Alma Mater Studiorum
– Università degli Studi di Bologna –

Dipartimento di Ingegneria Elettrica

Dottorato di Ricerca in Ingegneria Elettrotecnica

– Ciclo XXIII –

Settore Scientifico Disciplinare di afferenza: ING-IND 31

Numerical strategies for the solution of the magneto-fluid-dynamic problem at Low Magnetic Reynolds Numbers.

Presentata da:
Dott. Chiara Latini

Coordinatore Dottorato:
Chiar.mo Prof. Francesco Negrini

Tutor:
Prof. Ing. Carlo Borghi

Corelatore:
Prof. Ing. Andrea Cristofolini

Esame finale anno 2011

To the DIE (happy) team.

CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF ALGORITHMS

CHAPTER 1

INTRODUCTION

For a successful technology, reality must take precedence over public relations, for nature cannot be fooled.

(Richard Feynman)

In 2004, the 10 Mach hypersonic flight demonstration by the NASA research aircraft *X-43* boosted the research activity about the atmospheric and space hypersonic flight [1], [2], [3]. Magnetohydrodynamics (MHD) is being regarded as a promising field for the applications it has in this area [4], [5], [6], [7], [8], [9].

MHD has been proposed as an added tool in the development of the air breathing engines needed for hypersonic flight exploitation: an MHD system has been suggested to control the fluid dynamics at the inlet of the scramjet [10], [11], [12]. In the Russian AJAX project, MHD techniques are utilized to bypass kinetic energy of the working fluid from the supersonic diffuser to the nozzle, reducing the flow velocity in the combustion chamber to acceptable value, even for high vehicle Mach numbers [13], [14], [15].

An induced MHD interaction can modify the flow field in the boundary layer of a re-entry hypersonic vehicle, moving the shock front away from the body walls, or generating a torque allowing an active flow control without movable parts [16], [17], [18], [19]. The work presented in this thesis is focused on a numerical analysis of the shock moving effect. This process results in a mitigation of the heat fluxes at the aircraft surface.

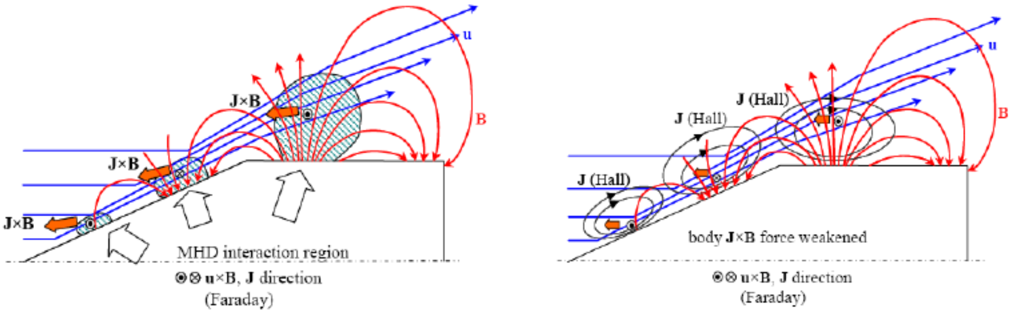
In the re-entry phase, for speeds higher than Mach 12, the temperature at the vehicle surfaces can reach 4000-5000 K. The flow is therefore weakly ionized, and can interact with an electromagnetic field. By equipping the fuselage with a magnet system, capable to generate fields at the order of Tesla, the Lorentz force acting on the charges convected with the fluid generates an electromotive force. The resulting current $\mathbf{J}$ couples with magnetic field generating a body force $\mathbf{J} \times \mathbf{B}$ acting on the flow.

Several magnets configuration have been proposed and tested in wind tunnels facilities [20], [21], [22] [23], [24], [25], [26], [27], [28]. The modeling effort presented here concentrates on a axi-symmetric configuration for the magnets that generates a current flow in the azimuthal direction, positioned where the MHD interaction is stronger, that is, where the flow field is almost orthogonal to the magnetic flux density [29], [30].

The combined effect of the Faraday electric field $\mathbf{E}$ and of the magnetic field causes the current to diffuse in the *Hall* direction $\mathbf{E} \times \mathbf{B}$. It results in a dampening effect on the azimuthal current, that is, in an overall decrease of the MHD body force.

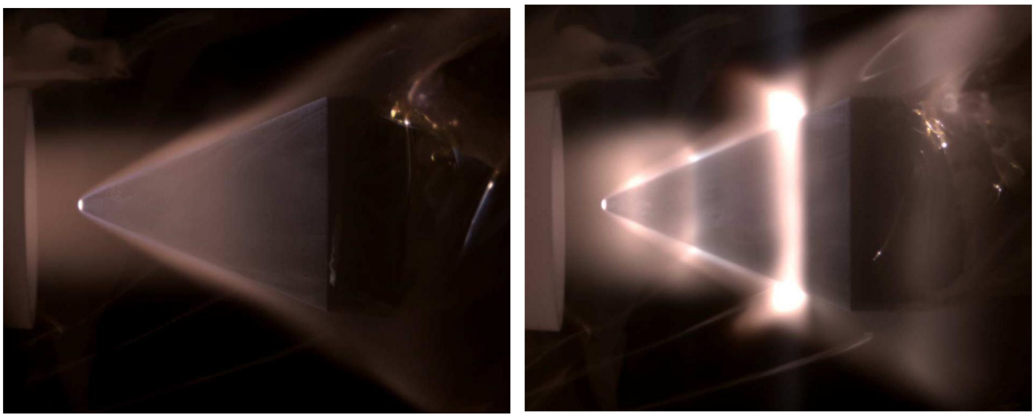
A sketch of the MHD interaction configuration around a conical test body is given figure 1.1; a picture of the flow around a conical test body with and without MHD interaction is shown in fig 1.2. The current and field components in the direction $\mathbf{u} \times \mathbf{B}$, where $\mathbf{u}$ is the mass flow velocity, are referred to as the *Faraday* components, whereas the components in the plane $(\mathbf{B}, \mathbf{E} \times \mathbf{B})$ are named *Hall* components.

The diffusion effect of the current in the Hall direction is modeled by defining a tensor conductivity, that introduces an anisotropy in the electrodynamic problem formulation, that is, in the formulation of the partial differential



(a) **(b)**

Figure 1.1: MHD interaction regions (a) and Hall currents (b).



(a) **(b)**

Figure 1.2: Flow pattern without MHD interaction (a) and with MHD interaction (b).

equation to be solved to determine the current entities. The physical anisotropy generates computational difficulties in the finite element solution, that does not reach convergence in several cases. The strategy to overcome this difficulty is linked to the solver algorithm behavior that has to take into account the preferred direction set by the physical problem. Once obtained the solution, a proper treatment of the finite element results for the potential permits an higher accuracy in the field evaluation.
The modeling activity presented in this work has been performed within the CAST project, funded by the Italian Space Agency (ASI), aimed at developing an integrated experimental-computational environment to explore new possible configurations for aerospace transport.

In the following, chapters 2, 3 present the physical and numerical models respectively, chapter 4 addresses the computational issues touched on above, chapter 5 presents the simulation results obtained.

CHAPTER 2

RESISTIVE MAGNETOHYDRODYNAMICS: THE PHYSICAL MODEL.

Cerca le cose difficili e sarai salvo era scritto. Perchè la salvezza sta nella ricerca. Anche se non si trova. Anche se non si sa cosa si cerca. Anche se non si sa che si cerca.

Rosario Magrì

Abstract. Magnetohydrodynamics is a continuum description of a charged fluid flow in an electromagnetic field. The continuum assumption results in a time and length scale definition for the phenomena that can be described by the model, and allows some simplifications in both the fluid dynamic and electrodynamic description. Fluid dynamics is modeled by the Navier-Stokes equations, with added source terms in momentum and energy equations accounting for the Faraday law and the electromagnetic energy contribution. Electrodynamics is described by the generalized Ohm's law, and by the Maxwell-Faraday and Ampere-Maxwell equations, where the displacement current is neglected. The convection current results divergence free, and the physical model is stated by imposing null divergence for the conduction current expressed by the generalized Ohm's Law. For the studied hypersonic flight applications the magnetic Reynolds number is low, and the magnetic field induced by the current can be neglected compared to the imposed magnetic field, constant in time. The electric field results therefore irrotational, and a scalar potential can be defined.

Introduction.

The physical model the work presented in this thesis is based on, is a one-fluid continuum model. The continuum assumption for the mass flow leads to the classical Navier Stokes equations formulation, with the added terms due to the electromagnetic interaction between the charged fluid and the electromagnetic field. The model is completed by the (generalized) Ohm's law that links the current density to the electric field it generates, and by the Maxwell-Faraday and Ampere-Maxwell equations. This is treated in section 2.1. The generalized Ohm's law describes a conducting fluid where the conductivity has not a scalar nature. It depends on the action of a magnetic field on a collision dominated plasma: collisions favor the diffusion of the charged particles. A tensor expression for the generalized Ohm's law is the strategy adopted to numerically treat the issues implied by anisotropy. This tensor formulation is described in section 2.2. Finally, section 2.3 discusses the dynamic equation for the magnetic field and the adopted low magnetic Reynold numbers approximation.

2.1 Magneto-hydrodynamic equations.

For a many particle plasma and long time scale phenomena, the plasma reaches through collisions a Maxwellian equilibrium and the charge separations saturate. In these conditions, it is possible to formulate a macroscopic model for a plasma as a single fluid. To quantify the time and length scale implied by this assumptions, let us introduce the electrons Debye length and the plasma frequency:

$$\lambda_D = \left(\frac{\epsilon_0 kT}{ne^2}\right)^{1/2} = 69.0 \cdot \left(\frac{T}{n}\right)^{1/2} \quad m \tag{2.1}$$

$$\omega_p = \left(\frac{ne^2}{\epsilon_0 m_e}\right)^{1/2} = 2\pi \cdot 8.97 \cdot n^{1/2} \quad s^{-1} \tag{2.2}$$

where T is the plasma temperature, n is the electron number density, m_e is the electrons mass. The Debye length defines the shielding effect the plasma has on the Coulomb field generated by the single charged particles in it: for distances greater than the Debye length, the Coulomb potential generated by each charge drops and the interactions between charged particles are no longer two body coulomb collisions, but can be described by the action of a mean electromagnetic field on each charge. For the plasma involved in hypersonic flight applications, the Debye length is of order of $4 \cdot 10^{-6}$, much less of the length scales of the phenomena under study.

The plasma frequency describes the response of the plasma to a distortion of the neutrality condition: the charge density fluctuates with an harmonic oscillator behavior, whose oscillation frequency is given by ω_p. For hypersonic flight plasmas, the order of magnitude of the characteristic frequency is $10^9 s^{-1}$. By multiplying the two quantities 2.1, 2.2 we obtain:

$$\lambda_D \omega_p = \left(\frac{kT}{m_e}\right)^{1/2} \simeq \bar{C}_e \tag{2.3}$$

where $\bar{C}_e$ is the mean thermal speed of electrons. Equation 2.3 implies that electrons travel a distance λ_D in a time $\omega_p^{-1}/2\pi$ before the restoring force acts. The distance λ_D is therefore the limiting distance to observe charge neutrality deviations, and the time $\omega_p^{-1}/2\pi$ is the limiting time to observe such a variation. Furthermore, any disturbance of angular frequency greater than ω_p doesn't affect the charge neutrality assumption. Holding this assumptions, a current density in the plasma can be defined as the sum of the current densities due to each conducting species, that is, electrons and ions. The time scale restrictions posed above, permits to neglect the displacement currents compared to the conduction current:

$$\frac{|\partial \mathbf{D}|/|\partial t|}{|\mathbf{J}|} \sim \frac{D_c}{t_c J_c} \sim \frac{\epsilon_0 E_c}{t_c \sigma_c E_c} = \frac{\epsilon_0}{\sigma_c t_c} \ll 1. \tag{2.4}$$

where t_c is the characteristic time scale of the phenomena under study, and σ_c is the characteristic plasma conductivity. For 2.4 to hold it has to be $t_c \gg \epsilon_0/\sigma_c$. Since:

$$\sigma_c \sim \frac{ne^2}{m_e \nu_{eH}}, \tag{2.5}$$

where ν_{eH} is the average collision frequency between electrons and heavy particles, 2.4 holds if

$$t_c \gg \nu_{eH}/\omega_p^2. \tag{2.6}$$

, as mplied by **??**. For the continuum fluid dynamic approximation to hold, it has to be $t_c \gg \nu_{eH}^{-1}$, so 2.6 implies that if $t_c >> \omega_p^{-1}$ the displacement current can be neglected.
Since

$$\frac{\rho_{ch}|\mathbf{u}|}{|\mathbf{J}|} \sim \frac{\epsilon_0}{\sigma_c t_c} \tag{2.7}$$

$$\frac{\rho_{ch}|\mathbf{E}|}{|\mathbf{J} \times \mathbf{B}|} \sim \frac{\epsilon_0}{\sigma_c t_c}, \tag{2.8}$$

where ρ_{ch} is the charge density and $\mathbf{u}$ is the flow velocity, if $t_c >> \omega_p^{-1}$ the convection current can also be neglected compared with the conduction current, so as the force acting on the fluid because of its charge, compared

. . . .

with the Hall forces $\mathbf{J} \times \mathbf{B}$.
It is now possible to state the magnetohidrodynamic (MHD) equations:

$$\frac{\partial \rho}{\partial t} + \nabla(\rho \mathbf{u}) = 0 \tag{2.9}$$

$$\frac{\partial(\rho \mathbf{u})}{\partial t} + \nabla(\rho \mathbf{uu} + p\mathbf{I} - \tau) = \mathbf{J} \times \mathbf{B} \tag{2.10}$$

$$\partial(\rho e_i)/\partial t + \nabla[(\rho e_i + p)\mathbf{u}] - \nabla \cdot (\tau \cdot \mathbf{u}) + \nabla(\mathbf{F_T}) = \mathbf{J} \cdot \mathbf{E} \tag{2.11}$$

$$\mathbf{J} = \sigma(\mathbf{E} + u \times \mathbf{B}) - \beta_e \frac{\mathbf{J} \times \mathbf{B}}{B} + \beta_e \beta_i \frac{(\mathbf{J} \times \mathbf{B}) \times \mathbf{B}}{B^2} \tag{2.12}$$

$$\nabla \times \mathbf{B} = \mu_0 \mathbf{J} \tag{2.13}$$

$$\nabla \times \mathbf{E} = -\frac{\partial \mathbf{B}}{\partial t}. \tag{2.14}$$

where it have been introduced the Hall parameters β_e, βi for electrons and ions, given by:

$$\beta_e = \frac{\omega_e}{\nu_{en}} \tag{2.15}$$

$$\beta_i = \frac{\omega_i}{\nu_{in}} \tag{2.16}$$

where $\omega_{e,i}$ are the cyclotron frequencies for ions and electrons respectively, and $\nu_{en,in}$ are the collision frequencies between electrons, ions and the neutral particles.
Equations 2.9 to 2.11 are the Navier-Stokes equations with the added terms due to the electromagnetic interaction between the plasma conduction current $\mathbf{J}$ and the electromagnetic field. Equation 2.12 is known as the generalized Ohm's law, written for weakly ionized gases, and will be discussed in more detail in the next section. The last two equations are the Maxwell equations for the electric and magnetic fields. The electric field is determined by the conduction current $\mathbf{J}$ through the Ohm's law equation, since no charge density appears in the MHD equations. The magnetic field is the superimposition of the imposed and induced magnetic fields. Equation 2.13 determines the induced part only.
A last comment about equation 2.13: it implies that the conduction current is divergence free in the MHD approximation,

$$\nabla \cdot \mathbf{J} = 0 \tag{2.17}$$

For a complete treatment of the collisional plasmas theory, and of the magnetohydrodynamics formulation, the reader is referred, for example, to [31], [32], [33], [34], [35].

. . . .

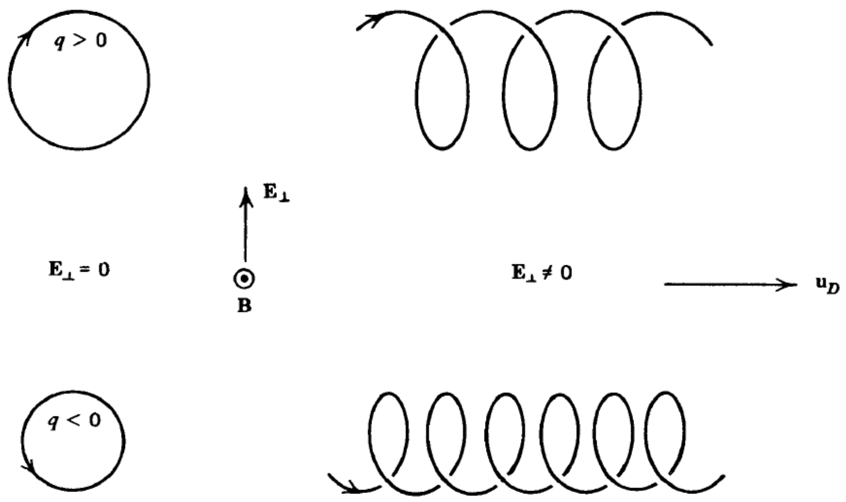


Figure 2.1: Charged particles trajectories in crossed electric and magnetic fields.

2.2 Diffusion in a magnetic field: anisotropy and the conductivity tensor.

In order to give some insight on the equation 2.12, let us consider the behavior of singly ionized positive ions and electrons, in a weakly ionized plasma.
In presence of a uniform electric field, charged particles move in a direction parallel or antiparallel to the field, depending on their charge. A conduction current in the field direction is observed.
In presence of both an uniform electric a magnetic field, charged particles, regardless of the charge sign, reach a drift velocity u_D in the direction $\mathbf{E} \times \mathbf{B}$:

$$u_D = \frac{\mathbf{E} \times \mathbf{B}}{B^2}. \tag{2.18}$$

The particle motion is shown in figure 2.1 for a z aligned magnetic field and an y aligned electric field [33]. Since both positive and negative charges have the same drift velocity, no conduction current would be observed.
The situations described above correspond to the limiting cases of a *strongly collision dominated*, and a *collisionless* plasma, respectively. In presence of many collisions, the motion of the particles around the magnetic field lines is often interrupted, and the bending effect of the field on the particle trajectories can be neglected, resulting in a motion along the electric field direction. The opposite condition maximises the action of the magnetic field, generating a drift motion in the direction $\mathbf{E} \times \mathbf{B}$. The relative incidence of collisions in presence of a magnetic field is described by the Hall parameter 2.15: the

ratio between the frequency of the gyroscopic motion and the collision frequency defines which of the behaviors described above has to be expected. For the intermediate cases, with $\mathbf{E} = E_y\hat{i}$ and $\mathbf{B} = B_z\hat{k}$, the following equations describe the electron motions:

$$U_{ex} = \frac{\beta_e\mu_e}{1+\beta_e^2}E_y \tag{2.19}$$

$$U_{ey} = -\frac{\mu_e}{1+\beta_e^2}E_y \tag{2.20}$$

where $\mu_e = e/m_e\nu_{en}$ is the electron mobility, related to β_e by:

$$\beta_e = \mu_e B. \tag{2.21}$$

As expected, electrons drift mainly in the electric field directions for low β_e, in the direction $\mathbf{E}\times\mathbf{B}$ for high Hall parameters. The electron conduction current is written

$$J_{ex} = -\frac{\sigma_e}{1+\beta_e^2}E_y = (\sigma_e)_\perp E_y \tag{2.22}$$

$$J_{ey} = \frac{\sigma_e\beta_e}{1+\beta_e^2}E_y = -(\sigma_e)_H E_y \tag{2.23}$$

where $\sigma_e = ne^2/m_e\nu_{en}$ is the scalar electron conductivity. Since $(\sigma_e)_H$ goes to zero for low β_e values, equations 2.19 show that for low Hall parameters the electron conductivity has a scalar nature. Equations analogue to 2.19 can be worked out for ions, obtaining:

$$U_{ix} = \frac{\beta_i\mu_i}{1+\beta_i^2}E_y \tag{2.24}$$

$$U_{iy} = \frac{\mu_i}{1+\beta_i^2}E_y \tag{2.25}$$

where $\mu_i = e/m_i\nu_{in}$ is the ion mobility, related to β_i by:

$$\beta_i = \mu_i B. \tag{2.26}$$

The ratio between the diffusion velocities components will determine the relative contribution of each species to the conduction current in the considered direction.

The total conduction current will be written as:

$$J_x = en(U_{ix} - U_{ex}) \sim -\frac{\sigma_e\beta_e}{(1+\beta_e\beta i)^2 + \beta_e^2}E_y = -(\sigma)_H E_y \tag{2.27}$$

$$J_y = en(U_{iy} - U_{ey}) \sim \frac{\sigma_e(1+\beta_e\beta i)}{(1+\beta_e\beta i)^2 + \beta_e^2}E_y = (\sigma)_\perp E_y \tag{2.28}$$

. . . .

where the total conductivity has been approximated with the electron conductivity. The module of the total conduction current J is given by

$$J = \frac{\sigma}{\sqrt{1+\beta_e^2}} E_y. \tag{2.29}$$

The current module is then maximum when the Hall parameter is minimum, *i.e.* when the charges drift in the $\mathbf{E} \times \mathbf{B}$ is negligible.
Equations 2.27 can be written for a general orientation of $\mathbf{E}$ in the vector form 2.12, or in a cartesian tensor form:

$$J_\alpha = \sigma_{\alpha\beta} E_\beta. \tag{2.30}$$

For $\beta i \beta_e \ll 1$ the conductivity tensor in 2.30 can be expressed as[1]:

$$\boldsymbol{\sigma} = \frac{\sigma}{1+\beta_e^2} \begin{pmatrix} 1+\beta_x^2 & \beta_x\beta_y - \beta_z & \beta_x\beta_z + \beta_y \\ \beta_x\beta_y + \beta_z & 1+\beta_y^2 & \beta_y\beta_z - \beta_x \\ \beta_x\beta_z - \beta_y & \beta_y\beta_z - \beta_x & 1+\beta_z^2 \end{pmatrix} \tag{2.31}$$

and 2.12 takes the form:

$$\mathbf{J} = \boldsymbol{\sigma}\mathbf{E} \tag{2.32}$$

In the reference frame moving with the flow, where the total electric field is written:

$$\mathbf{E}' = \mathbf{E} + \mathbf{u} \times \mathbf{B}, \tag{2.33}$$

we obtain:

$$\mathbf{J} = \boldsymbol{\sigma}\mathbf{E}' = \boldsymbol{\sigma}(\mathbf{E} + \mathbf{u} \times \mathbf{B}). \tag{2.34}$$

2.3 Low magnetic Reynolds numbers regime.

A description of the dynamic evolution of the magnetic field $\mathbf{B}$, can be obtained by substituting 2.13 and the generalized Ohm's law for scalar conductivity:

$$\mathbf{J} = \sigma(\mathbf{E} + \mathbf{u} \times \mathbf{B}), \tag{2.35}$$

in 2.14. The resulting equation is [33]:

$$\frac{\partial \mathbf{B}}{\partial t} = \frac{1}{\mu_0 \sigma} \nabla^2 \mathbf{B} + \nabla \times (\mathbf{u} \times \mathbf{B}) \tag{2.36}$$

The first term in the right hand side of equation 2.36 recalls a diffusion equation: for $\mathbf{u} = 0$ the field lines tend to diffuse in space and decay in time.

[1] See [36], [37]

The second term can be interpreted by noting that the flow variation through an area S whose bound is attached to flow streamlines is given by:

$$\frac{d\Phi}{dt} = \int\int_S \left[\frac{\partial \mathbf{B}}{\partial t} - \nabla \times (\mathbf{u} \times \mathbf{B})\right] \cdot d\mathbf{S}. \tag{2.37}$$

Equations 2.36 and 2.37 imply that for infinitely conducting fluid the variation of the flux of the magnetic field through the surface S would be null, that is, the magnetic field lines are convected with the fluid. The second term can therefore be interpreted as the magnetic field variation due to its convection with the fluid.

In the general case, the magnetic field exhibits both convection and diffusion. The ratio of the convection to the diffusion is expressed by the magnetic Reynolds number $\Re_m$:

$$\Re_m \equiv u_c L c/(\mu_0 \sigma_c)^{-1} = u_c L_c \sigma_c \mu_0. \tag{2.38}$$

where L_c is the characteristic length of the phenomena under study.

For the applications studied in this work, the magnetic Reynolds number is much less than unity, that is, the convection of the magnetic field lines is negligible. Furthermore, since the characteristic value of the induced magnetic field can be obtained from 2.35 and 2.13 as:

$$B_c^{ind} \sim L_c \mu_0 J_c = L_c \mu_0 \sigma_c u_c B_c \tag{2.39}$$

the magnetic Reynolds number can be interpreted as the ratio of induced to total magnetic induction.

