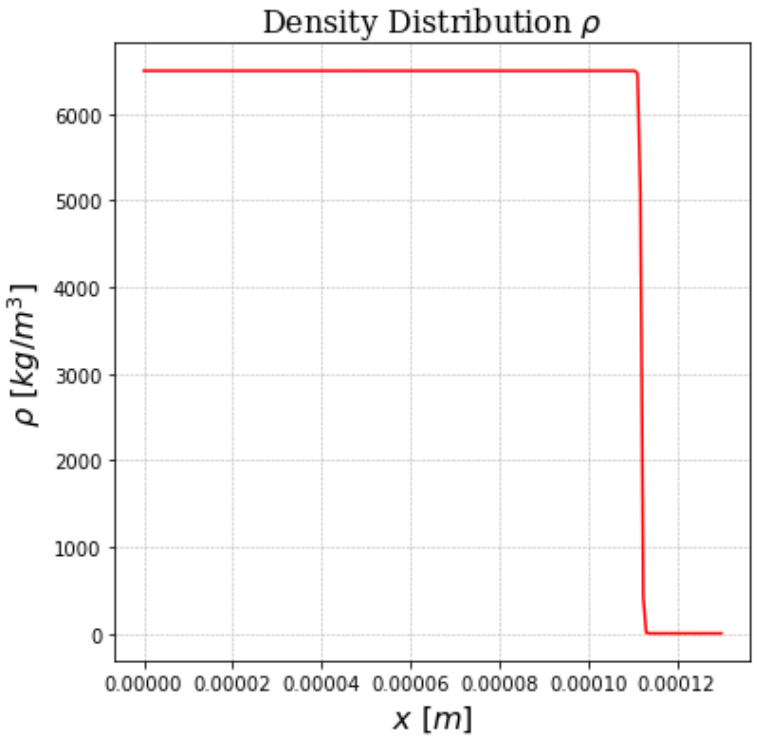


Figure 8.10: Lorentz Force acting on the liquid and air regions along wall-normal direction

As already said before, a smooth change for the fluid properties, like the density in Elmer, is needed since it is impossible to have a steep variation of this characteristic between the inside and the outside of the liquid film. Thereby, a hyperbolic tangent function (5.9) has been used for the evolution of the density and dynamic viscosity over the interface, as shown in figure (8.11)



(a)



(b)

Figure 8.11: Figure (a) the liquid and gas distribution in terms of density, where zinc film has a density ρ_l and air has a density ρ_g; figure (b) shows the density function described over the interface. The variables are in their dimensional form; ρ refers to density function $\rho_{l,g}$

Chapter 9

Steady Liquid Film Model in the Presence of a Magnetic Field as Initial Condition

In the previous chapter the velocity profile was expected a priori; the next step, discussed hereby, is to confirm if the velocity profile (6.8), provided to Elmer as initial condition in the presence of an external magnetic field, assumes the same shape computed in [19] and shown in the chapter 8.

9.1 Domain

The domain remains the same one of the previous chapter in terms of dimensions, as shown in figure (9.1)

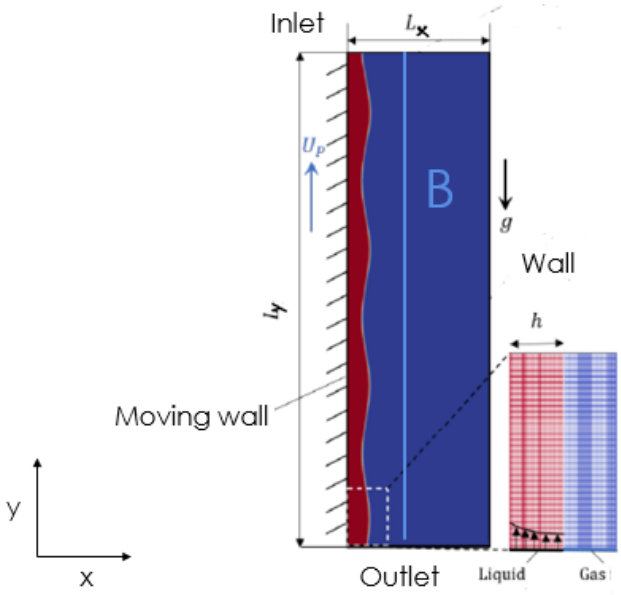


Figure 9.1: Schematic of the flow configuration used for validation purposes in Elmer: flow domain and zoom on the near-wall mesh in. B indicates the uniform magnetic field [16]

Where

$$L_x = 0.001\ m \quad L_y = 0.02\ m$$

9.2 Meshing

The mesh features explained in the previous chapter, have been kept for this case as well. Thereby, the interface length in Gmsh is a bit larger than the real magnitude h to chill the capture of the interface for the level set.

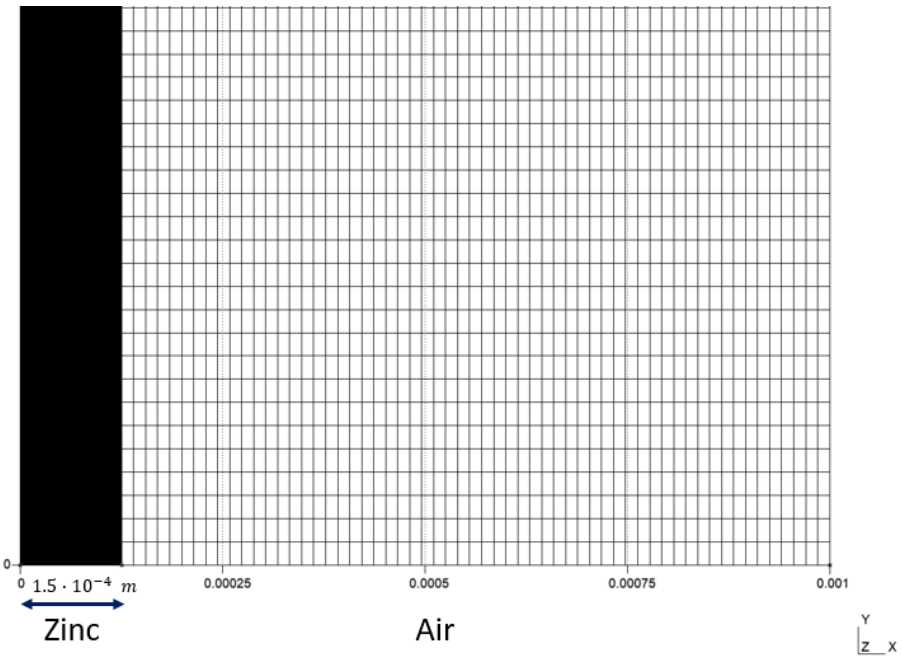


Figure 9.2: Position of the interface for the meshing in Gmsh

Moreover, the number of elements of the mesh is again the same as in chapter 8, as depicted in figure (9.3)

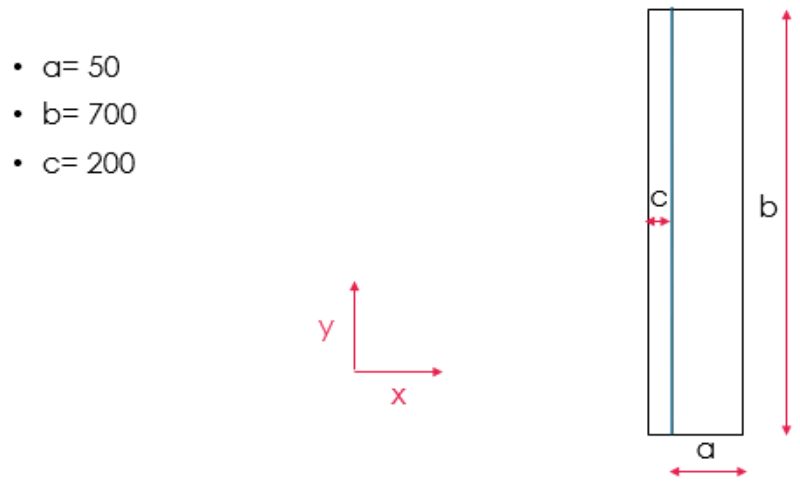


Figure 9.3: Number of the elements in the dimensions of the domain

9.3 Test Case Description

Hereunder, a concise delineation of the case set-up has been reported. The full input file is in the appendix 11

9.3.1 Simulation Set-Up

The same tricks for the evolution in time of the case without the magnetic field have been adopted in this example. Indeed, an adaptive time step has been set with a starting value of $0.01\ s$; then it changes by keeping the Courant number fixed at a certain value until the time step reaches a maximum of 10000 iterations. Thus, the total computational time is

$$T = 0.01 \times 10000 = 100\ [s]$$

The CFL condition is again $CFL = 0.5$, like $U_p = 0.5\ m/s$ and $\Delta x = 7.4 \cdot 10^{-7}\ m$ as well. The advection velocity U_p of the CFL condition corresponds to the strip velocity.

9.3.2 Body Forces

According to [16] the body force is gravitational and its value and direction can be specified in the section "Body Force" of the decision tree diagram in Elmer. As one can see in figure (9.1), the force direction is along y downwards. In the same section, Lorentz force must be specified since it appears as volume force in the Navier-Stokes momentum conservation equation.

9.3.3 Initial Condition

The equation describing the free surface for the level-set has been taken from [7]. Before appealing to [7], the same uniform interface of the previous chapter had been adopted. Nevertheless, by keeping the default features of the solvers, divergence problems had occurred. To overcome this complication, two possible solutions could be adopted.

- Picard iteration instead of Newton
- Reduction of the relaxation factor (< 0.1) both for Picard and Newton method

Nevertheless, nothing of these tricks had worked and they had caused an excessive increase in computational time either in the case with $B = 1\ T$ and with $B = 0.5\ T$. Hence, the optimal solution, proposed by [7], consists of providing an already deformed shape that is triangular, as shown in figure (9.4).

The adopted configuration is the one in figure (9.4a); indeed, the other one in figure (9.4b) has determined a diverged solution. Concerning the level-set, the implicit function ϕ describing the free surface is

$$-x + \left(\frac{1}{800}\right)(0.1 - y) = 0 \tag{9.1}$$

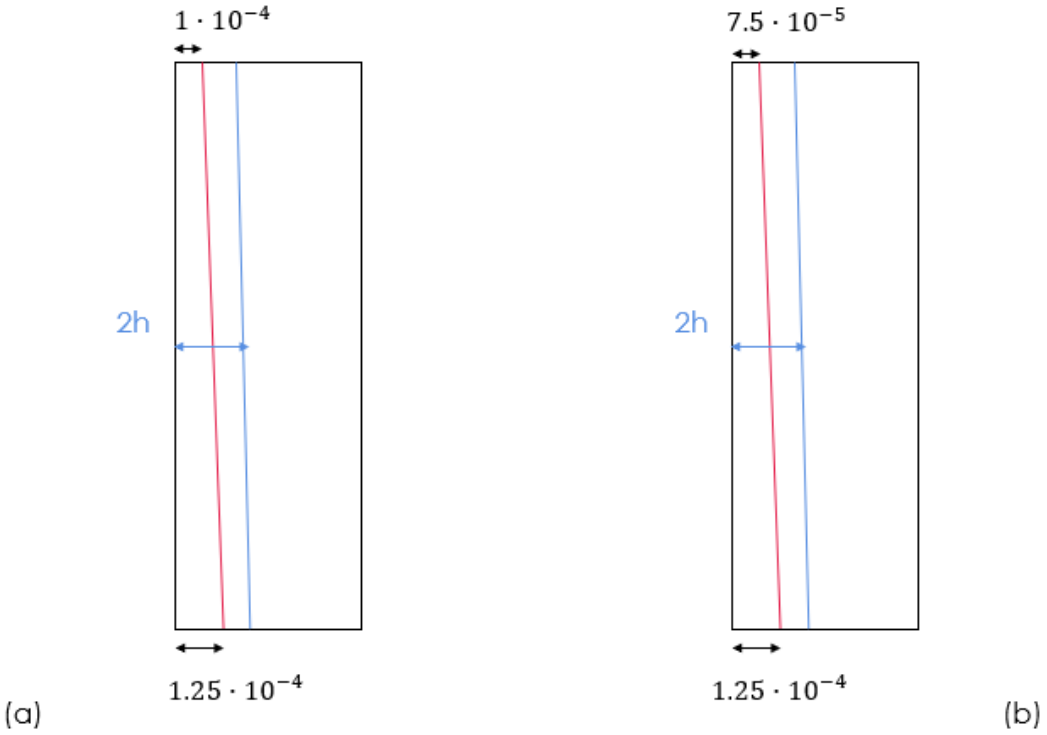


Figure 9.4: Figures (a) and (b) display two adopted triangular shapes for the interface. The blue line indicates the position $(2h)$ of the interface in Gmsh

.

In Elmer, the equation (9.1) refers to a surface variable called tx, where $tx(0)$ and $tx(1)$ correspond respectively to x and y coordinates. Concerning the advection velocity for the level-set, it coincides with the initial velocity of the Navier-Stokes equations. In this case, the velocity is zero along the normal wall direction (x), while the cross-stream velocity is the same one of the case without a magnetic field expressed in equation (6.8)

$$v(x) = \frac{1}{2}\frac{g}{\nu_l}x^2 - \frac{g}{\nu_l}hx + U_p$$

Where h is the film thickness and can be determined by fixing its corresponding dimensionless quantity $\hat{h}$

$$h = \hat{h}[h] = \hat{h}\sqrt{\frac{\nu_l U_p}{g}} = 7.51 \cdot 10^{-5}\ m$$

$$\hat{h} = 0.5 \quad U_p = 0.5\ m/s \quad g = 9.81\ m/s^2 \quad \nu_l = 4.461 \cdot 10^{-7}\ m^2/s$$

9.3.4 Material

In addition to what has already been defined in 6.3.4, the presence of an applied magnetic field has been added in the "Material" section in Elmer by calling it B_0 in wall-normal direction. Elmer computes the induced magnetic field in the material caused by the external source according to the equation (7.7)

[7] suggests using an asymptotic law for the level-set functions describing the evolution of the

dynamic viscosity μ_l and the mowing wall velocity U_p. Nevertheless, the application of this trick has not produced any improvement in terms of convergence; thus, the original functions have been kept. The physical properties of zinc and air have already been listed in table (8.1) and have been kept for this simulation as well.

9.3.5 Solvers

Five solvers have been used.

- 2 for level-set equation: computation of the interface and its reinitialization
- 1 for Navier-Stokes equations
- 1 for magnetic induction equation
- 1 for post-processing

The full equations system, solved by Elmer, is presented, as follows

$$\begin{cases} \nabla \cdot \mathbf{U} = 0 \quad Mass \quad conservation \quad equation \\ \rho_{l,g}\left(\frac{\partial \mathbf{U}}{\partial t} + (\mathbf{U}\cdot\nabla)\mathbf{U}\right) - \nabla(2\mu_{l,g}\mathbf{E}_{l,g}) + \nabla p_{l,g} = \rho_{l,g}\mathbf{f}_{ext} \quad Momentum \quad equation \\ \frac{\partial \phi}{\partial t} + \mathbf{U}\cdot\nabla\phi = 0 \quad Level-set \quad equation \\ \frac{\partial \mathbf{B}}{\partial t} - \frac{1}{\sigma_m \mu}\nabla\times\nabla\times\mathbf{B} - \nabla\times(\mathbf{U}\times\mathbf{B}) = 0 \quad Magnetic \quad induction \quad equation \end{cases}$$

As already declared before in 6.3.5, the umfpack has been used as a direct method for linear system resolution, due to the linearity of the equations visible in the system above. Figure (9.5) illustrates the steps followed by Elmer for the computation of the solution

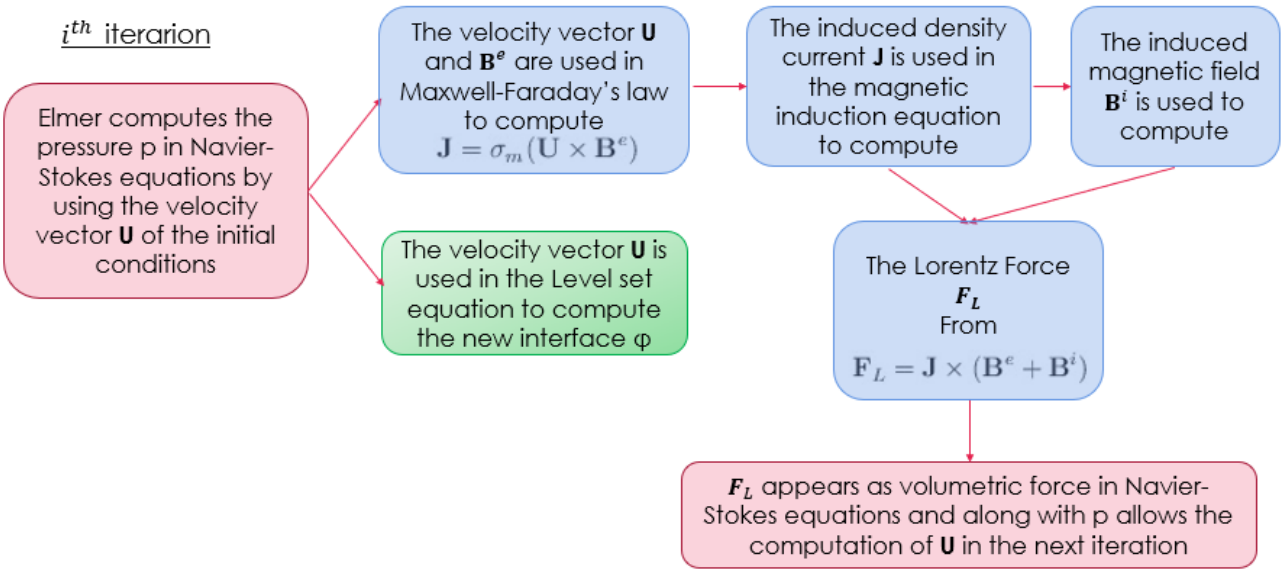


Figure 9.5: Order of the resolution of the equation in Elmer

The Newton iteration method for the non-linear convection term in Navier-Stokes equations has been held; nevertheless, the relaxation factor has been decreased further.

$$\lambda = 0.01$$

Indeed, a higher relaxation factor had caused the interruption of the simulation. The setting of the level set reinitialization has been kept as in the case without the magnetic field.

9.3.6 Boundary Conditions

The interested domain as depicted in figure (9.1) has four boundaries according to the numerical case in [16].