The physical model adopted in this work, will therefore neglect the induced magnetic field and will consider the only imposed magnetic field. Since it is constant, the equation 2.14 implies that the electric field is irrotational, and a scalar potential can be defined:

$$\nabla \times \mathbf{E} = 0 \tag{2.40}$$

$$E = -\nabla \phi. \tag{2.41}$$

The generalized Ohm's law 2.34 can be written as:

$$\mathbf{J} = \boldsymbol{\sigma}(-\nabla\phi + \mathbf{u} \times \mathbf{B}). \tag{2.42}$$

that, with 2.40 and 2.17 completes the electrodynamic model.

. . . .

CHAPTER 3

ELECTRODYNAMICS: THE NUMERICAL MODEL.

If people reach perfection they vanish, you know.

T.H. White (The Once and Future King)

Abstract. The electrodynamic equation has been discretized by applying the finite element method, in the weighted residual formulation.
The finite element method consists in an expansion of the unknown solution of the differential problem on a basis of linear functions. The differential operator is evaluated for the approximated solution. By minimizing the integral error of this evaluation on the computational domain, the coefficients of the expansion are obtained as solution of a linear system $Ax = b$. The solution vector component are the nodal values of the unknown function. The key idea of the finite element method lies in the choice of the basis function, each of which is not null only in a cluster of elements around a node. The contribution of each element can be therefore computed separately and then the contributions add up to build the final solving system.
The chapter presents the overall procedure and the implementation of the technique for the electrodynamic case.

Introduction

The finite element method is widely applied to solve differential problems described by elliptic PDE equations. The weak solution obtained describes the physics of the problem with high accuracy, even if linear approximating shape functions are chosen. The model presented in this chapter applies the finite element method to the electrodynamic equations, obtaining the Hall potential generated by the interaction between the plasma flow and the magnetic field, the electric field and the current density generated into the plasma.

3.1 The finite element method: an overview

The finite element method (FEM) has been adopted in this work to solve the differential problem arising from the electrodynamic modeling. The key concepts of the FEM approach will be presented in this section, for a general self-adjoint problem in one dimension. The generalization to the multidimensional case and can be found, for example, in [38], [39], [40].
Let $\mathbf{L}$ be a self-adjoint differential operator on the function u, and consider the problem:

$$\mathbf{L}[u] = f(x) \qquad u \in [a, b] \tag{3.1}$$

with the boundary conditions:

$$\mathbf{B}(u) = 0 \qquad for \quad u = a \quad or \quad u = b \tag{3.2}$$

where $\mathbf{B}(u)$ is a differential operator on the domain boundary. The finite element method seeks for an approximate solution by building an integral formulation of 3.1:

$$(v, \mathbf{L}[u] - f) = 0 \tag{3.3}$$

with the *inner* product notation:

$$(u, v) := \int_a^b vudx. \tag{3.4}$$

Equation 3.2 has to hold for all square integrable *test* (or *weight*) functions v on the domain, that is for all $v \subset L^2$. The unknown functions u, v, can be substituted at this point by the approximating functions U, V:

$$(V, \mathbf{L}[U] - f) = 0 \tag{3.5}$$

U, V belong to the n-dimensional subspaces S^n, $\hat{S}^n$ of L^2 called respectively the *trial* space and the *test* space. By choosing a basis

$$N_j(x), M_j(x) \qquad j = 1, \cdots, n \tag{3.6}$$

. . . .

of both spaces, the functions U, V can be expressed as a linear combination of the basis functions:

$$U = \sum_{j=1}^{n} c_j N_j(x) \tag{3.7}$$

$$V = \sum_{j=1}^{n} d_j M_j(x) \tag{3.8}$$

The problem 3.5 becomes

$$\sum_{j=1}^{n} d_j(M_j, \mathbf{L}[U] - f) = 0 \qquad j = 1, \cdots, n. \tag{3.9}$$

Since 3.9 has to be satisfied for all it implies that

$$(M_j, \mathbf{L}[U] - f) = 0 \qquad j = 1, \cdots, n. \tag{3.10}$$

where appears the *residual* $\mathbf{L}[U] - f$, that names the method. By identifying the test space $\hat{S}^n$ with the trial space S^n, and choosing the same basis $N_j, j = 1, \cdots, n$ the *Galerkin*'s method is obtained:

$$(N_j, \mathbf{L}[U] - f) = 0 \qquad j = 1, \cdots, n. \tag{3.11}$$

that is:

$$\sum_{k=1}^{n} c_k(N_j, \mathbf{L}[N_k] - f) = 0 \qquad j = 1, \cdots, n. \tag{3.12}$$

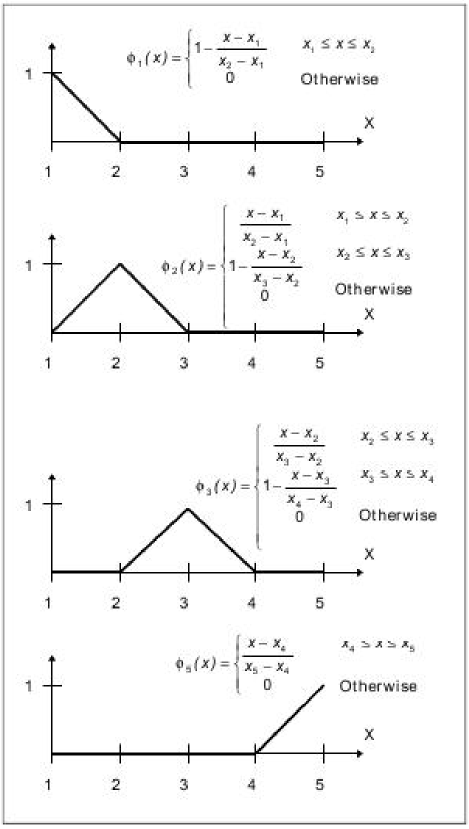
Equation 3.12 represents a linear algebraic system in the unknown values c_k. The success of the finite element method relies mainly on a convenient choice of the basis functions, that makes the integrals 3.12 easy to evaluate. In particular, if the basis functions N_j are non zero only on the elements containing the node j, the inner products 3.12 split into the contributions of the elements where N_j is not null. The most common example in one dimension is given by the so called *hat* functions, shown in figure 3.1. Equation 3.12 is integrated by part so as to reduce the derivative order on the trial functions, as in the procedure outlined in the next section.

3.2 The numerical model formulation

The numerical model applies the concept presented above to the electrodynamic equation 2.42. This section presents the tridimensional treatment but the same approach has been adopted for the two-dimensional axial symmetric formulation.

. . . .

Figure 3.1: Hat function in one dimension for a four element discretization of a one dimensional domain, as shown in [41]



The unknown potential $\phi(x, y, z)$ is piecewise approximated by using the shape functions $N_i(x, y, z)$, $i = 1, \cdots, n$:

$$\phi^*(x, y, z) = \{N^T(x, y, z)\}\{\phi\} \tag{3.13}$$

where ϕ^* represents the approximated potential, $\{N^T(x, y, z)\}$ is the shape functions' vector, and $\{\phi\}$ is the vector of the unknown nodal values. It implies that the electric field and the current density are approximated as:

$$\mathbf{E}(x, y, z)^* = [-\nabla N]\{\phi\} \tag{3.14}$$

$$\mathbf{J}(x, y, z)^* = \boldsymbol{\sigma}([-\nabla N]\{\phi\} + \mathbf{u} \times \mathbf{B}) \tag{3.15}$$

where u is the flow velocity, $\mathbf{B}$ is the magnetic flux density, and $\boldsymbol{\sigma}$ is the conductivity tensor. As in equation 3.12, the weighted residual of the continuity equation for the current density is set to zero:

$$\int_\Omega N\nabla \cdot \mathbf{J}^* d\Omega = 0 \tag{3.16}$$

Equation 3.16 is integrated by part, *i.e.* the divergence theorem is applied to the vector $N\mathbf{J}(x,y,z)^*$:

$$\int_\Omega N\nabla\cdot\mathbf{J}^* d\Omega = \int_{\partial\Omega} N J_n^* d\partial\Omega - \int_\Omega \nabla N\cdot\mathbf{J}^* d\Omega \tag{3.17}$$

By substituting 3.14 in 3.17 the final formulation is obtained:

$$-\int_\Omega \nabla N\cdot\boldsymbol{\sigma}(\mathbf{u}\times\mathbf{B})d\Omega - \int_{\partial\Omega} N J_n^* d\partial\Omega = -\int_\Omega \nabla N\cdot\boldsymbol{\sigma}[\nabla N]\{\phi_*\}d\Omega. \tag{3.18}$$

Equation 3.18 is an algebraic linear system in the unknown nodal values $\{\phi\}$ of the potential, since the integrals on the domain can be split on the shape functions supports Ω_i and evaluated, for example, by the Gauss quadrature rule. Consider the integrals:

$$\int_{\Omega_i} \nabla N_i\cdot\boldsymbol{\sigma}[\nabla N]\{\phi^*\}d\Omega_i \tag{3.19}$$

The support Ω_i is made up of *nel* elements containing the node i, therefore the integral in 3.19 can be written as the sum:

$$\int_{\Omega_i} \nabla N_i\cdot\boldsymbol{\sigma}[\nabla N]\{\phi^*\}d\Omega_i = \sum_{h=1}^{nel}\int_{\Omega_h} \nabla N_i\cdot\boldsymbol{\sigma}[\nabla N]\{\phi^*\}d\Omega_h \tag{3.20}$$

Consider tetrahedral elements with nodes i, j, k, m: the approximated potential 3.13 on the element is written as

$$\phi^* = N_i\phi_i + N_j\phi_j + N_k\phi_k + N_m\phi_m. \tag{3.21}$$

By substituting in 3.19, the following expression is obtained:

$$\begin{aligned}&\int_{\Omega_i} \nabla N_i\cdot\boldsymbol{\sigma}[\nabla N]\{\phi^*\}d\Omega_i = \\ &\sum_{h=1}^{nel}\int_{\Omega_h} \nabla N_i\cdot\boldsymbol{\sigma}[\nabla(N_i\phi_i + N_j\phi_j + N_k\phi_k + N_m\phi_m)]d(\Omega_h),\end{aligned} \tag{3.22}$$

that is:

$$\begin{aligned}&\int_{\Omega_i} \nabla N_i\cdot\boldsymbol{\sigma}[\nabla N]\{\phi^*\}d\Omega_i = \\ &\sum_{h=1}^{nel}[\int_{\Omega_h} \nabla N_i\cdot\boldsymbol{\sigma}[\nabla(N_i d(\Omega_h)]\phi_i + [\int_{\Omega_h} \nabla N_i\cdot\boldsymbol{\sigma}N_j d(\Omega_h)]\phi_j \\ &+ [\int_{\Omega_h} \nabla N_i\cdot\boldsymbol{\sigma}N_k d(\Omega_h)]\phi_k + [\int_{\Omega_h} \nabla N_i\cdot\boldsymbol{\sigma}N_m d(\Omega_h)]\phi_m.\end{aligned} \tag{3.23}$$

Equation 3.23 can be written as:

$$\begin{aligned}&\int_{\Omega_i} \nabla N_i \cdot \boldsymbol{\sigma}[\nabla N]\{\phi^*\} d\Omega_i = \\ &\sum_{h=1}^{nel} (a_{ii}\phi_i + a_{ij}\phi_j + a_{ik}\phi_k + a_{im}\phi_m).\end{aligned} \tag{3.24}$$

with:

$$a_{ii} = \int_{\Omega_h} (\nabla N_i \boldsymbol{\sigma} \nabla N_i) d\Omega_h \tag{3.25}$$

$$a_{ij} = \int_{\Omega_h} (\nabla N_i \boldsymbol{\sigma} \nabla N_j) d\Omega_h \tag{3.26}$$

$$a_{ik} = \int_{\Omega_h} (\nabla N_i \boldsymbol{\sigma} \nabla N_k) d\Omega_h \tag{3.27}$$

$$a_{im} = \int_{\Omega_h} (\nabla N_i \boldsymbol{\sigma} \nabla N_m) d\Omega_h \tag{3.28}$$

The element $(ijkm)$ contributes to the i-th equation with the coefficients a_{ii}, a_{ij}, a_{ik}, a_{im}, to the j-th equation with the coefficients a_{ji}, a_{jj}, a_{jk}, a_{jm}, and so on.
The contributions can be organized in a 4×4 matrix, called *element matrix*:

$$K_{ijkm} = \begin{pmatrix} a_{ii} & a_{ij} & a_{ik} & a_{im} \\ a_{ji} & a_{jj} & a_{jk} & a_{jm} \\ a_{ki} & a_{kj} & a_{kk} & a_{km} \\ a_{mi} & a_{mj} & a_{mk} & a_{mm} \end{pmatrix} \tag{3.29}$$

The coefficient $a_{\alpha\beta}$ of the element matrix K_{ijkm} adds up to the coefficient labeled $\alpha\beta$ in the global matrix, referred to as the *stiffness matrix*. The process of building up the stiffness matrix from the element matrices is called the *assembly* process. The RHS vector is built by the same procedure, and the final linear system is obtained.
The solution gives the nodal values for the potential, and from 3.14 the values of the electric field and the current density can be computed. The solution process is analyzed in chapter 4.

3.3 The numerical model implementation

The implementing code has been developed partly in FORTRAN 90, and partly in FORTRAN 77 for both the two and three dimensional cases. Figure 3.2 shows the code structure for the three dimensional case. The main modules' features are listed below:

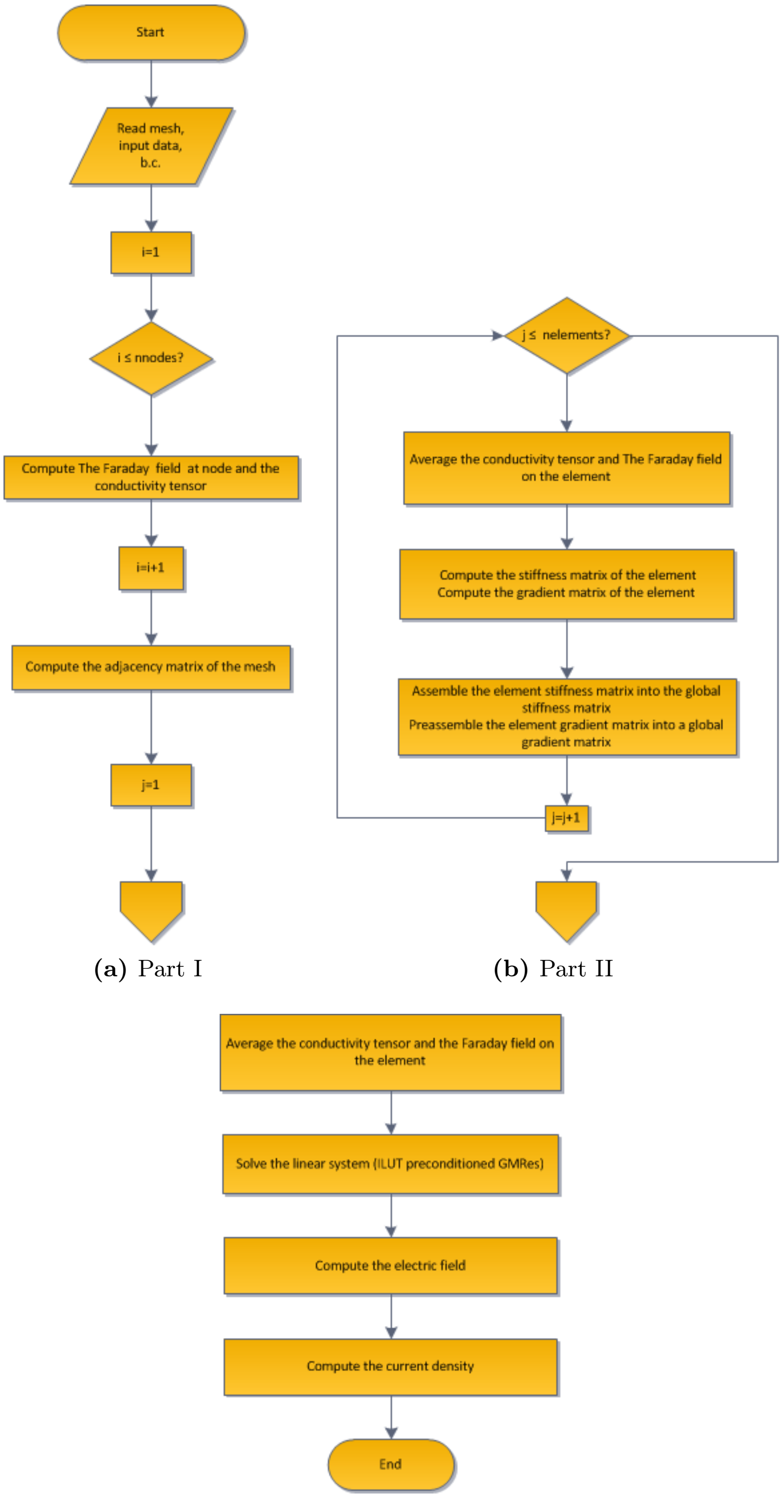
. . . .

- type definitions and variable declarations,
- declarations of the workspace variable for the iterative solver,
- reading of the mesh data, of the input physical quantities and boundary conditions,
- stiffness matrix computation,
- conductivity tensor computation at each node,
- Faraday field computation at each node,
- computation of mesh adjacency matrix,
- computation of the element stiffness matrix and R.H.S.,
- average of the conductivity tensor and Faraday field on the element,
- imposition of boundary conditions,
- assembling the element stiffness matrices and the R.H.S.,
- preconditioning,
- solution of the linear system to obtain electric potential, electric field and current density,
- computation of the electric field and of the current density.

The input-output structure of the code is shown in table 3.1.

Table 3.1: Input output structure for the electrodynamic code.

Code segment	**Quantities**	**Specifications**
Mesh	Nodes	(x, y, z) coordinates of the mesh nodes
	Faces of Tetrahedra	nodes indexes, area
	Tetrahedra	nodes indexes, volume
Input Physical quantities	Velocity field	
	Electrons mobility	Given by the kinetic model
	Electrons density	
	Magnetic flux density	
Boundary conditions	Current density normal to the boundary	Neumann boundary condition
	Electric potential on the boundary	Dirichlet boundary condition
Output Physical quantities	Electric potential	
	Electric field **E**	$\mathbf{J} \times \mathbf{E}$, source term for the fluid dynamic momentum equation
	Current density **J**	$\mathbf{J} \cdot \mathbf{E}$, source term for the fluid dynamic energy equation.


Start
Read mesh, input data, b.c.
i=1
i ≤ nnodes?
Compute The Faraday field at node and the conductivity tensor
i=i+1
Compute the adjacency matrix of the mesh
j=1
j ≤ nelements?
Average the conductivity tensor and The Faraday field on the element
Compute the stiffness matrix of the element
Compute the gradient matrix of the element
Assemble the element stiffness matrix into the global stiffness matrix
Preassemble the element gradient matrix into a global gradient matrix
j=j+1
Average the conductivity tensor and the Faraday field on the element
Solve the linear system (ILUT preconditioned GMRes)
Compute the electric field
Compute the current density
End

(a) Part I

(b) Part II

(c) Part III

Figure 3.2: Flowchart for the tridimensional electrodynamic code.

CHAPTER 4

COMPUTATIONAL METHODS

What is the use of a book, without pictures or conversations?

Lewis Carroll.

Abstract. This chapter presents the numerical techniques adopted to obtain the finite element solution of the electrodynamic equation.

The peculiar structure of the conductivity matrix generates a non-symmetric, highly anisotropic stiffness matrix. The associated system is solved by a GMRes iterative algorithm, but the convergence behavior of the solver is not optimal if a standard preconditioning is applied. These difficulties are linked to the physics of the MHD phenomenon under study, and have been overcome by developing a reordering method that takes into accounts the preferred directions identified by the physical configuration.

The accuracy of the field evaluation has been increased by implementing an error recovery technique. Finally, the implemented code has been analyzed to evaluate the time costs and the memory usage efficiency. The analysis described has been carried out for the two dimensional code. Nevertheless, the methods applied can be extended to the three dimensional case.

. . . .

Introduction

The finite element method applied to the anisotropic physical configuration described in the previous chapters, generates a large sparse linear system to be iteratively solved. The anisotropy of the conductivity tensor results in a non symmetric, highly anisotropic stiffness matrix, that generates several computational difficulties. A particular attention has therefore to be paid in the choice and in the optimization of the solution strategy. This chapter outlines the numerical techniques adopted to obtain the solution, and presents the methods developed and implemented to overcome the numerical difficulties and improve the quality of the approximation.

4.1 Linear solvers and the GMRes Algorithm

Several strategies can be applied to obtain the solution of the linear system $Ax = b$, where A is a sparse[1] $n \times n$ matrix. For finite element applications, when the number n ranges from thousands to millions, an iterative approach has to be adopted. It basically consists in identifying an iteration operator G such that:

$$x_{k+1} = Gx_k + f. \tag{4.1}$$

The limit of this sequence, if the limit exists, is the solution x:

$$x = Gx + f. \tag{4.2}$$

It can be seen, by expanding x on a basis of eigenvectors of G, that the existence of the limit is guaranteed if the spectral radius $\rho(G)$ of the matrix is less than one:

$$\rho = \max_i \{\lambda_i\} \tag{4.3}$$

$$\rho < 1 \tag{4.4}$$

where λ_i are the matrix eigenvalues.
A widely used class of iterative linear solvers are referred to as the *projection methods*, because they seek for the solution vector x by projecting it onto a subspace K of $\mathbb{R}_n$. The residual $r_k = b - Ax_k$ of the projected vector x_k is required to be orthogonal to another subspace L of $\mathbb{R}_n$. Projection methods generalize the idea that, starting from an initial guess, the solution point can be reached by taking steps in mutually orthogonal directions (i.e. by traveling along the directions defined by an orthogonal basis of the search subspace K). The approximation at the step k is defined as a linear combination of the previous search directions (i.e., the approximation at the

[1] A sparse matrix is a matrix primarly populated with zeroes.

. . . .

step k is defined as a vector of the subspace spanned by the traveled search directions). The distance to be covered in each step is defined by a residual minimization condition, corresponding to an orthogonality requirement (e.g., a directional derivative minimization requires the orthogonality between the gradient vector and the search direction itself [42], [43]).
In this approach, the G operator is defined by the choice of the two spaces K and L.
A popular choice for the space K is the so-called Krilov subspace generated by the initial residual $r_0 = b - Ax_0$. The m-th Krilov subspace K_m is defined as:

$$K_m(A, r_0) \equiv span\{r_0, Ar_0, A^2 r_0, \dots, A^{m-1} r_0\} \tag{4.5}$$

For general non-symmetric matrices, one of the most effective projection methods currently available is the GMRes, first proposed by Saad and Schultz in 1986 [44]. GMRes builds an orthogonal basis of the space K_m, and minimizes the euclidean norm of the residual over all the vectors in that space. Step by step, the dimension of the approximating Krilov subspace increases and the accuracy of the solution improves. The basis is built up by the Arnoldi procedure, that consists in a step by step Gram-Schmidt orthogonalization of the set $A^i r_0$. Its formulation is given, for example, in Saad [45] and is reported below:

Algorithm 4.1 Arnoldi procedure

```
Choose a vector $v_1$ of norm 1
for $j = 1, 2, ..., m$ do
    Compute $h_{ij} = (Av_j, v_i)$ for $i = 1, 2, ..., j$
    Compute $w_j \leftarrow Av_j - \sum_{i=1}^{j} h_{i,j} v_i$
    $h_{j+1,j} = ||w_j||_2$
    if $h_{j+1,j} = 0$ then
        Stop
    end if
    $v_{j+1} = w_j / h_{j+1,j}$
end for
```

The algorithm 4.1 can be written in matrix form, by introducing the $(m+1) \times m$ Hessenberg matrix $\overline{H}_m$:

$$\overline{H}_m = \begin{pmatrix} h_{1,1} & h_{1,2} & h_{1,3} & \dots & h_{1,m} \\ h_{2,1} & h_{2,2} & h_{2,3} & \dots & h_{2,m} \\ 0 & h_{3,2} & h_{3,3} & \dots & h_{3,m} \\ \vdots & \ddots & \ddots & \ddots & \vdots \\ 0 & 0 & 0 & h_{m,(m-1)} & h_{m,m} \\ 0 & 0 & 0 & & h_{(m+1),m} \end{pmatrix} \tag{4.6}$$

. . . .

The matrix entries are defined by the lines 3, 5 in the algorithm 4.1. Each column j contains the projection of the vector Av_j on the previous vectors of the basis. The $j+1th$ element in the column is the norm of the $j+1th$ basis vector. Thus, in matrix form, lines 3, 5 and 9 of 4.1 can be rewritten as

$$AV_m = V_{m+1}\overline{H}_m \tag{4.7}$$

where V_m is the matrix whose column are the vectors $v_j, j = 1, \ldots, m$.
The algorithm breaks down when the computed Krilov subspace K_j is invariant under A. In this case, the approximation computed from any projection method on the subspace K_j is exact [45]. Once a basis has been found, GMRes expresses a generic vector of the approximating space by its expansion on the basis:

$$x = x_0 + V_m y \tag{4.8}$$

where y is the components' vector and x_0 is the initial guess to the solution. By posing $v_1 = r_0/||r_0||_2 = r_0/\beta$ the residual can be written as

$$b - A(x_0 + V_m y) = \tag{4.9}$$
$$= r_0 - AV_m y \tag{4.10}$$
$$= \beta v_1 - V_{m+1}\overline{H}_m y \tag{4.11}$$
$$= V_{m+1}(\beta e_1 - \overline{H}_m y) \tag{4.12}$$

where e_1 is the first vector of the canonic basis of $\mathbb{R}_m$. The residual norm is then computed as:

$$J(y) \equiv ||b - A(x_0 + V_m y)||_2 = ||\beta e_1 - \overline{H}_m y||_2 \tag{4.13}$$

The GMRes approximation at step m is the vector

$$x_m = x_0 + V_m y_m \tag{4.14}$$

where y_m minimizes the norm 4.13.
This minimization condition is equivalent to setting the subspace L equal to AK_m. Since the new search subspace is obtained as the orthogonalization of AK_m, this condition asks for the approximation at step $m-1$ to be orthogonal to the approximation at step m.
After n iterations, Gmres builds up an exact approximation for $\mathbb{R}_n$. Since n in practice is high, Gmres is stopped when the residual is reduced to a certain fraction of the initial residual, i.e. when

$$||r^k||_2 \leq \epsilon ||r^0||_2 \tag{4.15}$$

The quality of the approximation at the $m-th$ step is linked to the spectrum of the matrix A. Convergence is fast if eigenvalues of A are clustered away from the origin, see Fig. 4.1, and if the matrix is near normal:

. . . .

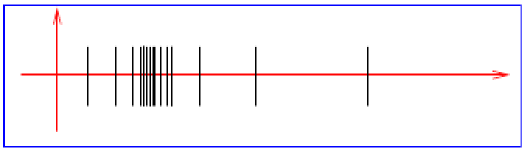


Figure 4.1: Optimal distributions of the eigenvalues for the iterative solution of the system.

$$AA^{+} \simeq A^{+}A \tag{4.16}$$

where A^{+} is the transpose conjugate of A. For further details, the convergence behavior of GMRes is discussed, for example in [44],[45].

4.2 Preconditioning and ILU-type algorithms.

The convergence of any iterative method solving the linear system $Ax = b$ depends on the distribution of the eigenvalues of the matrix A. A straightforward way to improve the convergence is therefore to solve an equivalent system characterized by a well conditioned matrix, *i.e.* by a matrix with an optimal distribution of the eigenvalues. It can be achieved by multiplying both sides of the equation by a matrix M^{-1}:

$$M^{-1}Ax = M^{-1}b \tag{4.17}$$

If the matrix $M^{-1}A$ has a better eigenvalue distribution than the original matrix A, the system can be more easily solved.
In practice, the matrix $M^{-1}A$ needs not to be formed explicitly, because the preconditioning operation is included in the solver's algorithm, as can be seen in algorithm 4.2. It is apparent from 4.2 that the matrix M^{-1} needs only to be multiplied by a vector, and this operation should not be too expensive to perform in terms of time costs.
By choosing $M \equiv A$ the preconditioning becomes the explicit inversion of the system. It is obviously meaningless to explicit compute the inverse of A, but it's possible to choose M as an *incomplete factorization* of A:

$$M = LU \tag{4.18}$$

$$A = LU - R \tag{4.19}$$

where the L, U matrices approximate a factorization of the matrix A, and R is the residual of the approximate factorization.
It's known that for non-singular matrices, it is possible to build an exact factorization into a lower (L) and upper (U) triangular factors by applying the Gaussian elimination procedure reported in algorithm 4.3 (for further

Algorithm 4.2 Right preconditioned GMRes

```
 1: Choose a vector $x_0$ and a dimension $m$
 2: Compute $r_0 = Ax_0 - b$, $\beta = ||r_0||_2$, $v_1 = r_0/\beta$
 3: Arnoldi
 4: for $j = 1, 2, ..., m$ do
 5:    $z_j \leftarrow M^{-1}v_j$
 6:    $w \leftarrow Az_j$
 7:    for $i = 1, 2, ..., j$ do
 8:       $h_{ij} \leftarrow (w, v_i)$
 9:       $w \leftarrow w - h_{i,j}v_i$
10:    end for
11:    $h_{j+1,j} = ||w||_2$
12:    $v_{j+1} = w/h_{j+1,j}$
13: end for
14: Define $V_m = [v_1, \dots, v_m]$ and $\overline{H}_m = \{h_{i,j}\}$.
15: Form the approximate solution
16: $x_m = x_0 + M^{-1}V_m y_m$
17: where $y_m = argmin_y||\beta e_1 - \overline{H}_m y||_2$, $e_1 = [1, 0, \dots, 0]^T$
18: Restart
19: If satisfied stop, else set $x_0 \leftarrow x_m$ and goto 2
```

details, see, for example, [46],[47],[48]).

The main issue of this procedure is that the L, U factors are generally not sparse even if the matrix is. It is therefore necessary to define a method to limit the number of non zero entries in the LU factors, *i.e* to reduce the number of the *fill-ins* generated in the factorization process. Such procedures are referred to as *Incomplete LU* (ILU) factorizations [49], [45]. Clearly, the cruder the dropping strategy, the poorer the improvement in performances and robustness obtained for the iterative solver. On the other hand, the cruder the dropping strategy, the lower the time and memory cost of the preconditioners. It is usually preferable, anyway, to choose a good factorization, because it guarantees an increased stability of the iterative solver.

Different ILU-based preconditioners are defined by different dropping strategies they adopt to obtain the approximated sparse L, U factors. We outline here the ILU(p) and ILUT methods, and refer the reader to [45], [50], [51] for a complete treatment.

Define the non-zero pattern of A as the set :

$$NZ(A) = \{(i, j), 1 \leq i, j \leq n | a_{i,j} \neq 0\} \tag{4.20}$$

The ILU(0) method performs the L, U factorization and drops all the non-zero terms outside the non-zero pattern of A. During the Gauss elimination, therefore, every non null coefficient in a position where A has a zero entry is neglected. Figure 4.2 shows a sketch of the non null entries distibution for an

Algorithm 4.3 Gaussian Elimination

```
1: for $i = 2, \dots, n$ do
2:    for $k = 1, \dots, i-1$ do
3:       $a_{ik} \leftarrow a_{ik}/a_{kk}$
4:       for $j = k+1, \dots, n$ do
5:          $a_{ij} \leftarrow a_{ij} - a_{ik}a_{kj}$
6:       end for
7:    end for
8: end for
```

example sparse matrix B, its exact L, U factors and the ILU(0) approximated ones.

ILU(0) produces obviously a rough approximation of the L, U factors of A, as for general matrices they would be dense. In order to improve the approximation quality, it has been introduced the concept of *level of fill*.

The *level of fill* gives an order of magnitude estimate of the size of the non zero entries generated in the Gaussian elimination process: the higher the level, the smaller the element. In order to match this definition, a *level of fill* is attributed to each element in a Gaussian elimination process, and initially set to:

$$lev(a_{ij}) = 0 \qquad if \quad a_{ij} \neq 0, \qquad or \quad i = j \tag{4.21}$$

$$lev(a_{ij}) = \infty \qquad elsewhere \tag{4.22}$$

During the process, the *level of fill* is updated each time the element is modified in row 5 of algorithm 4.3:

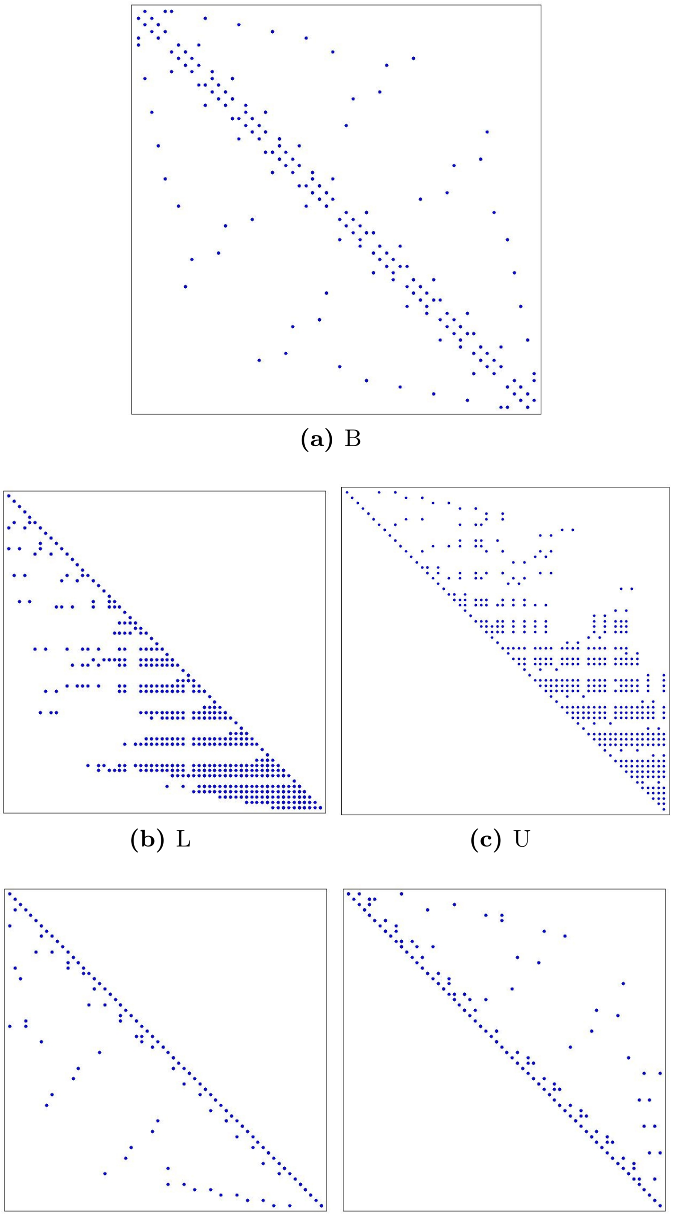
$$lev(a_{ij}) = min\{lev(a_{ij}), lev(a_{ik}) + lev(a_{kj}) + 1\} \tag{4.23}$$

Equation (4.23) shows that the *level of fill* never increases during the elimination: a non null entry in a position belonging to $NZ(A)$ mantains a zero *level of fill* throughout the process. At the end of the procedure 4.3 all the terms having a *level of fill* higher than a certain threshold p, are discarded.

The strategy outlined above is based on the assumption that the size of a matrix element with level k can be estimated by ϵ^k, $\epsilon \leq 1$. This is verified by, for example, diagonally dominant matrices processed in algorithm 4.3, since each element is divided by the corresponding diagonal term, but it is not true in the general case.

As the threshold increases, a greater number of element is retained, and better approximations are obtained. The $ILU(p)$ preconditioner algorithm is reported in 4.4. The dropping strategy adopted in algorithm 4.4 is based on topologic considerations and on the poor order of magnitude estimate of the elements' module outlined above.

No explicit evaluation of the element module is done during the procedure,



(a) B

(b) L

(c) U

(d) IL

(e) IU

Figure 4.2: The matrix B and the exact and ILU(0) approximated L U factors.

and it can generate low quality approximations. In order to overcome this issue, it can be defined a new algorithm, that takes into account the actual module of the discarded elements. It can been achieved by setting a threshold on the module values. In the ILU-type computational algorithm,

Algorithm 4.4 ILU(p)algorithm

```
For all $a_{ij} \neq 0$ set $lev(a_{ij}) = 0$
for $i = 2, \ldots, n$ do
  if $lev(a_{ik}) \leq p$ then
    for $k = 1, \ldots, i-1$ do
      $a_{ik} \leftarrow a_{ik}/a_{kk}$
      for $j = 1, \ldots, n$ do
        $a_{ij} \leftarrow a_{ij} - a_{ik}a_{kj}$
        Update level of fill of non zero $a_{ij}$
      end for
    end for
  end if
  Replace any element in row $i$ with level $a_{ij} > p$ by zero.
end for
```

the L, U factors are stored in the same matrix. If there is a row in this matrix populated by high values elements, too many elements would be retained if the only threshold-based dropping strategy is applied.
In order to control this behaviour, it is possible to set the maximum number r of elements to be retained in each row: the r largest elements in the L and U parts of the row are preserved.
The above consideration outline the ILUT algorithm reported in 4.5 (see [45]).

Algorithm 4.5 ILUT algorithm

```
 1: for $i = 1, \ldots, n$ do
 2:   $w \leftarrow a_{i*}$
 3:   for $k = 1, \ldots, i-1$ do
 4:     if $w_k \neq 0$ then
 5:       $w_k \leftarrow w_k/a_{kk}$
 6:       Apply a dropping rule to $w_k$
 7:       if $w_k \neq 0$ then
 8:         $w \leftarrow w - w_k * u_{k*}$
 9:       end if
10:     end if
11:   end for
12:   Apply a dropping rule to row $w$
13:   $l_{ij} \leftarrow w_j$ for $j = 1, \ldots, i-1$
14:   $u_{ij} \leftarrow w_j$ for $j = i, \ldots, n$
15:   $w \leftarrow 0$
16: end for
```

The algorithm is defined by the dropping rules in lines 5, 10. The first dropping process is based on the element values:

1. The user defined threshold τ is weighted by the current row 2-norm: $\tau_i = \tau * ||w||_2$,

2. All the row element smaller than τ_i are set to 0.

In line 10, the dropping process of line 5 is repeated. Then, the p largest elements in the L, U part of the row are retained, and the others are set to 0.

It can be noted that if it is possible to arrange the A matrix coefficients so as to minimize the number and the size of the *fill-in*s generated in the Gauss procedure, a softer dropping strategy can be applied, obtaining a better L, U factors approximation. The following sections will outline a method to obtain this result.

4.3 Reordering methods and preconditioning.

The mesh adopted for the finite element discretization can be viewed as a graph, *i.e.* a collection of connected nodes. The adjacency matrix of a graph is defined by:

$$m_{ij} = 1 \qquad if \quad nodes \quad i, j \quad are \quad connected \tag{4.24}$$

$$m_{ij} = 0 \qquad otherwise \tag{4.25}$$

The matrix generated in the finite element discretization has the same coefficients distribution as the adjacency matrix of the discretizing mesh. If the labeling of the mesh nodes is changed, the position of the non null elements of M is changed according to the new labels. An optimal choice of the labels would therefore result in an optimal distribution of the non null entries. This optimization is treated in subsection 4.3.2.

If the discretized physical problem shows a preferred direction, an appropriate choice of the nodes' numbering direction would produce an optimal distribution of non null coefficients' size. This is treated in 4.3.3.

4.3.1 Graph theory nomenclature.

A graph is a collection of nodes connected by arcs. The following summary outlines some notions and notations about undirected graphs[2]:

- If V is a finite non-empty set and E is a collection of unordered pairs (v_i, v_j) of V, then $G(V, E)$ is a graph.

[2] An undirected graph is a graph whose nodes are connected by undirected arcs. This definition excludes self connected nodes, since the set E cannot contain duplicates.

- The elements of $V = V(G)$ and $E = E(G)$ are called vertices and edges respectively.
- If $\{v_i, v_j\} \in E$, *i.e.* if v_i, v_j are connected by an edge, then v_i, v_j are said to be *adjacent.*
- The *degree* of a vertex is the number of vertices adjacent to him.
- A *path* in a graph is a sequence of connected nodes.
- A level structure, $L(G)$, of a graph G is a partition of the set $V(G)$ into levels $L_1, L_2, \ldots, L_k$ such that:
  1. All vertices adjacent to vertices in level L_1 are either in L_1 or in L_2,
  2. All vertices adjacent to level L_k are in either level L_k or L_{k-1},
  3. For $1 < i < k$, all vertices adjacent to level L_i are in either level L_i or L_{i-1} or L_{i+1}.

Graphs can be used to represent the structure of a matrix, such a representation being particularly useful for sparse matrices analysis. Some notions are given in the following:

- Given a $N \times N$ matrix A, we can define a graph $G(V, E)$ where V has n elements $\{v_1, \ldots, v_n\}$ representing the rows in A, and $\{v_i, v_j\} \in E$ if $a_{ij} \neq 0$
- The bandwidth of a matrix A is the maximum distance of a non null element from the main diagonal:

$$b(A) = \max\{|i - j|\, a_{ij} \neq 0\} \tag{4.26}$$

- A matrix is said to be an M-matrix if
  1. $a_{i,i} > 0$ for $i = 1, \ldots, n$
  2. $a_{i,j} \leq 0$ for $i \neq j \;\; i, j = 1, \ldots, n$
  3. A is non singular
  4. A^{-1} is non-negative
- The *envelope* (or *profile*) of A contains the elements of A to the right of the first non null element of each row, and to the left of the main diagonal:

$$Env(A) \equiv \{(i, j), j \leq i \quad and \quad \exists k \geq i \quad | \quad a_{kj} \neq 0\} \tag{4.27}$$

For further details, the reader is referred to [52], [53].

. . . .

Algorithm 4.6 Cuthill-MCKee algorithm

```
Select a starting node $v_r$ of minimal degree
Define a permutation vector $p$
Define a vector $q \leftarrow 0$
$p_1 \leftarrow r$
for $i = 1, \ldots, n$ do
    for $j = 1, \ldots, n$ do
        $l = 1$
        if $\{v_i, v_j\} \in E$ then
            $q_l \leftarrow j$
            $l = l + 1$
        end if
        Reorder $q$ by increasing degree of $v_{q_j}, j = 1, \ldots, l$
        for $m = 1, \ldots, l$ do
            $step = 0$
            if $p_k \leq j \quad \forall k = 1, \ldots, n$ then
                $p_{(i+step)} \leftarrow j$
                $step \leftarrow step + 1$
            end if
        end for
    end for
end for
```

4.3.2 The Reverse Cuthill McKee Algorithm

The bandwidth of a sparse matrix A can be reduced by labelling the connected nodes with near indexes. The first proposed strategy to obtain such a reduction is due to E. Cuthill and J. McKee [54], and reported below:

1. Select a starting node and relabel it as 1.

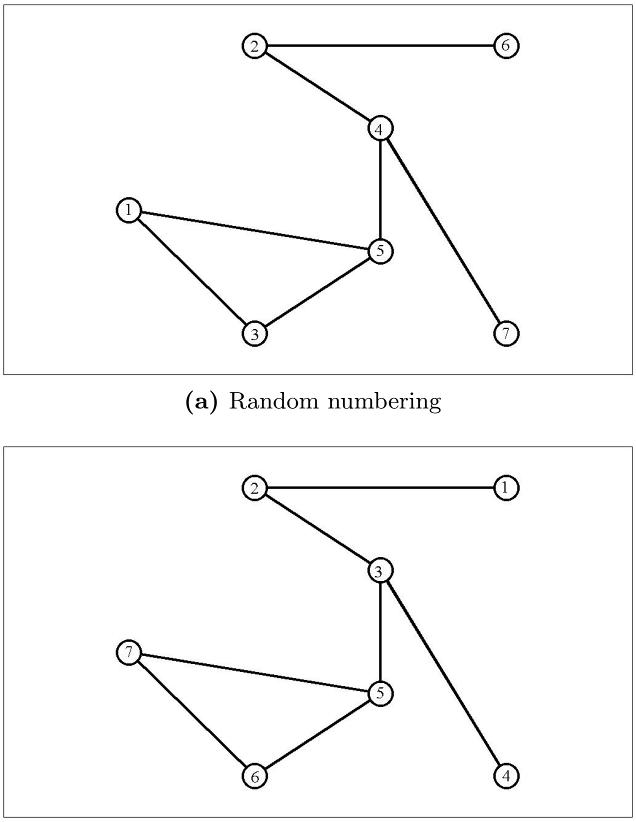
2. The nodes adjacent to this node are numbered in sequence beginning with 2 in the order of their increasing degree. These nodes are at distance 1 from node 2. They are said to be at the second level.

3. These procedure is repeated for each node in second level in sequence, *i.e.* first for node 2, then for node 3, then for node 4, *etc.*

4. The above procedure is repeated for the nodes at each successive level.

An example of Cuthill McKee (CM) renumbering is shown in Fig. 4.3, and an implementing algorithm is outlined in 4.6.

Once selected a starting node, the CM algorithm performs a breadth first traversal (referred to as *Breadth First Search*, BFS) of the graph, labeling the neighboring nodes by increasing degree. This procedure minimizes the



(a) Random numbering

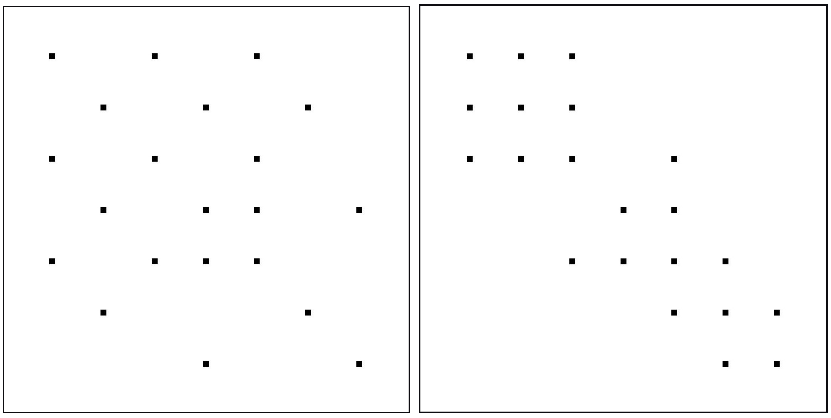
(b) CM Numbering

Figure 4.3: Cuthill-McKee renumbering of an example undirected graph.

label distance between connected nodes and generates a vertices partitioning, where vertices with the same depth in BFS are on the same level.
The relevance of this technique in computations is linked to the effect it has on the adjacency matrix of the graph, that is, on the stiffness matrix generated in a finite element discretization, where non null elements arise in positions labeled by connected nodes. Figure 4.4 shows the adjacency matrices for the graphs of Fig. 4.3. It is apparent that the reordering results in a bandwidth and profile size reduction of the matrix.

In 1971, A. George proposed to reverse the CM numbering, achieving in most cases a further profile reduction of the matrix [3] [56]. The resulting algorithm is referred to as Reverse Cuthill McKee (RCM).
The computational advantages obtained by applying a reordering stem from the effects of a reduced profile on the incomplete factorization [57], [52], [58], [59], [60].

[3] It has been demonstrated by W.H. Liu, and A.H. Sherman in 1976 [55].



(a) Random numbering

(b) RCM Numbering

Figure 4.4: Reverse Cuthill-McKee renumbering effect on an example sparse matrix.

Recall the incomplete elimination step of algorithm 4.3. The j loop applies to all row elements to the right of the first non null element in each row. To give an order of magnitude estimate of the number of non null terms introduced in the L, U factors by the Gauss procedure (*fill-ins*), consider a worst case arrangement of the non-null coefficients in A, for a given bandwidth b. Assume that the only non null out-diagonal element in each row i, with $i \leq (b+1)$ is in the first column, whereas for $i > (b+1)$ it is labeled $(b+j, j)$. If no cancellation is taken into account, the number of *fill-ins* introduced satisfies:

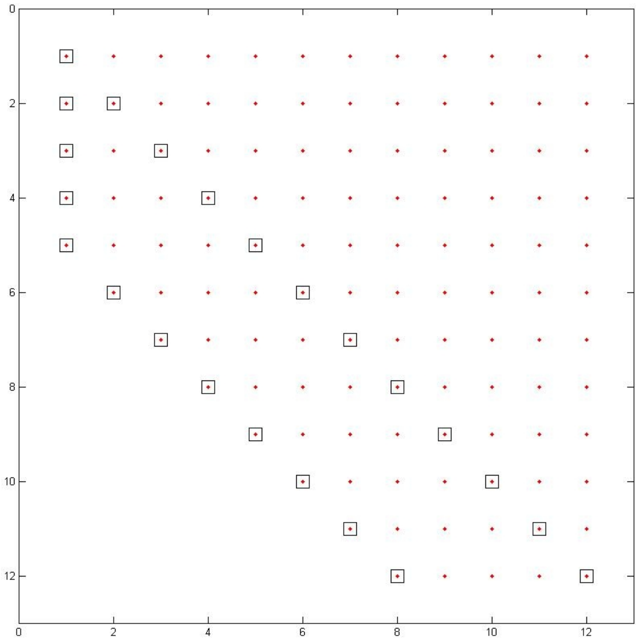
$$N_{fill} \leq \frac{N(N-1)}{2} + (b-1)(N-b) + \frac{(b-2)(b-1)}{2}. \tag{4.28}$$

Since $b < N$, 4.28 shows that a reduction of the bandwidth leads to a reduction of the *fill-ins* generated by the Gauss procedure. In other words, an optimal ordering generates the best elimination sequence for the matrix A, *i.e.* the one that introduces the lower number of new non null elements [58] [61]. The matrix A for the above example and the non null entries introduced by Gauss elimination are shown in Fig. 4.5.