Wall

Homogeneous Dirichlet boundary conditions have been applied for the two velocity components. It means to write in Elmer

$$u = 0$$
$$v = 0$$

Conversely, nothing has been specified for the induction equation; it means a homogeneous Neumann condition for the induced magnetic field component $\mathbf{B}^i$

$$\nabla \mathbf{B}^i \cdot \hat{n} = 0$$

Moving Wall

The wall velocity in the streamwise direction has been defined.

$$v = U_p$$

Whereas for the wall-normal component, a homogeneous Dirichlet condition has been applied.

$$u = 0$$

Concerning the induction equation, a homogeneous Dirichlet condition referred to the induced magnetic field component $\mathbf{B}^i$ has been set.

$$B_x^i = 0$$
$$B_y^i = 0$$

The physical meaning behind these boundary conditions has already dealt with in 8.3.6

Inlet and Outlet

The wall-normal velocity has a value equal to zero.

$$u = 0$$

Conversely, for the streamwise velocity, no numerical quantity has been written; it involves a homogeneous Neumann boundary condition implicitly.

$$\frac{\partial v}{\partial x} \cdot \hat{n} = 0$$

The same condition has been associated with the induced magnetic field in the two directions.

$$\nabla \mathbf{B}^i \cdot \hat{n} = 0$$

Figure (9.6) recaps the initial and boundary conditions used for this simulation.

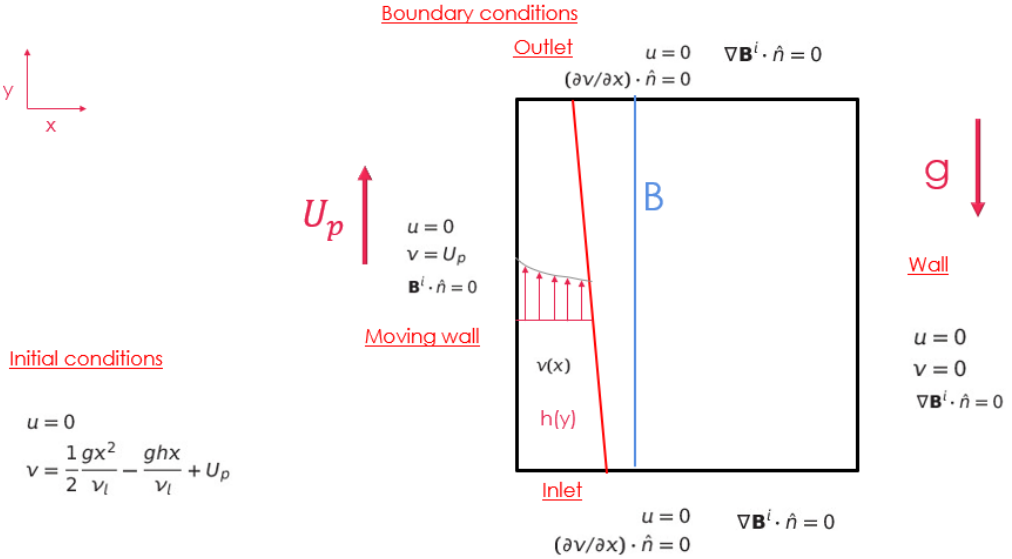


Figure 9.6: Full set of initial and boundary conditions used for the test case

9.4 Post-processing

The velocity of the moving wall is $U_p = 0.5\ m/s$ like in the previous chapters. The applied external magnetic field in this simulation has two possible values.

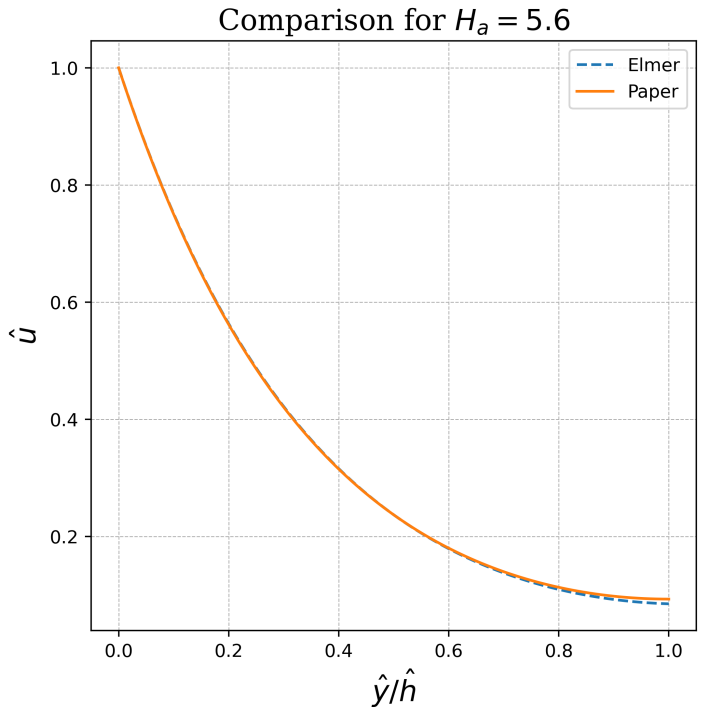
$$\mathbf{B}^e = (B_0, 0) = 0.5\ T \quad and \quad \mathbf{B}^e = (B_0, 0) = 1\ T$$

The listed parameters in table (8.1) allow computing the corresponding Hartmann numbers

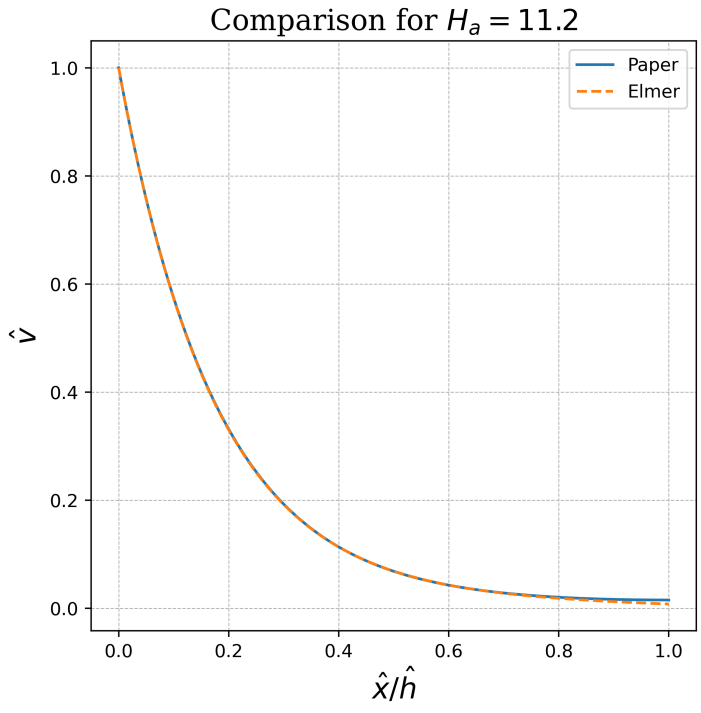
$$B_0 = [B] = 0.5 \rightarrow H_a = \frac{[B]}{\sqrt{\frac{\sigma_m U_p}{g \rho_l}}} = 5.6$$

$$B_0 = [B] = 1 \rightarrow H_a = \frac{[B]}{\sqrt{\frac{\sigma_m U_p}{g \rho_l}}} = 11.2$$

The comparison between the expected results of cross-velocity profile computed according to the equation (8.5), from [19], and those got by Elmer have been depicted in figure (9.7).



(a)



(b)

Figure 9.7: Comparison between the cross velocity profile computed in Elmer and that one obtained from equation (8.5). In the legend Paper indicates the report [19]. The variables are in their dimensionless form

The computation of equation (8.5) has been realized in Matlab (appendix 11). Data have been collected in the liquid film at position $y = 0.01\ m$ of the interested domain. One can note the reduction of the velocity profile compared to that one in chapter 6 without a magnetic field, as shown in figure (9.8)

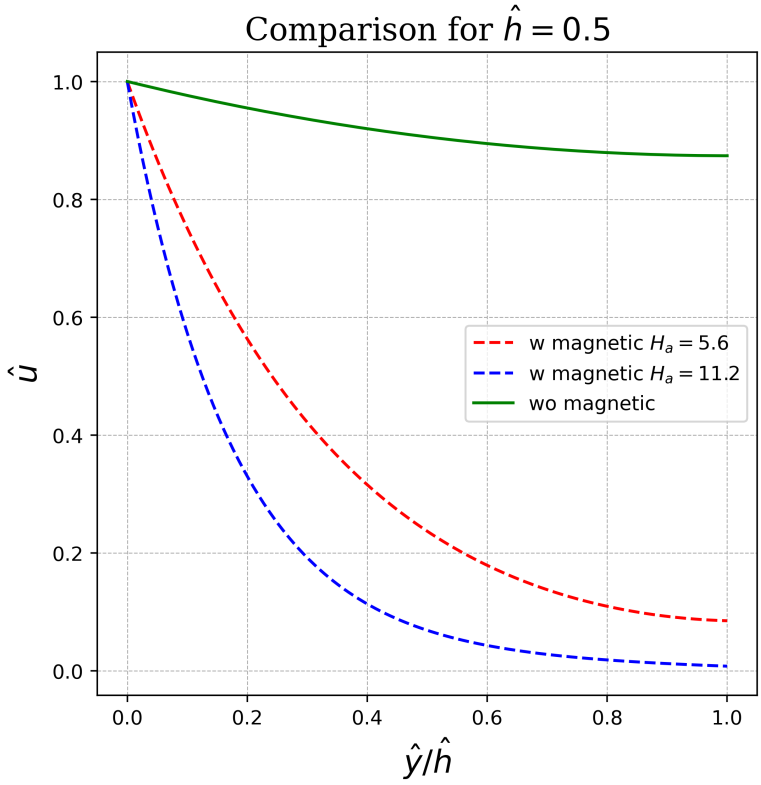
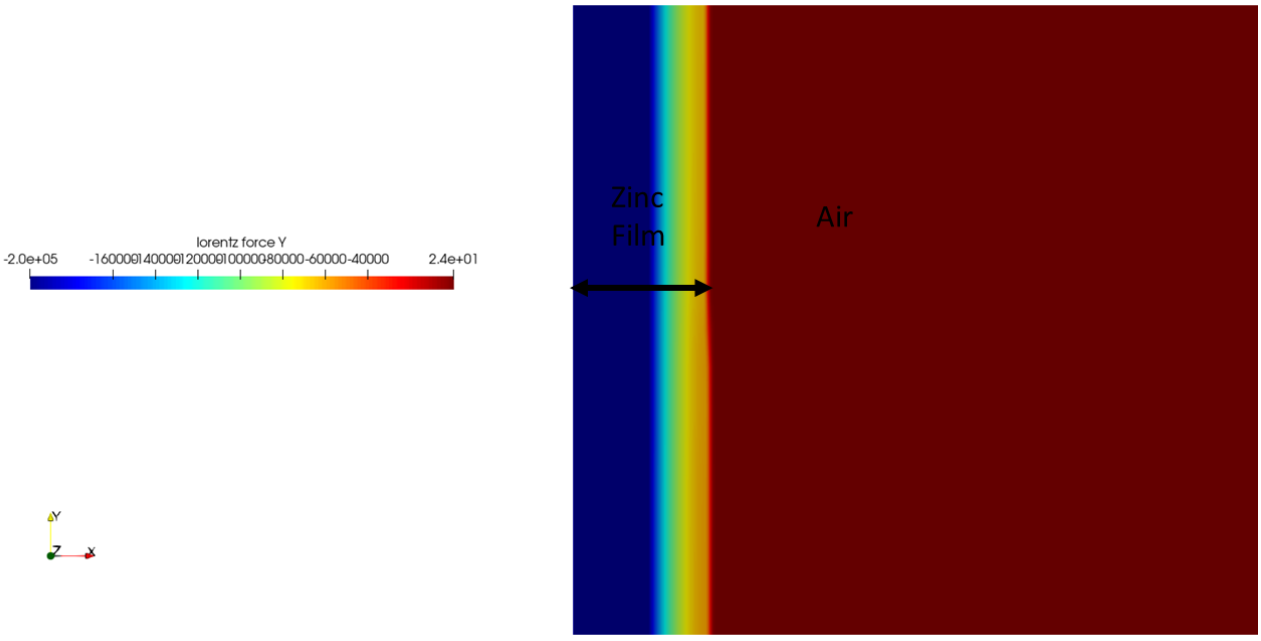
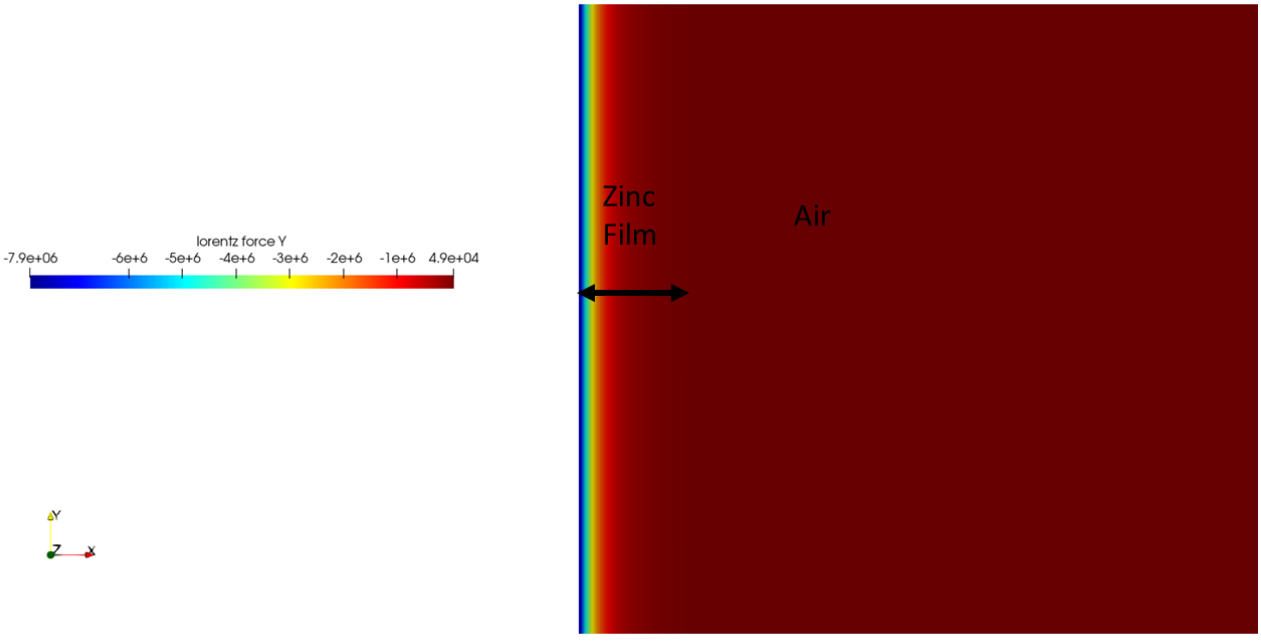


Figure 9.8: Comparison between the velocity profiles in presence (green) and in absence (blue and red) of the magnetic field. The variables are in their dimensionless form

This difference is caused by the Lorenz force acting downwards on the liquid film. Indeed, this force assumes negative values in the liquid film as evident in figure (9.9). This is explicable thanks to the relation between the Lorentz force $\mathbf{F}_L$ and the current density $\mathbf{J}$ as delineated in the equation (8.9).



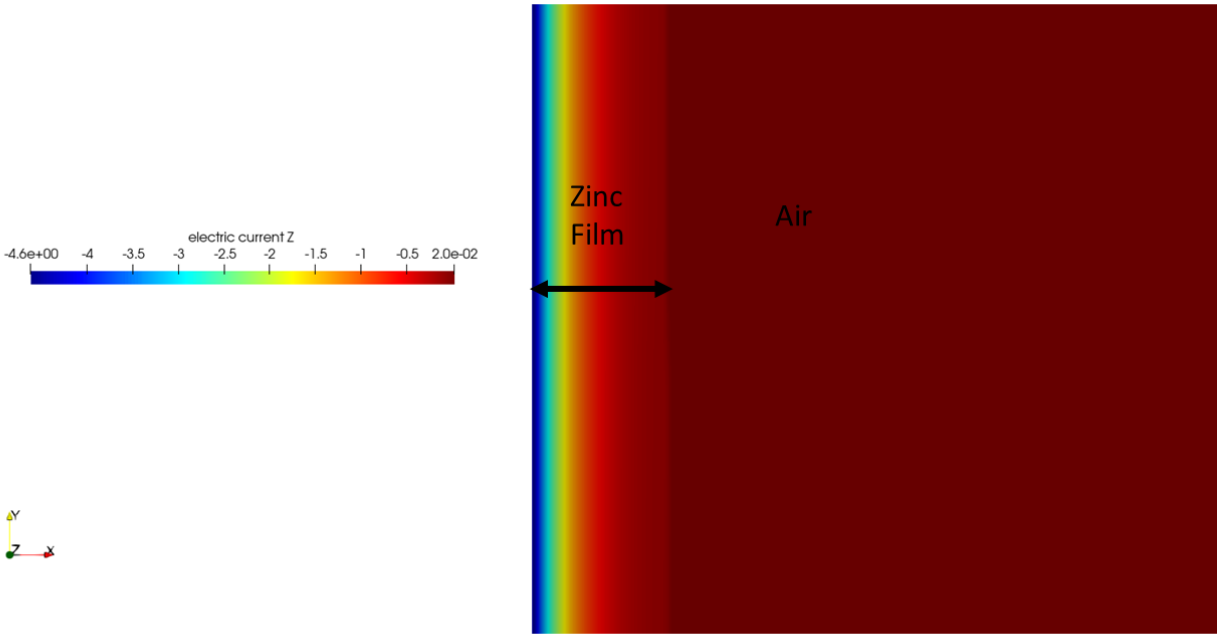
(a)



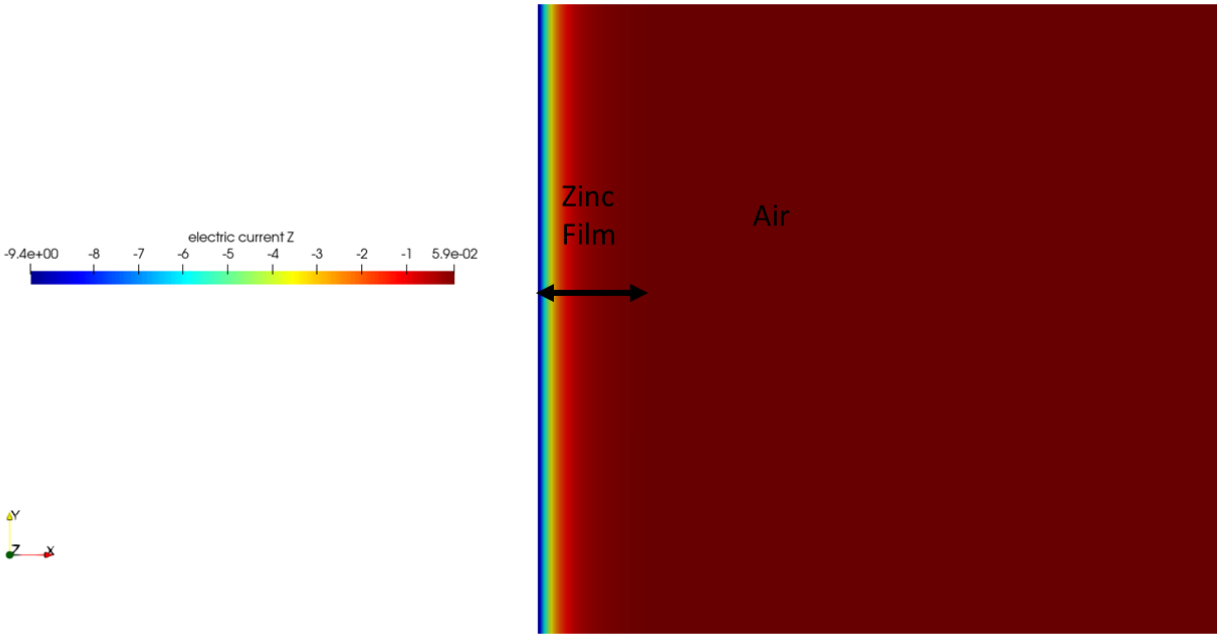
(b)

Figure 9.9: Lorentz Force acting on the liquid and air regions along the streamwise direction. Figure (a) refers to $\mathbf{B} = 0.5\ T$ and figure (b) refers to $\mathbf{B} = 1\ T$

The induced current $\mathbf{J}$ comes from Ohms's law (7.11) which determines an induced magnetic field as computed in (8.8). This last equation shows the presence of just one component $J_z = H^i_{y,x} - H^i_{x,y}$, as visible in figure (9.10).