A last comment about the starting node for the renumbering. An optimal choice maximizes the depth of the level structure, and minimizes its width. The first strategy aimed at this scope, suggested by Cuthill and McKee, simply picks up a node with the lowest number of connections, *i.e.* a node of minimal degree. A further step is due to Gibbs, Poole and Stockmeyer, [53], who applied the concepts of *peripheral* and *semiperipheral* node. A *peripheral* node is a node that generates a level structure of maximal depth in the graph, *i.e.* it has the maximum distance by another node in the graph.

....

Figure 4.5: Non null entries distribution (square shaped markers), and *fill-ins* (dots) in Gauss procedure for the example in text.



Algorithm 4.7 Pseudo-peripheral node

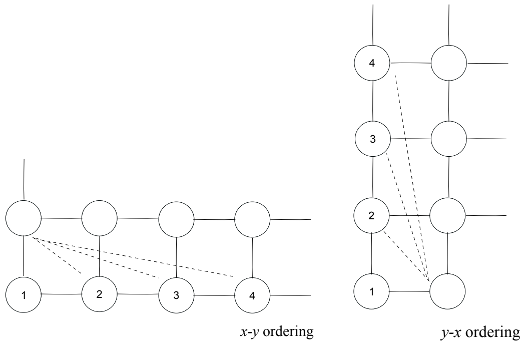
```
Select a starting node $x \leftarrow v_r$ of minimum degree
Generate the level structure $L_1, L_2, \ldots, L_s$ rooted at $x$
for $u_k \in L_s$, of minimum degree do
    Generate the level structure $M_1, M_2, \ldots, M_t$ rooted at $u_k$
    if $k > r$ then
        $x \leftarrow u_k$
    end if
end for
Reorder $x$ is a *semiperipheral* node
```

Finding such a node would imply a search amongst all the minimal degree nodes of the graph, and it obviously would not be cost effective. Gibbs, Poole and Stockmeyer suggested instead to apply the heuristic 4.7 to find a semiperipheral node. It is not guaranteed the algorithm selects a *peripheral* node, but it most likely will find a “boundary node”, with a deep level structure.

Algorithm 4.7 produces a further reduction in the profile of the stiffness matrix.

. . . .

Figure 4.6: Natural $x-y$ and $y-x$ orderings for a discretizing mesh.



For the mesh configurations studied in this work, anyway, where boundary nodes are usually nodes of minimal degree, the procedure 4.7 can be omitted, and good results are obtained by selecting a minimal degree starting node.

4.3.3 The weighted reverse Cuthill McKee: algorithm.

The RCM ordering is based on the mesh topology, and has no preferred spatial direction. The physical problem studied in this work, on the contrary, has a preferred spatial direction defined by the imposed magnetic field.
A work by Clift and Tang [62] suggests that the proper strategy to introduce a direction sensitivity into the RCM algorithm can be derived by analyzing the discretization of the 2D anisotropic PDE:

$$\frac{\partial}{\partial x}(K\frac{\partial \phi}{\partial x}) + \frac{\partial}{\partial y}(\frac{\partial \phi}{\partial x}) \tag{4.29}$$

on a rectangular grid, with K =1000.
Clift and Tang [62] provide a comparison between the performances of an iterative solver applied to the problem 4.29 for an $x-y$ and $y-x$ ordering of the mesh nodes (see Fig. 4.6). A remarkable improvement is obtained when using the second.

In the cited paper, the Authors note that the anisotropy of the equation results in a different size for the stiffness matrix entries, depending on the direction of the nodes connections: connections aligned in the x directions have values $O(K/K+1)$, whereas connections along the y direction have values $O(1/K+1)$, as can be easily seen by discretizing the equation (4.29) and normalizing the resulting coefficients by the diagonal element.
The values of the *fill-ins* entries in the L factors are related to the matrix coefficients [62], [63], [64]:

Theorem 4.3.1. *Let A be an M-matrix and let $v_{i_1}, \ldots, v_{i_m}$ be a path in the* *graph of A where*

$$v_{i_1}, \ldots, v_{i_m} \in \{v_{i_1}, \ldots, v_{i_{j-1}}\}, \tag{4.30}$$

then for $i > j$:

$$l_{ij} \geq \frac{a_{ii_1} a_{i_1 i_2} \ldots a_{i_m i_j}}{d_{i_1} d_{i_2} \ldots d_{i_m} d_{i_j}}, \quad d_k = a_{kk} \tag{4.31}$$

The theorem shows that for the case of equation (4.29) connections oriented in the y direction tend to generate smaller terms in the L factor.
Equation 4.29 is equation 2.42 written for a uniform **B** field along the y direction, and a constant flow velocity perpendicular to it, along the x direction.
The conductivity tensor for these conditions is given by:

$$\boldsymbol{\sigma} = \frac{\sigma}{1+\beta_e^2} \begin{pmatrix} 1 & 0 & \beta_e \\ 0 & 1+\beta_e^2 & 0 \\ -\beta_e & 0 & 1+\beta_e^2 \end{pmatrix}. \tag{4.32}$$

The electrodynamic equation becomes:

$$\frac{\partial}{\partial x}(\frac{\partial \phi}{\partial x}) + \frac{\partial}{\partial y}[(1+\beta_e^2)\frac{\partial \phi}{\partial y}] \tag{4.33}$$

It follows from theorem 4.3.1 that a natural $x-y$ ordering has to be preferred in this case.
Table 4.1 shows the performances of an ILUT preconditioned GMRes for the problem 4.33 discretized on a structured triangular mesh, for the $x-y$ and $y-x$ ordering respectively.
As expected, best performances are obtained for the first ordering. Figure 4.7, 4.8 show a sketch of the matrix in the two cases. Triangular shaped markers depict low value elements $O(10^{-3})$ not visible in the gray scale representation. It can be seen that the nonzero pattern is almost the same, but the distribution of the coefficients modules is remarkably different, and causes the difference in the performances.

The objective of a reordering is to reduce the number and the module of the *fill-ins* entries. RCM achieves the first goal but misses the second, having no sensitivity to the preferable direction set by the physical conditions. It follows from the previous discussion that the RCM algorithm should be modified to take into account the anisotropy of the problem. A straightforward way to introduce such a dependence in the algorithm is to assign a *strength* to each connection in the matrix graph, and to reorder the nodes in the direction of the weakest connection.
The reordering technique adopted is therefore characterized by the definition

. . . .

Table 4.1: Performances of an ILUT(*level of fill* 20; threshold 1.0×10^{-4}) preconditioned GMRes on a structured triangular mesh, for constant $\mathbf{B} = 2T$ along the y direction.

	x-y ordering			y-x ordering		
Nodes	Iters	Conv. rate	Residual	Iters	Conv. rate	Residual
30	7	$1,5$	$1,3 \times 10^{-11}$	194	$5,7 \times 10^{-02}$	$4,1 \times 10^{-08}$
40	8	$1,8$	$2,7 \times 10^{-11}$	277	$4,0 \times 10^{-02}$	$3,9 \times 10^{-08}$
50	9	$1,2$	$2,2 \times 10^{-11}$	513	$2,1 \times 10^{-02}$	$3,7 \times 10^{-08}$
70	12	$0,92$	$3,2 \times 10^{-11}$	1113	$9,9 \times 10^{-03}$	$3,1 \times 10^{-08}$

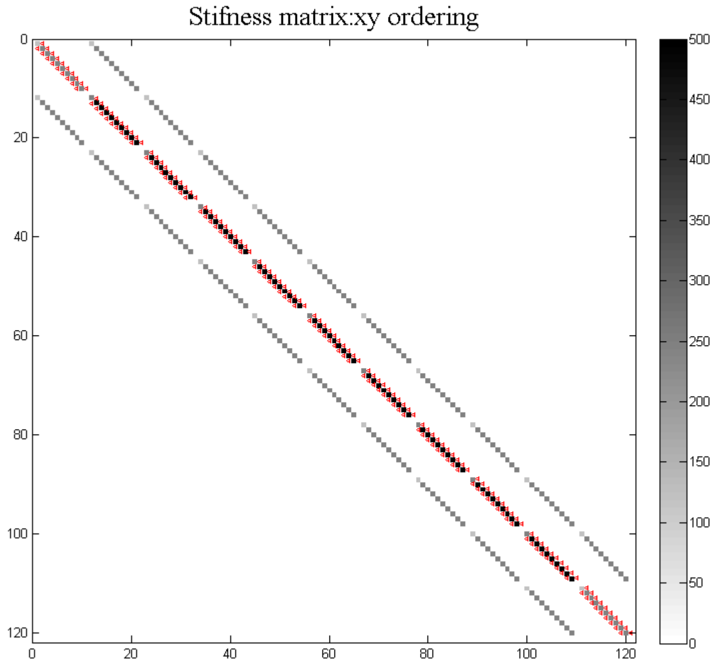


Figure 4.7: Pattern of the non-zero coefficient of the stiffness matrix for the problem described in text (absolute values): xy ordering.

of the connection strength. For the electrodynamic problem considered in this work, equations (4.32) and (4.33) suggest to identify the direction of the weakest connection with the one perpendicular to the magnetic flux density vector, since $\beta_e = \mu\mathbf{B}$. Once assigned a strength to each edge in the matrix graph, the nodes are reordered subsequently in each RCM level by increasing values of the scalar product of the mesh edges with the magnetic flux density vector. This algorithm is referred to in the following as Weighted RCM (WRCM) [65].

. . . .

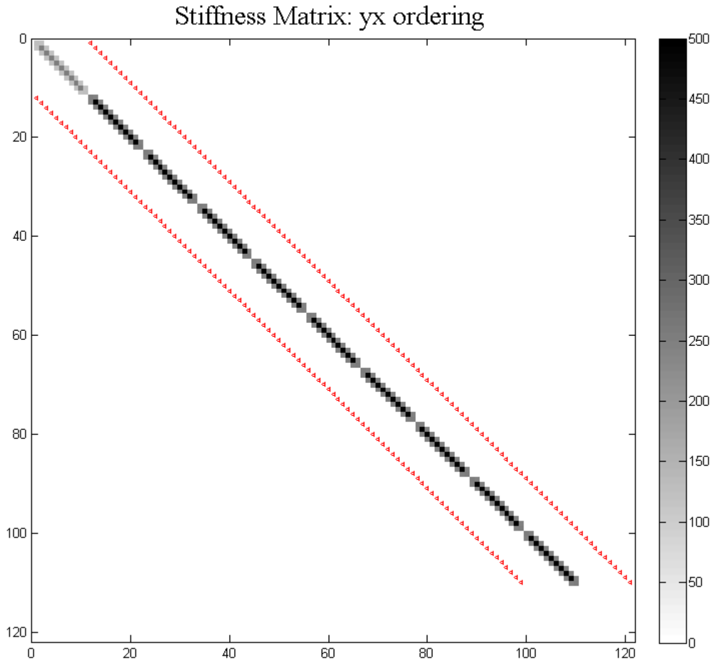


Figure 4.8: Pattern of the non-zero coefficient of the stiffness matrix for the problem described in text (absolute values): yx ordering

The dependence of the coefficient size on the magnetic field in the general case, for structured meshes, is given in the following calcution. For the sake of simplicity, the value of the mobility is assumed equal to one, and the multiplicative factor in the expression of $\boldsymbol{\sigma}$ has been ignored, since it doesn't affect the coefficient size distribution.
The element matrix coefficient $K_{m,n}$ for linear shape functions $N_j(x,y)$ is given by:

$$K_{m.n} = \int_V \nabla N_m^T \boldsymbol{\sigma} \nabla N_n dV \tag{4.34}$$

where,

$$\nabla N_j^T = (\frac{\partial N_j(x,y)}{\partial x}, \frac{\partial N_j(x,y)}{\partial y}) \tag{4.35}$$

and, in the assumed approximations:

$$\boldsymbol{\sigma} = \begin{pmatrix} 1+B_x^2 & B_xB_y \\ B_xB_y & 1+B_y^2 \end{pmatrix}. \tag{4.36}$$

. . . .

The element matrix coefficient is given by:

$$(\nabla N_{m,x}, \nabla N_{m,y}) \cdot \begin{pmatrix} \nabla N_{n,x} + \nabla N_{n,x} B_x^2 + B_x B_y \nabla N_{n,y} \\ \nabla N_{n,y} + \nabla N_{n,y} B_y^2 + B_x B_y \nabla N_{n,x} \end{pmatrix}. \tag{4.37}$$

that gives:

$$(\nabla N_{m,x}, \nabla N_{m,y}) \cdot \begin{pmatrix} B_x(\mathbf{B} \cdot \nabla \mathbf{N}_n) + \nabla N_{n,x} \\ B_y(\mathbf{B} \cdot \nabla \mathbf{N}_n) + \nabla N_{n,y} \end{pmatrix}. \tag{4.38}$$

The element matrix becomes therefore:

$$\begin{aligned} K_{m,n} &= (\mathbf{B} \cdot \nabla \mathbf{N}_n)(\nabla N_{m,x} B_x + \nabla N_{m,y} B_y) \\ &\quad + \nabla N_{m,x} \nabla N_{n,x} + \nabla N_{m,y} \nabla N_{n,y}. \end{aligned} \tag{4.39}$$

The final expression is therefore:

$$K_{m,n} = \int (\mathbf{B} \cdot \nabla N_m)(\mathbf{B} \cdot \nabla N_m) + (\nabla N_m \cdot \nabla N_n). \tag{4.40}$$

The contributions of the element matrixes add up to build the stiffness of the finite element formulation. It is apparent from equation **??** the dependence of the stiffness matrix coefficients on the scalar product between the magnetic flux density and the mesh edges, since the gradient of the linear shape functions are perpendicular to each edge of the triangular elements.

Two strategies can be adopted in preferred direction sensitive reorderings. The first reorders each RCM level by increasing degree, and applies the weakest connections ordering only to nodes having the same degree. The second possibility is to neglect the nodes degree, and to reorder accounting only for the connection strenghts. WRCM chooses the second approach, and does not apply a minimal degree ordering in each level. This choice enhances the sensitivity of the algorithm to the preferred direction, but increases the matrix profile. Nevertheless, this effect can be neglected provided the RCM levels are low populated. It can be obtained by a proper choice of the root node of the reordering, as highlighted in section 4.3.2.
It can be observed that the Dirichlet boundary conditions configuration could also play a role in the choice of the root node. If the potential is defined on the *m-th* grid point, the *m-th* row coefficients in the A matrix will be zero everywhere but in the position (m, m), where the coefficient is 1.
Now consider the incomplete elimination step given in algorithm 4.3. For each *i-th* row, the k loop sets to 0 all the elements before the diagonal. To eliminate the first element, each row is combined with the first one, to eliminate the second element with the second one, and so on. The j loop accomplishes this task. By looking at the step 5 in the algorithm 4.3, it can be easily seen that the elements $a_{k,j}$ generate *fill-ins* if they are not null: if

. . . .

the elements $a_{i,j}$ were null before the elimination of the *k-th* element in the row, they would become not null after the j loop unless the out-diagonal entries of the *k-th* row were null, that is, unless the *k-th* row corresponds to a point on a Dirichlet edge in the mesh.
Since each row is reduced by all the preceding rows, the previous considerations would imply that if the first rows in the matrix are "Dirichlet rows", the *fill-ins* could be reduced. These observations however apply to the first row only, or to a matrix configuration where "Dirichlet rows" are contiguous; otherwise the "Dirichlet rows" following a "non-Dirichlet row" would be modified by the procedure, and would not preserve their structure.
The algorithm could be forced to choose the starting node on the Dirichlet edge in order to have a "Dirichlet row" as first row, or even to choose all the n Dirichlet nodes as first n nodes, but it would imply a larger bandwidth, affecting the algorithm efficiency.
For the WRCM algorithm presented here, the root node has been chosen as a minimal degree node, since it has been noted that a well rooted RCM numbering it's enough to ensure good results in terms of profile reduction.

The WRCM procedure is shown in Fig. 4.9 and is coded in FORTRAN 90. The mesh is represented as a collection of *nodes* and *edges* types, on which acts a searching routine, returning the permutation vector for the nodes. Reordering is performed by a *quick-sort* implementing routine.

4.3.4 The weighted reverse Cuthill McKee: performances.

As a first step, the WRCM procedure has been tested for two simple configurations of the magnetic flux density, for the problem 4.33 defined on a triangular mesh of a square domain. For high values of electrons' mobility, the solver showed low rate convergence, or no convergence at all, for the examined cases. The definition of the field is given in the following, for a magnetic flux density aligned along straight lines, linearly decreasing (case I):

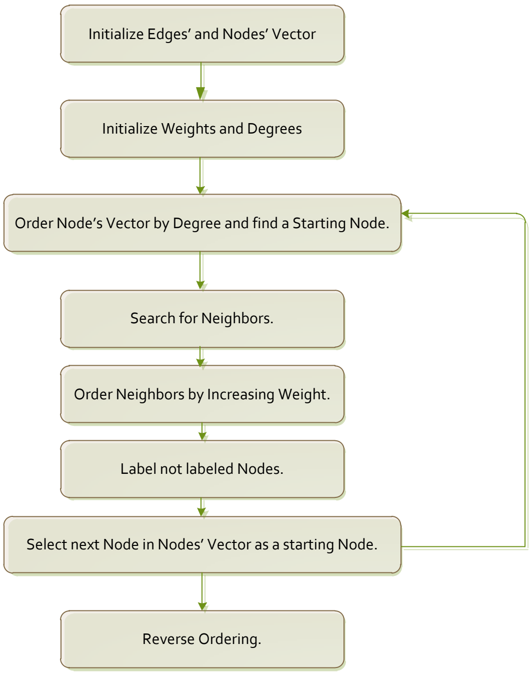
$$B_x = B_{1x} - x(B_{1x} - B_{2x})/L, \tag{4.41}$$
$$B_y = B_{1y} - y(B_{1y} - B_{2y})/L, \tag{4.42}$$

where

$$B_{1x} = 3.0T, \tag{4.43}$$
$$B_{2x} = 1.0T, \tag{4.44}$$
$$B_{1y} = 1.0T, \tag{4.45}$$
$$B_{2y} = 3.0T, \tag{4.46}$$
$$L = 0.1m. \tag{4.47}$$

. . . .

Figure 4.9: Code structure for the WRCM reordering routine.


Initialize Edges' and Nodes' Vector
Initialize Weights and Degrees
Order Node's Vector by Degree and find a Starting Node.
Search for Neighbors.
Order Neighbors by Increasing Weight.
Label not labeled Nodes.
Select next Node in Nodes' Vector as a starting Node.
Reverse Ordering.

and for a **B** field aligned along circular lines (case II, see Fig: 4.10):

$$B_x = -2.0y/(x^2 + y^2), \tag{4.48}$$

$$B_y = 2.0x/(x^2 + y^2). \tag{4.49}$$

The choice of the boundary conditions aimed to reproduce the physical configuration observed in the hypersonic MHD applications. The plasma region with high conductivity values is confined between the shock and the vehicle surface. The current path has therefore to be contained into this region. This situation is reproduced by setting to zero the normal current density on the boundary of the computational domain, *i.e.* by imposing a Neumann condition on the corresponding boundary edges. On the test square domain, the following boundary conditions have been set (see Fig: 4.10):

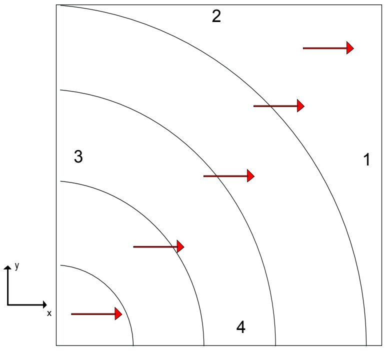
$$J_n = 0 \qquad edges \quad 1, 2, 3, \tag{4.50}$$

$$\phi = 0 \qquad edge \quad 4, \tag{4.51}$$

where J_n is the current density normal to the considered edge. The tests have been run on an Intel Xeon E5405 processor, and the GMres convergence criterion is given by [44]:

$$||r^{(k)}||_2 \leq ||r^{(0)}||_2, \qquad eps = 10^{-14} \tag{4.52}$$

Figure 4.10: **B** field (black) and velocity field (red) configuration for the second test case.



where $r^{(k)}$ is the residual vector after the k-th iteration.
The main computational difficulties arise at increasing values of the electron mobility, so tables 4.2, 4.3, 4.4, 4.5 and figures 4.12, 4.11 show the code performances for increasing mobilities for the given magnetic flux density layout.

Table 4.2: Performances of an ILUT (level of fill 20; threshold 1.0×10^{-4}) preconditioned GMRes: case I, unstructured, RCM reordered mesh.

Mobility (1/T)	Iters (No reord)	Iters (RCM)	Convergence rate (RCM)	Residual (RCM)
10	No Convergence	18	0.6	8.1×10^{-09}
20	No Convergence	21	0.5	1.3×10^{-08}
30	No Convergence	27	0.4	1.5×10^{-08}
40	No Convergence	29	0.4	1.5×10^{-08}
50	No Convergence	31	0.3	4.8×10^{-08}
70	No Convergence	32	0.4	1.1×10^{-08}
90	No Convergence	32	0.4	1.0×10^{-08}

Each table reports the number of iterations needed to reach the convergence in case of no reordering (second column), and when the specified ordering is applied (third column). For the latter, the convergence rate and the residuals are reported also (fourth and fifth column).
A comparison shows an outstanding improvement if the solver performance when an RCM or a WRCM is applied. WRCM shows a good stability and better performances for case II. Table 4.6 shows the outcomes of the application of the RCM and WRCM reordering to a strongly anisotropic case study. The code models the MHD interaction around a cone, in order

Table 4.3: Performances of an ILUT (level of fill 20; threshold 1.0×10^{-4}) preconditioned GMRes: case I, unstructured, WRCM reordered mesh.

Mobility (1/T)	Iters (No reord)	Iters (WRCM)	Convergence rate (WRCM)	Residual (WRCM)
10	No Convergence	18	0.6	8.1×10^{-09}
20	No Convergence	21	0.5	1.1×10^{-08}
30	No Convergence	30	0.4	1.4×10^{-08}
40	No Convergence	31	0.4	1.3×10^{-08}
50	No Convergence	31	0.4	4.3×10^{-08}
70	No Convergence	32	0.4	1.1×10^{-08}
90	No Convergence	31	0.4	9.2×10^{-09}

Table 4.4: Performances of an ILUT (level of fill 20; threshold 1.0×10^{-4}) preconditioned GMRes: case II, unstructured, RCM reordered mesh.

Mobility (1/T)	Iters (No reord)	Iters (RCM)	Convergence rate (RCM)	Residual (RCM)
10	31	19	0.6	8.3×10^{-09}
20	21	14	0.8	8.8×10^{-09}
30	24	17	0.7	2.4×10^{-09}
40	36	18	0.6	2.7×10^{-09}
50	60	21	0.5	2.9×10^{-09}
70	243	43	0.3	4.9×10^{-10}
90	1164	109	0.1	9.2×10^{-10}

Table 4.5: Performances of an ILUT (level of fill 20; threshold 1.0×10^{-4}) preconditioned GMRes: case II, unstructured, WRCM reordered mesh.

Mobility (1/T)	Iters (No reord)	Iters (WRCM)	Convergence rate (WRCM)	Residual (WRCM)
10	31	17	0.6	1.4×10^{-08}
20	21	13	0.9	2.2×10^{-09}
30	24	12	0.9	9.0×10^{-09}
40	36	17	0.7	8.2×10^{-10}
50	60	20	0.6	3.5×10^{-10}
70	243	21	0.5	3.0×10^{-09}
90	1164	37	0.3	2.1×10^{-09}

....

Figure 4.11: Iterations' number as a function of the mobility for case I, unstructured mesh.