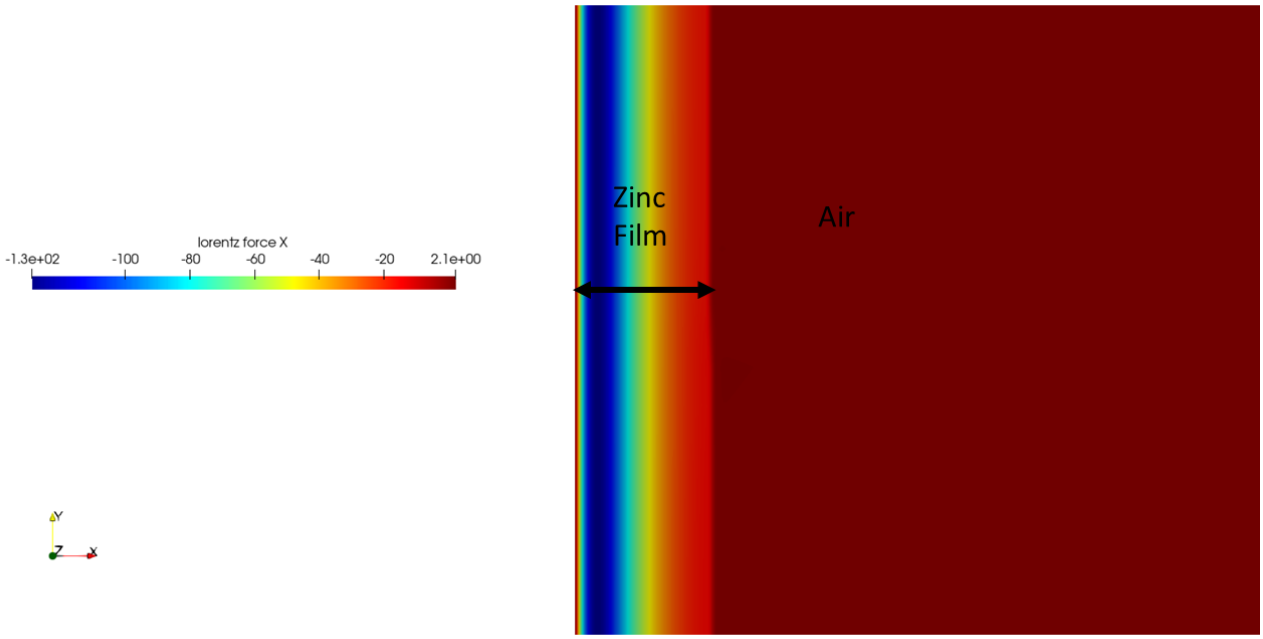
(a)



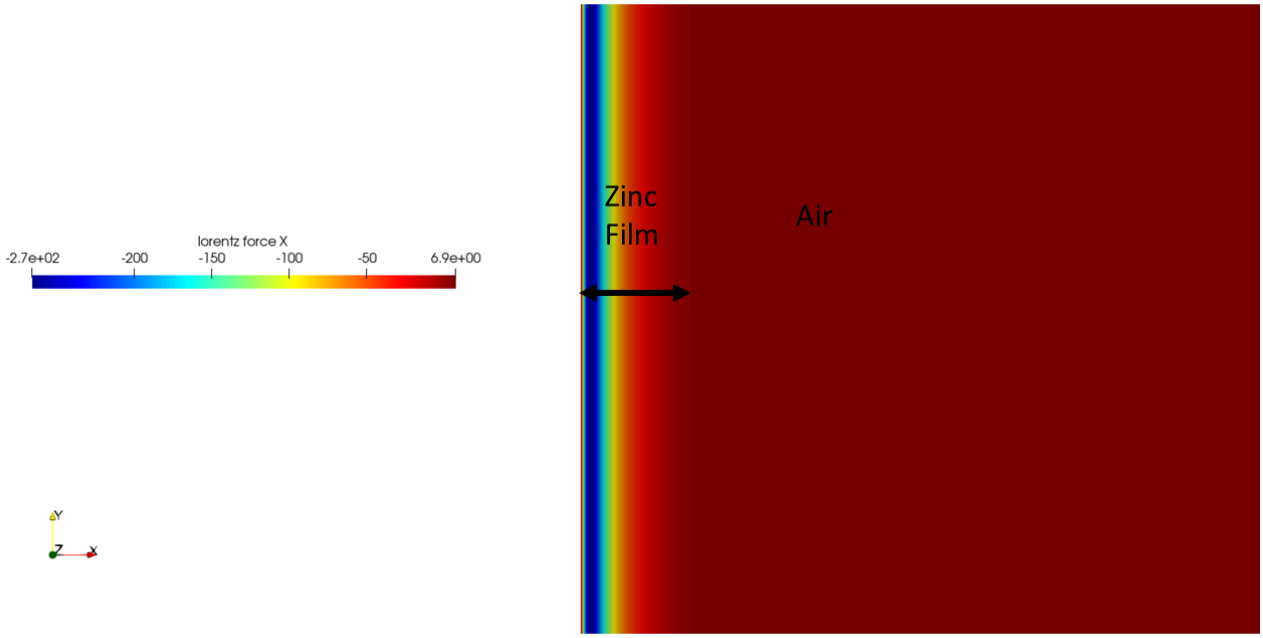
(b)

Figure 9.10: Induced current originated in the domain. Figure (a) refers to $\mathbf{B} = 0.5\ T$ and figure (b) refers to $\mathbf{B} = 1\ T$

Equation (8.9) demonstrates the presence of two components of the Lorenz force. The contribution along x is represented in figure (9.11)



(a)



(b)

Figure 9.11: Lorentz Force acting on the liquid and air regions along the wall-normal direction. Figure (a) refers to $\mathbf{B} = 0.5\ T$ and figure (b) refers to $\mathbf{B} = 1\ T$

Chapter 10

Steady Liquid Film Model: Expected Velocity Profile in the Presence of a Gaussian Magnetic Field as Initial Condition

The final step needed to confirm the electromagnetic model [19] is to simulate a Gaussian magnetic field acting on the liquid film. In this case, the velocity profile, provided as initial condition, is the one computed in [19].

10.1 Domain

The geometry is the same as in the previous cases, nevertheless, the magnetic field changes in the streamwise direction, as depicted in figure (10.1)

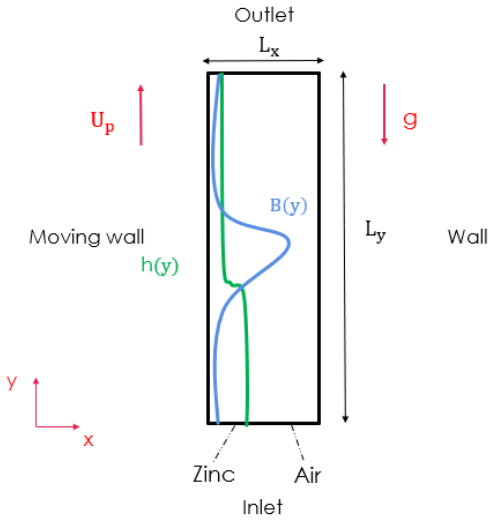


Figure 10.1: Schematic of the flow configuration used for validation purposes in Elmer. B indicates the Gaussian magnetic field and $h(y)$ indicates the zinc free surface

Where

$$L_x = 0.001\ m \quad L_y = 0.02\ m$$

10.2 Meshing

The mesh features explained in the previous chapters have been kept for this case as well. Thereby, the interface length in Gmsh is a bit larger than the real magnitude $h(y)$ to chill the capture of the interface for the level set. In this case, the interface position in Gmsh is at a distance of

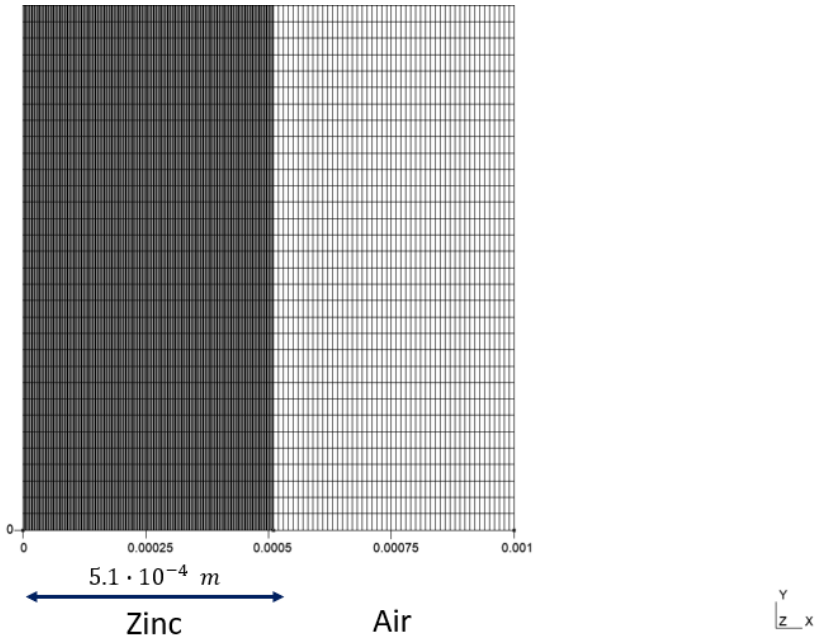


Figure 10.2: Position of the interface for the meshing in Gmsh

$1.5 \max(h(y))$. Moreover, the number of elements of the mesh has been incremented since the interface is at a major distance compared to the geometry in chapters 6 and 8, as depicted in figure (10.3)

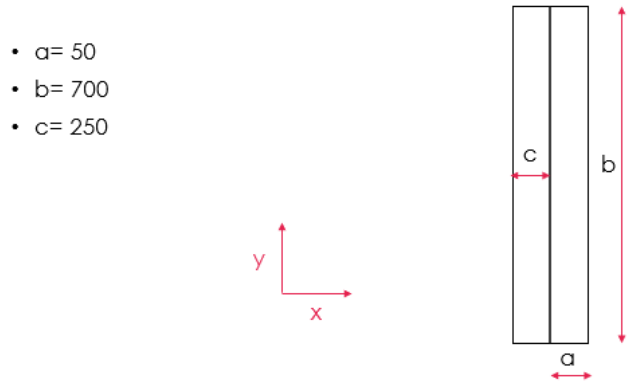


Figure 10.3: Number of the elements in the domain dimensions

10.3 Test Case Description

Hereunder, a concise delineation of the case set-up has been reported. The full input file is in the appendix 11

10.3.1 Simulation Set-Up

As in the previous cases, an adaptive time step has been set. The starting value of the time step is $0.01\ s$; then it changes by keeping the Courant number fixed at a certain value until the time step reaches a maximum of 10000 iterations. Thus, the total computational time is

$$T = 0.01 \times 10000 = 100\ [s]$$

The CFL condition is again $CFL = 0.5$, like $U_p = 0.5\ m/s$ and $\Delta x = 7.4 \cdot 10^{-7}\ m$ as well. The advection velocity U_p of the CFL condition corresponds to the strip velocity.

10.3.2 Body Forces

According to [16] the body, force is gravitational, and its value and direction can be specified in Elmer's section "Body Force" of the decision tree diagram. As one can see in figure (10.1), the force direction is along y downwards. In the same section, Lorentz force must be specified since it appears as volume force in the Navier-Stokes momentum conservation equation.

10.3.3 Initial Condition

In chapter 9 the equation describing the free surface for the level-set had been taken from [7]. In this case, the provided interface is already deformed according to the warping model computed numerically [19]. Nevertheless, this interface had generated some numerical issues being an excessively steep model for Elmer. Thereby, by using the same data of the interface, a hyperbolic tangent function $(h(y))$ has been chosen to interpolate the free surface points, as shown in figure (10.4).

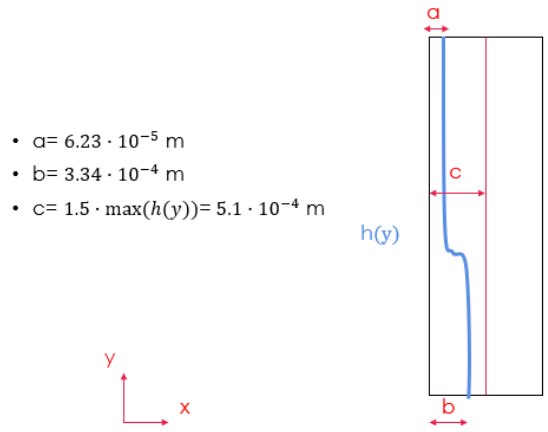


Figure 10.4: Free surface function

The function has been written below.

$$h = \frac{1}{2}\left(3.34 \cdot 10^{-4} - 6.23 \cdot 10^{-5}\right)\left(1 + \tanh\left(\frac{-y + 0.01}{E}\right)\right) + 6.23 \cdot 10^{-5} \tag{10.1}$$

Where (10.1) takes the extremes of the interface and the smooth parameter E assumes a value that ensures a smooth trend, as depicted in figure (10.5)

$$E = 2.5 \cdot 10^{-3}$$

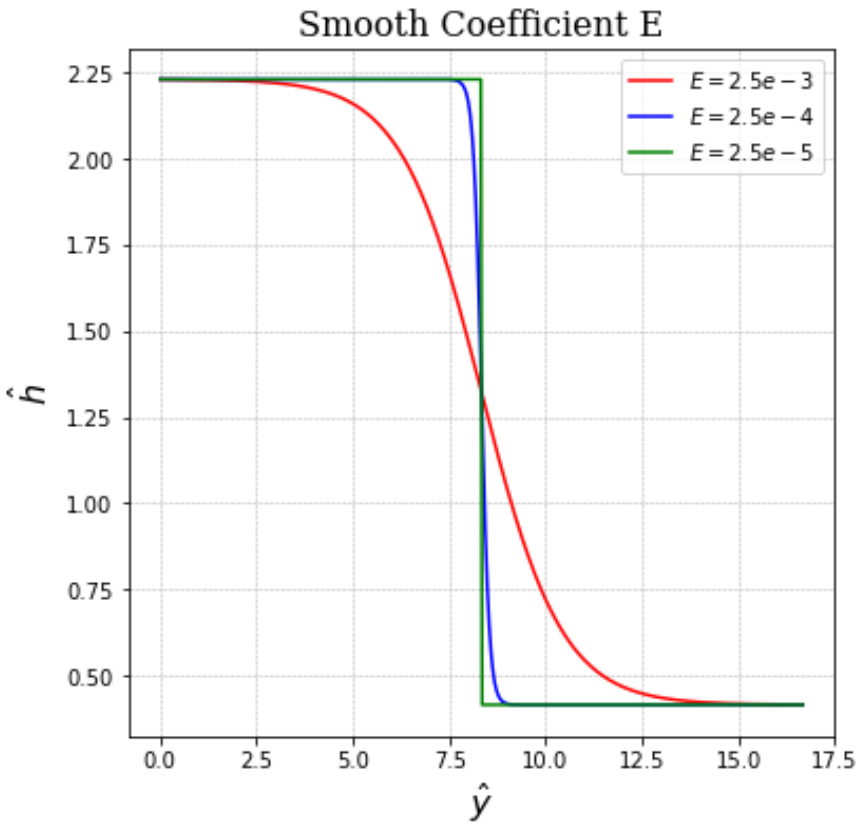


Figure 10.5: Influence of the parameter E on the interface evolution. The variables are in their dimensionless form

.