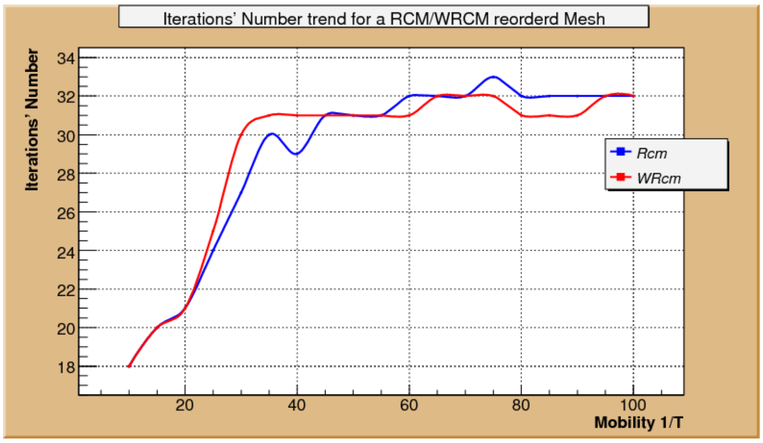
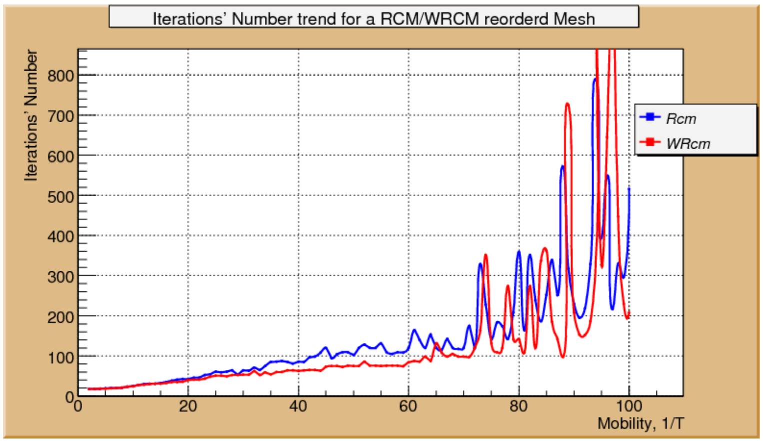


Figure 4.12: Iterations' number as a function of the mobility for case II, unstructured mesh.



to analize the results of an experimental campaign described in [23]. The improvement in performances introduced by the application of WRCM to this case is apparent.

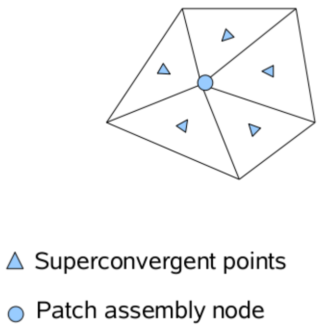
4.4 Field evaluation from the finite element solution.

This section addresses the problem of an optimal evaluation of the fields, when the potentials are known from the finite element solution.
The direct differentiation of the FEM potentials causes a loss in the accuracy of the approximation, and the obtained FEM fields show discontinuities at

Table 4.6: Performances of an ILUT (level of fill 20; threshold 1.0×10^{-4}) preconditioned GMRes: a strongly anisotropic test case, comparison between different orderings.

Ordering	Iters	Convergence rate	Residual
None	No conv.	-	-
RCM	501	2.2×10^{-02}	1.2×10^{-14}
WRCM	358	3.1×10^{-02}	9.0×10^{-14}

Figure 4.13: An element *patch* for the SPR technique.



elements' edges.
The strategy presented in the following *recovers* the loss in accuracy by a *a posteriori* treatment of the FEM data, *i.e*, it defines a procedure to evaluate at best the derivatives once the solution for the potential is given by the FEM method. Since the recovered values have higher accuracy, they can be used to give an estimate of the errors on the FEM fields.

4.4.1 The Superconvergent Patch Recovery: the technique.

The Superconvergent Patch Recovery (SPR) technique consists basically in finding out an optimal polynomial approximation for each component of the fields on a cluster of elements containing a given node, referred to as a *patch* (see fig 4.13).
The order p of the approximating polynomial is defined by the order of the elements. For the linear elements in two dimensions adopted in this work, the polynomial will be a linear function of the coordinates.
The procedure to build such polynomials can be determined by considering the following theorem, derived by Hermann [66]:

Theorem 4.4.1. *Consider the differential equation*

$$\mathbf{Lu} = \mathbf{S}^T\mathbf{ASu} = \mathbf{p} \tag{4.53}$$

where $\mathbf{L}$ *is the self-adjoint operator defined by* $\mathbf{S}$*, and* $\mathbf{A}$ *and* $\mathbf{p}$ *are known matrices of position.*
The minimization of the energy functional Π *defined as:*

$$\Pi = \frac{1}{2}\int_\Omega (\mathbf{Su})^T\mathbf{ASu}d\Omega - \int_\Omega \mathbf{u}^T\mathbf{p}d\Omega \tag{4.54}$$

gives at an absolute minimum the exact solution $\mathbf{u} = \bar{\mathbf{u}}$*, and it is equivalent to minimization of another functional* Π^* *defined as*

$$\Pi^* = \frac{1}{2}\int_\Omega [\mathbf{S}(\mathbf{u}-\bar{\mathbf{u}})]^T\mathbf{A}[\mathbf{S}(\mathbf{u}-\bar{\mathbf{u}})]d\Omega \tag{4.55}$$

For the electrodynamic problem treated in this work, $\mathbf{u}$ represents the potential, and $\mathbf{S}$ is the gradient operator. The first outcome of the Hermann theorem is therefore that the *weighted sum of squares* of errors on the fields has a minimum when the approximate gradients $\mathbf{Su}$ approach the exact ones $\mathbf{S}\bar{\mathbf{u}}$.

The Hermann theorem gives a first indication about the approximating polynomials: they should fit the sampling points in a least square manner, in order to guarantee the convergence to exact solution. The theorem further states that a choice that minimizes 4.55 converges to exact values for the gradients. The order of convergence depends on the sampling points. Equation 4.55 is a continuum least square problem formulation. It's known [67] that a minimizing solution is given by the least square interpolation of the function $\mathbf{S}\bar{\mathbf{u}}$ at the Gauss-Legendre points in each element of the *patch*. The approximation is exact to the order $p + 1$, if the Gauss points are of order p. The error on the approximated gradients has therefore order $p + 1$, the same order of the errors on the potential. This property is known as *superconvergence*. For a detailed proof of the superconvergent behavior of the SPR approximation, see, for example, [68, 69, 70, 71, 72]. It's worth noting that the error estimate given above is valid for structured meshes. Nevertheless, superconvergent behavior for unstructured mesh has also been observed, and proved in some cases. The SPR method, proposed by Zinkiewicz and Zhu [73, 74], is at present a widely applied recovery method, and its effectivity in terms of performances and computational costs have been proven, see for example [75], though several improvements have been developed for particular applications.

The practical implementation of the technique is simple, and it's outlined below [38]:

1. Identify the Gauss points of order p in each element. The coordinates of the k-th point are (x_k, y_k).

2. Evaluate the FEM gradients components $\hat{\sigma}_i(x_k, y_k)$ at the Gauss points.

3. For each element *patch*, build the polynomial:

$$\bar{\sigma}_i^* = \mathbf{p}(x, y)\mathbf{a}_i \tag{4.56}$$

where:

$$\mathbf{p}(x, y) = [1, x, y, \cdots, y^p] \tag{4.57}$$

$$\mathbf{a}_i = [a_1, a_2, \cdots, a_{p+1}]^T \tag{4.58}$$

$$\tag{4.59}$$

that minimizes

$$\Pi = \frac{1}{2}\sum_{k=1}^{n}[\bar{\sigma}_i(x_k, y_k) - \mathbf{p}(x_k, y_k)\mathbf{a}_i] \tag{4.60}$$

4. The coefficients $\mathbf{a}_i$ are obtained as

$$\mathbf{a}_i = \mathbf{A}^{-1}\mathbf{b}_i \tag{4.61}$$

where:

$$\mathbf{A} = \sum_{k=1}^{n} \mathbf{p}_k^T \mathbf{p}_k \tag{4.62}$$

$$\mathbf{b}_i = \sum_{k=1}^{n} \mathbf{p}_k^T \hat{\sigma}_i(x_k, y_k) \tag{4.63}$$

$$\tag{4.64}$$

5. The boundary values are recovered from interior *patch*es, see figure 4.14.

4.4.2 The Superconvergent Patch Recovery: implementation.

Figure 4.15 shows the code structure for the Superconvergent Patch Recovery routines. The procedure is straightforward:

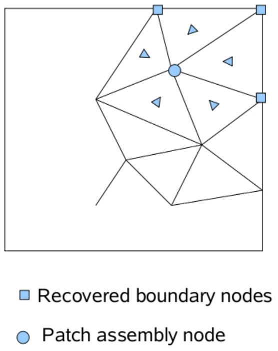
- For linear triangular elements the center of gravity can be identified as the superconvergent point[4]

- For each internal element *patch*, evaluate the least square plane approximating the Gauss points for the each field component (*i.e.* the x, y components, for the axial-symmetric field configuration adopted in the two dimensional model presented in this work).

[4] The superconvergence of the center of gravity has not been fully proved yet, see for example [74].

. . . .

Figure 4.14: Recovery of boundary gradients.



- Compute the superconvergent nodal values by the plane equations.

- Recover the boundary values from the superconvergent values computed for the connected internal nodes.

- The errors are evaluated as the euclidean norm of the difference between FEM fields and SPR fields. The FEM fields have been obtained by averaging the FEM gradients computed on each element of the *patch*.

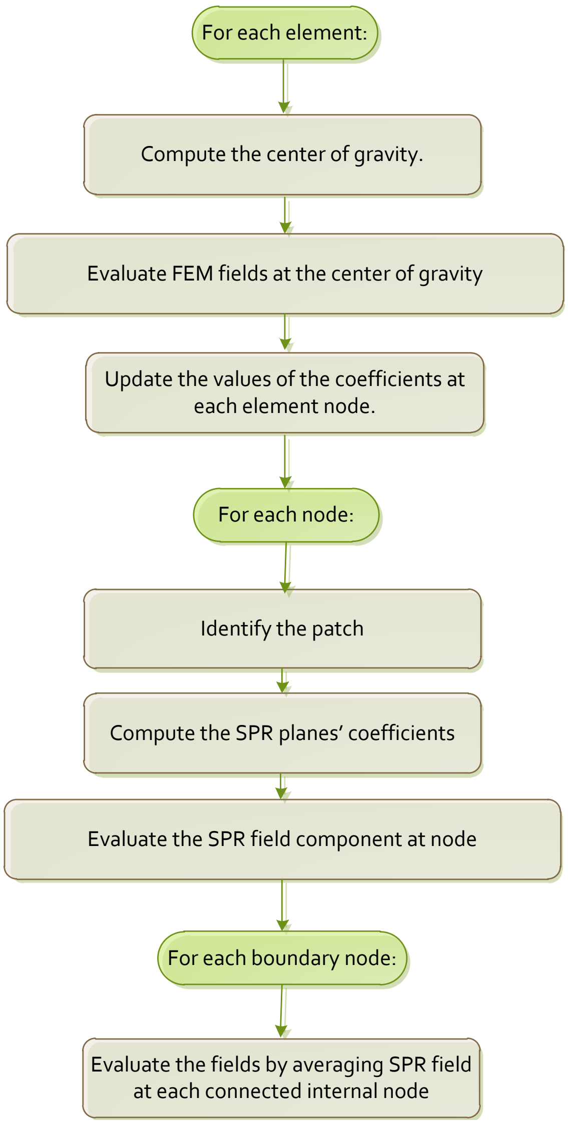
For a structured mesh the expected convergence order would be ($O(h)^2$), where h is the elements size, for an unstructured mesh the convergence order is greater than 1.5, see [74]. Some details about the calculations are given in the following.
The least square planes for the x, y field components are written as:

$$E_x^*(x,y) = a_x x + b_x y + c_x \tag{4.65}$$
$$E_y^*(x,y) = a_y x + b_y y + c_y \tag{4.66}$$

The least square minimization generates a linear system for the determination of the unknown coefficients, whose Cramer solution is given by:

$$a_x = \frac{|N_{ax}|}{|A|} \qquad a_y = \frac{|N_{ay}|}{|A|} \tag{4.67}$$
$$b_x = \frac{|N_{bx}|}{|A|} \qquad b_y = \frac{|N_{by}|}{|A|} \tag{4.68}$$
$$c_x = \frac{|N_{cx}|}{|A|} \qquad c_y = \frac{|N_{cy}|}{|A|} \tag{4.69}$$

Figure 4.15: Code structure for the SPR routine.


For each element:
Compute the center of gravity.
Evaluate FEM fields at the center of gravity
Update the values of the coefficients at each element node.
For each node:
Identify the patch
Compute the SPR planes' coefficients
Evaluate the SPR field component at node
For each boundary node:
Evaluate the fields by averaging SPR field at each connected internal node

where:

$$|N_{ax}| = \lambda\beta^2 + \theta\mu + \alpha\epsilon\gamma - \lambda\epsilon - \theta\beta\gamma - \alpha\beta\mu \tag{4.70}$$
$$|N_{bx}| = -\delta\mu - \lambda\alpha\beta - \alpha\theta\gamma + \delta\beta\gamma + \lambda\theta + \alpha^2\mu \tag{4.71}$$
$$|N_{cx}| = -\delta\epsilon\gamma - \theta\mu\alpha - \lambda\theta\beta + \delta\mu\beta + \theta^2\gamma + \lambda\epsilon\alpha \tag{4.72}$$
$$|A| = \delta\epsilon + 2\alpha\theta\beta - \delta\beta^2 - \theta^2 - \alpha^2\epsilon. \tag{4.73}$$

. . . .

The least square plane coefficients are functions of the coordinates and of the field values at the superconvergent points. The Greek letters above define the summations over the *patch* of the superconvergent points' contribution:

$$\alpha = \sum_{i=1}^{N} x_g^i \quad \beta = \sum_{i=1}^{N} y_g^i \quad \gamma = \sum_{i=1}^{N} E_{xg}^i \tag{4.74}$$

$$\delta = \sum_{i=1}^{N} (x_g^i)^2 \quad \epsilon = \sum_{i=1}^{N} (y_g^i)^2 \quad \theta = \sum_{i=1}^{N} x_g^i y_g^i \tag{4.75}$$

$$\lambda = \sum_{i=1}^{N} E_{xg}^i x_g^i \quad \mu = \sum_{i=1}^{N} E_{xg}^i y_g^i \tag{4.76}$$

The SPR method has been applied to analyze the outcomes of a 9.2 Mach simulation of the flow around a blunt test body. The results are shown in Fig. 4.16, where the error is mapped as the ratio between the euclidean norm of the difference between FEM field and SPR field, and the norm of the SPR field itself. By comparing Fig. 4.16 with figures 4.17 it is apparent that greater errors arise at the shock layer boundaries, and at the edges of a mesh refinement region. The high error percentages observed at the middle of the rightmost edge are explained by noting that boundary values are recovered by averaging the SPR values computed at the connected internal nodes. Since in this case the connected nodes are located in the shock zone, the discontinuities in the physical quantities result in a poor estimate for the average on the boundary. This difficulty could be overcome by a overall mesh refinement, or by simply assigning to each boundary node the SPR field value computed for the nearest internal node connected with it.

4.5 Performance analysis: code profiling.

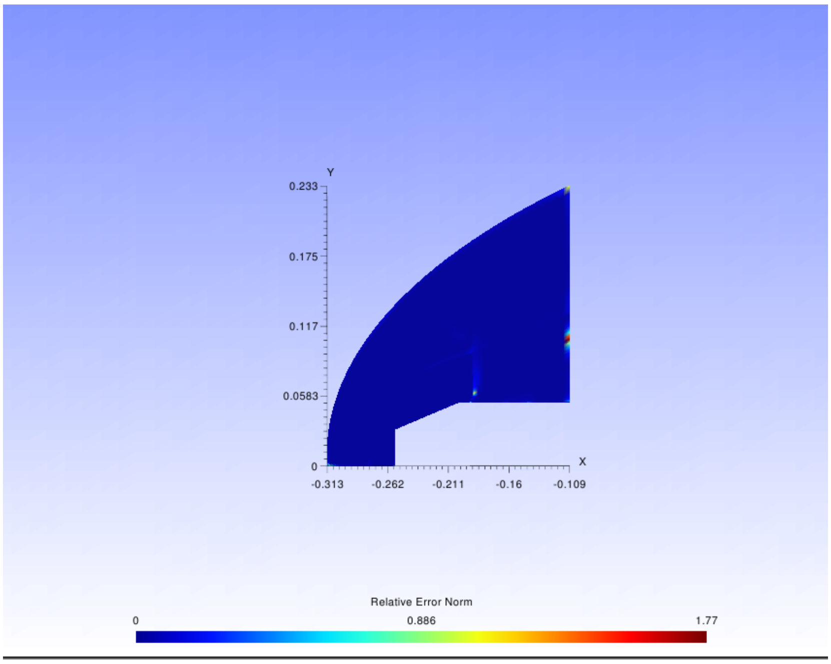
As a final testing step, the code has been analyzed by using a profiling tool. The aim of such analysis is to identify the most time expensive code sections, and to study the memory usage efficiency.

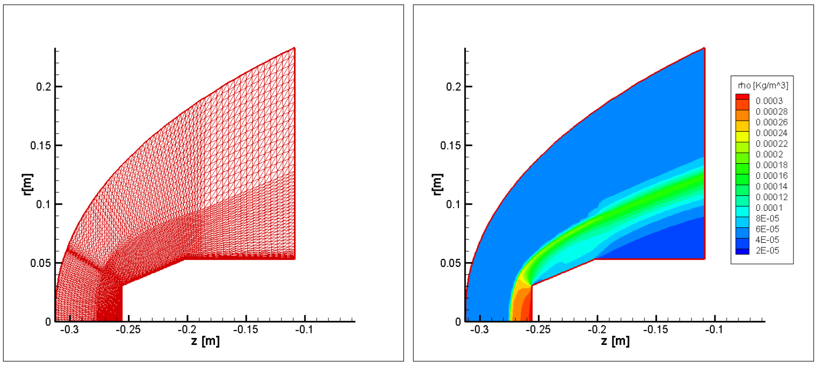
The time costs analysis has been performed through gProf, the GNU profiling utility. The outcomes are summarized in Table 4.7. The table shows the percentage of the total running time spent in each listed function, the number of calls, and the average time per call spent in the function itself and all its descendants, for a test run with 6565 nodes.

It can be seen that the most time consuming functions are, as can be expected, the preconditioner *iluk*, the solver *GMRes* and the LU solver *lusol*, needed by both the solver and the preconditioner.

The reordering modules that build the mesh edges, sort the nodes and renumber them, respectively, have percentages lower than six, i.e. they have a low time cost when compared with the significant improvements they induce in

Figure 4.16: Relative norm of the error in the fields computation by comparison with the superconvergent estimate.





(a) **(b)**

Figure 4.17: Mesh structure and density distribution for the test case cited in text.

the solver performances. It has also to be noted that nodes are reordered just one time at the beginning of the simulation, whereas the matrix arising in the FEM electrodynamic discretization is solved at each fluid-dynamics time step.

The second phase of the analysis described in this section aimed to test the

Table 4.7: Time profile for the 2D electrodynamic code.

Name	% Time	Calls	Total/Call [ms]
iluk (preconditioner)	29.73	1	220.00
GMRes (solver)	14.86	25	5.19
lusol (LU factors comp.)	10.81	28	2.86
nodes' relabeling	5.41	1	40.00
nodes' sorting	2.70	6565	$<$ 0.01
edges' building	1.35	1	10.00

cache usage efficiency of the code. A good management of the read/write operations in the program work flow maximizes the successful accesses to the cache (*cache hits*) and minimizes therefore the execution time by avoiding, as far as possible, frequent data and instructions exchange between the processor and the RAM, typically slower than cache memories by a factor 100. The compiler itself (*e.g.* the Intel compiler adopted for this work [76]) provides options to exploit the cache at best(*loop unrolling, loop fusion, loop interchange, cache padding* support), but aggressive optimizations produce larger binaries and could cause unexpected code behaviors. The only reliable solution is therefore a well structured code, and a cache profiling tool helps identifying the inefficient code segments that have to be optimized.

The results presented below have been obtained by running *Cachegrind* with the *Callgrind* extensions. *Cachegrind* is the cache profiling tool supplied by the *Valgrind* instrumentation framework. It runs the code and tracks the cache access operations, counting the cache misses for different cache levels (data and instruction caches for level 1, and the level 2 unified cache). *Callgrind* builds a call graph of the program's run, collecting data about the caller/callee relationships between functions, the number of calls, the number of instructions, and so on. All these results are linked to the source code lines and can be grouped by function. Table 4.8 shows the data about the cache configuration, automatically detected by Callgrind: the cache size, the cache line size and the associativity. The machine is equipped with a quad-core processor, and each couple of cores shares a 6M L2 cache. Cachegrind detects only a 6MB cache and will therefore simulate this configuration. It doesn't constitute a problem, because the code is not engineered for parallel execution and will therefore exploit only one L2 cache, i.e. the 6M detected by Callgrind. Tables 4.9, 4.10 report the results obtained for a test run with 2560 nodes, with the same configuration applied for the time profiling.

For each event type, the table shows the most expensive functions (ordered by decreasing self cost percentage), and the corresponding cost percentage. Self cost refers to the cost of the function itself (callee costs are not included). The event types tracked by *Callgrind* are:

. . . .

- Data read misses for level 1 (D1mr) and level 2 (D2mr)
- Data write misses for level 1 (D1mw) and level 2 (D2mw)

Table 4.8: Cache configuration detected by Callgrind.

Cache	Level	Dimension	Line	Associativity
Instruction	1	32KB	64 B	8-way associative
Data	1	32KB	64 B	8-way associative
Unified	2	6MB	64 B	24-way associative

Table 4.9: Cache profile for the 2D electrodynamic code: costs grouped by function, level 1 cache.

D1mr	%	D1mw	%
lusol (LU factors comp.)	42.18	nodes' relabeling	36.10
scalar product	15.04	matrix-vector multiplic.	12.24
GMRes (solver)	13.67	lusol (LU factors comp.)	11.82
iluk (preconditioner)	11.49	iluk (preconditioner)	11.68
nodes' relabeling	5.06	solver initializations	5.06

Table 4.10: Cache profile for the 2D electrodynamic code: costs grouped by function, level 2 cache.

D2mr	%	D2mr	%
lusol (LU factors comp.)	69.05	nodes' relabeling	24.99
scalar product	16.32	iluk (preconditioner)	15.91
matrix-vector multiplic.	10.86	matrix-vector multiplic.	15.43
nodes' relabeling	1.61	lusol (LU factors comp.)	11.60
GMRes (solver)	0.88	solver initializations	7.25

In order to understand the reasons of the detected cache misses, the source code has to be analyzed.

Cachegrind produces a report of the cache data for each code line. The source line annotations for the *lusol* routine are shown in the following for exampling purpose:

....

```
     D1mr    D1mw     D2mr   D2mw

        0      28        0     28   subroutine lusol(n, y, x, alu, jlu, ju)
        .       .        .      .  c
        0       0        .      .          do 40 i = 1, n
  112,000 112,000        0 76,369             x(i) = y(i)
  112,028       0       .             do 41 k=jlu(i),ju(i)-1
1,415,344        0 925,579      .       x(i) = x(i) - alu(k)* x(jlu(k))
        .       .    .          .    41        continue
        0       0    1          .   40     continue
        .       .    .          .  c     backward solve.
        .       .    .          .  c
       28       0    2          .   do 90 i = n, 1, -1
  108,536       .    .          .      do 91 k=ju(i),jlu(i+1)-1
1,450,400        1 773,310      .        x(i) = x(i) - alu(k)*x(jlu(k))
        .       .    .          .   91      continue
  197,680       0  111,742      .             x(i) = alu(i)*x(i)
        .       .        .      .   90   continue
        .       .        .      .  c
        .       .        .      .    return
        .       .        .      .  c----------------end
```

This routine solves the system $(LU)x = y$. *Alu* and *jlu* variables contain the LU factors stored in CSR format [45]. It can be easily seen by looking at the highlighted lines that the D1mr misses are mainly due to random accesses to the x vector, that cannot be avoided, because they depend on the algorithm structure and on the CSR storage strategy. The same considerations apply to the nodes' relabeling procedure (the main responsible for D2mr misses, see Table 4.9), that implies a random access to the node's vector. Similar analysis have been carried out on the other functions, and the code turned out to be effective in terms of cache accesses.

CHAPTER 5

SIMULATION RESULTS

Any man who can drive safely while kissing a pretty girl is simply not giving the kiss the attention it deserves.