In Elmer, the equation (10.1) refers to a surface variable called tx, where $tx(0)$ and $tx(1)$ correspond respectively to x and y coordinates. Concerning the advection velocity for the level-set coincides with the Navier-Stokes equations' initial velocity. In this case, the velocity is zero along the wall-normal direction (x), while the cross-stream velocity is the same one of the cases with a magnetic field expressed in the chapter 8

$$\begin{aligned} v(x) = -U_p \Bigg[&\tanh\left(\frac{h}{[h]} H_a \frac{B}{[B]}\right)\left(1 + \frac{1}{H_a^2 (B/[B])^2}\right)\sinh\left(\frac{x}{[x]} H_a \frac{B}{[B]}\right) + \\ &- \cosh\left(\frac{x}{[x]} H_a \frac{B}{[B]}\right)\left(1 + \frac{1}{H_a^2 (B/[B])^2}\right) + \frac{1}{H_a^2 (B/[B])^2}\Bigg] \end{aligned} \tag{10.2}$$

Where h is computed in (10.1) and B is the Gaussian function $B(x)$ which has been specified in the "Material" section. Furthermore, the table (10.1) recaps the reference values used in Elmer for the equation (10.3)

$Elmer$	$Variable$	$Value$
Up	U_p	$0.5\ m/s$
$h1$	$[h]$	$1.50 \cdot 10^{-4}\ m$
$y1$	$[y]$	$0.0012\ m$
B_max	$[B]$	$0.5\ T$
$h1$	$[x] = [h]$	$1.50 \cdot 10^{-4}\ m$
Ha	H_a	$5.6\ [nondim]$

Table 10.1: Names of the reference parameters employed in Elmer

U_p is the strip velocity and is the reference quantity for the velocity vector; $[h] = [x]$ is computed as follows

$$\hat{h} = \frac{h(y)}{[h]} = \frac{h(y)}{\sqrt{\frac{\nu_l U_p}{g}}} \tag{10.3}$$
$$U_p = 0.5\ m/s \quad g = 9.81\ m/s^2 \quad \nu_l = 4.461 \cdot 10^{-7}\ m^2/s$$

They are the reference quantities respectively for wall-normal coordinate x and film thickness h. Moreover, $[y]$ is the reference value for the streamwise coordinate

$$[y] = \frac{[h]}{\epsilon} \tag{10.4}$$

Where ϵ is the film parameter

$$\epsilon = (C_a)^{1/3} = \left(\frac{\mu_l U_p}{\sigma_t}\right)^{1/3} \tag{10.5}$$

Where C_a is the capillary number and σ_t is the surface tension. Table (10.2) resumes the values of the parameters of the equation (10.5)

$Variable$	$Value$
U_p	$0.5\ m/s$
μ_l	$0.0029\ Pa \cdot s$
σ	$0.8\ N/m$

Table 10.2: Adopted parameters for C_a computation

Eventually, after having imposed H_a and the strip velocity U_p, the computation of the reference quantity for the magnetic field is immediate thanks to Hartmann number definition

$$[B] = \frac{H_a}{\sqrt{\frac{\sigma_m U_p}{g \rho_l}}} \tag{10.6}$$

Table (10.3) recaps the reference quantities employed in the simulation

$Reference\ Quantity$	$Definition$	$Expression$
$[v]$	U_p	U_p
$[h]$	$(\mu_l[u]/(g\rho_l))^{1/2}$	$(\mu_l U_p/(g\rho_l))^{1/2}$
$[x]$	$[h]$	$(\mu_l U_p/(g\rho_l))^{1/2}$
$[y]$	$[h]/\epsilon$	$[(\mu_l U_p/(g\rho_l))^{1/2}]/C_a^{1/3}$
$[B]$	$sup(B(y))$	$H_a/(\sigma_m U_p/(g\rho_l))^{1/2}$

Table 10.3: Reference quantities adopted in the model. C_a is the capillary number (10.5)

10.3.4 Material

The presence of an applied magnetic field has been added in the "Material" section in Elmer by calling it $B(y)$ in the wall-normal direction. In this case, as said above, the magnetic field evolves in streamwise direction y according to a Gaussian trend.

$$B(y) = [B]\left[(1 - 5 \cdot 10^{-3}) \exp\left(-\frac{(y - 0.01)^2}{2\gamma^2[y]^2}\right) + 5 \cdot 10^{-3}\right] \tag{10.7}$$

Where γ is the magnetic field standard deviation; γ values lower than 1 have a major wiping effect on the liquid film, nevertheless they have led to divergence problems, determining the stop of the simulation. Thereby, $\gamma = 3$ has been elected. $[B]$ and $[y]$ are the reference quantities for the magnetic field and for the streamwise coordinate according to table (10.3). Elmer computes the induced magnetic field in the material caused by the external source according to the equation (7.7). The physical properties of zinc and air have already been listed in table (8.1).

10.3.5 Solvers

Five solvers have been used.

- 2 for level-set equation: computation of the interface and its reinitialization
- 1 for Navier-Stokes equations
- 1 for magnetic induction equation
- 1 for post-processing

The full equations system, solved by Elmer, is presented, as follows

$$\begin{cases} \nabla \cdot \mathbf{U} = 0 \quad Mass \quad conservation \quad equation \\ \rho_{l,g}\left(\frac{\partial \mathbf{U}}{\partial t} + (\mathbf{U} \cdot \nabla)\mathbf{U}\right) - \nabla(2\mu_{l,g}\mathbf{E}_{l,g}) + \nabla p_{l,g} = \rho_{l,g}\mathbf{f}_{ext} \quad Momentum \quad equation \\ \frac{\partial \phi}{\partial t} + \mathbf{U} \cdot \nabla\phi = 0 \quad Level - set \quad equation \\ \frac{\partial \mathbf{B}}{\partial t} - \frac{1}{\sigma_m \mu}\nabla \times \nabla \times \mathbf{B} - \nabla \times (\mathbf{U} \times \mathbf{B}) = 0 \quad Magnetic \quad induction \quad equation \end{cases}$$

As already declared before in 6.3.5, the umfpack has been used as a direct method for linear system resolution; whereas, the Newton iteration method has been applied for the non-linear convection term in Navier-Stokes. The relaxation factor value of the previous simulation has been held

$$\lambda = 0.01$$

Indeed, a higher relaxation factor has caused the interruption of the simulation. The setting of the level set reinitialization has been the same case without the magnetic field. Figure (10.6) illustrates the steps followed by Elmer for the computation of the solution

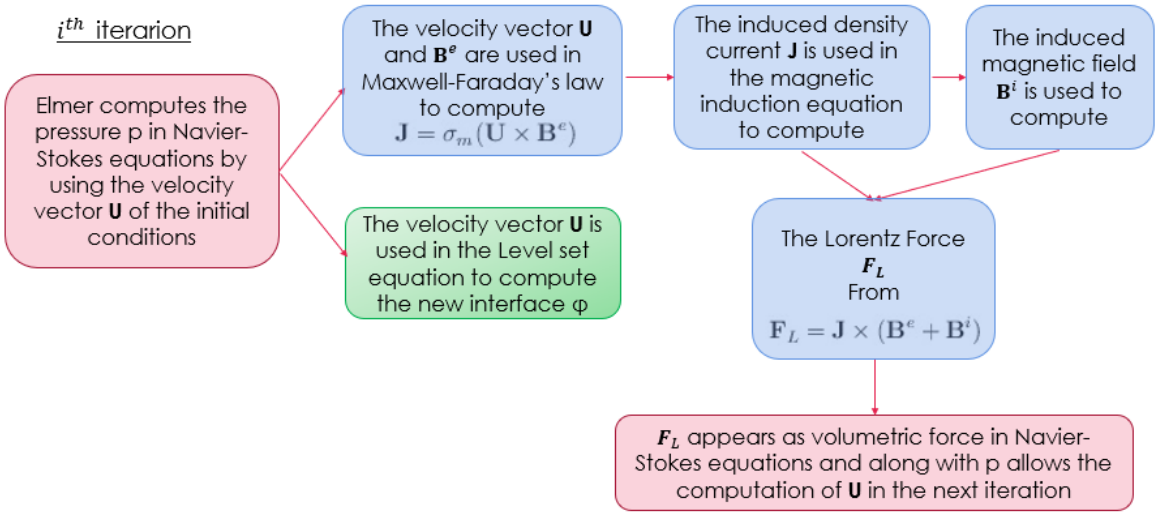


Figure 10.6: Order of the resolution of the equation in Elmer

10.3.6 Boundary Conditions

The interested domain as depicted in figure (10.1) has four boundaries according to the numerical case in [16].

Wall

Homogeneous Dirichlet boundary conditions have been applied for the two velocity components. It means to write in Elmer

$$u = 0$$

$$v = 0$$

Conversely, nothing has been specified for the induction equation; it means a homogeneous Neumann condition for the induced magnetic field component $\mathbf{B}^i$

$$\nabla \mathbf{B}^i \cdot \hat{n} = 0$$

Moving Wall

The wall velocity in the streamwise direction has been defined.

$$v = U_p$$

Whereas for the wall-normal component, a homogeneous Dirichlet condition has been applied.

$$u = 0$$

Concerning the induction equation, a homogeneous Dirichlet condition referred to the induced magnetic field component $\mathbf{B}^i$ has been set.

$$B_x^i = 0$$
$$B_y^i = 0$$

The physical meaning behind these boundary conditions has already dealt with in 8.3.6

Inlet and Outlet

The wall-normal velocity has a value equal to zero.

$$u = 0$$

Conversely, for the streamwise velocity, no numerical quantity has been written; it involves a homogeneous Neumann boundary condition implicitly.

$$\frac{\partial v}{\partial x} \cdot \hat{n} = 0$$

The same condition has been associated with the induced magnetic field in the two directions.

$$\nabla \mathbf{B}^i \cdot \hat{n} = 0$$

Figure (10.7) sums up the initial and boundary conditions used for this simulation.

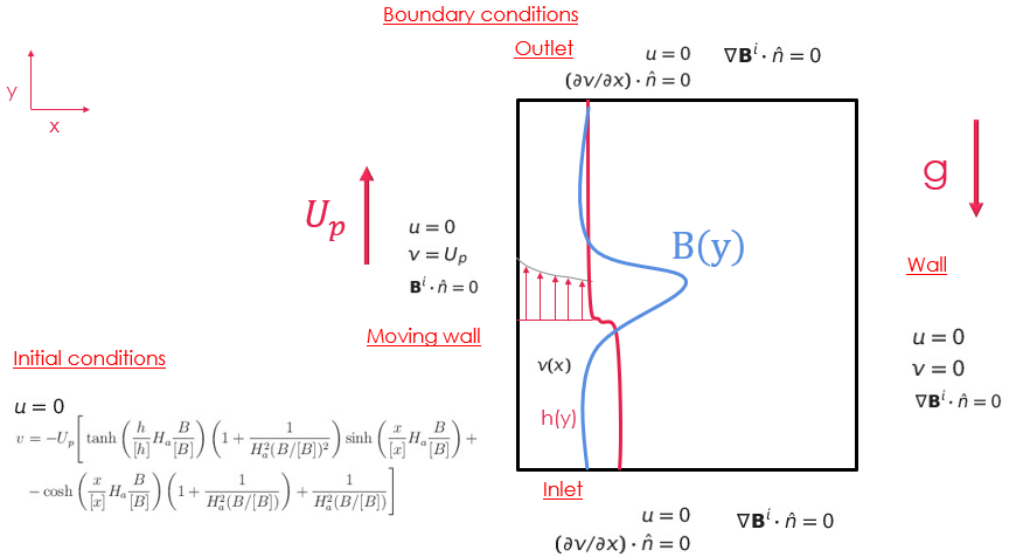


Figure 10.7: Full set of initial and boundary conditions used for the test case

10.4 Post-Processing

The simulation has been run by using the parameters illustrated in table (10.1) keeping the velocity of the strip constant $U_p = 0.5\ m/s$, in particular, two values of Hartmann number have been tested

$$H_a = 3 \rightarrow [B] = 0.26\ T$$
$$H_a = 5.6 \rightarrow [B] = 0.5\ T$$

Once the Hartmann number and the physical properties have been fixed, Elmer can compute the velocity profiles described in equation (10.2). Figure (10.9) shows the comparison between the predicted solutions from the electromagnetic model [19] provided to Matlab (11) and the results computed in Elmer. The velocity profiles have been selected in three different regions of the domain, as shown in figure (10.8)

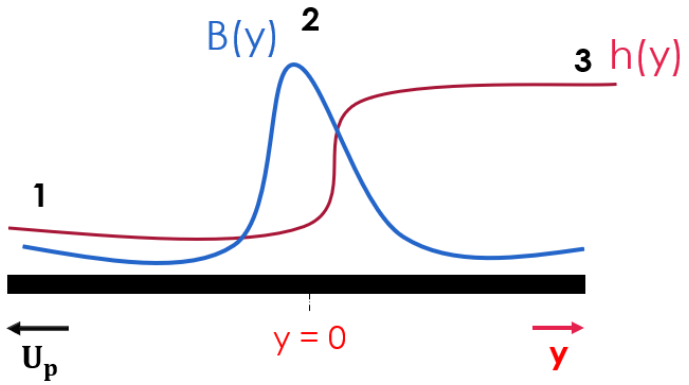
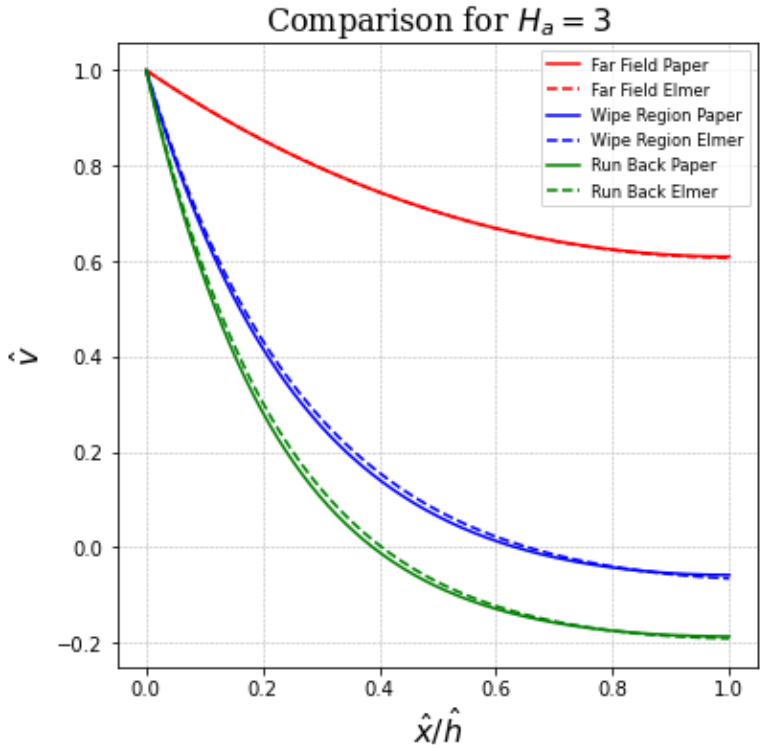


Figure 10.8: Sketch of the jet wiping regions: a liquid film with an interface $h(y)$ is dragged on a plate moving at a speed U_p. This film enters in a zone where a magnetic field $B(y)$ is applied. 1 is the far field region, 2 is the jet wiping region and 3 is the run back region.

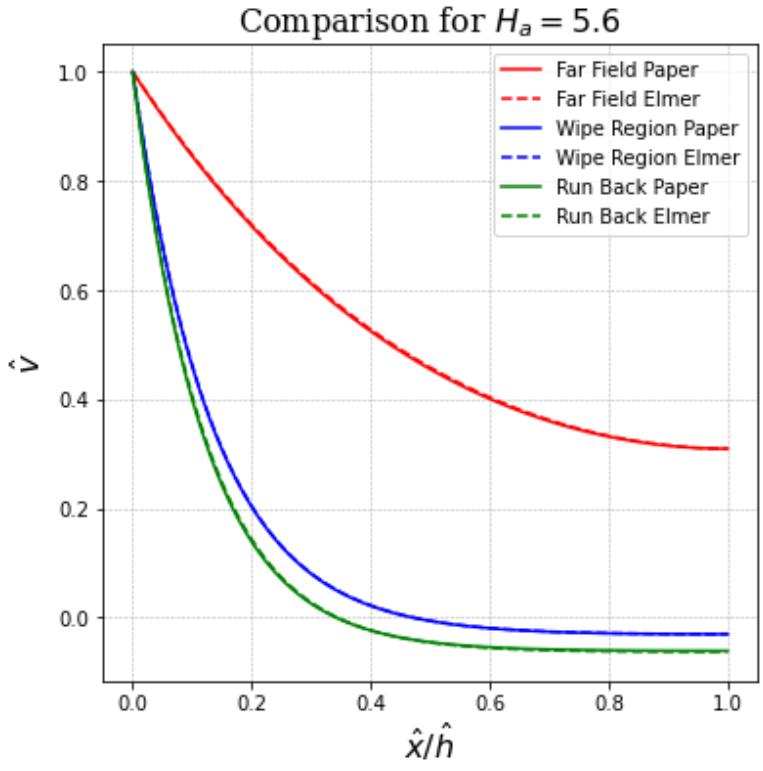
Table (10.4) displays the three regions' coordinate in streamwise direction according to the geometry of this simulation depicted in figure (10.1).

$Region$	$Dimensional\ coordinate\ h$	$Non-dimensional\ coordinate\ h/[h]$
$Run\ back$	$0.007\ m$	5.83
$Jet\ wiping$	$0.01\ m$	8.33
$Far\ field$	$0.014\ m$	11.6

Table 10.4: Three regions coordinates in streamwise direction



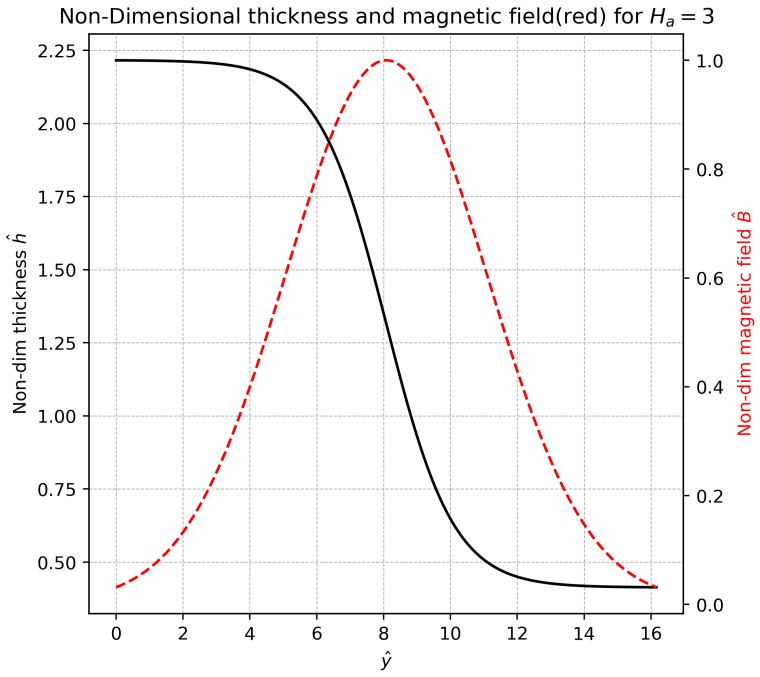
(a)



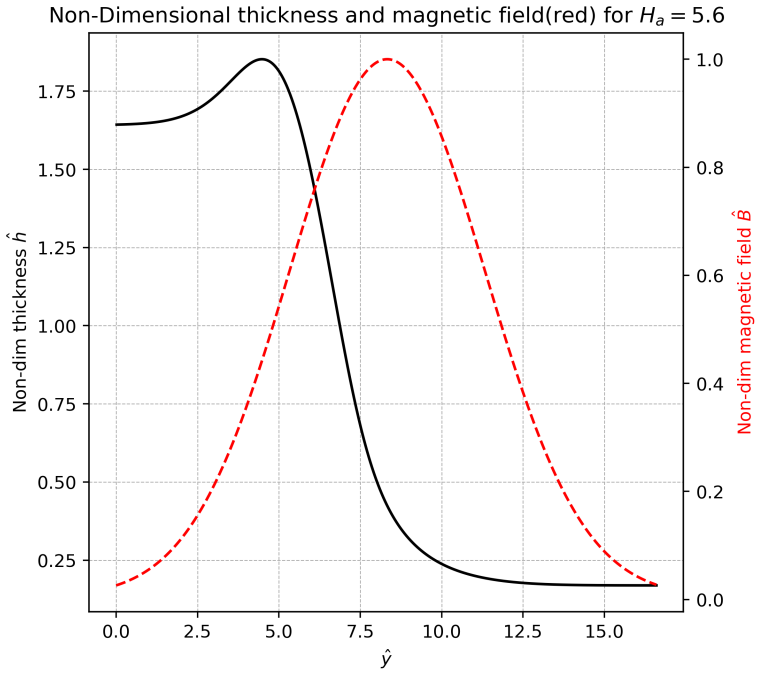
(b)

Figure 10.9: Comparison between the cross velocity profile computed in Elmer and that one obtained from equation (10.3). In the legend Paper indicates the report [19]. The variables are in their dimensionless form

Figure (10.10) shows the interaction between the zinc film thickness h with the magnetic field B according to the electromagnetic model [19].