Albert Einstein.

Abstract. This chapter presents the simulations results obtained from the electrodynamic code. The two dimensional code has been tested by modeling the MHD interaction around two axial symmetric blunt body configurations, for scalar and tensor conductivity. The shock displacement effect, the pressure distribution modification and the heat fluxes mitigation have been shown to be of order of 10% in the scalar conductivity case, whereas they can be hardly appreciated in presence of tensor sigma. For the three dimensional code, the preliminary testing phase is briefly overviewed.

. . . .

Introduction

The electrodynamic code has been worked out within the CAST[1] project, funded by the Italian space agency (ASI). The project aimed at developing an integrated experimental-computational environment to explore new possible configurations for aerospace transport. In this context, it has been studied the MHD interaction between the plasma surrounding a re-entry vehicle and the magnetic field generated by magnets placed on the vehicle body, in order to evaluate the stagnation heat fluxes reduction and pressure field modification.
The proposed configurations have been numerically modeled and then experimentally tested in the hypersonic nozzles located at the CIRA [77] and ALTA [?] facilities. The results presented in the following sections simulate blunt test body configurations in different flow condition.
The last section outlines the preliminary results obtained for the tridimentional code.

5.1 Two dimensional electrodynamics.

This chapter describes the simulations of the MHD interaction in the hypersonic flow generated by the GHIBLI wind tunnel at CIRA. The chosen working point for the nozzle is characterized by an enthalpy of $H_0 = 22MJ/Kg$ and $P_0 = 2atm$. The flow conditions are specified for each presented test case.

5.1.1 Test cases.

The first simulation, referred to as test case I, considers a truncated-cone body geometry, in an hypersonic air flow. The considered magnetic field at the body surface is shown in figure 5.1, whereas figures 5.2 show the the Mach number distribution and ionization degree as calculated by the CIRA fluid-dynamic code. It can be noted by looking at Figure 5.2 that the ionization degree is almost constant on the computation domain. The scalar electrical conductivity is given by :

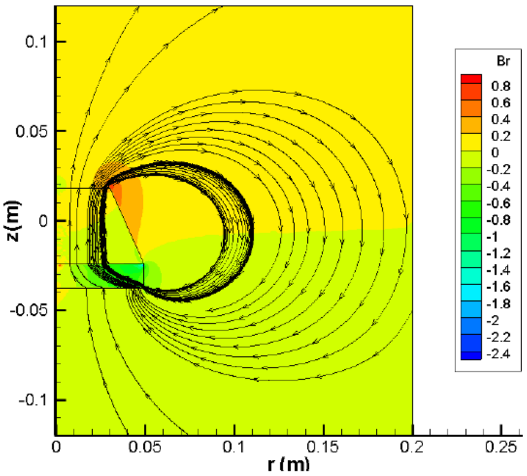
$$\sigma_e = \frac{n_e e^2}{m_e \bar{\nu}_{eH}} \tag{5.1}$$

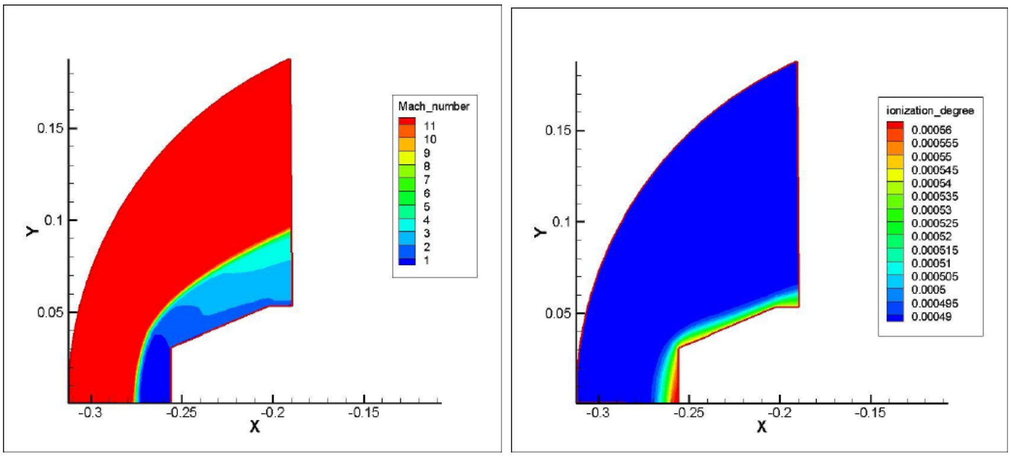
where e is the electron charge, m_e is the electron mass, and $n_e/\bar{\nu}_{eH}$ is the ratio between the electron number density and electron-heavy particles collision frequency, that can be can be assumed to be constant, for a constant ionization degree. The electron conductivity has therefore been assumed to be constant on the domain, whereas the electron mobility has been varied

[1] New Configurations for Aerospace Transport

. . . .

Figure 5.1: Magnetic flux density at the test body surface.



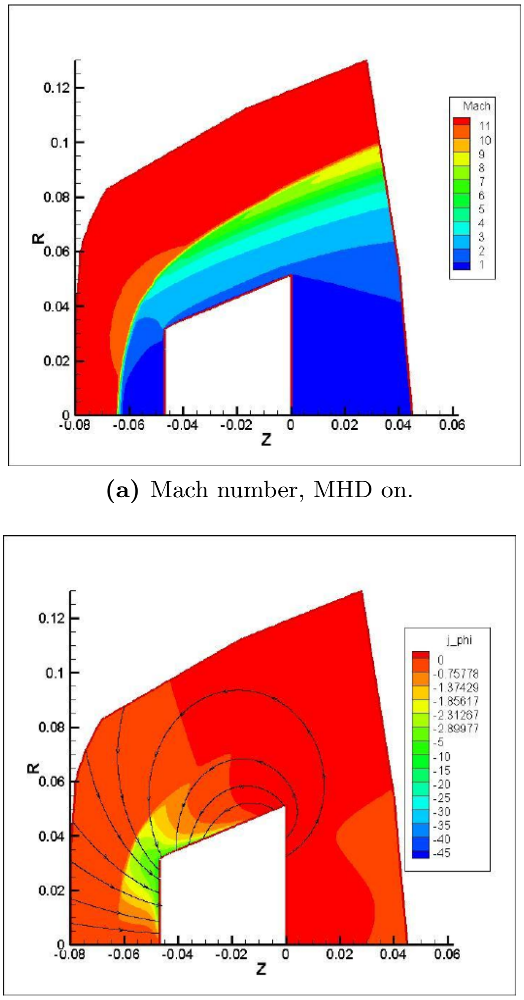


(a) Mach number, no MHD.

(b) Ionization degree.

Figure 5.2: Flow conditions for test case I.

as the inverse plasma density, since it depends on the inverse of $\bar{\nu}_{eH}$. The electron collision rate has been evaluated under the assumption of instantaneous electrons relaxation from [78]. In Fig 5.3a are shown the results for the distribution of the Mach number in presence of MHD interaction. A plot of the azimuthal component of the adimensional current density and of the B field lines are shown in Fig. 5.3b. The expected profile of pressure and temperature gradient on the test body wall with and without MHD interaction, as well as the computed percentage variations, are shown in figures 5.4a, 5.4b, 5.5a, 5.5b and 5.6a, 5.6b, 5.7a, 5.7b.

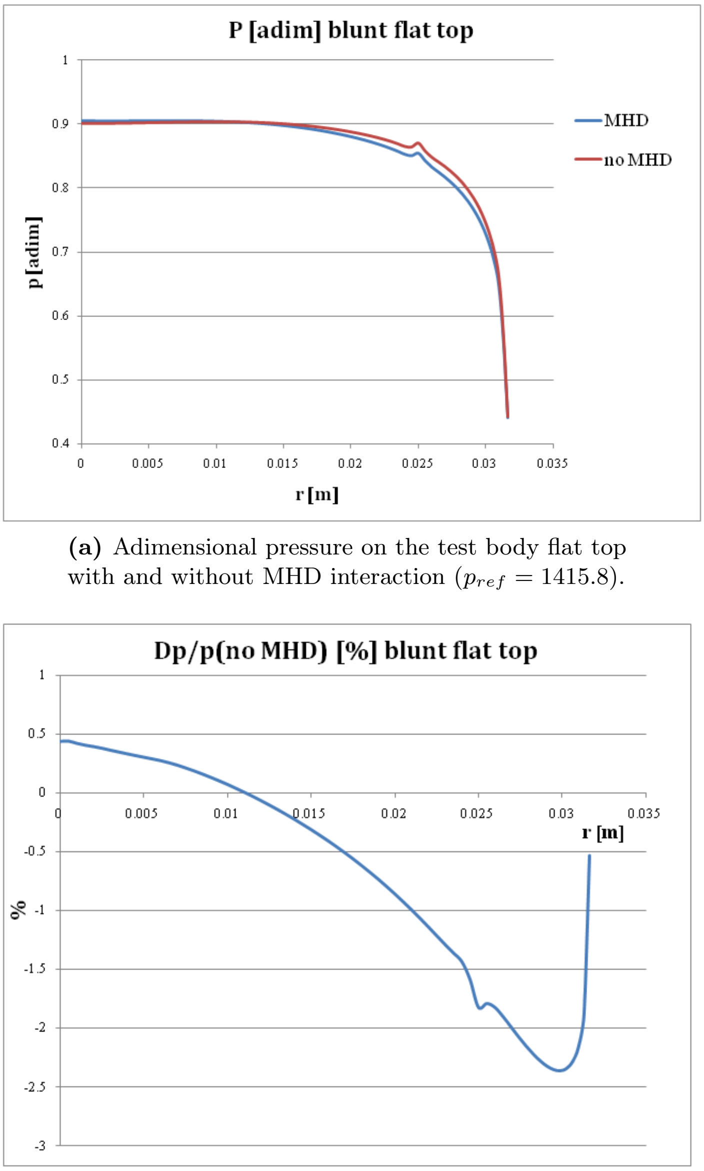


(a) Mach number, MHD on.

(b) Azimuthal current density around the test body (adimensional, $J_{ref} = 1415.8A/m^2$), MHD on.

Figure 5.3: Mach number and azimuthal current density for test case I.

It's apparent from the calculation the reduction in temperature gradients expected at the test body flat top, at the order of some percentage. It has to be noted that this order of magnitude has been obtain for a scaled model, and the involved quantities are one order of magnitude lower than the corresponding quantities in the final application. The pressure reduction can be explained with modification of the momentum streamlines induced by the interaction. The second test case presented in this section aims at comparing the Faraday current densities in different conductivity conditions.

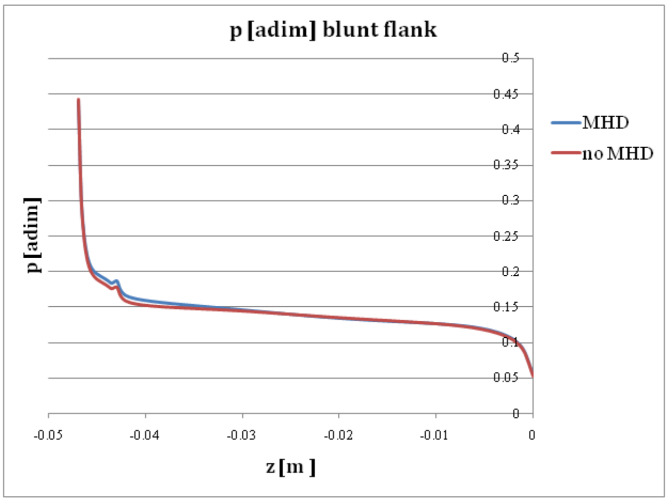


(a) Adimensional pressure on the test body flat top with and without MHD interaction ($p_{ref} = 1415.8$).

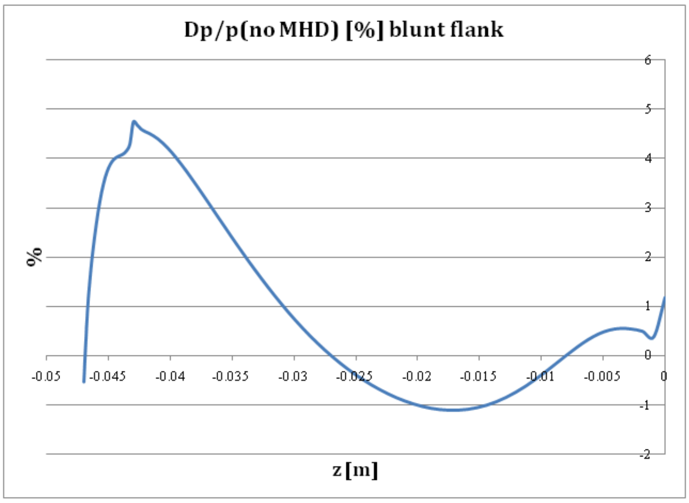
(b) Percentage variation of pressure due to MHD interaction on the test body flat top.

Figure 5.4: Pressure behavior on the test body top with and without MHD interaction.

Argon has been chosen as working gas, since the higher electron density provided maximizes the effect under study.
A sketch of the geometry chosen for the comparison is shown in Fig. 5.8. The **B** field is produced by a 56500 A current flowing on a circular 29 mm radius winding. With this configuration, a 1 T magnetic flux density is obtained.



(a) Adimensional pressure on the test body flank with and without MHD interaction ($p_{ref} = 1415.8 Pa$).



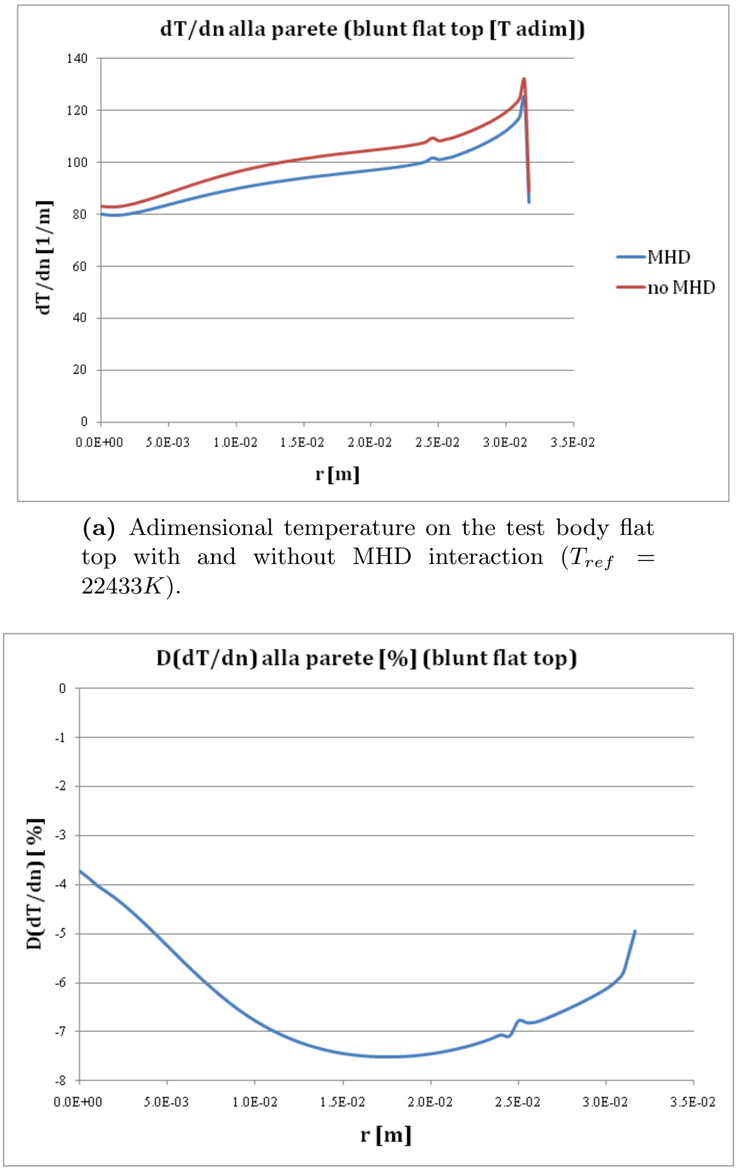
(b) Percentage variation of pressure due to MHD interaction on the test body flank.

Figure 5.5: Pressure behavior on the test body flank with and without MHD interaction.

on the axis of the body faced to the flow. The assumed free stream conditions are listed below:

- $Mach = 15.7$
- $p = 2.013603698 \quad [Pa]$
- $T = 36.098665 \quad [K]$

. . . .

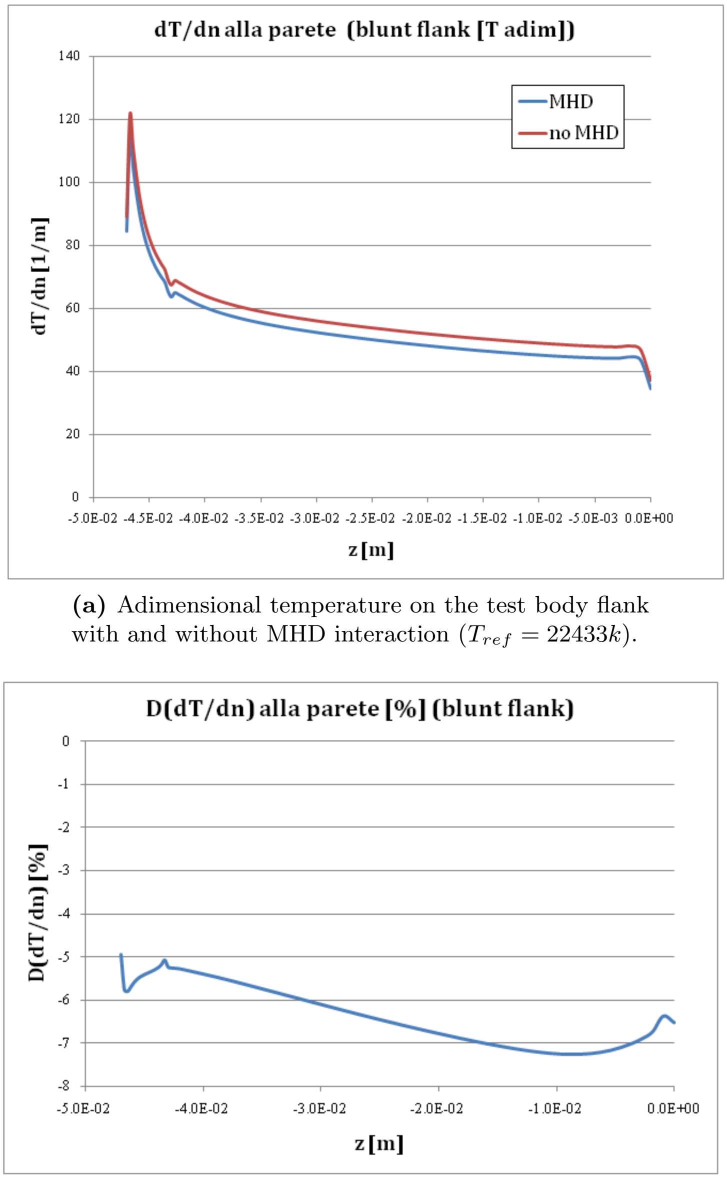


(a) Adimensional temperature on the test body flat top with and without MHD interaction ($T_{ref} = 22433K$).

(b) Percentage variation of temperature due to MHD interaction on the test body flat top.

Figure 5.6: Temperature behavior on the test body flank with and without MHD interaction.

A scalar conductivity has been assumed on the calculation domain. In this case the only component of the current density is the azimuthal one, and can be promptly evaluated the cross product between the velocity and the **B** field times the conductivity. It should also be noted that, under these assumptions, the MHD source terms appearing in the fluid dynamic equa-



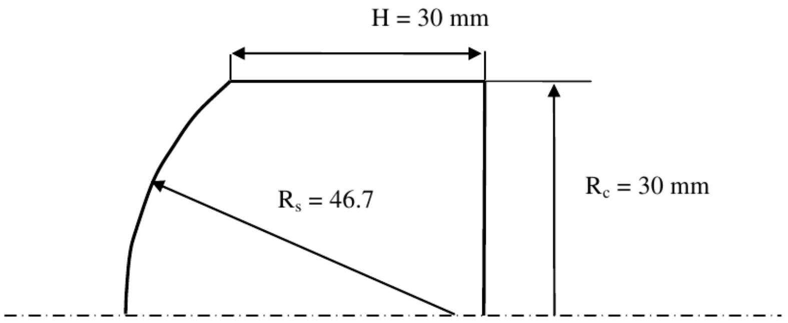
(a) Adimensional temperature on the test body flank with and without MHD interaction ($T_{ref} = 22433k$).

(b) Percentage variation of temperature due to MHD interaction on the test body flank.

Figure 5.7: Pressure behavior on the test body flank with and without MHD interaction.

tions depend on the product σB^2, which is defined and does not vary during the solution. Each solution can be considered as a family of solutions with constant MHD interaction parameter (*i.e.* constant σBe^inter2). Thus, the entire required range of *Rem* can be explored on a single solution by varying the conductivity of a factor α and the **B** field of a factor $\alpha^{-\frac{1}{2}}$. The current

Figure 5.8: Sketch of the geometry for the test case II.



density in the plasma will be increased by a $\alpha^{-\frac{1}{2}}$ factor, leaving the source terms unmodified. This expedient can be useful because exploring MHD regimes with high conductivity will inevitably require large calculation domains, and time expensive computations [79]. The conductivity has been modeled as a step function, set to zero outside the shock layer. The results for this condition are reported in figures 5.9a, 5.9b, 5.9c ($\sigma = 10 \quad S/m$) and in figures 5.10a, 5.10b ($\sigma = 100 \quad S/m$).

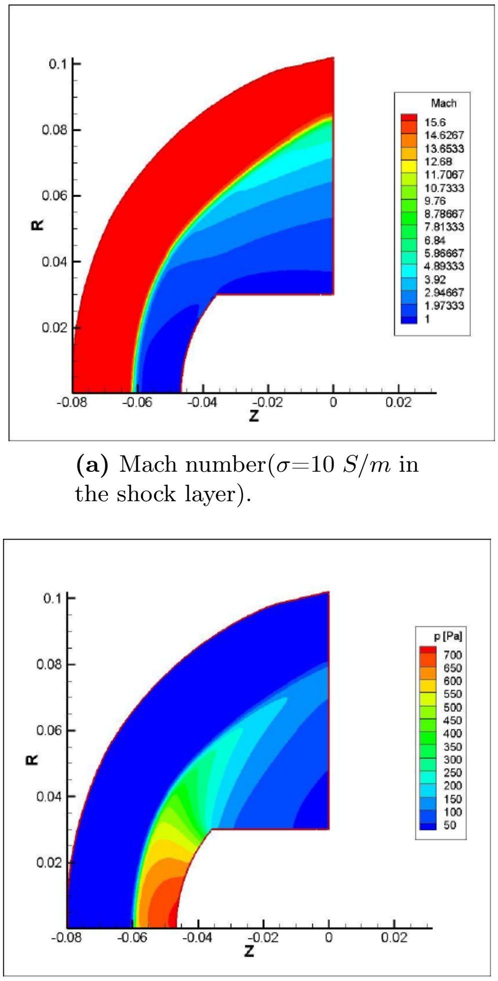
The scalar conductivity is increased by an order of magnitude. The shock is moved away from the test body nose and the increase of this effect for increasing σ by comparing figures 5.9b and 5.10a. It has to be noted that the scalar conductivity condition simulated in this test case, is the most favorable for the MHD effects to be appreciated, since no Hall currents are present on domain, and the Faraday current sustaining the MHD phenomenon is maximized.

5.1.2 Integration.

The electrodynamic code described in this work couples with the two dimensional fluiddynamics developed at CIRA through data exchange by file. The coupling schema is reported in figure 5.11. The electrodynamic code ELDYNSOLV interfaces with the fluidodynamical mesh by building the elements onto the the grid points defined by the fluidodynamics, in order to avoid the accuracy loss implied by interpolation.

This section presents the computational analisys of the MHD interaction around a truncated cone shaped test body, to be tested in the GHIBLI nozzle with air as working gas [80], at the same conditions adopted for the simulations presented in section 5.1.1.

The magnetic field needed for the MHD interaction around the blunt test body is produced by a single permanent magnet, with 1.4T residual induction. The height, the minor and the major radius of the truncated cone are



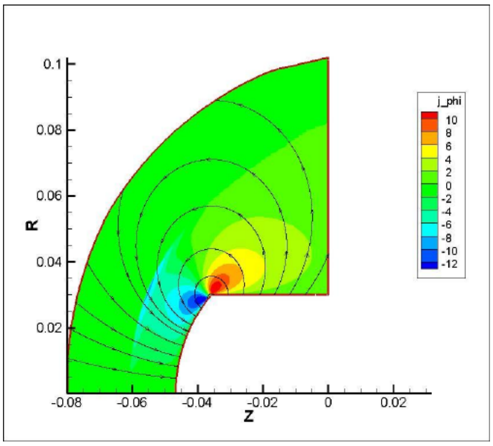
(a) Mach number(σ=10 S/m in the shock layer).

(b) Pressure (σ=10 S/m in the shock layer).

Figure 5.9: Results for the test case II, σ=10 S/m condition.

45 mm, 96 mm and 60 mm respectively. The final configuration of the test body is shown in figure 5.12. The structured mesh for the finite element computation is shown in figure 5.13. The magnetic field configuration is the same given in section 5.1.1 and has been interpolated on the domain. Computed and interpolated fields are shown in figures 5.1, 5.14 respectively.

The first results presented have been obtained by neglecting the free stream ionization: the currents are therefore totally contained in the shock region. The conductivity has been assumed to be scalar; as previously noted, in such case the only Faraday current is present and the MHD effect is therefore maximized (see 5.15). By observing figure 5.15 it is apparent the shock wave displacement, whereas figure 5.16 shows a reduction of wall pressure



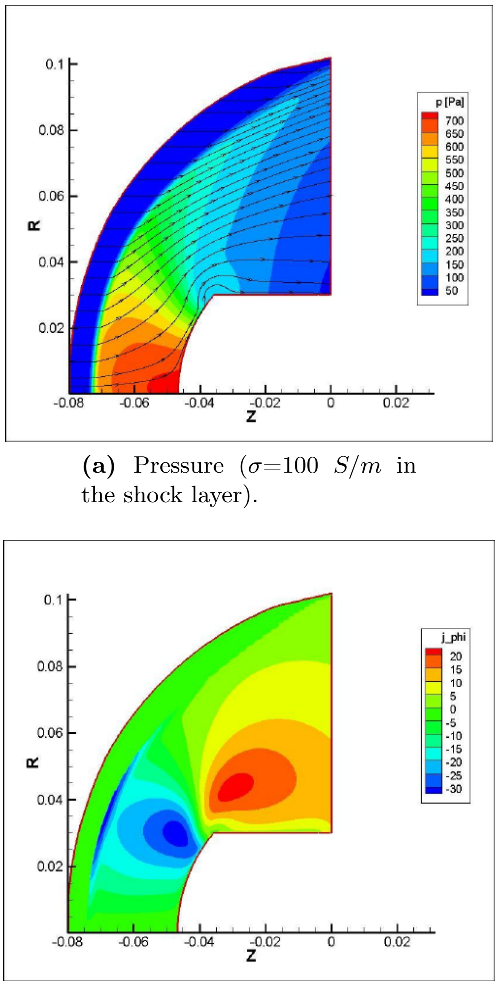
(c) Current density and B field lines (σ=10 S/m in the shock layer).

Figure 5.9: Results for the test case II, σ=10 S/m condition.

and a heat fluxes mitigation of about 10% in the stagnation zone [81].

The second test considers a tensor conductivity and a ionized free stream. The tensor nature of the conductivity has a dampening effect on the Faraday current sustaining the MHD interaction. Just looking at the Faraday current distribution in figure 5.17a it can be easily noticed that its module is lower than in the scalar conductivity case. It can also be noted that a non-zero value of conductivity in the free stream affect the current distribution that assumes considerable values also outside the shock, as shown in figure 5.17b. The Hall currents' magnitude is highly dependent on the electron mobility. In the simulation presented here, the free stream mobility has been assumed constant and equal to 29 T^{-1}, and has been estimated as inversely proportional to the mass density inside the shock. The approximate nature of this assumption affects the current densities evaluation. A more accurate value for this quantity has to be obtained by kinetics models, still under development within the CAST collaboration.

The effect on the flow field of Hall currents is evident if we compare figure 5.15 with figure 5.17: the reduction of the shock displacement effect is apparent. The effect on wall heat fluxes and wall pressure, shown in figure 5.18, are also less significant with respect to the previous case.



(a) Pressure (σ=100 S/m in the shock layer).

(b) Current density (σ=100 S/m in the shock layer).

Figure 5.10: Results for the test case II, σ=100 S/m condition.

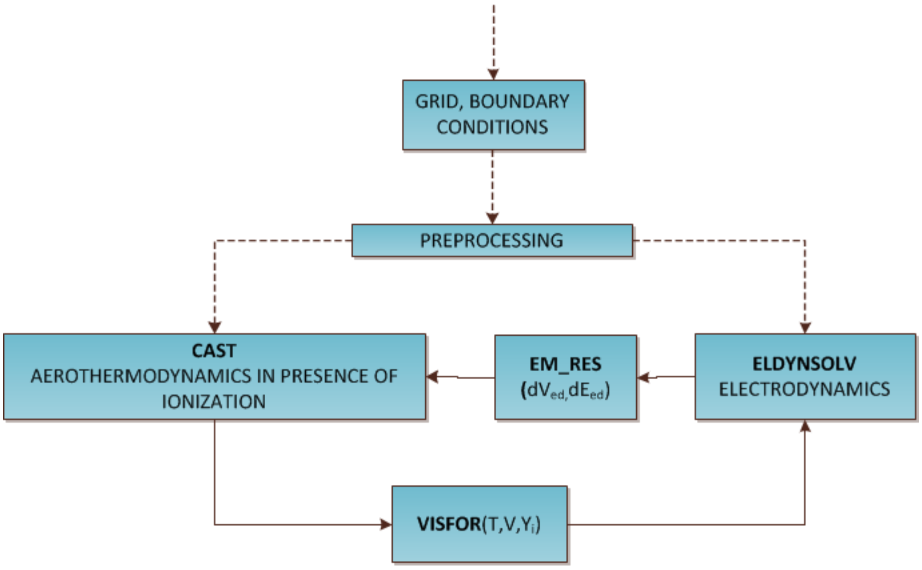
5.2 Three dimensional electrodynamics.

This section overviews the preliminary testing phase of the three dimensional electrodynamic code. The code has been run in two simple geometric configurations, by imposing a uniform flow field and an uniform magnetic flux density. The values of electron density and electron charge, have been assumed equal to one, as well as the modules of the flow velocity and of the magnetic flux density.

The tetrahedral meshes and the post processing views for the simulations have been obtained through the open source code *Gmesh* [82].

The first test case is defined over the unitary cube shown in figure 5.19 where

Figure 5.11: Coupling schema for the electrodynamic code.



the imposed boundary conditions are also depicted. The tetrahedral mesh is shown in figure 5.21a. The boundary conditions have been chosen in order to obtain a constant field along the y direction. The only non null current density component allowed by the imposed conditions is along the y direction. Its module is defined by the null x field requirement. The assumed conditions are:

$$\mathbf{B} = (0, 0, B_0) \tag{5.2}$$

$$\boldsymbol{\beta} = (0, 0, \beta_0) \tag{5.3}$$

$$\mathbf{u} = (0, u_0, 0), \tag{5.4}$$

$$\tag{5.5}$$

the electrodynamic equation is therefore:

$$\mathbf{J} = \frac{\sigma}{1+\beta_0^2} \begin{pmatrix} 1 & -\beta_0 & 0 \\ \beta_0 & 1 & 0 \\ 0 & 0 & 1+\beta_0^2 \end{pmatrix} \begin{pmatrix} E_x + u_0 B_0 \\ E_y \\ E_z \end{pmatrix} \tag{5.6}$$

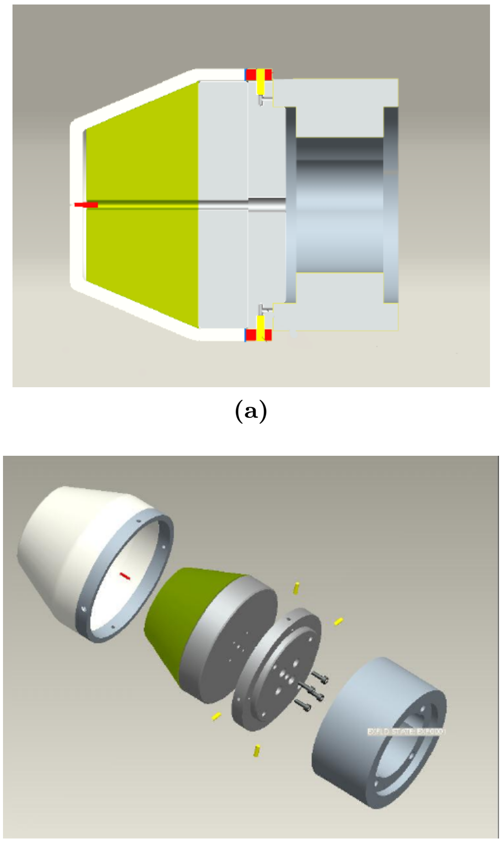
The first two equations are therefore:

$$J_x = \frac{\sigma}{1+\beta_0^2}(E_x + u_0 B_0 - \beta E_y) \tag{5.7}$$

$$J_y = \frac{\sigma}{1+\beta_0^2}(\beta E_x + \beta u_0 B_0 + E_y) \tag{5.8}$$

$$\tag{5.9}$$

. . . .



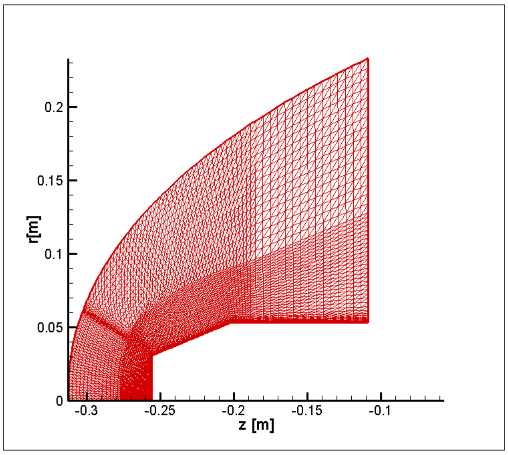
(a)

(b)

Figure 5.12: Test body configuration.

The $J_y = 0$ together with the $E_x = 0$ request are satisfied by imposing the J_x module equal to $\sigma u_0 B_0$, as shown in figure 5.19. Equations 5.7 give therefore an expected electric field in the y direction $E_y = -\beta u_0 B_0$, obtained in the simulation for unitary velocity and magnetic flux density modules, as shown in figure 5.21b. Figure 5.22 shows the analogous results obtained for a cylindrical ring section of lenght 4 m, constant flow field, and a radial magnetic field $\mathbf{B} = (B_0/r)\hat{e}_r$, with boundary conditions given in figure 5.20.

Figure 5.13: Triangular structured mesh for the electrodynamic computations.



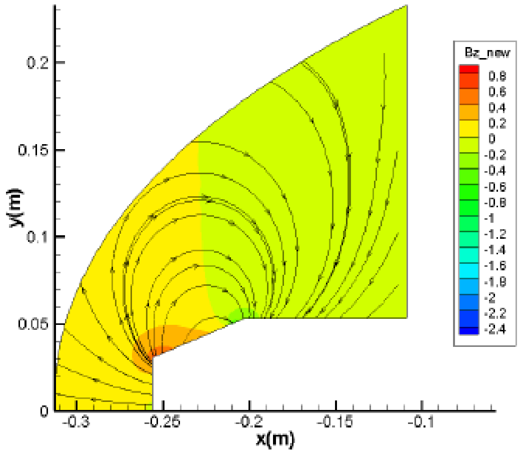


Figure 5.14: Interpolation of the magnetic field on the computational domain.

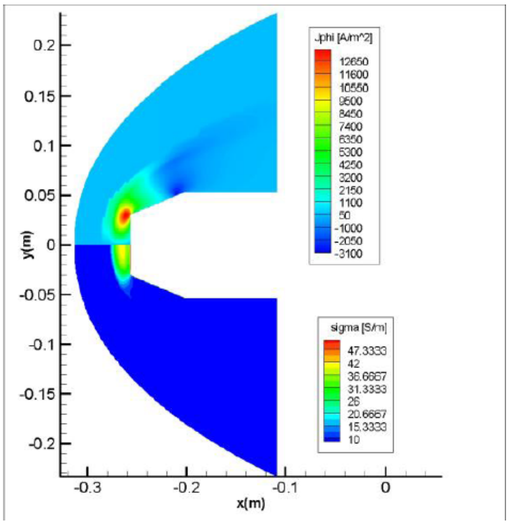
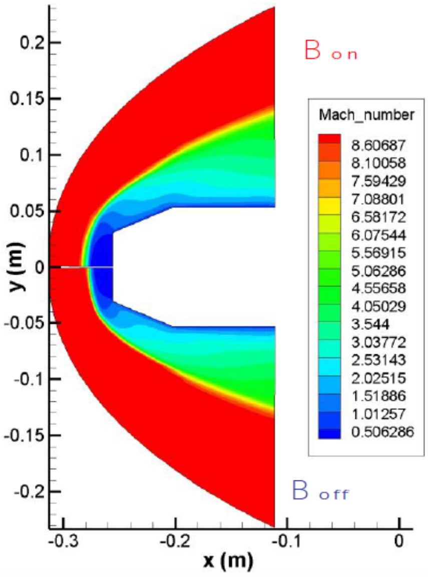
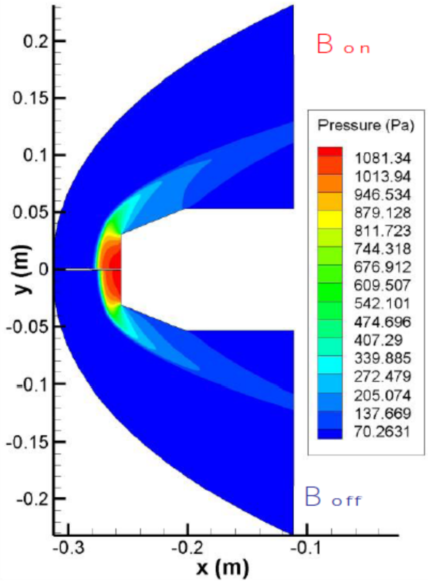


Figure 5.15: Sigma and Faraday current distribution in the scalar conductivity case.

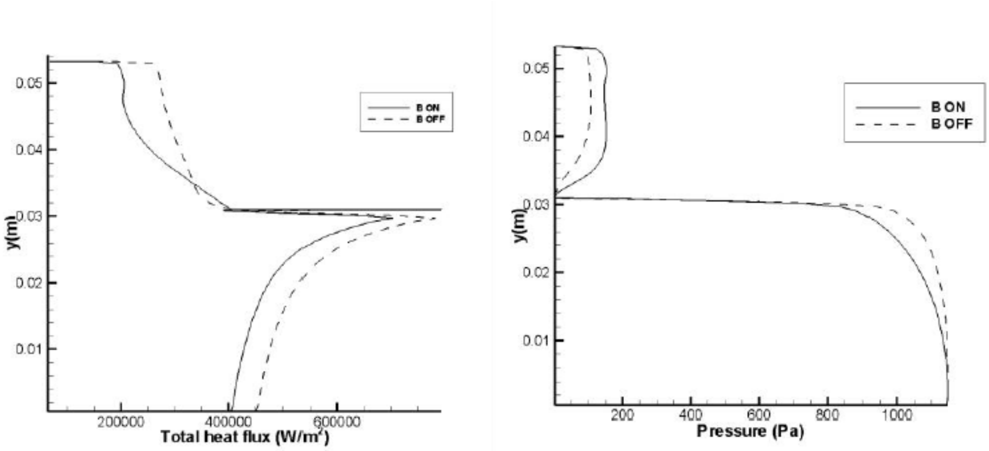


(a) Mach number



(b) Pressure

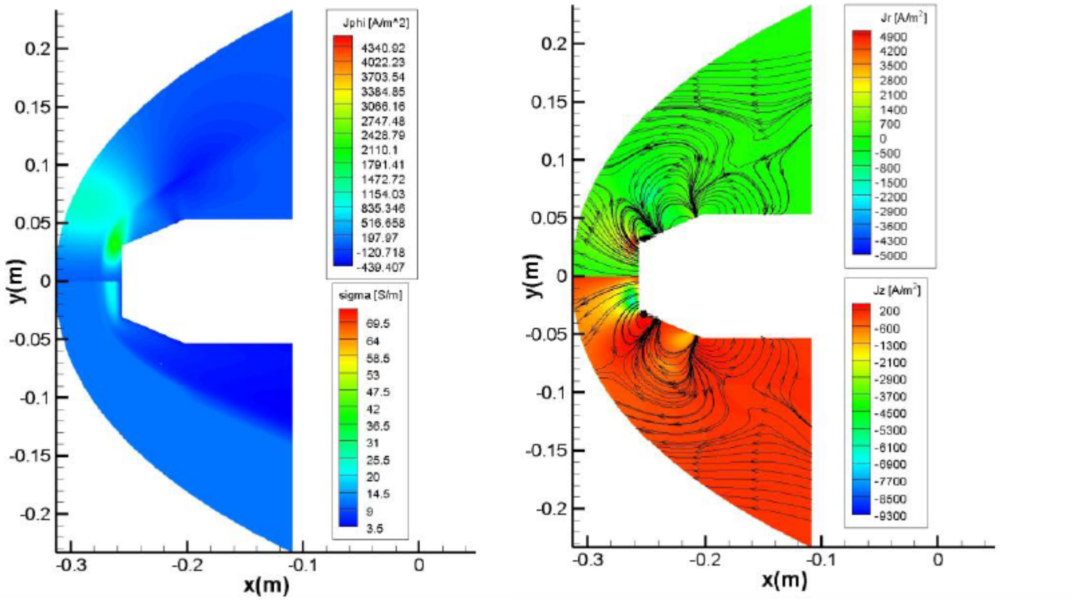
Figure 5.15: Differences in terms of Mach (a) and pressure (b) distributions between magnetic field off and magnetic field on cases (scalar sigma).



(a) Wall heat fluxes

(b) Wall pressure

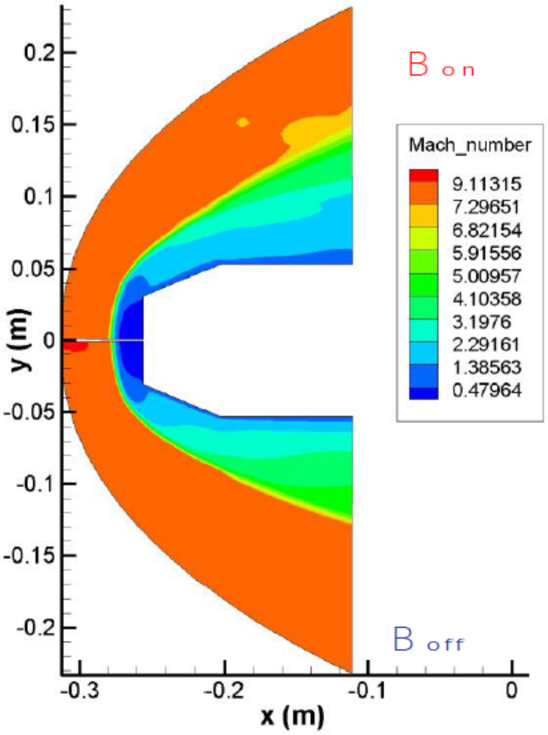
Figure 5.16: Differences in terms of wall heat fluxes (a) and wall pressure (b) distributions between magnetic field off and magnetic field on cases (scalar sigma).



(a) Mach number

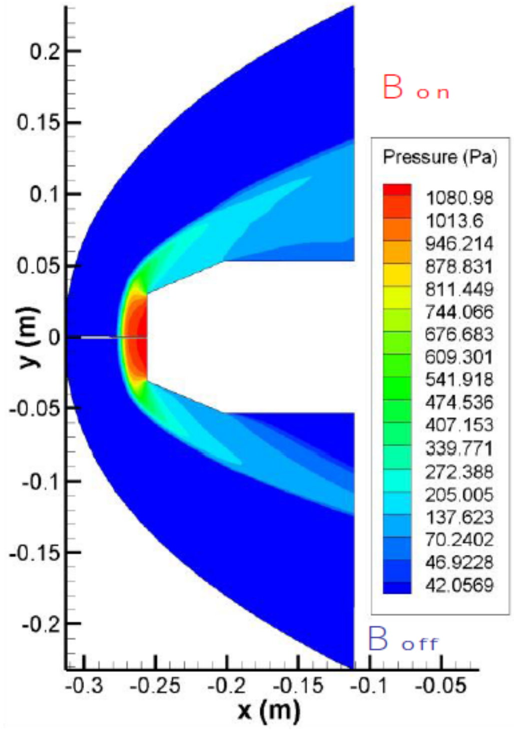
(b) Pressure

Figure 5.17: Sigma and Faraday current distribution (a), Hall currents components and streamlines (b).



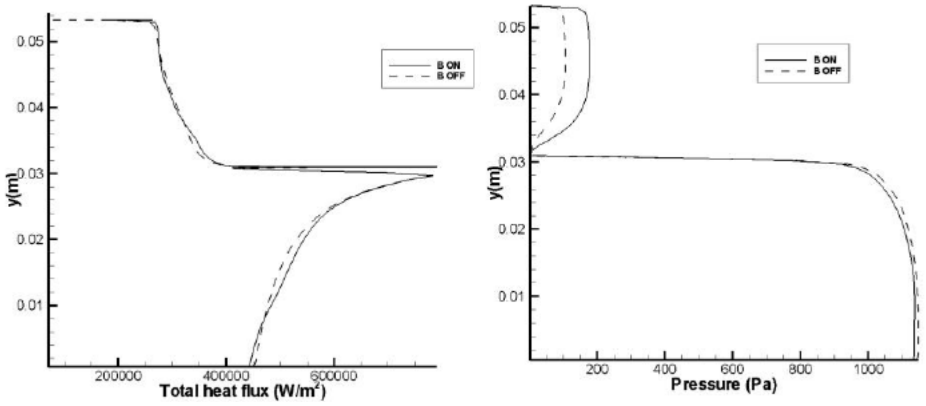
(a) Mach number

. . . .



(b) Pressure

Figure 5.17: Differences in terms of Mach (a) and pressure (b) distributions between magnetic field off and magnetic field on cases (tensor sigma).



(a) Wall heat fluxes **(b)** Wall pressure

Figure 5.18: Differences in terms of wall heat fluxes (a) and wall pressure (b) distributions between magnetic field off and magnetic field on cases (tensor sigma).

. . . .

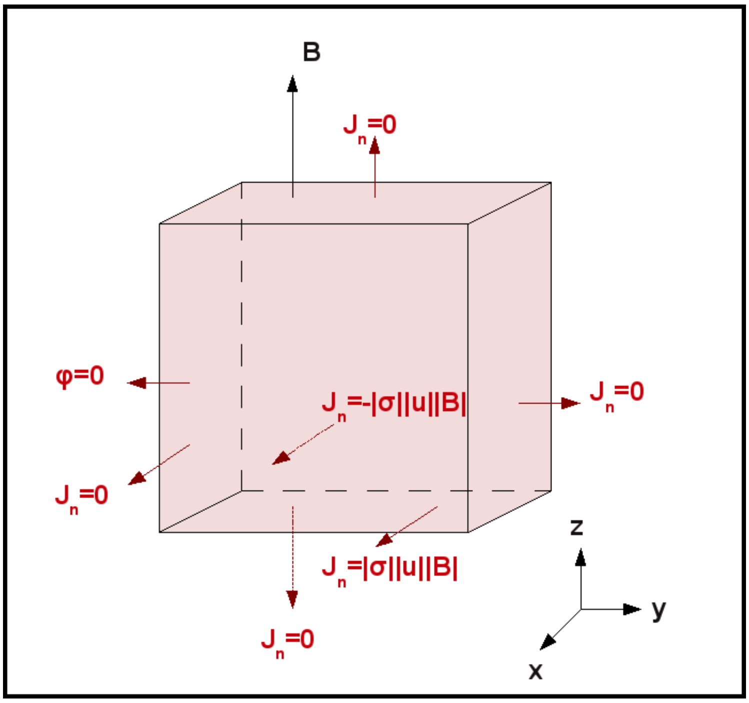


Figure 5.19: Boundary conditions for the test case described in text, cubic domain.