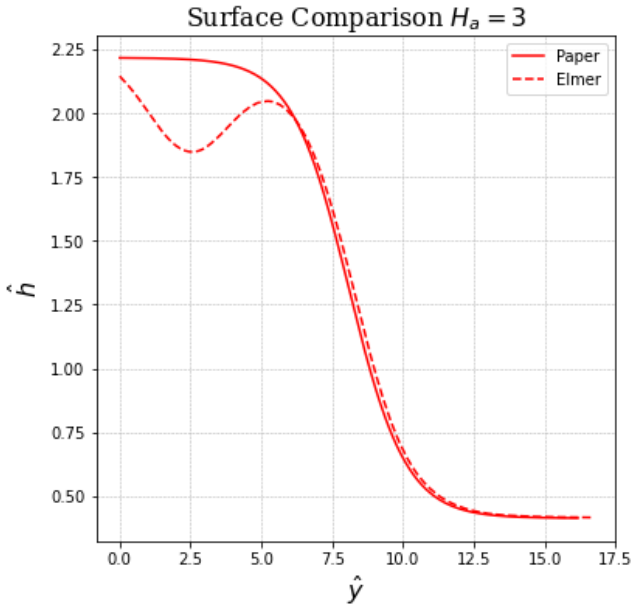
(a)



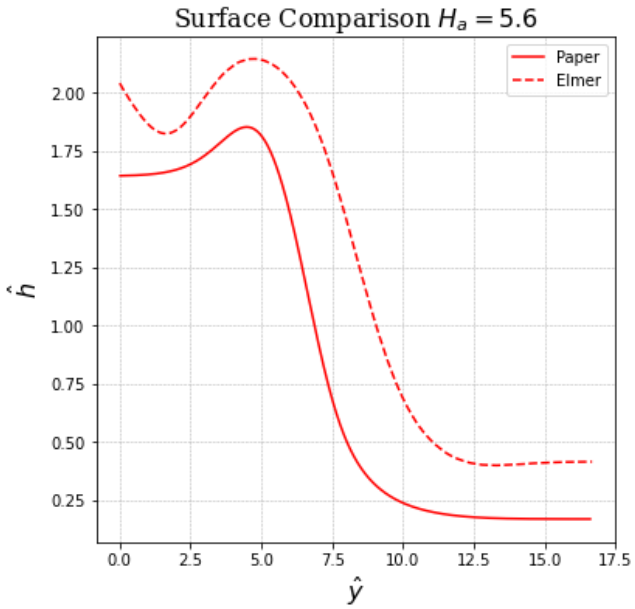
(b)

Figure 10.10: Interaction of the film thickness with the magnetic field for two values of Hartmann number. The variables are in their dimensionless form

With respect to the initial free surface, a small hollow has occurred in the run back region, for both Hartmann numbers. For $H_a = 3$ the concavity is smaller than that one for $H_a = 5.6$, as shown in figure (10.11), since the magnitude of the magnetic field with $H_a = 3$ is lower. This mismatch is caused by the dynamic evolution of the problem and not by the boundary conditions, since their change has produced the same results.



(a)



(b)

Figure 10.11: Comparison between the initial free surface and the one computed in Elmer. The variables are in their dimensionless form

Chapter 11

Conclusions

This manuscript has presented the validation campaign of the numerical results of the electromagnetic model simulating the magnetic action dumping against the undulations on a zinc film after the jet wiping. The investigation realized in Elmer has confirmed the expected results presented in [19].

To facilitate the comprehension of the magnetic induction solver in Elmer, the Hartmann problem (chapter 7) has been a key example. Particularly, this case has allowed understanding the right combination of boundary conditions for the MHD problem this manuscript deals with.
The test cases realized in Elmer represent a simplified model since just a steady external magnetic field has been considered. Moreover, the magnetization of zinc has been neglected. Indeed, Elmer's "magnetic induction" solver does not consider the magnetization **M** of the material since it does not appear in the magnetic induction equation. Regardless, zinc is a diamagnetic material and its magnetic susceptibility χ_m can be approximated to unity like the air's one. Thus, this simplification has not affected the results, although it has not permitted considering the hysteresis effect of the induced magnetic field in the zinc. Another severe limitation has been represented by the only possibility to use the "magnetic induction" solver mentioned above for the resolution of the magnetic induction equation. Indeed, the other magnetic field solvers can not be matched with the Navier-Stokes equation as explained in the model's guide [20].
While the convergence in the simulations concerning the uniform magnetic field has been reached, there have been some numerical limitations in the last test case, dealing with a Gaussian magnetic field. These constraints, particularly the huge time computing and the inability of parallel computing (MPI), have not permitted verifying the convergence and investigating on the unsteady case. In addition, the differences between the surface in the initial condition and the one after the computation in Elmer, which is striking for $H_a = 5.6$, deserve a wider inquiry which has not been possible to realize.

In the light of the considerations mentioned above, a further study should relate to overcoming the numerical limitations caused by the resolution schemes with the scope of verifying the convergence for the test case with a Gaussian magnetic field [19]. Thereby, a deeper investigation should be led in terms of numerical methods implemented in Elmer's solvers, particularly one should examine the relaxation factor's influence in the convergence and possibility to provide the free surface coordinates to level-set solver. Besides, the actual version of this software does not allow the parallel computing (MPI) for umfpack method, which is the best resolution scheme for this MHD problem; thereby, it would need to modify the actual version of Elmer to insert the magnetization effect in the "magnetic induction" solver.

Bibliography

[1] *Backward discretization formula*. URL: https://en.wikipedia.org/wiki/Backward_differentiation_formula.

[2] N.A. Battista. *An Introuction to Magnetohydrodynamics*. Stony Brook University. 2010.

[3] *Boundary conditions in Magnetostatics*. URL: https://quickfield.com/help/QuickField.chm/html/Theory/BoundaryConditionsInMagnetostatics.htm.

[4] Daniel Chopp. *Computing Minimal Surfaces via Level Set Curvature Flow*. Vol, 106,pp. 77-91. Journal of Computational Physics, 1993.

[5] W. G Dallin. *Continuous Hot-Dip Galavanizing 2016*. 2018.

[6] P.A. Davidson. *Introduction to Magnetohydrodynamics*. University of Cambridge, 2017.

[7] M. Dumont et al. *New DC Electromagnetic Wiping System for Hot-Dip Coating*. 2011.

[8] R. Ernst et al. *Working principle of an Electromagnetic Wiping System*. Vol.45,No.1,pp. 59-71. 2009.

[9] A.E. Fisher, M.G Hvasta, and E. Kolemen. *Study of liquid metal surface wave damping in the presence of magnetic fields and electrical currents*. Vol. 19,pp. 101-106. 2019.

[10] J.P. Hartnett and M. Kostic. *Heat Transfer to Newtonian and Non-Newtonian Fluids in Rectangular Ducts*. Vol. 19, pp. 247-356. 1989.

[11] F. W. Hughes and A. J. Brighton. *Theory and Problems of Fluid Dynamics*. McGRAW-HILL, 1999.

[12] Gustavo Krause. *Introducción a la Dinámica de Fluidos Computacional*. Apunte de Catedra. Universidad Nacional de Córdoba, 2020.

[13] M. Loss. "Implementation of level set methods for moving interfaces on adaptive grids," Politecnico di Torino. MA thesis. 2018.

[14] P. Mazzoldi, M. Nigro, and c. Voci. *Fisica Volume II*. EdiSes, 2017.

[15] M.A. Mendez, A. Gosset, and J.M. Buchlin. *Integral Modelling of The Jet Wiping Process: Part1, Introduction and Self-Similar Formulation*. Technical Note 216a. 2018.

[16] M.A. Mendez et al. *Dynamics of the Jet Wiping Process via Integral Models*. 2020.

[17] Stanley Osher and Ronald Fedkiw. *Level Set Methods and Dynamic Implicit Surfaces*. Springer, 2003.

[18] A Panchal. "Study of Liquid Metal MHD Flows Using OpenFoam". MA thesis. 2019.

[19] Fabio Pino and Miguel Mendez. *Integral modelling of the jet wiping: Electromagnetic model*. 2021.

[20] Peter Råback. *Elmer Models Manual*. CSC - IT Center for Science, 2013.

[21] Peter Råback. *Elmer Solvers Manual*. CSC - IT Center for Science, 2013.

[22] Peter Råback. *ElmerGrid Manual*. CSC - IT Center for Science, 2013.

[23] Peter Råback and M Malinem. *Overview of Elmer*. CSC - IT Center for Science. 2013.

[24] P. Saravanan and S. Srikanth. *Surface Defects and their Control in Hot Dip Galvanized and Galvannealed Sheets*. Vol. 5, Issue 11,pp. 11-23. 2018.

Appendix A: Codes

Steady Liquid Film Model Without Magnetic Field in Elmer

```
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!!! Parameters                                                        !!!
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!

$ bw=1e-5 ! viscosity bandwidth
$ E=1e-5 ! smooth coefficient
$ rhoMin=1.0 ! air density [kg/m^3]
$ rhoPlus = 6500.0 ! zinc density [kg/m^3]
$ muMin = 0.00001 ! air dynamic viscosity  [Pa*s]
$ muPlus = 0.0029 ! zinc dynamic viscosity [Pa*s]
$ nuPlus =  muPlus/rhoPlus ! zinc kinematic viscosity [m^2/s]
$ Up= 0.5 ! strip velocity [m/s]
$ g= 9.81 ! gravity acceleration [m/s^2]
$ h = 0.5 ! non-dimensional film thickness
$ h1 = sqrt(nuPlus*Up/g) ! film thickness reference quantity [m]
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!

Header
  Mesh DB "." "."
  Include Path ""
  Results Directory ""
End



Simulation
  Max Output Level = 5 ! For verbosity
  Coordinate system = "Cartesian 2D"
```

```
29   Coordinate mapping(3) = 1 2 3
30
31   Simulation Type = Transient ! Steady state
32
33   Timestepping Method = String BDF
34   BDF Order = Integer 2
35
36   Timestep sizes = Real 0.01
37   Timestep Intervals = Integer 10000
38
39   Timestep Function
40   Real Procedure "LevelSet" "LevelSetTimestep"
41   LevelSet Courant Number = Real 0.5
42
43   Steady State Max Iterations = Integer 1
44   Output Intervals = Integer 500
45   Post File = case.vtu
46 End
47
48
49
50 Body 1
51   Name = "Vessel"
52   Equation = 1
53   Material = 1
54   Body Force = 1
55   Initial Condition = 1
56 End
57
58
59
60 Body Force 1
61   Name = "BodyForce 1"
62   Flow Bodyforce 2 = -9.816
63 End
64
65
66
67 Initial Condition 1
68
69
70   Surface = Variable Coordinate 1
```

```
71        Real MATC "-tx+h*h1" ! Free Surface function
72    Velocity 1 = 0
73
74    Velocity 2 = Variable Coordinate 1
75         Real MATC "if (tx<=h*h1) (0.5*g*tx^2)/nuPlus - h*h1*tx*g/
              nuPlus +Up; else 0;"
76  End
77
78
79  Equation 1
80    Active Solvers(4) = 1 2 3 4
81  End
82
83
84
85  Material 1
86    ! Air has negative values of LS marker function
87    Name = "Zinc-air"
88  ! Hyperbolic tangent functions describing fluid properties
89    Density = Variable Surface
90        Real MATC "(rhoPlus-rhoMin)*(0.5*(1+tanh(3.14*tx/E)))+rhoMin"
91    Viscosity = Variable Surface
92        Real MATC "(muPlus-muMin)*(0.5*(1+tanh(3.14*tx/E)))+muMin"
93
94    LevelSet Bandwidth = Real $ bw
95    Levelset Velocity 1 = Equals Velocity 1
96    Levelset Velocity 2 = Equals Velocity 2
97  End
98
99
100
101 Solver 1
102   Equation = String "Level Set"
103   Variable = String Surface
104   Variable DOFs = Integer 1
105   Procedure = File "LevelSet" "LevelSetSolver"
106   Steady State Convergence Tolerance = Real 1e-6
107
108   Stabilize = Logical True
109   Nonlinear System Relaxation Factor = Real 1.0
110   Nonlinear System Max Iterations = Integer 1
111   Nonlinear System Convergence Tolerance = Real 1.0e-6
```

```
112
113   Linear System Solver = direct
114   Linear System Direct Method = umfpack
115 End
116
117 Solver 2
118   Equation = String "Reinitialize"
119   Variable = String Renorm
120   Variable DOFs = Integer 1
121   Procedure = File "LevelSet" "LevelSetDistance"
122   Extract Interval = Integer 2
123   Reinitialize Interval = Integer 10
124 End
125
126 Solver 3
127   Equation = "Navier-Stokes"
128   Linear System Solver = direct
129   Linear System Direct Method = umfpack
130
131   Stabilize = Logical True
132   Nonlinear System Relaxation Factor = Real 0.5
133   Nonlinear System Max Iterations = Integer 10
134   Nonlinear System Convergence Tolerance = Real 1.0e-3
135   Nonlinear System Newton After Iterations = 10
136   Nonlinear System Newton After Tolerance = 0
137 End
138
139 Solver 4
140    Exec Solver = always
141    Procedure = File "SaveData" "SaveMaterials"
142    Parameter 1 =  String Density
143    Parameter 2 = Viscosity
144 End
145
146
147 Boundary Condition 1
148   Target Boundaries = 3
149   Name = "wall"
150   Velocity 1 = 0
151   Velocity 2 = 0
152 End
153
```

```
154 Boundary Condition 2
155   name = "moving wall"
156   target boundaries = 1
157   Velocity 1 = 0
158   Velocity 2 = $ Up ! [m/s]
159
160 End
161
162 Boundary Condition 3
163   name = "Inlet"
164   target boundaries = 4
165   Velocity 1 = 0
166 End
167
168 Boundary Condition 4
169   name = "Outlet"
170   target boundaries = 2
171   Velocity 1 = 0
172 End
```

Steady Liquid Film Model Without Magnetic Field in Matlab

```
clear all
close all
clc

mu=0.0029; %zinc dynamic viscosity [Pa*s]
rho=6500; %zinc density [kg/m^3]
nu=mu/rho; %zinc kinematic viscosity [m^2/s]
g=9.8; %gravity acceleration [m/s]
Up=0.5; %strip velocity [m/s]
h=[0.5 1 1.5]; %non-dimensional film thicknesses
h1=sqrt(nu*Up/9.8); %reference quantity of the film thickness [m]
tx=h.*h1; %film thickness [m]

for i=1:length(tx)
  x=linspace(0,tx(i),625); %wall-normal coordinates [m]
  v=(0.5*g*(x).^2)/nu -tx(i)*(x).*g/nu +Up; %stream cross velocity [
     m/s]
  v1=v./Up; %non-dimensional velocity
  x1=x./tx(i); %non-dimensional wall-normal coordinate
  plot(x1,v1)
  lgd{i} = strcat('$\hat{h}=$',num2str(h(i)));
  hold on
  grid on
end
legend(lgd,'Interpreter','latex')
xlabel("x")
ylabel("U_y")
```

Hartmann Problem in Elmer

```
$ Ha= 15 ! Hartmann number

Header
  CHECK KEYWORDS Warn
  Mesh DB "." "."
  Include Path ""
  Results Directory ""
End

Simulation
  Max Output Level = 5
  Coordinate System = Cartesian
  Coordinate Mapping(3) = 1 2 3
  Simulation Type = Transient
  Steady State Max Iterations = 1
  Output Intervals = 10
  Timestepping Method = BDF
  BDF Order = 1
  Timestep Intervals = 50
  Timestep Sizes = 1
  Solver Input File = case.sif
  Post File = case_15.vtu
End

Constants

End

Body 1
  Target Bodies(1) = 1
  Name = "Body 1"
  Equation = 1
  Material = 1
  Body Force = 1
  Initial condition = 1
End

Solver 1
  Equation = Navier-Stokes
  Variable = Flow Solution[Velocity:2 Pressure:1]
```

```
  Procedure = "FlowSolve" "FlowSolver"
  Stabilize = True
  Optimize Bandwidth = True
  Steady State Convergence Tolerance = 1.0e-5
  Nonlinear System Convergence Tolerance = 1.0e-4
  Nonlinear System Max Iterations = 1
  Nonlinear System Newton After Iterations = 3
  Nonlinear System Newton After Tolerance = 0.0
  Nonlinear System Relaxation Factor = 0.5
  Linear System Solver = Iterative
  Linear System Iterative Method = BiCGStabL
  Linear System Max Iterations = 2000 ! 1000
  Linear System Convergence Tolerance = 1.0e-5
  Linear System Preconditioning = ILU1
  Linear System Residual Output = 1
End

Solver 2
  Equation = "Magnetic field solver"
  Variable = Magnetic Field
  Procedure = "MagneticSolve" "MagneticSolver"
  ! Exec Solver = Never
  Variable DOFs = 3
  Exported Variable 1 = -dofs 3 lorentz force
  Exported Variable 2 = -dofs 3 electric current
  Stabilize = True
  Optimize Bandwidth = True
  Steady State Convergence Tolerance = 1.0e-3
  Nonlinear System Convergence Tolerance = 1.0e-4
  Nonlinear System Max Iterations = 1
  Nonlinear System Newton After Iterations = 3
  Nonlinear System Newton After Tolerance = 0.0
  Nonlinear System Relaxation Factor = 0.5
  Linear System Solver = Iterative
  Linear System Iterative Method = BiCGStabL
  Linear System Max Iterations = 2000
  Linear System Convergence Tolerance = 1.0e-4
  Linear System Preconditioning =  ILU1
  Linear System Abort Not Converged = False
  Linear System Residual Output = 1
  Linear System Precondition Recompute = 1
End
```

```
83
84 Equation 1
85   Name = "Equation 1"
86   Active Solvers(2) = 1 2
87 End
88
89 Body Force 1
90   Lorentz Force = Logical True
91 End
92
93 Material 1
94   Name = "Zinc"
95   Viscosity = 0.0029
96   Density = 6500
97   Magnetic Permeability = 1.2e-6
98   Electric Conductivity = 1.6e7
99   Applied Magnetic Field 2 = $ Ha/(0.015*sqrt(1.6e7/(0.0029)))
100 End
101
102 Initial Condition 1
103   Name = "InitialCondition 1"
104   Velocity 2 = 0
105   Velocity 1 = 0
106 End
107
108 Boundary Condition 1
109   Target Boundaries(2) = 2 4
110   Name = "walls"
111   Noslip Wall BC = True
112   Magnetic Field 1 = Real 0
113   Magnetic Field 2 = Real 0
114   Magnetic Field 3 = Real 0
115 End
116
117 Boundary Condition 2
118   Target Boundaries(1) = 1
119   Name = "inlet"
120   Velocity 1 = 0.0015
121   Velocity 2 = 0.0
122 End
123
124 Boundary Condition 3
```

```
125   Target Boundaries(1) = 3
126   Name = "outlet"
127   Velocity 2 = 0
128 End
```

Steady Liquid Film Model: Expected Velocity Profile in Presence of a Magnetic Field as Initial Condition in Elmer

```
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!!!Parameters
   !!!
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!

$ bw=0.03 ! 1e-5 ! delta coefficient of level set
$ E=1e-5 ! smooth coefficient
$ rhoMin=1.0 ! air density [kg/m^3]
$ rhoPlus = 6500.0 ! zinc density [kg/m^3]
$ muMin = 0.00001 ! air dynamic viscosity  [Pa*s]
$ muPlus = 0.0029 ! zinc dynamic viscosity [Pa*s]
$ nuPlus =  muPlus/rhoPlus ! zinc kinematic viscosity [m^2/s]
$ Up= 0.5 ! strip velocity [m/s]
$ g= 9.81 ! gravity acceleration [m/s^2]
$ h = 0.5 !non-dimensional film thickness
$ h1 = (nuPlus*Up/g)^0.5 ! reference quantity of film thickness [m]
$ sigma= 1.6e7 ! electrical conductivity
$ B= 0.5 ! reference quantity of magnetic field [T]
$ Ha= B*sqrt(sigma*Up/(rhoPlus*g)) ! Hartmann number
$ relax = 0.1
$ A= tanh(h*Ha)*(1+1/(Ha^2))
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!

Header
  Mesh DB "." "."
  Include Path ""
  Results Directory ""
End



Simulation
  Max Output Level = 5 ! For verbosity
  Coordinate system = "Cartesian 2D"
  Coordinate mapping(3) = 1 2 3

```

```
36   Simulation Type = Transient ! Steady state
37
38   Timestepping Method = String BDF
39   BDF Order = Integer 2
40
41   Timestep sizes = Real 0.01
42   Timestep Intervals = Integer 10000
43
44   Timestep Function
45   Real Procedure "LevelSet" "LevelSetTimestep"
46   LevelSet Courant Number = Real 0.5
47
48   Steady State Max Iterations = Integer 1 ! 1
49   Output Intervals = Integer 100
50   Post File = File case.vtu
51 End
52
53
54
55 Body 1
56   Name = "Vessel"
57   Equation = 1
58   Material = 1
59   Body Force = 1
60   Initial Condition = 1
61 End
62
63
64
65 Body Force 1
66   Name = "BodyForce 1"
67   Flow Bodyforce 2 = -9.816
68   Lorentz Force = Logical True
69 End
70
71
72
73 Initial Condition 1
74 ! free surface function
75   Surface = Variable Coordinate 1
76      Real MATC "-tx+h*h1"
77   Velocity 1 = 0
```

```
78
79    Velocity 2 = Variable Coordinate 1
80          Real MATC "if (tx<=h*h1) (-(A*sinh(Ha*tx/h1)-cosh(Ha*tx/h1)
              *(1+1/(Ha^2))+1/(Ha^2))*Up); else 0;"
81
82  End
83
84
85
86  Equation 1
87    Active Solvers(5) = 1 2 3 4 5
88  End
89
90
91
92  Material 1
93    ! Air has negative values of LS marker function
94    Name = "Zinc-air"
95
96    Density = Variable Surface
97       Real MATC "(rhoPlus-rhoMin)*(0.5*(1+tanh(3.14*tx/E)))+rhoMin"
98    Viscosity = Variable Surface
99       Real MATC "(muPlus-muMin)*(0.5*(1+tanh(3.14*tx/E)))+muMin"
100
101   LevelSet Bandwidth = Real $ bw
102   Levelset Velocity 1 = Equals Velocity 1
103   Levelset Velocity 2 = Equals Velocity 2
104   Magnetic Permeability = 1.2e-6
105   Electric Conductivity = $ sigma
106   Applied Magnetic Field 1 = $ B
107   Applied Magnetic Field 2 = 0
108   Applied Magnetic Field 3 = 0
109
110 End
111
112 Solver 4
113   Equation = Navier-Stokes
114   Variable = Flow Solution[Velocity:2 Pressure:1]
115   Procedure = "FlowSolve" "FlowSolver"
116   Stabilize = True
117   Optimize Bandwidth = True
118   Steady State Convergence Tolerance = 1.0e-5
```

```
119
120   Nonlinear System Convergence Tolerance = 1.0e-4
121   Nonlinear System Max Iterations = 1
122   Nonlinear System Newton After Iterations = 100 ! 3
123   Nonlinear System Newton After Tolerance = 0.0
124   Nonlinear System Relaxation Factor = $relax
125
126   Linear System Solver = Iterative
127   Linear System Iterative Method = BiCGStabL
128   Linear System Max Iterations = 2000
129   Linear System Convergence Tolerance = 1.0e-4
130   Linear System Preconditioning = ILU0
131   Linear System Abort Not Converged = False
132   Linear System Residual Output = 1
133   Linear System Precondition Recompute = 1
134
135   Linear System Solver = direct
136   Linear System Direct Method = umfpack
137
138   Apply Integral BCs = Logical True
139 End
140
141 Solver 2
142   Equation = String "Level Set"
143   Variable = String Surface
144   Variable DOFs = Integer 1
145   Procedure = File "LevelSet" "LevelSetSolver"
146   Steady State Convergence Tolerance = Real 1e-6
147
148   Stabilize = Logical True
149   Nonlinear System Relaxation Factor = Real 1.0
150   Nonlinear System Max Iterations = Integer 1
151   Nonlinear System Convergence Tolerance = Real 1.0e-6
152   Nonlinear System Newton After Iterations = 100
153   Nonlinear System Newton After Tolerance = 0.0
154
155   Linear System Solver = direct
156   Linear System Direct Method = umfpack
157 End
158
159 Solver 3
160   Equation = String "Reinitialize"
```

```
161   Variable = String Renorm
162   Variable DOFs = Integer 1
163   Procedure = File "LevelSet" "LevelSetDistance"
164   Extract Interval = Integer 2
165   Reinitialize Interval = Integer 10
166 End
167
168
169 Solver 1
170   Equation = "Magnetic field solver"
171   Variable = Magnetic Field
172   Procedure = "MagneticSolve" "MagneticSolver"
173
174 ! There is no 2D formulation, this equation always assumed 3D
      problem.
175   Variable DOFs = 3
176   Exported Variable 1 = -dofs 3 lorentz force
177   Exported Variable 2 = -dofs 3 electric current
178   Optimize Bandwidth = True
179
180   Steady State Convergence Tolerance = 1.0e-5
181   Nonlinear System Convergence Tolerance = 1.0e-4
182   Nonlinear System Max Iterations = 1
183   Nonlinear System Relaxation Factor = $relax
184   Nonlinear System Newton After Iterations = 100
185   Nonlinear System Newton After Tolerance = 0.0
186
187   Linear System Solver = Iterative
188   Linear System Iterative Method = BiCGStabL
189   Linear System Max Iterations = 2000
190   Linear System Convergence Tolerance = 1.0e-4
191   Linear System Preconditioning = ILU0
192   Linear System Abort Not Converged = False
193   Linear System Residual Output = 1
194   Linear System Precondition Recompute = 1
195
196   Linear System Solver = direct
197   Linear System Direct Method = umfpack
198 End
199
200 Solver 5
201    Exec Solver = always
```

```
202     Procedure = File "SaveData" "SaveMaterials"
203     Parameter 1 =  String Density
204     Parameter 2 = Viscosity
205 End
206
207
208 Body Force 1
209   Name = "BodyForce 1"
210   Flow Bodyforce 2 = -9.816
211   Lorentz Force = Logical True
212 End
213
214 Boundary Condition 1
215   Target Boundaries = 3
216   Name = "wall"
217
218   Noslip wall BC = True
219   Velocity 1 = 0
220   Velocity 2 = 0
221 End
222
223 Boundary Condition 2
224   name = "moving wall"
225   target boundaries = 1
226   ! Normal-Tangential Velocity = True
227   Velocity 1 = 0
228   Velocity 2 = $ Up ! [m/s]
229   Magnetic Field 1 = 0
230   Magnetic Field 2 = 0
231   Magnetic Field 3 = 0
232
233
234 End
235
236 Boundary Condition 3
237   name = "Inlet"
238   target boundaries = 4
239   Velocity 1 = 0
240
241
242
243 End
```

```
244
245 Boundary Condition 4
246   name = "Outlet"
247   target boundaries = 2
248   Velocity 1 = 0
249
250 End
```

Steady Liquid Film Model: Expected Velocity Profile in Presence of a Magnetic Field as Initial Condition in Matlab

```
%% Steady Liquid Film Model: Expected Velocity Profile...
...in Presence of a Magnetic Field as Initial Condition
clear all
close all
clc


% PARAMETERS
g=9.8; %gravity acceleration [m/s^2]
mu_l=0.0029; %zinc dynamic viscosity [Pa*s]
rho_l=6500; %zinc density [kg/m^3]
nu_l=mu_l/rho_l; %zinc kinematic viscosity [m^2/s]
sigma_m=1.6e7; %zinc electrical conductivity [S/m]
Up=0.5; % strip velocity [m/s]
h=.5; % non dimensional film thickness
B1=0.5; % reference quantity  of the magnetic field [T]
h1=sqrt(nu_l*Up/g);% reference quantity of the film thickness [m]

% VELOCITY PROFILE COMPUTATION
x=linspace(0,h*h1,1000); % normal wall direction [m]
B=0.5; % uniform magnetic field [T]
Ha=B1*sqrt(sigma_m*Up/(rho_l*g)); % Hartmann number
% the sign minus is needed to change the reference system
A=tanh(h*B/B1*Ha).*(1+1/(Ha^2*(B/B1)^2));
v=-(A.*sinh((x./h1)*Ha*B/B1)-cosh((x./h1)*Ha*B/B1).*(1+1/(Ha^2*(B/B1
    )^2))+1/(Ha^2*(B/B1)^2)).*Up;

% DIMENSIONLESS PLOT
x1=x./(h*h1); %non-dimensional wall-normal coordinate
v1= v./Up; %non-dimensional cross stream velocity
figure (1)
hold on
grid on
plot(x1,v1)
xlabel("$\frac{\hat{x}}{\hat{h}[h]}$",'Interpreter','latex')
ylabel("U_y(x)")
```

Steady Liquid Film Model in Presence of a Magnetic Field as Initial Condition in Elmer

```
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!!!Parameters
   !!!
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!

$ bw=0.03 ! delta coefficient of level set
$ E=1e-5 ! smooth coefficient
$ rhoMin=1.0 ! air density [kg/m^3]
$ rhoPlus = 6500.0 ! zinc density [kg/m^3]
$ muMin = 0.00001 ! air dynamic viscosity  [Pa*s]
$ muPlus = 0.0029 ! zinc dynamic viscosity [Pa*s]
$ nuPlus =  muPlus/rhoPlus ! zinc kinematic viscosity [m^2/s]
$ Up= 0.5 ! strip velocity [m/s]
$ g= 9.81 ! gravity acceleration [m/s^2]
$ h = 0.5 ! non-dimensional film thickness
$ h1 = (nuPlus*Up/g)^0.5 ! reference quantity of film thickness [m]
$ sigma= 1.6e7 ! electrical conductivity [S/m]
$ B= 0.5 ! uniform magnetic field equal to [B]
$ Ha= B*sqrt(sigma*Up/(rhoPlus*g)) !Hartmann number
$ relax = 0.01
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!

Header
  Mesh DB "." "."
  Include Path ""
  Results Directory ""
End



Simulation
  Max Output Level = 5 ! For verbosity
  Coordinate system = "Cartesian 2D"
  Coordinate mapping(3) = 1 2 3

  Simulation Type = Transient ! Steady state
```

```
36
37   Timestepping Method = String BDF
38   BDF Order = Integer 2
39
40   Timestep sizes = Real 0.01
41   Timestep Intervals = Integer 20000
42
43   Timestep Function
44   Real Procedure "LevelSet" "LevelSetTimestep"
45   LevelSet Courant Number = Real 0.5
46
47   Steady State Max Iterations = Integer 1
48   Output Intervals = Integer 200
49   Solver Input File = case.sif
50   Post File = case.vtu
51 End
52
53
54
55 Body 1
56   Name = "Vessel"
57   Equation = 1
58   Material = 1
59   Body Force = 1
60   Initial Condition = 1
61 End
62
63
64
65 Body Force 1
66   Name = "BodyForce 1"
67   Flow Bodyforce 2 = -9.816
68   Lorentz Force = Logical True
69 End
70
71
72
73 Initial Condition 1
74 ! free surface function
75   Surface = Variable Coordinate 1, Coordinate 2
76      Real MATC "-tx(0)+(1/800)*(0.1-tx(1))"
77   Velocity 1 = 0
```

```
78
79    Velocity 2 = Variable Coordinate 1
80          Real MATC "if (tx<=h*h1) (0.5*g*tx^2)/nuPlus - h*h1*tx*g/
              nuPlus +Up; else 0;"
81
82  End
83
84
85
86  Equation 1
87    Active Solvers(5) = 1 2 3 4 5
88  End
89
90
91
92  Material 1
93    ! Air has negative values of LS marker function
94    Name = "Zinc-air"
95
96    Density = Variable Surface
97       Real MATC "(rhoPlus-rhoMin)*(0.5*(1+tanh(3.14*tx/E)))+rhoMin"
98    Viscosity = Variable Surface
99       Real MATC "(muPlus-muMin)*(0.5*(1+tanh(3.14*tx/E)))+muMin"
100
101   LevelSet Bandwidth = Real $ bw
102   Levelset Velocity 1 = Equals Velocity 1
103   Levelset Velocity 2 = Equals Velocity 2
104   Magnetic Permeability = 1.2e-6
105   Electric Conductivity = $ sigma
106   Applied Magnetic Field 1 = $ B
107   Applied Magnetic Field 2 = 0
108   Applied Magnetic Field 3 = 0
109
110 End
111
112 Solver 4
113   Equation = Navier-Stokes
114   Variable = Flow Solution[Velocity:2 Pressure:1]
115   Procedure = "FlowSolve" "FlowSolver"
116   Stabilize = True
117   Optimize Bandwidth = True
118   Steady State Convergence Tolerance = 1.0e-5
```

```
119
120   Nonlinear System Convergence Tolerance = 1.0e-4
121   Nonlinear System Max Iterations = 1
122   Nonlinear System Newton After Iterations = 100 ! 3
123   Nonlinear System Newton After Tolerance = 0.0
124   Nonlinear System Relaxation Factor = $relax
125
126   Linear System Solver = Iterative
127   Linear System Iterative Method = BiCGStabL
128   Linear System Max Iterations = 2000
129   Linear System Convergence Tolerance = 1.0e-4
130   Linear System Preconditioning = ILU0
131   Linear System Abort Not Converged = False
132   Linear System Residual Output = 1
133   Linear System Precondition Recompute = 1
134
135   Linear System Solver = direct
136   Linear System Direct Method = umfpack
137
138   Apply Integral BCs = Logical True
139 End
140
141 Solver 2
142   Equation = String "Level Set"
143   Variable = String Surface
144   Variable DOFs = Integer 1
145   Procedure = File "LevelSet" "LevelSetSolver"
146   Steady State Convergence Tolerance = Real 1e-6
147
148   Stabilize = Logical True
149   Nonlinear System Relaxation Factor = Real 1.0
150   Nonlinear System Max Iterations = Integer 1
151   Nonlinear System Convergence Tolerance = Real 1.0e-6
152   Nonlinear System Newton After Iterations = 100
153   Nonlinear System Newton After Tolerance = 0.0
154
155   Linear System Solver = direct
156   Linear System Direct Method = umfpack
157 End
158
159 Solver 3
160   Equation = String "Reinitialize"
```

```
161   Variable = String Renorm
162   Variable DOFs = Integer 1
163   Procedure = File "LevelSet" "LevelSetDistance"
164   Extract Interval = Integer 2
165   Reinitialize Interval = Integer 10
166 End
167
168
169 Solver 1
170   Equation = "Magnetic field solver"
171   Variable = Magnetic Field
172   Procedure = "MagneticSolve" "MagneticSolver"
173
174 ! There is no 2D formulation, this equation always assumed 3D
      problem.
175   Variable DOFs = 3
176   Exported Variable 1 = -dofs 3 lorentz force
177   Exported Variable 2 = -dofs 3 electric current
178   Optimize Bandwidth = True
179
180   Steady State Convergence Tolerance = 1.0e-5
181   Nonlinear System Convergence Tolerance = 1.0e-4
182   Nonlinear System Max Iterations = 1
183   Nonlinear System Relaxation Factor = $relax
184   Nonlinear System Newton After Iterations = 100
185   Nonlinear System Newton After Tolerance = 0.0
186
187   Linear System Solver = Iterative
188   Linear System Iterative Method = BiCGStabL
189   Linear System Max Iterations = 2000
190   Linear System Convergence Tolerance = 1.0e-4
191   Linear System Preconditioning = ILU0
192   Linear System Abort Not Converged = False
193   Linear System Residual Output = 1
194   Linear System Precondition Recompute = 1
195
196   Linear System Solver = direct
197   Linear System Direct Method = umfpack
198 End
199
200 Solver 5
201    Exec Solver = always
```

```
202     Procedure = File "SaveData" "SaveMaterials"
203     Parameter 1 =  String Density
204     Parameter 2 = Viscosity
205 End
206
207
208 Body Force 1
209   Name = "BodyForce 1"
210   Flow Bodyforce 2 = -9.816
211   Lorentz Force = Logical True
212 End
213
214 Boundary Condition 1
215   Target Boundaries = 3
216   Name = "wall"
217
218   Noslip wall BC = True
219   Velocity 1 = 0
220   Velocity 2 = 0
221 End
222
223 Boundary Condition 2
224   name = "moving wall"
225   target boundaries = 1
226   ! Normal-Tangential Velocity = True
227   Velocity 1 = 0
228   Velocity 2 = $ Up
229   Magnetic Field 1 = 0
230   Magnetic Field 2 = 0
231   Magnetic Field 3 = 0
232
233
234 End
235
236 Boundary Condition 3
237   name = "Inlet"
238   target boundaries = 4
239   Velocity 1 = 0
240
241
242 End
243
```

```
244 Boundary Condition 4
245   name = "Outlet"
246   target boundaries = 2
247   Velocity 1 = 0
248
249 End
250
251 ! Solver 1 :: Reference Norm = 5.29208635E-07
252 ! Solver 2 :: Reference Norm = 6.00803071E-01
```