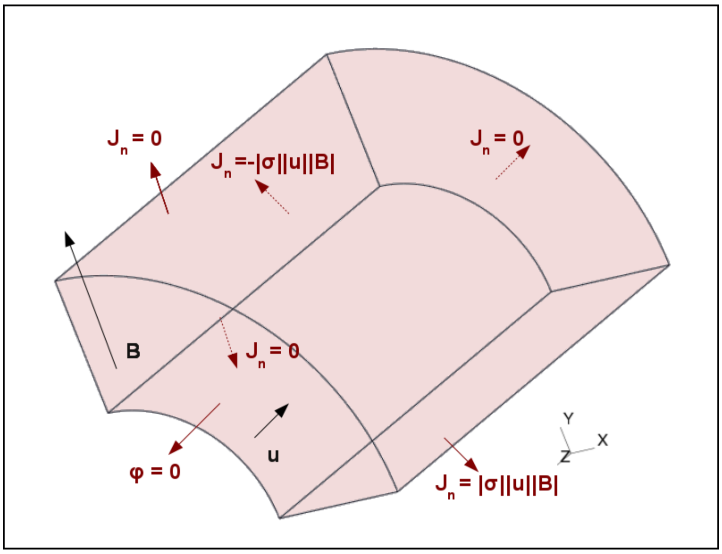
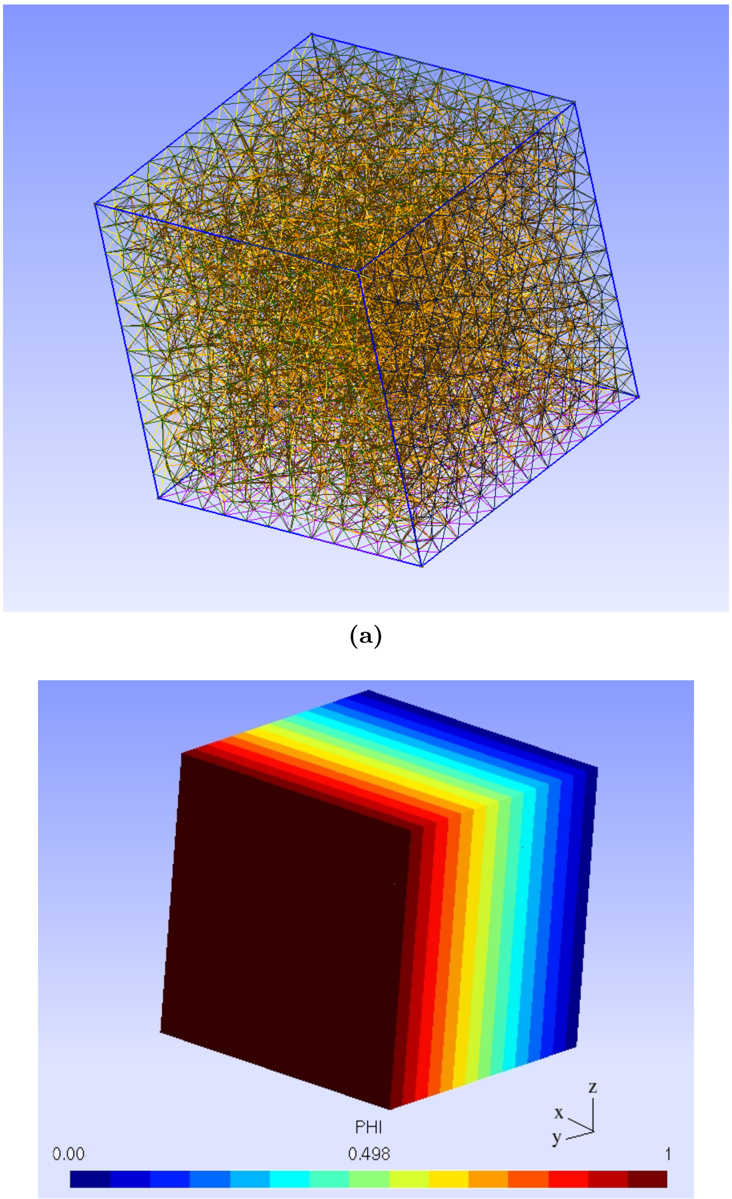


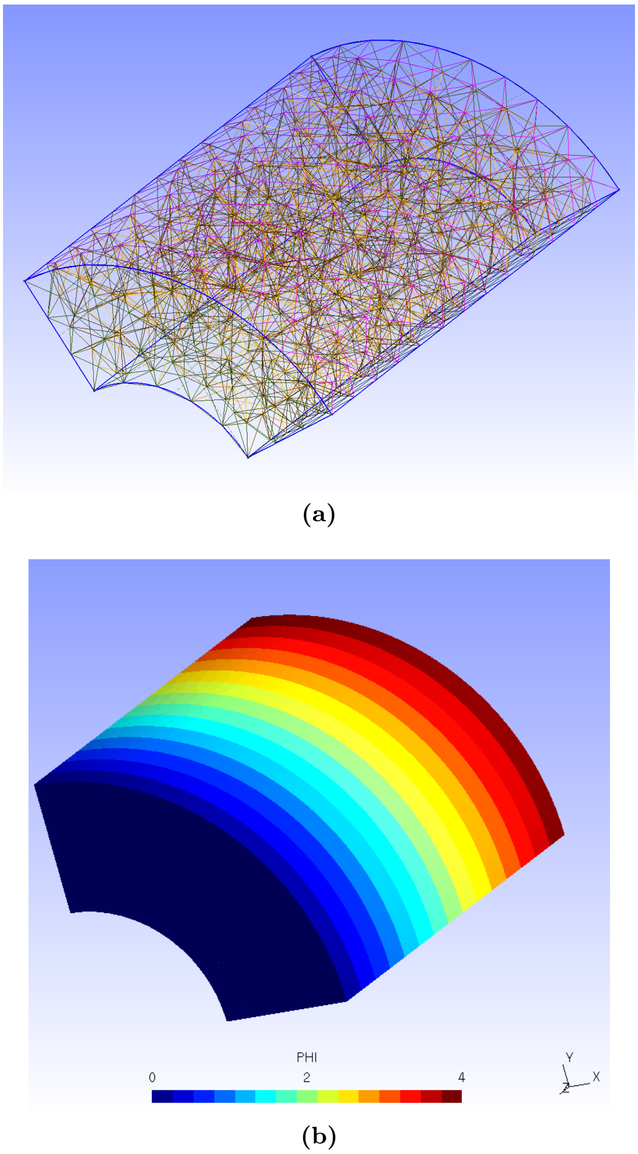
Figure 5.20: Boundary conditions for the test case described in text, cylindrical ring section.



(a)

(b)

Figure 5.21: Tetrahedral mesh (a) and Hall potential (b) for the cubic domain.



(a)

(b)

Figure 5.22: Tetrahedral mesh (a) and Hall potential (b) for the cylindrical ring section.

CHAPTER 6

CONCLUSIONS

Anyone who tells a lie has not a pure heart, and cannot make a good soup.

Ludwig van Beethoven

A research should provide answers. Or, at least, should improve the questions, and find out better strategies to get the answers.

The question the work presented in this thesis should answer is about the possibility to apply the MHD interaction to control the flux around an hypersonic re-entry vehicle. Or, more realistically, should supply a computational tool that, in conjunction with the experimental data, could clarify the feasibility and effectiveness of this idea.

In order to make possible such an evaluation, it has been necessary to overcome some computational issues linked to the anisotropic physical nature of the problem. It has been done, and some results have been obtained.

From the numerical point of view, the solver reaches convergence and has time costs much smaller than fluid-dynamics computation times. The technique developed to improve the solver convergence behavior, reduces the convergence time and solves the convergence difficulties. The implementation of the Superconvergent Patch Recovery Technique gives higher accuracy in the fields evaluation, when required.

The results say that, in the scaled test body configuration, for an electron number density of about $10^{19} - 10^{20}$ and a conductivity of the order of ten Siemens the interaction gives a modification of the heat fluxes of about 10%, but the results also highlight the dampening effect the Hall diffusion has on the MHD effectiveness.

The model assumes to have as an input a reliable estimate of the electron density and of the conductivity, in the interaction region, and further developments are necessary to fulfill this requirement. The tridimensional extension of the model has been completed, and it needs the fluid dynamics coupling to give its answers, or, at least, to begin asking its questions.

Appendices

APPENDIX A

ACRONYMS

FEM Finite Element Method

GMRes Generalized Minimum Residual (method)

ILU Incomplete Lower Upper (factorization)

ILUT Incomplete Lower Upper (factorization) with Threshold

LU Lower Upper (factorization)

MHD MagnetoHydroDynamics

PDE Partial Differential Equation

Rem Magnetic Reynolds Number

RCM Reverse Cuthill McKee

SPR Superconvergent Patch Recovery

WRCM Weighted Reverse Cuthill McKee

BIBLIOGRAPHY

[1] F. Curran, J. Hunt, N. Lovell, and G. Maggio. he benefits of hypersonic airbreathing launch systems for access to space. In *Proceedings of the 39th AIAA/ASME/SAE/ASEE Joint Propulsion Conference*. 2

[2] Charles R. McClinton, Vincent L. Rauscha, Robert J. Shawb, Unmeel Methac, and Chris Nafteld. Hyper-x: Foundation for future hypersonic launch vehicles. *Acta Astronautica*, 57(2-8):614–622, July-October 2005. 2

[3] Charles R. McClinton, Vincent L. Rausch, Luat T. Nguyen, and Joel R. Sitz. Preliminary x-43 flight test results. *Acta Astronautica*, 57(2-8):266 – 276, 2005. 2

[4] Shang J.S. Recent research in magneto-aerodynamics. *Progress in Aerospace Sciences*, 37:1–20, 2001. 2

[5] Shang J.S. et al. Magneto-aerodynamic interactions in weakly ionized hypersonic flow. In *Proceedings of the 40th Aerospace Sciences Meeting and Exibit*, 2002. 2

[6] V.A. Bityurin et al. Assessment of hypersonic mhd concepts. In *Proceedings of 28th Plasmadynamics and Lasers Conference*, 1997. 2

[7] V.A Bityurin, A.N.Bocharov, and J.T. Lineberry. Mhd aerospace applications. In *Proceedings of 13th International Conference on MHD Power Generation and High Temperature Technologies*, volume III, pages 793–814, Beijing, PRC, October 1999. 2

[8] S. O . Macheret et al. Mhd power extraction from cold hypersonic air-flow with external ionization. In *Proceedings of the 30th Plasmadynamics and Lasers Conference*, 1999. 2

[9] D.J. Moorhouse and C.F. Suchomel. Exergy methods applied to hypersonic vehicle challenge. In *Proceedings of 32sd Plasmadynamics and Lasers Conference*, 2001. 2

[10] R.J. Litchford, U.B. Metha, and J.T. Lineberry. Magnetohydrodynamic augmented hypersonic air-breathing propulsion. In *Space Transportation Day*, 2000. 2

[11] D.M. Van-Wie et al. Numerical investigation of mgd interaction in non-equilibrium plasma flow in the models of supersonic intake. In *Proceedings of the 3rd Workshop on Magneto-Plasma-Aerodynamics in Aerospace Applications*, 2001. 2

[12] R.K. Agarwal. Numerical simulation of compressible viscous mhd flows to study the possibility of reducing supersonic drag of blunt bodies and scramjet inlets. In *AFOSR Grant No.F49620-99-1-0005*, 1999. 2

[13] E.P. Gurianov and P.T. Harsha. Ajax: new directions in hypersonic technology. In *Proceedings of 27th Plasmadynamics and Lasers Conference*, 1996. 2

[14] A.L. Kuranov et al. Modern condition of development and perspective researches of hypersonic technologies under the 'ajax' concept. In *Proceedings of 2nd Workshop on Magneto-Plasma-Aerodynamics in Aerospace Applications*, 2000. 2

[15] V.A. Zeigarnik et al. Dimension optimization of mhd generators for hypersonic aircraft of 'ajax' concept. In *Proceedings of 2nd Workshop on Magneto-Plasma-Aerodynamics in Aerospace Applications*, 2000. 2

[16] J.T. Lineberry et al. Prospects for mhd flow control for hypersonics. In *Proceedings of 31st Plasmadynamics and Lasers Conference*, 2000. 2

[17] S.V. Bobashev et al. Influence of mhd interaction on shock-wave structures in supersonic diffuser. In *Proceedings of the 2nd Workshop on Magneto-Plasma-Aerodynamics in Aerospace Applications*, 2000. 2

[18] J.T. Lineberry and V.A. Bityurin et al. Cylinder with current in hypersonic flow. In *Proceedings of the 3rd Workshop on Magneto-Plasma-Aerodynamics in Aerospace Applications*, 2001. 2

[19] W.W. Vuillermont and U. Shumlack. Multi temperature, thermal and ion fraction effects over wedge and blunt body shapes in hypervelocity flow. In *Proceedings of ICOPS2000*, 2000. 2

[20] F. Scortecci, F. Paganucci, and L. d'Agostino. Experimental investigation of shock-wave/boundary-layer interactions over an artificially heated model in hypersonic flow. In *Proceedings of the 8th AIAA Int.*

Space Planes and Hypersonic Systems and Technologies Conference, Norfolk, VA, April 1998. 2

[21] V.A. Bityurin et al. Studies on mhd interaction in hypervelocity ionized air flow over aero-surfaces. In *Proceedings of 34th AIAA Plasmadynamics and Lasers Conference*, 2003. AIAA-2003-1365. 2

[22] V.A. Bityurin et al. Mhd activity in russia. In *LG-MHD International Workshop on Magneto-Hydro-Dynamics Applications: State of the Art*, 2003. 2

[23] A. Cristofolini et al. Experimental investigation on the mhd interaction around a sharp cone in an ionized argon flow. In *Proceedings of 37th AIAA Plasmadynamics and Lasers Conference*, 2006. AIAA-2006-3075. 2, 50

[24] C. A. Borghi, M. R. Carraro, A. Cristofolini, A. Veefkind, L. Biagioni, G. Fantoni, A. Passaro, M. Capitelli, and G. Colonna. Magnetohydrodynamics interaction in the shock layer of a wedge in a hypersonic flow. *IEEE Transaction on Plasma Science*, 2006. 2

[25] A. Cristofolini, C.A. Borghi, M.R. Carraro, G. Neretti, A. Passaro, G. Fantoni, and L. Biagioni. Hypersonic mhd interaction on a conical test body with a hall electrical connection. *Plasma Science, IEEE Transactions on*, 36(2):530 –541, April 2008. 2

[26] A. Cristofolini, C. A. Borghi, G. Neretti, V. M. Granciu, A. Passaro, D. Baccarella, and P. Caredda. Experimental investigation on the mhd interaction in a hypersonic flow around a blunt body. In *Proceedings of 38th Fluid Dynamics Conference and Exhibit*, 2008. AIAA 2008-4287. 2

[27] A. Cristofolini, C. A. Borghi, G. Neretti, V. M. Granciu, A. Passaro, D. Baccarella, and P.Caredda. Experimental investigation on the mhd interaction in a hypersonic flow around a blunt body. In *Proceedings of 40th AIAA Plasmadynamics and Lasers Conference*, 2009. AIAA 2009-3727. 2

[28] Andrea Cristofolini Carlo A. Borghi Gabriele Neretti Andrea Passaro Damiano Baccarella Antonio Schettino and Francesco Battista. Experimental activities on the mhd interaction in a hypersonic air flow around a blunt body. In *Proceedings of 41th AIAA Plasmadynamics and Lasers Conference*, 2010. AIAA 2010-4490. 2

[29] C.A. Borghi, M.R. Carraro, and Cristofolini A. Analysis of magnetoplasmadynamic interaction in the boundary layer of a hypersonic vehicle. *Journal of Spacecraft and Rockets*, 41(4), 2004. 2

. . . .

[30] C.A. Borghi, M.R. Carraro, and A. Cristofolini. An axi-simmetric hall configuration for mhd interaction in hypersonic flows. In *Proceedings of 36th AIAA Plasmadynamics and Lasers Conference*, 2005. AIAA-2005-4785. 2

[31] L.I. Sedov. *A course in Continuum Mechanics*. Walters-Nordhoff, 1981. 9

[32] P.A. Davidson. *An Introduction to Magnetohydrodynamics*. Cambridge Texts in Applied Mathematics, 2001. 9

[33] M. Mitchner and Charles H. Kruger. *Partially Ionized Gases*. John Wiley & Sons Inc, 1973. 9, 10, 12

[34] A. Ferrari. Fluidi e plasmi in astrofisica. Lecture Notes. 9

[35] G. Turchetti. Meccanica dei continui. Lecture Notes. 9

[36] C.A. Borghi, M.R. Carraro, and A. Cristofolini. Numerical modeling of mhd interaction in the boundary layer of hypersonic flows. *IEEE Transactions on Magnetics*, 39(3):1507–1510, 2003. 12

[37] A. Cristofolini, C. A. Borghi, and C. Latini. Numerical analysis of the mhd interaction in argon hypersonic flows. In *Proceedings of the 17th International Conference on MHD Energy Conversion*, September 2009. 12

[38] O. C. Zienkiewicz R. L. Taylor and J. Z. Zhu. *The Finite Element Method: its basis and fundamentals*. Elsevier Butterworth-Heinemann, 2005. 17, 52

[39] Owe Axelsson and Vincent A. Barker. *Finite element solution of boundary value problems: theory and computation*. Society for Industrial and Applied Mathematics, Philadelphia, PA, USA, 2001. 17

[40] C. Hirsh. *Numerical computation of internal and external flows. Vol. 1 : Fundamentals of numerical discretization*. Wiley, 1990. 17

[41] Daniel Swenson. Finite element concept. `www.mne.ksu.edu/~swenson/Courses/ME862/Lectures/lect04.pdf`. vii, 19

[42] J. R. Shewchuk. An introduction to the conjugate gradient method without the agonizing pain. Technical report, Pittsburgh, PA, USA, 1994. 28

[43] Magnus R. Hestenes and Eduard Stiefel. Methods of conjugate gradients for solving linear systems. *Journal of Research of the National Bureau of Standards*, 49:409–436, Dec 1952. 28

[44] Youcef Saad and Martin H. Schultz. Gmres: a generalized minimal residual algorithm for solving nonsymmetric linear systems. *SIAM J. Sci. Stat. Comput.*, 7(3):856–869, July 1986. 28, 30, 47

[45] Youcef Saad. *Iterative methods for sparse linear systems.* SIAM, Philadelphia, 2003. 28, 29, 30, 31, 34, 60

[46] A. R. HouseHolder. *The Theory of Matrices in Numerical Analysis.* Dover, 1975. 31

[47] S. Pohjolainen. Matrix algebra. `http://matpc41.ee.tut.fi/PNF:byName:/ML02/Matrixtes/Chapter2.pdf`. 31

[48] T. Young and M. Mohlenkamp. Introduction to numerical methods and matlab programming for engineers. `www.math.ohiou.edu/courses/math344/lecture12.pdf`. 31

[49] Michele Benzi. Preconditioning techniques for large linear systems: a survey. *J. Comput. Phys.*, 182:418–477, November 2002. 31

[50] Matthias Bollhöfer and Yousef Saad. ILUPACK - preconditioning software package. `http://www-public.tu-bs.de/~bolle/ilupack/`. 31

[51] Y. Saad. ILUT: a dual threshold incomplete ILUfactorization. *Numerical Linear Algebra with Applications*, 1:387–402, 1994. 31

[52] Alan George and Joseph W. Liu. *Computer Solution of Large Sparse Positive Definite Systems.* Prentice Hall Professional Technical Reference, 1981. 36, 38

[53] Norman E. Gibbs, Jr. Poole, William G., and Paul K. Stockmeyer. An algorithm for reducing the bandwidth and profile of a sparse matrix. *SIAM Journal on Numerical Analysis*, 13(2):236–250, 1976. 36, 39

[54] E. Cuthill and J. McKee. Reducing the bandwidth of sparse symmetric matrices. In *Proceedings of the 1969 24th national conference*, ACM '69, pages 157–172, New York, NY, USA, 1969. ACM. 37

[55] Wai-Hung Liu and Andrew H. Sherman. Comparative analysis of the cuthill-mckee and the reverse cuthill-mckee ordering algorithms for sparse matrices. *SIAM Journal on Numerical Analysis*, 13(2):198–213, 1976. 38

[56] John Alan George. *Computer implementation of the finite element method.* PhD thesis, Stanford, CA, USA, 1971. 38

[57] Robert Bridson and Wei-Pai Tang. A structural diagnosis of some ic orderings. *SIAM J. Sci. Comput.*, 22:1527–1532, May 2000. 38

. . . .

[58] S. Parter. The use of linear graphs in gauss elimination. *SIAM Review*, 3(2):119–130, 1961. 38, 39

[59] I. S. Duff and G. A. Meurant. The effect of ordering on preconditioned conjugate gradients. *BIT*, 29:635–657, December 1989. 38

[60] Laura C. Dutto. The effect of ordering on preconditioned gmres algorithm, for solving the compressible navier-stokes equations. *International Journal for Numerical Methods in Engineering*, 36(3), 1993. 38

[61] Lyle Pussell and S. Y. Trimble. Gram-schmidt orthogonalization by gauss elimination. *Am. Math. Monthly*, 98(6):544–549, 1991. 39

[62] Simon S. Clift and Wei-Pai Tang. Weighted graph based ordering techniques for preconditioned conjugate gradient methods. *BIT Numerical Mathematics*, 35(1):30–47, 1995. 41

[63] P. A. Forsyth E. F. D'Azevedo and Wei-Pai Tang. Towards a cost-effective ilu preconditioner with high level fill. *BIT Numerical Mathematics*, 32(3):442–463, 1992. 41

[64] J. A. Meijerink and H. A. van der Vorst. An iterative solution method for linear systems of which the coefficient matrix is a symmetric m-matrix. *Mathematics of Computation*, 31(137):pp. 148–162, 1977. 41

[65] Andrea Cristofolini, Chiara Latini, and Carlo Angelo Borghi. A weighted reverse cuthillâmckee procedure for finite element method algorithms to solve strongly anisotropic electrodynamic problems. *J. Appl. Phys.*, 109(033301), 2011. 43

[66] Leonard R. Herrmann. Interpretation of finite element procedure as stress error minimization procedure. *Journal of the Engineering Mechanics Division*, 98(5):1330–1336, September/October 1972. 51

[67] G.Szegö. *Orthgonal polynomials*, volume 23. Amer. Math. Soc. Colloquium Publications, 1975. 52

[68] Q.D. Zhu and Qun-Lin. *Superconvergence Theory of Finite Element Methods*. Hunan Science and Technology Press, Hunan,China, 1989. 52

[69] Zhimin Zhang. Ultraconvergence of the patch recovery technique. *Math. Comput.*, 65:1431–11437, 1996. 52

[70] Zhimin Zhang and J. Z. Zhu. Analysis of the superconvergent patch recovery technique and a posteriori error estimator in the finite element method (ii). *Computer Methods in Applied Mechanics and Engineering*, 163(1-4):159 – 170, 1998. 52

[71] Zhimin Zhang. Ultraconvergence of the patch recovery technique ii. *Math. Comput.*, 69:141–158, 2000. 52

[72] Qiding Zhu and Qinghua Zhao. Spr technique and finite element correction. *Numerische Mathematik*, 96:185–196, 2003. 52

[73] O. C. Zienkiewicz and J. Z. Zhu. Superconvergent patch recovery and a posteriori error estimates. part 1: the recovery technique. *International Journal for Numerical Methods in Engineering*, 33(7):1331–1364, 1992. cited By (since 1996) 709. 52

[74] O. C. Zienkiewicz and J. Z. Zhu. Superconvergence and the superconvergent patch recovery. *Finite Elements in Analysis and Design*, 19(1-2):11 – 23, 1995. 52, 53, 54

[75] I. Babuska, T. Strouboulis, C. S. Upadhyay, S. K. Gangaraj, and K. Copps. Validation of a posteriori error estimators by numericalin other words approach. *International Journal for Numerical Methods in Engineering*, 37(7):1073–1123, 1994. 52

[76] Intel fortran compiler optimizing applications. `http://software.intel.com/file/6397`. 58

[77] The cira homepage. `http://www.cira.it/html/inglese/home/`. 63

[78] Carl E. Baum. Electron thermalization and mobility in air. Technical Report XII, Air Force Weapons Laboratory, 1965. 64

[79] Andrea Cristofolini and Carlo Borghi. Cast-dt-30: Nota tecnica sull'implementazione di modelli per mfd (basso rem). Technical report, Centro Italiano Ricerca Aerospaziale (CIRA), 2009. 70

[80] Andrea Cristofolini and Francesco Battista. Alta-cast-dt-27: Dossier sul tc4a: Resoconto test. Technical report, Centro Italiano Ricerca Aerospaziale (CIRA), 2009. 70

[81] Andrea Cristofolini and Francesco Battista. Cast-dt-65: Rapport di integrazione modello per la magnetofluidodinamica. Technical report, Centro Italiano Ricerca Aerospaziale (CIRA), 2010. 72

[82] C. Geuzaine and J.-F. Remacle. Gmsh: a three-dimensional finite element mesh generator with built-in pre- and post-processing facilities. *International Journal for Numerical Methods in Engineering*, 79(11):1309–1331, 2009. 73

. . . .