Steady Liquid Film Model: Expected Velocity Profile in Presence of a Gaussian Magnetic Field as Initial Condition in Elmer

```
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!!!Parameters                                                            !!!
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!

$ bw=1e-5  ! viscosity bandwidth ! 0.03
$ rhoMin=1.0 ! air density [kg/m^3]
$ rhoPlus = 6500.0 ! zinc density [kg/m^3]
$ muMin = 0.00001 ! air dynamic viscosity  [Pa*s]
$ E=1e-5 !smooth coefficient for the functions describing fluid
    properties
$ muPlus = 0.0029 ! zinc dynamic viscosity [Pa*s]
$ nuPlus =  muPlus/rhoPlus ! zinc kinematic viscosity [m^2/s]
$ Up= 0.5 ! strip velocity [m/s]
$ g= 9.81 ! gravity acceleration
$ E_ls = 1e-2/4 ! smooth coeffecent for level set interface
$ sigma_m= 1.6e7 ! electrical conductivity [S/m]
$ sigma= 3 ! magnetic field standard deviation
$ y1= 0.0012
$ Ha= 3 ! Hartmann number
$ B_max= Ha/sqrt((sigma_m*Up)/(rhoPlus*g)) ! reference quantity of
    the magnetic field [T]
$ h1 = (nuPlus*Up/g)^0.5 ! reference quantity of the film thickness
    [m]
$ relax = 0.01

!!!!!!!!Functions!!!!!!!!!!!!!!!!!!
$ function B(y) import sigma,B_max,y1 { _B =  ((1 - 5e-3)*exp(-((((y
    -0.01)/y1)^2)/(2*sigma^2)) )+ 5e-3)*B_max }! Gaussian magnetic
    field

$ function h(y) import  E_ls { _h = (3.34e-4-6.23e-5)*0.5*(1+tanh((-
    y+0.01)/E_ls))+6.23e-5} ! free surface function


$ function V(x,y) import Ha,h1,Up,sigma,E_ls,B_max,y1  { _V = -(tanh
    ((h(y)/h1)*Ha*(B(y)/B_max))*(1+1/((B(y)/B_max)^2*Ha^2))*sinh((B(y
    )/B_max)*Ha*x/h1)-cosh((B(y)/B_max)*Ha*x/h1)*(1+1/((B(y)/B_max)
```

```
  ^2*Ha^2))+1/((B(y)/B_max)^2*Ha^2))*Up} ! cross velocity profile

!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!


Header
  Mesh DB "." "."
  Include Path ""
  Results Directory ""
End



Simulation
  Max Output Level = 5 ! For verbosity
  Coordinate system = "Cartesian 2D"
  Coordinate mapping(3) = 1 2 3

  Simulation Type = Transient ! Steady state

  Timestepping Method = String BDF
  BDF Order = Integer 2

  Timestep sizes = Real 0.01
  Timestep Intervals = Integer 20000

  Timestep Function
  Real Procedure "LevelSet" "LevelSetTimestep"
  LevelSet Courant Number = Real 0.5

  Steady State Max Iterations = Integer 1
  Output Intervals = Integer 50
  Post File = File case.vtu
End



Body 1
  Name = "Vessel"
  Equation = 1
  Material = 1
  Body Force = 1
```

```
  Initial Condition = 1
End



Body Force 1
  Name = "BodyForce 1"
  Flow Bodyforce 2 = -9.816
  Lorentz Force = Logical True
End


Initial Condition 1

  Surface = Variable Coordinate 1, Coordinate 2
     Real MATC "-tx(0) +(3.34e-4-6.23e-5)*0.5*(1+tanh((-tx(1)+0.01)/
        E_ls))+6.23e-5"

  Velocity 1 = 0
   Velocity 2 = Variable Coordinate 1, Coordinate 2
      Real MATC "if (tx(0)<=3.18e-4) V(tx(0),tx(1)); else 0;"

End



Equation 1
  Active Solvers(5) = 1 2 3 4 5
End



Material 1
  ! Air has negative values of LS marker function
  Name = "Zinc-air"

  Density = Variable Surface
     Real MATC "(rhoPlus-rhoMin)*(0.5*(1+tanh(3.14*tx/E)))+rhoMin"
  Viscosity = Variable Surface
     Real MATC "(muPlus-muMin)*(0.5*(1+tanh(3.14*tx/E)))+muMin"

  LevelSet Bandwidth = Real $ bw
```

```
111   Levelset Velocity 1 = Equals Velocity 1
112   Levelset Velocity 2 = Equals Velocity 2
113   Magnetic Permeability = 1.2e-6
114   Electric Conductivity = $ sigma_m
115   Applied Magnetic Field 1 = Variable Coordinate 2
116      Real MATC " ((1 - 5e-3)*exp(-((((tx-0.01)/y1)^2)/(2*sigma^2)) )
         + 5e-3)*B_max "
117   Applied Magnetic Field 2 = 0
118   Applied Magnetic Field 3 = 0
119
120 End
121
122 Solver 4
123   Equation = Navier-Stokes
124   Variable = Flow Solution[Velocity:2 Pressure:1]
125   Procedure = "FlowSolve" "FlowSolver"
126   Stabilize = True
127   Optimize Bandwidth = True
128   Steady State Convergence Tolerance = 1.0e-5
129
130   Nonlinear System Convergence Tolerance = 1.0e-4
131   Nonlinear System Max Iterations = 1
132   Nonlinear System Newton After Iterations = 3
133   Nonlinear System Newton After Tolerance = 0.0
134   Nonlinear System Relaxation Factor = $relax
135
136   Linear System Solver = direct
137   Linear System Direct Method = umfpack
138
139   Apply Integral BCs = Logical True
140 End
141
142 Solver 2
143   Equation = String "Level Set"
144   Variable = String Surface
145   Variable DOFs = Integer 1
146   Procedure = File "LevelSet" "LevelSetSolver"
147   Steady State Convergence Tolerance = Real 1e-6
148
149   Stabilize = Logical True
150   Nonlinear System Relaxation Factor = Real 1.0
151   Nonlinear System Max Iterations = Integer 1
```

```
152   Nonlinear System Convergence Tolerance = Real 1.0e-6
153
154   Linear System Solver = direct
155   Linear System Direct Method = umfpack
156 End
157
158 Solver 3
159   Equation = String "Reinitialize"
160   Variable = String Renorm
161   Variable DOFs = Integer 1
162   Procedure = File "LevelSet" "LevelSetDistance"
163   Extract Interval = Integer 2
164   Reinitialize Interval = Integer 10
165   Filename = File "Matlab.dat"
166 End
167
168
169 Solver 1
170   Equation = "Magnetic field solver"
171   Variable = Magnetic Field
172   Procedure = "MagneticSolve" "MagneticSolver"
173
174 ! There is no 2D formulation, this equation always assumed 3D
     problem.
175   Variable DOFs = 3
176   Exported Variable 1 = -dofs 3 lorentz force
177   Exported Variable 2 = -dofs 3 electric current
178   Optimize Bandwidth = True
179
180   Steady State Convergence Tolerance = 1.0e-5
181   Nonlinear System Convergence Tolerance = 1.0e-4
182   Nonlinear System Max Iterations = 1
183   Nonlinear System Relaxation Factor = $relax
184
185   Linear System Solver = Iterative
186   Linear System Iterative Method = BiCGStabL
187   Linear System Max Iterations = 2000
188   Linear System Convergence Tolerance = 1.0e-4
189   Linear System Preconditioning = ILU0
190   Linear System Abort Not Converged = False
191   Linear System Residual Output = 1
192   Linear System Precondition Recompute = 1
```

```
193
194   Linear System Solver = direct
195   Linear System Direct Method = umfpack
196 End
197
198 Solver 5
199    Exec Solver = always
200    Procedure = File "SaveData" "SaveMaterials"
201    Parameter 1 =  String Density
202    Parameter 2 = Viscosity
203 End
204
205
206 Body Force 1
207   Name = "BodyForce 1"
208   Flow Bodyforce 2 = -9.816
209   Lorentz Force = Logical True
210 End
211
212 Boundary Condition 1
213   Target Boundaries = 3
214   Name = "wall"
215
216   Noslip wall BC = True
217   Velocity 1 = 0
218   Velocity 2 = 0
219 End
220
221 Boundary Condition 2
222   name = "moving wall"
223   target boundaries = 1
224   Velocity 1 = 0
225   Velocity 2 = $ Up ! [m/s]
226   Magnetic Field 1 = 0
227   Magnetic Field 2 = 0
228   Magnetic Field 3 = 0
229
230
231 End
232
233 Boundary Condition 3
234   name = "Inlet"
```

```
235   target boundaries = 4
236   Velocity 1 = 0
237
238
239 End
240
241 Boundary Condition 4
242   name = "Outlet"
243   target boundaries = 2
244   Velocity 1 = 0
245
246
247 End
248
249 ! Solver 1 :: Reference Norm = 5.29208635E-07
250 ! Solver 2 :: Reference Norm = 6.00803071E-01
```

Steady Liquid Film Model: Expected Velocity Profile in Presence of a Gaussian Magnetic Field as Initial Condition in Matlab

```
%% Steady Liquid Film in Presence of a Gaussian Magnetic Field
clear all
close all
clc


%FIXED PARAMETERS
prompt1='Hartmann Number=';%defining Hartmann number [adim]
Ha= input(prompt1);
g=9.8;%gravity acceleration [m/s^2]
mu=0.0029; %zinc dynamic viscosity [Pa*s]
rho=6500; %zinc density [kg/m^3]
nu=mu/rho; %zinc kinematic viscosity [m^2/s]
Up=0.5; %strip velocity [m/s]
sigma_t=0.8; % zinc suraface tension [N/m]
E=1e-2/4; %smooth parameter
sigma=3; %magnetic field standard deviation
sigma_m=1.6e7; %zinc electrical conductivity [S/m]


%VARIABLES COMPUTATION
epsilon=(mu*Up/sigma_t)^(1/3); %Film Parameter
h1=sqrt(nu*Up/9.8); % reference quantity of film thickness [m]
y1=h1/epsilon; %reference quantity of streamwise coordinate [m]
y_dim=linspace(0,0.02,1001); %streamwise coordinates [m]
B_max=Ha/sqrt((sigma_m*Up)/(rho*g)); %reference quantity of magnetic
    field [T]
h_dim=0.5*(3.344042480877588e-04-6.237787763685626e-05)*(1+tanh((-
    y_dim+0.01)/E))+6.237787763685626e-05; %film thickness [m]
B_dim=((1 - 5e-3)*exp(-((((y_dim-0.01)/y1).^2)./(2*sigma^2)) )+ 5e
    -3)*B_max; %magnetic field [T]


%VELOCITY PROFILES COMPUTATION
for i=[350 500 700]
x_dim=linspace(0,h_dim(i),1001); %wall-normal coordinates [m]
A1=tanh((h_dim(i)./(h1)).*(B_dim(i)/B_max)*Ha).*(1+1./(Ha^2*(B_dim(i
    )/B_max).^2));
```

```
36 v_dim=-(A1.*sinh((x_dim./h1).*Ha.*B_dim(i)/B_max)-cosh((x_dim./h1).*
       Ha.*B_dim(i)/B_max).*(1+1./(Ha^2.*(B_dim(i)/B_max).^2))+1./(Ha
       ^2.*(B_dim(i)/B_max).^2)).*Up;
37
38 figure(1)
39 x1=x_dim/h_dim(i); %non-dimensional wall-normal coordinate [adim]
40 v1=v_dim/Up; %non-dimensional cross stream velocity [adim]
41 plot(x1,v1);
42 grid on
43 hold on
44 end
45
46 %PLOTS
47 legend('Run Back','Wipe Region','Far Field')
48 xlabel('$\hat{x}/\hat{h}$','Interpreter','latex')
49 ylabel('$\hat{v}$','Interpreter','latex')
50 title('Comparison among velocity profiles')
51
52 figure(2)
53 yyaxis left
54 plot(y_dim,B_dim)
55 xlabel("y [m]")
56 ylabel("B [T]")
57 yyaxis right
58 plot(y_dim,h_dim)
59 ylabel("h [m]")
60 grid on
61 title('Dimensional thickness and magnetic field')
```

Appendix B: Finite Element Method

The finite element method is a powerful numerical technique and, as declared in the previous chapter, it is at the base of Elmer. A brief description of the fundamentals of this technique are below described. The information used for the drafting of this chapter can be found in [12].

Overview

The finite element method (FEM) is one of the most popular methods used to resolve contour values problems numerically. It derives from the weighted residual method, where the solution can be written as a finite expansion, without considering a particular test function:

$$\phi(\mathbf{x}) \approx \hat{\phi}(\mathbf{x}) = \sum_{j=1}^{M} a_j \psi_j(\mathbf{x}) \tag{1}$$

Where **x** is for the vector (x, y, z). The coefficients a_j are unknown and have to be found, while the functions $\psi_j(\mathbf{x})$ are the shape functions or test functions. The approximate solution $\hat{\phi}$ is known as the test solution and the choice of different types of test functions gives rise to the different weighted residual methods.
Assume no particular test function $\psi_j(\mathbf{x})$, It is now required that approximate solution $\hat{\phi}$ is equal to the function ϕ over a set of M points or nodes with coordinate $\mathbf{x}_k$

$$\hat{\phi}(\mathbf{x}_k) = \phi(\mathbf{x}_k) = \sum_{j=1}^{M} a_j \psi_j(\mathbf{x}_k) \quad k = 1, \ldots, M \tag{2}$$

We can write the (2) in matrix form:

$$\mathbf{\Phi} = [\mathbf{\Psi}]\mathbf{a}$$

Where $\psi_{k,j} = \psi_j(x_k)$ are the elements of the matrix $[\mathbf{\Psi}]$ and $\phi(\mathbf{x}_k)$ and a_k are the elements of the vectors $\mathbf{\Phi}$ and **a** respectively. By premultiplying both members of the above expression by the inverse of the matrix $[\mathbf{\Psi}]$ yields an expression for the coefficients a_j of the expansion (2):

$$[\mathbf{\Psi}]^{-1}\mathbf{\Phi} = \mathbf{a}$$

whereby

$$a_j = \sum_{k=1}^{M} \psi_{jk}^{-1} \phi(\mathbf{x}_k) \quad j = 1, \ldots, M$$

Where ψ_{jk}^{-1} is the element jk of the matrix $[\mathbf{\Psi}]^{-1}$. Now, by replacing the previous equality in (2):

$$\hat{\phi}(\mathbf{x}_k) = \phi(\mathbf{x}_k) = \sum_{j=1}^{M} \sum_{k=1}^{M} \psi_{jk}^{-1} \phi(\mathbf{x}_k) \psi_j(\mathbf{x}_k)$$

We can rewrite the expression above as:

$$\hat{\phi}(\mathbf{x}_k) = \sum_{k=1}^{M} \tilde{\phi}_h N_k(\mathbf{x})$$

where $\tilde{\phi}_h = \phi(\mathbf{x}_k)$ is the dependent variable and

$$N_k(\mathbf{x}) = \sum_{j=1}^{M} \psi_{jk}^{-1} \psi_j(\mathbf{x}_k)$$

is the shape function. In a computational point of view, computing the inverse matrix $[\mathbf{\Psi}]^{-1}$ for $M >> 1$ and arbitrary test functions ψ_j can be very expensive. Moreover, the result in this condition is likely to be inaccurate. However, there is a very simple strategy to avoid this problem, which is to choose a basis of test functions such that:

$$\psi_j(\mathbf{x}_k) = \begin{cases} 1 & if \quad j = k \\ 0 & if \quad j \neq k \end{cases}$$

Hence, $[\mathbf{\Psi}] = [\mathbf{\Psi}]^{-1} = \mathbf{I}$, thereby, the coefficients a_j of (2) are the nodal values of ϕ. In this way, test functions can be defined as interpolation functions of the nodal value within a small region of the domain bounded by the neighbouring nodes.

Local Test Functions of Compact Support

The discretization process in the finite element method consists of subdividing the spatial domain Ω into small non-overlapping Ω^e subdomains called elements. These elements are defined by the nodes that make up the domain and their shape depends on the dimension of the problem and the type of discretization chosen. In one-dimensional domains the finite elements are line segments given by the two nodes located at the ends of each one of them, while in two-dimensional domains triangles or quadrilaterals are usually used, which are defined by the nodes at the vertices. In three-dimensional domains, different polyhedra are used, such as tetrahedra (four faces), hexahedra (six faces), triangular prisms, among others. As we will see in this section, in addition to the vertex nodes, a finite element can have more nodes arranged on its sides or faces or in its interior, which allow to increase the order of the approximation. Like the grids used in the finite difference method, a finite element mesh can be structured or unstructured, but in this type of discretization the process offers much greater flexibility for modeling complex geometries.
Figures (1a) and (1b) show, as an example, two 2D meshes, a structured quadrilateral mesh and an unstructured triangle mesh, both used to model the flow around an airfoil. According to its formulation, the approximation proposed by the finite element method is written in terms of an interpolation based on the nodal values of the dependent variable and of the shape functions N_k

$$\hat{\phi}(\mathbf{x}_k) = \sum_{k=1}^{M} \tilde{\phi}_h N_k(\mathbf{x}) \tag{3}$$

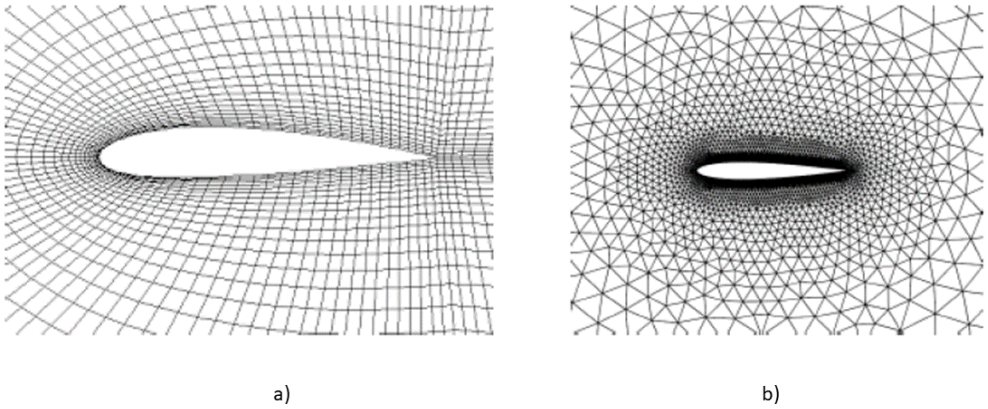


Figure 1: Figure a) structured finite element mesh of quadrilaterals. Figure b) unstructured mesh of triangles. They are both used for modeling two-dimensional flow around an airfoil airfoil.[12]

where the functions must comply with the imposed condition:

$$\begin{aligned} N_k(\mathbf{x}_k) &= 1 \quad j = k \\ N_j(\mathbf{x}_j) &= 0 \quad j \neq k \end{aligned} \tag{4}$$

The simplest way to satisfy these conditions is to use compact support form functions, which means that a function N_k is zero over the entire domain except in a relatively small region of the domain that includes the $\mathbf{x}_k$ coordinate node. This region is made up of the elements that share that node, so that, within a given element Ω^e, the only non-zero shape functions are those corresponding to the nodes that are part of the element. Thus, the approximate solution $\hat{\phi}(\mathbf{x})$ in the region $\mathbf{x} \in \Omega^e$ is written in terms of an interpolation expressed only by the nodal values of the nodes belonging to the Ω^e element and the corresponding shape functions,namely:

$$\hat{\phi}(\mathbf{x}) = \sum_{k^e=1}^{M^e} \tilde{\phi}_{k^e} N^e_{k^e}(\mathbf{x}) \quad \mathbf{x} \in \Omega^e \tag{5}$$

where M^e is the number of nodes that made up the element Ω^e, k^e denotes the local numeration of the element nodes and $N^e_{k^e}$ are the locally defined shape functions.

Interpolation in One-Dimensional Domains

Consider a 1D case $0 \leq x \leq l$ discretized by $N-1$ elements of two nodes each one, like figure (2)

$$\Omega^e_k = [x_k, x_{k+1}] \quad k = 1, \ldots, M$$

A one-dimensional two-node element allows a linear interpolation of the function $\phi(x)$ to be constructed within each element. In global terms this approximation inside the element Ω^e_k is:

$$\hat{\phi}(x) = \tilde{\phi}_k N_k(x) + \tilde{\phi}_{k+1} N_{k+1}(x) \quad x \in \Omega^e_k \tag{6}$$

where the shape functions are:

$$x_k \leq x \leq x_{k+1} \quad \begin{cases} N_k = \frac{x_{k+1}-x}{x_{k+1}-x_k} \\ N_{k+1} = \frac{x-x_k}{x_{k+1}-x_k} \end{cases}$$

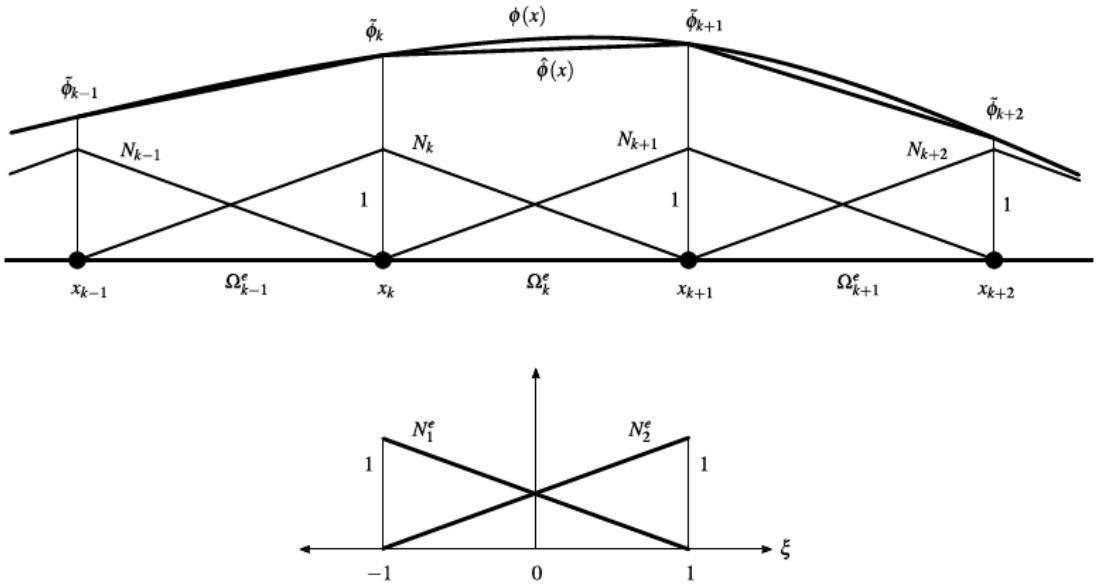


Figure 2: Schematic of the finite element approximation of a one-dimensional function. one-dimensional function.[12]

At the local level, i.e., at the level of each element, the interpolation (6) can be written in relation to a local ξ coordinate, referred to a normalized system with origin at the center of the element Ω^e_k

$$\hat{\phi}(\xi) = \tilde{\phi}_1 N^e_1(\xi) + \tilde{\phi}_2 N^e_2(\xi) = \sum_{k^e=1}^{2} \tilde{\phi}_{k^e} N^e_{k^e}(\xi) \quad -1 \leq \xi \leq 1$$

where, in this case, the local shape functions are

$$-1 \leq \xi \leq 1 \quad \begin{cases} N^e_1 = \frac{1-\xi}{2} \\ N^e_2 = \frac{1+\xi}{2} \end{cases} \tag{7}$$

the transformation from the local system to the global one is given by:

$$x = \frac{1}{2} x_{k+1}(1+\xi) + \frac{1}{2} x_k (1-\xi) \tag{8}$$

where x_k and x_{k+1} are the coordinates of the ends of the one-dimensional element. Note that the ξ coordinate inside the element is precisely the linear interpolation of the value of x at the ends. The expressions for N^e_1 and N^e_2 in equation (7) are the two locally defined shape functions for a linear interpolation. If we wish to obtain a higher order of interpolation, more values of the function ϕ inside the element must be known. That is, we must define interior nodes in the element where the values of ϕ are specified. Thus, for a second-order interpolation, a node is placed at the center

of the element so that the approximation is made in terms of three parabolic shape functions which, obviously, must satisfy the condition (4):

$$-1 \leq \xi \leq 1 \quad \begin{cases} N_1^e = \frac{\xi}{2}(\xi - 1) \\ N_2^e = 1 - \xi^2 \\ N_3 = \frac{\xi}{2}(\xi + 1) \end{cases} \tag{9}$$

These definitions are shown graphically in figure (3)

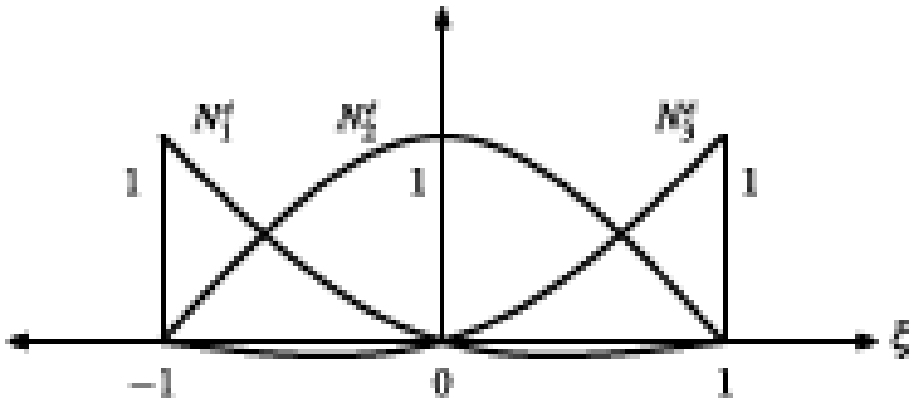


Figure 3: Schematic of shape functions for quadratic interpolation within a one-dimensional three-node element.[12]

Naturally, the discretization of one-dimensional domains has the simplicity that the geometry of a finite element can only be linear and is defined by the two end nodes. In contrast, in multidimensional domains the complexity increases substantially since, in principle, elements of arbitrary shapes could be defined. However, bearing in mind that it is not desirable to increase the complexity of the problem unnecessarily, the shapes of the finite elements are usually reduced to simple figures such as triangles and quadrilaterals in two-dimensional domains and polyhedra with few faces in three-dimensional domains.

Shape Functions Continuity

The way to define local interpolation functions is not unique, but depends on the characteristics one wishes to impart to the approximation. By means of the shape functions (7) and (9) we construct an approximation function $\hat{\phi}$ in terms of the nodal values $\tilde{\phi}_k$, which is continuous over the entire domain including the discretization nodes ($\mathbf{x}_k$ coordinate points), but its derivative is not defined at the ends of the elements. This continuity condition of the approximation function is a consequence of the test functions used in the expansion since the continuity of $\hat{\phi}$ is given directly by the continuity of the N_k functions, whose derivatives are not defined at the end nodes of the elements. The continuity requirements on the shape functions are imposed according to the order of the differential equations to be solved, similar to what happens in the traditional weighted residue method, where the application of the differential operator on the test solution makes it necessary,

in principle, for the latter to be derivable

Constant test functions within each element lead to approximation functions that are piece wise constants which do not satisfy the continuity of the function at the nodes. Test functions that satisfy the conditions (4) but have discontinuities in the first derivative, such as linear functions (7) or parabolic functions (9), lead to approximation functions that satisfy the continuity of $\hat{\phi}$ at all nodes including those at the boundaries of the elements, but not of its first derivative. In that case the shape functions are said to be of C^0 continuity. Shape functions satisfying the conditions (4) and which also satisfy:

$$\begin{aligned} N_k^{'}(\mathbf{x}_k) &= 1 \\ N_j^{'}(\mathbf{x}_j) &= 0 \quad j \neq k \end{aligned} \tag{10}$$

where $N_k^{'}$ is the derivative, allow us to construct an approximation where both $\hat{\phi}$ and its first derivative are continuous and are said to be of C^1 continuity. In general, a basis of shape functions is of C^s continuity when they allow to construct an approximation in which both the variable $\hat{\phi}$ and its s first derivatives are continuous at all the nodes of the discretization.

Continuity C^0 shape functions can be obtained using Lagrange interpolation polynomials, which are defined as the lowest degree polynomials that verify the correspondence between a set of distinct x_i points and their images f_i. The functions given in (7) and (9) are, precisely, particular cases of the Lagrange interpolation polynomials, for two and three points, respectively. In general, taking into account that the local form function $N_{k^e}^e$ satisfies $f_{k^e} = 1$ and $f_{j^e} = 0$ $(j^e \neq k^e)$, the functions N^e on a one-dimensional element of M^e nodes based on the Lagrange interpolation polynomials are:

$$N_{k^e}^e(\xi) = \prod_{1 \leq m \leq M^e} \frac{\xi - \xi_m}{\xi_{k^e} - \xi_m} = \frac{(\xi - \xi_1) \ldots (\xi - \xi_{k^e-1}) (\xi - \xi_{k^e+1}) \ldots (\xi - \xi_{M^e})}{(\xi_{k^e} - \xi_1) \ldots (\xi_{k^e} - \xi_{k^e-1}) (\xi_{k^e} - \xi_{k^e+1}) \ldots (\xi_{k^e} - \xi_{M^e})} \tag{11}$$

The interpolation order of the polynomial must not be confused with the degree of continuity of the shape function. Although by means of this expression we can obtain high interpolation orders, the shape functions are in all cases of continuity C^0.

To obtain shape functions of continuity C^1 or higher, Hermite interpolation polynomials are usually used. It is worth mentioning that, despite the possibility of constructing high-order interpolation functions, these are not generally used, since in this case, systems of equations with relatively dense (multiple) matrices (multiple non-zero elements), which increases the computational cost of solving the system computational cost to solve the system.

Approximation of The Differential Equations Solution

Taking into account the above definitions, in this section we will see how the finite element method is applied to solve differential equations. This development shares many of the features of the traditional weighted residue method, but the use of compact support form functions requires additional treatment to satisfy the continuity requirements of the approximate solution. This often requires

"weakening" the boundary value problem to reduce the continuity requirement on the shape functions, thus allowing simpler, essentially C^0 continuity functions to be used. On the other hand, finite element schemes almost unanimously use the Galerkin method, whereby the weight functions in the integrals of the residue are the compact support shape functions that interpolate the nodal values of the variable inside the elements.

Weak Form of the Differential Equation

The spatial discretization process using the finite element method relies on the integral form of the differential equation, which transforms the boundary value problem into a variational one. Consider for this analysis a boundary value problem constituted by the Poisson equation and an appropriate set of boundary conditions:

$$-\nabla^2\phi = s \quad in \quad \Omega \tag{12}$$

with

$$\begin{aligned} \phi &= \bar{\phi} \quad in \quad \Gamma_{\bar{\phi}} \\ \hat{\mathbf{n}} \cdot \nabla\phi &= \frac{\partial\phi}{\partial n} = \bar{q} \quad in \quad \Gamma_{\bar{q}} \end{aligned} \tag{13}$$

where s is a source term which depends on the $\mathbf{x}$ position and $\hat{\mathbf{n}}$ is the normal vector going out the contour $\Gamma_{\bar{q}}$

Equation (12) together with the boundary conditions (13) are known as the strong formulation of the boundary value problem. A function $\phi(\mathbf{x})$, of at least C^2 continuity (up to the second continuous derivative) that satisfies the differential equation and the boundary conditions is called a classical solution of the problem. To obtain an approximate solution by the finite element method we write the weighted residual formulation:

$$\int_\Omega W_m R_\Omega \, d\Omega = \int_\Omega W_m(\nabla^2\phi + s) \, d\Omega = 0 \tag{14}$$

where the expression is zero because we are considering the residual of the classical solution, which satisfies the differential equation exactly. Note that, if we were to intrude the test solution (3) into equation (14), in principle we would need the shape functions to have bounded derivatives at least up to second order (hence, the first derivatives must be continuous), since the presence of singularities in the derivatives (basically, unbounded values of the derivatives) can generate problems in the evaluation of the integrals. In other words, this formulation would imply that the shape functions must be at least of continuity C^1, so as to guarantee that the second derivatives are bounded over the whole domain. Generally speaking, we can say that for a differential operator $\mathcal{L}$, the above formulation requires that the shape functions be of continuity C^{s-1}, where s is the order of the highest order derivative in the differential operator. This means that, for the Poisson problem which is of second order, form functions of continuity C^0, such as Lagrange interpolation polynomials, could not be used.

To overcome this limitation, we proceed to "weaken" the formulation, by applying the Gaussian divergence theorem to the first term of the residual:

$$\int_\Omega W_m \nabla^2 \phi \; d\Omega = \int_\Omega [\nabla \cdot (W_m \nabla \phi) - \nabla W_m \cdot \nabla \phi] \; d\Omega = \int_\Gamma W_m (\hat{\mathbf{n}} \cdot \nabla \phi) d\Gamma - \int_\Omega W_m \cdot \nabla \phi \; d\Omega$$

Replacing this equality in equation (14) and assuming that the function W_m is zero on the Dirichlet contour $\Gamma_{\bar{\phi}}$, the Neumann boundary condition arises naturally (hence they are often called natural boundary conditions) and gives the weak formulation of the problem of values on the contour:

$$\int_\Omega \nabla W_m \cdot \nabla \phi \; d\Omega - \int_{\Gamma_{\bar{q}}} W_m \bar{q} d\Gamma = \int_\Omega W_m s \; d\Omega \tag{15}$$

where the condition (13) was introduced.

The weak formulation (also called variational form) of the problem allows to reduce the continuity requirement on the function ϕ, which can now be of C^0 continuity, provided that the weight functions W_m are differentiable (even if their first derivative is not continuous). The classical solution of the problem (12) with boundary conditions (13) obviously satisfies the integral equation (15), but this weak formulation only satisfies the differential equation and the Neumann boundary conditions that were naturally introduced to the problem. The fulfillment of the Dirichlet boundary conditions is achieved by correctly choosing the space to which the solution of (15) belongs, which must be such that the solution satisfies (13) in $\Gamma_{\bar{\phi}}$. In terms of the test solution (3), this is achieved by imposing the prefixed values on the Dirichlet contour to the corresponding nodal values, which will no longer be unknowns of the problem. The uniqueness of the weak solution of the problem (12) (i.e., the solution of its weak formulation), is guaranteed by means of the Lax-Milgram lemma.

Once the weak formulation of the problem has been obtained, the next step to obtain the approximate solution by means of the finite element method is to introduce the test solution (3) into the integral formulation. Therefore, considering our Poisson model and using the Galerkin method, we have that:

$$\int_\Omega \nabla N_m \cdot \left[\sum_{k=1}^{M} \tilde{\phi}_k \nabla N_k \right] d\Omega - \int_{\Gamma_{\bar{q}}} N_m \bar{q} \; d\Gamma = \int_\Omega N_m s \; d\Omega \quad m = 1, \ldots, M$$

where M is the problem nodes quantity. Regrouping terms in the expression yields:

$$\sum_{k=1}^{M} \left[\int_\Omega \nabla N_m \cdot \nabla N_k \; d\Omega \right] \tilde{\phi}_k = \int_\Omega N_m s \; d\Omega + \int_{\Gamma_{\bar{q}}} N_m \bar{q} \; d\Gamma \quad m = 1, \ldots, M \tag{16}$$

With the expression (16) we construct the system of algebraic equations whose solution allows us to know the nodal values of the approximate solution of the approximate solution:

$$[\mathbf{K}] \tilde{\phi} = \mathbf{f} \tag{17}$$

where

$$K_{mk} = \int_\Omega \nabla N_m \cdot \nabla N_k \; d\Omega$$

$$f_m = \int_\Omega N_m s \; d\Omega + \int_{\Gamma_{\bar{q}}} N_m \bar{q} \; d\Gamma$$

Note that the system (17) requires some additional treatment in order to correctly consider boundary conditions. In the presence of Dirichlet boundary conditions (essential conditions), the nodal variables are known and must therefore be extracted from the linear system, adding their contribution to the right-hand member and reducing the size of the system. Basically, if there is a Dirichlet condition at node d such that the nodal variable is known ($\tilde{\phi}_d = \bar{\phi}_d$), the column d of the matrix $[\mathbf{K}]$ multiplied by the prefixed value $\bar{\phi}_d$ is subtracted from the right-hand member vector and the row d of the system is cancelled. Thus, the number of equations of the algebraic system is in general not equal to the number of nodes of the discretization, unless there are no regions with essential boundary conditions. On the other hand, taking into account that the shape functions are compactly supported and are generally low-order polynomials, it is convenient to perform the integrals numerically at the local level by means of Gaussian quadratures and, subsequently, to compute the contribution of each element to each node of the global system, in what is known as the global matrix assembly.